Resource Constrained Scheduling Challenges

Vladimir Liberzon, PMP, GPSF, GCMF
Augustin Purnus, PMP
Introduction

• Resources are limited in most projects. Project schedules that allow reliable decision-making must take into account all existing constraints including resource, supply, space and financial limitations.

• No less important is the quality of the project model.

• To be used for reliable project planning and what-if evaluations required for informed decision-making, a good project model must reflect reality.

• People can only rely on the scheduling results if the software they use takes into account all information used by the people when scheduling their projects manually.
Introduction

• When resources are not limited, the Critical Path Method (CPM) easily finds the best project schedule but CPM does not consider resource, supply, space, and cost constraints.

• With limited resources, calculating the best schedule is a complicated task that has no mathematical solution achievable within reasonable time.

• Project management packages use different algorithms for resource constrained scheduling and produce different results for the same projects.
Introduction

• In this presentation, we will discuss the main activity, resource, and resource assignment properties and project constraints that should be modeled by project scheduling software, as well as the tools and techniques used in resource-constrained schedule analysis.

• Let’s start with discussing the main requirements for a reliable project model.
Project Data: Activities

Most activities belong to one of the following main types:

- **Duration** type activities: for these activities, duration is their initial information and does not depend on the volume of work to be done.

- **Productivity** type activities: these activities are characterized by the volume of work to be done that is usually measured in the physical units. Activity duration depends on the productivity of assigned resources.

- **Level of Effort** or **Hammock** type activity: duration of such activity is determined by the external events so that this activity lasts from one event to another.

- **Milestone**: an activity with zero duration that usually reflects some event in a project schedule.
Project Data: Activities

Activity properties include:

- **Volume** – amount of work to be done, usually measured in physical units (cubic meters, tons, etc.)
- **Duration** that can be defined as initial data for Duration type activities and is calculated for Productivity and Hammock type activities
- **Calendar** that defines the time periods when an activity can be executed
- Whether an activity is **splittable** when the assigned resources are needed on activities with higher priorities
- Whether an activity is **continuous** and must be done without any interruption (example: concrete pouring)
Activity properties include:

- **Priority**: assigned manually, it is used in resource-constrained scheduling

- Whether activity duration is **adjustable**, i.e. may be increased if necessary with the corresponding adjustment of assigned resource workloads

- **ASAP/ALAP**: defines whether an activity must be executed As Soon or As Late As Possible

- **Time constraints** such as Start No Earlier Than or Finish No Later Than
Project Data: Activities

Activity **cost and material requirements** can be defined as:

- **fixed** amount that does not depend on activity volume or duration,
- amount **per unit of volume**, 
- amount **per unit of duration**

Activities are of the same **Type** if they share the same properties such as cost and material requirements per volume unit, resource requirements and productivity.

Corporate databases may contain the **norms** for different types of activities, resources, and assignments that must be used in project models (e.g. material requirements per volume unit, unit costs, resource productivity, etc.)
Project Data: Renewable Resources

Project resources can be divided into two main classes:

- **Renewable resources** that may be used again after they finished the job they have been assigned to
- **Consumable resources** that are spent on activities they have been assigned to and thus cannot be used again

Renewable resource (labor, machines) properties include:

- available **quantities** at different time periods
- resource **calendar**
- **per hour cost**
- **per hour consumption of project materials** (a machine may use certain amount of fuel or electricity per work hour)
Consumable resource properties include:

- **unit** of measure,
- **unit cost** that may be different at different time periods.

Consumable resources do not have special calendars.

Resources (both renewable and consumable) can be **produced** on some project activities (e.g. mobilization for renewable resources and supply for materials), they also may be **spent** or **removed** from the project team on other activities (usually milestones).
Activity duration depends on assigned resource quantity, productivity, and workload. Assignment **quantity** is the number of resource units assigned to do the work. Assignment **productivity** is the volume of work an assigned resource does in one work hour. Assignment **workload** is percent of resource’s work time needed to execute an activity. A resource’s **workload** is 100% for full time assignments and can be less than 100% for part-time assignments.
Partial Resource Assignments

- Most software tools manage and assign resource hours and this may cause problems with project scheduling and reporting.
- Let’s consider a simple project that consists of three independent activities with the same resource A assigned to these activities with 60% workload (0.6 resource hour per hour).
- Suppose we have only 2 units of resource A (or 2 resource hours per hour).
- If we consider the required resource hours only, then all three activities can be done in parallel, which is wrong as none of these 3 resource units can execute two activities simultaneously.
- Leveling resource hours may produce unfeasible schedules.
Partial Resource Assignments

Before resource leveling

After resource leveling
Project Data: Resource Assignments

Renewable resources may be assigned different ways:

- **Team assignment** means that resources assigned as a team do the job together and, if a resource belonging to the assigned team is unavailable, the whole team won’t work.
- **Independent assignment** means that resources may do their job independently of each other and at different time.

Independent assignments are used for modeling work in several shifts.
Simulating work in several shifts

• Let’s assume that resource A works in the first shift and resource B works in the second shift and both have the same productivity of 1 piece per hour on Activity 1 with the volume of work comprising 100 pieces.

• We have assigned both resources independently and Spider Project has calculated what work and when will be done in each shift.
Variable Resource Assignments

- Assignments may be variable when activity resource requirements are defined as a range, specifying the minimum and maximum quantity and workload of the assigned resources.

- It won’t make sense to delay an activity execution just because one out of several assigned resources is unavailable at the moment of the activity’s possible start. With variable resource assignment, an activity execution will be started by the reduced crew and additional resources may join if they become available before the activity is finished. Moreover, even at a reduced pace the activity will finish earlier.
Variable Resource Assignments

- In this example, resource A assignment on activity 1 is variable: 3 units were assigned but the minimum quantity required for the activity start is one unit.

- Activity 1 volume of work is 200 units, activity 2 volume is 100 units, resource productivity is 2 units per hour on activity 1 and 5 units per hour on activity 2.

- After project leveling, activity 1 execution was started by 2 units of resource A and an additional unit joined in after activity 2 was finished.
Resource assignment properties may also include assignment cost and assignment material consumption that may be set as fixed or as an amount per volume or duration unit.

Another way of resource assignment is assigning resource skills rather than concrete resources. Resources have the same skill if they can execute a certain type of activity although perhaps with different productivity and cost.

Project resource-constrained scheduling with automatic resource assignments based on assigned resource skills and availability is called skill scheduling.
Skill Scheduling

In this sample project, resource skill A was assigned to activities 1, 2 and 3, and resource A1 was assigned to activity 4. Both resources A1 and A2 have skill A. Spider Project automatically selected available resources that have required skills.

Activity 3 execution was delayed because both resources with the required skill were busy. The dashed arrowhead line shows the activity on which the resource that has delayed the activity start was used. We call it resource dependency that helps to analyze resource-constrained schedules.
Project Data: Activity Dependencies

• Standard FS, SS, FF and SF links connect activity start and finish points with No Earlier Than dependencies.

• SS and FF dependencies usually mean that two activities may be executed in parallel but with some lag. That is, the next activity can start after a certain amount of work on a preceding activity is done, defining a certain minimal “distance” between crews on the preceding and succeeding activities. This distance is usually measured in volume units and the lag defined this way is called the volume lag.

• The Time lag defines time interval between an activity’s start or finish and another activity’s start or finish, and is used mostly for duration type activities. Time lag can have its own calendar.
Project Data – Activity Dependencies

• It could be also useful to use **point-to-point dependency** that we call **double-lag dependency** when a certain point of a preceding activity is linked to some point of the succeeding activity. These points may be defined as Starts-Plus-Volume Lags.

• In real life, the way a project is executed depends on its status. When major milestones are late, people apply corrective actions, use additional resources or more advanced technology on the rest of the project.

• It is useful to develop **conditional networks** where these corrective actions are applied automatically when needed.
Conditional Networks

• Conditional networks are created using special activities called Switches. Switch is a zero-duration activity that has two states: when the state is Yes, Spider Project selects certain links emanating from the Switch activity; when the state is No, Spider Project ignores these and selects other links and corresponding activities.

• Conditional networks may be used for what-if analysis or for modeling different ways of project execution that depend on certain conditions.

• Let’s look at the example in the next slide
Conditional Network Example

• Switch selects whether to use one or two resource units on the next job.

• If it happens before February 16, one resource is sufficient to finish on time but if the Switch activity is late, then two resources will be used and the project will be executed differently.
Resource Constrained Scheduling

• Resource leveling heuristic determines the rules by which the activities competing for the same overallocated resource will be delayed.

• Some packages use a single rule, others offer a choice of leveling heuristics to use for scheduling, still others employ sophisticated approaches using several rules or iterative processes for finding the best solution.

• Different packages create different resource constrained schedules for the same project. Therefore project planners must not blindly rely on the schedules offered by their tools and must look into these schedules for possible improvements.
Let's look at the sample schedule below. This schedule was obtained using Critical Path Method and resource A is overallocated. Activities 1 and 2 constitute the Critical Path, activities 3 and 4 have 11-day total float, but this CPM schedule is not feasible and needs adjustment.
Resource-Constrained Scheduling

The most commonly used heuristic for resource leveling is assigning maximum priority to activities with the least total float in CPM schedule. Applying this rule to our sample project, we get the following schedule. Project duration is longer but resource overallocation is resolved.
Resource Critical Path

• Activity **floats** in resource-constrained schedules must be calculated taking into account resource constraints on both the forward and backward passes.

• A sequence of activities with the least total float (usually zero) in a resource-constrained schedule is called **Resource Critical Path** or **Critical Chain**.

• In our example, it is the sequence of activities 1, 2, and 4 (and, of course, the Start and Finish milestones).

• When resources are not limited, Resource Critical Path is the same as traditional Critical Path, therefore Critical Path is a particular case of Resource Critical Path.
Resource-Constrained Scheduling

- But Resource-constrained floats and Resource Critical Path depend on the algorithms that have been used for creating resource-constrained schedule.
- Since different leveling algorithms may create different schedules for the same project, resource-constrained float is a characteristic of both the activity and the schedule. The same activity can be critical in one resource-constrained schedule and have a positive float in another.
- This must be of particular attention for project planners who use different project-scheduling tools and/or import project models from one tool to another.
In the optimal schedule for our sample project (below), Resource Critical Path consists of activities 1, 2 and 3. In this schedule, activity 4, that was critical in the earlier-shown schedule, has 2-day resource-constrained total and free floats.
Resource Critical Path

• Resource-constrained floats and Resource Critical Path must be calculated taking into account all existing constraints.
• Other constraints (material supply, funding, space) can also be taken into account if you use Spider Project leveling engine.
• Unfortunately, not every PM tool calculates correct resource-constrained total floats and Resource Critical Path.
Let's have a look at the resource-constrained schedule created for our sample project by Microsoft Project software:

You may notice that this schedule is not optimal and resource constrained total floats are wrong. Microsoft Project shows that activity 3 is critical despite having a 9-day free float and that activities 1 and 2 have 8-day total floats although actually they are resource critical.
Cost and Material Leveling

• Projects can have not only renewable resource constraints but also financial, material supply, and space constraints that must also be taken into account.

• Spider Project models not only the expenses and material consumption but also project funding and material supply.

• Project resource leveling can include cost and material leveling.

• An example on the next slide shows another Sample Project where cost leveling was applied.

• Materials are leveled similarly.
Before and After Cost Leveling

<table>
<thead>
<tr>
<th>Name</th>
<th>Duration, Days</th>
<th>Cost [Remain]</th>
<th>Financ Level</th>
<th>Jan 20</th>
<th>Feb 20</th>
<th>Mar 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Leveling</td>
<td>47.00</td>
<td>1 000</td>
<td>1 000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity 1</td>
<td>10.00</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity 2</td>
<td>25.00</td>
<td>350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity 3</td>
<td>12.00</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity 4</td>
<td>15.00</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity 5</td>
<td>20.00</td>
<td>150</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finish</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Financing</td>
<td></td>
<td>1 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transh 1</td>
<td></td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transh 2</td>
<td></td>
<td>350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transh 3</td>
<td></td>
<td>350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Duration, Days</th>
<th>Cost [Remain]</th>
<th>Financ Level</th>
<th>Jan 20</th>
<th>Feb 20</th>
<th>Mar 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Leveling</td>
<td>47.43</td>
<td>1 000</td>
<td>1 000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity 1</td>
<td>10.00</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity 2</td>
<td>25.00</td>
<td>350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity 3</td>
<td>12.00</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity 4</td>
<td>15.00</td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity 5</td>
<td>20.00</td>
<td>150</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finish</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Financing</td>
<td></td>
<td>1 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transh 1</td>
<td></td>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transh 2</td>
<td></td>
<td>350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transh 3</td>
<td></td>
<td>350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Financing & Expenses:
- Total cost: Before - 297.71, After - 367.50

Financing:
- Before: 1 000.00
- After: 1 000.00

Expenses:
- Before: 0.00
- After: 0.00
Space Constraints Leveling

• Spatial constraints are modeled using special material resource. But space is usually required by a group of activities rather than a single activity.

• Spatial resources are consumed at the start of the first activity of the group and are produced when the last of the group activities was finished.

• Spatial resource production enables another activity group to consume it, that is to occupy the released space.
Resource Constrained Scheduling

• A PM tool that calculates the shortest feasible resource-constrained schedule can save a lot of money to its users. Nowadays, however, a few planners pay attention to the quality of resource schedules offered by their tools, or prefer manual leveling that rarely produces the best result but is very time-consuming.

• After project schedule is approved, a project planner may want to maintain the scheduled order of activity execution. Automatic resource leveling may change this order when project data has been changed (e.g. after the actual data is entered). When changes are undesirable and it is necessary to maintain the approved order of activity execution, it is nice to have this option in the automatic resource leveling settings.
Resource Dependencies

• **Resource dependencies** are very useful for resource-constrained schedule analysis. They show why an activity is delayed for a longer time than required by activity logic dependencies.

• Activity A precedes Activity B with the resource dependency if activity B was delayed because a resource required for activity B execution was busy on activity A.

• Resource dependencies are created as a result of current schedule analysis and are reconsidered every time that the project is rescheduled.

• They are not the same as soft links people create when leveling projects manually and trying to fix resource conflicts.
Flex

• Total and Free Floats show whether an activity can be delayed without delaying the project’s finish or the scheduled dates of other activities.

• Other types of activity reserves are Start and Finish Flexes. Flex is a metric of activity duration flexibility, introduced by Rafael Davila.

• Activity Start Flex shows the time difference between the earliest activity start and the planned activity start in the current schedule without violating any constraints and without delaying activity finish.

• Activity Finish Flex shows the time difference between the latest activity finish and the planned activity finish in the current schedule without violating any constraint and without delaying activity start.

• Both metrics mean increasing activity duration by keeping activity start or finish intact.
Drag

• Activity Drag is the amount of time that an activity on the critical path adds to the project duration.
• Alternatively, it is the maximum amount of time that one can shorten the activity duration before it’s no longer on the critical path or before its duration becomes zero.
• Activity Drag is a metric developed by Stephen Devaux.
• Let’s look at the Drags and Flexes of activities of the sample project presented in the next slide.
Sample Project

This is a CPM schedule and the required resource quantity (8) exceeds their availability (7)
Sample Project

Sample Project after leveling – duration increased from 16.5 to 17 days
Sample Project

• Activity Drags show what will happen to the project duration if activity duration is zero.

• You can notice that for some activities Drags are negative, which indicates that minimizing activity duration makes the project duration longer.

• In this project, activities have zero Start Flex, but one of these has a positive Finish Flex. It means that we can delay its finish without impacting other activity dates.

• Let’s look at the Spider Project function called schedule adjustment. When we use it, Spider Project looks for the optimal project schedule by increasing activity durations with a simultaneous adjustment of assigned resource workloads.
Adjusted schedule

- Project duration has not changed but resource workloads on some activities are now less than 100% and we can adjust the number of resources required for activities execution.
Resource Adjustment

- After resource adjustment, the scheduled project duration is 13.25 days instead of 17 days.
Conclusions

• This example shows that project schedule optimization works should not be finished after the automated resource leveling even if your software creates the optimal schedules.
• Using Drags can help you find the best ways for schedule crashing,
• Using Flexes can help you find the way to do the same work with less resources,
• Schedule and resource adjustment can substantially improve your schedule without involving additional resources and costs.
Thank you

smt@spiderproject.com