Increased Submarine Production and Maintenance Throughput via Intelligent Scheduling

Robert Richards
Stottler Henke Associates Inc. (SHAI)
1650 S. Amphlett Blvd., Suite 300
San Mateo, CA 94402
Richards@StottlerHenke.com

Submarine production has benefited significantly from more intelligently allocating resources and managing other constraints, thereby increasing efficiency and reducing overall project duration. Intelligent scheduling is being leveraged by General Dynamics Electric Boat (EB) for the scheduling of various aspects of submarine construction, to increase the speed of production. Scheduling is an NP-complete problem, that is the size of the solution space grows exponentially as the size of the project grows linearly and therefore problems of any reasonable size cannot be solved simply mathematically. Unfortunately, most commercial project management software does not benefit from intelligent scheduling technology. Most 'solutions' such as resource leveling greatly simplify the problem and thus result in far suboptimal results. Intelligent scheduling includes a strategy that leverages scheduling heuristics learned from many of the world's best human schedulers, including those involved in submarine scheduling, in order to solve complex scheduling challenges in reasonable amounts of time (minutes).

The goal of scheduling is to optimize the synchronization of resources and other constraints to minimize the duration of the project. For submarine production and maintenance, resources include human resources, equipment resources, and physical-space resources. This paper reviews some of the literature on this topic showing different techniques and results showing the major difference in schedule duration due to the scheduling engine. Real-world experience from EB is provided to further illustrate the real-world impact, and lessons learned. Thus, without adding one extra resource, submarines are being produced by EB more rapidly just by utilizing better scheduling technology.

INTRODUCTION

Scheduling, at its most basic, is the process of assigning tasks to resources over time, with the goal of optimizing the result according to one or more objectives [1]. Scheduling is heavily used in ship and submarine production to minimize the time and cost associated with the completion or production of small to large, simple to complex projects. The Aurora scheduling framework is one example of a general-purpose intelligent scheduler that has been successfully applied to a variety of domains [2], [3], including submarine production. Intelligent scheduling combines graph analysis techniques with heuristic scheduling techniques to quickly produce an effective schedule based on a defined set of tasks and constraints [4]. This typically includes the following:

Temporal: Tasks must be scheduled between the project start and end dates; each task has duration and an optional start date and an optional end date.

- Calendar: Tasks can only be scheduled during working shifts; tasks cannot be scheduled on holidays.
- Ordering: Tasks can optionally be assigned to follow either immediately after/before another task or sometime after/before another task; optionally with a specific offset time in between.
- Resource: Each task can require that resources be available for the task to be scheduled.

The framework distills the various operations involved in creating a schedule that respects all of these constraints into reconfigurable modules that can be exchanged, substituted, adapted, and extended. This framework is used as a foundation to create domain-specific scheduling tools that respect the constraints specific to domains. Additionally, heuristics are tuned on a domain-specific basis to ensure a high-quality schedule for a given domain.

The scheduling framework consists of two primary components: the engine and the user interface. Both components may be customized to create a domain-specific scheduling tool.

This paper describes lessons learned from working on some of the world's most complex scheduling challenges, specifically submarine production.

HEURISTICS: IMPORTANCE OF

Scheduling is an NP-complete problem, that is the size of the solution space grows exponential time and therefore problems of any reasonable size cannot be solved simply mathematically. Most 'solutions' such as resource leveling greatly simplify the problem and thus result in far suboptimal results. Stottler Henke has employed a strategy that includes leveraging scheduling heuristics learned from many of the world's best human schedulers in order

to solve complex scheduling challenges in reasonable amounts of time.

Consider the following extremely simple example (which is therefore easier to use to illustrate this point) where:

- Three activities, called Activity 1, 2, and 3, from three different orders are all competing for time on similar machines in a particular work center.
- The priority is highest (or the due date is soonest) for Activity 1 and lowest for Activity 3.
- Two different machines exist, A which is expensive and precise and B, which costs less and has higher throughput.
- Machine A is required for Activity 3, but it can also process activities 1 and 2, though it is not efficient to do so.

Let's look at a solution from a simple scheduler: Activity 1 is chosen first for assignment, since it has the highest priority, and it so happens that at the moment Activity 1 can begin, only Machine A is available, so Machine A is assigned to Activity 1. Activity 2 is assigned to Machine B, which has become available soon after Machine A. Activity 2 is soon completed, owing to Machine B's fast production rate. When Activity 3 is finally examined, its required machine, Machine A, is busy, and worse, busy on an activity that it wasn't essential for. Meanwhile, Machine B is idle.

Obviously, this is a suboptimal solution since a different assignment would have prevented Machine B from being idle and prevented expensive Machine A from being assigned to a task that didn't need it. Of course, a more complex scheduler could "look ahead" to see if the cheaper machine might be soon available, but for any such workaround there's a corresponding example that still causes problems. And each of these rules has to be anticipated and created by the scheduling system software developer.

Perhaps a scheduling system could be written that systematically tried every possible solution and selected the best, and therefore optimal, one. In the example above, the number of possible solutions is 2 choices for Activity 1 times 2 choices for Activity 2 times 2 choices for Activity 3 = only 4 possible solutions. However, consider an activity list consisting of only 30 simple resource assignments where (for simplicity's sake) only one resource is required for each activity. Assume on average 4 meaningfully distinct choices (e.g. different machines) for each activity. This means that there are 30 distinct decisions with 4 choices each, so the number of solutions is 4 x 4 x 4 .... x 4 =

$$4^{30} = \text{over a million trillion possible solutions,}$$

which are clearly impractical to systematically search. This calculation was based on an extreme over simplification, the more realistic, and complicated planning problem is much more difficult. This is the essence of NP-Complete problems. The widely recognized and clearly applicable NP-Completeness Theorem states that to guarantee an optimal solution to an NP-Complete problem requires exponential time (e.g. M^N where M is the average number of options per choice and N is the number choices) which is clearly impractical in this case, since N is typically in the thousands. An optimal solution can simply not be guaranteed for this application.

Therefore, to determine near-optimal solutions in reasonable timeframes requires good heuristics learned from actual human experts on a large number of situations. We have developed both general heuristics for producing good solutions and the techniques and architecture to incorporate domain specific knowledge and heuristics into the planning system. Our expertise includes substantial experience eliciting the required knowledge and cognitive processes from expert planners, then mimicking those processes in software to create advanced intelligent planning and scheduling systems. To wit, Aurora mimics the *decision-making process of expert schedulers*.

FLEXIBLE / RECONFIGURABLE ARCHITECTURE

To achieve maximum flexibility, we designed Aurora to have a number of components that could be plugged in and matched to gain varied results. The scheduling system permits arbitrary flexibility by allowing a developer to specify what code libraries to use for different parts of scheduling. Each of the pluggable components must extend the corresponding general base class that defines the entry-point methods. This allows the objects that are integral to Aurora to interact with them successfully. The libraries may make use of any of the Aurora objects (such as activities and resources) that pass through the interface. These objects provide support for additional attribute caching, permitting domains to make use of custom properties in the scheduling heuristics. The primary pluggable components include a preprocessor; a scheduling queue prioritizer; the actual scheduler, which usually applies several scheduling methods; a conflict solution manager; and a postprocessor. See Figure 1 for a more detailed breakdown of configurable operations.

From this reconfigurable Aurora architecture, we have been able to build quite varied complex and successful scheduling systems; accomplishments range from scheduling the downlinks of US Air Force satellites [5] [7] & scheduling related to space debris tracking [6], to scheduling medical residents during their education at Harvard's Medical School, to scheduling the final assembly of the Boeing 787 jetliner and various other aircraft for Boeing, as well as similar operations for Bombardier and Learjet, to combining intelligent scheduling with Critical Chain Project Management (CCPM), to scheduling the manufacturing facilities of pharmaceutical production.

Further details regarding some of these accomplishments and lessons learned from the experience are provided in the sections below.

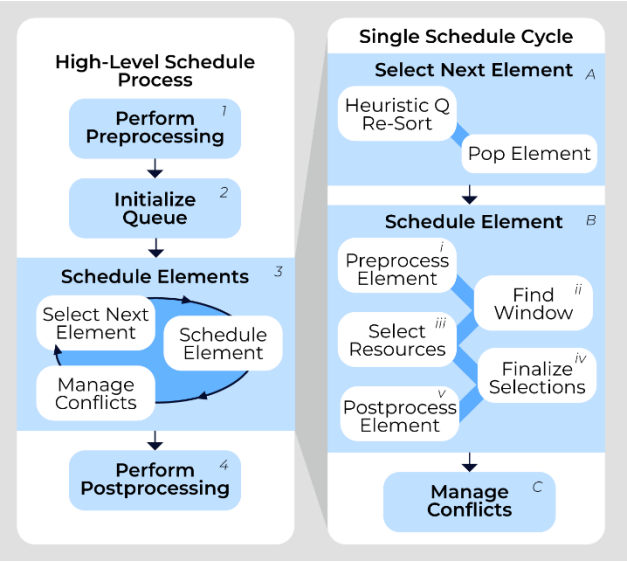

Figure 1. Aurora's reconfigurable scheduling system process breakdown.

GENERAL DYNAMICS ELECTRIC BOAT

Aurora is being leveraged by General Dynamics Electric Boat (EB) for the scheduling of various aspects of submarine construction, to increase the speed of production. To help maximize efficiency, customizations have been provided to further benefit EB and to provide greater efficiency to the users of Aurora, that is, the user interface has been adapted to make the EB specific use cases even more streamlined.

Electric Boat needed a project management and scheduling tool that could not only support the large models and complex constraints and resource requirements found in submarine construction, but additionally wanted the option of performing of Critical Chain Project Management (CCPM). Aurora's critical chain capability was originally developed for Boeing for aircraft manufacturing which faces many similar scheduling complexities found in submarine production. Due to the complex project management and scheduling challenges found at Boeing, the resulting product CCPM implemented in Aurora has resulted in Aurora being the world's most capable critical chain software solution.

Here are some of the powerful and many times unique capabilities of Aurora that Electric Boat leverages:

- Multiple-pass intelligent resource-constrained scheduling – This intelligent resource leveling tool results in shorter project schedules than the single-pass resource leveling option provided in Microsoft Project and Primavera P6.
- Mixed-mode scheduling – Aurora provides both forward and backward scheduling, available on a task-by-task basis.
- Schedule Rationale – For each task, Aurora provides the rationale/explanation on why it was scheduled where it was scheduled, so it is easy to determine what changes could be made for a task to occur earlier.
- Interface with other scheduling tools – Since the benefits of scheduling are only a mouse click away, Aurora is designed to interface with Microsoft Project, Primavera P6, Artemis, TeamCenter and others. You do not have to change the way you work to receive the benefits of Aurora's scheduling engine.
- 64-bit version can handle projects into the hundreds of thousands of tasks.
- Ability to run how the client wants to run. Aurora can be run as a web-based application or a standalone application under Windows, Mac and Linux.
- Supports more types of constraints beyond finish-to-start, start-to-start, finish-to-finish & start-to-finish, including:
  - physical space constraints, including taking into account the creation and elimination of the space during the project,
  - ergonomic constraints,
  - shift-based constraints,
  - resource Links — graphically depict when resource availability is driving the start date.
- Supports complex human classifications
  - E.g., occupation plus various skills and or certifications, and Aurora optimizes taking these into consideration.
- The ability to leverage knowledge about resource constrained task placement during execution. That is, during execution tasks many times start and complete at different times than calculated during scheduling, therefore resources may become available for a task that was originally scheduled later but could be done now, Aurora understands the details of the schedule and finds these opportunities, thus shortenkng the execution and utilizing resources that otherwise would lie idle. Thus, Aurora determines in real time what is best to work on to minimize project execution time.
- When performing Critical Chain:
  - Ability to take variability of tasks in a chain into account in buffer consumption. That is, if a chain consists of a series of low variability tasks at the beginning then a few high variability tasks at the end of the chain, standard buffer consumption reports could give an overly optimistic view of the situation.
  - Ability to handle short-duration tasks, and update buffer reports on any timeframe (e.g., once every hour).

Thanks to Aurora's modeling capabilities, GDEB now has a tool that allows them to see the level of detail necessary, so the model of reality reacts correctly to the actual reality, resulting in a level of trust, so GDEB is providing the best path forward. An example of the need for modeling human

resources with details beyond just an occupation, such as occupation plus a set of specializations and/or certifications, includes specializations that certain welders have. For example, there may be a resource set of welders, all of whom can perform Shielded Metal Arc Welding, then there may be subsets that can also perform Gas Tungsten Arc Welding, there can also be different levels such as apprentice or master. One welder may fall into many different subsets and to make a different resource set by hand for each and maintain this is overly complicated. It is better to have a dataset with the welders and the skills and let Aurora manage the details and allocate the welders optimally.

One of the unique and powerful capabilities in Aurora is the explanation facility. Aurora provides an explanation capability that shows the rationale for why every task is scheduled where it is, that is, each task includes the reasons why it is scheduled at its current time. This is a powerful capability that provides transparency into why the schedule is scheduled the way it is and builds trust by the users. Figure 2 shows a sample explanation. What is usually seen is that the start date may be affected by a start-no-earlier than constraint, then the start date may be later due to one or more predecessors not completing until later, and then finally the actual scheduled start date may be further delayed due to a resource not becoming available until after all the predecessors have completed.

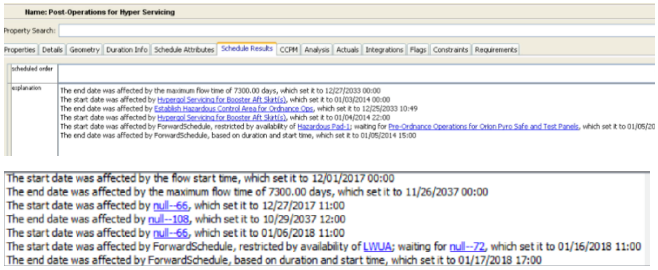

Figure 2. Automatically generated explanation

EB has some of the most sophisticated fabrication capabilities in the world, however, to increase efficiency sometimes it is best to outsource/farm out less specialized work. Aurora already provided many of the graphical and tabular reports to help the user determine what is best to outsource. Aurora has been modified to provide a convenient interface for visualizing which tasks can be outsourced and providing a one-click option to outsource a task that adjusts the actual model appropriately, see Figure 3.

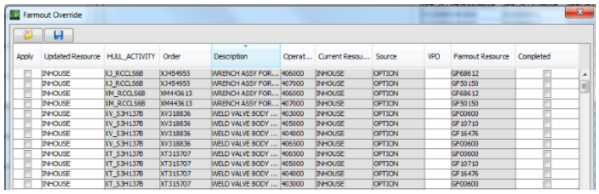

Figure 3. Farmout /Outsource interface

Aurora for EB has been enhanced to provide the ability to handle less than perfect data sources, such as having an override for the status of work-in-progress tasks, so schedulers can easily override data from external sources to match reality when those external sources have not yet been fully updated. For example, the latest data may include information about open tasks that actually have zero (0) duration remaining. This may occur if an operation which has an initial estimate of 10 hours, experiences unforeseen circumstances that cause the operation to actually need more than 10 hours to complete. However, the current external system that data is read from simply calculates the remaining duration from the original duration minus the hours worked. Once the hours exceed the original duration, Aurora will see the remaining duration as zero (0). Therefore, a dialog is provided, see Figure 4, that shows all the open operations and their currently calculated remaining durations. The user has the option to change any of the remaining durations or to mark an operation complete. This information can also be saved out separately and later read back in if desired.

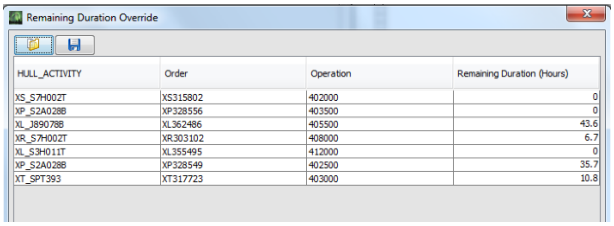

Figure 4. Remaining duration override interface

Overall, EB mostly needed enhancements related to ways to increase the efficiency of the user experience. That is, certain data that is read into Aurora from external systems is not updated in a way that Aurora needs for various reasons. For example, when an operation is outsourced in the external system it means the actually outsource process steps will be commenced. This is not appropriate for situations where long-term scheduling is occurring, and outsourcing is used to meet deadlines that may occur months or years in the future. The ability to easily outsource items to test long-term schedules is useful and necessary, but it is not desired to start the outsourcing process since more changes may occur during the interim and the actual outsourcing specifics may change.

Concurrent and Non-concurrent Constraints

Many domains benefit from the concept non-concurrent constraints, due to the fact that workspace is limited and there are many situations where tasks should not be performed too close to each other at the same time. Figure 5 shows non-concurrent constraint for tasks A, B and Figure 6 shows concurrent constraints for task B, A, & C.

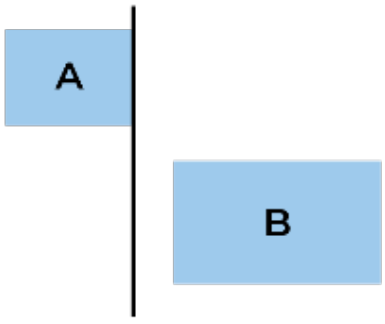

Figure 5. Non-concurrent tasks

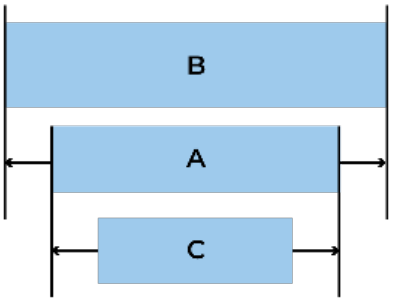

Figure 6. Concurrent tasks

A variation of the non-concurrent constraint is the ability to mark activities as being 'hazardous' to other activities. The result of such a hazardous marking means that Aurora will never schedule the hazardous activities to occur simultaneously with any of the activities it is hazardous to. Graphical enhancements now allow for hazard activities to be denoted in the PERT Chart, with special arrows emanating from the activity causing the hazard and pointing to the activities affected. Figure 7 illustrates hazardous constraints.

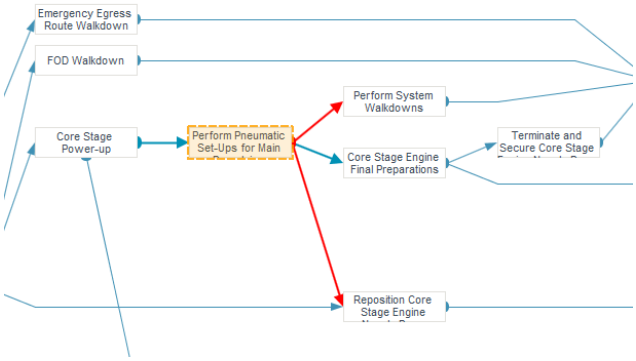

Figure 7. Hazardous constraints shown with red arrows

INTELLIGENT SCHEDULING BENEFITS

The use of Aurora for scheduling has typically meant that 10%+ more tasks can be accomplished with the same resources in the same amount of time (or the same tasks accomplished in 10%+ less time) when compared with other scheduling methods.

One real-world example considers the analysis of a refinery turnaround project. Note that no Microsoft Project results are provided because the MS Project software could not successfully resource-level this project.

The project network consists of over 2,500 activities. A view of the network is shown in Figure 8. Note 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 8.

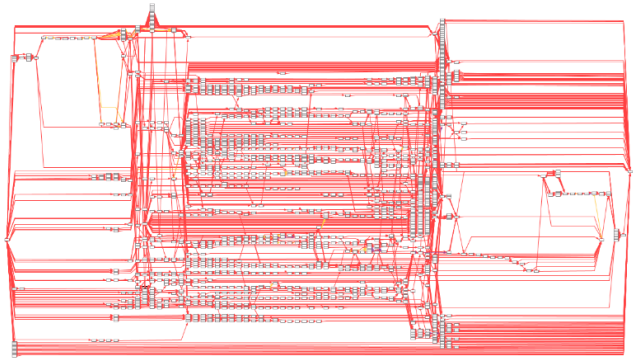

Figure. 8. Turnaround Project Network

The results of the analyses are shown in Figure 9.

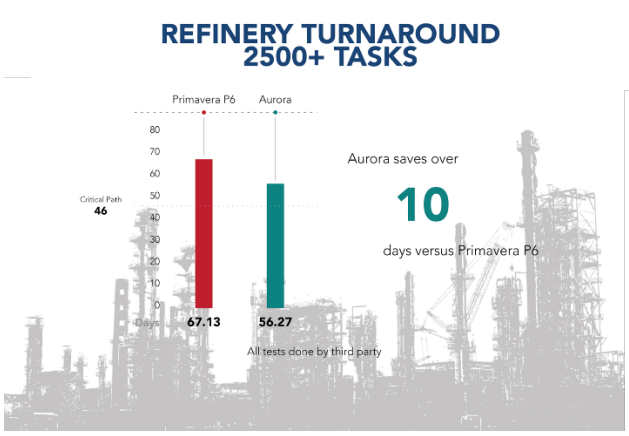

Figure 9. Scheduling Results – Refinery Project

The difference in absolute terms is over 10 days. There are a few ways to compare these results; the simplest is to simply compare overall durations, using Aurora's intelligent scheduling results as the basis: Primavera P6 resource-leveling is over 19% longer than intelligent scheduling. Using the Primavera P6 resource-leveling as the bases: Intelligent scheduling is over 16% shorter than Primavera P6 resource-leveling.

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 %
Aurora 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 Aurora is over 100%.

These results demonstrate the significant benefit of leveraging Aurora's intelligent scheduling. Recall that everything besides the method for scheduling is the same in both cases. Leveraging Aurora 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 Aurora 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, updating the schedule can be done quickly, and the updated schedule will be shorter than if one used resource-leveling only. Therefore, every time a schedule update is performed, the overall benefit of leveraging Aurora's intelligent scheduling increases.

CONCLUSIONS

General Dynamics Electric Boat's submarine production has benefited significantly from intelligently scheduling and modeling to the detail necessary so that the schedule itself reacts correctly to real-world changes during execution, thereby increasing efficiency and reducing overall project duration. Intelligent scheduling includes a strategy that leverages scheduling heuristics learned from many of the world's best human schedulers, including those involved in submarine scheduling, in order to solve complex scheduling challenges in reasonable amounts of time.

The capabilities that have benefited GDEB includes:

- Large multi-project support, able to handle 100,000+ tasks per project.
- Multiple-pass intelligent resource-constrained scheduling, resulting in shorter projects and greater transparency.
- Mixed-mode scheduling, supporting both forward and backward scheduling, available on a task-by-task basis.
- Schedule explanations for each task providing greater understanding and transparency.
- Scheduling and re-scheduling occur wall clock time fast, so many what-ifs / scenarios can be performed rapidly.
- Support for various constraint types, which allow for the correct modeling of GDEB realities.

So now the user has the ability to model their situation to the level of detail required, can find optimal schedules, and finally perform the scheduling in such a short amount of time that various other what-if scenarios can be performed as desired.

REFERENCES

[1] M. L. Pinedo, Scheduling. Cham: Springer International Publishing, 2016.

[2] A. Kalton, "Applying an Intelligent Reconfigurable Scheduling System to Large-Scale Production Scheduling," presented at the International Conference on Automated Planning & Scheduling (ICAPS) 2006, Ambleside, The English Lake District, U.K., 2006.

[3] R. Richards, "Critical Chain: Short-Duration Tasks & Intelligent Scheduling in e.g., Medical, Manufacturing & Maintenance," presented at the 2010 Continuous Process Improvement (CPI) Symposium, Cal State University, Channel Islands, 2010.

[4] Kalton, A., R. Richards. (2008) Advanced Scheduling Technology for Shorter Resource Constrained Project Durations. AACE International's 52nd Annual Meeting & ICEC's 6th World Congress on Cost Engineering, Project Management and Quantity Surveying. Toronto, Ontario, Canada. June 29 – July 2, 2008.

[5] Mohammed, J., Stottler, R., "Rapid Scheduling of Multi-tracking Sensors for a Responsive Satellite Surveillance Network," Proceedings of the Infotech@Aerospace 2010 Conference, Vol. 1, AIAA, Reston, VA, 2010.

[6] Stottler, R., Thompson, R., "Globally Optimized Scheduling for Space Object Tracking," Proceedings of the Infotech@Aerospace 2011 Conference, Vol. 1, AIAA, Reston, VA, 2011.

[7] D. Stottler and K. Mahan, "Automatic, Rapid Replanning of Satellite Operations for Space Situational Awareness (SSA)," presented at the Advanced Maui Optical and Space Surveillance Technologies Conference, 2015.