Logic Analysis of Resource-Leveled Schedules (MS Project)

By Thomas Boyle, PE, PSP, PMP

Effective management of resources – i.e. planning, procuring, mobilizing, and deploying – is a core competency for successful companies in project-focused industries like construction. Most scheduling tools based on the Critical Path Method (CPM) – like Microsoft Project – can generate project schedules without resources, but they also include methods for assigning, analyzing, and "leveling" project resources. In this context, "leveling" means selectively delaying some work (compared to the CPM-based schedule) pending the completion of other, more important works that demand the same resources.

This simple description could imply the imposition of a certain logical/sequential relationship between two competing tasks (i.e. the "less important" work can only start after the "more important" work is finished with the resources) – sometimes called "soft logic". Unfortunately, the leveling engine in Project 2010 does not appear to use, much less preserve, any such soft logic. Consequently, logical analysis of the leveled schedule – including interpretation of Total Slack to determine critical path or driving logical path – appears invalid.

Figure 1 is a simplified CPM model of a construction project involving multiple trades working in multiple areas. The model includes realistic resource loading, but the logical links have been limited to "hard logic" only (i.e. physical constraints). In other words, there is no preferential logic to guide the resource deployments. The default 5dx8h weekly calendar is universally applied, and a deadline of 25Feb'04 has been imposed. The unleveled CPM schedule includes a forecast completion that is nearly 3 months ahead of the deadline, but resources are severely over-allocated – the schedule appears unrealistic and needs to be leveled.

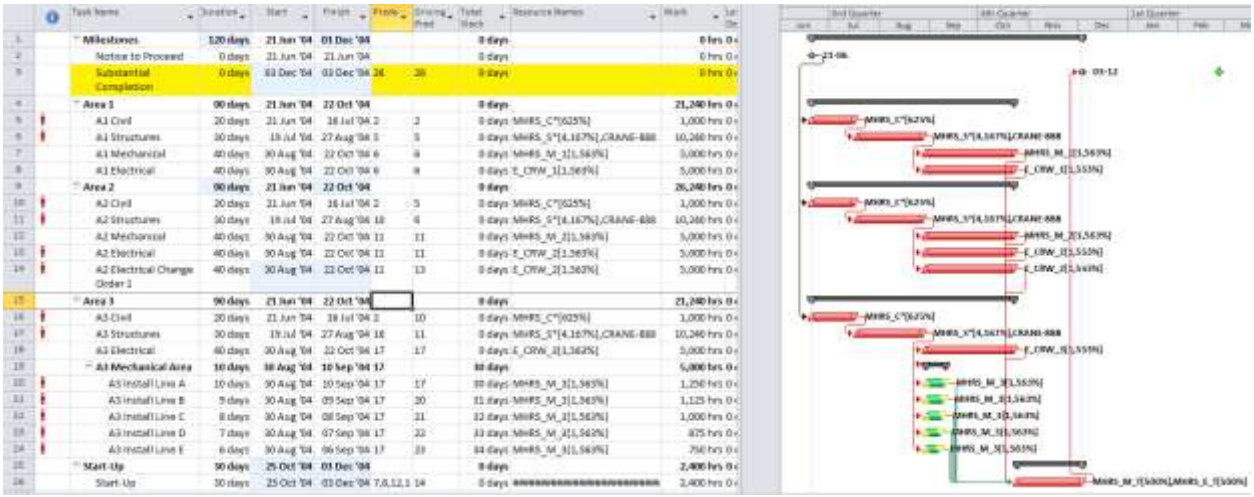

Figure 1: Simple Construction Project with Resource Loading

Specifically:

1. Three civil works tasks are running concurrently, but there is only sufficient manpower to run them sequentially. (Figure 2.)

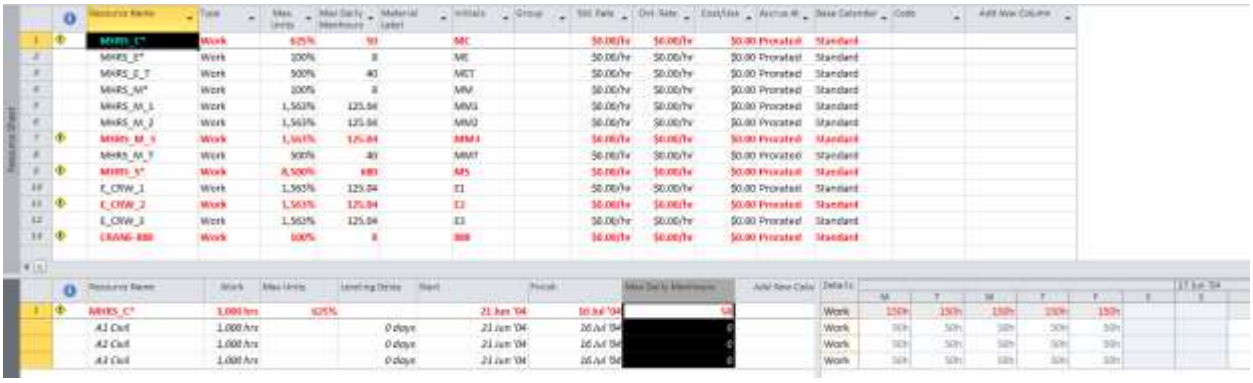

Figure 2: Over-Allocation of Civil Works Manpower

2. Three structural tasks are also running concurrently, and these require both manpower (Figure 3) and a crane (Figure 4), which is the limiting resource. They must be done sequentially.

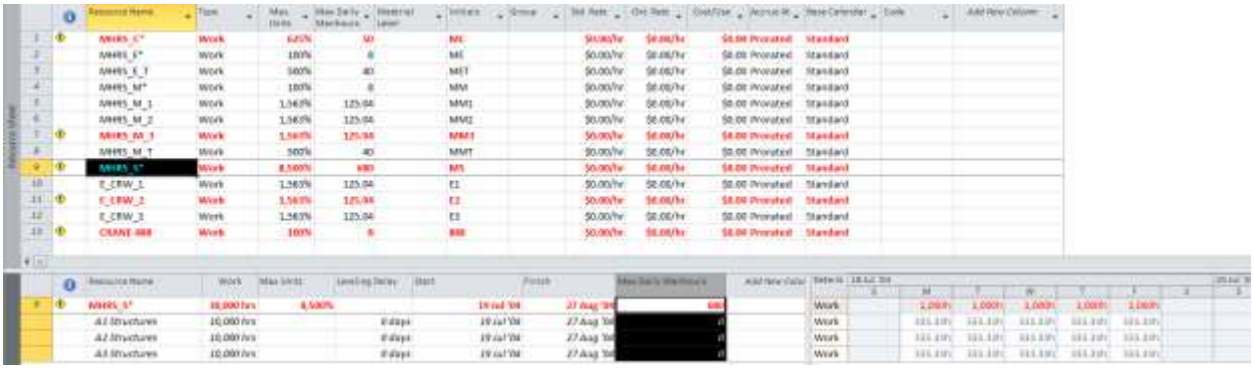

Figure 3: Over-Allocation of Structural Erection Manpower

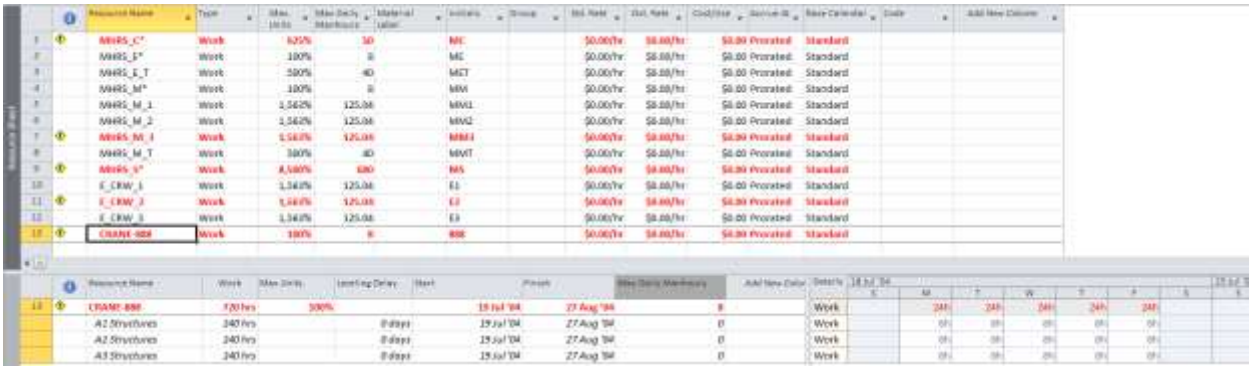

Figure 4: Over-Allocation of Crane for Structural Erection

3. There is room to install the five separate manufacturing lines concurrently, but there is only enough skilled manpower to install them sequentially. (Figure 5).

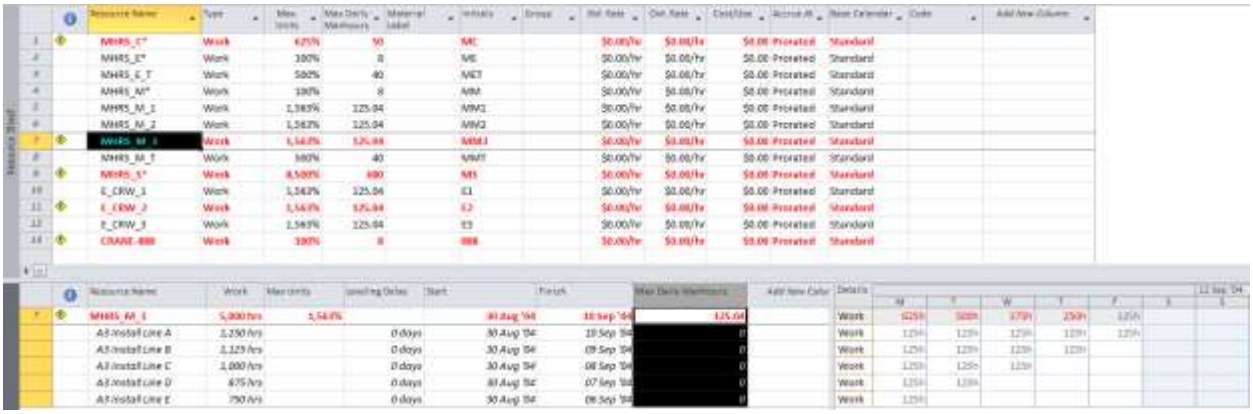

Figure 5: Over-Allocation of Area 3 Specialized Mechanical Installation Manpower

4. An electrical change order has been approved in area 2, but this requires the same specialized crew that is already working there. The Change-order work must be delayed (Figure 6).

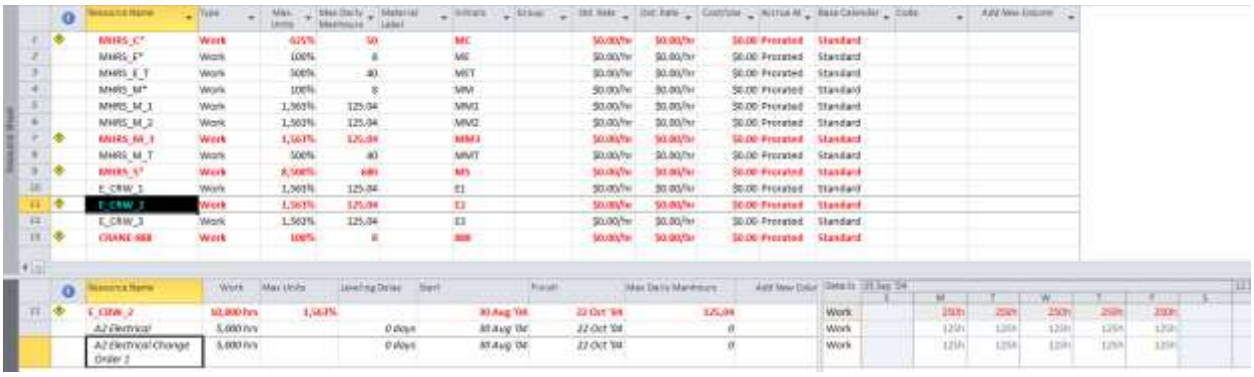

Figure 6: Over-Allocation of Area 2 Specialized Electrical Manpower

It is a simple matter to remove the over-allocations by manually executing Project's leveling engine using near-default conditions (Figure 7).

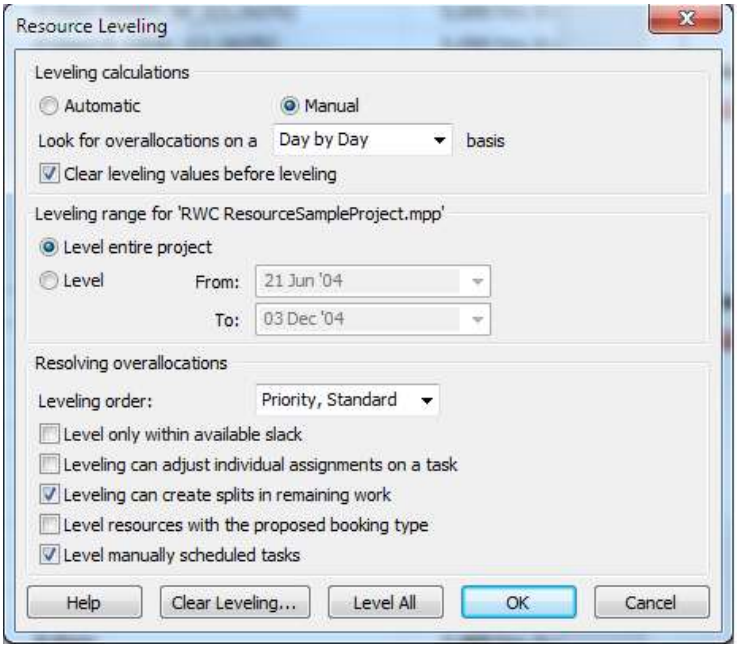

Figure 7: Resource Leveling Options

The leveling engine resolves the over-allocations by selectively delaying those tasks (and task resource assignments, if specified) which are judged to be lower-priority according to Project's proprietary rules. Figure 8 illustrates the results of the leveling exercise:

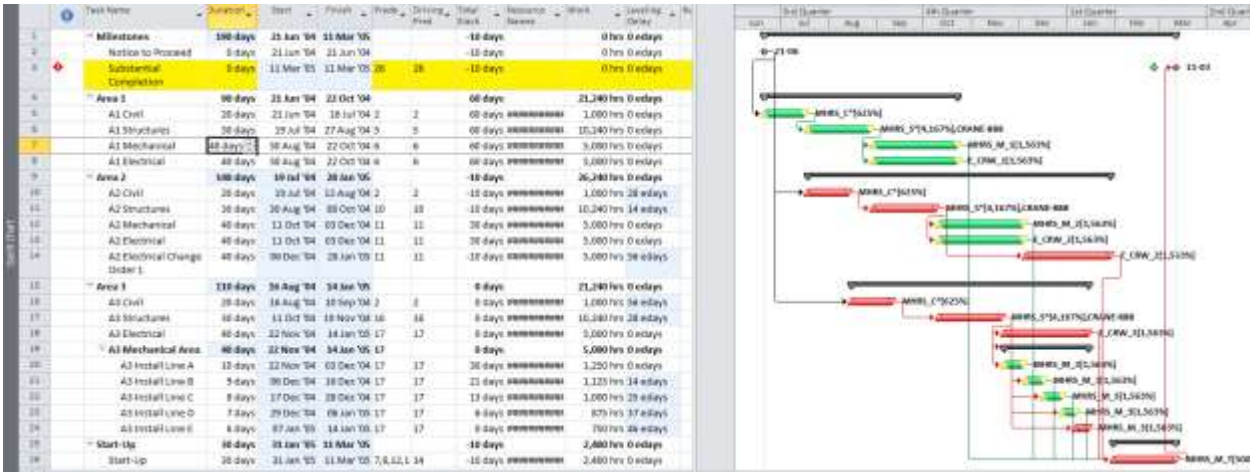

Figure 8: Resource-Leveled Schedule

1. The primary artifact of the leveling process is the "leveling delay" task property, which is in units of elapsed-duration (i.e. "edays"). The leveling delay is incorporated into the forward-pass schedule calculation, pushing the early start dates of the affected tasks. (Separate leveling delays can also be applied to resource assignments, which can extend task durations. This has not been done here and is generally not recommended when assigned resources are expected to work concurrently – e.g. Crane and structural erection crew.) Leveling delay is also incorporated into the backward pass, removing "phantom slack" from logically-connected tasks.

2. Through the task leveling delay, the civil, structural, mechanical, and electrical tasks have been re-scheduled sequentially.
3. Substantial Completion has been delayed until two weeks after the deadline, resulting in 10 days of negative slack on the milestone and its logical driving predecessors.
4. There is not an obvious (-10d) total-slack path from beginning to end of the project.

Figure 9 illustrates the use of BPC Logic Filter to determine the driving path logic of the Substantial Completion task after leveling. The driving path is comprised of four tasks and two milestones separated by gaps, and the intervals of the gaps are determined by the "leveling delay." Unfortunately, this does not describe a "resource constrained critical path." In fact, the obviously critical tasks without leveling delay – including the first (i.e. "A1") Civil and Structural works and the A2 Electrical works – now have high values of total slack and are shown far from the critical path. Consequently, it is clear that logical path analysis – including any evaluation of Total Slack – is not consistent with the rule-based resource leveling algorithm used by Microsoft Project.

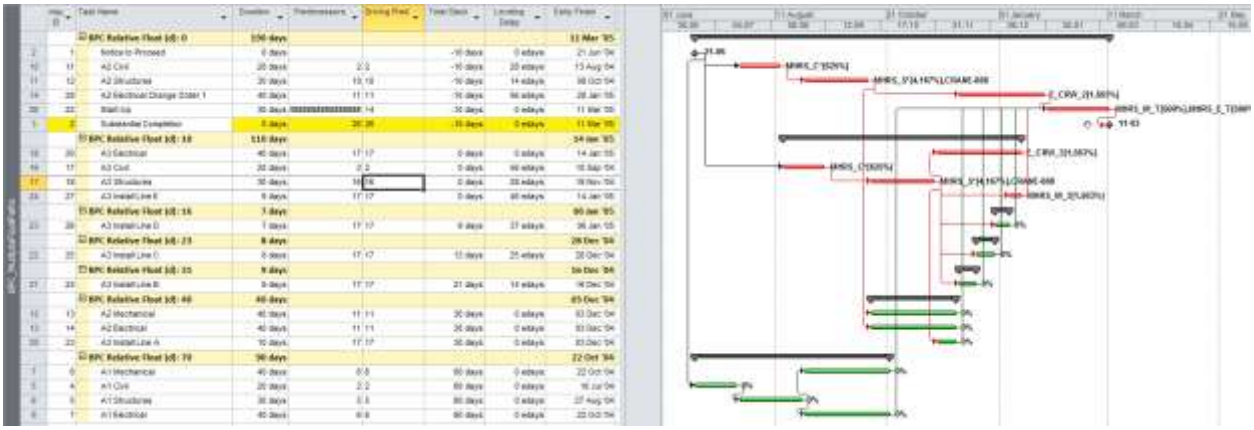

Figure 9: Logic Analysis of Leveled Schedule

Figure 10 illustrates the un-leveled schedule, revised to include preferential logic for avoiding resource conflicts. The preferential logic results in schedule dates that are identical to those of the leveled schedule seen earlier. As shown in Figure 11, however, the logic paths are clear and consistent with the real resource constraints of the project. The "BPC Relative Float (d): 0" group appears to represent the true resource constrained critical path for the project.

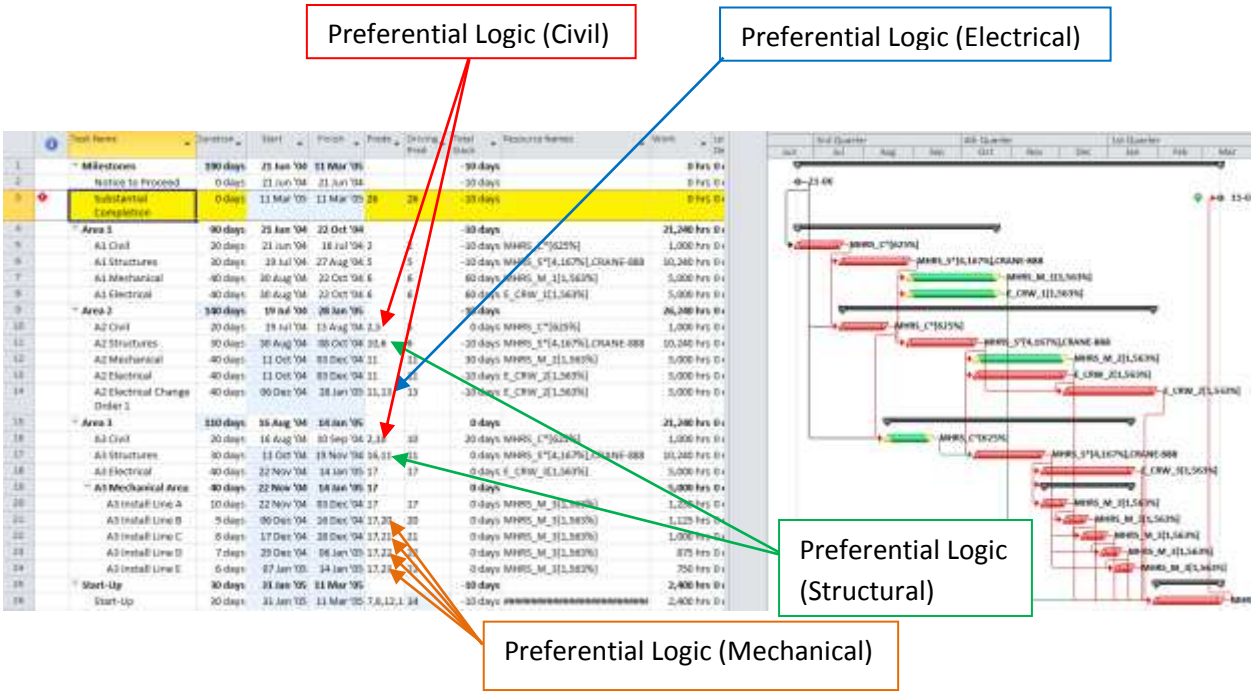

Figure 10: Preferential (Soft) Logic in Unleveled Schedule

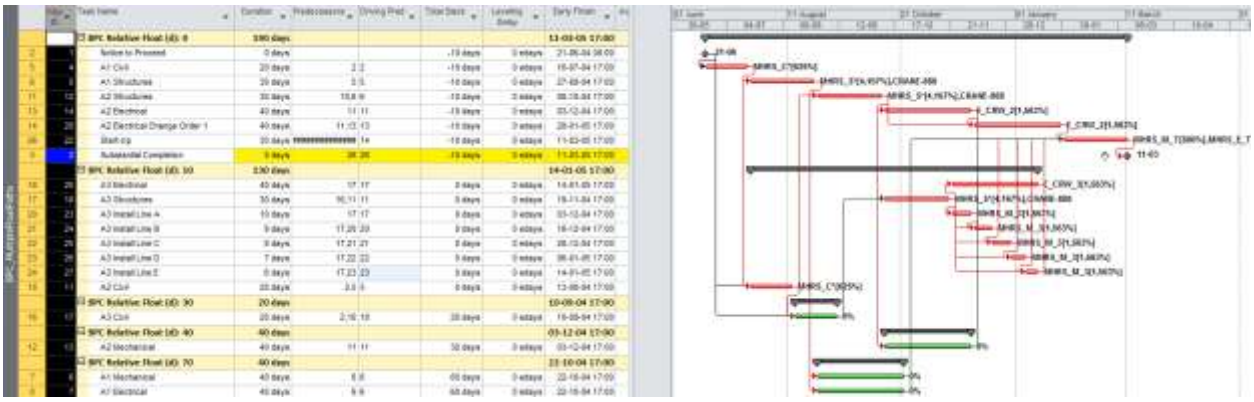

Figure 11: Logic Analysis of Unleveled Schedule with Preferential Logic

In conclusion, Microsoft Project's proprietary resource leveling engine offers a convenient tool for resolving resource conflicts in project schedules, and this functionality seems heavily used and highly valued in some industries. It does not appear appropriate, however, for use in complex projects where formal logical sequencing of tasks - including identification of Critical Path or Critical Chain - is required. In particular, Project's "Critical" flag will fail to mark the complete critical path in a resource-leveled schedule. Consequently, a project specification that requires both a logic-driven schedule basis and heuristic resource leveling appear contradictory.

[31Mar'16: The latest (pending) release of BPC Logic Filter now includes resource leveling constraints in the logical path analysis. I've written another article to summarize and amend this one: The Resource-Constrained Critical Path - Logic Analysis of Resource-Leveled Schedules (MS Project), Part 2 .]