GPM® and *Forensic* Total Float

By Dr. Gui Ponce de Leon, PE, PMP, LEED AP, PMA Consultants LLC

Abstract

The critical path method (CPM) is widely used as a project management tool. Basic to CPM is for the planner to draw a project network first, and then to use CPM software to calculate activity dates and total floats, establish the project completion date and locate the critical path. Alas, all is not well in CPM utopia! The moment the planner takes control of activity dates by using constraint dates or resource leveling, total float is reduced (some would say sequestered); and once actual dates are introduced in a CPM network, the analyst loses total floats and the critical path can no longer be calculated left of the data date. The ability to schedule an activity later than its early dates without sacrificing total float, and to determine total floats and the critical path for the *as-built* portion of a schedule (left of the data date), is solved by the graphical path method (GPM®). This paper describes the float aspects of GPM, while emphasizing the concept of GPM *forensic total float* and its role in retrospective schedule analysis, whether in updating or forensic scheduling. A compendium of CPM and GPM float concepts is provided for historical context.

I. The CPM Alternative

CPM was developed in the late 1950s as a *prospective* method for planning and scheduling complex projects [Kelley & Walker, 1959]. In CPM, planning states which activities must occur and in which logical order for the project to complete, and ends with a *network diagram* of all activities in logical sequence ***without dates.*** With activities' *elapsed times* and the network logic, CPM scheduling first calculates early start and finish times for all activities. In conventional CPM, using the project completion as the late finish of the last activity, late finish and start times for all activities are then calculated. If the time from early start to late finish available to any activity exceeds its elapsed time, it has positive total float and is *a floater;* if not, the activity has total float of zero and is deemed *critical.*

When late time calculations are limited by the project completion date, a floater may be delayed within its total float range without delaying completion of the project. CPM's emphasis on late times is thus essential to timely project completion; however, it is typical of CPM to continue to focus on the earliest possible activity starting times, thereby ***relinquishing the ability*** to interpret float as also measuring available schedule gain, not just delay.

As a prospective method, the application of CPM was for an *entire* project, *before* any work was started (data date on project start date). When introduced, CPM did not consider updating and maintaining the network and schedule current with progress once the project started. In the early 1960s, various *updating* methods for planning, scheduling and controlling what remains of a *partly completed* project were introduced [Moder & Phillips, 1964, p 265; O'Brien, 1965; Antill & Woodhead, 1965, p 183; Shaffer et al., 1965]. Prevailing methods statused completed activities differently, but shared the common limitation of applying CPM calculations only to the portion of a network right of the *data date*. As originally practiced, updating did not extend to applying the CPM calculations left of the data date.

II. Retrospective Calculations – The CPM *Workaround*

Once actual dates are introduced in the scheduling process, the CPM algorithms cease to function for the portion of the network left of the data date. CPM calculations go inactive to the past of the data date for two reasons: 1) total floats can no longer be calculated using the CPM equation of late finish date (actual finish) less early finish date (also actual finish); and 2) even if total float, in the conventional CPM sense, could be inferred by observation, corrections would have to be made as, for any completed activity, actual dates may not have occurred on the *earliest possible* dates, a necessary condition for CPM to correctly calculate activity as-built total floats.

With CPM unable to calculate total floats in the past, the critical path cannot be located for the *statused* portion of a schedule, or for the as-built schedule. Analysts work around this CPM forensic void by *pushing the data date out of the way*. To deploy CPM calculations, analysts using a *baseline* [AACE, 2009, p 17], hold the data date at the project start or at the start of the period being evaluated. Statusing is limited to making the durations of activities experiencing progress match actual or would-be remaining durations and to actualizing logic, and *excludes introducing* actual dates.

© 2010 PMA Consultants LLC

Originally Published as Part of the PMICOS 2010 Conference

To deploy the CPM calculations, analysts working with as-built schedules perform prospective simulations of the as-built in its entirety, or for an as-built window [Ponce de Leon, 1984; Keane & Caletka, 2008, p 141]. For the as-built schedule, these simulations reset the data date back to the project start, remove actual dates, set activity durations as the time between actual start and finish dates and actualize logic, yielding a prospective model of the as-built. CPM is able to calculate simulated early dates and total floats for the ***actual durations,*** which reveals the *as-built critical path*. Prospective simulations of the as-built schedule (or an as-built schedule window) work around CPM's inability to calculate total floats left of the data date by *moving* the data date to the project start (or to a prior window data date).

III. Problems Stemming from the CPM *Modus Operandi*

A CPM schedule chock-full of early dates that do not make use of available total floats is counterintuitive to those responsible for delivering the project. Left on their own to develop more realistic working schedules, field personnel use bar charts and other methods to schedule the work, regrettably, often disconnected from the CPM schedule.

CPM limits options by treating total float as only measuring leeway with respect to activity late dates (and therefore, project completion). Moreover, in CPM schedules, when an activity is scheduled later than its early dates through a no-earlier-than start constraint or a resource-constrained date, its total float is correspondently reduced by the difference between the constraint date or resource-constrained date and the early start date.

Knowing that completed activities and the start of in-progress activities are left out of CPM calculations, few practitioners feel compelled to actualize logic ties and lags–the *forgotten step* in updating [O'Brien & Plotnick, 2010, p 535]. It does not help that schedulers' propensity for ignoring logic/lag status left of the data date is encouraged by out-of-sequence rules available from CPM software [Glavinich, 2004, p 173; Keane & Caletka, 2008, p 65; Wickwire et al., 2003, p 478]. In addition, without actual dates figuring in total float calculations, the rigor applied to recording actual dates is inconsistent amongst schedulers [Wickwire et al., 2003, p 477; O'Brien & Plotnick, 2010, p 536].

Although actual activity start/finish dates do not affect the CPM calculations, they document the as-built schedule; to qualify as an as-built schedule the cause of delays need not be shown so long as the delay effect is shown [AACE, 2009, p 21] (for an activity, the delay effect is the activity-level variance). However, without total floats, there is no algorithmic way to calculate the critical path. "The as-built critical path cannot be directly computed using CPM logic since networked computations that generate float values can be generated only to the future (right) of the data date" [AACE, 2009, p 99]. This guidance is only partly correct as total floats are needed for criticality, *not just floats*.

Some analysts overcome CPM's retrospective void by using *observational* methods that rely on updates. "The closest the analyst can determine the as-built critical path is to cumulatively collect from successive schedule updates the activities that reside on the critical path between the data date and the data date of the subsequent update" [AACE, 2009, p 99]. One such method gleans floats from updates and groups and filters driving activities to identify the as-built critical path. The authors warn that float values can be manipulated by a planner and should never be used as the sole indicator of determining the location of the driving path or as-built critical path [Keane & Caletka, 2008, p 240].

IV. The Graphical Path Method Alternative

The graphical path method (GPM) is a rule-based planning/scheduling method that uses a network of activities in logical sequence to calculate the project completion and total floats and locate the critical path [Ponce de Leon, May & July 2008]. However, owing to different scheduling principles, alternate algorithms, and computer graphics relying on ***objectbase*** principles that enable ***gestural*** technology, there are vast differences and enhancements. To underscore how GPM retools planning and scheduling–and project management for that matter, consider the following:

- Imagine a method that allows stakeholders to place activities between early and late dates or, even late dates, without using constraint dates or resorting to resource leveling, and without ***sacrificing*** total floats.
- Imagine having activities 100% completed while still computing total floats and pinpointing the location of the critical path and near critical paths *left of the data date*.
- Imagine a graphical, interactive system that allows stakeholders to work on any wall, in full view of and while engaging all stakeholders, using a stylus or hand-touch to work with the activities in real-time.

© 2010 PMA Consultants LLC

Originally Published as Part of the PMICOS 2010 Conference

[A] Working Right of the Data Date–*Planning*

Unique to GPM is placing activities on planned dates, not necessarily CPM early dates, as the network is being built [Ponce de Leon, May 2009]. As the network progresses, using logic ties and object dates (for activities, floating milestones and fixed events), GPM continuously calculates link gaps and total floats. Link gaps on non-driving logic ties are displayed in Figure 1 (22-day gap between Foundations and Erect Steel, 7-day gap between Erect Steel and Masonry, etc.). With GPM relying on network logic, an activity cannot be scheduled before early dates, although if forced out of sequence, negative gaps arise. Activities on other than early dates may float in either direction.

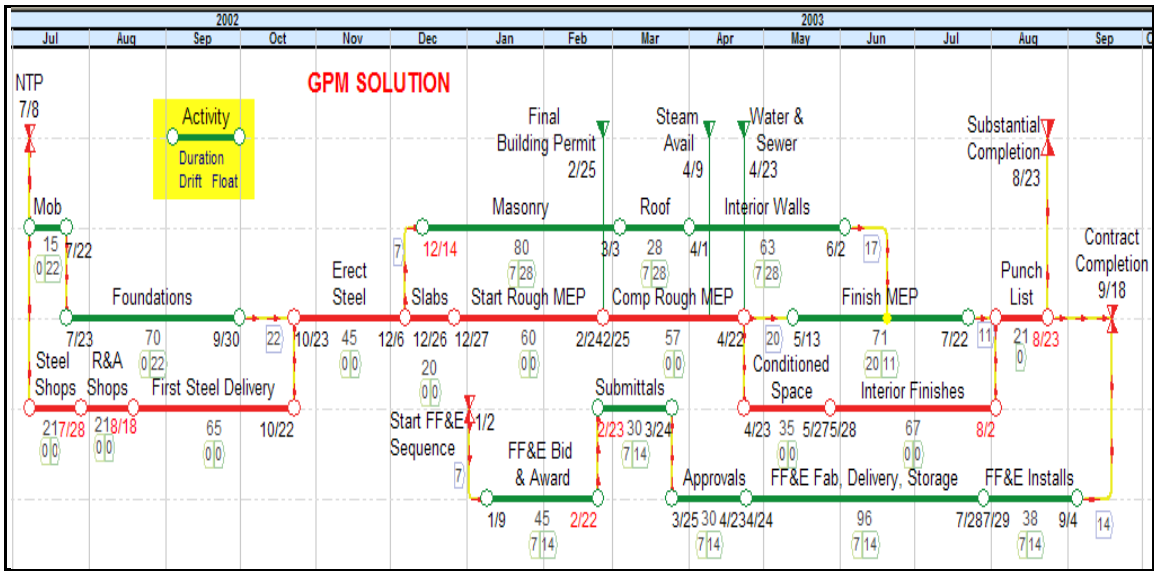


Figure 1. GPM Schedule on ***Planned Dates*** Displaying Gaps, Drifts and Floats

An activity that may be delayed while not causing an overrun in the project completion or an interim completion date, has available *float* and is a *floater*. In Figure 1, Interior Walls is an example with float of 28 days. Other examples include, FF&E Installs (float of 14 days) and Foundations (float of 22 days). An activity that may gain schedule and/or extend to earlier planned dates while not forcing an earlier project start or interim release, has available *drift*, and is a *drifter*. Interior Walls and FF&E Installs have drift, 7 days each. Foundations, on the other hand, has no drift as the Mob-Foundations logic chain is on early dates. For every activity, float plus drift equals CPM total float.

The concept of CPM free float as float available to an activity when all its successors start on their respective early dates is not flexible enough to capture GPM concepts. Instead, GPM calculates activity *buffer* and activity *drift-buffer*. Buffer measures forward float available to an activity from its *planned dates* without delaying any of its successors from their respective *planned dates*. Drift-buffer measures the extent of gain available to an activity in its *planned dates* without forcing any of its predecessors into earlier dates.

Buffer for an activity is calculated as the minimum gap on the logic ties to all its successors. For instance, Interior Walls has a buffer of 17 days. For an activity, drift-buffer is calculated as the minimum gap on the logic ties from all its predecessors. For instance, Masonry has a drift-buffer of 7 days and Roof has zero drift-buffer.

Because pacing is inherent in GPM scheduling (activities may be planned to start later than the early start dates), GPM has the capability to measure pacing algorithmically. For an activity without driving predecessors, i.e., all logic ties merging into the activity have positive gap, the smallest of the gaps (its drift-buffer), measures the extent of pacing. Further, an activity with a zero-gap predecessor(s) but with positive drift may be part of a paced logic chain. In Figure 1, Interior Walls without drift-buffer but with drift of 7 days is part of a paced logic chain, starting with Masonry.

Figure 2 portrays the CPM ***early-dates*** solution of the GPM schedule in Figure 1. Purely for comparison purposes, with activities in their early dates, all drift values are necessarily zero. For instance, Interior Walls and FF&E Installs have zero drift; their drift values have been subsumed in the float value, as their floats equal their total floats. Again, as demonstrated in Table 1, GPM drift plus GPM float exactly equals CPM total float.

© 2010 PMA Consultants LLC

Originally Published as Part of the PMICOS 2010 Conference

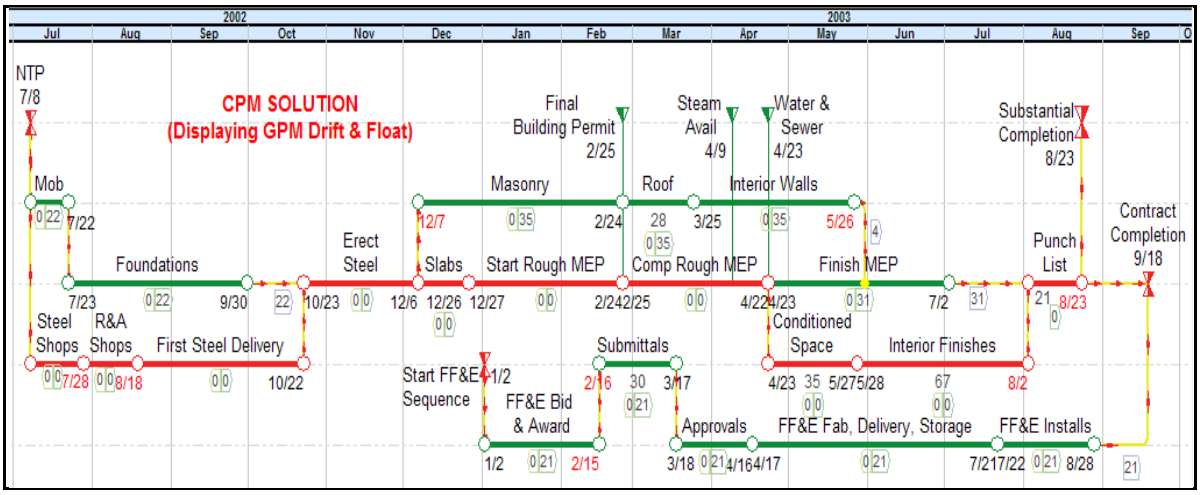


Figure 2. Alternate CPM Schedule on ***Early Dates*** has Zero Drifts and Floats Equal Total Floats

The GPM concept that an activity scheduled between early and late dates has float available *in either direction* is a novel idea to CPM schedulers. In fact, with GPM, total float of an activity is unaffected even if the activity is scheduled in its *late dates.* With CPM, when an activity is scheduled later than its early dates through a no-earlier-than start constraint or leveling date, its total float is correspondently reduced. Figure 3 illustrates how total floats are reduced when activities in a CPM schedule are in the *GPM planned dates.* Table 1 contrasts all three solutions: 1) the GPM schedule with activities in *planned dates*; 2) the conventional CPM schedule with activities in *early dates*; and 3) the CPM schedule with GPM planned dates as no-earlier-than constraint dates.

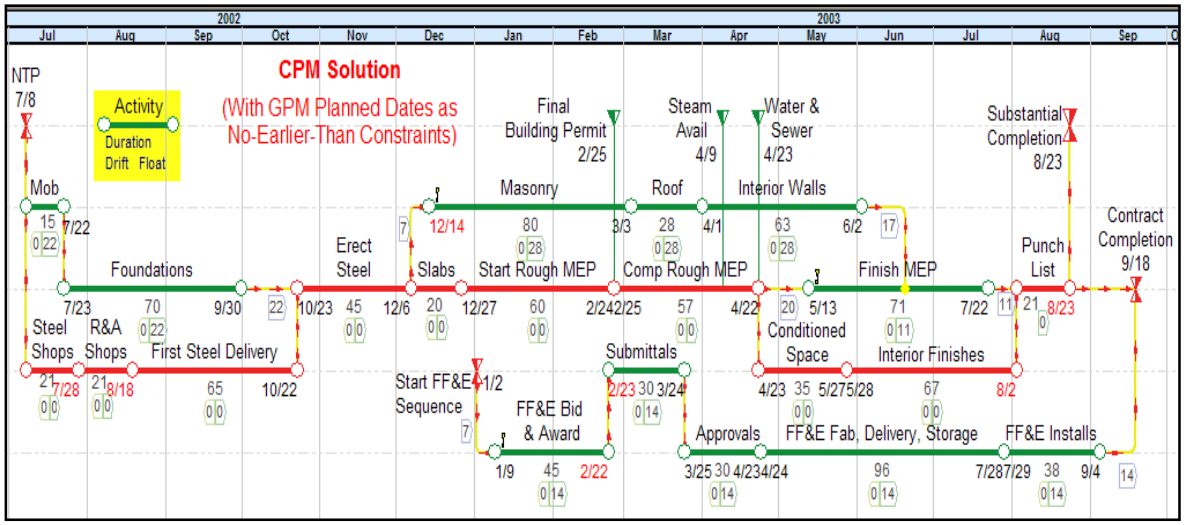


Figure 3. Alternate CPM Schedule on ***Planned*** Dates Has Total Floats Reduced by Constraint Dates

Table 1 Contrasting GPM and Two CPM Solutions Relative to Drifts, Floats and Total Floats

Selected Activities (logic chains are color coded)	GPM *Planned* Dates Solution			CPM *Early* Dates Solution			CPM *Planned* Dates Solution		
	Drift	Float	Total Float	Drift	Float	Total Float	Drift	Float	Total Float
Mobilization	0	22	22	0	22	22	0	22	22
Foundations	0	22	22	0	22	22	0	22	22
Masonry	**7**	**28**	**35**	**0**	**35**	**35**	0	**28**	**28**
Interior Walls	**7**	**28**	**35**	**0**	**35**	**35**	0	**28**	**28**
Finish MEP	**20**	**11**	**31**	**0**	**31**	**31**	0	**11**	**11**
FF&E Bid and Award	**7**	**14**	**21**	**0**	**21**	**21**	0	**14**	**14**
FF&E Installs	**7**	**14**	**21**	**0**	**21**	**21**	0	**14**	**14**

© 2010 PMA Consultants LLC

Originally Published as Part of the PMICOS 2010 Conference

[B] Working Left of the Data Date–*Statusing*

When statusing a GPM schedule, completed and in-progress activities are on actual start/finish and actual start dates, respectively, which, except for critical activities, are not necessarily the earliest possible actual dates. Logic ties are corrected if actual dates for related activities are out of sequence. A completed activity that *could have* floated to later actual dates without delaying the project has forensic float. A completed activity that *could have* drifted to earlier actual dates without forcing an earlier project start has forensic drift. Any completed activity with forensic drift plus forensic float equal to or less than zero (relative to the calculated completion date) is on the as-built critical path.

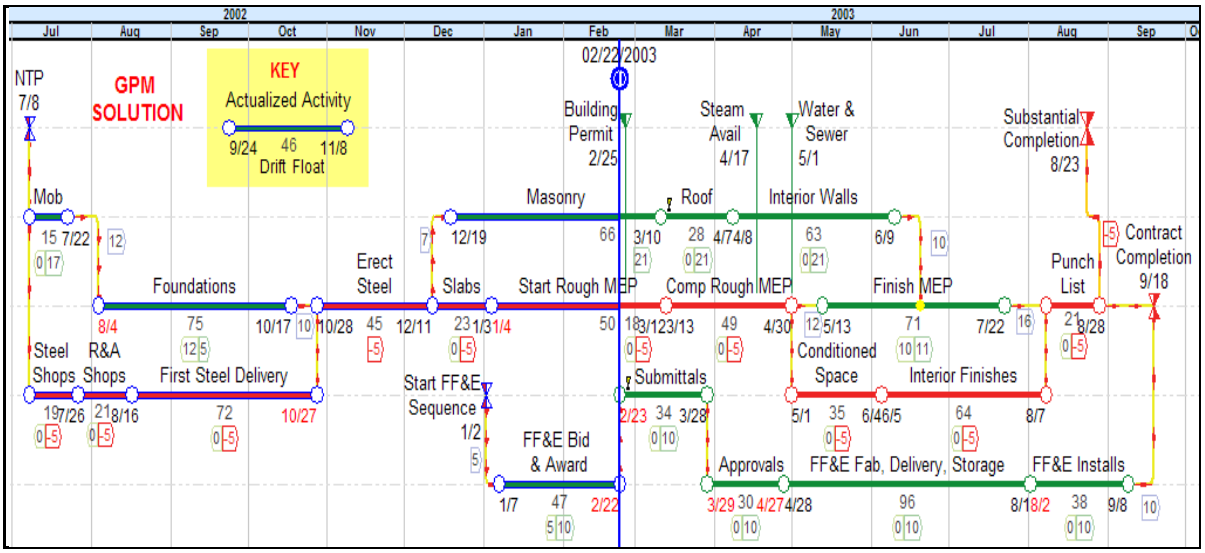


Figure 4. GPM Schedule Statused as of 22-Feb-03 Displaying Forensic Gaps, Drifts and Floats

In Figure 4, the critical path right of the data date has total float of -5 days. The as-built portion of the critical path has forensic drift of zero, forensic float of -5 days, and forensic total float of -5 days. There are three completed, non-critical activities: Mob, Foundations, and FF&E Bid & Award. Foundations connects to the critical path at Erect Steel; its forensic float of 5 days reflects the forensic gap in the logic tie to Erect Steel (10 days) and the forensic float for Erect Steel (-5 days). FF&E Bid & Award is on a logic chain ending on FF&E Installs; its forensic float is the float of the logic chain right of the data date, however, with activities right of the data date on early dates, their drifts are zero.

Once a data date is introduced 1) the data date becomes the boundary for drift calculations right of the data date, and 2) the project start date remains the boundary for forensic drift calculations left of the data date. In this sense, GPM and CPM algorithms are in sync, GPM not permitting drifting, and CPM not permitting early dates, before the data date. As it relates to total floats and the critical path left of the data date, GPM is not stymied by actual dates because GPM algorithms do not rely on actual dates to calculate total floats–total floats on either side of the data date are calculated from logic tie gaps. Excluding out-of-sequence progress, forensic gaps are greater than or equal to zero.

In the GPM solution, the forensic drift of 12 days for Foundations points out and measures delayed-start pacing vis-à-vis the controlling Steel Delivery logic chain. In the baseline (Figure 1), Foundations had zero drift and 22 days of float. In the update in Figure 2, the 12-day gap in the logic tie from Mobilization measures 12 days of pacing (delaying the start) of Foundations with respect to the Steel Delivery portion of the critical path.

[C] The GPM *Retrospective* Perspective on Total Float

Forensic drifts and forensic floats are a function of forensic gaps, which are commonly fixed (when the predecessor and successor activities are assigned actual dates), ***and*** gaps, which refresh to current planned dates. From update to update, as activities are statused left of the data date, and re-planned right of the data date, forensic floats and forensic drifts continuously refresh algorithmically, just as their counterparts right of the data date—with total float calculations *remaining active* on both sides of the data date line. This means that, as a dynamic attribute, like the critical path itself, forensic total floats convert to as-built floats in the final update (data date equal to the project completion date).

© 2010 PMA Consultants LLC

Originally Published as Part of the PMICOS 2010 Conference

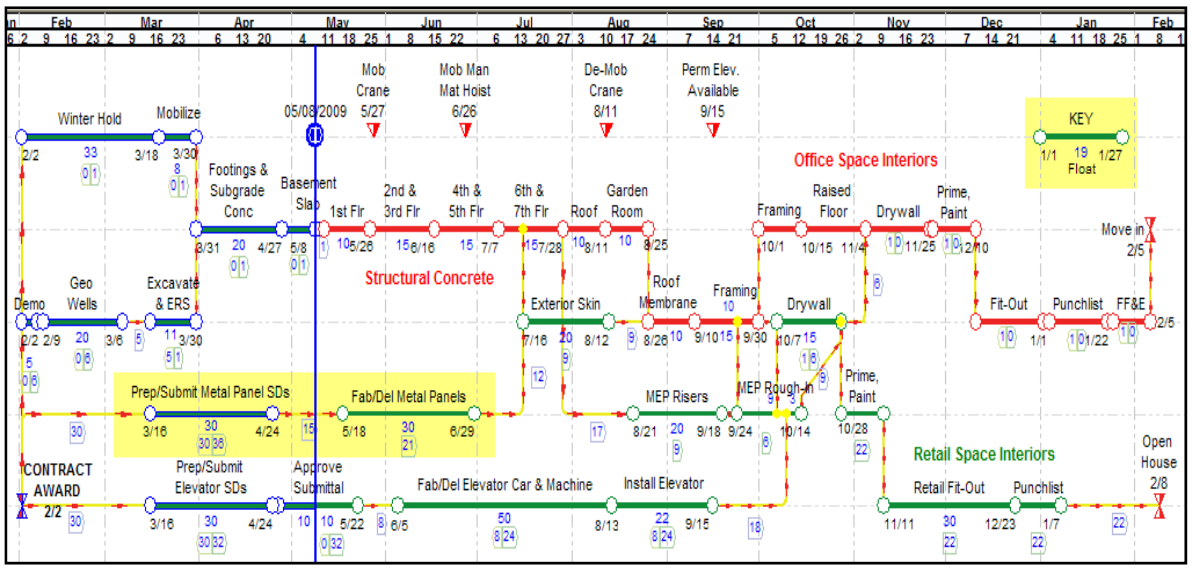


Figure 5. Forensic Gaps and Forensic Floats Left of Data Date of 08 May 09

In the May update in Figure 5, a critical path break occurs at the data date, which, is in fact, the gap on the logic tie between Basement Slab and 1st Floor. The 1-day gap represents either a delay or pacing of critical path concrete work, starting with 1st Floor, depending upon the baseline timeframes and information from as-built records (pacing would imply that the critical path had one day of float between scheduled and contractual completion).

Notice that the Fab/Del Metal Panels activity is being paced; in fact, 15 days have elapsed between the completion of the submittal activity and the planned start for Fab/Del Metal Panels. The float of 21 days of this activity is the sum of the gap of 12 days to its successor Exterior Skin, and the successor's float of 9 days.

In the July update in Figure 6, the forensic float of Prep/Submit Metal Panel SDs refreshes from 36 to 26 days based on the forensic float status of Fab/Del Metal Panels (16 days) and the forensic gap (10 days). The forensic float of Fab/Del Metal Panels is controlled by the float of Exterior Skin, a remaining activity (right of data date).

Similarly, the forensic float of 1day for Footings & Subgrade Conc increases to 5 days in the July update (Figure 6) due to the forensic gap that developed between its successor Basement Slab and 1st Flr. Forensic floats and forensic drifts (and therefore forensic total floats) of other completed activities similarly react to downstream gaps.

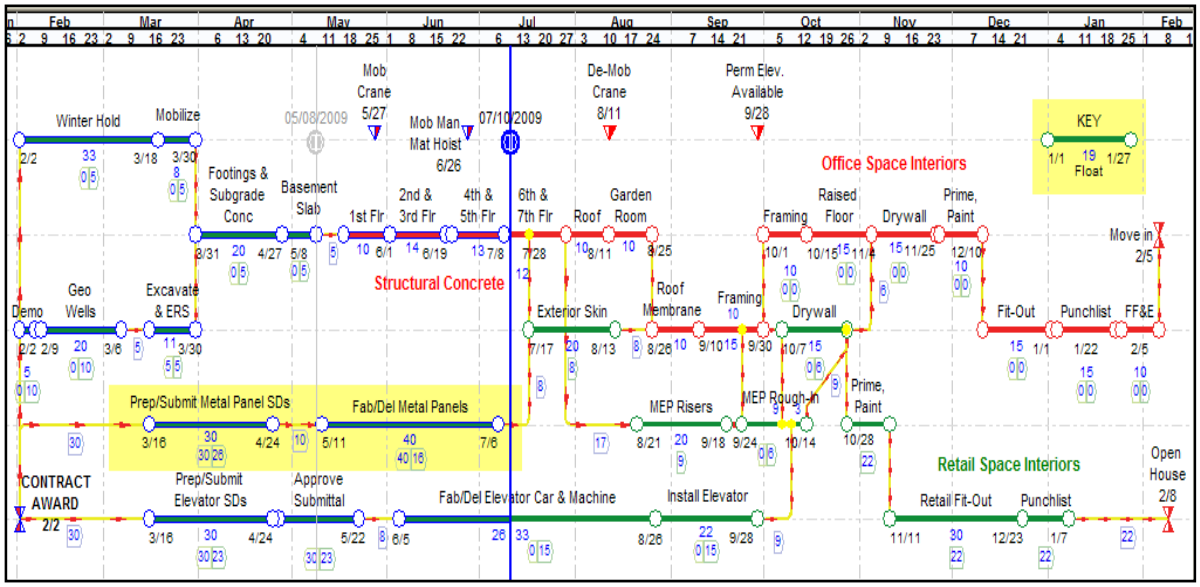


Figure 6. Forensic Gaps and Forensic Floats Left of Data Date of 10 Jul 09

© 2010 PMA Consultants LLC

Originally Published as Part of the PMICOS 2010 Conference

[D] Out of Sequence Progress

Out-of-sequence progress occurs when the actual sequencing of two connected activities contradicts their logic tie. CPM schedulers, rather than rectifying the logic tie to reflect reality, turn to shortcuts offered by their scheduling software, with progress override and retained logic as the two common calculation rules [O'Brien & Plotnick, 2010, pp 540-541]. The AACE International Recommended Practice No. 29 (RP 29) provides alternate guidelines for restating CPM logic ties to simulate as-built performance [AACE, 2009, pp 74-76] resembling updating practice from the 1970s when the common fix was to correct logic to reflect the actual dates [Wickwire et al., 2003, p 478].

In a GPM update, out-of-sequence progress causes negative forensic gap and reduces forensic total float. *Corrected logic* rules (with *omitted logic* options) fix the ambiguity through equivalent zero-gap logic ties or FS logic (in the latter case by subdividing the predecessor and/or successor) [Lu & Lam, 2009]. In the first two cases in Figure 7, the successor's actual start is earlier than what the logic tie dictates causing negative gaps if the logic ties are not rectified. Once corrected to preserve good logic, the float of the predecessor increases by the amount of the initial negative gap.

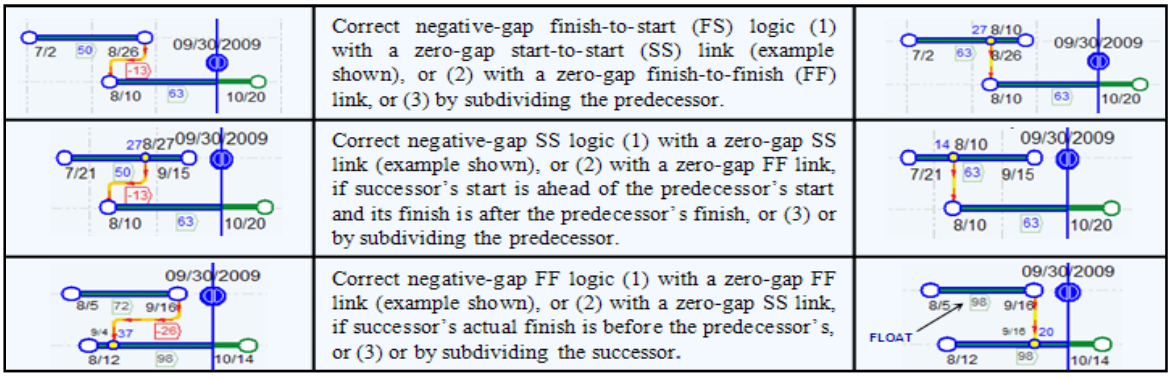


Figure 7. Corrected Logic Rule Options (Data Date of 09/30/09)

[E] As-Built Critical Path

The as-built critical path is located with forensic total floats. Normally, a completed activity with forensic float and drift of zero, has zero forensic total float. A break in the as-built critical path stands out and is rectified by recognizing the concealed delay posing as a non-driving link (positive forensic gap). Consider the following two snapshots of the second update. In Figure 8, there is no as-built path with activities having ***both*** zero forensic drift and forensic float.

Figure 9 reveals the as-built critical path with the addition of Rebar Hold, a concealed delay in the as-built condition that prevents earlier actual start date for its successor critical activity. There is no as-built critical path through the Excavate & ERS activity because the activity was paced with respect to the controlling Winter Hold-Mobilize path.

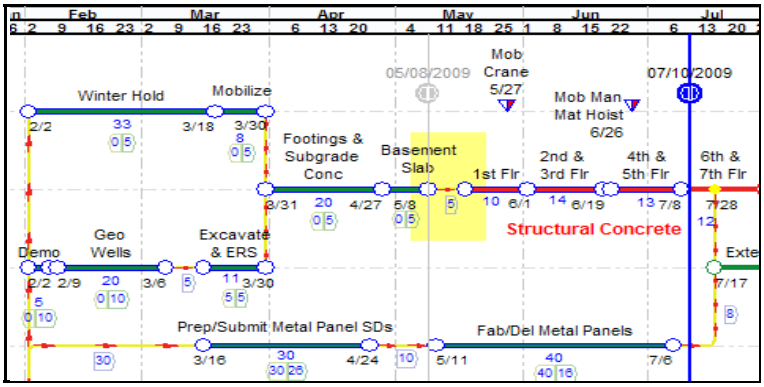


Figure 8. Apparent As-Built Critical Path Break

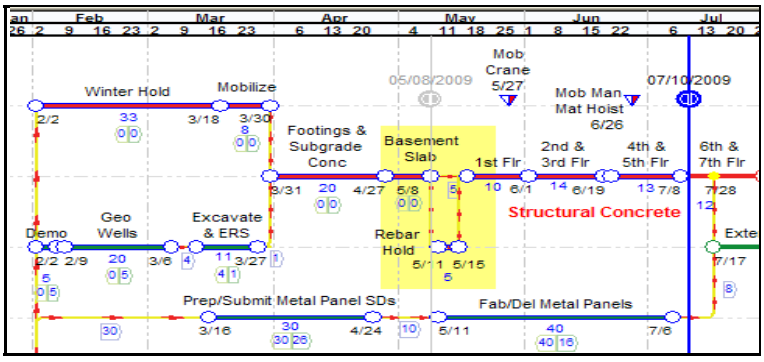


Figure 9. As-Built Critical Path Revealed

© 2010 PMA Consultants LLC

Originally Published as Part of the PMICOS 2010 Conference

V. Compendium of CPM and GPM Floats

Arrow Diagramming Method (ADM) – The Kelley & Walker network [1959] uses arrows between activity tail (i) and head (j) nodes to represent activity i,j. An activity that may start upon the finish of another is connected to the predecessor through a common node or dummy activity, which limits ADM notation to finish-to-start (FS) logic (*FS logic rule*). Using ADM notation (TE/TL is early/late node time), *total float* is interpreted as from late finish (TLj) minus early start (TEi), subtract the duration for the activity. *Free float* is interpreted as from *earliest* early start of all successors (TEj) minus activity early start (TEi), subtract the duration. *Interfering float* is total float less free float or TLj minus TEj. *Independent float* is interpreted as from TEj (*earliest* early start of all successors) less TLi (*latest* late finish of all predecessors), subtract the duration, but it must be at least zero.

CPM with a Backward Pass Premise – CPM assumes a project start date, from which the completion date is to be calculated (forward pass premise); this leads to floats rooted in *early dates* and *head nodes*. If the assumption is the project completion date, from which an optimal project starting date is to be calculated, the emphasis switches from early dates to late dates, with floats rooted in *late dates* and *tail nodes*. With a backward pass premise, free float, actually *backward free float*, becomes: from activity late finish (TLj) minus *latest* late finish of all predecessors (TLi), subtract the activity duration [Zimmerman, 1967]. *Interfering float* and *independent float* also have to be restated, but total float calculation remains unchanged. There are many projects where the completion date is a given (e.g., completion of a school in time for fall opening), and the goal of the planning and scheduling process is to assist stakeholders to determine an optimal starting date (including contingency), not necessarily the earliest starting date.

Activity-on-Node Network (AON) – Fondahl [1962] led an independent, parallel effort to develop an approach for manually applying CPM. The method substitutes a *circle and line* for the *arrow* notation, but holds onto the FS logic rule. The method calculates activities' early start and finish *dates* and late start and finish *dates*. Due to the FS logic rule, float calculations for activities are essentially the same, except they are driven by *activity dates* rather than tail and head node times. Fondahl originated the concept of *driving links*, and calculated *link lags* for non-driving links, albeit within the confines of the FS logic rule. For any link, the *link lag* value (not to be confused with PDM lead/lag) is interpreted as the early start of the successor minus the early finish of the predecessor.

Precedence Diagramming Method (PDM) – As originally developed, PDM extended the AON model to allow three types of relationships [IBM, 1968; O'Brien, 1969, pp 99-115]. The start-to-finish (SF) relationship type was added by Ponce de Leon [1970, pp 42-47; Lu & Lam, 2009]. In PDM, an activity may start or finish ***sometime*** after a predecessor has started or finished. The minimum *waiting time* between the predecessor/successor-connected relevant dates is the *lead/lag* factor. The relevant dates for each PDM dependency stem from the logic type. The calculation of free float changes for the four PDM relationship types, as free float can no longer be stated only in terms of start of successor and finish of predecessor. Instead, *free float* measures days the predecessor may be delayed from its early dates without delaying any of its successors from their respective early dates. Equations for total float and interfering float remain unchanged. PDM was eventually extended to calculate Fondahl's link lag concepts for start-to-start (SS), finish-to-finish (FF) and SF relationship types [Ponce de Leon, 1972, pp 120-138].

Resource-Constrained Scheduling (RCS) – In CPM schedules conditioned on resource limits, r*esource-constrained float* measures days an activity may be delayed from its recalculated early dates while breaching neither resource limits nor completion [Kim & de la Garza, 2003]. Resource-constrained float is calculated, per Kim & de la Garza, by removing what they deem *phantom float* from remaining float associated with the recalculated RCS early dates.

GPM, Forward Pass Premise – GPM provides five *prospective* float attributes. 1) Float measures days an activity may slip from and/or extend beyond planned dates without the network overrunning the project completion date or an interim completion date. As GPM float results from planner-devised logic, and has GPM drift as an inherent counterpart, it is neither *scheduled float* [Antill & Woodhead, 1965, p 65] nor *remaining float* [Kim & de la Garza, 2003]. 2) Drift measures days an activity may backslide from planned dates and/or extend to earlier planned dates without forcing an earlier project start or release date. 3) Float plus drift equals total float. 4) Buffer equals the minimum of all link gaps to the activity's successors. 5) Drift-buffer equals the minimum of all link gaps from the activity's predecessors. In addition, *forensic floats, forensic drifts and forensic total floats* (left of the data date counterparts) are provided. As forensic float is measured from actual dates forward, both forensic drift and forensic float are required to properly calculate forensic total float.

© 2010 PMA Consultants LLC

Originally Published as Part of the PMICOS 2010 Conference

Link gaps are key in GPM calculations as all floats and drifts originate at the gap level. *Gap* measures days the predecessor may slip and/or extend without delaying its successor. *Drift-gap* measures days the successor may gain and/or extend to earlier dates without forcing the predecessor back. PDM logic leeway (i.e., GPM gap), originally calculated by this author [1972, pp 120-138], has been posited as *inter-activity float*/*relationship slack* [Winter, 2004].

Relationship Diagramming Method (RDM) – RDM is a variant of ADM, with PDM features, that focuses on the rationale for each relationship [Plotnick, 2006]. RDM introduces *just-in-time float (JTF)* to measure float of support activities leading to an *i-node* or merge point that is selected to be driven by another chain of activities. RDM calculates a *multi-calendar float (MF)* that homogenizes activity floats on a chain having dissimilar calendars to a uniform calendar. As with PDM logic, total floats for the start and finish nodes may differ [Ponce de Leon, 1972, pp 139-177]. Thus, RDM calculates *start float* and *finish float*, with *total float* as the lesser of the two. RDM adopts ADM or PDM calculations for free and independent floats.

Activity Float Interpretations – *Activity float* measures activity slippage from early dates without altering the *early schedule tree*, after allowing activities on the same chain a proportionate share of the chain's float based on activity-to-chain duration [Ponce de Leon, 1983]. *Discrete activity float* assigns to each activity on a path a proportionate share of the path's combined total float and free float, using activity-to-path duration as the basis for proportioning [Woolf, 2007]. If PDM calculations allow start float and finish float for an activity to differ, float embedded, if any, within the activity start and finish dates is *internal activity float* [Pickavance, 2005, p 579]. The difference between the duration assigned to an activity and the time necessary for the activity is deemed *activity float* [Keane & Caletka, 2008, p 265].

Other Float Interpretations – *Contract float* measures days between contractual and projected completion available to offset critical path delay, pursuant to applicable contract clauses [Ponce de Leon, 1985]. It is also called *terminal* or *end float* [Keane & Caletka, 2008, p 191] and *external float* [Pickavance, 2005, p 594]. S*equestered float* is that concealed by a preferential link or other technique for the primary intent of reducing float [Ponce de leon, 1985; Keane & Caletka, 2008, p 193]. A *float map* lists activities in the baseline and contemporaneous updates along with their respective total float values from update to update to highlight activities that are consistently critical in contemporaneous updates [Keane & Caletka, 2008, p 238]. A *float trend* sorts baseline activities and remaining activities in the updates, from update to update, by ascending total float; and plots, from update to update (x-axis), the total float (y-axis) for the 0%, 10%, 50%, 90% and 100% percentiles [Kilpatrick, 1972].

VI. Summary and Conclusions

Two float aspects of CPM that hinder the analysis of schedule performance are expounded: 1) once constraint dates or resource leveling is introduced in a CPM schedule, total floats are sacrificed and the critical path can no longer be trusted; and 2) once actual dates are introduced, CPM loses the ability to calculate total floats in the past (left of the data date) and the critical path can no longer be located for the as-built portion. These maladies afflicting CPM are well documented in scheduling literature. Practitioners have devised workarounds within the confines of CPM.

The GPM updating mantra is for the planner to apply equal rigor to logic definition right and left of the data date. For out-of-sequence progress, the GPM *corrected logic* rules (including *omitted logic options)* better portray as-built conditions than what results from rules currently practiced by schedulers using features of CPM scheduling software. A payback from more accurate as-built network logic is true forensic total floats and more accurate schedule analysis results. Calculation of true total floats, even with constraints, and true forensic total floats as tools to locate the critical path, have been *missing links* in CPM scheduling. That void has now been dealt with by the GPM scheme of thought.

Pacing by *delaying the start* of an activity in a GPM schedule is obvious: the activity has drift and its total float exceeds its float. Pacing in a CPM schedule is accomplished with constraint dates or resource leveling, either of which conceals the pacing and reduces total float. In addition, a CPM as-built schedule has difficulty when pacing converts into delay, because prospective simulations necessary to ready the as-built for collapsing erase all traces of pacing.

Researchers and commentators have proposed many measures of float beyond the original four used by Kelley & Walker. Over 35 float variations (5 are relationship-related), are catalogued. With three years of actual GPM application on projects ranging from $15M to $2B, it is posited that *gap* and *drift-gap* (based on ***planned*** vs. early dates), *float, drift, total float* and *contract float* yield sufficient float information for project management and control.

© 2010 PMA Consultants LLC

Originally Published as Part of the PMICOS 2010 Conference

References

AACE International (2009). *Recommended Practice for Forensic Schedule Analysis No. 29R-03.* Atlanta, GA.

Anthill, J. & Woodhead, R. (1965). *Critical Path Methods in Construction Practice*. New York, NY: John Wiley & Sons, Inc.

Fondahl, J. W. (1962). *A Non-computer Approach to the Critical Path Method for the Construction Industry (2nd ed.)*, pp 25-30. Stanford University: Department of Civil Engineering, Stanford, CA.

Glavinich, T. (2004). *Construction Planning & Scheduling (2nd ed.).* AGC of America, US.

IBM. (1968). *Project Control System/360, Application Description.* IBM Application Program, H20-0222-2.

Keane, P. & Caletka, A. (2008). *Delay Analysis in Construction Contracts.* Chichester, UK: Blackwell Publishing Ltd.

Kelley, J. & Walker, M. (December, 1959). *Critical Path Planning and Scheduling*, pp 162-163. Proceedings of the Eastern Joint Computer Conference, Boston, MA.

Kilpatrick, D. (1972). *Float Trend Charts, Project Planning by Network Techniques*, pp 191-225. New York, NY: John Wiley & Sons, Inc.

Kim, K. & de la Garza, J. (2003). *Phantom Float*, pp 507-517. Journal of Construction Engineering and Management, Vol. 129, No. 5.

Lu, M. & Lam, H. (2009). *Transform Schemes Applied on Non-Finish-to-Start Logical relationships in Project Network Diagrams*, pp 863-873. Journal of Construction Engineering Management, Vol. 135, No. 9.

Moder, J. & Phillips, C. (1964). *Project Management with CPM and PERT*. New York, NY: Reinhold Publishing Corp.

O'Brien, J. (1965). *CPM in Construction Management*, pp 148-157. New York, NY: McGraw-Hill.

O'Brien, J. (1969). *Scheduling Handbook*, pp 99-102. New York, NY: McGraw-Hill.

O'Brien, J. & Plotnick, F. (2010). *CPM in Construction Management (7th ed).* New York, NY: McGraw-Hill.

Pickavance, K. (2005). *Delay and Disruption in Construction Contracts (3rd ed.).* London, UK: Mortimer House.

Plotnick, F. (2006). *RDM–Relationship Diagramming Method*. AACE International Transactions, Atlanta, GA.

Ponce de Leon, G. (1970). *Precedence Network-Based CPM, An Introduction*. Ann Arbor, MI: Townsend & Bottum, Inc.

Ponce de Leon, G. (1972). *Extensions to the Solutions of Deterministic and Probabilistic Project Network Models.* PhD diss., Univ. of Michigan, Civil Engineering, College of Engineering, Ann Arbor, MI.

Ponce de Leon, G. (1983). *Activity Float Entitlement*. Strategem, *A Project Controls Journal* 1(3), 1-2. Ann Arbor, MI: Project Management Associates, Inc.

Ponce de Leon, G. (1984). *Setting Up and Adjusting the As-Built Schedule*. Strategem, *A Project Controls Journal 2*(1), 5-6. Ann Arbor, MI: Project Management Associates, Inc.

Ponce de Leon, G. (1985). *Float Ownership: Specs Treatment*. Strategem, *A Project Controls Journal* 2(2), 1-3, 14-15. Ann Arbor, MI: Project Management Associates, Inc.

Ponce de Leon, G. (May, 2008). *Graphical Planning Method (A New Network-based Planning/Scheduling paradigm).* PMICOS Annual Conference, Chicago, IL.

Ponce de Leon, G. (July, 2008). *Project Planning using Logic Diagramming Method.* AACE International Transactions, Atlanta, GA.

Ponce de Leon, G. (2009). *GPM®: An Objectbase Project Networking Method.* PMICOS Annual Conference, Boston, MA.

Shaffer, L.; Ritter, J. & Meyer, W. (1965). *The Critical Path Method*, pp 53-60. New York, NY: McGraw-Hill.

Wickwire, J., Driscoll, T., Hurlbut, S. & Hillman, S. (2003). *Construction Scheduling: Preparation, Liability, and Claims.* Frederick, MD: Aspen Publishers.

Winter, R. (2004). *Longest Path Value (to the rescue).* PMICOS Annual Conference, Montreal, Canada.

Woolf, M. B. (2007). *Faster Construction Projects with CPM Scheduling*, p 336. New York, NY: McGraw-Hill.

Zimmerman, L. S. (1967). *A Network Approach to Resource Scheduling*, pp 17-38. MS Thesis, University of Illinois, Civil Engineering, College of Engineering, Urbana, IL.

© 2010 PMA Consultants LLC

Originally Published as Part of the PMICOS 2010 Conference