Reduced Project Duration & Improved Critical Resource Determination via Intelligent Scheduling: Navy and Other Applications

Robert A Richards
Stottler Henke Associates, Inc.
1670 S. Amphlett Blvd., Suite 310
San Mateo, CA 94402
Richards@StottlerHenke.com

In complex production and maintenance environments, such as ship production, the method of allocating resources and managing other constraints significantly affects the efficiency of progress as well as the overall project duration. Resources include human resources, equipment resources and physical-space resources. Due to the inherent complexity of resource allocation and constraint management for such complex production and maintenance environments, the project durations can be two-times, or more, longer than needed. Unfortunately, most commercial project management software does not benefit from such intelligent scheduling technology. Software that does not leverage intelligent scheduling may also determine incorrect critical resources. A resource is critical if the project duration would be shorter if more of that resource were available. So not using intelligent scheduling will result in longer than necessary schedules and might direct users in the wrong direction per critical resources, thus leading to wasteful acquisition of more resources to shorten the schedule when this is an option.

KEY WORDS: computers in construction; maintenance; scheduling; shipbuilding; systems engineering; U.S. Navy

INTRODUCTION

When developing a project plan there are various ways of turning the goals of the project into a set of activities and an overall schedule. Usually one of the first steps after determining all the activities that must be performed as part of the project is to determine any and all time-based dependencies between activities (technical dependencies). That is, for any activity, all the activities that must be completed before that activity can start must be modeled. Once all the activities and the technical constraints are modeled for those activities, the next stage may be to insert a default duration for each of the activities as well as a start date for the project.

At some stage during the planning process, the issue of the *who*, *what*, and *where* for each of the activities has to be addressed. That is, for each activity: who is going to perform this activity, what equipment or tools, etc. are going to be needed to perform this activity, and where is this activity going to be performed; or put another way, what are the resources that are needed for each of the activities and how are these needed resources going to be modeled, if at all? The *who* is the most common resource; people are required for almost every activity (exceptions include curing processes that may require only space & time). The *what* may or may not be important to every activity; for example, if all that is required for the person to complete the activity is their computer, most likely that computer is available to them 24 hours a day and thus that resource is not normally modeled; however, many activities require equipment that is limited, such as forklifts, tractors, and possibly special tools that are shared among many people. The *where* can also become a very important resource because activities performed by people require space; for construction being performed in a room, for example, only so many people and equipment can fit in that room and certain activities may not be compatible simultaneously.

So the person who is modeling the project has to decide how these resources are going to be handled. Many times they are not explicitly modeled, but a scheduler may realize that a critical piece of equipment cannot be used in parallel and may implicitly model such resource constraints by putting in technical links for certain tasks that use that resource, so that that resource is not overburdened. This may result in an overly constrained model of reality, which may result in a flow that is longer than would be possible if the scheduler had handled the situation via a fully resource-loaded schedule

There is a trade-off between how complex the model is and how many resources are actually in that model and the potential benefits of using a more complex model. As the model becomes more complex/realistic, then the project management software can be used to greater effect to verify there are no conflicts. Furthermore, it can consider all the resource constraints to *potentially* develop an efficient overall schedule that realistically models the real-world situation. For example, it is easy to develop an original model and then discover (when resource requirements are modeled) that the currently available resources at certain junctures in the project are not sufficient, so one or more of the tasks at these junctures will have to be rescheduled to manage the constraint. In addition, the project management software can also provide graphic depictions of the resource allocations across the project, and from this information it may be easy to discover that by increasing the number of a few inexpensive resources many bottlenecks can be eliminated. Again, when using project software (e.g., Microsoft Project), resource leveling means resolving conflicts or over allocations in the project plan by allowing the software to re-arrange tasks automatically to resolve the conflicts. Unfortunately, *the challenge of resource leveling is non-trivial*.

Let's return to the non-resource constrained situation. In this case the scheduling engine needs to take into account all the technical constraints when determining the schedule. In the mathematical sense this problem is solvable and every project management software package should output the same result. However, once resources are introduced the problem becomes much more complex. This can be understood intuitively by considering all the resources that could be required to complete an activity. A single real-world activity could require multiple people each needing specific skills, each of the people may need to have access to specific pieces of equipment which are in limited supply, and furthermore, the space where the activity occurs is shared by other activities so this activity cannot occur when some or all of those other activities are occurring. There could be other types of constraints that may need to be considered also. It is obvious that the resource constrained situation is significantly more complex than the purely temporal case. Mathematically, the resource-constrained project scheduling problem is NP-hard (nondeterministic polynomial-time hard). This means that there is realistically no way to guarantee that the result provided is the optimal result.

So in complex production and maintenance environments, such as ship production, the method of allocating resources and managing other constraints significantly affects the efficiency of progress as well as the overall project duration. Due to the inherent complexity of resource allocation and constraint management for such complex production and maintenance environments, the project durations can be two-times, or more, longer than needed.

It is likely that *most users of commercial project management software are NOT aware that the results from the resource leveling process are not optimal, and could be improved upon significantly*. It is ironic or at least disappointing that project teams that have put in the significant effort and cost to create a resource-constrained model could reap huge time and cost savings simply by running their already built model through different scheduling engines.

Figure 1 illustrates the potential impact of different scheduling algorithms on a resource-loaded project network. It includes the *critical path* for reference (i.e., the schedule assuming infinite resources) via white (non-filled) boxes. Also shown is the *resource-constrained critical path* (RC-CP) of the same project schedule when taking into account limited resources. The only difference in determining the schedules is the actual scheduling algorithm. When a less efficient scheduling method is used, the unnecessarily long schedule (shown with darker (blue) filled-in rectangles) will give an erroneous impression of the time in which the project could potentially be completed. The schedule with lighter (orange) filled-in rectangles is a more efficient RC-CP schedule. The *only* difference was the scheduling engine applied to the problem.

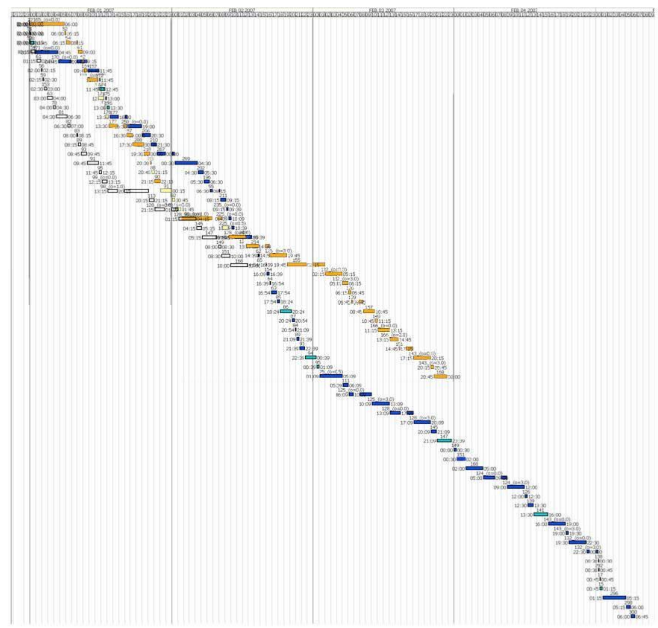


Fig. 1: Comparison of Resource-Constrained Critical Paths

Software that does not leverage intelligent scheduling may also determine incorrect critical resources. A resource is critical if the project duration would be shorter if more of that resource were available. That is, one project management software might indicate that more of resource x is needed if the project duration is to be shortened, while a more intelligent scheduler might not only reveal a shorter schedule but may determine that resource x is not critical (although some other resource(s) is/are). So not using intelligent scheduling will result in longer than necessary schedules and might direct users in the wrong direction per critical resources, thus leading to wasteful acquisition of more resources to shorten the schedule when this is an option.

The following sections will provide more details on the challenge of resource constrained scheduling and the benefits of intelligent scheduling.

Most commercial Project Management (PM) software, such as Microsoft Project and Primavera P6, provides a *resource-leveling* capability with graphical support to assist users in better understanding resource usage and to optimize resource utilization *by hand*.

The goal of *resource-leveling* in PM software is to provide the user with *a* valid resource-loaded schedule that does not have any over-allocated resources. Most PM software does *not* try to optimize the allocation of resources in order to generate the shortest resource-leveled schedule. Even though all PM software does not purport to provide an optimized schedule, it is likely that many users of PM Software are NOT aware that the results from the resource-leveling process are not optimal, and could be improved upon significantly.

So in this paper,

- *Resource-leveling* will refer to the functionality provided commercial project management software, and
- *intelligent scheduling* technology will refer to resource-constrained scheduling that attempts to optimize the utilization of resources to minimize the project duration.

This paper shows that the scheduling method used affects the project duration even for relatively small projects consisting of a few dozen tasks, and the effect becomes more pronounced as the number of tasks grows and number of types and quantity of resources expand. Some of the literature on this topic reviewing different techniques and results, which reveals the major difference in schedule duration due to the scheduling method, is also reviewed. Many lessons will be drawn from Stottler Henke's own work in this area. Since the early 1990s, Stottler Henke has been working with NASA, Boeing, Bombardier, the US Navy and other companies to develop intelligent scheduling technologies and software using artificial intelligence and other techniques. Lessons learned from experience at the Naval Submarine Support Facility (NSSF) and at Boeing per the manufacture of the B787 Dreamliner are provided. Without adding one extra resource, an entire project can be shortened significantly; therefore the application of more-efficient resource allocation and constraint management methods can improve the planning & production process, enhancing the productivity of ship production.

The following sections take real-world examples analyzed by the author in the oil refinery turnaround and aerospace domains. To complement this, results from the literature are also included to emphasis the fact that these differences are found in all domains for all types and sizes of projects. In addition to the efficiency that can be realized by utilizing intelligent scheduling, the following sections show the significant effects of using different resource-leveling techniques found in different PM software.

The main thrust of this paper is to

demonstrate and explain the differences,

not try to rank specific software. Part of this is because different techniques will show different results when applied to different problems. So it would not be objective to draw any conclusions regarding the efficiency of different PM software from such a small sample. However, it is educational to have a reference as a baseline for comparison. The obvious choice is Microsoft Project (MS Project) as it is so widely known and almost all the literature on this topic includes MS Project results.

WHY IS SCHEDULING DIFFICULT? A SIMPLE DIFFICULT ILLUSTRATION

To illustrate the difficulty of resource-constrained scheduling, a small project network will be used. It is fortunate that these effects can be seen at this scale, because, due to the inherent complexity of the resource-constrained scheduling problem, it is difficult/impossible to visualize what is occurring for larger networks. Figure 2 displays the illustrative network (Demeulemeester et al., 1994).

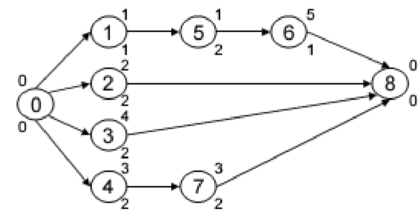


Fig. 2: Illustrative Network

The information in Figure 2 (on the left) is defined as follows:

- Activity/Task name/number: # inside circle
- Activity/Task duration in days # above node
- Resource units required: # below node.

The Critical Path (i.e., scheduling assuming infinite resources) is seven (7) workdays as shown in the Gantt chart in Figure 3 (a five-day work week is assumed in the illusration).

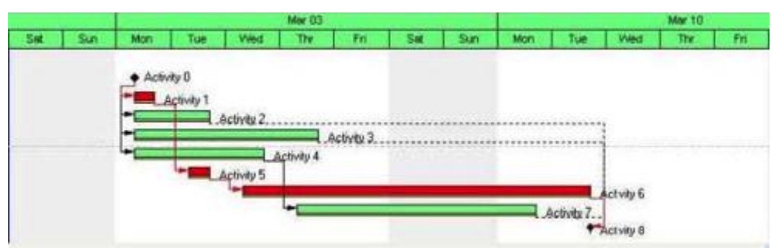


Fig. 3: Gantt Chart Showing the Critical Path (Assuming a Five-day Work Week)

Next a resource limit is set.

- 5 units of resource available.

Resource-leveling in Primavera P6 results in the following, see Figure 4.

Fig. 4: Example Results of Resource-Leveling

So, the resource-leveled schedule can be 8 days.
However, if this same problem is resource-leveled in MS Project 2003 or 2007 the result is 9 days (Leus, 2004, and confirmed by the author) as shown in Fig. 5:

Compare this to the Primavera P6 result that takes 8 days, shown in Figure 7 below.

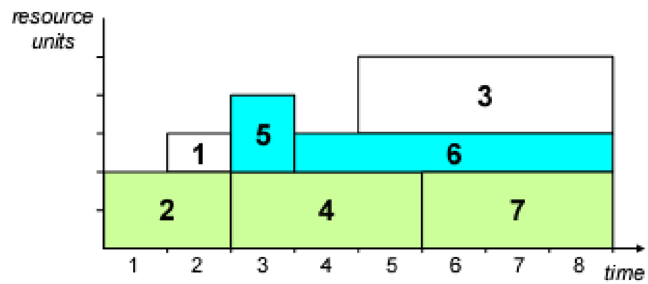
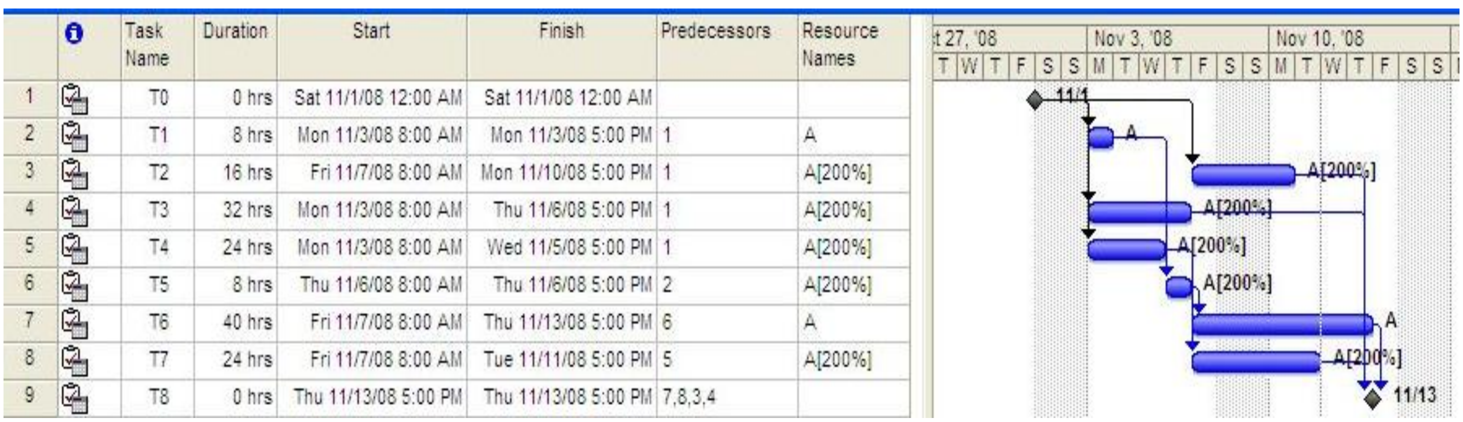


Fig. 7: Primavera P6 Scheduling Solution

Notice how different the results are.

Since the problem is small enough, the actual globally optimal schedule can be found, and the minimum resource-loaded project duration is 7 units of time.



	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	T0	0 hrs	Sat 11/1/08 12:00 AM	Sat 11/1/08 12:00 AM		
2	T1	8 hrs	Mon 11/3/08 8:00 AM	Mon 11/3/08 5:00 PM	1	A
3	T2	16 hrs	Fri 11/7/08 8:00 AM	Mon 11/10/08 5:00 PM	1	A[200%]
4	T3	32 hrs	Mon 11/3/08 8:00 AM	Thu 11/6/08 5:00 PM	1	A[200%]
5	T4	24 hrs	Mon 11/3/08 8:00 AM	Wed 11/5/08 5:00 PM	1	A[200%]
6	T5	8 hrs	Thu 11/6/08 8:00 AM	Thu 11/6/08 5:00 PM	2	A[200%]
7	T6	40 hrs	Fri 11/7/08 8:00 AM	Thu 11/13/08 5:00 PM	6	A
8	T7	24 hrs	Fri 11/7/08 8:00 AM	Tue 11/11/08 5:00 PM	5	A[200%]
9	T8	0 hrs	Thu 11/13/08 5:00 PM	Thu 11/13/08 5:00 PM	7,8,3,4	

Fig. 5: Resource-Leveled in MS Project 2003 or 2007

Looking at the Gantt charts you can see that different programs lay out the tasks quite differently.

Figure 6 provides another way of looking at the MS Project results.

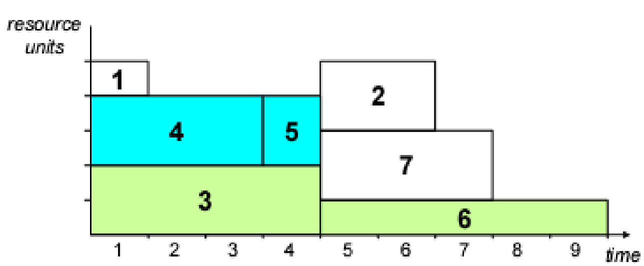


Fig. 6: Alternate View of MS Project Results

Observe how the problem is represented in Figure 6 and 7; it looks similar to a puzzle. A reader is encouraged to see if they can find the optimal solution. This should cerebrally illustrate the inherent difficulty of resource schedule optimization.

This illustration should hint at the level of complexity that occurs as many more different types of resource constraints are introduced. For example, in many domains, such as ship construction and maintenance as well as in aircraft assembly there can be multiple resources per task. In many cases physical space becomes a limited resource, i.e., only so many workers will fit in a given area, and some actions may permanently eliminate possible workspace, thus physical space becomes an important resource that needs to be managed.

Critical Resource

This illustration also demonstrates the error in critical resource determination. The Critical Path is 7 workdays, the resource units as 'limited' to five (5). The results from Primavera P7 and MS Project imply that this resource was critical because they determined resource-constrained durations longer than 7 workdays. That is, one way to shorten the schedule duration would be to try increasing the number of resources available to more than 5 and then re-calculate the schedule, if this was an option to try to get the duration back down to 7 workdays.

Since the project *can* actually be performed in 7 workdays using five units of the resource, the resource is NOT critical. So these tools not only gave inefficient results they incorrectly imply a critical resource.

CONSTRUCTION EXAMPLES FROM LITERATURE

Kastor & Sirakoulis (2009) have run some resource-leveling comparisons in the construction domain.

Housing Project

The first series of tests is taken from a real housing project, which consisted of 96 houses. The focus was on the concreting of these 96 houses. The project has 98 activities, and one (1) renewable resource (concrete) was considered.

In the table below are displayed the results ordered by efficiency. The duration and percentage deviation longer than the infinite resource critical path (CPM) are shown.

The Primavera P6 has many settings that can be changed to potentially get better/different resource-leveling results. Here are all the results from Kastor & Sirakoulis (2009) for MS Project, Open Workbench and Primavera P6 having different settings. Random three or four letter descriptors are provided for the various settings so the reader can cross-reference the same settings across the two series/instances.

Rule	**1st Instance Duration**	**Percentage deviation from CMP (%)**
Primavera P6 LST	709	52.80
Primavera P6 Default	709	52.80
MS Project Standard	744	60.34
Primavera P6 PWM	744	60.34
Primavera P6 LFT	744	60.34
Primavera P6 EPWM	823	77.37
Primavera P6 MSLK	823	77.37
Primavera P6 SPT	893	92.46
Open Workbench	863	85.99

Note again how simple the resources are, only 1 resource and the variability between the results is significant.

Shopping Mall Project

The second series of comparisons is taken from the construction of a shopping mall consisting of 19 buildings. The schedule consisted of 668 activities and 7 renewable resources.

Here are all the results from Kastor & Sirakoulis (2009) for MS Project, Open Workbench and Primavera P6 having different settings. Random three or four letter descriptors are provided for the various settings so the reader can cross-reference the same settings across the two series/instances.

Rule	**1st Instance Duration**	**Percentage deviation from CMP (%)**
Primavera P6 LST	308	29.41
Primavera P6 Default	308	29.41
MS Project Standard	314	31.93
Primavera P6 PWM	319	34.03
Primavera P6 LFT	319	34.03
Primavera P6 EPWM	308	29.41
Primavera P6 MSLK	327	37.39
Primavera P6 SPT	336	41.18
Open Workbench	832	249.58

Drawing an approximate curve fit of the average of these results; Figure 8 gives an insightful figure showing how inefficient the results can get by just using an inappropriate resource-leveling technique. This is not even considering the benefits of intelligent scheduling.

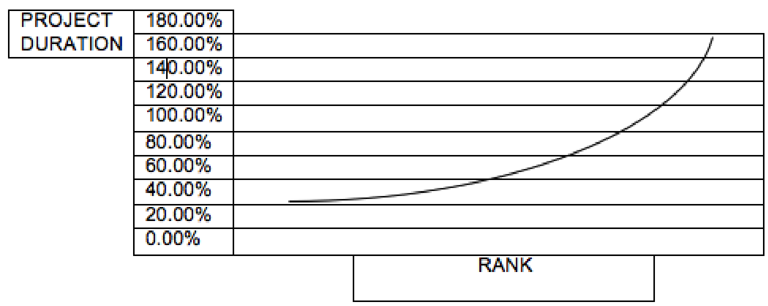


Fig. 8: Performance differences for different resource-leveling techniques, showing % increase from Critical Path

REFINERY TURNAROUND EXAMPLE

This section considers the analysis of a real refinery turnaround project (See Figures 9 & 10).

1
1
2
2
2
3
3
4

Fig. 9: Refinery

The project network consists of over 2,500 activities. A view of the network is shown in Figure 10. Note the the red lines link tasks with Finish to Start constraints, this network also has some start-to-start constraints that are shown with yellow lines, some may be seen in the upper-left portion of the network shown in Figure 10.

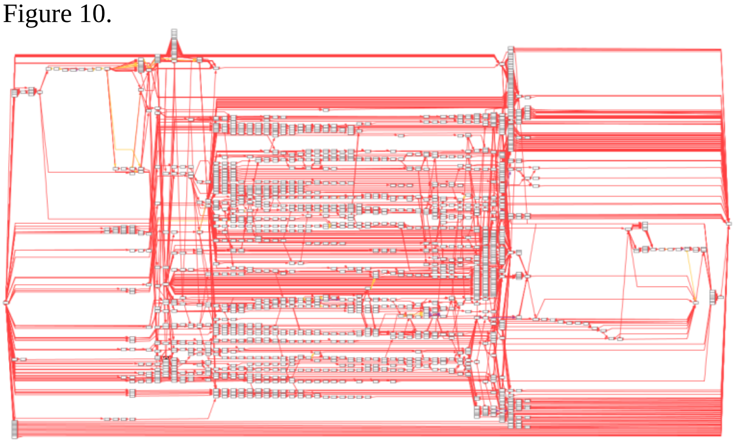


Fig. 10: Turnaround Project Network

The results of the analyses are as follows (note that no MS Project 2007 results are provided because the author could not successfully resource level this project in MS Project 2007):

Primavera P6 resource-leveling	**67 days, 3 hours**
Intelligent Scheduling	**56 days, 6.5 hours**

The difference in absolute terms is over 10.5 days. There are a few ways to compare these results; the simplest is to simply compare overall durations, using the intelligent scheduling results as the basis:

> Primavera P6 resource-leveling is **19.3% longer** than intelligent scheduling (67.125 - 56.27) / 56.27.

Using the Primavera P6 resource-leveling as the bases:

> Intelligent scheduling is **16.2% shorter** than Primavera P6 resource-leveling (67.125 - 56.27) / 67.125

Another valuable perspective lies in comparing the resource-constrained result with the Critical Path, that is, the situation assuming unlimited resources. Why is this perspective valuable? Because the Critical Path is the best case scenario, and the valid schedule when considering resources must always be longer than the Critical Path, so the length longer than the Critical Path is the only portion of the total project duration that the resource-leveling or intelligent scheduling can affect.

The Critical Path for the refinery turnaround project is **46 days**.

Primavera P6 resource-leveling results longer than Critical Path: 21.125 days
Percent longer than Critical Path 45.9 %

Intelligent scheduling results longer than Critical Path: 10.27 days
Percent longer than Critical Path 22 %

The percent difference between days more than Critical Path for Primavera P6 versus intelligent scheduling is
105.70%.

These results demonstrate the significant benefit of leveraging intelligent scheduling. Recall that everything besides the method for scheduling is the same in both cases. Leveraging intelligent scheduling saved over 10.5 days, and all of the associated costs with all the resources that are needed, as well as the lost revenue from the refinery being unavailable.

Of course the cost savings and other benefits of leveraging intelligent scheduling are huge for the initial plan, but even more potential benefit comes in the *execution phase* of the project, where unexpected circumstances need to be dealt with. By leveraging intelligent scheduling, rescheduling can be done quickly and the updated schedule will be shorter than if one used resource-leveling only. Therefore, every time a re-schedule is performed, the overall benefit of leveraging intelligent scheduling increases.

NAVY & AEROSPACE EXAMPLES

= 67.125 days

This section draws lessons from the ship domain, both airship (i.e. airplane) and submarines. Stottler Henke is applying intelligent scheduling to both of these domains and has found them to be extremely complex from a resource scheduling perspective, however, the domains are similar in the types of challenges. To be specific, Stottler Henke has implemented and continues to implement intelligent scheduling at multiple aerospace firms, including Boeing and for the Navy per submarine maintenance at the Naval Submarine Support Facility (NSSF).

In this ship domain resources include

- human resources,

- equipment resources and
- physical-space resources.

Human resources alone can be quite complex; for example, tasks may require multiple individuals, each may need to have certain occupations (such as a mechanic), and there may be additional requirements for some or all the individuals to have additional specific skills/certifications. Added to this are a multitude equipment resources, ranging from shop machines to massive cranes. In ship production / maintenance, and aerospace, physical space is also a limiting resource; for example, when conducting submarine maintenance, physical-space limitations are common, as well as the physical-space limitation of submarine ingress and egress. As if these considerations were not complex enough, other constraints are present, such as constraints on certain types of work (e.g., hot work), when other conditions are in effect.

Commercial Airplane Project

The example used in this section is drawn from a Boeing aircraft assembly process. The entire assembly process consists of multi-thousands of tasks, and most tasks have a multitude of resource constraints like those described above. A sub-project of the entire project, which will be used for the analysis discussed in this section, consists of about 300 tasks.

In order to utilize this project with commercial off-the-shelf (COTS) project management software it had to be simplified somewhat because of details that could not be modeled in some PM products. This simplified model was scheduled in MS Project 2007, Primavera P6 and *Aurora,* Stottler Henke's intelligent scheduler and PM tool. The following is a summary of the results:

Intelligent Scheduler	= 102.5 days
Primavera P6	= 115 days
MS Project 2007	= 145.6 days

The

> MS Project 2007 resource-leveling is **42% longer** than Aurora's intelligent scheduling.

Primavera P6 did much better than MS Project. However there is still a significant difference between these results and what is possible utilizing intelligent scheduling. The

> Primavera resource-leveling is **12% longer** than Aurora's intelligent scheduling.

As is obvious from this (simplified) real-world case, the differences in the scheduling results are huge. As more and more of the entire assembly process is modeled, the disparity between resource-leveling and intelligent scheduling only increases. As shown with all the examples, different commercial project management tools calculate different results and those results may be far from what could be calculated with current intelligent scheduling technology.

For more complex problems the critical resources and the near critical resources can become complex. For example, even if two software packages show the same critical resource, they almost certainly calculated different schedules and thus different resource allocations. Thus the effects of increasing the quantity of a critical resource will have different changes to the resulting schedule in each software after it reschedules with the increase in the critical resource. These characteristics are seen in the Boeing aircraft assembly process project when scheduled with different software packages.

CONCLUSIONS

In this paper we have shown that resource-constrained schedules and therefore resource-constrained project management is greatly affected by the underlying scheduling engine – more so as the project becomes larger and includes greater numbers of resource requirements and other non-technical constraints. From the literature and our experience there are situations where projects using these commercial tools could benefit significantly from intelligent scheduling technology and/or different resource-leveling techniques.

We have demonstrated the effect the scheduling engine can have on the resulting schedule. The primary conclusion is that the underlying scheduling engine can greatly impact the results. History has proven that it is so far impossible to build a scheduling solution that is best in all situations, so a beneficial approach would be to

> maintain a pool of possible scheduling engines or engine configurations, and apply all of them to projects.

Because of their differing strengths and weaknesses, some would perform more effectively in some situations than in others. Once all had been applied to a given project, the best engine for the purpose could then be selected for subsequent resource-constrained critical path application during the execution phase. If possible, the best solution could then be further tailored to maximize the benefit if it was customizable, but the key point is that the scheduling system has a significant impact on a project and should be given corresponding consideration.

If increasing resources is an option, different software can show different resources as being the critical resource. In the worst case software that performs poorly at scheduling may show a resource as critical when if a better scheduling technique was applied that resource might be found to not only be not critical, but increasing that resource would not even shorten the schedule.

Consider the amount of work that is put into developing a project network: days, weeks, or months, before selecting the resource-level option. Presently there may be significant amounts of time and effort put into optimizing the results of the resource-leveling results in order to derive a shorter project Now, with a trivial amount of additional effort after the network development, a *significantly* shorter duration project can be calculated automatically, initially saving substantial amounts of time and effort per hand optimizing the initial results, then, even more importantly, during the execution of the project.

REFERENCES

Demeulemeester, E., Herroelen, W.S., Simpson, W., Baroum, S., Patterson, J.H. & Yang, K.-K. (1994). On a paper by Christofides et al. for solving the multiple-resource constrained, single project scheduling problem. *European Journal of Operational Research, 76*, 218-228.

A. Kastor, K. Sirakoulis (2009). The effectiveness of resource levelling tools for Resource Contraint Project Scheduling Problem. *International Journal of Project Management, 27*, 493-500.

Leus, R. (2004). The generation of stable project plans. *4OR, the Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, *2*(3), 251-254.