Proceedings of the 2014 Winter Simulation Conference
A.Tolk, S.Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J.A. Miller, eds.

A SIMULATION BASED HEURISTIC APPROACH TO A RESOURCE INVESTMENT PROBLEM (RIP)

Scott R. Schultz
Jonathan Atzmon

Mercer University
1400 Coleman Ave.
Macon, GA 31207, USA

ABSTRACT

A simulation-based heuristic approach is presented for a resource investment problem (RIP). This version of the RIP considers the trade-off between the number of resources, project makespan and resource utilization. A "win-win" goal is a reduction in project makespan while improving resource utilization. The RIP heuristic is presented as an executive for RCAN, a simulation tool that produces solutions to the multi-mode resource constrained project scheduling problem (RCPSP). The RIP heuristic uses feedback from RCAN to iteratively modify a set of renewable resources. The heuristic is shown to be effective on real-world, large-scale depot maintenance projects. In addition, the simulation tool uses a priority rule approach to schedule project tasks for the RCPSP problem. Therefore the RCPSP priority rule has a major effect on the RIP heuristic. An analysis is presented showing how various priority rules impact the RIP heuristic's ability to reduce the makespan while maintaining or increasing resource utilization.

1 INTRODUCTION

Depot maintenance is the complete inspection and repair of an aircraft, often taking from 3 to 6 months. Efforts are in progress to significantly reduce this cycle time, also known as the project makespan. These efforts include implementation of lean manufacturing practices, performing pre-inspections of aircraft to ensure availability of spare parts, and assigning the proper mix of skilled trade resources to various stages of the depot maintenance process. In this paper, we consider how the availability of skilled trade resources, known in literature as renewable resources, impact project makespan.

Because this work is concerned with reducing the makespan within a depot maintenance setting, we are particularly concerned with the tradeoff between the investment in renewable resources and the reduction in makespan. Ordinarily, the addition of resources to reduce a project's makespan results in a reduction in resource utilization. However, we have observed that the proper mix of resources, hereafter referred to as the resource pool, may not only reduce makespan but can improve resource utilization, a "win-win" proposition.

This planning and scheduling of depot maintenance can be modeled as an activity network where the activities or tasks have precedence relationships. These tasks are executed by a set of renewable resources which are scheduled on a 3 shift operation. In literature, this problem of determining the impact of resource allocation on project scheduling is known as the Resource Investment Problem (RIP), see Mohring (1984). If task durations are dependent on the quantity of assigned resources, then the problem becomes a Multi-mode Resource Investment Problem (MRIP)

978-1-4799-7486-3/14/$31.00 ©2014 IEEE

The RIP and MRIP are classes of project scheduling problems. The typical objective of RIP and MRIP problems is to minimize the cost of procuring resources used in executing a project. Mohring (1984) was first to introduce the RIP problem and included a deadline constraint. He also proved RIP to be NP-Hard. While literature on RIP and MRIP is sparse, authors have looked at various extensions. For example Sabzehparvar et al. (2008) extend the RIP problem with a mathematical model allowing for multiple modes in which activities can be performed, thus MRIP. Hsu and Kim (2005) also consider the MRIP problem and present heuristic rules that consider due date constraints and resource usage. Shadrokh and Kianfar (2005) allow for tardiness in the RIP problem by including a penalty cost in the objective when the project is not completed by its due date. Najafi and Asimi (2009) also allow for tardiness using a discounted cash flow objective. In this paper, a strict objective function is not considered; instead, tradeoffs between size and mix of resource pools are compared to the project's makespan.

The RIP and MRIP problems are particularly challenging combinatorial optimization problems because they contain the Resource Constrained Project Scheduling Problem (RCPSP) which is shown by Blazewicz, Lenstra, and RinnooyKan (1983) to be NP-Hard. NP-Hard is a classification of discrete optimization problems from computational complexity theory. Being NP-Hard suggests that a polynomial-time algorithm for solving the problem is unlikely to be found. Thus alternative procedures that find good, but not necessarily optimal solutions are sought, particularly for large problem instances. For example, Najafi and Niaki (2006), and Shadrokh and Kianfar (2007) use heuristics in the form of genetic algorithms to obtain solutions for RIPs. These papers tend to limit problem sizes to 60 activities in the network and 5 resources. The actual problems encountered in depot maintenance may contain over 1000 activities and 100 resources. To address the large-scale depot maintenance RIP and MRIP problems, Schultz and Fithian (2011) present a simulation based analysis tool, called RCAN. RCAN requires an analyst to provide a depot maintenance project network and a pool of renewable resources. RCAN uses these inputs and a priority rule based heuristic within a simulation setting to produce the project's makespan and performance measures such as resource utilization. The analyst can evaluate the results of an RCAN simulation run, adjust the resource pool, and rerun the simulation to determine the impact of the new resource pool.

This paper presents two enhancements to RCAN: 1) an executive containing a heuristic to automate the RIP analysis; and 2) a brief study of how the priority rule impacts the solution time and quality for large-scale real-world RIPs.

2 MODELING APPROACH

2.1 Model Logic

Schultz and Fithian (2011) present a simulation model for modeling a generic Resource Constrained Project Scheduling Problem. The basic premise of the simulation is that the project tasks are entities which are processed by seizing resources from the resource pool. When a task has completed processing, all its successor tasks are generated. A task can only begin processing if all its predecessors have been processed; and, the minimum required skilled resources to begin the task are available. Note a task can require more than one type of skilled resource.

A task has parameters indicating the minimum and maximum number of personnel, by skill code, which can be assigned to the task. The simulation will always attempt to assign the maximum personnel, but will begin processing a task if the minimum is available. Once the task begins processing, it will periodically check to see if additional resources have become available, and will continue to try and seize additional resources until the maximum is met.

The task duration time is a linear function of its processing hours and the number of assigned personnel. For example, consider a 4 hour task which requires a minimum of 2 Electricians, but allows a maximum of 4 Electricians to be assigned. If only 2 Electricians are available, the task will take 4/2=2 hours to perform. However, if 4 Electricians are available at the start, the task will take 4/4=1 hour to perform. But, if 2 Electricians are available at the start and then 2 more become available an hour into the job, then the task will take 1.5 hours to perform.

The version of RCAN presented by Schultz and Fithian (2011) uses a simple priority rule. This priority rule is to randomly choose tasks from a list of available tasks. For a task to be "available", the following two conditions must be met: all its predecessor tasks must be complete; and its minimum required resources must be available.

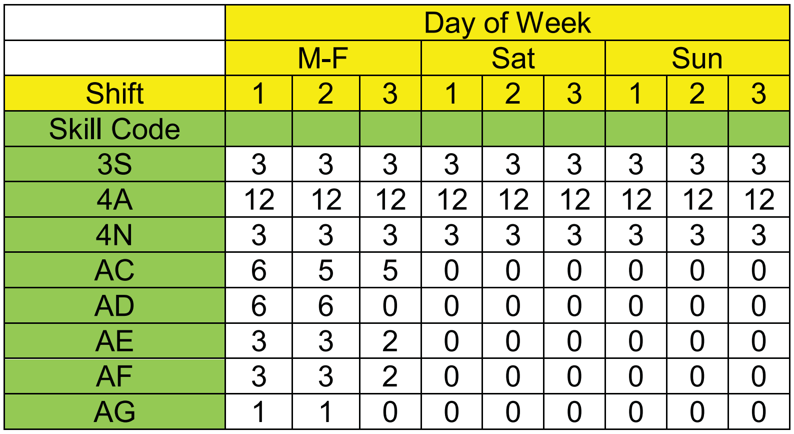
The simulation will run until all tasks have completed.

2.2 Model Inputs

Key inputs for modeling a resource constrained project scheduling problem using RCAN include: a listing of project tasks; the project network completely defined including the ability to model start-finish, finish-start, finish-finish and start-start relationships; the minimum number of skilled tradesmen needed to begin the task; the maximum personnel that can simultaneously perform the task; a triangular distribution for task processing hours; a percent task completion value to allow mid-project modeling; and a resource pool defined by shift and day of week.

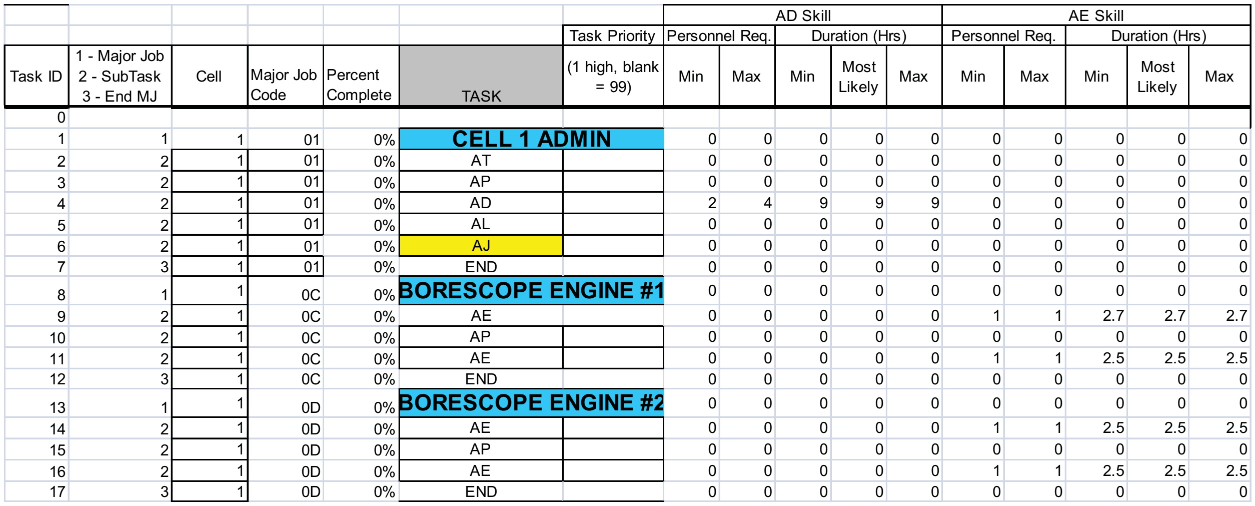
The resource pool is the amount of skilled trade resources available during each operational shift. An example of a resource pool is provided in Table 1. Note the resource pool remains constant Monday through Friday, but can be defined differently for Saturday and Sunday.

Table 1: Example resource pool



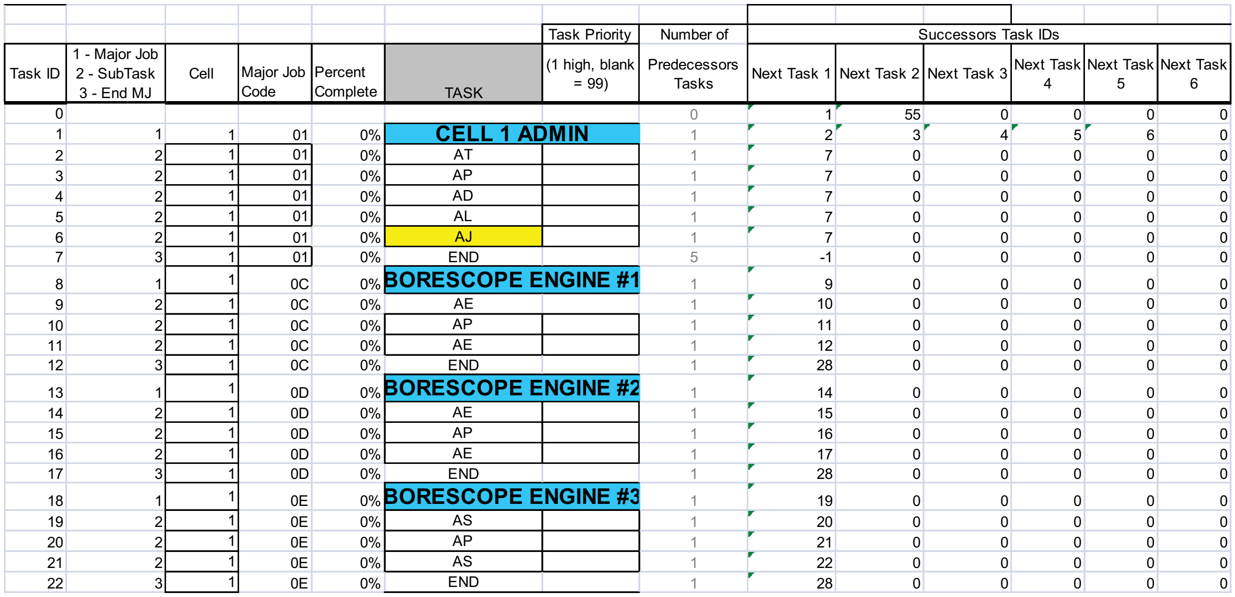
	Day of Week								
	M-F			Sat			Sun		
Shift	1	2	3	1	2	3	1	2	3
Skill Code									
3S	3	3	3	3	3	3	3	3	3
4A	12	12	12	12	12	12	12	12	12
4N	3	3	3	3	3	3	3	3	3
AC	6	5	5	0	0	0	0	0	0
AD	6	6	0	0	0	0	0	0	0
AE	3	3	2	0	0	0	0	0	0
AF	3	3	2	0	0	0	0	0	0
AG	1	1	0	0	0	0	0	0	0

Figure 1 shows a partial listing of project tasks with resource requirements and processing hours. The tasks are configured using a "major job" concept. A major job consists of a start task which has no processing hours or resource requirements. The major job contains a series of subtasks which contain the processing time and resource requirements. The last task within a major job is the "End" task which again has no process time or resource requirements, but is used to represent the completion of the major job. This approach of using a major job, subtask and end task construct allows for modeling start-start, start-finish, finish-start, finish-finish relationships between major jobs. Figure 2 shows the network relationship for these same tasks. The network relationships are fully defined by providing a list of successor tasks.



							AD Skill					AE Skill				
						Task Priority	Personnel Req.		Duration (Hrs)			Personnel Req.		Duration (Hrs)		
Task ID	1 - Major Job 2 - SubTask 3 - End MJ	Cell	Major Job Code	Percent Complete	TASK	(1 high, blank = 99)	Min	Max	Min	Most Likely	Max	Min	Max	Min	Most Likely	Max
0																
1	1	1	01	0%	CELL 1 ADMIN		0	0	0	0	0	0	0	0	0	0
2	2	1	01	0%	AT		0	0	0	0	0	0	0	0	0	0
3	2	1	01	0%	AP		0	0	0	0	0	0	0	0	0	0
4	2	1	01	0%	AD		2	4	9	9	9	0	0	0	0	0
5	2	1	01	0%	AL		0	0	0	0	0	0	0	0	0	0
6	2	1	01	0%	AJ		0	0	0	0	0	0	0	0	0	0
7	3	1	01	0%	END		0	0	0	0	0	0	0	0	0	0
8	1	1	0C	0%	BORESCOPE ENGINE #1		0	0	0	0	0	0	0	0	0	0
9	2	1	0C	0%	AE		0	0	0	0	0	1	1	2.7	2.7	2.7
10	2	1	0C	0%	AP		0	0	0	0	0	0	0	0	0	0
11	2	1	0C	0%	AE		0	0	0	0	0	1	1	2.5	2.5	2.5
12	3	1	0C	0%	END		0	0	0	0	0	0	0	0	0	0
13	1	1	0D	0%	BORESCOPE ENGINE #2		0	0	0	0	0	0	0	0	0	0
14	2	1	0D	0%	AE		0	0	0	0	0	1	1	2.5	2.5	2.5
15	2	1	0D	0%	AP		0	0	0	0	0	0	0	0	0	0
16	2	1	0D	0%	AE		0	0	0	0	0	1	1	2.5	2.5	2.5
17	3	1	0D	0%	END		0	0	0	0	0	0	0	0	0	0

Figure 1: Project task listing



						Task Priority	Number of	Successors Task IDs					
Task ID	1 - Major Job 2 - SubTask 3 - End MJ	Cell	Major Job Code	Percent Complete	TASK	(1 high, blank = 99)	Predecessors Tasks	Next Task 1	Next Task 2	Next Task 3	Next Task 4	Next Task 5	Next Task 6
0							0	1	55	0	0	0	0
1	1	1	01	0%	CELL 1 ADMIN		1	2	3	4	5	6	0
2	2	1	01	0%	AT		1	7	0	0	0	0	0
3	2	1	01	0%	AP		1	7	0	0	0	0	0
4	2	1	01	0%	AD		1	7	0	0	0	0	0
5	2	1	01	0%	AL		1	7	0	0	0	0	0
6	2	1	01	0%	AJ		1	7	0	0	0	0	0
7	3	1	01	0%	END		5	-1	0	0	0	0	0
8	1	1	0C	0%	BORESCOPE ENGINE #1		1	9	0	0	0	0	0
9	2	1	0C	0%	AE		1	10	0	0	0	0	0
10	2	1	0C	0%	AP		1	11	0	0	0	0	0
11	2	1	0C	0%	AE		1	12	0	0	0	0	0
12	3	1	0C	0%	END		1	28	0	0	0	0	0
13	1	1	0D	0%	BORESCOPE ENGINE #2		1	14	0	0	0	0	0
14	2	1	0D	0%	AE		1	15	0	0	0	0	0
15	2	1	0D	0%	AP		1	16	0	0	0	0	0
16	2	1	0D	0%	AE		1	17	0	0	0	0	0
17	3	1	0D	0%	END		1	28	0	0	0	0	0
18	1	1	0E	0%	BORESCOPE ENGINE #3		1	19	0	0	0	0	0
19	2	1	0E	0%	AS		1	20	0	0	0	0	0
20	2	1	0E	0%	AP		1	21	0	0	0	0	0
21	2	1	0E	0%	AS		1	22	0	0	0	0	0
22	3	1	0E	0%	END		1	28	0	0	0	0	0

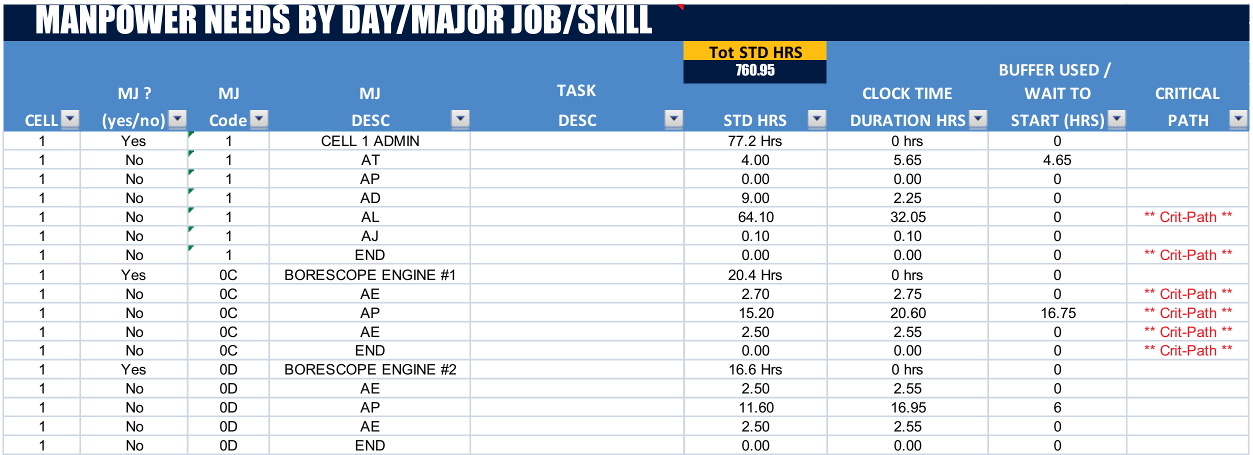
Figure 2: Project network relationships

2.3 Model Outputs

Throughout the simulation run, task specific information, such as start time and completion time, are captured and reported to both a Microsoft Project output file and Excel output file. In addition, shift summary information is also provided to the Excel output file.

Recall that minimizing the makespan is the primary objective for modeling and analyzing the project network. To reduce the makespan, improvements must be made to the critical path. The critical path is identified in both the Microsoft Project output and in the Excel output file. The Excel file, see Figure 3, not only shows which tasks are on the critical path, but also displays how long a delay occurred from when the task could begin until it actually started. Note that a task can begin whenever all its predecessor tasks have completed, but will not begin until the minimum resource requirement is available. For example, in Figure 3 all subtasks under the ‘Borescope Engine #1’ major job are on the critical path.

However, the task requiring AP mechanics was delayed from starting by 16.75 hours because those mechanics were busy performing other tasks. Thus adding additional AP mechanics to the resource pool should improve the makespan.

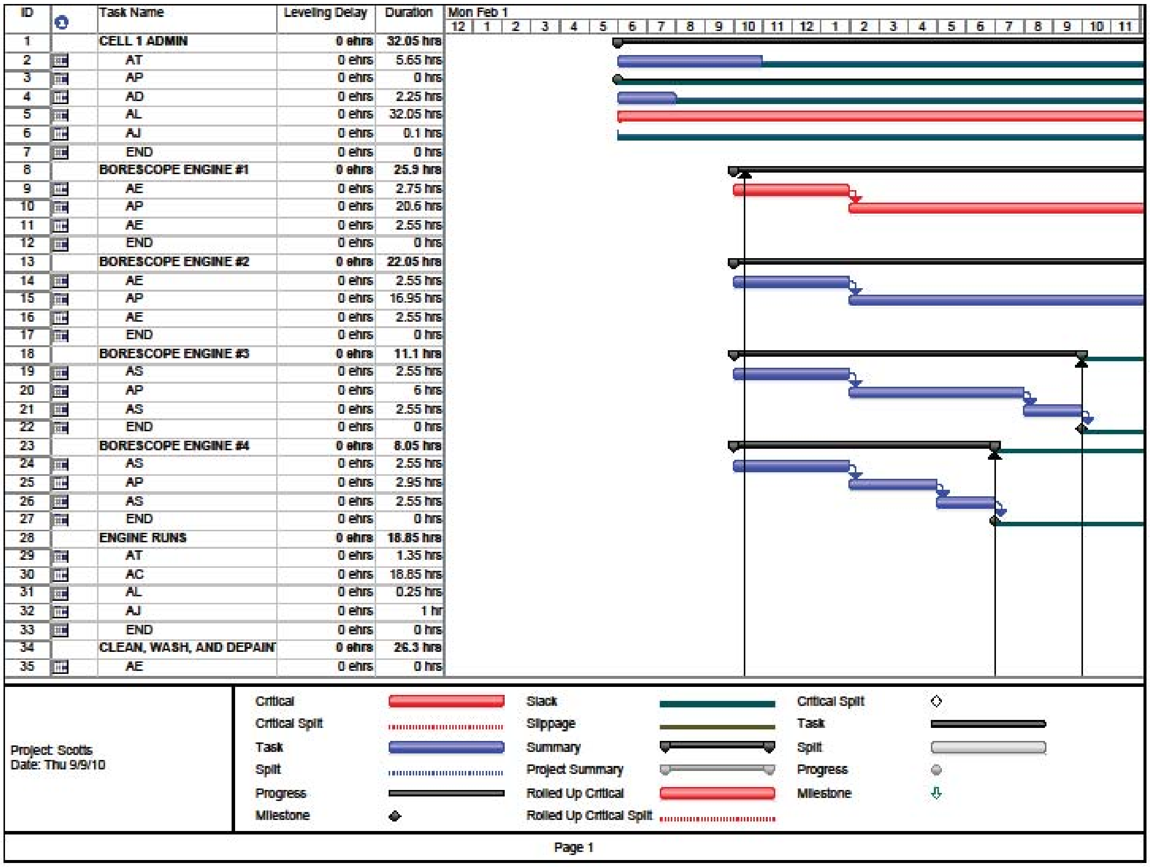


MANPOWER NEEDS BY DAY/MAJOR JOB/SKILL

					Tot STD HRS 760.95		BUFFER USED /	
	MJ ?	MJ	MJ	TASK		CLOCK TIME	WAIT TO	CRITICAL
CELL	(yes/no)	Code	DESC	DESC	STD HRS	DURATION HRS	START (HRS)	PATH
1	Yes	1	CELL 1 ADMIN		77.2 Hrs	0 hrs	0	
1	No	1	AT		4.00	5.65	4.65	
1	No	1	AP		0.00	0.00	0	
1	No	1	AD		9.00	2.25	0	
1	No	1	AL		64.10	32.05	0	** Crit-Path **
1	No	1	AJ		0.10	0.10	0	
1	No	1	END		0.00	0.00	0	** Crit-Path **
1	Yes	0C	BORESCOPE ENGINE #1		20.4 Hrs	0 hrs	0	
1	No	0C	AE		2.70	2.75	0	** Crit-Path **
1	No	0C	AP		15.20	20.60	16.75	** Crit-Path **
1	No	0C	AE		2.50	2.55	0	** Crit-Path **
1	No	0C	END		0.00	0.00	0	** Crit-Path **
1	Yes	0D	BORESCOPE ENGINE #2		16.6 Hrs	0 hrs	0	
1	No	0D	AE		2.50	2.55	0	
1	No	0D	AP		11.60	16.95	6	
1	No	0D	AE		2.50	2.55	0	
1	No	0D	END		0.00	0.00	0	

Figure 3: Partial Excel report showing critical path and delays due to limited resources

Figure 4 is the Detailed Gantt Chart view of the Microsoft Project file for the same project as depicted in Figure 3. The Gantt chart view helps identify which tasks may be causing the delay for starting a critical task. For example, in Figure 3, AP tasks under both BoreScope Engine #1 and Borescope Engine #2 are delayed by 16.75 and 6 hours respectively. Looking at Figure 4, the AP tasks for BoreScope Engine #3 and Engine #4 are being performed in parallel with Engine #1 and Engine #2. Engine #3 and Engine #4 complete first, followed by Engine#2 and then Engine #1. Thus the lack of AP resources are causing delays at Engine #1 and #2.



ID	Task Name	Leveling Delay	Duration
1	**CELL 1 ADMIN**	**0 ehrs**	**32.05 hrs**
2	AT	0 ehrs	5.65 hrs
3	AP	0 ehrs	0 hrs
4	AD	0 ehrs	2.25 hrs
5	AL	0 ehrs	32.05 hrs
6	AJ	0 ehrs	0.1 hrs
7	END	0 ehrs	0 hrs
8	**BORESCOPE ENGINE #1**	**0 ehrs**	**25.9 hrs**
9	AE	0 ehrs	2.75 hrs
10	AP	0 ehrs	20.6 hrs
11	AE	0 ehrs	2.55 hrs
12	END	0 ehrs	0 hrs
13	**BORESCOPE ENGINE #2**	**0 ehrs**	**22.05 hrs**
14	AE	0 ehrs	2.55 hrs
15	AP	0 ehrs	16.95 hrs
16	AE	0 ehrs	2.55 hrs
17	END	0 ehrs	0 hrs
18	**BORESCOPE ENGINE #3**	**0 ehrs**	**11.1 hrs**
19	AS	0 ehrs	2.55 hrs
20	AP	0 ehrs	6 hrs
21	AS	0 ehrs	2.55 hrs
22	END	0 ehrs	0 hrs
23	**BORESCOPE ENGINE #4**	**0 ehrs**	**8.05 hrs**
24	AS	0 ehrs	2.55 hrs
25	AP	0 ehrs	2.95 hrs
26	AS	0 ehrs	2.55 hrs
27	END	0 ehrs	0 hrs
28	**ENGINE RUNS**	**0 ehrs**	**18.85 hrs**
29	AT	0 ehrs	1.35 hrs
30	AC	0 ehrs	18.85 hrs
31	AL	0 ehrs	0.25 hrs
32	AJ	0 ehrs	1 hr
33	END	0 ehrs	0 hrs
34	**CLEAN, WASH, AND DEPAIN**	**0 ehrs**	**26.3 hrs**
35	AE	0 ehrs	0 hrs

Project: Scotts
Date: Thu 9/9/10

Page 1

Figure 4: Microsoft Project detailed Gantt chart view

3 RIP HEURISTIC

3.1 Background

The purpose of the RCAN simulation analysis tool is to analyze the impact of resource allocation decisions on RCPSPs. This tool helps an analyst understand the tradeoff between the size and makeup of the resource pool and the makespan of the project. Ordinarily one might reason that as resources are added to a resource pool to reduce the project makespan, that resource utilization would decrease. However, we have observed that for many large-scale real-world depot maintenance projects that adding resources can in fact reduce the makespan while increasing resource utilization, a "win-win".

3.2 Heuristic Development and Description

During the simulation run of a depot maintenance project, useful information is gathered that can be used in a heuristic to determine how to adjust the resource pool. Of particular value is the critical path and the "wait to start" time as depicted in the last two columns of Figure 1. To obtain a task's "wait to start" time, a timer is begun when all the task's precedence conditions have been met. The timer counts time until the minimum resources are available for starting the task. In other words, the "wait to start" time is the time from which a task could be scheduled, but is not scheduled due to lack of resources, until the task is scheduled.

Therefore, adding resources to the resource pool can reduce the "wait to start" time for tasks. With the goal of reducing makespan, priority should be given to reducing the "wait to start" time for tasks on the critical path. This leads to a simple heuristic.

1 – Determine an initial resource pool. See section 3.3 for a discussion on obtaining this initial resource pool.
2 – Use the RCAN simulation tool to obtain the critical path and "wait to start" times.
3 – Find the task on the critical path which has the largest "wait to start" time and identify the skill code which performs that task.
4 – Add one resource of the skill code identified in step 3 to each active shift of the resource pool.
5 – Repeat steps 2 through 4 until the "wait to start" times for all tasks on the critical path is 0.0.

3.3 Initial Resource Pool

Because the RIP heuristic adds resources each iteration, the initial resource pool is the minimum feasible resource pool for the given depot maintenance project. As described earlier, both a minimum and maximum personnel requirement is provided for each task, by skill code. The minimum represents the number of resources required to begin a task, while the maximum represents the most resources that can simultaneously perform the task. Thus if the minimum resource requirement for task *i* and for skill code *j*, is defined as $minR_{ij}$, then the initial resource pool is defined in (1) as:

$$\max_i\{minR_{ij}\}\ \forall\ j \tag{1}$$

For example, using Figure 1, the minimum resource pool would contain two AD resources and one AE resource.

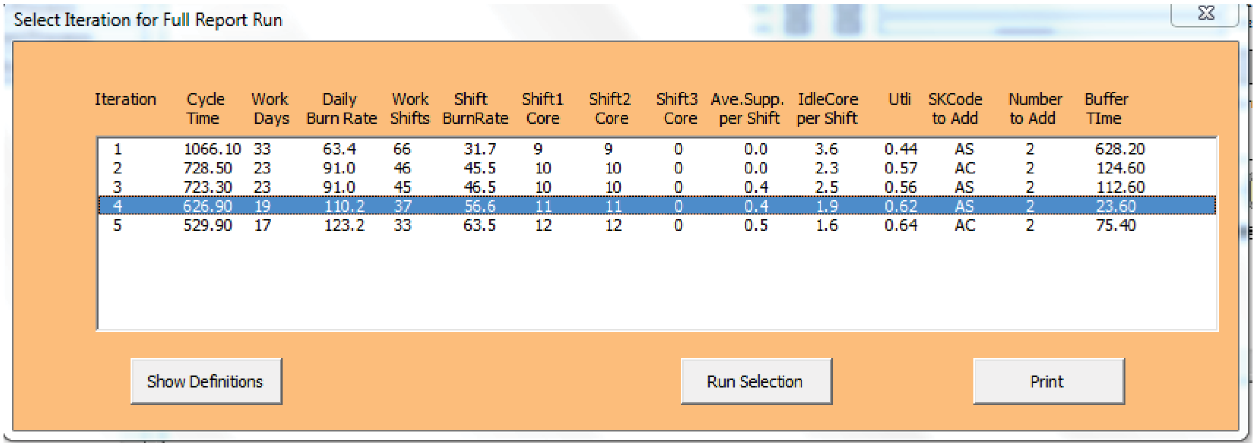
3.4 Problem Instance

As an example of the RIP heuristic in action, the third stage of depot maintenance on a military aircraft is used. This stage contains 25 major jobs and over 100 tasks. The tasks are performed by three skill codes, AC, AS, and AV. Work is scheduled over 2 shifts Monday-Friday with no weekend work. The total project contains 2040 man-hours of work. This depot maintenance project was simulated using the RCAN simulation tool controlled by the RIP heuristic described in section 3.2.

One other note is that while RCAN uses a triangular distribution to represent project task times, discrete values were provided for this actual depot maintenance project. Therefore the minimum, most likely, and maximum durations for each task's triangular distribution were set to the same value as can be seen in figure 1, creating a deterministic simulation. Thus, multiple runs of the simulation to determine means and confidence intervals is unnecessary.

3.5 Results

The RIP heuristic is designed to present various scenarios for an analyst to select from rather than a single "optimal" scenario. Figure 5 is the report showing the results of the RIP heuristic applied to the example problem. The first iteration represents the results using the initial resource pool which contained 9 resources on shift 1 and shift 2. The makespan of the depot maintenance project using this initial resource pool is 1066.1 hours or 33 work days. The sum of the "wait to start" times over all tasks for this first iteration is 628.2 hours. The average utilization of the 18 resources is 44%. The skill code with the most "wait to start" time is the AS skill code.



Select Iteration for Full Report Run

Iteration	Cycle Time	Work Days	Daily Burn Rate	Work Shifts	Shift BurnRate	Shift1 Core	Shift2 Core	Shift3 Core	Ave.Supp. per Shift	IdleCore per Shift	Utli	SKCode to Add	Number to Add	Buffer TIme
1	1066.10	33	63.4	66	31.7	9	9	0	0.0	3.6	0.44	AS	2	628.20
2	728.50	23	91.0	46	45.5	10	10	0	0.0	2.3	0.57	AC	2	124.60
3	723.30	23	91.0	45	46.5	10	10	0	0.4	2.5	0.56	AS	2	112.60
4	626.90	19	110.2	37	56.6	11	11	0	0.4	1.9	0.62	AS	2	23.60
5	529.90	17	123.2	33	63.5	12	12	0	0.5	1.6	0.64	AC	2	75.40

Figure 5: Simulation results for each iteration of the RIP heuristic

By the fifth iteration of the RIP heuristic, the makespan has significantly decreased to 529.9 hours while the resource utilization has increased to 64%. The number of resources has increased by 10 personnel, 5 on each shift.

3.6 Discussion of Results

Although not as dramatic as shown in Figure 5, this response pattern is typical of other depot maintenance projects. Initially, the makespan (or cycle time) decreases significantly in early iterations, along with an increase in resource utilization. At some point however, the reduction in makespan between iterations slows dramatically, and resource utilization begins to decline.

4 PRIORITY RULE ASSESSMENT

4.1 Background on Priority Rules

The RCAN simulation tool uses a priority rule approach to dispatch project tasks from a list of available tasks as described in section 2.1. A second objective of this paper is to evaluate how various priority rules impact the performance of the RIP heuristic on large-scale, real-world projects. The following is a brief description of the priority rules under investigation as described in Alvarez-Valdes and Tamarit (1989).

- SPT – Shortest Processing Time. This rule dispatches the task from the "available list" which has the least number of hours required to process the task, $\min\{d_j\}$.
- LFT – Late Finish Time. This rules dispatches the task from the "available list", $\min\{LST_j + d_j\}$. LST refers to the late start time. The late start time is the latest time a task can start found during the backward pass using the well-known Critical Path Method.
- Min Slack – Minimum Slack Time. This rule dispatches the most critical task from the "available list", $\min\{ LST_j - EST_j\}$. EST refers to the early start time. The early start time is the earliest time a task can start found on the forward pass using the well-known Critical Path Method.
- Random – This rule selects a task at random from the "available list". Including this rule is helpful in determining if a rule performs worse than a completely random selection.
- Max Resources – Maximum Required Resources. One condition for a task to be "available" for dispatch is that its minimum required resources must be available. Thus is might seem prudent to give priority to processing those tasks which require a large number of minimum resources. The "Max Resources" rule selects the task from the "available list" which requires the largest number of resources.

4.2 Problem Instances

The large-scale, real-world projects being investigated are depot maintenance projects. Four different project networks are investigated. Primary parameters of interest are the number of project tasks, the number of different skill codes, and the initial size of the resource pool. Table 2 contains the parameter values for each of the four project networks.

Table 2: Problem Instances

Project Name	Number of Tasks	Variety of Skill Codes	Number of Renewable Resources per Shift
Project 2012_12_17	904	20	64
Project 2013_9_5	498	10	45
Project 85000037 CT	490	27	85
Project Cell 1	140	11	35

4.3 Analytical Results

Figures 6 to 9 compare the performance of the five priority rules on the four project networks. Each figure displays the resulting makespan achieved while performing the RIP heuristic for each of the five priority rules. The resource pool is identical on the first iteration. On subsequent iterations, the resource pool is increased by a single resource. Iterations continue until all tasks on the critical path have no “wait to start” time. Thus, the number of iterations can vary over the five priority rules.

4.4 Discussion of Results

The objective of the RIP heuristic is to reduce the makespan and to achieve the reduction in the fewest number of iterations since subsequent iterations use larger resource pools. Thus priority rules, represented by the lines on the figures which are closest to both axis’, are desirable. For example, in Figure 6, the Min Slack and LFT rule produce a better (lower) makespan than the other three rules for iteration 1. In general, Min Slack and LFT continue to produce better makespan than the other rules for all subsequent iterations. Each subsequent iteration represents a larger resource pool by a single resource. Thus a priority rule represented on figures 6-9 that produces the lowest makespan in the earliest iterations is desirable. From figures 6-9, both Min Slack and LFT in general produce results that outperform the other priority rules. This finding is consistent with Davis and Patterson (1975) who concluded that Min Slack performed consistently better than other priority rules when tested on fifty seven computer generated networks.

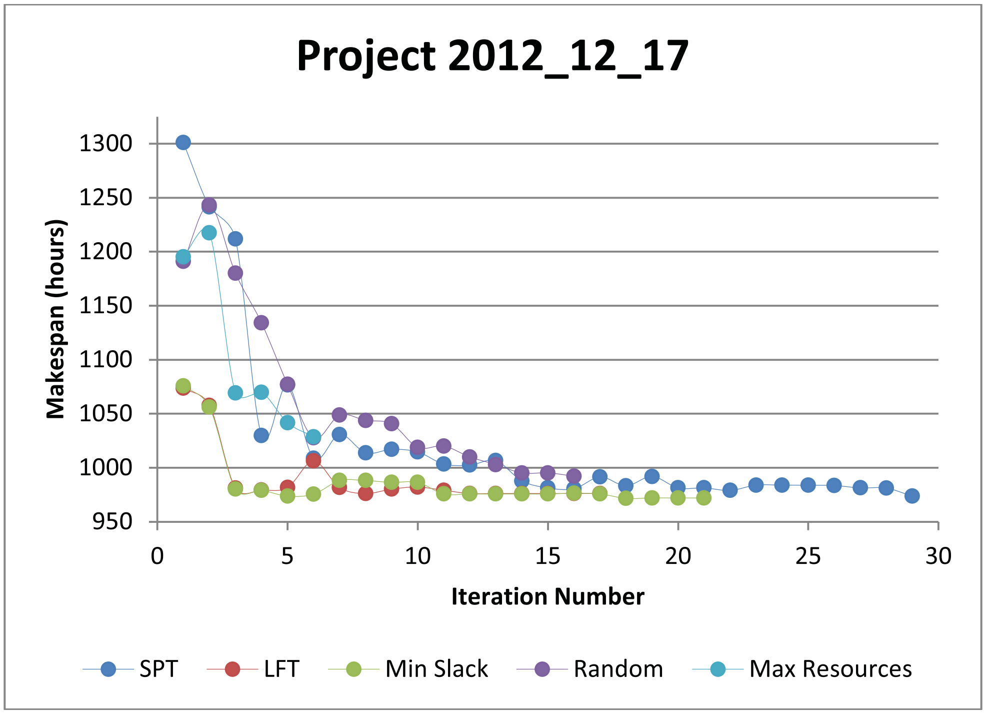


Figure 6: Makespan comparison of RIP Heuristic for Project 2012_12_17

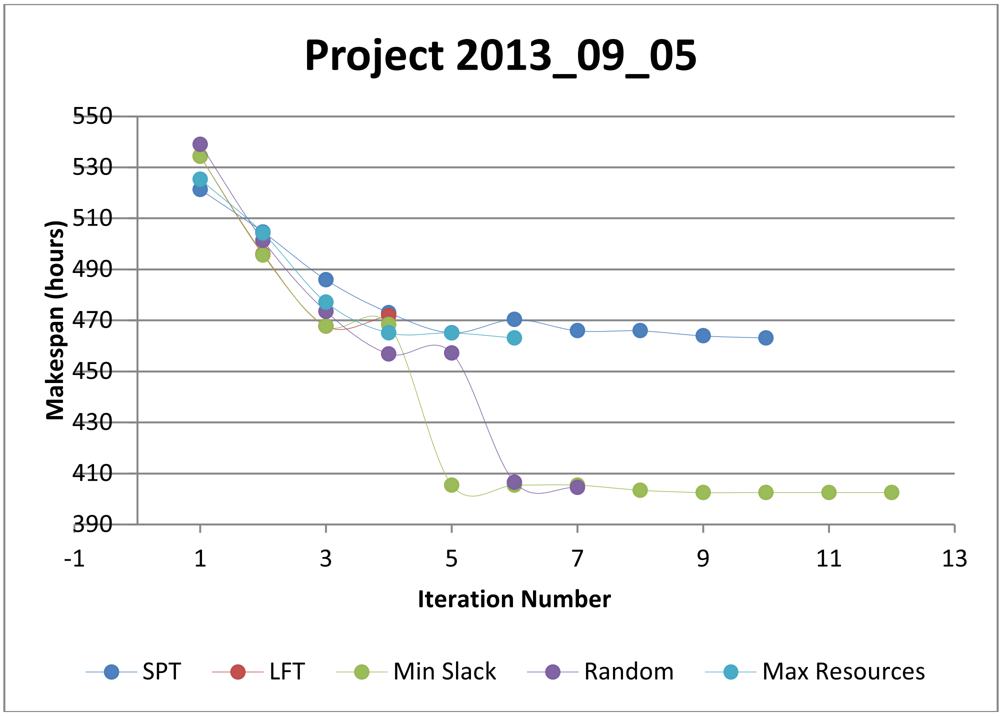


Figure 7: Makespan comparison of RIP Heuristic for Project 2013_09_05

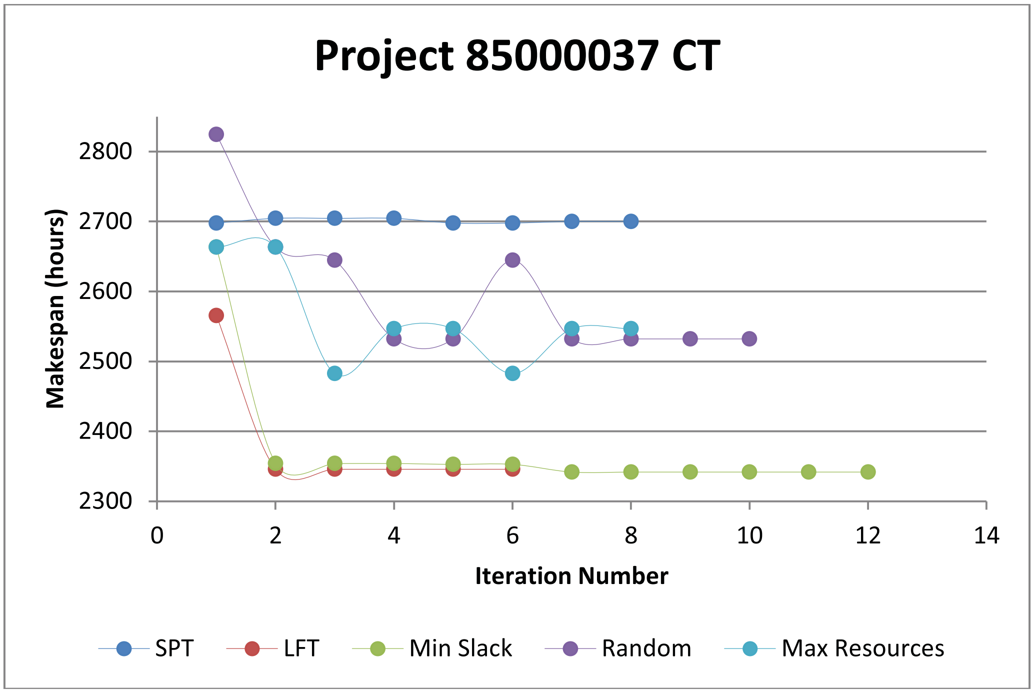


Figure 8 : Makespan comparison of RIP Heuristic for Project 85000037 CT

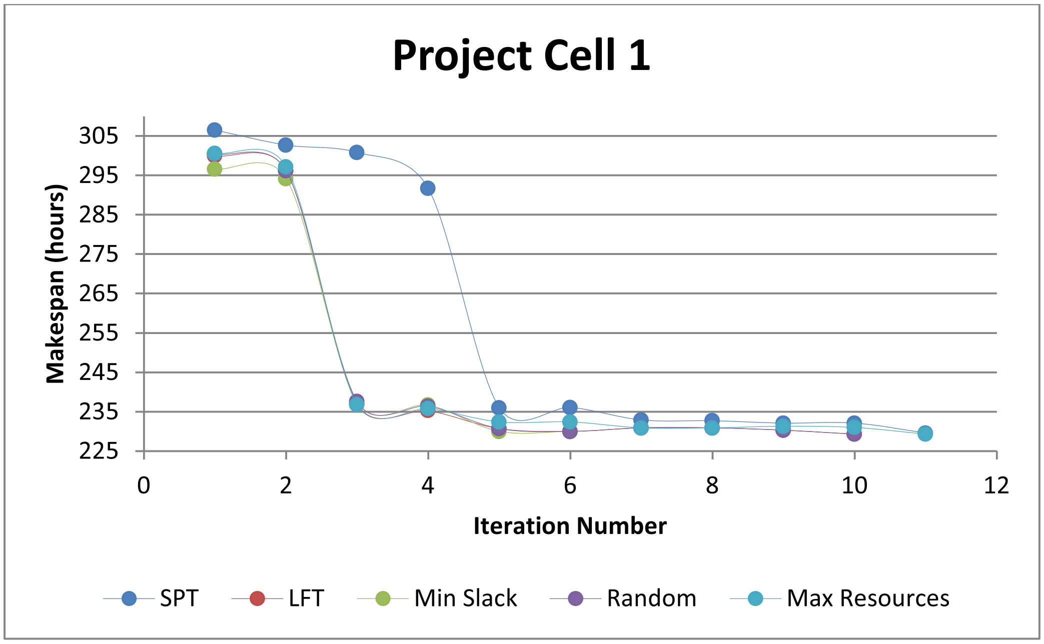


Figure 9 : Makespan comparison of RIP Heuristic for Project Cell 1

5 SUMMARY

Two enhancements have been presented to an RCPSP simulation tool called RCAN. The first enhancement is an RIP heuristic executive as part of the RCAN tool. This RIP heuristic is shown to be effective in selecting a resource pool which reduces project makespan while increasing resource utilization, a "win-win" proposition. The second enhancement is the analysis and selection of the priority rule which dispatches tasks within RCAN's underlying RCPSP solver. The Min Slack and LFT priority rules are shown to be the most effective for RCAN's RIP heuristic.

REFERENCES

Alvarez-Valdes, R. and J.M. Tamarit. 1989. "Heuristic Algorithms for Resource-Constrained Project Scheduling: A review and an Empirical Analysis", *Advances in Project Scheduling,* edited by R. Slowinski and J. Weglarz, 113-134. Elsevier, Amsterdam, the Netherlands.

Blazewicz J., J. Lenstra, A. RinnooyKan. 1983. "Scheduling subject to resource constraints: classification and complexity". *Discrete Applied Mathematics* 5: 11-24.

Davis, E. and J. Patterson. 1975. "A comparison of Heuristic and Optimum Solutions in Resource-Constrained Project Scheduling", *Management Science*, 21: 944-955.

Hsu, C.C. and D.S. Kim. 2005. "A new heuristic for the multi-mode resource investment problem," *The Journal of the Operational Research Society*, 56: 406-413.

Mohring, R.H.. 1984. "Minimizing costs of resource requirements in project networks subject to a fixed completion time," *Operations Research*, 32: 89-120.

Najafi, A.A. and F. Azimi. 2009. "A priority rule-based heuristic for resource investment project scheduling problem with discounted cash flows and tardiness penalties," *Mathematical Problem in Engineering*.

Najafi, A.A., and S.T.A. Niaki. 2006. "A genetic algorithm for resource investment problem with discounted cash flows," *Applied Mathematics and Computation*, 183: 1057-1070.

Savzehparvar, M., S,M. Seyed-Hosseini, and S. Nouri. 2008. "A Mathematical model for the multi-mode resource investment problem," *Journal of Industrial Engineering International*, 4: 25-32.

Schultz, S.R., P. Fithian. 2011. "A practical approach for modeling large scale resource investment problems," In *Proceedings of the 2011 Industrial Engineering Research Conference,* edited by T. Doolen and E. Van Aken. Reno, Nevada.

Shadrokh, S. and F. Kianfar. 2005. "A genetic algorithm for resource investment problems, enhanced by the revised Akpan method", *Scientia Iranica*, 12: 90-98.

Shadrokh, S. and F. Kianfar. 2007. "A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty", *European Journal of Operational Research*, 181: 86-101.

AUTHOR BIOGRAPHIES

SCOTT SCHULTZ is an Associate Professor of Industrial Engineering and an Associate Dean at Mercer University in Macon, Georgia USA. He holds a M.S. in Integrated Manufacturing Systems Engineering and Ph.D. in Industrial Engineering from North Carolina state University in Raleigh, North Carolina, USA. His email address is schultz_sr@mercer.edu.

JONATHAN ATZMON is pursuing a B.S.E with a specialization in Industrial Engineering and M.S.E in Engineering Management at Mercer University in Macon, Georgia USA. His email address is jonathan.atzmon@live.mercer.edu.