An Algorithm for Identifying and Prioritizing non-Critical Path Schedule Delays[1]

Niall McShane

Abstract

An algorithm is presented which extends the critical path method with new metrics at the task and path level to identify and prioritize non-critical path delays and guide the project manager to intervene when such delays threaten the project completion date. This method does not rely on statistical simulation (e.g. Monte Carlo) techniques or require a-priori knowledge of potential risk and is responsive to early warning signs of delays within the project. It has particular applicability in projects where the scale and complexity of the project network precludes any individual understanding all of the detail in the project schedule.

Keywords: *Monitoring & Control; Project Networks; Risk, Uncertainty*

Introduction

Critical Path Method or CPM has been around for almost 70 years. Starting in late 1956, the Integrated Engineering Control Group at DuPont, with support from Remington Rand Corporation, began to research the possibility of using newly emerging computer processing capabilities to overcome challenges in traditional project planning and scheduling procedures for large scale construction projects. The approach used activity on arrow diagrams to map out dependencies between tasks and attached durations and costs to the activities to allow schedule and cost analysis to be performed. The longest path through the network, the Critical Path, could be calculated and the cost of alternative schedules, achieved by crashing task durations to improve schedule or allowing non-critical tasks to float to level resources, could easily be determined. The results of this work were published by Kelley and Walker (1959). Concurrently, the US Navy was working with Booz, Allen & Hamilton on a similar problem related to research and development projects associated with the Polaris Fleet Ballistic Missile program (Malcolm et al, 1959). Where duPont's emphasis was on managing cost and schedule for deterministic schedules, the Navy's concern was with the unpredictability that was inherent in research and development projects.

[1] How to cite this paper: McShane, N. (2026). An Algorithm for Identifying and Prioritizing non-Critical Path Schedule Delays, *PM World Journal*, Vol. XV, Issue III, March.

Consequently, while the Navy's Program Evaluation Review Technique or PERT shared a lot in common with CPM, including the use of activity on arrow diagrams and the concept of a critical path, PERT used three point estimates to compute an expected duration for each task and attempted to predict a probabilistic critical path but lacked a cost model or the idea of analyzing cost-schedule tradeoffs. Over time, these competing methods converged, adopting the best practices of each method, especially as software tools emerged to support project scheduling. This hybrid critical path method quickly spread to construction and other industries and became a de-facto standard in the project manager's toolkit. However, as early as the 1960's people were already recognizing that both CPM and PERT inadequately addressed the risk of schedule delays; in particular delays to non-critical path activities.

A key development in the early 1960's was the introduction of Monte Carlo simulation (Van Slyke, 1963) which extended the schedule analysis to include a large number of simulations of the project network using task durations that are derived randomly from the range of possible durations estimated for each task. A key output of this type of schedule analysis was a criticality index which reflects the probability that a particular task will become critical during project execution. This helped to expose areas of the schedule which were non-critical in the deterministic view but which had high risk of becoming critical if some combination of tasks in those paths took longer than their expected duration.

Further advances in trying to manage the inherent risk of variability in schedule durations such as cruciality (Williams,1992) and schedule sensitivity (Elmaghraby, 2000) attempted to quantify not only the probability that certain tasks or paths could become critical, but also the anticipated impact that such developments would have on the overall project schedule. Another key development in the late 1960's was the US Federal Government's cost/schedule control systems criteria which evolved into a more flexible approach of earned value analysis, EVA or earned value management, EVM (Anbari, 2003) which is used to track variances in the cost and schedule performance of the project.

A more comprehensive critique of critical path based approaches was offered by Goldratt (1997) who argued that CPM is inherently flawed because it focuses on managing tasks instead of managing uncertainty, resources and human behavior. His solution proposes to cut task estimates to a minimum, replace total float and free float with project buffers and feeding buffers, eliminate multi-tasking and define a critical chain which considers both scheduling and resource constraints. Management of the project proceeds by managing resource contention and tracking the rate of buffer consumption. This approach requires significant cultural change within an adopting organization and there is some controversy over whether this is truly a revolution in project management or a management fad that is difficult to implement in practice (Herroelen et al, 2002 and Raz et al, 2003).

While these approaches have helped to mitigate risk, they all require a level of insight or skill to define the degree of uncertainty in a project's tasks or a level of cultural change and trust between workers and management that many project based organizations cannot achieve. Even if these skill levels or cultural norms can be achieved, there may still be truly unexpected events which the planning team failed to recognize. As project size and complexity increase, it becomes increasingly difficult for these methods to adequately address the risk.

Context

Many years ago, while working for a large US based corporation, I had a role which involved extensive work bench marking best practices within the corporation. One division that I bench marked was involved in a multi-disciplinary mega project which involved thousands of people. They had developed project scheduling to an art form and were more than happy to share their expertise which included the use of Monte Carlo simulation, Earned Value Management and milestone protection buffers among other best practices. They also employed the practice of structured planning workshops and layered schedules that allowed them to manage the complexity of a project involving tens of thousands of individual tasks and numerous subprojects managed independently by separate project teams. The structured planning workshops produced layers of schedules that were vertically integrated by dependencies showing deliverables to and from each subproject. Figure 1 below shows an illustrative example of a top level schedule comprising 4 subprojects with deliverables; D1 through D7 being produced and consumed among those various subprojects. This defines an agreed interface between the subprojects that the project managers of each subproject commit to.

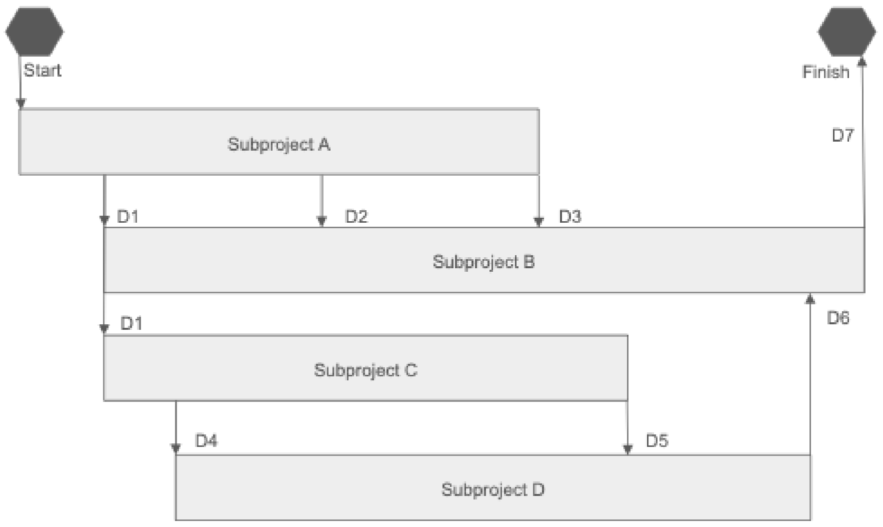


FIGURE 1: TOP LEVEL SCHEDULE

Each subproject at one layer was then broken down into the sub-tasks that consumed the inputs to that task and produced its outputs (see Figure 2 below). There might have been up to ten or more layers in the project with increasing detail as you descended the layers until, at the lowest layer, no task was scheduled for more than a week. At the lowest level of the schedule, individual critical path networks were used to manage the lowest level details and customized software was used to integrate the schedule dependencies between the different layers and the different subprojects so that an overall critical path through the entire schedule could be calculated.

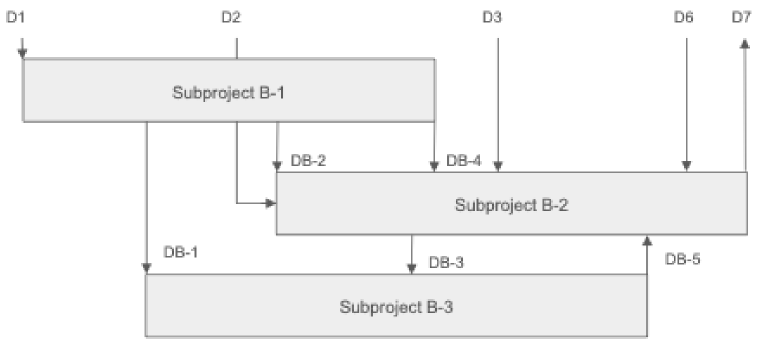


FIGURE 2: SUBPROJECT B

This level of detail was, of course, incomprehensible by any individual and so project reporting was done by rolling up the status of the project to higher levels. Senior managers and executives would only ever see the top 2 or 3 layers of the schedule and would manage the project by the trends that they saw there. Mid-level managers might be looking at 3 or 4 layers of intermediate detail reflecting their specific area of responsibility while individual project managers would manage the lowest layers of a relatively small section of the overall project. Despite the obvious success that this team was having, the result was like the famous parable of the blind men and the elephant - none of the folks who could see the day-to-day detail of what was happening with individual tasks could picture the overall status of the project while those who were responsible for the overall schedule, or even a significant portion of it, did not have visibility to the day-to-day challenges arising within all areas of the project. Imagine the challenge of managing non-critical path schedule delays in such an environment.

While implementing the same sort of structured planning and integrated schedule practices in other parts of the organization, I developed the ideas that will be described in this paper. At the time, the scale of the projects that we were implementing were much smaller than those that the benchmarked division had been doing and so these ideas sat and gathered dust for a long time but I recently came across some papers relating to this and decided that it was time to bring them forward and share them with the profession. In the intervening years, most of the software world where I spent my career has moved away from critical path scheduling to an agile approach in which a backlog of work is burned down over multiple fixed duration iterations and progress is measured by the burndown rate of the backlog. However, critical path scheduling still has a role to play in other disciplines and I believe that these ideas have the potential to add significant new tools to the portfolio of project managers in those fields.

Problem Statement

Considering the context described above, and the general problem of how to manage non-critical path schedule delays in large or complex projects, I will use an example to illustrate the limitations of existing critical path scheduling methods and the risk of non-critical path schedule delays (See Figure 3 below).

Suppose a project has a subproject that employs a process that is executed on a serial production line. The process is scheduled to take 1 day per unit to complete and it has to be executed on 50 consecutive units. The production of these units is scheduled to take 50 days. There is an option to add a second production line but that involves significant cost and, since the overall task has 20 days of float, the project manager decided during the planning phase to retain only a single production line. These units are one of several, unrelated deliverables that the project manager is responsible for providing to the overall project.

The project proceeds without delay until it reaches the point at which the units have to go through this process. When the first units reach this step, it takes 2 days instead of 1. At the weekly project meeting, with only 2 units completed, the project manager notes that the aggregate task has slipped by 3 days and the project team explains the issue and says that they are working on a solution to adjust the process in an effort to reduce the time required back to its original 1 day estimate per unit. The project manager sees that there are still 17 days of float on the task. She is confident that the team will be able to make the necessary adjustments to hold the schedule delay within its available float and approves the team's plan to pursue changes to the process.

At the next weekly project meeting, only one additional unit has been completed and the overall task is showing a delay of 10 days. In addition to the extra day lost due to the process taking more time, 3 days were lost when the line had to be stopped and reworked to implement changes that the team hoped would recover the schedule. The team has also added another 3-day planned shutdown in the coming week to make further process changes in the belief that they can still recover the original yield rate. At this point, half of the available float has been consumed but there are still 10 days of float remaining and the project manager is focused on a delay that is impacting the critical path for one of her other deliverables. The team remains confident that just one more tweak to their process will yield the necessary improvement to achieve the 1 unit per day performance that they had estimated and no further action is assigned.

Another week passes and despite 3 more days of downtime to apply changes to the process, the yield remains stubbornly at 1 unit every 2 days. The team worked overtime to complete 2 additional units but only 5 units have been produced with 15 days elapsed. In the project meeting the team reports that they have finally understood the root cause of the problem and it is not solvable with their current capabilities. They have identified a potential solution but that would require a much more significant retooling of the process which would take months to develop. They propose that the project should accept the current process and start up a second production line to recover the schedule at significant cost to the project. It will take 4 weeks to bring the second production line to an operational state but the team has calculated that they will produce 10 additional units during that period using the first line and they will still have 35 days to run on two lines allowing them to complete the remaining 35 units within the originally scheduled float for this task. This effectively reduces the total float on this activity to zero.

Reluctantly the project manager agrees to the plan of action proposed by the team and signs off on the additional cost to bring up a second production line. Assuming the line can be brought up in 4 weeks they should be able to hold the schedule for completing all 50 units. She communicates this update to the overall project. Significantly, this is the first time that the delay has visibility outside of the subproject team.

Another week goes by and even with overtime yielding a further 3 units, the project team reports that their proposed mitigation plan will not work. All available production lines are currently busy and there is no opportunity to bring up another line for at least 60 days. At this stage, the project is out of options. The process cannot be improved in time to accelerate the production of the remaining units and additional production capacity is not available either. The project manager orders a change to the schedule to show the remaining 42 units taking 2 days each and even with overtime increasing the yield to 3 per week, the remaining duration is projected at 14 weeks (70 days). The task is now showing a 40 day delay from its original plan and negative 20 days of float to the overall project schedule. This sudden and dramatic shift triggers a chain reaction up through the higher layers of the schedule and the project manager learns that this is now the critical path for the entire project. Furthermore, because of sequencing issues in other areas of the project which have pushed dependent tasks out of the windows in which expensive resources have been scheduled to be available, it has added significant additional cost as well as compounding delays in downstream tasks.

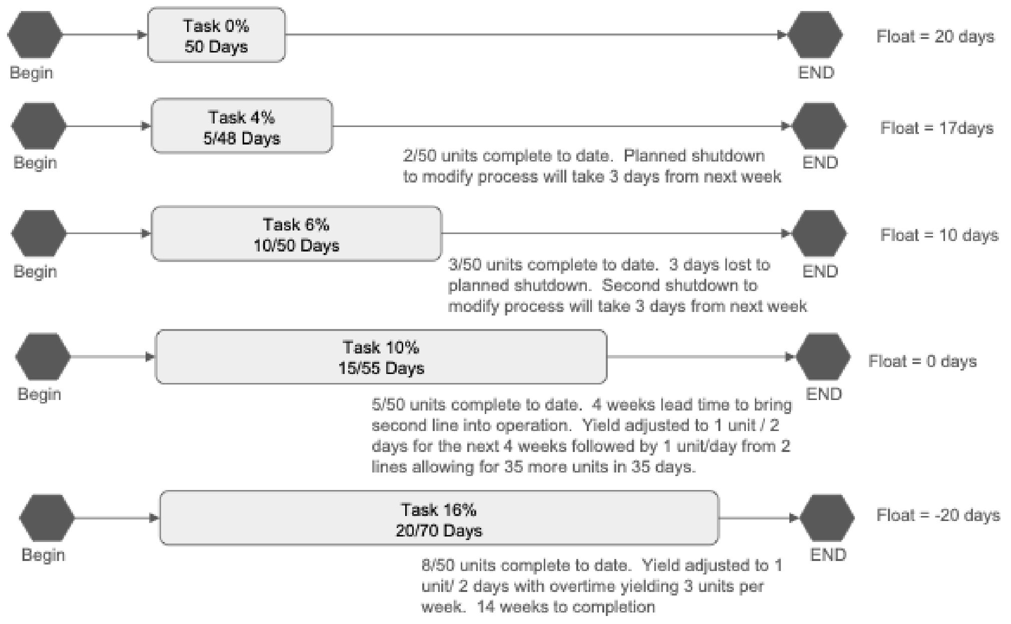

Task 0%
50 Days
Begin
END
Float = 20 days
Task 4%
5/48 Days
2/50 units complete to date. Planned shutdown to modify process will take 3 days from next week
Float = 17days
Task 6%
10/50 Days
3/50 units complete to date. 3 days lost to planned shutdown. Second shutdown to modify process will take 3 days from next week
Float = 10 days
Task 10%
15/55 Days
5/50 units complete to date. 4 weeks lead time to bring second line into operation. Yield adjusted to 1 unit / 2 days for the next 4 weeks followed by 1 unit/day from 2 lines allowing for 35 more units in 35 days.
Float = 0 days
Task 16%
20/70 Days
8/50 units complete to date. Yield adjusted to 1 unit/ 2 days with overtime yielding 3 units per week. 14 weeks to completion
Float = -20 days

Figure 3: Example of Non-Critical Path Schedule Delay

The higher level project is completely blindsided by this development because it appeared on their radar only a week ago when the production of these units first became critical and the lower level project manager had reported a seemingly reasonable plan to contain the issue by bringing a second line on. This looks like a crisis that has come out of nowhere even though it has been brewing for four weeks, hidden for most of that time by the available float in the original schedule. The lower level project manager didn't necessarily do anything wrong. She was managing within her float and focusing on other deliverables which were on her critical path. She didn't have visibility to the potential downstream effects of a major delay in this non-critical path within her scope and the people managing those downstream tasks didn't have any advanced warning that there was risk to these deliverables. Monte Carlo simulations and risk based scheduling would not necessarily have identified these issues because, to do so would require some a-priori knowledge of the potential risk and that knowledge was either not available or had been discounted during the initial planning.

Method

Let us now look at an algorithm that could help to identify issues with non-critical path schedule delays and bring earlier attention to them. Consider the sample CPM network illustrated in Figure 4 below:

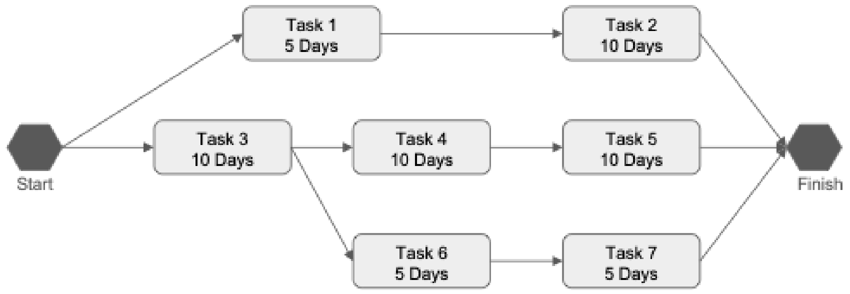


FIGURE 4: SAMPLE CPM NETWORK

This figure shows a network having seven tasks and two milestones (start and finish). There are three paths through the network from start to finish; Paths 1-2, 3-4-5, and 3-6-7. CPM determines the critical path through this network by traversing each path in turn and summing the durations of each activity. Path 1-2 has a duration of 15 days, path 3-4-5 has a duration of 30 days and path 3-6-7 has a duration of 20 days. Thus, the critical path is path 3-4-5 and the critical path duration is 30 days.

Float is defined as the amount of time a task can be delayed without impacting the project. There are two types of float, free float which is the amount of delay which can be accommodated without impacting the start of another activity and total float which is the amount of delay which can be accommodated without impacting the project end date. In the network shown in Figure 4, there is a total float of 15 days on path 1-2 and 10 days on path 3-6-7. Considering free float, Task 1 has no free float but Task 2 has 15 days of free float. Similarly, Tasks 3 and 6 have no free float but Task 7 has 10 days.

When tracking a project against plan, a project manager must pay particular attention to the tasks on the critical path. There is no float on the critical path and any delay to these activities will cause a delay to the project. Delays to non-critical tasks, however, will not cause a delay to the project until all of the available float has been used.

As we have seen, the problem which a project manager faces is that the early tasks on non-critical paths through the project network may use up all of the available float for that path, causing the later tasks to become critical and increasing the overall risk of the project being delayed. In a simple project network, it is possible for the project manager to understand the implications of any delay to non-critical tasks and initiate appropriate corrective actions. In very complex networks however, with hundreds or even thousands of tasks and paths, and potentially multiple separately managed subprojects, this is not possible. In such circumstances, it is difficult for the project manager to concentrate adequately on anything other than the critical path and perhaps a few other near-critical paths or paths that have been flagged as high risk by Monte Carlo simulations etc.

The method proposed is an extension to existing CPM algorithms. It adds new metrics at the task and path level which automate the analysis of non-critical schedule delays.

The first metric which is defined is a measure of schedule progress called Duration Performance Index (DPI). Unlike the schedule performance index defined in Earned Value Management, DPI is concerned only with durations – not with costs. DPI is defined as follows:

$$DPI = \textit{Planned Task Duration} / \textit{Actual Task Duration}$$

Although actual activity duration is only known when an activity is complete, DPI may be calculated for work in progress as:

$$DPI = (\textit{Planned Task Duration} / \textit{Actual Task Duration to Date}) * \%\ \textit{complete}$$

Values of DPI > 1 indicate tasks which are ahead of schedule. DPI < 1 indicates tasks which are behind schedule.

The second metric defined by this method is at the path level. This is the threshold DPI (tDPI) which is defined as follows:

$$tDPI = \textit{Planned Path Duration} / \textit{Critical Path Duration}$$

tDPI apportions the available float across all tasks in a path. If the value of DPI for the current task in a given path is less than the tDPI for the path, that task has been delayed to the point where it is consuming the available float for later tasks in the path.

As an example, consider Figure 5 below:

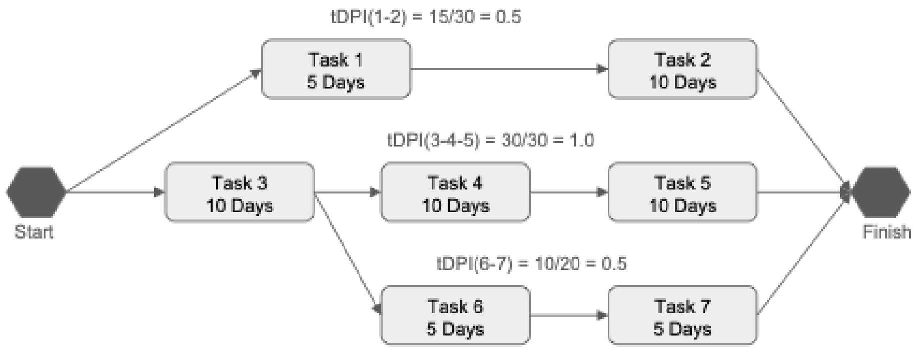


FIGURE 5: SAMPLE CPM NETWORK WITH tDPI

This shows the values of tDPI for the network in Figure 4. Clearly tDPI for the critical path is 1.0 since no delay can be accommodated on this path. For path 1-2, tDPI is 0.5 – each task in this path can take twice its planned duration without impacting the project's end date. Path 3-6-7 must be handled slightly differently. Using the formula above for tDPI would yield a value of 20/30 = 0.67. This implies that DPI for Task 3 may go as low as 0.67 before the project's end date is at risk. Since Task 3 is also on path 3-4-5 which is the critical path, we know that this is not the case. Any delay in Task 3 impacts the project's end date.

Thus, we can see that, where multiple paths originate from a given task, the tDPI which applies to that task is the value associated with the longest of the resulting paths. In other words, it is the largest (highest risk) value of tDPI which can be calculated from that task. This is also the path with the least amount of total float.

The correct tDPI for path 3-6-7 is 1.0 since task 3 is on the critical path but we can also define tDPI for sub path 6-7 which does have float. Here we discover another attribute of tDPI. Path 6-7 starts 10 days into the project. The critical path duration remaining at this point is only 20 days. Thus, tDPI for path 6-7 is calculated as 10/20 = 0.5. This is clearly the correct value since each task on path 6-7 can indeed take twice its planned duration without impacting the project's end date. Refining the formula for tDPI we can write:

tDPI = Planned Path Duration / Remaining Critical Path Duration

To illustrate the use of DPI and tDPI, consider the following examples:

- If Task 1 is successfully completed in 5 days, as planned, DPI = 1.0. tDPI for path 1–2 is 0.5 so there is no problem. Furthermore, Task 2 now has 25 days left to complete so the tDPI for path 1-2 has changed to 10/25 = 0.4 indicating that there is reduced risk on this path.
- If Task 1 takes 10 days instead of the planned 5 days, DPI = 0.5. This is equal to tDPI for path 1-2. Task 2 still has an appropriate amount of float and there is no apparent increase in risk to the project.
- Finally, consider the case where Task 1 takes 15 days to complete. DPI = 0.33 which is less than tDPI for path 1-2. There are now only 15 days remaining to the project end date and Task 2 is planned to take 10 days to complete. Clearly, Task 1 has consumed some of the float apportioned to Task 2 and Task 2 can no longer take twice its planned duration without impacting the end date. tDPI for path 1-2 has changed as a result of progress to date and is now equal to 10/15 = 0.67.

Again, we can refine the definition of tDPI:

tDPI = Remaining Planned Path Duration / Remaining Critical Path Duration

This sensitivity of tDPI to progress on the project is an important strength of the metric. Each time the project is updated, new values of tDPI are created for every path, accurately reflecting the current status of the project. Better than expected progress on a path will cause tDPI to be lowered, reflecting lower risk associated with that path. Slower than expected progress will cause tDPI to rise, reflecting higher risk. Similarly progress on the critical path, which is better than or worse than plan, will affect tDPI for all other paths in the network.

Implementation Note: Although tDPI is defined for each path, it is possible to calculate the tDPI from the available CPM data for each task in the network. The total float on a path is constant for all tasks on that path at a given point in time. Thus, for any path in the network,

Remaining Planned Path Duration = Remaining Critical Path Duration - Total Float

And therefore:

tDPI = (Remaining Critical Path Duration - Total Float) / Remaining Critical Path Duration

This formula also solves the problem of determining which tDPI to use for a task from which multiple paths originate since the existing CPM algorithms correctly determine the total float based on the longest path emanating from each task.

As stated previously, if DPI for the current task in a given path is less than tDPI for that path, that task has been delayed to the point where it is consuming the available float for later tasks in the path. This fact has been demonstrated by the examples given. Thus, by comparing DPI and tDPI for all active tasks in the network, the project manager may analyze the delays to non-critical tasks and decide where intervention is required. In a very large network however, even this residual analysis may prove very time consuming for the project manager.

A third metric is thus defined to further automate the analysis of the project. This metric is the Intervention Index (II) which is defined as follows:

$$II = (tDPI-DPI) / (tDPI + DPI)$$

This relative difference formula is used to normalize the value of II into a range between -1 and +1 to make comparison of values from tasks with widely different DPI and tDPI values possible.

An intervention index may be calculated for each active task in the network and then used to sort the tasks so that those with higher II values appear at the top of the list. Any value greater than zero indicates that DPI is lower than tDPI and intervention should be considered.

Note that traditional earned value analysis would also flag non-critical path tasks that are behind schedule but earned value lacks the integration of float into the analysis. These metrics do not replace earned value but the inclusion of float into these metrics provides an alternative perspective that allows for the prioritization of non-critical path schedule delays that indicate risk to the overall project schedule.

Implementation

Considering the previous example of the 50 day task to produce 50 units at a rate of 1 per day: at the first reporting meeting when only 2 of the 50 planned units (4%) are complete after 5 days and the task still has 17 days of float remaining, the DPI for this task would be:

$$DPI = (Planned\ Task\ Duration / Actual\ Task\ Duration\ to\ Date) * \%\ complete$$

$$DPI = (50 / 5) * 4\% = 0.4$$

This value is clearly far below 1 which immediately gives rise to concern. The example didn't specify the overall critical path duration but let's assume that this is 150 days from the date of this first sign of a problem. In that case:

$$tDPI = (\text{Remaining Critical Path Duration} - \text{Total Float}) / \text{Remaining Critical Path Duration}$$

$$tDPI = (150 - 17) / 150 = 0.887$$

Clearly DPI for this task is significantly lower than tDPI and the intervention index, II is:

$$II = (tDPI - DPI) / (tDPI + DPI)$$

$$II = (0.887 - 0.4) / (0.887 + 0.4) = 0.378$$

In the second reporting period, these metrics would show even more concern. With 10 days elapsed, only 3 of 50 units have been produced and the float has reduced to just 10 days.

$$DPI = (50 / 10) * 6\% = 0.3$$

$$tDPI = (145 - 10) / 145 = 0.931$$

$$\mathrm{II} = (0.931 - 0.3) / (0.931 + 0.3) = 0.513$$

Note that the tDPI for this path is moving closer to 1 which is the tDPI of the critical path and the II continues to rise.

After three weeks, with all float gone and only 5 of 50 units produced, when the problem was first communicated outside of the subproject, the metrics indicate:

$$DPI = (50 / 15) * 10\% = 0.333$$

$$tDPI = (140 - 0) / 140 = 1.0$$

$$II = (1.0 - 0.333) / (1 + 0.333) = 0.5$$

Despite a slight improvement in the value of DPI, and II resulting from the team working overtime the fact that tDPI is now 1.0 makes it very clear to the project manager and to the overall project that this task is in trouble. DPI is an objective measure of task performance that is not mediated by the team's optimism that they can recover the original plan and the tDPI and II metrics provide quantitative prioritization of the potential impact to the overall project.

Finally, after 4 weeks when the team realizes that its mitigation plans are not viable, we have to adjust how we calculate the metrics. Since this has now become the critical path, tDPI should equal 1.0. However, because this is not an authorized delay to the project, we need to recognize the negative float by forcing the overall project end date to its originally committed date. This results in metrics as follows:

$$DPI = (50 / 20) * 16\% = 0.4$$

$$tDPI = (135 - (-20)) / 135 = (135 + 20) / 135 = 1.148$$

$$II = (1.148 - 0.4) / (1.148 + 0.4) = 0.483$$

Note that, if the team maintains the yield of 3 units per week and completes the work in 70 days according to the revised schedule, taking a total 90 days to produce all 50 units, the metrics will continue to show moderate improvement over time, reflecting the impact of the overtime worked to achieve 3 units per week. At task completion, the metrics would be:

$$DPI = (50 / 90) * 100\% = 0.556$$

$$tDPI = (65 - (-20)) / 65 = (65 + 20) / 65 = 1.31$$

$$II = (1.31 - 0.556) / (1.31 + 0.556) = 0.404$$

Of course, the project could opt to re-baseline to reflect the delay, in which case, DPI would be 1.0, tDPI would be 1.0 and II would be 0. There are legitimate arguments for and against such re-baselining as there are with standard CPM.

Pilot Data

This method has not been tried on any live projects to date. However, I have created a simulation that has allowed me to explore how the metrics would work in various scenarios. This simulation was instrumental in refining the method to include normalization of the II metric since it quickly became clear that a simple delta (tDPI - DPI) or ratio (tDPI / DPI) would yield extreme results that could mask other significant variances that also needed attention. The simulation generates the following data for each task:

- Planned Task Duration: a random number between 1 and 20 days.
- Actual to Date: a random number between 1 and the planned task duration.
- Percent Complete: a random number between 0 and 100%
- Estimate to Complete: calculated as actual to date divided by percent complete minus the actual to date.
- Remaining Critical Path Duration: a random number between 100 and 200 days.
- Total Float: a random number between zero and remaining critical path duration minus the remaining planned duration of the task

Using these inputs, the simulation calculates the values of DPI, tDPI and II for each task and color codes the resulting values to highlight tasks with lower DPI, higher tDPI and higher II relative to others within the simulation. The number of tasks per simulation is relatively low to make visual inspection of the results easier but I have run many simulations and the model can be scaled up to any desired level for further investigation. Note: the estimate to complete does not factor into the metrics in any way but provides a useful sanity check on the relative II scores by indicating how much of the available float would be consumed if the task were allowed to continue at its current pace without improvement. ETC is not really a particularly useful metric on its own because it fails to take into consideration any special circumstances (e.g. unavailable resources, missing parts etc) that have caused delays up to the current time but which may not persist throughout the remaining portion of a task.

In the following tables, I present a few of these simulations and provide commentary on the results to illustrate how effective these metrics are in differentiating between various task scenarios.

Task	Planned Task	Actual to Date	Percent Comp	Est to Complete	Rem CP	Total Float	DPI	tDPI	II
1	3	1	4.02%	23.9	141	57	0.121	0.596	0.663
2	6	3	15.08%	16.9	128	116	0.302	0.094	-0.526
3	13	1	81.36%	0.2	189	55	10.576	0.709	-0.874
4	7	5	18.26%	22.4	194	44	0.256	0.773	0.503
5	3	2	6.01%	31.3	179	123	0.09	0.313	0.553
6	19	9	84.77%	1.6	199	94	1.789	0.528	-0.545
7	13	1	86.70%	0.2	118	29	11.271	0.754	-0.875
8	13	9	6.22%	135.8	147	123	0.09	0.163	0.29
9	10	1	97.04%	0	108	3	9.704	0.972	-0.818
10	9	8	79.37%	2.1	110	2	0.893	0.982	0.047
11	3	2	4.35%	44	144	135	0.065	0.063	-0.021
12	7	1	26.26%	2.8	130	91	1.838	0.3	-0.719
13	15	8	76.35%	2.5	171	2	1.432	0.988	-0.183

TABLE 1: SIMULATED SCENARIO 1 TO EXPLORE METRICS EFFECTIVENESS

In this scenario, tasks 1, 4, 5 and 8 stand out as having the highest II values. Task 1 was planned for 3 days and, with 1 day elapsed, has only achieved 4% completion to date. The value of DPI is very low reflecting the poor performance to date. With 57 days of float remaining, the value of tDPI is moderate but the value of ETC indicates that, without a change in the rate of completion for this task, it could consume a substantial portion of that remaining float. It is likely that, for such a short duration task with so little elapsed time and such low progress to date, the ETC value may not be accurate but it is also possible that this task has simply been massively underestimated. The burden of responding to an inquiry about this task to determine which scenario is accurate is well worth the effort for the health of the overall project. If the task really represents very little risk, it should drop off the list, or at least show substantial improvement, in the subsequent reporting cycle and, if not, this would provide an early warning and an opportunity to take corrective action.

Task 5 is another short duration task with an extremely low DPI value. Due to a significant float of 123 days on this task, the value of tDPI is also quite low and so, while this task also merits attention, the risk to the overall project appears to be lower. Similarly for task 8, despite extremely low DPI and an ETC that actually exceeds the remaining float, the value of II, while positive, is relatively low. This reflects the fact that there remains a lot of time to address the issues with this task and take corrective action. As a project manager, I would want to follow up on both of these tasks and it could be that Task 8 actually represents a greater risk to the project than Task 5 because it is planned as a longer duration task and it has more elapsed time while only achieving similar progress of ~6% towards completion. This means that the current status of Task 8 may be more reflective of a future reality than for Task 5.

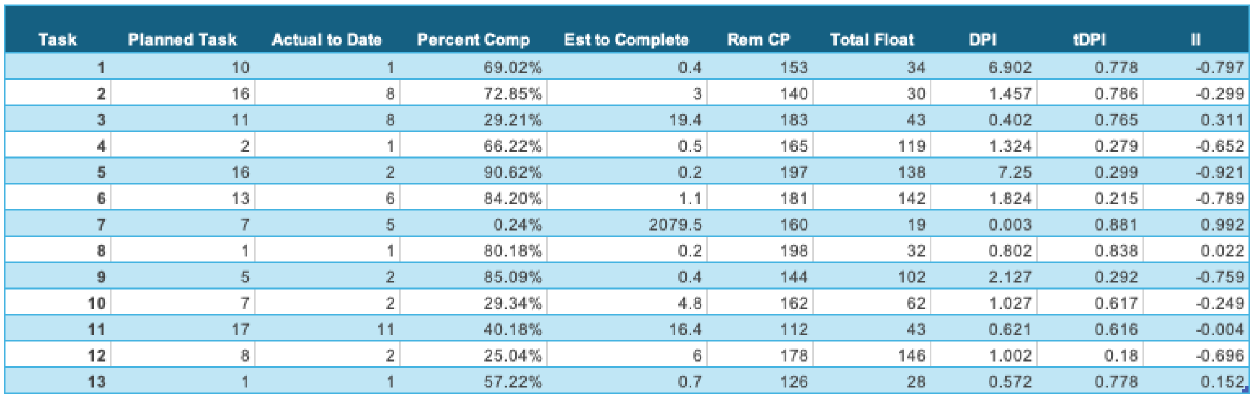
Note that, through the mystery of random number generators, the value of Total Float and DPI for tasks 5 and 8 are the same but what gives task 8 a lower tDPI and hence, a lower II, is the lower

Remaining Critical Path Duration relative to task 5. This makes the Total Float of 123 days for Task 8 more significant than the same float value for Task 5.

Task 4 presents a different case. Here, although the task is significantly behind schedule, performance is not as bad as in the previous two cases. However, the tDPI metric shows that this task has less available float and, in fact, the ETC suggests that without some improvement, this task could consume more than half of the remaining float available to it. This task certainly merits some investigation, especially if other tasks are dependent on this.

The only other task in this scenario to have a positive value of II is Task 10. DPI for this task is marginally lower than tDPI but the tDPI value indicates that the task is near critical. In fact, the ETC calculation suggests that the task is likely to be completed just in time to avoid any significant delay but before the next weekly check-in. While this task may not require any intervention, it certainly merits close monitoring to ensure that it does indeed complete no later than currently projected.

Tasks 2 and 11 in this scenario are also worth analyzing. Both of these tasks are significantly behind schedule with low or very low values of DPI. However, the large remaining float values for these tasks mean that despite these very low DPI scores, they are still greater than the corresponding tDPI values and, as a result, the II values are negative indicating that these delays pose minimal risk to the project schedule.



Task	Planned Task	Actual to Date	Percent Comp	Est to Complete	Rem CP	Total Float	DPI	tDPI	II
1	10	1	69.02%	0.4	153	34	6.902	0.778	-0.797
2	16	8	72.85%	3	140	30	1.457	0.786	-0.299
3	11	8	29.21%	19.4	183	43	0.402	0.765	0.311
4	2	1	66.22%	0.5	165	119	1.324	0.279	-0.652
5	16	2	90.62%	0.2	197	138	7.25	0.299	-0.921
6	13	6	84.20%	1.1	181	142	1.824	0.215	-0.789
7	7	5	0.24%	2079.5	160	19	0.003	0.881	0.992
8	1	1	80.18%	0.2	198	32	0.802	0.838	0.022
9	5	2	85.09%	0.4	144	102	2.127	0.292	-0.759
10	7	2	29.34%	4.8	162	62	1.027	0.617	-0.249
11	17	11	40.18%	16.4	112	43	0.621	0.616	-0.004
12	8	2	25.04%	6	178	146	1.002	0.18	-0.696
13	1	1	57.22%	0.7	126	28	0.572	0.778	0.152

TABLE 2: SIMULATED SCENARIO 2 TO EXPLORE METRICS EFFECTIVENESS

In this second scenario, task 7 clearly stands out as the most urgent issue to address. With 5 days gone in a task scheduled for 7 days, virtually no progress has been made. Total float is relatively low and the ETC calculation indicates a huge problem if progress doesn't step up urgently. The DPI metric reflects the almost total lack of progress while tDPI of 0.881 shows the near criticality of the task. An II value of 0.992 is close to the extreme that this metric can achieve. With 19 days of float remaining, and a task originally scheduled for 7 days, all may not be lost. There may be some special cause of delay affecting this task and if that obstacle can be removed, the task may be able to complete within its remaining float and the project schedule may be preserved.

Task 3 with an II of 0.311 may also need some intervention, especially if there are dependent tasks. While the ETC suggests that this task should complete within its remaining float, the project risk would be reduced if performance on this task can be improved.

Tasks 8 and 13 also have positive II values but these are small duration tasks that are very close to being complete and despite having relatively high tDPI values they have adequate float remaining and should not be a major concern.

Arguably, task 11 could be of greater concern than either Task 8 or 13. Despite having a DPI that is marginally higher than its tDPI, and having 43 days of float remaining, the current progress of this task suggests that a significant portion of that float could be consumed before the task completes. This task is similar to Task 3 which has a positive II value due to its DPI being lower than its tDPI meaning that it is already consuming float that should be apportioned to later tasks. Task 3 also has a higher tDPI resulting from its relatively lower float compared to its remaining critical path duration.

Conclusions

I have presented an algorithm for analyzing non-critical path schedule delays in large and complex projects which removes the need for individual project managers to have either a-priori knowledge of where risk is likely to arise in the project or an all-encompassing knowledge of project networks that may involve hundreds or thousands of tasks and potential critical paths. This method provides a deterministic way of identifying and prioritizing non-critical path schedule delays that rise to the level of concern where additional attention by the project manager and project team is warranted. It also provides an objective view of the data that is independent of assurances by the project team that everything is ok and they can still achieve the planned date within the available float.

In the example project presented above, the availability of these metrics could provide visibility of this issue to the larger project team up to 4 weeks earlier than traditional approaches which would potentially allow for alternative outcomes to be realized, saving the project significant time and money.

Simulation of schedule data supports the idea that these metrics can effectively identify and prioritize non-critical path schedule delays within a project. While there are unique factors that may trigger potential false positives, especially for very short duration tasks or tasks with very low completion levels, these factors can be readily identified in response to such tasks being flagged by the algorithm and the benefit of identifying legitimate high risk delays outweighs the burden of dealing with such false positives.

These metrics are easily calculated from data that already exists in all CPM tools and their implementation and widespread deployment have the potential to significantly improve non-critical path schedule risk management.

The algorithm presented in this paper is intended to supplement existing methods like earned value analysis, Monte Carlo simulation, resource based scheduling etc rather than to replace those methods. It provides another perspective for the project manager to observe the state of their project and make informed decisions about the actions that are necessary to ensure a successful outcome.

Field deployment will be necessary to begin to establish the best practices for interpreting these metrics and identifying triggers for when to intervene as well as decisions about whether and when to re-baseline, how to handle significant replans etc. A necessary pre-condition for such field deployment will be integration of these metrics into existing CPM tools, either natively or as add-on packages.

References

Anbari, F. T. (2003). Earned Value Project Management Method and Extensions. *Project Management Journal*, *34*(4), 12–23. https://doi.org/10.1177/875697280303400403

Elmaghraby, S. E. (2000). [Rev. of *On criticality and sensitivity in activity networks*]. *European Journal of Operational Research*, *127*(2), 220–238. https://doi.org/10.1016/S0377-2217(99)00483-X

Goldratt, E. M. (1997). *Critical chain*. North River Press.

Herroelen, W., Leus, R., & Demeulemeester, E. (2002). Critical Chain Project Scheduling: Do Not Oversimplify. *Project Management Journal*, *33*(4), 48–60. https://doi.org/10.1177/875697280203300406

Kelley, J. E., & Walker, M. R. (1959). Critical-path planning and scheduling. *Papers Presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference*, 160–173. https://doi.org/10.1145/1460299.1460318

Malcolm, D. G., Roseboom, J. H., Clark, C. E., & Fazar, W. (1959). Application of a Technique for Research and Development Program Evaluation. *Operations Research*, *7*(5), 646–669. https://doi.org/10.1287/opre.7.5.646

Raz, T., Barnes, R., & Dvir, D. (2003). A Critical Look at Critical Chain Project Management. *Project Management Journal*, *34*(4), 24–32. https://doi.org/10.1177/875697280303400404

Van Slyke, R. M. (1963). Monte Carlo methods and the PERT problem. *Operations research*, *11*(5), 839–860.

Williams, T. M. (1992). Criticality in Stochastic Networks. *The Journal of the Operational Research Society*, *43*(4), 353. https://doi.org/10.2307/2583158

About the Author

Niall McShane

Washington State, USA

Niall McShane is a retired senior manager and corporate executive. He first managed projects at the European Space Agency in the 1980's as an employee of Logica Space and Defence Systems Ltd before joining Motorola as a project manager, project management advocate and manager of project offices. He was responsible for several significant accomplishments in project management at Motorola including the establishment of an Engineering and Project Management Council which set standards for a career ladder in project management, and the creation of a project management framework for the corporation. Later in his career he served as an engineering manager with Motorola and as a test manager and customer success executive with IBM.

Niall lives near Seattle and enjoys sailing and playing Pétanque. He can be contacted at nialljmcshane@gmail.com