Enhancing Resource-Leveling via Intelligent Scheduling: Turnaround & Aerospace Applications Demonstrating 25%+ Flow-Time Reductions

Rob Richards, Ph.D.
Principle Scientist
Stottler Henke Associates, Inc.
Background & Perspective

Stottler Henke

- Artificial Intelligence Research & Development
  - Software company

- Video: Project Management Experience
Resources & Critical Path (Resource Loaded)

- Large organizations developing and building complex systems rely on schedules and project management.
- Many CPPM projects are resource constrained (in reality, even if not modeled that way)
- Resource constraints (e.g., labor, space, equipment) greatly complicates the scheduling problem.
  - Hence a ‘reason’ to ignore
Where in the PM Space?

- Project Management
  - ...
  - Critical Path (Resource Constrained)
    - ...
    - Scheduling / Level Resources
      ⇔ ⇔
    - ...
  - ...

Scheduling Background / Comparisons

• Resource-Constrained Scheduling is NP-Complete, takes exponential time for optimal solution
  – I.e., it is a hard problem
  – Approximate methods are needed
• Most automatic scheduling systems use simple one-pass algorithms
• Standard constraint-based approaches are far less computationally efficient (Aurora takes advantage of structure of scheduling problems and heuristics)
Why Important? / Motivation

- So much work is put into developing project plan before hitting the schedule / Level Resources … button Days, Weeks, Months
- What if your resulting schedule is 10% longer than it needs to be because of the scheduling engine?
- Would you care?
How about 25+% longer?
Motivation: Visual

• Following figure shows.
  – Critical Path
  – Resource Constrained Critical Path (theoretically correct)

• The **goal** is the **shortest** correct schedule
Scheduling Engine Comparison
## Construction Examples
(Kastor & Sirakoulis, 2009)

<table>
<thead>
<tr>
<th>Product</th>
<th>1st Example</th>
<th>2nd Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Duration</td>
<td>Deviation from CPM (%)</td>
</tr>
<tr>
<td>Primavera P6</td>
<td>709</td>
<td>52.8</td>
</tr>
<tr>
<td>MS Project</td>
<td>744</td>
<td>60.34</td>
</tr>
<tr>
<td>Open Workbench</td>
<td>863</td>
<td>85.99</td>
</tr>
</tbody>
</table>
Different Resource-Leveling Techniques

• Deviation from Critical Path Duration
Benefits of Sophisticated Underlying Scheduler

• Results in a better initial schedule

• Execution: Schedule is more flexible and better able to accommodate change.
  – Schedule is “self-aware” of what tasks can most easily be moved. I.e., tasks store information about what placed it where it is placed.
  – Quickly reschedule as if resources on late task are not available until after its estimated end time.
Maybe Only for ‘Big’ Problems?

• Let’s look at a toy problem …
• ‘Simple’ problem with only 7 real tasks and 2 milestones.
‘Simple’ Network details

- Number superscript of circle is duration in days
- Number subscript of circle is resources needed
- There is only 1 type of resource
Critical Path of Network

- Solution when infinite resources available
  - Find longest path = 1 + 1 + 5 = 7
- So Critical Path is 7 days
Gantt Chart of Critical Path

- Note: Sat/Sun are not workdays
Set Resource Pool to 5

- Only one type of resource to make the problem ‘simple’
• Note: now some resources are overloaded
• Resource level to solve over allocation
Resource-Leveled in MS Project = 9 days
Simple Enough, Right?

- Another view of the solution
But there is a better solution … P6 Model: Resource Leveled = 8 days
Simple?

Critical Path = 1 + 1 + 5 = 7
1 resource
5 total units
End of Story… Not quite

• There is an even better solution
• 7 days
• So this ‘simple’ problem could not even be solved well by the world’s ‘premier’ project management tools.
• Can you solve this ‘simple’ problem in 7 days?
Constraints Add Complexity

- Technical constraints (E.g., F-S, F-F, S-F, lags)
- Resource constraints
- Labor constraints
- Usage constraints – e.g., tool can only be used for so many hours continuously and/or during a day.
- Spatial constraints – e.g.,
  - job requires a certain location or type of space;
  - two elements should (or should not) be next to each other
- Ergonomic constraints – individual limitations on work conditions
Visualizing More Complex Situations

- No good methods shown to date
- Closest way is by similar problems
  - E.g., Tetris game, Tetris cube
Tetris

- Shapes similar to resource profile of individual tasks
- Holes when playing Tetris represent resource allocation inefficiencies.
  - E.g., black regions in figure to the right
More realistic to scheduling multiple types of resources per task is the Tetris Cube. If not pieced together properly then will not fit in box.

Video
Refinery Turnaround
Leveraging Intelligent Scheduling Technology
Turnaround Project
Network 2,500+ Tasks
Results: 2,500+ Turnaround

- Primavera P6 67.125 days
  - Performed by 3rd party
- Aurora 56.27 days
- Primavera P6 19.3% longer than Aurora
- Critical Path is 46 days
  - P6 is 21.125 days longer than CP
  - Aurora is 10.27 days longer than CP
  - So % diff over CP is >100%
Long-Term Refinery-Related Upgrade

MS Project 2007 = 1,627 days
Primavera P6 = 1,528 days
Primavera P3 = 1,258 days
Intelligent scheduling (Aurora) = 1,240 days
300 Task Example: Aerospace Application

Multiple Resource Types Needed for most tasks
300 Task Example: Network in Aurora
Results: 300 Task Example

- MS Project 2003 145.6 days
- MS Project 2007 145.6 days
- Primavera P6 115 days
  - Performed by 3rd party
- Deltek Open Plan 110 days
- Aurora 102.5 days
Results

• Multiple sources reveal the effect of the Scheduling Engine
• For larger projects (>1,000): Aurora has been able to find project durations **SIGNIFICANTLY** shorter than other software for the same data set.
• Much of the potential improvement offered by modeling resources is being squandered.
• Resource leveled schedules are sub-optimal
Planning & Execution

- Initial Schedule benefits
- Execution benefits even MORE
  - If scheduler is inefficient, every delay will be magnified because re-allocation of resources will be deficient
Benefits of Sophisticated Underlying Scheduler

• Results in a better initial schedule

• **Execution**: Schedule is more flexible and better able to accommodate change.
  – Schedule is “self-aware” of what tasks can most easily be moved. I.e., tasks store information about what placed it where it is placed.
Analogy: Chess

Chess mathematically is similar to resource loaded scheduling.

– Easy: Create basic rules to play
– Hard: Win against other intelligent players

Resource Leveling in most software is analogous to 'Easy' chess solution

Each move analogous to execution mode update, challenge continues throughout game/plan
Take Aways

- Scheduling engine is critical
- Paying up to 100% penalty due to the scheduling engine
- Changing to an improved scheduling engine is probably the greatest potential improvement available to your project
  - Just press a different button
- Use more than 1 scheduling engine
Questions?

• Robert Richards, Ph.D.
  Stottler Henke Associates, Inc.
  Richards@StottlerHenke.com
Thank You For Attending!