

Fall 12-13-2013

Coordinating Job Release Dates with Workdays: a Job Shop Application to Utilities Field Service Scheduling

Ryan Lawrence Pelkey
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds



Part of the [Engineering Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Pelkey, Ryan Lawrence, "Coordinating Job Release Dates with Workdays: a Job Shop Application to Utilities Field Service Scheduling" (2013). *Dissertations and Theses*. Paper 1510.
<https://doi.org/10.15760/etd.1510>

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

Coordinating Job Release Dates with Workdays: A Job Shop Application to Utilities Field

Service Scheduling

by

Ryan Lawrence Pelkey

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science
in
Systems Science

Thesis Committee:
Wayne Wakeland, Chair
James Strathman
Thaddeus Shannon

Portland State University
2013

Abstract

A local utility company processes a variety of jobs each day including meter reading, service shut-offs, emergency response, and customer service work. For the Company, a specific workflow begins with automated meter-reading (AMR) and ends with collections/service shut-offs (CSOs) for accounts with excessively late payments (AMR-CSO workflow). There are considerable and systemic sources of variability in both the workload and resource demands of the AMR-CSO workflow including order arrival, order release schedules, order batch-sizing and maintenance scheduling.

This project draws on theory from the job-shop problem to explore possible means to mitigate this variability. We hypothesized that controlling various forms of input variability would lead to reduced downstream workload variability. Using discrete event simulation we tested a variety of measures to reduce input variability in the workflow. Consistent with other literature we find that various workload control tactics have limited impact on output measures and system performance.

However, we found that system is much more sensitive to resource capacity variability. One input control tactic we call Targeted Release allowed us to reduce Company capacity variability which suggested significantly improved outcomes. These initial results are promising for both the Company and for future investigation of tactics to mitigate resource capacity variability.

Acknowledgements

I would like to thank my committee for bending over backwards to accommodate a very tight timeline and for helping me turn this into a worthwhile document. In particular I want to thank Wayne Wakeland, who has been instrumental in guiding my education and career. Thanks goes to many at the Company but in particular to Lisa Muirden and Chuck Muehleck for providing data and discussion about Company processes and policies. This project would not have been possible without the support and love of my parents Gary and Chris Pelkey. My brothers and best friends Matthew, Andrew and Jeremy always knew the proper proportion of distraction, laughter and motivation I needed. Mostly though, they were just hilarious. Thanks to Dave Lenar for his valuable insights on the issues I explored and the frequently bad way I wrote about them. There are too many friends and classmates to name but in particular thanks to Natalie Perrin, Josh Linden-Levy, Patricia Fox and Ben Pearson for listening to all my silly ideas. Thanks to Maura Hovious for the pie and haircuts and kitty videos. Lastly thanks to Ben Harrington for constantly reminding me how much fun life should be. This would have taken even longer without him.

Table of Contents

Abstract.....	i
Acknowledgements.....	ii
List of Tables.....	v
List of Figures.....	vi
1 INTRODUCTION.....	1
1.1 Background.....	1
1.2 Problem Description.....	2
1.2.1 Workflow.....	2
1.2.2 AMR – Batch Sizes.....	3
1.2.3 AMR – Order Release.....	4
1.2.4 Scheduled Wait-times.....	6
1.2.5 Job Dispatching.....	9
1.2.6 Maintenance.....	10
1.2.7 Project Goals.....	11
2 LITERATURE REVIEW.....	13
2.1 Operation Scheduling.....	13
2.2 Cyclic Scheduling.....	14
2.3 Workload Control (WLC).....	15
3 Methods.....	20
4 SIMULATION.....	27
4.1 Design.....	27
4.1.1 Entity Creation.....	27
4.1.2 Signaling Track.....	31
4.1.3 Meter Processing.....	31
4.1.4 Billing.....	33
4.1.5 Wait-Times.....	35

4.1.6 Credit/Shut-off Processing.....	35
4.2 Validation.....	39
5 EXPERIMENTAL RESULTS.....	45
5.1 Order Batch Size.....	45
5.2 Order Review/Release.....	47
5.3 Targeted Release through Wait-times.....	51
6 DISCUSSION.....	70
REFERENCES.....	75
APPENDIX A – SIMULATIN DATA TABLES.....	78

List of Tables

Table 1 – AMR-CSO Event schedule by day-of-week	7
Table 2 – Workdays to complete CSO orders	9
Table 3 – Cluster 4 Create Module Specifications.....	30
Table 4 – AMR Driver Schedule.....	32
Table 5 – B-Level Tech Schedule	39
Table 6 – AMR Driver Schedule for ORR.....	49
Table 7 – CSO Orders Released by Day of Week.....	49
Table 8 – Workday Targeted Operation Schedule	53
Table 9 – Cluster 1 Read Dates.....	57
Table 10 – Outputs for top performing experiments relative to Base	69

List of Figures

Figure 1 – Each Cluster’s contribution to total meters and total CSO orders..... 4

Figure 2 – Number of CSO order released by day of week..... 8

Figure 3 – Workflow operation sequence 24

Figure 4 – 2011 Meters read per day 28

Figure 5 – 2011 AMR Drivers per day..... 28

Figure 6 – Distribution of B-Level Techs per day 37

Figure 7 – Distribution of A-Level Techs per day 37

Figure 8 – Scheduled A-Level Techs per Order in Queue..... 38

Figure 9 – Simulation Meters by Cluster. 40

Figure 10 – CSO Orders by Cluster 41

Figure 11 – Average CSO Machine Queue Length 43

Figure 12 – Average Maximum CSO Machine Queue Length 43

Figure 13 – Average Days in CSO Machine Queue..... 44

Figure 14 – Average Time in CSO Machine Queue 46

Figure 15 – Average CSO Machine Queue Length 47

Figure 16 – Average Maximum CSO Machine Queue Length Attained 47

Figure 17 – Average Time in CSO Machine Queue 50

Figure 18 – Average CSO Machine Queue Length 51

Figure 19 – Average Maximum CSO Queue 51

Figure 20 – TR Scenario Comparisons Time in CSO Machine Queue 59

Figure 21 – TR Comparisons Average CSO Machine Queue Length 59

Figure 22 – TR Comparisons Maximum CSO Machine Queue Length..... 60

Figure 23– Average Days in CSO Machine Queue..... 63

Figure 24 – Average CSO Machine Queue Length 64

Figure 25 – Maximum CSO Machine Queue Length..... 64

Figure 26 – Time in CSO Machine Queue..... 66

Figure 27 – Average CSO Machine Queue Length 67

Figure 28 – Maximum CSO Machine Queue Length..... 67

1 INTRODUCTION

1.1 Background

A local utility company, hereafter the Company, processes a variety of jobs each day including meter reading, service shut-offs, emergency response, and customer service work. Due to a high degree of unpredictability of emerging work as well as location of scheduled work, and varying time-constraints for different job-types, the Company utilizes a highly adaptive scheduling policy. For example, highly-skilled service technicians, A-level technicians, are assigned a slate of customer service jobs at the beginning of a shift. But as emergency jobs arrive at the Company, these service techs will be pulled from customer service jobs to resolve the emergency; because service shut-offs require a lower level of technical skill they are primarily assigned to lower-skilled technicians, B-level techs. Because service shut-offs must be completed before 5pm and customer service work must be completed before midnight, a large number of shut-off orders on a day might overwhelm the capacity of the B-level techs and mean that A-level techs will be pulled from service work to complete them.

In each of these examples, and other events not described, service work may be pushed to later hours of the day, resulting in overtime premiums and poor customer service. While the Company makes every attempt to avoid these negative consequences, the option to delay customer service time has been a convenient release valve to alleviate job bottlenecks.

However, the Company is approaching a change to their customer service model which will make delaying customer service appointments even more costly.

The Company will implement much smaller customer service appointment windows. Failing to complete jobs by their scheduled service window will result in financial penalty for the

Company. This means the Company will no longer be able to utilize flexible scheduling to shift capacity in order to meet its variable workload demands. One strategy to cope with workload variability is to increase total capacity. However, excess capacity will result in idleness when demands are low. The costs associated with idleness are inevitably passed on to customers in the form of higher rates. This thesis will explore methods of reducing workload variability with the goal of minimizing resource drain from other workflows.

1.2 Problem Description

1.2.1 Workflow

For the Company, a specific workflow begins with automated meter-reading (AMR) and ends with collections/service shut-offs (CSOs) for accounts with excessively late payments. This will be called the AMR-CSO workflow. We can think of each operation performed in the AMR-CSO workflow as a machine, and each job as an input to that machine. A job, or more accurately a batch of jobs begins as a set of meters to be read on a given day. The batch is released to the first machine in the flow – AMR drivers.

There are roughly 700,000 meters in the Company's system. These meters are grouped into 137 AMR routes. These 137 individual routes are clustered into 21 clusters – groups of routes read over a 21 consecutive-workday period. In order that the amount a customer is billed from month to month be kept consistent, NW Natural tries to keep a 30 day billing period. That is, every attempt is made to read each meter 30 days from its last reading. In practice, however, there is a typical range of 28 - 33 days for meter reading. Further, by Public Utility Commission regulation customers may only be billed once per month.

After the meters are read, the jobs become a batch of bills to be processed by the ‘billing machine.’ Some bills will not be paid on time. These jobs will then be processed by the ‘late notice machine,’ and so on. The final output in this workflow is a CSO order for bills that remain unpaid.

1.2.2 AMR – Batch Sizes

Jobs (meters to be read) are not input one at a time, but rather are batched so that a set will be processed together. A significant source of input variability is the meter batch sizes. We will refer to batches of meters as clusters. As was outlined earlier every cluster will ultimately produce a batch of CSO orders. Not all clusters are created equal. Clusters have varying number meters in them and some clusters are likely to produce more CSO orders than other routes.

There are typically 7 routes per cluster. This means that 7 routes will be read per day. The routes were grouped into their specific clusters so as to minimize travel time between clusters. Less emphasis was placed on daily input levels (number of meters per cluster) and output levels (probability of generating CSO orders). Some clusters have total meters numbering in the low 20,000s while others have in the low 40,000s. Some clusters may produce significantly more CSOs than others which will lead to highly variable resource demands from day to day during the final stage of the workflow.

Figure 1 shows the percentage of each cluster’s contribution to the total number of meters in the system and the total number of CSO orders in the system. We can see that there is tremendous variability between clusters.

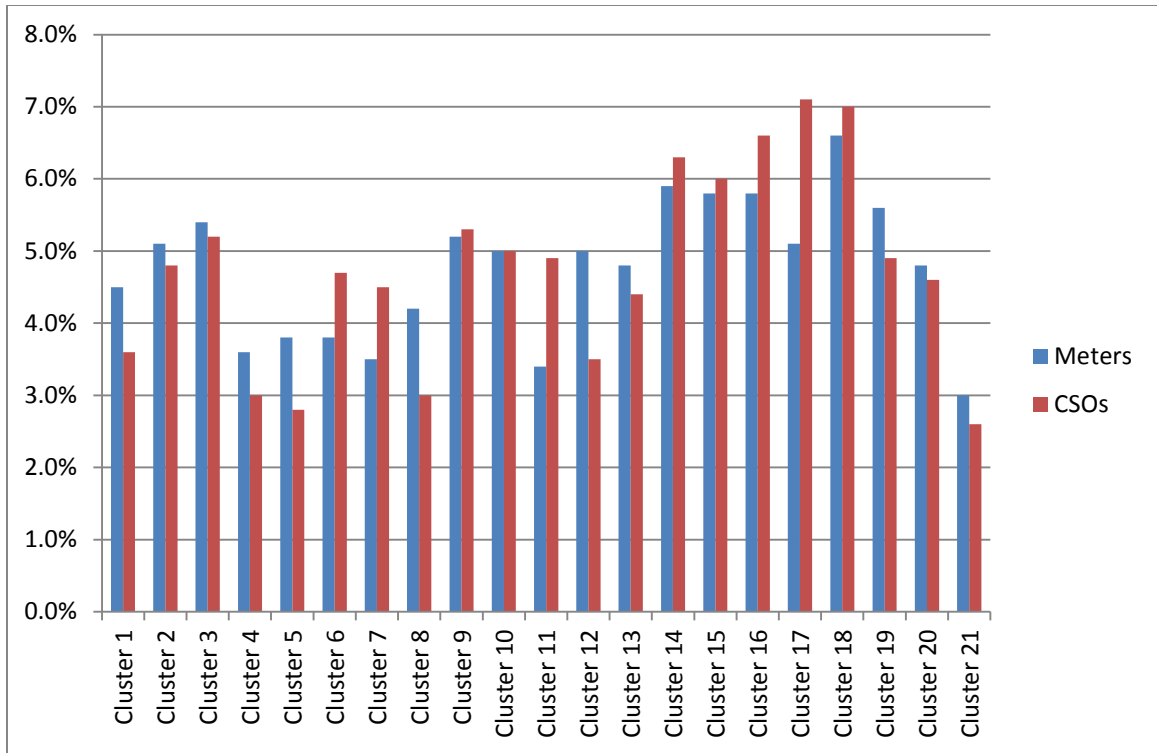


Figure 1 – Each Cluster’s contribution to total meters and total CSO orders

1.2.3 AMR – Order Release

Orders (clusters) are input to the system when they are read by AMR drivers. The clusters are read once roughly every 30 days, roughly sequentially by cluster number. On or around the beginning of a month, Cluster 1 will be due to be read and the drivers will read those specific meters. The next day the Cluster 2 meters will be available and so forth. The meters are read 5 days a week. Clearly 21 is not evenly divisible by five. So there is no consistency as to which day of the week a cluster will be read. Cluster 1 could be read on a Monday one month and a Wednesday the next month.

One goal of the schedule is full AMR driver utilization. AMR routes are read every workday, Monday through Friday. At the beginning of a shift, an AMR driver is assigned that day’s route. If she finishes the route before the shift is finished, she may begin work on the next

day's route. For this reason, consecutive clusters tend to be contiguous; tomorrow's routes are next to today's routes in order to minimize travel time between them and enable working ahead.

Releasing some jobs (clusters) early allows the AMRs to be done slightly ahead. Combined with the fact that there are typically 29 days, rather than 30, scheduled between meter readings to allow AMRs to be done slightly behind, AMRs can be managed without coming up against a hard deadline and thus incurring overtime costs.

There are a variety of ways this order release strategy can create downstream variability. First, the early practice of releasing orders early runs the risk of routes getting too far ahead of schedule. This can mean that billing periods can be too short, leading to amount variability for the customer, as well as the danger that they could be billed twice in one month. To avoid this, AMR routes must be delayed, leading to low resource utilization, and wasted money for the Company, as well as customers.

Second, the fact that the day of the week a particular cluster is read shifts from month to month limits visibility about where the Company will need to position its resources.

Resources might be in a convenient position to work Cluster 5 on a Wednesday, following working Cluster 4 on a Tuesday. But there might be a big difference in travel demands to work Cluster 5 coming off a weekend. This is a potentially fascinating issue, but it is outside the bounds of this project.

The third and most critical way the order release process creates variability downstream also arises due to the inconsistency of which day of the week a cluster is read on. This is

specifically related to the scheduled wait-times within the system which will be discussed in the following section.

1.2.4 Scheduled Wait-times

The scheduled wait-times between operations are another source of stochasticity. If a meter is read on day 1, then the series of operations in the workflow is outlined below along with the day in the cycle it takes place:

- 1 – AMR
- 2 – Bill Issued
- 17 – Bill Due
- 24 – One Week Late Notice
- 40 – Shut-Off Date Notice
- 44 – Route Desk
- 45 – In the Field for Shut-Off
- 55 – Shut-off Due Day

For our purposes the most important day in the workflow cycle is 45, the day jobs are released to the final machine in the flow – the ‘CSO machine.’ The scheduled timing of these releases ensures that jobs will be scheduled to be released to this machine on Fridays. This is critical to note, because the ‘CSO machine’ may not be run on Fridays through Sundays. Therefore jobs released on Fridays can be thought of as sitting in the queue at the machine, simply waiting until Monday when the machine can be run again. This means that a disproportionate number of orders are in the queue Monday on mornings.

To make this clear Table 1 shows events in the workflow as well as the day in the work cycle it is to be performed. For instance, if a meter is read on a Monday, the bill for that account will be sent on Tuesday and, should it become a CSO order, it will be released to the CSO

machine on a Wednesday. As we can see, meters read on both Wednesday and Thursday will be released to the CSO machine on Monday.

We must also take into consideration the input preference described in section 1.2.3.

Because the company cannot perform the CSO operation on Fridays, they have more resources to perform other work. A common practice is to utilize these additional resources to perform the AMR job relative to other days of the week. This means that we should expect more meters to be released to the CSO machine on Tuesdays than on Wednesdays or Thursdays.

	Event (Day in Cycle)			
	Read Day (1)	Bill Sent (2)	Bill Due (17)	Released to CSO (45)
Day of Week	Monday	Tuesday	Wednesday	Wednesday
	Tuesday	Wednesday	Thursday	Thursday
	Wednesday	Thursday	Friday	Monday
	Thursday	Friday	Monday	Monday
	Friday	Monday	Tuesday	Tuesday

Table 1 – AMR-CSO Event schedule by day-of-week

The data confirm that these processes have the expected consequences. Monday had by far most CSO orders released with 15400, followed by Tuesday with 12000, Wednesday with 9600 and Thursday with 8500 (Figure 2).

We can see now how these scheduled wait-times, the order release decisions and the variability in batch-sizes can align to create substantial downstream workload variability. To illustrate, we can refer to Figure 1. Clusters 17 and 18 are not only significant contributors to the total meters read, but they are also both heavy contributors to CSO orders. Consider the impact of these clusters being read on a consecutive Wednesday and Thursday. This would

likely mean an explosive CSO queue on the eventual Monday these orders would hit the shut-off machine.

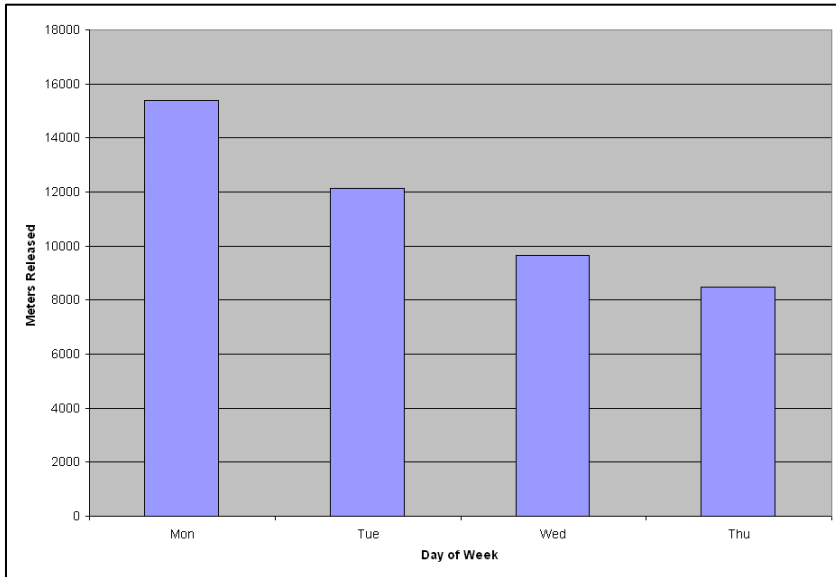


Figure 2 – Number of CSO order released by day of week

An additional problem is that for the jobs released on Wednesday and Thursday the delivery date for these jobs falls on a non-work day. Specifically, jobs released on a Wednesday are due on a Saturday, and jobs released on a Thursday are due on a Sunday. This forces managers to move these jobs up in order to ensure completion before the due date. Late delivery for these jobs is not considered an option.

These problems reduce the workable lead-time for the last machine in the flow, as the final due date is set from the time of the AMR and does not change. As you can see, there is a supposed scheduled lead-time for these jobs of 11 days. But the workable number of days for jobs is actually much less than that. The table below shows the typical effective lead times for jobs by the day of their release to the machine. The actual lead times could be even less in the case of weeks which have holidays on Mondays, etc. The average effective lead-

time for any batch of shut-off jobs is 6.2 days. This means that nearly half of the scheduled lead-time for these jobs is rendered non-existent by non-workdays.

Scheduled Release Day	Number of Workdays
Monday	8
Tuesday	7
Wednesday	6
Thursday	5
Friday	5

Table 2 – Workdays to complete CSO orders

1.2.5 Job Dispatching

Completion of work may not be a problem in principle. There is no reason that all jobs cannot be accomplished with managed oversight. The problem is that this process does demand managed oversight. Ideally, jobs would be released to a machine without any queue and only the number of jobs that could be processed in the processing time interval would be released to the machine. This would allow the jobs to be processed and exit the machine with negligible queuing time and leave the machine ready to receive additional jobs at the beginning of the next release period.

As currently organized however, there is necessarily a queue waiting for many jobs, and in practice there is a queue waiting for all jobs. Any queue of jobs waiting to be processed by a machine means that management must dispatch the jobs to the machines for processing. Numerous dispatch rules have been developed for use in machine-shop floor management. The Company follows at least two distinct and well researched dispatching rules for dispatching jobs to the CSO machine. On any day, some batch of jobs has ‘matured.’ That is, a batch of jobs is within three days of its due date and has become a high-priority. There

is a specific class of technician, B-level techs. The main priority of this class of tech is to perform the shut-off operation. The high-priority shut-off orders will be assigned first and they will be assigned first to the B-level techs. If there are a very high number of high-priority shutoffs, the number that exceeds the capacity of B-level techs will be assigned to A-level techs. This dispatching rule could either be thought of as First Come First Served (FCFS), or as Shortest Due Date (SDD). As all jobs have the same cycle- and (effectively) processing-times, the due dates are entirely determined by the input date. Therefore, for this problem the two dispatching rules are equivalent.

Since there are between five and seven workdays between release and due date, when jobs are released to the final machine they are not considered an immediate priority. Once all high-priority orders are dispatched, lower-priority orders are assigned as slack permits. They are dispatched to field technicians not on by FCFS or SDD, but by a proximity rule – if a tech is projected to be near the order, it will be dispatched to them. This is done as a standard part of a B-level tech’s workloading. However, these low-priority orders will only be assigned to A-level techs if they have a light workload on that day. As this is field work, travel time to the job is considered. Therefore, travel time can be thought of as a portion of the total processing time. Jobs are assigned if the travel time is below a certain threshold.

The dispatching rule in this case can be thought of as Shortest Processing Time (SPT).

1.2.6 Maintenance

The Company must regularly pull field technicians out of the field for a variety of meetings. As the heading of this section suggests, these meetings can be thought of as routine maintenance of shop ‘machines.’ These meetings must be scheduled during standard work hours. While some effort is made to schedule these meetings according to season, daily

workloads are considered exogenous variables. Since there is little control or projectability of daily workloads, meeting schedules are made largely independent of daily workloads. This means that frequently large numbers of workers are pulled off the field and are unavailable to process jobs, on days with high job demands.

Compounding the problem, these meeting schedules are not well communicated to the job-dispatching department. While efforts have been made to increase communication between the meeting-scheduling and job-dispatching departments, these meetings continue to catch dispatchers by surprise, leading to emergency rescheduling. A frequent result is that jobs must be scheduled for later in the day and technicians must be kept later than their scheduled shift. This is a significant cause of over-time hours for the company.

1.2.7 Project Goals

As we can see from the preceding account, there are considerable and systemic sources of variability in both the workload and resource demands of the AMR-CSO workflow. This variability regularly impacts other workflows in the Company system. Ideally the AMR-CSO workflow would be entirely independent, never demanding higher value resources from other workflows.

We hypothesize that the variability of inputs, specifically the batch-sizes, order release mechanism, and intra-system wait-times schedule, are the conspicuous cause of much of the downstream workload variability. This project will explore possible means to mitigate this variability by proposing an integrated approach to workload control through the use of a novel billing cycle model which will:

1. Correlate job release dates with work days;

2. Incorporate novel order review/release rule;
3. Provide a framework to balance workloads;
4. Create predictable periods of low workload in order to schedule maintenance.

Using discrete event simulation, this project will assess the contribution of each facet of the workflow to the variability measure. The ultimate goal of this project is to determine the extent to which variability can be limited and workloads and resource demands can be more tightly grouped around the mean. We believe that there is in principle some allocation of meters to routes, grouped into some set of clusters, which when processed over an efficient workflow schedule will lead to a near-optimal and highly consistent resource demand level.

2 LITERATURE REVIEW

The billing/shut-off cycle workflow can be thought of as a machine or job shop. Specifically, as this work is repetitive month after month, this workflow can be thought of as a cyclic schedule job shop problem. An ideal cyclic schedule will systematize uniform output of end-items. If end-items can be produced at a uniform rate, all contributing materials and processes can be made uniform [1]. However, the schedule employed by the Company is not designed to generate uniform outputs.

2.1 Operation Scheduling.

The basis of the job shop problem is the question of how to schedule order processing operations on the various machines in a shop. There are numerous variations on this problem including the single machine problem [2][3][4][5][6][7][8][9], the multiple machine problem, identical machine problem, and a variety of others. The central question of this class of problems is how to order the jobs that need to be processed in such a way that minimizes some performance measure such as total makespan, work in progress, or average job tardiness.

If different sections of a job can be scheduled on different machines, then the scheduling process is two step: determine what length of job will be scheduled on which machine, then determine the order in which each machine will process its various assignments. Each step has been shown to be equivalent to a traveling salesman problem. Yalaoui and Chu propose first a branch and bound algorithm [10] then simplify it as a heuristic [11] solution to the job scheduling step while Tahar et al. [12] offer a linear programming solution. The method is fast and achieves balanced workloads across the parallel stations. But these solutions do

nothing to address the variability described in Section 1. In fact, these solutions proposed are largely consistent with the approach currently employed by the Company.

2.2 Cyclic Scheduling

The sub-topic of cyclic scheduling is small compared to many other sub-topics, but still well researched and diverse. Broadly speaking, research can be divided into investigations of single product and multiple product lines. As this project investigates a line producing a single product, we restrict the review to those papers.

Graves et al. appear to be among the first to develop the cyclic schedule concept. They investigated a circuit manufacturing plant which scheduled jobs according to input/output needs and local sequencing rules at each machine [13]. They found that these procedures lead to a large work-in-process inventory. Holding set-up and transfer times negligible, they formulate the problem first as a combinatorial optimization problem which they deem unsolvable. They then develop a heuristic scheduling algorithm by which they set a desired output rate then determine the schedule of tasks and production rate to achieve the desired output. This method was successful in both reducing throughput time, as well as stabilizing workforce assignments.

Hall provides a qualitative assessment of the benefits of cyclic scheduling [1]. Aldakhilallah and Ramesh develop two scheduling heuristics for cyclic, re-entrant job shop environments which produce a single product [14]. They use a mixed-integer program to determine cyclic schedules for a repetitive production re-entrant job shop with a predetermined sequence of operations with known processing, set-up and material handling times as well as a specialization of that environment in which setup for a job can begin on a machine before completion of the previous operation. They attempt to minimize flow-time (work-in-

process) for a given cycle-time (throughput). While their algorithms are shown to be both efficient and effective, they are so by re-ordering the sequence of operations.

Kouvelis and Karabati explore cyclic scheduling of unpaced, synchronous production lines [15]. They develop an implicit enumeration algorithm which approaches optimality and can solve realistic sized problems. Wójcik investigates repetitive manufacturing systems of multiple processes utilizing a shared processing resource. [16]. He uses a constraint propagation program to narrow the possible solution set of conflict-free schedule, which ensure that the shared resource is not requested by more than one process during a single processing interval.

2.3 Workload Control (WLC)

The problem of workload variability is well known. As Irastorza and Deane wrote over 40 years ago, workload variability results in costs from idle machinery and labor, overtime wages, or the costs of utilizing resources for out of the ordinary operations [17]. While there is wide recognition of input variability, research tends to deal with coping with variability, rather than affecting it directly. Eilon et al. state that, “if the arrival of jobs, their processing requirements and facilities are given, the only control parameter at the disposal of the dispatcher is...the order in which the job should be processed [6]. Melnyk et al. observe that “[f]actors such as arrival rates, shop loads and processing times...are treated as set by forces outside the control of shop personnel [19].

Shimoyashiro et al. treat the problem of load balancing through input scheduling [20]. They treat a shop as a vector of processing capacities and a job as a vector of processing needs, and develop an algorithm for matching similar vectors.

Bechte [21] provides a broad overview of various aspects of load-oriented manufacture control including order entry, order release, and operation sequencing (dispatching). His discussion is oriented toward non-cyclic systems with variable inputs which are loaded during weekly planning periods. He develops a job release protocol which releases jobs until a load limit is reached at a single work station. This system also relies heavily on dispatching rules and the entire system requires significant human management.

Land and Gaalman [22] review a number of WLC and find each suffering from an assumption of stationarity and stability of both jobs and capacities. However, in order to achieve such stationarity, the WLC reviewed would likely require lower-overall throughput to ensure queues maintain expected norms.

There are many proposed methods to cope with input uncertainty. A major area of investigation in the job shop literature is Order Review/Release (ORR) strategies.

In an early investigation of ORR, Baker [23] develops a load-oriented rule for a very simple single machine job shop. Baker finds little benefit of ORR and concludes that proper dispatching is much more critical to shop success. Curiously however, Baker suggests situations in which input control may be useful, such as reducing confusion on an overloaded shop floor or when there are frequent changes to a master schedule, yet he does not test either of these scenarios in his simulation.

Melnyk et al. [24] suspect the reason ORR methods have had little impact in practice on decreasing lead-times is that models treat the planning system as a stochastic process outside the control of the system being investigated. In a simulation experiment they find that load smoothing prior to ORR has a high impact on tardiness and flow-time variance measures,

while ORR had a greater impact on work-in-process measures. Also, the combination of workload smoothing and controlled release diminishes the importance of complicated dispatching rules.

A later study by Melnyk et al. [19] studies the interaction effects of variance control, ORR and dispatching rules. They find that ORR can have positive impact on shop floor functioning, but only in the presence of variance control both at the planning stage (load-leveling) and at the shop floor. Consistent with previous research, they find that without any system control, the Shortest Processing Time (SPT) dispatching rule outperforms all other dispatching rules. However improving job processing time variance made SPT the worst performer among all dispatching rules.

Philipoom and Fry [25] note that the majority of studies on ORR take for granted that all orders will be accepted by the job shop, independent of shop conditions. They investigate the effect of rejecting orders if accepting them will push the shop above a maximum work threshold. By varying the threshold for rejecting work, Philipoom and Fry show that for three different order release mechanisms some increase in rejecting orders has beneficial effects on mean flow time and various tardiness measures. Importantly, they show that a work path release mechanism is never worse and frequently better than a shop load release mechanism.

Ragatz and Mabert consider the case in which due date assignment and operating decision rules are interdependent. They note that if due-date feedback influences job flow times may make setting each very difficult if due dates do not stabilize quickly. Without quick convergence, it may be necessary to have a short-circuiting rule [26].

Davis et al. [27] develop a variety of workload imbalance measures which go beyond the common convention of input stochasticity. They show that high levels of worker flexibility are an effective strategy to cope with imbalanced workloads.

Bott and Ritzman [28] found that even with ample capacity slack, workload variability can hamper on-time delivery and severely impact total inventory. Further their results suggest that the second most critical factor affecting workload variability, after product complexity, is demand variability.

One area of the WLC literature investigates pull or feed-forward systems [29][30]. The goal of these approaches is to avoid machine starvation and their by decrease lead times by pulling orders forward to machines which fall below an established queue threshold. This course is not an option for the Company in question as the release times to each machine in the AMR-CSO workflow are regulated by public authority.

2.3.1 Order Release Mechanism (ORM)

An ORM is the timing convention by which orders are released to the shop floor. There are a variety of ORMs considered in the literature including Deterministic Input, where orders are released at constant intervals; Closed Loop Input, where the number of orders in the shop is held constant; CONWIP, a form of Closed Loop Input where the WIP is kept constant; and Starvation Avoidance which focusses on the bottleneck machine[31]. These techniques also fit into Wisner's categories of Finite Loading policies, where orders are released when certain shop floor conditions are met, or Infinite Loading policies, where orders are released at a predetermined release date, regardless of shop conditions [32]

But by regulation, the Company must use a form of Infinite Loading or Deterministic Input ORM. It must release a batch of orders roughly every 30 days.

2.3.2 Machine Unavailability

There is a surprising lack of research concerning the variability of machine capacity. In a study of the CONWIP ORM, Hopp and Spearman write that a majority of the variability present in a job shop system is due to shop floor conditions such as random machine failures, periodic adjustments and inattention from the operator [33]. In an investigation of various environmental factors on the performance various ORR techniques, Cigolini et al. found that the availability of machines can have a great impact on performance [34].

This has been a brief review of some of the concepts we will explore below. We feel this project builds on much of this work. In particular we expect to add to prior work on workload control by measuring the impact of various types of input variability. We also feel our case study fills a gap by providing a platform to test the effects of adjusting multiple forms of input variability simulataniously.

3 Methods

In many ways the AMR-CSO problem is similar to the identical parallel machine scheduling solution described by Tahar et al. and Yalaoui and Chu. The machines (technicians) are identical in so far as they are capable of performing the same operation in parallel. The jobs are in fact batches which are split between the various machines. Setup costs are a major consideration for scheduling machines (technicians). The Tahar et al. solution is a close model for the current scheduling approach to AMR-CSO. With some modification, we will rely on their notation to describe the current scheduling process.

R	set of meters,
r	index of meters ($r = 1, \dots, R$),
J	set of routes, where a route is vector of meters $\langle r_1, \dots, r_n \rangle$ for all $r \in R$, such that all meters appear in one and only one route,
j	route index ($j = 1, \dots, J$),
\mathcal{J}	a unique map of R to J ,
K	set of route clusters, where a cluster is a vector of routes $\langle j_1, \dots, j_n \rangle$ for all $j \in J$, such that all routes appear in one and only one cluster,
k	cluster index ($k = 1, \dots, K$),
\mathcal{K}	a unique map of J to K ,
O	total operations,
o	index of operations ($o = 1, \dots, O$),
P	set of technicians to staff operation stations,
p	index of technicians ($p = 1, \dots, P$),
θ_{ko}	processing time of operation o on cluster k ,
C_{ko}	completion time of operation o on cluster k ,
S_{ko}	start time of operation o on cluster k ,
W_i	scheduled wait-time following operation O_i , ($w = 1, 2, \dots, O-1$)

A major difference between the two problems is that Tahar et al. and Yalaoui and Chu assume that each job requires only a single, identical operation, which is performed by all available machines. Jobs in AMR-CSO require multiple operations for completion and, more importantly, at any given start-time there is more than one operation performed in the workflow. These operations pull from the same set of technician resources, however, so they are modeled as a single consideration. However, this difference can still be treated as a routing problem based on setup costs of machines on operations as will be seen.

If each $k \in \mathcal{K}$ is a job, the final goal is to determine $Q_{k,p}$, some length of k , (for instance, a specific AMR route, or set of CSOs) to be allocated to technician p . This is a two-step process. At any S_{k_o} there is at least one job k scheduled to receive the operation o , $o = 1, \dots, O$, represented by the set O_k . The first step in scheduling is determining which technicians will be assigned to each operation. For Tahar et al. this problem is determining the sequence of jobs to process based on setup costs between specific jobs, which they solve using Little's traveling salesman algorithm.

Our problem involves the setup costs for a technician at for a specific operation, $\sigma_{p,o}$. Using a matrix of setup times, the same algorithm could produce O_p , a schedule of all technicians to one and only one operation at S_{k_o} . Once O_p has been created, the second step is determining $Q_{k,p}$. Both steps are represented simultaneously in the objective function below.

For Tahar et al., the goal under this formulation is to minimize the maximum makespan for all jobs by first allocating sections of jobs to all machines and then optimizing the job order on each machine based on sequence-dependent setup times. The Company's goals for AMR-CSO scheduling are slightly different. The goal here is to minimize the total penalty for tardy

or early completion of jobs. As described by Panwalker et al.[3] if each $k \in O_k$ has due time D_{ko} then the tardiness and earliness of k are

$$T_{ko} = \max(0, C_{ko} - D_{ko}) \text{ and } E_{ko} = \max(0, D_{ko} - C_{ko})$$

respectively. If A is a schedule of $Q_{k,p}$ on O_p , and the total penalty for earliness and tardiness for A is given by

$$f(A) = (\alpha_1 E_{ko} + \alpha_2 T_{ko})$$

The problem is thus minimax $f(A)$ by utilizing $Q_{k,p}$ and O_p as decision variables for

$$\sum_{p \in O_p} \sum_{k \in O_k} (\sigma_p + Q_{k,p})$$

subject to

$$\sum_{k=1}^K \sum_{p=1}^P (\sigma_p + Q_{k,p}) = \theta_{ko}$$

$$\sigma_p + Q_{k,p} \geq 0, \quad O_k \cap O_p = o; k = 1, \dots, K; p = 1, \dots, P$$

$$\sigma_p + Q_{k,p} = 0, \quad O_k \cap O_p = \emptyset; k = 1, \dots, K; p = 1, \dots, P$$

There are many features of AMR-CSO that allow it to be modeled as a cyclic schedule. The process is repetitive – the AMR-CSO workflow processes jobs that are largely similar month after month. Individual jobs even re-enter the workflow for reprocessing every month. The sequence of operations is standard across all jobs. Under the Tahar et al. paradigm, which is the basic process utilized currently, the AMR-CSO is trivially cyclic.

The actual \mathcal{H} is considered set and roughly every 30 days, \mathcal{H} starts over with cluster K_1 at operation O_1 . If a schedule A were cyclic in a meaningful sense, then it would repeat every Z time steps such that:

$$A_{S_{1,1}} = A'_{S_{1,1}+Z} = A''_{S_{1,1}+2Z} = \dots, A^{\alpha}_{S_{1,1}+\alpha Z}, \quad \forall S_{k,o}, k = 1, \dots, K; o = 1, \dots, O;$$

However, no such Z exists. Due to the causes of variability outlined above, daily schedules are unlikely to repeat at all, and even if they happen to do so, there is no reason to expect that the following day's schedule would be a repeat of the previous cycle's schedule. O_5 may require 10 technicians one day and require only 3 the next. In a truly cyclic schedule the number of resources necessary at any operation would be largely consistent from day to day.

The cycle time is effectively infinite. The consequence of this fact is that schedules must be highly managed on at least a daily basis and in reality managed throughout the day. Our goal is to determine the extent to which variability can be eliminated and the scheduling procedure be rendered cyclic. If a cyclic schedule can be created for the AMR-CSO workflow, it could reduce both production and management costs.

One major source of variability is the input variability – the difference in meters per cluster and likelihood of downstream work generation. The generation of a truly cyclic schedule would require inputs that were balanced or nearly balanced. This would be accomplished by redrawing the routes and re-clustering them so that each route has a similar number of meters and each cluster is likely to lead to a similar number of CSO orders.

The task of converting the current $\mathcal{H} \rightarrow \mathcal{H}^*$, some K which balances workloads across the workflow, is beyond the scope of this project. To even begin to do so one would need to

know how such an input schedule would interact with the operation schedule – the scheduled wait-times between operations.

Instead we will investigate possible means of reducing workload variability in order to achieve a specific and realistic Z. To do this we will first determine the extent to which the operation schedule contributes to workload variability using discrete event simulation in a multi-step process. First a simulation will be built to recreate the behavior of the actual system. The simulation will pass clusters through a series of decision points and processing modules. The size (number of meters) and probability of passing through various decision points will be informed by real Company data from a representative year.

Model specifications will be given in the next section, but generally if the operation schedule is defined as the set of wait-times W , between operations such that $(w = 1, 2, \dots, O-1)$, the figure below illustrates a workflow of operations and wait buffers as well as the basic pattern of the simulation:

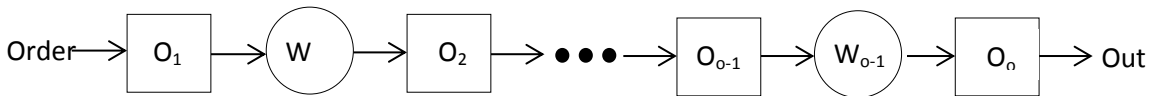


Figure 3 – Workflow operation sequence

The simulation will be validated against other sample years from company data. Once we are confident the simulation reliably reflects the real system we will investigate a variety of tactics to reduce input variability and determine their impact on downstream workloads. Our first conjecture is that the variability in batch-sizes creates variability in CSO machine queue length and order processing times. As currently scheduled there are at least two sources of lot-related variability. Neither the routes nor clusters of routes were created with balancing in mind. Some clusters have in the low 20,000s of meters while others are as large as the low

40,000s. Secondly, some clusters are consistently more likely to produce downstream workloads in the form of CSOs. The regrettable truth is that customers in more affluent clusters have a higher ability to pay their bill than customers in less affluent areas.

We will assume the existence of some λ^* which balances both the input and downstream workloads. We will examine the impact these rerouted clusters have on downstream workload variability.

Our second conjecture is that the order release mechanism creates variability by inputting orders without respect for downstream timing. We will attempt to adjust the release mechanism so that downstream workload patterns are predictably level.

The third conjecture is that wait-times between events create workload variability by releasing orders on non-workdays and thereby inefficiently stacking jobs and creating long queues and processing times. We will reschedule the wait-times so that they target releases with workdays.

In these three experiments, our goal is not workload reduction, but workload balancing. We do not expect or seek reduction in average or total processing times or utilization metrics. We will measure impact of experiments by the average length of CSO processing queue, maximum length of CSO processing queue, and the average time an order spends waiting in the CSO processing queue. We expect to see minimum and maximum values closer to the mean. We will be particularly concerned if there is a reduction in the number of additional high-value labor hours necessary. If there is little variability in resource demand, then theoretically the Company could set actual resource level at this demand and therefore

minimize the need for the AMR-CSO workflow to consume higher value resources from other workflows.

4 SIMULATION

4.1 Design

In order to investigate these questions a Discrete Event Simulation (DES) was created using Rockwell Arena 11.00.00 - CPR 7 and run on an HP Compaq nc6400. In consultation with the Company, 2011 was determined to be a representative year and all data used to inform the simulation was obtained from Company records from this year. This section will provide a detailed explanation of the base simulation so that deviations from it can be described briefly in the chapter 5 Experiments.

4.1.1 Entity Creation

The specific entities for this simulation are meters. The individual meters will move through the simulation and seize the work of various resources such as AMR drivers and technicians to perform the CSO. The first decision to be made is how to load new entities into the system. In Arena, entities are created with a CREATE module which specifies the Entity Type (meters), the Time Between Arrivals and the Entities per Arrival.

Our first step is to determine how many meters should be created with each arrival. We know that meters arrive to the system every workday. One possible approach is to look at the actual number of entities read per day and load that number into the queue each day. In 2011 the Company read a total of 8360296 meters over 252 work days. When the number of meters read per day is plotted as a histogram of 20 bins (Figure 4), the distribution is highly normal with a mean of 33200 and a standard deviation 8330.

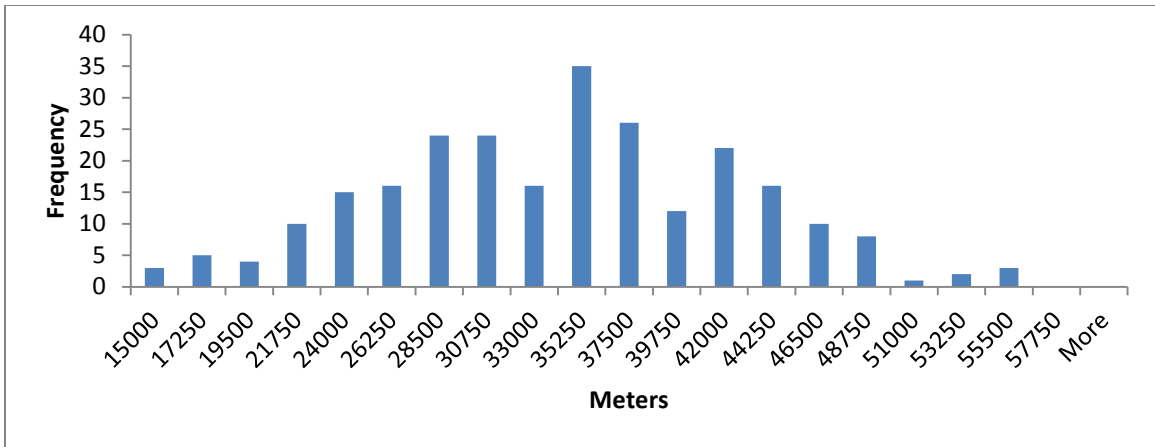


Figure 4 – 2011 Meters read per day

However, if we look at the number of AMR drivers per day we find that this distribution, too, is highly normal (Figure 5) with a mean of 7.21 and a standard deviation of 1.2. This approach seems to put the cart before the horse. It seems highly likely that the normality of the distribution of meters read is due to the normal distribution of meter readers.

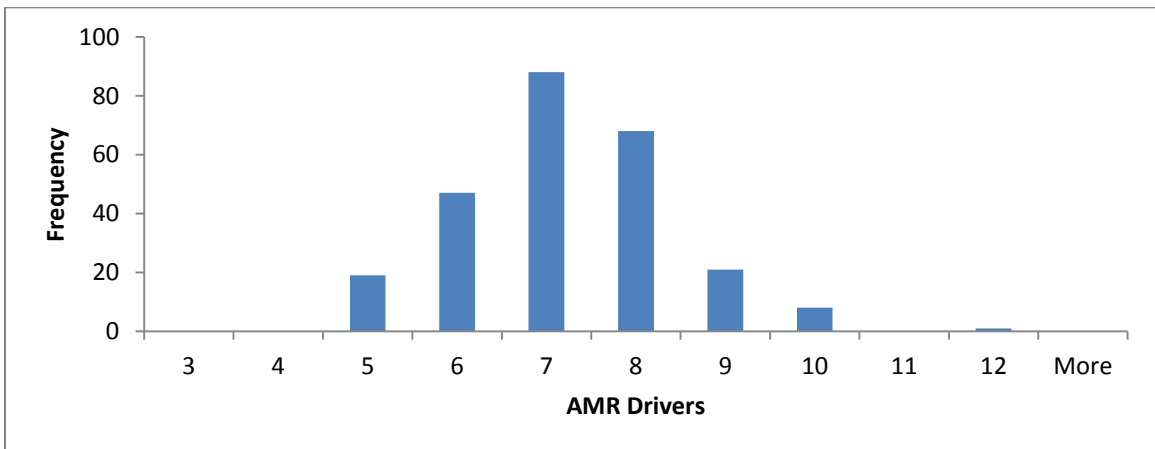


Figure 5 – 2011 AMR Drivers per day

A superior approach is to look at the number of meters actually queued by the system each day. The meters are grouped into 21 specific clusters, $(K_1, K_2, \dots, K_{21})$ and queued cyclically by cluster number. Each workday the next cluster in the sequence is queued and meters in the

cluster are made available to be read by AMR drivers. The simplest way to model this system is to have a single CREATE module create the average number of meters per each cluster.

If we count the number of meters in each cluster each month, we find, unsurprisingly, that it is highly similar to the distribution of meters read with a mean of 33200 and STD of 8030.

However, part of our thesis is that the system is prevented from reaching a steady state by differences in input variability. Creating cluster inputs around a single mean assumes that there is not difference in the actual means of the clusters. The simplicity of this approach is attractive, but in order to be certain that it adequately captures the behavior we are interested in we will check the differences between the clusters.

An entire year of data gives 12 observations for each cluster. As was outlined above, clusters vary by size, but within each cluster there is variability from month to month due to seasonality and service turn-ons and shut-offs. We assume the rough normality of each individual cluster and use a single factor ANOVA test to determine if the means of the clusters are significantly different. 21 groups gives 20 between-group degrees of freedom, and 231 within-group degrees of freedom, yielding an F Critical-value of 1.62 at a .05 level of significance. The F-test on the data gives an F-value of 23.1, so we can reject the null hypothesis and conclude that the means of at least some clusters are significantly different from others.

For our purposes it is unnecessary to determine which clusters are significantly different. Separating only those clusters from the rest and modeling the others collectively would add a layer of complexity over simply modeling each cluster individually. Each cluster is

represented by a unique create module specifying a unique distribution for Entities per Arrival. Distributions were derived from Company data using the Arena Input Analyzer.

The final step in generating inputs is to determine the time between each arrival. As we have outlined, roughly every workday the meters belonging to the succeeding cluster are made available to be read by AMR drivers. Therefore, one CREATE module per day should have an arrival event. With 21 clusters, there will typically be 29 days between arrival events for each cluster, including weekend days. For details of this scheduling system from a theoretical perspective please refer to section 1.3.6.

From a purely modeling perspective, scheduling strictly 29 days between arrival events would mean that arrivals would happen on Saturdays and Sundays. Since no AMR can happen on those days the entities arriving would wait in the processing queues for unrealistically long times which would skew output data. A strict 29 day inter-arrival time would also lead to too many arrival events over the course of a year. To avoid these problems each CREATE module is given an inter-arrival time as a triangular distribution in units of days with 29 days being both the minimum value and the mode value, and the maximum value being 31 days. To avoid partial days, the values are rounded to the nearest integer as ANINT(TRIA(29, 29, 31)). For an example of the details of a CREATE module see Table 3.

Name	Cluster4
Entity Type	Cluster4Meters
Entities per Arrival	1.93e+004 + EXPO(5.74e+003)
Time Between Arrivals	ANINT(TRIA(29, 29, 31))
First Creation	3.29
Time Units	Days

Table 3 – Cluster 4 Create Module Specifications

4.1.2 Signaling Track

In order to understand various aspects of the Processing portion of the simulation, it will be necessary to first describe an independent track within the simulation which may be referred to as the 'Signaling Track.' The Signaling Track creates a single entity, the Signaling Entity, every day at 8.01 hours. The entity proceeds immediately to a SIGNAL module which sends a signal of value '1' to the entire simulation. Next the entity moves to a DELAY module and is delayed for 10 hours. This means that the entity is released at simulation-time 6pm every day. It then moves to an ASSIGN module, which assigns the number of bills to be processed that day. More will be described about this module later. The final active module in the track is another SIGNAL module which sends a signal of value 2 to the entire simulation. The entity is then disposed and exits the simulation permanently.

4.1.3 Meter Processing

We can now discuss the major portion of the simulation – the meter processing. After an arrival event the batch of meters proceeds to a PROCESS module. This module simulates the job of AMR drivers reading the meters. Each meter will seize one driver for a specified processing-time. After the processing is finished, the driver is released back to the available resource pool and the entity exits the PROCESS module and proceeds to the next module. The maximum number of entities that can be processed at any time is equal to the total number of drivers in the resource pool.

The available data provide only the means to determine the average read rate per read day. Because these rates describe how long a resource is occupied by an entity, they are given as seconds/meter. The best fit processing time distribution for the data is 3 + LOGNORMAL(2.72, 1.28) seconds, where 2.72 is the LogMean and 1.28 is the LogSTD.

The meters entering the AMR Processing module must be processed by an available resource, in this case an AMR Driver. Arena provides two methods for making resources available, they could either be of a fixed capacity or on a schedule. Since the AMR Drivers do not work at all times, we use the scheduling method. The Drivers are scheduled Monday through Friday for a 7.5 hour shift. All that is left is to do is tell Arena how many Drivers are available during these shifts.

From the data we determine the likely number of Drivers working a generic shift is given by the distribution $ANINT(NORM(7.21, 1.2))$. However, we are told the Company prefers to schedule more drivers on Fridays because CSO orders may not be processed on Fridays and therefore more resources are available as AMR Drivers. We compare the number of Drivers by day of the week via single-factor ANOVA and verify that there is significant difference in the mean number of drivers by day of the week. While the mean for Fridays seems to be the most significantly different from the rest, as with the meter inputs, the simplest solution is to specify a unique distribution for each day of the week. The final schedule is given in Table 4.

Day of the Week	Technicians Working
Monday	$NORM(6.29, 2.09)$
Tuesday	$NORM(7.04, 1.06)$
Wednesday	$NORM(7.33, 1.16)$
Thursday	$NORM(6.8, 1.01)$
Friday	$WEIB(8.83, 4.82)-0.5$
Saturday	0
Sunday	0

Table 4 – AMR Driver Schedule

For Monday – Thursday, the number of technicians working is described by a normal distribution where the first number in the ordered pair is the average number of technicians working on that day of the week and the second number is the standard deviation. While the number of technicians working on Mondays – Thursdays is fairly similar, the number of

techs who worked Fridays during 2011 is described by a very different curve and a much higher mean number of technicians. When 9am on, say, a Monday arrives during a simulation run, a random number is generated according to the curve described by $NORM(6.29, 2.09)$. That is the number of technicians available to do work on that simulation day. In each case, the number generated is rounded to an integer so to avoid having partial resources, which cannot be used and would skew utilization metrics. The shift for these workers is 9am – 4:30pm. This job is not done on Saturday or Sunday, so no technicians are scheduled.

4.1.4 Billing

In the real system, at the end of a shift the data for the meters that have been read are uploaded to a central billing system. That night the bills are processed, those for customers receiving paper bills are printed and put in envelopes and mailed the following day. However, in an attempt to control downstream workload, the Company instituted a governor on the number of bills that are processed in a single day. The governor is generally around 38,000, but there is some variability.

In the simulation, once a meter is finished being processed in the AMR Processing module, it proceeds to a HOLD module called 'Hold for Billing.' The meters are held here until the HOLD module receives 'Signal 2' from the Signal Track at 6pm. At this time the module releases the minimum of either the total number of meters in the Hold for Billing Queue, or the number of meters allowed by the governor described by the distribution $TRIA(2.7e+004, 3.78e+004, 5.29e+004)$. The work of the bill processors is outside the bounds of this project and we simply assume that it is completed. Therefore we allow all the

meters released to proceed to a DECIDE matrix before proceeding to the Mail Bills PROCESSING module.

Since what this project is concerned with is how the variability at the input stream affects variability of the CSO orders, for the sake of computational efficiency we remove any meters which will not become CSO orders from the simulation. Again, the simplest method for doing this is to determine what percentage of all meters read will become CSO orders and apply this chance to all meters. But one of our hypotheses is that differences in rates by cluster will affect the downstream workload variability. Therefore for the base simulation each cluster will be given its own average rate of meters becoming CSO orders.

To accomplish this we took the simple average, $\frac{\text{ClusterX}(\text{total CSO orders})}{\text{ClusterX}(\text{total Meters read})}$, for the entire simulation year. Each meter passes through a DECIDE module to sort it by cluster, then it proceeds to a second DECIDE module which gives that cluster's likelihood of becoming a CSO order. For example, in 2011 .46% of all meters in Cluster 4 resulted in a CSO order, while .76% of all meters in Cluster 17 resulted in a CSO order. The DECIDE module flips a weighted-coin that comes up true .46% of the time (in the case of Cluster 4). When a false case occurs (clearly the majority of the time), the meter immediately exits the system permanently via the DISPOSE module. When a true case occurs, the meter moves on to be processed as a bill which will ultimately become a CSO order.

Like the work of the bill processors, the work of mailing the bills is not a concern of this project. However we do employ a Mailer resource in the Mail Bills PROCESSING module because it provides a convenient timing mechanism. The Mailer resource works the same time periods as the AMR Driver, Monday-Friday 9am-4:30pm. However, to ensure that no

meters are left in this module at the end of the shift, we make 1000 mailers available and make the processing time a constant .5 seconds per meter. When the meters reach the Mail Bills module at 6pm there are no resources working. This means that they must wait in the module queue until the following workday at 9am to be processed. Mailers do not work on Saturdays, so any meters read on Friday will wait in the queue until Monday.

4.1.5 Wait-Times

The next series of modules are timing and data modules. The timing modules ensure that events which require no work, such as bills reaching their due date or late notices being sent, occur on the schedule set by the Company. The data modules record certain data used for analysis such as the day a late notice was sent or how much time a CSO order spent in the processing queue.

4.1.6 Credit/Shut-off Processing

The final critical module in the simulation is the CSO PROCESSING module. There are two components to the total processing times for CSO orders – the travel time and the job processing time. The Company data includes En Route Time – the time of day the technician begins travel to the job, Order Start Time – the time of day the technician arrives at the meter and begins being processing the order, and Order End Time – the time of day the processing is completed. The period from En Route Time to Order End Time was used as the total processing time. In certain cases the En Route Time was later than the Order Start Time. These orders were excluded from the sample, totaling roughly 1,000 exclusions from a total of nearly 46,000. The remaining 45,000 entries were converted to total minutes. The mean processing time was 23.3 minutes and the likely processing time for an order is given by $-.001 + \text{LOGN}(22.6, 21.1)$ minutes, with a square error of $<.006$.

There are two different resources for the CSO Processing module – B-level technicians and A-level technicians. Each of these different resources is a member of the same set, but the number of each resource available at any given time is given by two different schedules. Each of the resources is scheduled Monday through Thursday, by regulation the CSO order may not occur on Fridays, for a 7.5 hour shift beginning at 9am.

Determining the number of resources to schedule for a day was not a matter of simply counting the number of resources who were listed as working per day. The CSO was the primary job responsibility of the B-level techs, but not so for the A-level techs. The A-level techs are regularly assigned the CSO orders as fill work if they do not have many customer service orders to complete, or are assigned a CSO order because they are conveniently located to one. Therefore as many as 30 technicians may appear to work on a single day, but their total work level is well below this FTE. Similarly the B-level techs may only be assigned a half day's worth of CSO orders and spend the rest of their time occupied with tasks such as corrosion mitigation or they may spend half of their day in a meeting. To determine how many of each technician to schedule per day the minutes spent working on each job were totaled for each day. These total minutes were then converted into 7-hour shift equivalents to yield the number of FTE for that day.

Analysis of the B-level technician daily FTEs revealed their distribution to be normal with a slight right skew, a mean daily FTE of 6.53 and a STD of 1.97 (Figure 6). This distribution fit to the data has a square error of .016. The daily FTEs of the A-level technicians is much more variable. It is best fit by a triangular distribution with minimum value (.001), mode .597 and maximum value 18 (Figure 7).

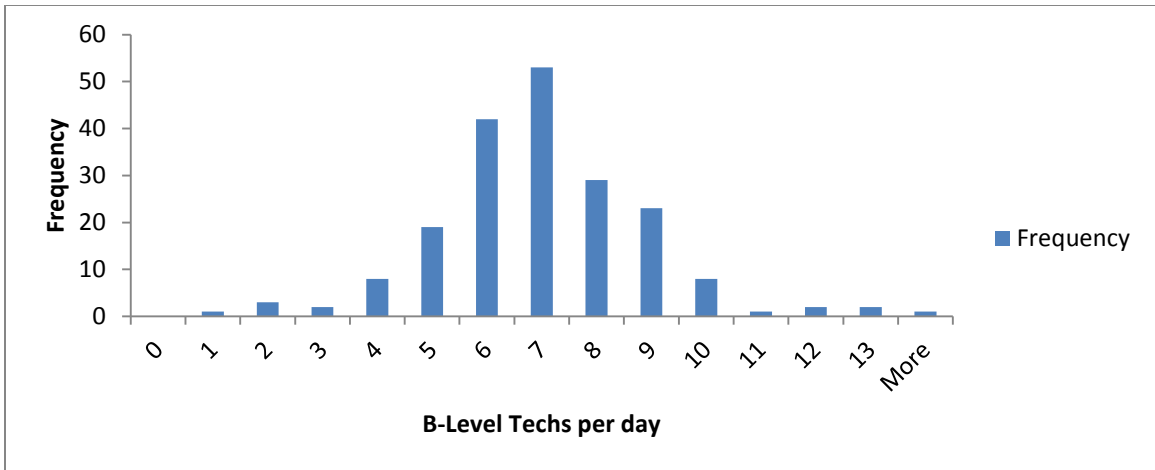


Figure 6 – Distribution of B-Level Techs per day

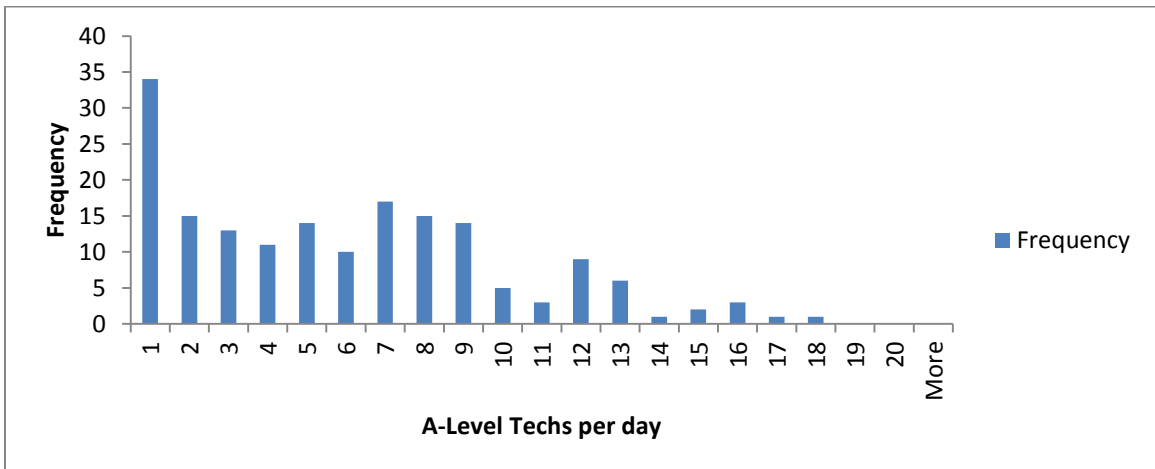


Figure 7 – Distribution of A-Level Techs per day

Like the number of AMR drivers being dependent on the day of the week, we suspected the daily FTEs of the technicians might not be entirely independent. It seemed possible that staffing levels could be affected by at least two variables – day of the week, and/or number of orders in the queue.

Like the AMR Drivers, the number of technicians scheduled by day of the week were compared using a single-factor ANOVA test. Using a P-value level of significance of .05, we cannot reject the null hypothesis in the case of either the A-level technician and the B-level

technician. There is not a significant difference in the means of FTE scheduled by day of the week. The A-level ANOVA test returns an F-value of 1.92 compared to the F-critical value 2.65, and a P-value of .13. The B-level ANOVA test returns an F-value of 1.13 compared to the same F-critical value, and a P-value of .34.

Next we determined the extent to which the number of CSO orders in the queue was correlated with the FTE scheduled for that day. The B-level staffing level is very weakly correlated to order level with a correlation coefficient of .32. The A-level staffing level has a stronger correlation with order level, but still not very strong with a correlation coefficient of .71. However, we are interested in how much of the variability in staffing of the A-level technicians can be explained by the queue size. Regression analysis on the two variables returns an R squared value $<.51$. If we inspect the line fit plot (Figure 8) we see that staffing levels are correlated well with low queue levels, but that queue level becomes increasingly unpredictable as it increases.

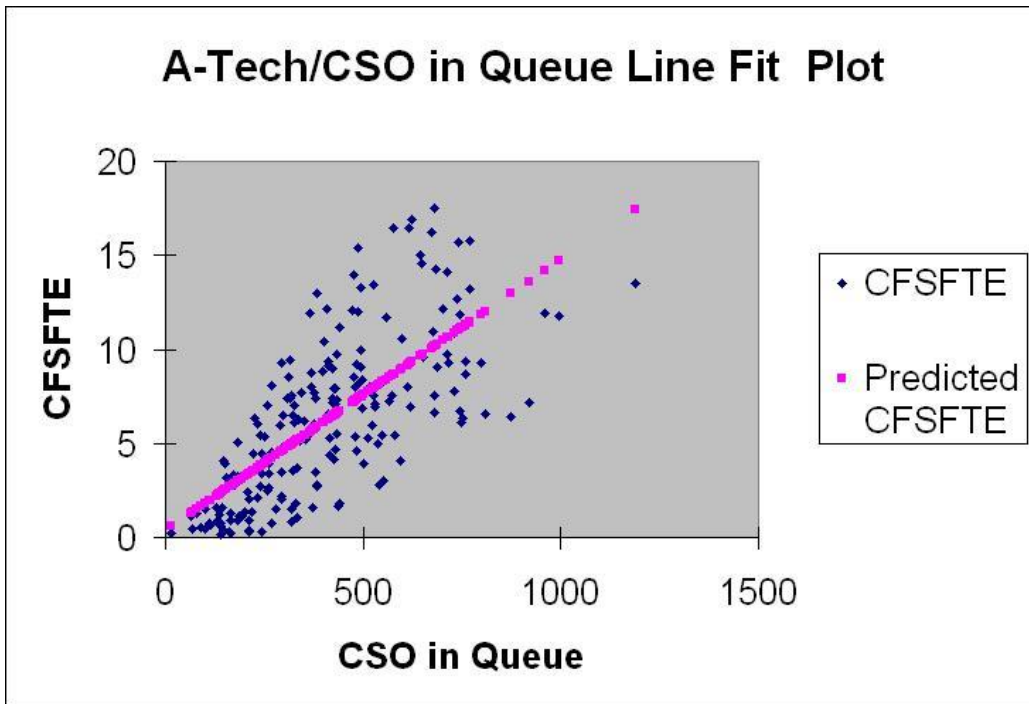


Figure 8 – Scheduled A-Level Techs per Order in Queue

We decide that tying simulation staffing levels to either day of week or queue length will not significantly improve the fidelity of the simulation to the real system. We use only the random distribution $ANINT(NORM(6.53, 1.97))$ for the B-level technician daily staffing level, and $ANINT(TRIA(-.001, .597, 18))$ for the A-level technicians daily staffing level. Both types of technicians are given the same working hours. As an example, the B-level technician schedule is given in Table 5. Shifts for this work are 9am – 4:30pm.

Day of the Week	Technicians Working
Monday	(6.53, 1.97)
Tuesday	(6.53, 1.97)
Wednesday	(6.53, 1.97)
Thursday	(6.53, 1.97)
Friday	0
Saturday	0
Sunday	0

Table 5 – B-Level Tech Schedule

4.2 Validation

We are able to use a variety of measures to validate our simulation and determine if it behaves similarly to the real system in ways that are critically important. We will compare values for: number of meters input to the system, number of CSO orders processed, maximum CSO machine queue length, average CSO machine queue length and average time spent by an order in the CSO machine queue. Because discrete event simulation has built-in variability, in order to get an accurate assessment of how the simulation is likely to behave we must look at averages of output values taken over multiple replications of the simulation. We run 30 replications of the simulation, each replication runs for 425 days with a 60-day warm-up period.

Our first question is how closely the simulation comes to capturing the inputs of the system. Does it successfully generate an accurate number of meters per cluster? The 95% confidence

intervals for the averages and range of returned values of total number of meters generated by each cluster are averaged over the 30 replications and displayed in Figure 9 below. The average error for all individual clusters is 2.6%. The total average number of meters generated by the simulation is $8.18E+06$ while the true value is $8.35E+06$, an error of 2%.

The most important metrics for our purposes concern the CSO machine queue. The first metric we will consider is the number of CSO orders processed per cluster for a simulation year. The data for the 30 replications are displayed in Figure 10. The simulation averages are again quite close to the true values. The average individual cluster error is 3.5%. The error of the simulated total CSO orders has an error of only 2.8% of the actual total.

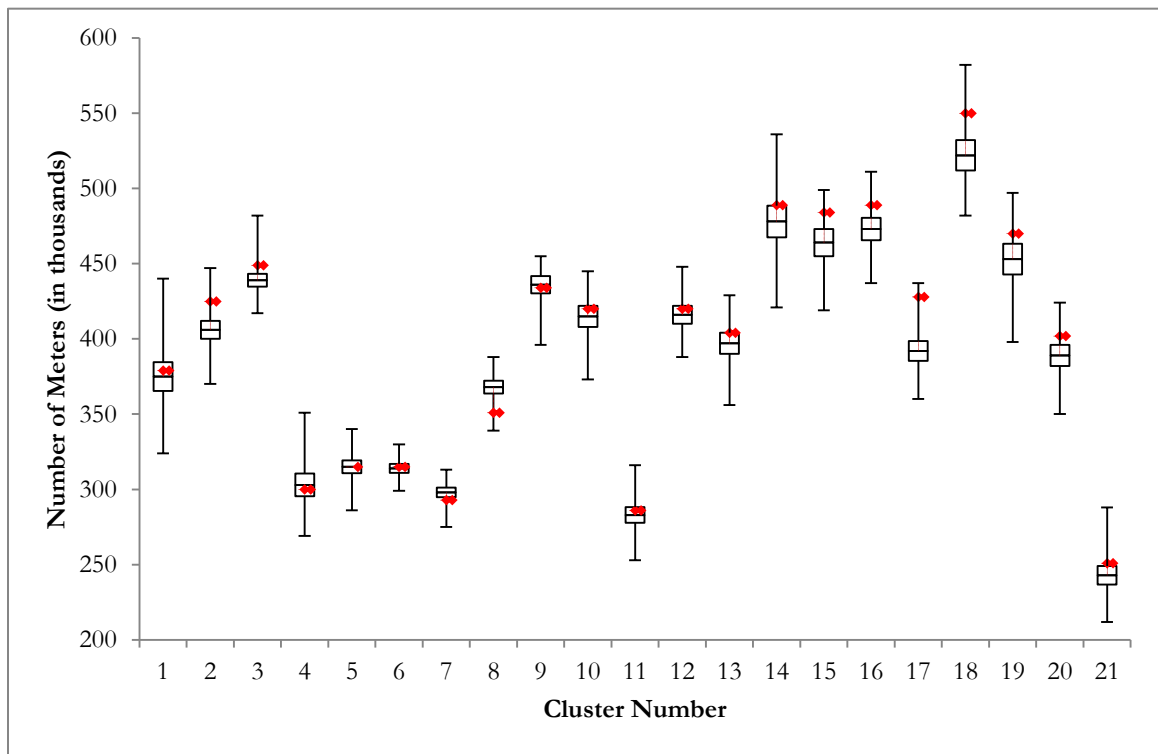


Figure 9 – Simulation Meters by Cluster, Data shown: simulated average, 95% confidence interval and range. True cluster monthly average shown in red.

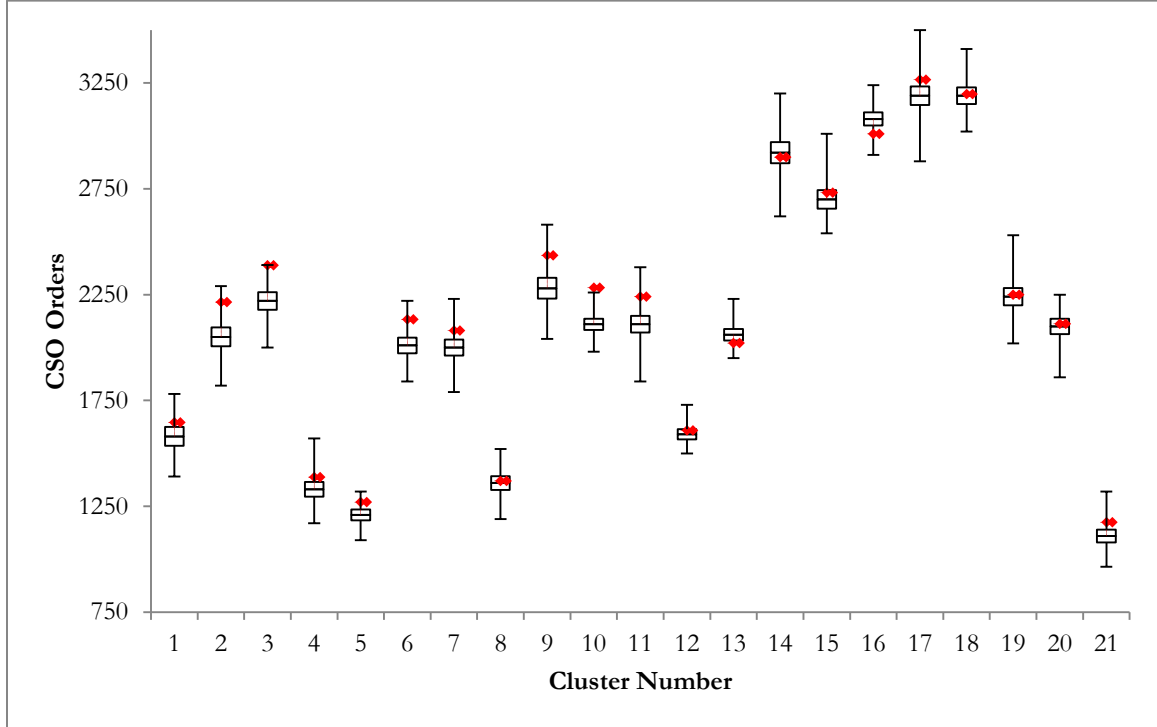


Figure 10 – CSO Orders by Cluster, Data similar to Figure 9

The final three metrics we will examine are CSO machine queue related. First we will look at how well the simulation estimates the average length of the CSO machine queue. The output metric for the simulation is extremely precise. It tallies the queue length data for every moment of the simulation run. Unfortunately the actual data for this number are imprecise. We had to estimate the value based on only two pieces of information – the number of orders released to the machine on a day, and the number of orders processed by the machine on a day. We make the simplifying assumption that all orders released to the machine are released at the beginning of the shift. Therefore, the queue reaches its maximum level at the beginning of the shift and its lowest level at the end of the shift, after all orders for that day have been processed. The formula for the maximum number of orders in the queue on day T is $(k + \sum_{y=0}^T R_y - P_{y-1})$, where k is the number of orders in the queue prior to the first

processing day $y=0$, R_y is the number of orders released on day y and P_y is the number of orders processed on day y .

Another simplifying assumption we make is a uniform rate of processing throughout a single shift. This allows us to assume the midpoint value between the maximum and minimum queue length for a day as the average of the total eight hour shift. Next we weight the minimum value of the queue for all idle time between working shifts. Many of these idle periods are overnight until the next day's shift, but many are over the weekend. Once all queue length values are properly weighted by hours, we sum them and divide by 8640 hours, the total number of hours we have data for. The result is an estimated average CSO machine queue length of 172 orders.

We measure the Average CSO Machine queue length during each replication over 30 replications of the simulation. The average of these averages is 146 orders with a half-width of 16 orders. The maximum average queue length observed for a single replication was 295 orders. Our simulated average is 15% different from our estimated actual average. The estimated actual average is 5% greater than the upper bound of the 95% confidence interval (130, 162) however, the estimated actual average is well within the simulated range 91.5-295. While a 5%-15% error is hardly ideal, it is also not tremendously far off the mark. We feel comfortable concluding that our simulated average is at least on the same scale as the estimated actual average queue length that it provides a meaningful analogue to the true system.

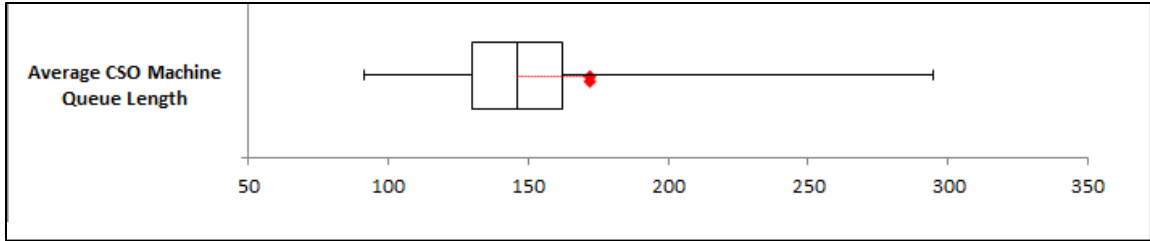


Figure 11 – Average CSO Machine Queue Length Values given in number of orders. True average shown in red.

Next we investigate the maximum queue length achieved during simulation. The data do not contain the true maximum number of orders waiting to be processed. Our estimation of the maximum queue length actually attained, as outlined above, is 1187. Figure 12 shows the CI and range for these outputs. The average maximum queue length achieved for all 30 replications 967. The confidence interval (897, 1037) does not quite capture the actual estimated maximum value 1187. From the upper bound of the confidence interval to the simulated mean, this is an error of 13%-19%.

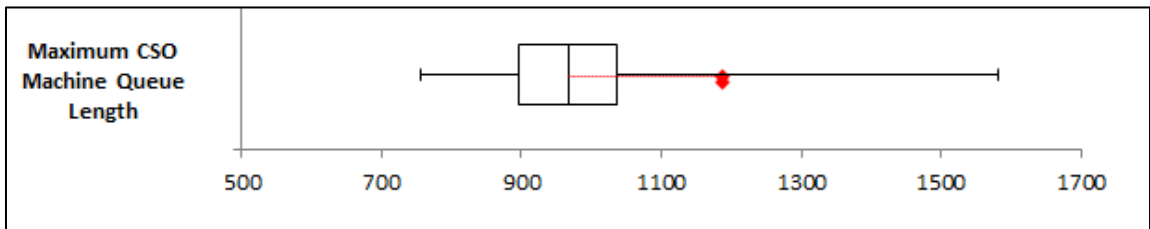


Figure 12 – Average Maximum CSO Machine Queue Length Values given in number of orders. True maximum shown in red.

The final comparison metric we use is the time an order spent in the CSO machine queue. Like the queue length, we recorded the average time each order spent in the queue for each replication and computed a 95% confidence interval for those replications (Figure 13). The average time spent in the queue over all replications is 1.21 days. Because we include travel time as part of the simulation process time, the actual time spent in-queue was estimated as an order's release date to the En Route time. The estimated average for actual time spent in

queue is 1.34 days. In this case, the actual estimated average is exactly the upper bound value of the confidence interval (1.08, 1.34).

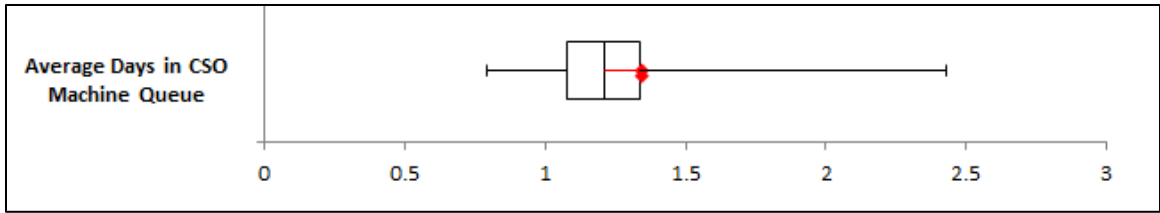


Figure 13 – Average Days in CSO Machine Queue Values given in days. True average shown in red.

The simulation processes CSO orders more efficiently than the real system. Since the overall inputs and outputs are very similar we can assume that this efficiency is gained on a per-day basis. The average queue length is shorter because the queue will frequently drop to 0 in the simulation. In reality, the queue only dropped to 0 on the last day of the year. There are likely real-world activities occupying the time of technicians that are unaccounted for in the simulation. We could likely achieve more accurate processing metrics by decreasing the number of resources available or slightly increasing the order processing time. However the goal is not perfect fidelity to the true system. We feel the performance of the simulation is close enough to the real system to provide meaningful comparison with alternative experimental scenarios.

5 EXPERIMENTAL RESULTS

5.1 Order Batch Size

Our first step is to investigate what effect the differences in cluster means have on downstream workloads. Each cluster is a batch of jobs. As has been said, our goal is not to reduce workloads, but to level them. Therefore we begin our experiments by redistributing the meters across \mathcal{K} such that $K_1 \approx K_2 \approx \dots \approx K_{21}$. To do this we give each cluster the same probability distribution to describe the number of meters created at each arrival event. Our starting point is the distribution describing the number of meters created for all workdays, independent of cluster. This distribution is normal, with a mean of $3.32E+004$ and a standard deviation of 8000.

In addition to redistributing the meters within the clusters, we assume that, should such a realignment actually occur, the routes themselves would be grouped into clusters such that they are likely to yield equivalent numbers of CSO orders. Therefore we also adjust the probability of each cluster generating a CSO order to be the system average of .54%. This simulation is named STD8000.

Now we test the effect of reducing only the standard deviation of the number of meters generated at each arrival event. We hold the mean as well as the likelihood of becoming a CSO order at the aforementioned levels. We investigate scenarios with both a 4000 meter standard deviation and a 2000 meter standard deviation. The average standard deviation of monthly meter inputs for all clusters is 4630 meters and only one cluster has a monthly standard deviation below 3000. It is very unlikely then that the system as a whole could be designed such that each cluster could have a monthly standard deviation as low as 2000

meters. However, we simulate at this level in order to project the potential savings an additional 50% reduction in input variability might have.

Results for the various experiments are presented in Figures 14-16 to show how each simulation performs against each other by our chosen comparison metrics. The results are quite surprising. As we would expect, realigning the clusters to have the same average number of meters and standard deviation shows improved performance at the STD8000 level. The average Days in CSO Queue, Average Queue Length and Maximum Queue Length each drop by 9%, 6% and 8% respectively. However, as we can see in Figure 14, we cannot say that these changes are even statistically significant because the confidence intervals for each highly overlap. Further, any gains that may exist at the STD8000 level begin to erode as the standard deviation is reduced. At the STD2000 level the results are indistinguishable from the Base simulation. For complete results see Table 13 in Appendix A.

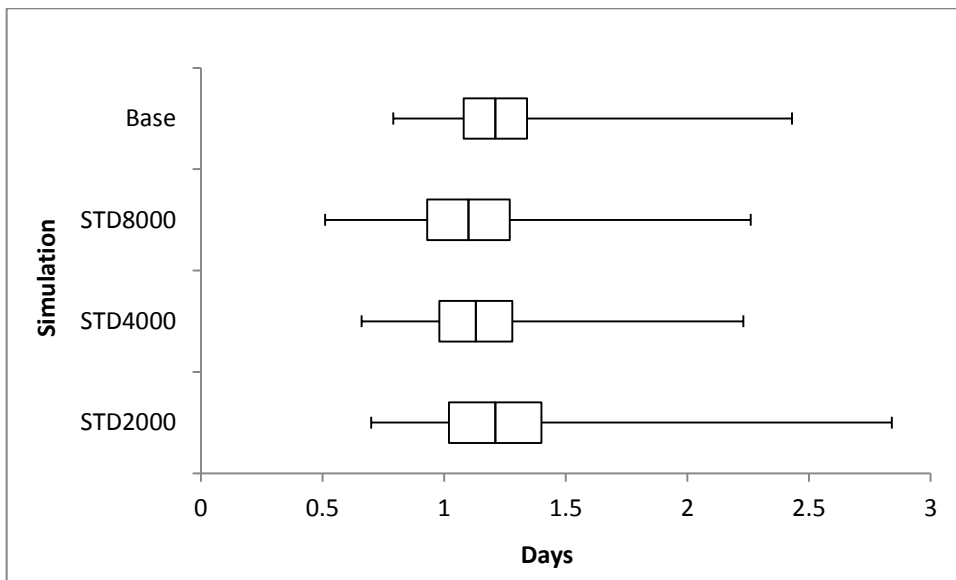


Figure 14 – Average Time in CSO Machine Queue

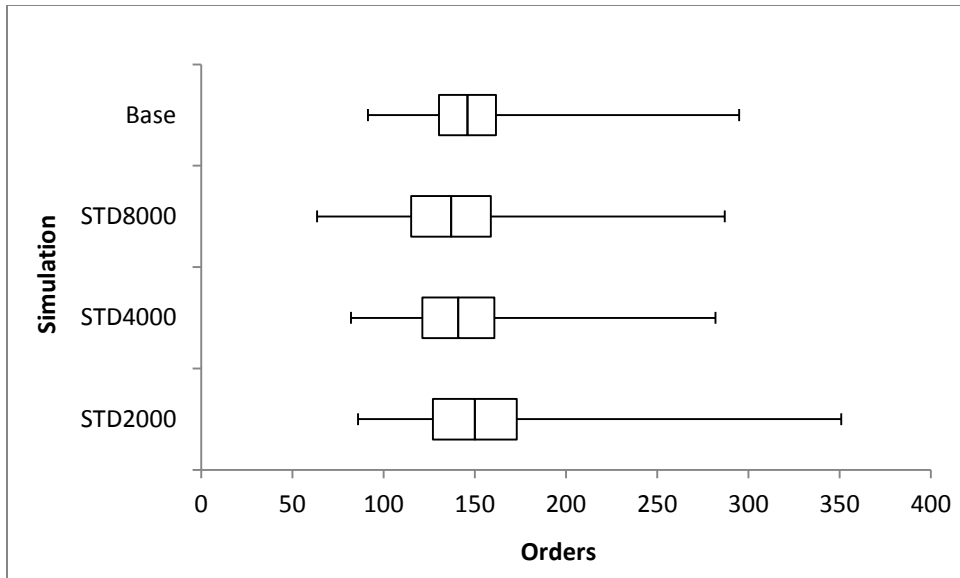


Figure 15 – Average CSO Machine Queue Length

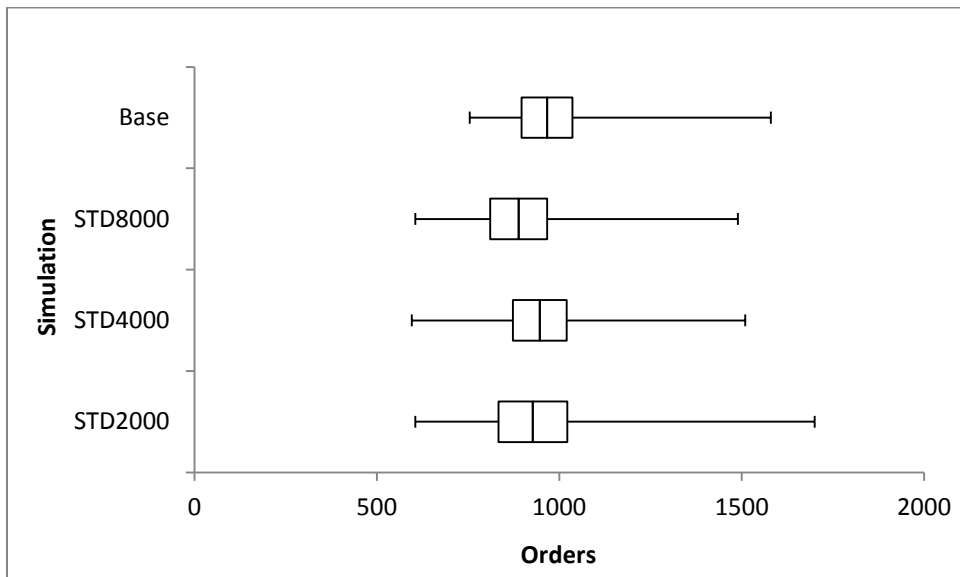


Figure 16 – Average Maximum CSO Machine Queue Length Attained

5.2 Order Review/Release

Having investigated the potential effects of reducing input variability through batch sizes, we turn to the possibility of reducing queue length variability through order release mechanisms. In the current system, an asymmetry between AMR workdays (five per week) and CSO workdays (four per week) forces order which are set to be released on Friday to be held until

Monday. This essentially doubles the queue length for Mondays relative to other workdays. We will examine two possibilities for mitigating this problem.

The first tactic is a simple order review/release mechanism. All meters must be read every 28 – 34 days. We can think of all the meters which are in the 28 – 34 day window since their last read as sitting in an order pool waiting to be released to the processing shop. The release mechanism is reading the meter. As we saw in Table 1, if a meter is read on a Wednesday or Thursday, then it will enter the CSO machine queue on Monday.

One way to level inputs to the CSO machine would be to release fewer orders to the system on the days which result in a CSO order release. That is, read fewer meters on Wednesdays and Thursdays. To do this we simply schedule fewer AMR drivers on these two days. We will cut the number of FTE roughly in half. Therefore half the number of meters will be read on these days. However, each day of the week an additional cluster of meters becomes available to be processed. If fewer of the meters are entering the system on Wednesday and Thursday, there will be many more waiting in the available job pool. So that this pool does not become backlogged, the work will have to be caught up on other days. If we remove on average 7 total drivers across the two days, we must make up for the lost work by shifting those drivers to other days. We also simplify the distributions so the values are integers and make the standard deviations slightly closer to the mean. After some experimentation we find that to even get the average number of releases per day in the same neighborhood, we must make the distributions quite disparate. The final distributions used are presented in Table 6.

Even at these distributions, the number of CSO orders released per day are likely different on average. The number of orders released Mondays and Tuesdays remains, on average, higher than on Wednesdays and Thursdays. Table 7 displays results for orders released by release day. The averages presented are yearly averages for each day, taken over 30 replications of the Base simulation input and resource levels.

Day of the Week	Technicians Working
Monday	(11, 1)
Tuesday	(11, 1)
Wednesday	(3, 1)
Thursday	(3, 1)
Friday	(11, 1)
Saturday	0
Sunday	0

Table 6 – AMR Driver Schedule for ORR The number of technicians working is described by a normal distribution. The first number in the pair is the mean and the second is the standard

Release Day	Average	STD	Half-Width	Minimum	Maximum
Monday	12600	561	209	11300	13600
Tuesday	12800	383	143	11500	13400
Wednesday	10900	298	111	10200	11400
Thursday	10400	397	148	9300	11200

Table 7 – CSO Orders Released by Day of Week

The number of orders released on Mondays and Tuesdays for a year are each approximately 2000 orders greater than both Wednesdays and Thursdays. Over a 52 week period, this is approximately 40 more orders per day, per week on these two days than the other two workdays. These outputs remain consistent when the meter create modules are given the same means at both the 8000 and the 4000 STD level.

While the number of orders released by workday are not perfectly uniform, these results are much more similar to each other than the real system. We do not pursue the possibility of making them more uniform because even at these levels, the outputs of interest do not

indicate an improvement. In fact, many of the measurements of interest may have worsened. Comparison results for what we call the ORR experiments are presented in Figures 17-19.

Utilizations of both technician types remained consistent across all simulations. With the exception of the STD4000 case, all metrics regarding CSO queue moved insignificantly upward. The only metrics to show signs of potential improvement are the measures of variability in the Maximum Queue length likely to be attained in the STD4000 case. The standard deviation decreased by 10% and the half-width decreased by 20%. While this suggests the actual value will be closer to the mean, the results indicate that the maximum queue length likely to be attained may potentially increase. These experiments suggest that load balancing through order release is likely to offset the gains of reducing batch-size variability. For complete results see Table 14 in Appendix A.

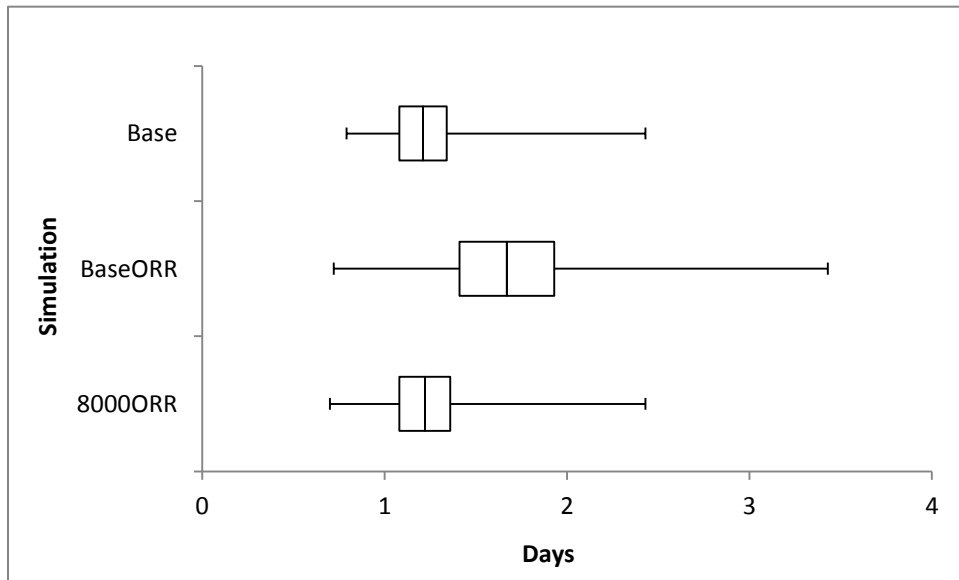


Figure 17 – Average Time in CSO Machine Queue, shows confidence intervals for the Base simulation, the Base simulation with the ORR modification and the STD8000 simulation with the ORR modification

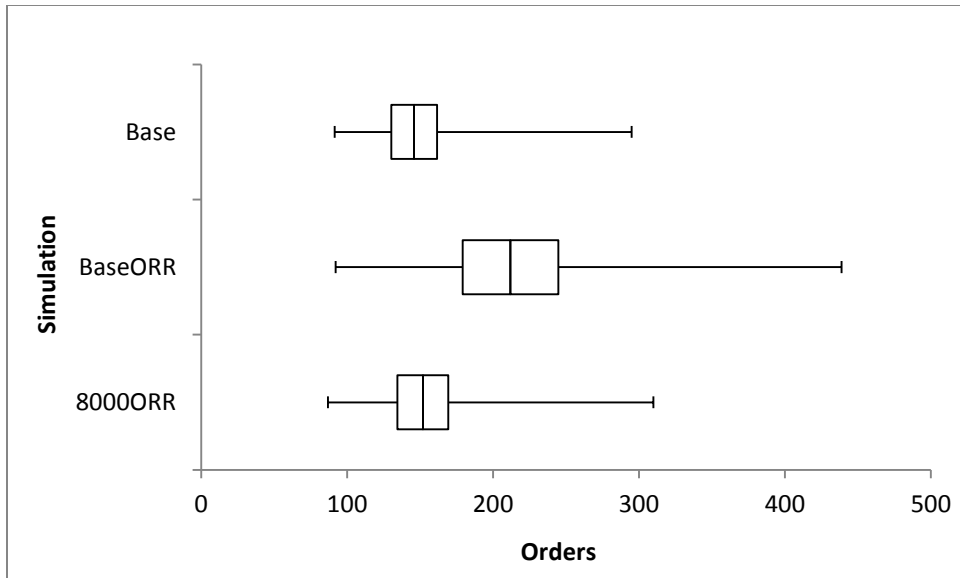


Figure 18 – Average CSO Machine Queue Length shows confidence intervals for the Base simulation, the Base simulation with the ORR modification and the STD8000 simulation with the ORR modification

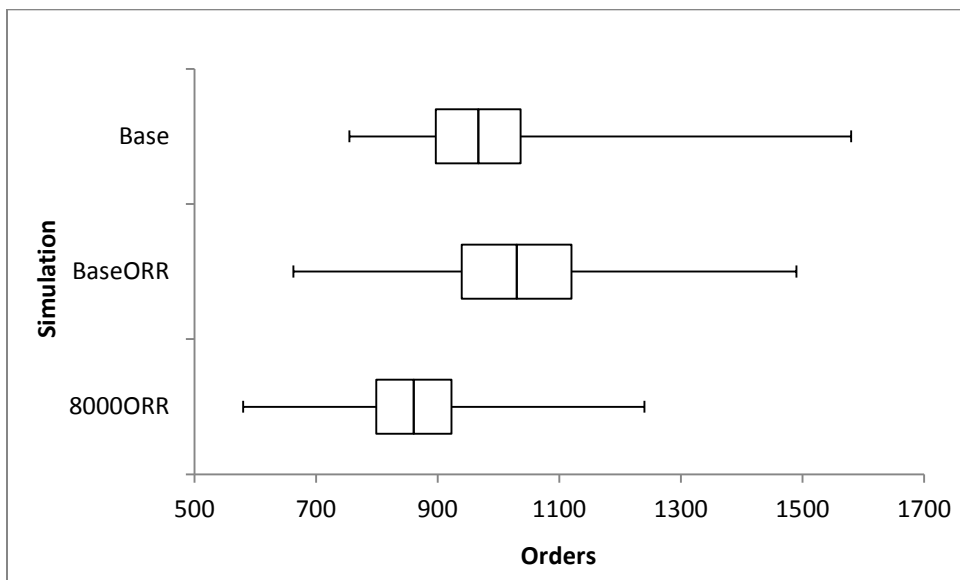


Figure 19 – Average Maximum CSO Queue shows confidence intervals for the Base simulation, the Base simulation with the ORR modification and the STD8000 simulation with the ORR modification

5.3 Targeted Release through Wait-times

The second tactic we will investigate is changing the wait-time between operations in order to target order release with operation workdays. This is a holistic approach combining

elements of both batch-sizing and ORR investigated above which we will refer to as Targeted-Release (TR).

As currently timed, a CSO order is scheduled to enter the CSO processing machine on its 45th day in the cycle. This timing works well for reads completed on a Monday or Tuesday. Each of their operations occur at the time they are scheduled to; the bill is sent on the second day in system, the late notice on the 24th and the order arrives at the CSO machine on the 45th.

However, because of weekends and the asymmetry between AMR days and CSO days meters read on Wednesday and Thursday do not reach the CSO machine until their 49th and 48th day in the system, respectively. Meters read on Thursday receive a full lead time to fulfill the CSO order. However, CSO orders have a due date 21 days from the day that the late notice is sent. Meters read on Wednesdays have arrive at and are completely processed by the Late Notice machine on a Friday. This due date does not change regardless of the fact that they must wait three days to be processed by the CSO machine. Therefore, the lead time for CSO orders originated on a Wednesday is effectively reduced by three days.

Dealing with this problem is a simple matter of scheduling operations at seven-day intervals. Rather than a bill being due on the 17th day in the cycle, the bill will be due on the 22nd day in the cycle. That is, if a meter is read on a Monday, the bill for that meter will be due three Mondays later. All major operations could be realigned on seven day intervals except the CSO operation. Following the pattern described above, if the CSO operation was scheduled for the 43rd or 50th day in the cycle, meters read on Fridays which become CSO orders will

be released to the CSO machine on a Friday. These orders will still have to be held until Monday and they will still have a shorter lead time.

Therefore we will schedule the CSO operation for either the 42nd or 49th day in the cycle for all meters read Tuesday – Friday and on the 43rd or 50th day in the schedule for meters read on Mondays. As the real life system under investigation has a customer service component to its considerations, we will use the longer of the two options. The final operations schedule is presented in Table 8.

While this schedule assures that operations will occur at nearly the same intervals across the calendar, it does not address the problem of stacking the release of two clusters worth of CSO orders on the same day. Under this scenario two clusters will still reach the CSO machine on Monday while one will reach it all other days. The asymmetry in number of workdays between the first and final machines will always produce this problem.

Operation	Day in System
AMR	1
Bill	2
Bill Due	22
Late Notice	29
Tue-Fri AMRs to CSO Queue	49
Mon AMRs to CSO Queue	50
Due Day	59

Table 8 – Workday Targeted Operation Schedule

This leads us to try again to balance the number of orders reaching the CSO machine across all workdays. As we have seen, restricting the number of meters read does not improve system performance. However, we have seen that realigning clusters by the number of meters per cluster and the likelihood that those meters will become CSO orders can improve system performance. Therefore one tactic might be, instead of making all clusters more

similar as we did in our first set of experiments, to make some clusters less likely to lead to CSO orders than others. This might have the effect that when two of these reduced-CSO-rate clusters land on Mondays, they will have a total number of CSO orders arriving that is roughly equivalent to the number arriving on all other workdays.

However, under the current model of 21 clusters being loaded consecutively over all workdays, there is no guarantee that these reduced-CSO-rate clusters will be released to the system on a Monday or a Tuesday, the days which will lead to a Monday CSO machine release. The TR scenario will investigate, then, a method of scheduling clusters to be released to the system on specific days of the week, rather than releasing strictly by cluster number sequence. In addition to its cluster number, each cluster will be assigned a release day of the week. For instance Cluster 1 might be assigned to Mondays and Cluster 2 would be assigned to Tuesdays. Further, Cluster 6 would be a Monday cluster and Cluster 7 would be a Tuesday cluster. If Monday clusters and Tuesday clusters typically resulted in 50% fewer CSO orders than the clusters assigned to other days of the week, then we should expect to see a balanced number of CSO orders released per day.

Following the cluster assignment procedure outlined above, Cluster 1 assigned to Monday, etc., we quickly see a problem. With 21 clusters, Cluster 21 would have to either be scheduled on the same day as Cluster 1 or there would have to be five weeks between Monday cluster releases while there would be four weeks between all other cluster releases. Neither of these options improves our solution, either some clusters remain doubled, or they are perpetually out of sync. To cope with this problem we assume a reorganization from 21 clusters to 20 clusters. We further assume it is possible to construe this organization such that eight of these clusters will produce 50% the number of CSO orders than the other 12

clusters. We admit that it may in fact be impossible, or at least very difficult, to create such a cluster organization. Whether it is possible is outside the purview of this project. Our question is whether such an order release mechanism could lead to a more predictable workload.

The system of 20 clusters cycling over the course of 20 workdays could actually work like clockwork. Following this arrangement though would lead a customer to have 13 meter reads, and therefore receive 13 bills, over the course of a calendar year. While a calendar of 13 months of four weeks apiece might be entirely sensible, reality and regulation being what they are we must again modify the TR proposal. We will schedule clusters not only to specific days of the week, but to specific days of the week of the month. Cluster 1 would be scheduled for the first Monday of each month. Cluster 6 would be scheduled for the second Monday of each month. This will ensure that each meter is read once and only once per month as state regulation mandates.

There will typically be 28 days between a cluster's releases into the system. But since months have an annoying fickleness regarding which day of the week they begin on, this proposal will also guarantee that there will occasionally be 35 days between reads for individual clusters. On a month that begins on a Tuesday, for instance, the first cluster to be released to the system will be Cluster 2, not Cluster 1. In fact, Cluster 1 will be the sixth order released to the system that month. Over the course of a year this will mean that there will be an average of 30 days between meter readings. But from month to month there will be strictly either 28 or 35 days between meter reads. While it may seem like this is an increase in the amount of variability, it is actually a decrease. The current system operation leads to a period of 28-33 days between reads. This requires three bits of information to describe. The TR

proposal requires only one bit to describe the time between meter reads. So while the range has increased slightly, the variability has actually decreased.

To simulate the TR scenario, we had to eliminate one cluster and redistribute its meters to the remaining 20. Without much more detailed analysis than is possible in this project, it was not possible to do this in such a way that would reflect the real organization of the current clusters. We entirely abandoned all real cluster information and instead created 20 clusters with a monthly mean of 35000 meters and a standard deviation of 8000 meters. The best direct comparison is to the BaseSTD8000 simulation which had monthly means of 34000 meters for the 21 clusters.

In order to schedule the arrivals according to the day-of-the-week schedule we outlined above, it was necessary to schedule the meter arrivals via Arena's Arrival Schedule option. This is very similar to the method used to schedule the resources, the user specifies the number of arrivals and the duration of those arrivals. It is a very fine point but it is important to stress that the user does not specify the number of entities in the Arrival Schedule. The user schedules the average number of arrival events which take place over some duration. Arena then uses that user specified average as the mean of a random distribution which it schedules over a time duration. This leads to some variability in arrival events even if the number specified by the user is constant. For instance, if the number of arrival events is an average of one per hour, there will be some hours in which no events occur and some hours when three or more events occur due to the random distribution Arena uses. However, given the large number of entities in our simulation as well as our own specified variability, the variability due to Arena is minimal in comparison.

A detailed schedule for Cluster 1 is presented in Table 18 in appendix A. The rule for this cluster is that its meters are read on the first Monday of every month. There is one entity for each arrival. In order to give the simulation a reasonable warm-up period, the schedule begins November 1, 2010. For the first seven hours of the simulation, there are 0 entity arrivals. In the eighth hour there is an arrival event. For the next 839 hours there are no arrival events. The 839 is the number of hours in a five week period, and 671 is the number of hours in a four week period. These numbers tell us the interval between the specified day of the week of a cluster. For clarity, the scheduled read dates for the meters in Cluster 1 are shown in Table 9.

Cluster 1 Scheduled Read Dates
November 4, 2010
December 6, 2010
January 3, 2011
February 7, 2011
March 7, 2011
April 4, 2011
May 2, 2011
June 6, 2011
July 4, 2011
August 1, 2011
September 5, 2011
October 3, 2011
November 7, 2011
December 5, 2011

Table 9 – Cluster 1 Read Dates

One of the goals of the TR scenario is a consistency of inputs. We modify the AMR driver schedule so that, unlike the true system, each day will have the same mean number of AMR drivers. We do not eliminate all variability however since the number working would continue to be subject to illness, vacation and maintenance meetings. We assign all days the

distribution derived from Company data describing the average number of AMR drivers independent of day-of-week, $ANINT(NORM(7.21, 1.2))$.

Under the TR scenario, we assume that clusters scheduled for Mondays and Tuesdays are 50% less likely to lead to CSO orders. The probability for these clusters is .35% while the probability for all other clusters is set to .7%. The final change we make for the TR simulation is to remove the billing governor. Should the governor be reached, it would prevent some meters from moving on and shift some CSO orders off of their target release date. Again, while this may not be entirely realistic, our goal is investigation of the potential effectiveness of Targeted Release.

We run simulations with the base configuration described above, TRBase, as well as a simulation with meter per cluster variability reduced to std 4000, TRSTD4000. Figures 20-22 display results of these as well as results of the Base STD8000 and STD4000 simulations. We can see that if anything the TR scenario slightly worsens system performance across all measures, though not significantly. All measures were up slightly under the TR scenario from their base counterparts. But only the increase in the Days in CSO queue is significant. The confidence intervals shifted from STD8000 (.64, .77) and STD4000 (.59, .7) to TRBase (.83, 1.08) and TRSTD4000 (.82, .95) respectively.

As constructed, the TR scenario does little to improve system performance and, if anything, may degrade it. Again we can see that a reduction in input variability has no effect on the downstream workloads as the outcomes for TRBase and TRSTD4000 are equivalent. However, TR offers at least one additional advantage over previously discussed schemes. Because the inputs are now tied to a specific subset of days in the month, and there are

regularly gaps in input days due to the irregularity of number of days in a month, and the timing of downstream release of orders to the CSO machine is tied directly to the input day, TR creates numerous days of predictably lower workloads.

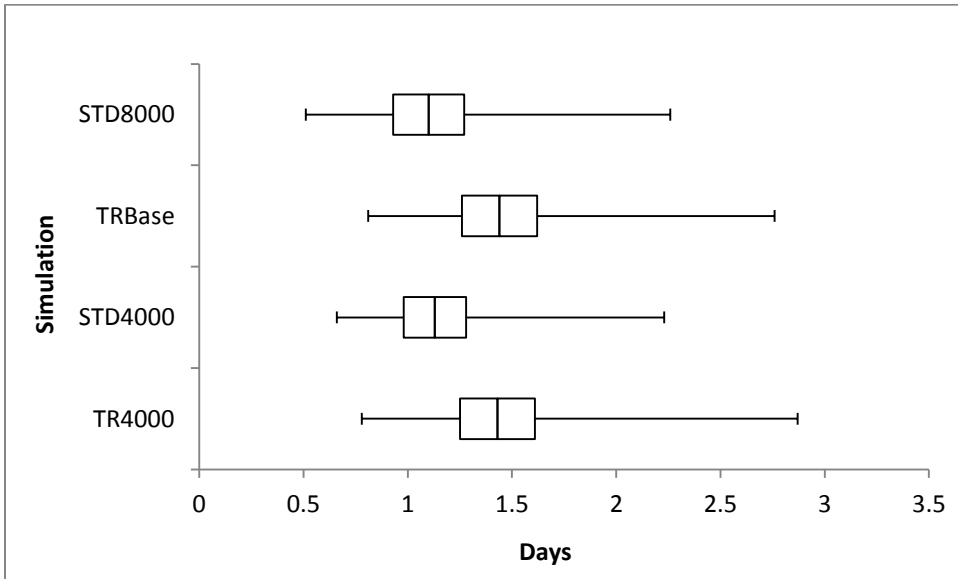


Figure 20 – TR Scenario Comparisons Time in CSO Machine Queue

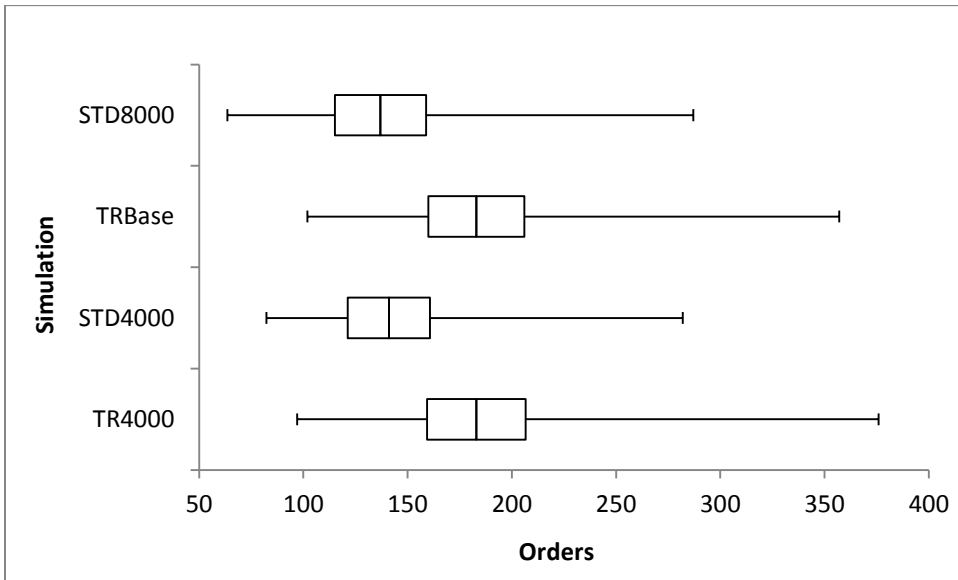


Figure 21 – TR Comparisons Average CSO Machine Queue Length

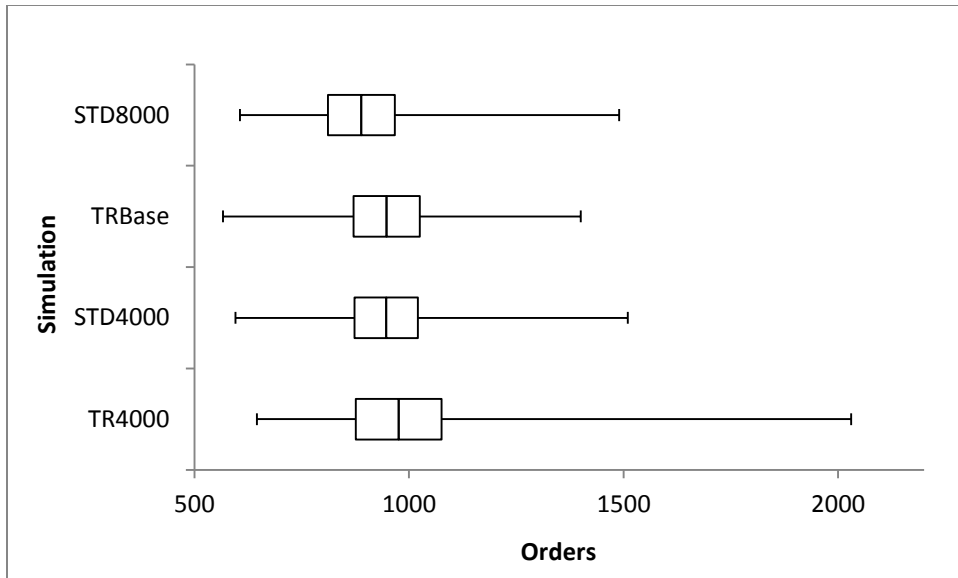


Figure 22 – TR Comparisons Maximum CSO Machine Queue Length

The advantage of reduced workload days presents itself in a few different ways. First, since the AMR machine is run only on the first 20 weekdays of any month, any additional days of a week within the same month would not require any AMR resources. For example, September 30, 2013 is the 21st weekday of that month. Under TR for the Company, there would be no AMR work on this day. Another way to think of this is that it is the fifth Monday of that month. Any fifth day of the week of any month would not have any AMR work scheduled.

Second, whenever these gaps in input occur, there will be no CSO orders released 49 days later (50 days in the case of a Monday). Therefore these days would also have a reduced workload, and therefore a reduced demand for CSO work resources. If the Company was able to get into a routine of accomplishing CSO orders on the day they were released to the machine (a routine it does not find itself in currently) then the demand for resources on these days would fall to zero.

Finally, the company currently schedules more AMR resources on Fridays than other days of the week in order to get ahead on the AMR orders. Since a goal of TR is input consistency, and the number of resources scheduled for a Friday is on average no different than other days, TR frees some resource hours on Fridays that could be allocated in other ways.

These three aspects of the TR arrangement mean that the Company gets back many resource hours in reduced demand. This time could be used to catch up on other work-types, or as 'Maintenance time.' As was stated in Chapter 1, the Company is required to hold numerous training and informational meetings throughout the year. While an attempt is made to schedule these meetings on low-volume workload days, it is difficult to do so. Inevitably resources must be pulled out of the field on days in which work will be pushed to later in the day. The TR arrangement provides known and entirely predictable periods of reduced workload. These days could easily be utilized for meetings.

In 2011 there were 20 fifth weekdays. This means 20 days when there would be no AMR demand, and roughly the same number of days with no, or reduced CSO demand. If we count these days by the average FTE they demand, that comes to ~1760 work hours. In 2012 there were 954 total required meeting hours for all B-level technicians. This means that there would be twice as many freed hours per year than would be necessary for meeting hours. The Company could therefore allocate many hours to holiday time by increasing the number of day-off slots available on these days. This would focus holiday time to days which are guaranteed to have a lower impact on work accomplished.

Over the course of a year, these 1760 hours are an average of 34 additional work hours per week, or around 5 FTE. Since the total number of employees is a static quantity, the

additional hours cannot affect the average number of employees available on workdays. But what these hours would do is reduce the amount of variability in resource availability. Due to the difficulty of projecting how such additional hours would impact variability, we choose to reduce the STD of resource availability by half. This may sound somewhat drastic, but it is not without basis. The current schedules for both AMR drivers and B-level technicians is roughly normal with mean 7 FTE and a STD of 2 FTE. Therefore in only 2.15% of days will the FTE be between 1 – 3 and even in these cases the number of FTE is likely to be closer to 3 than lower. This is a difference of just over 4 FTE from the mean of 7. This difference is covered by the 5 additional FTE the company gets back from TR. Over 260 workdays the Company is likely to have fewer than 3 FTE on only 6 days. These six days seem very likely to be covered by the 35 additional available work-hours per week. We can therefore assume that three (3) FTE is essentially a minimum that is extremely unlikely to be exceeded. A normal distribution with mean 7 and STD 1 basically accomplishes this scenario. Therefore, we set the FTE schedules for AMR and B-Level technicians to follow the curve $ANINT(NORM(7, 1))$ for those days they are scheduled to work.

Some preliminary runs of the TR simulation at these resource levels showed much improved productivity and very low utilization rates for both the B- and A-level technicians. One major goal of this project is to decrease the need to divert A-level technicians to the AMR-CSO workflow. Faced with the preliminary simulation results we attempted to reduce the number of A-level technicians available to the workflow. The original distribution describing the available resources is $ANINT(TRIA(-.001, .597, 18))$. The average number of A-level techs according to this distribution is ~ 6 , but the range 0-18. For simplicity sake we assume normality with a mean of 5 A-level techs and a STD of 1. Therefore we have lowered the

average number of resources by one and severely reduced the maximum likely number of resources. This distribution also all but guarantees there will be at least 2 A-level techs working CSO orders every workday. For final resource schedules for TRAdjustedResources (TRAdRes) see Tables 19 and 20 Appendix A.

Outputs for TRAdRes are shown along with TRBase for comparison in Figures 24-26 and the disparities are rather surprising. Again, the only differences between the TRBase and TRAdRes are that the resource levels for AMR and B-level technicians are less variable and the resource level for A-level technicians are less variable and *reduced*. Yet these changes result in dramatic decreases in Days in CSO Queue, Average CSO Queue Length and Maximum Queue Length. Average Days in Queue drops more than 35% from 1.44 days to .93 days with non-overlapping CI (1.26, 1.62) and (.82, 1.04) respectively. Average Queue Length also drops over 35% from 183 CSO orders in TRBase to just 118 during TRAdRes runs. Confidence intervals for these measures are also non-overlapping at (160, 206) and (103, 133). Maximum Queue Length falls 34% from 948 in TRBase to 627 in TRAdRes, with confidence intervals (871, 1025) and (576, 678).

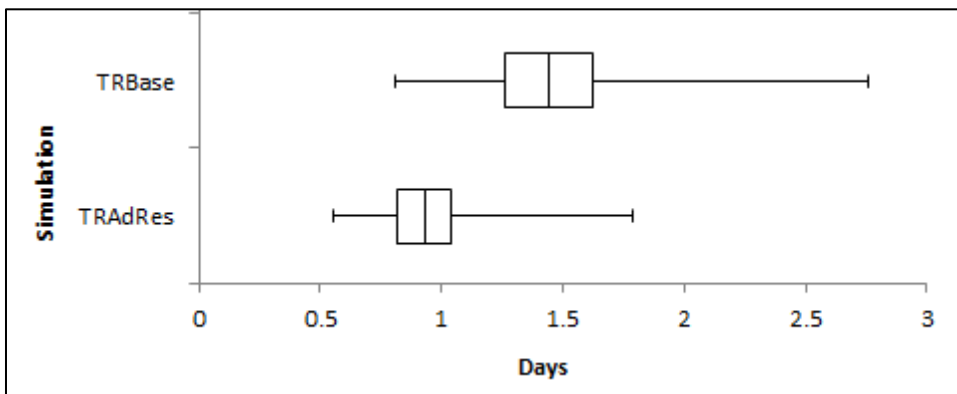


Figure 23– Average Days in CSO Machine Queue

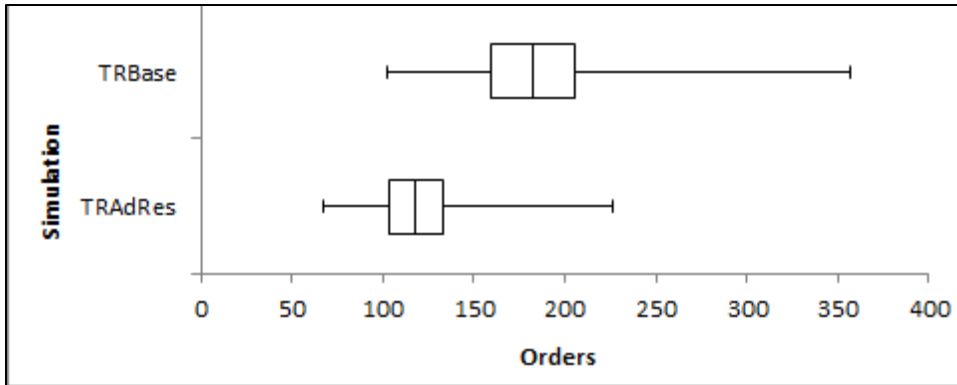


Figure 24 – Average CSO Machine Queue Length

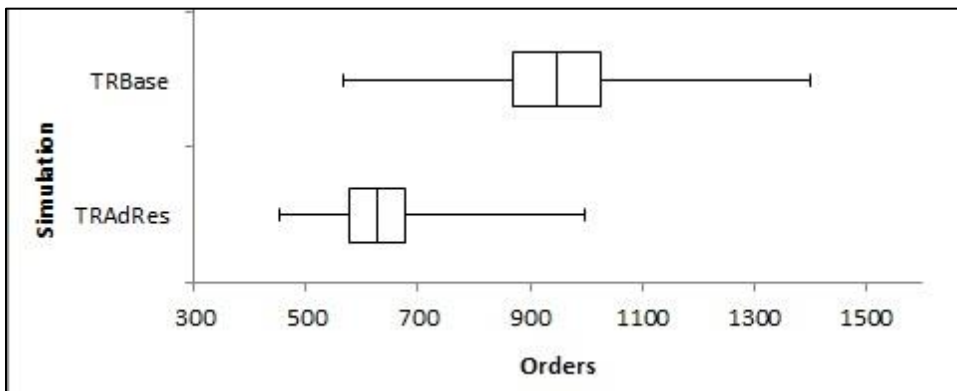


Figure 25 – Maximum CSO Machine Queue Length

In addition to the reduction in absolute values of each of these measurements is a significant reduction in variability by half-width, STD and total range of measurements. For example the maximum queue length attained in 30 replications of TRBase is 1,400 and the range of all measurements is 833. The maximum queue length attained in 30 replications of TRAdRes is 997 and the range of all measurements is just 543. The STD and Half-width for Average Days in Queue, Average Queue Length and Maximum Queue all drop by between 34% - 39% as well.

The area we actually see increases in are resource utilizations. The B-level resources have an average utilization of 1 during TRAdRes, up from .97 in TRBase. We can say this change is technically significant with confidence intervals (.99, 1.01) and (.96, .98) respectively. The increase in average utilization is larger for the A-level resources going from an average .93 in

TRBase to 1 in TRAdRes. Confidence intervals (.92, .94) and (.99, 1.01) are significantly different.

Two points about these utilization measures are important. First, utilization measures at these levels may seem unrealistic. Any manager who expected an employee to be working 100% of the time would find themselves perpetually frustrated by both employee performance and extraordinarily high turn-over rates. But we must remember that the simulation only accounts for seven hours of an eight and a half hour shift. Breaks and other downtime are excluded from our model. Further, the time required to complete operations was derived from data about actual work time. It is no surprise then that utilization rates would come very close to 100%.

Second, though the differences between the two simulations are minor, the fact that they increased is very telling. We interpret this as meaning that the resource availability levels used in the TRAdRes simulation better align with the work presented to them. The inputs to both simulations were the same. Performance measurements were worse and resource utilizations were lower in TRBase because the resource availability regularly dropped below work demand due to variability, and at other times was well above work demand.

Most encouraging is that the performance improvement in TRAdRes was achieved while committing significantly fewer A-level technicians to the work than in other simulation tests. This suggests that TR reduces the impact of the AMR-CSO workflow on other Company workflows. However, the current A-level technician commitment to AMR-CSO is based on a mix of convenience and conspicuous need. There will frequently be days in which no A-level resources are committed to the workflow, and other days when as many as 18 FTE are

committed to the workflow. In the TRAdRes scenario, the maximum and average number of resources are both significantly lower, but we assume that at least some level of A-level technician commitment every day.

This begs the question, is this arrangement possible in the real world currently? Could the Company change its resource staffing policy to something similar to what we suggest in TRAdRes and see improvement? To investigate this possibility we change both technician schedules in the Base simulation to the schedules that were used in TRAdRes. All other schedules and inputs are maintained at original settings. In reality the drastic variability in resource levels was caused by a variety of factors including mandatory meetings, holidays, health-related call outs, and workloads. We can think of this scenario as a Company initiative to hire enough resources to guarantee resources at the proposed simulation levels without making any changes to reduce these sources of variability. The results of this simulation, BaseAdRes, are displayed in Figures 26-28 below.

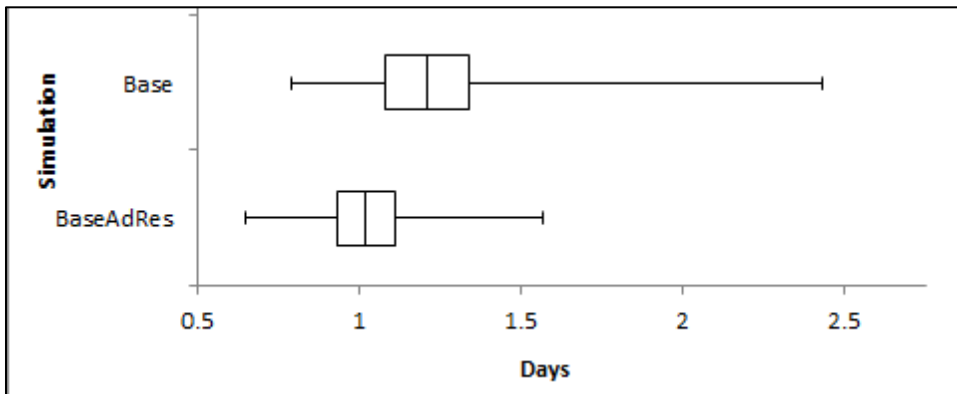


Figure 26 – Time in CSO Machine Queue

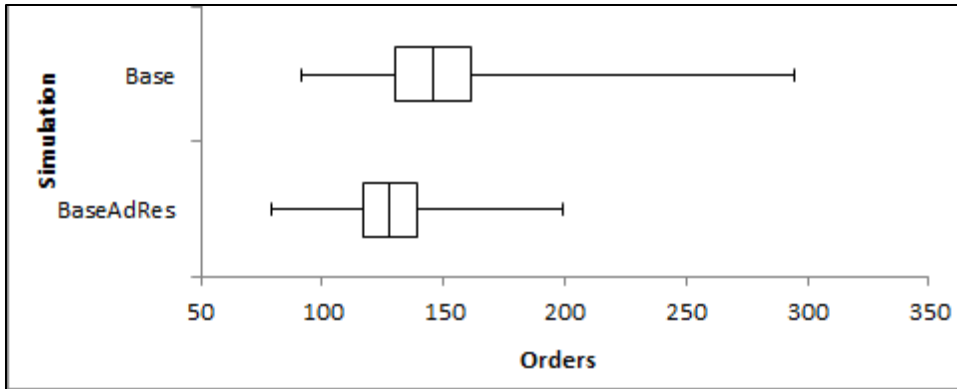


Figure 27 – Average CSO Machine Queue Length

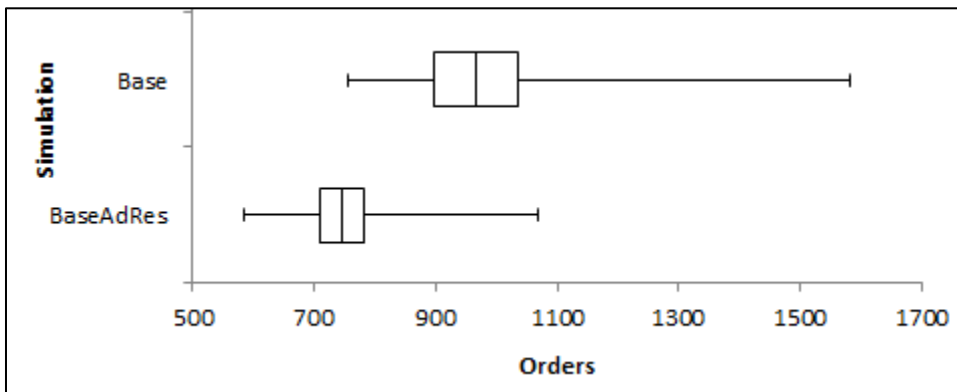


Figure 28 – Maximum CSO Machine Queue Length

As we can see the Adjusted Resource schedule improves the performance of the system. While the confidence intervals for each of the queue measurements do overlap the averages are considerably different. Average Days in Queue, Queue Length and Maximum Queue length dropped by 16%, 12% and 23% respectively. Also, the ranges for these values fell considerably because, while all the minimum values were lower, the maximum value returned in any of the 30 replications was significantly lower. The maximum values returned for each of the queue measurements fell by 35%, 33% and 32% respectively. We also see improvement in the resource utilization measurements. Again, it seems that stabilizing the workforce ensures that work demands are better matched to available resources. We can

conclude from this experiment that resource variability is a significant contributor to overall workload variability.

It would appear then that the Company could significantly improve performance by stabilizing its resource levels. But it must be remembered that the current volatility of resource levels is due to a variety of factors that are beyond the company's control. At the current overall resource capacity, the technicians must be shift to various workflows of the company in order to meet a variety of constraints. In order to achieve the stability to achieve the stability of the Adjusted Resource levels, the Company would be forced to add capacity. It is beyond the scope of this project to estimate the total number of hires necessary to accomplish this, but any addition of capacity should be avoided. Additional capacity, even if well utilized, will inevitably produce higher rates for customers. Where alternatives exist to additional capacity which can achieve similar results exist those options should be preferred.

Table 10 shows the output data of the best performing alternatives available to the company without adding capacity. Our work suggests that there appear to be gains simply by reorganizing the clusters so that they had roughly the same number of meters in them. Even at the STD level of 8000 meters every critical metric improves. The gains associated with this change though are not dramatic. The 8000 meter standard deviation seems actually unlikely. In reality, the average standard deviation for individual clusters is 4600, which is caused by seasonality, client/service migration, and economic factors. Therefore it is likely that the clusters after reorganization would actually have a lower standard deviations than 8000. However, our work does not suggest that reducing the standard deviation bears additional benefits and actually suggests that it could degrade system performance.

The best alternative for the Company is the Targeted Release scheme because of the opportunities it provides to stabilize resource variability as well as input variability. Our work suggests that has the potential to significantly reduce workload variability, increase throughput all while consuming fewer high value technicians, making them available to focus on higher value orders.

Simulation	Measurement	Days In CSO Queue	Average Queue Length	Maximum Queue Length	B-Utilization	A-Utilization
Base						
	Average	1.21	146	967	0.91	0.88
	STD	0.33	41.8	187	0.03	0.03
	Half-Width	0.13	15.6	69.7	0.01	0.01
	Minimum	0.79	91.5	755	0.86	0.84
	Maximum	2.43	295	1580	0.97	0.93
STD8000						
	Average	1.10	137	889	0.94	0.90
	STD	0.46	58.6	208	0.03	0.04
	Half-Width	0.17	21.9	77.7	0.01	0.01
	Minimum	0.51	63.6	606	0.89	0.83
	Maximum	2.26	287	1490	1	0.98
TRAdRes						
	Average	0.93	118	627	1	1
	STD	0.3	39.7	135	0.02	0.02
	Half-Width	0.11	14.8	50.6	0.01	0.01
	Minimum	0.55	67.7	454	0.97	0.97
	Maximum	1.79	226	997	1.03	1.03

Table 10 – Outputs for top performing experiments relative to Base

6 DISCUSSION

This work has explored means of reducing workload variability with an emphasis on control of input variability. Batch sizing and Order Review/Release have a long history in the Job Shop literature and are widely applicable. We also developed a tactic we call Targeted Release to target specific inputs with specific release dates. What we found is that the system is much more sensitive to variability in resource capacity than it is to input variability. The largest gains were seen under the TR scenario in which resource variability was reduced. The advantage came from predictable periods of low workload created from the TR schedule. This work supports the findings of Mehta and Uszoy [35] that insertion of idle time into a schedule can absorb random breakdown of machines without disrupting the production schedule.

We feel Targeted Release also has broader applicability, but it is critical to recognize that it is applicable to systems with endogenous control of their own inputs. The TR method for targeting release times to the machines within a job shop is possible because the Company has direct control over the timing of the inputs. In the vast majority of job shop research, orders arriving at the shop are modeled as a random variable with certain predictable parameters. The key to TR is that the orders are actually generated by the Company internally by the needs of their billing department. This is not a characteristic unique to the company under investigation or to utility companies in general. Some other areas of applicability would be paycheck printing, hospital surgery scheduling, and public transportation maintenance scheduling just to name a few.

We feel that a Targeted Release scenario has many powerful benefits in its ability to both streamline workload but most especially to create advantageous gaps in work. In the case investigated here, it created an environment which would allow a much more stable workforce than the standard operating system in place. Under these adjusted resource levels, TR dramatically outperformed all other options and achieved the goals of minimizing the impact of AMR-CSO on other Company work-flows.

The choice of using simulation to investigate potential changes to the AMR-CSO workflow proved extremely beneficial. Simulation allowed the flexibility to test a variety of assumptions about the impacts of input variability on the workflow in a short amount of time and across a variety of measures. More importantly, the use of simulation enabled the investigation of a hypothesis we had not previously considered critical to the overall investigation. Simulation provided an element of surprise. Had we pursued a closed form for say, the optimum allocation of meters to clusters, or the optimal ordering of clusters in order to reduce resource demand, we would not have had the opportunity to realize the impact of workload idle-time insertion.

However, our methodology is not without limitations. While particular detail was given to inputs, other aspects of the real system were generalized in the simulation. Only one type of work is modeled. The Company processes a wide variety of jobs. Technicians, particularly A-level technicians constantly alternate between job types. It is possible that this erodes competency. All technicians are treated as equally capable. In reality there is certainly disparity between individual abilities.

There are no emergency orders in the simulation. While the data used includes the consequences of emergency events, it averages those events across an entire year. An emergency event can cause tremendous delay in work as well as the reallocation of jobs among technicians and rerouting which would increase travel times. In the real world, all these events would happen at one time, affecting numerous technicians and jobs. They are dependent on each other. In the simulation, portions of these effects happen regularly and independently of each other.

Similar to the lack of emergency orders, there are no traffic events that may affect a large number of orders and technicians. Further, travel times are treated as independent, doled out as random variables. But they are likely to be dependent on a variety of factors such as weather, traffic, day of the week, total order level, seasonality.

Perhaps the biggest limitation of this work is that the overall design of the simulation is geared toward investigating inputs, whereas the most important finding relates to resources. While we feel the results are reasonable and suggestive of powerful potential system improvements, they should be seen as suggestive. To be able to draw concrete conclusions with respect to how inserted idle time might impact resource capacity variability, the simulation would have to be modified in a variety of ways. We would need to incorporate some specific impacts on resources and work including:

- Spatial component to work. Some technicians drive their trucks home at night, while others pick up and drop off their trucks as central hubs. Some sequences of work, like driving AMR routes on the coast, require technicians to spend nights in hotels overnight. Certain B-level technicians will always drive specific AMR routes. A-level

technicians tend to be assigned to the same service routes on a daily basis. CSO orders tend to be assigned to technicians specifically because they are grouped near each other. While some of these aspects may not be critical to an accurate representation of resource impact, many should be explicitly modeled.

- Capacity dependence. The resource pools for AMR drivers and B-level techs to perform CSO orders are simulated as independent. In reality these are the same pool.
- Explicitly define impact of meetings. The main finding relating to the impact of meetings on resource capacity was founded on reasonable but unverified assumptions. We first assume that removing technicians from the field for meetings contributes to variability in resource capacity. We then assume that the ability to schedule meetings during the workload gaps would have certain impacts on the amount of variability. Instead, the draw down in resources should be explicitly modeled. We could then shift this drawdown to the workload gap days, and assess the actual impact it could have.

We feel that the application of job shop methods has been illuminating to the subject of utility order processing. The framing allowed us insight into categorize aspects of the Company's procedures we might otherwise not have such as various order review/release concepts for allowing orders into the system, characterizing the dispatch rules used by the company when assigning orders and workloads to technicians, the benefits of workload variance reduction vs. variance management. We feel there are deeper possibilities to bring job shop methods to bear for utility companies in general, such as investigating the benefits of various dispatching rules under differing environmental scenarios such as break downs and emergency orders.

There is a body of literature on the insertion of idle times into a job shop schedule. Much of it deals with inserted idle time in order to achieve more efficient ordering of jobs so as to decrease tardiness penalties [36]. This type of scheduling, however, is not similar to our methods. There is also research concerning the insertion of idle time to absorb the impact of machine breakdowns such as Mehta and Uzsoy [35], O'donovan et al. [8], Akturk and Gorgulu [37] and Mason et al. [38]. Yet these works assume both that there is no control over order acceptance and that minimizing completion time of orders is the chief priority.

Our assumptions and goals in this work were slightly different. We assumed some control over order entry. Also, our goal was not to speed all orders through the system. Rather, our goal was to level workloads across the system and ensure consistent resource utilization. We feel our work could and should lead to further investigation of use of buffers to create workload idle times. If buffers are triggered at regular intervals rather than based on machine condition, visibility of order flow through a shop will be greatly increased, and idle times could be known well in advance. These factors could greatly increase the ability of accurate due date setting at the input level. While the overall time an order spends in the system may increase, total tardiness could be significantly improved.

There are many avenues for future research specific to this project as well. A superior estimation of the potential reduction in resource variability is needed, along with cost benefit and economic impact assessment. Should the Company be interested in pursuing an organizational change, a re-routing project to better fit the clusters to the needs specified here would be a fascinating challenge.

REFERENCES

- [1] R. W. Hall, "Cyclic scheduling for improvement," *Int. J. Prod. Res.*, vol. 26, no. 3, p. 457, Mar. 1988.
- [2] A. Seidmann, S. S. Panwalkar, and M. L. Smith, "Optimal assignment of due-dates for a single processor scheduling problem," *Int. J. Prod. Res.*, vol. 19, no. 4, p. 393, Aug. 1981.
- [3] S. S. Panwalkar, M. L. Smith, and A. Seidmann, "Common Due Date Assignment to Minimize Total Penalty for the One Machine Scheduling Problem," *Oper. Res. Oper. Res.*, vol. 30, no. 2, pp. 391–399, 1982.
- [4] J. B. Sidney, "Optimal single-machine scheduling with earliness and tardiness penalties," *Oper. Res.*, vol. 25, no. 1, pp. 62–69, 1977.
- [5] C. R. Bector, Y. P. Gupta, and M. C. Gupta, "Determination of an optimal common due date and optimal sequence in a single machine job shop," *Int. J. Prod. Res.*, vol. 26, no. 4, pp. 613–628, 1988.
- [6] P. S. Ow and T. E. Morton, "The single machine early/tardy problem," *Manag. Sci.*, vol. 35, no. 2, pp. 177–191, 1989.
- [7] J. Carlier and E. Pinson, "An algorithm for solving the job-shop problem," *Manag. Sci.*, vol. 35, no. 2, pp. 164–176, 1989.
- [8] R. O'Donovan, R. Uzsoy, and K. N. McKay, "Predictable scheduling of a single machine with breakdowns and sensitive jobs," *Int. J. Prod. Res.*, vol. 37, no. 18, pp. 4217–4233, 1999.
- [9] S. Melouk, P. Damodaran, and P.-Y. Chang, "Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing," *Int. J. Prod. Econ.*, vol. 87, no. 2, pp. 141–147, 2004.
- [10] F. Yalaoui and C. Chu, "Parallel machine scheduling to minimize total tardiness," *Int. J. Prod. Econ.*, vol. 76, no. 3, pp. 265–279, Apr. 2002.
- [11] F. YALAOUI and C. CHU, "An efficient heuristic approach for parallel machine scheduling with job splitting and sequence-dependent setup times," *IIE Trans.*, vol. 35, no. 2, pp. 183–190, 2003.
- [12] D. Nait Tahar, F. Yalaoui, C. Chu, and L. Amodeo, "A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times," *Int. J. Prod. Econ.*, vol. 99, no. 1–2, pp. 63–73, Jan. 2006.
- [13] S. C. Graves, H. C. Meal, D. Stefek, and A. H. Zeghmi, "Scheduling of re-entrant flow shops," *J. Oper. Manag.*, vol. 3, no. 4, pp. 197–207, Aug. 1983.
- [14] K. A. Aldakhilallah and R. Ramesh, "Cyclic scheduling heuristics for a re-entrant job shop manufacturing environment," *Int. J. Prod. Res.*, vol. 39, no. 12, pp. 2635–2657, 2001.
- [15] P. Kouvelis and S. Karabati, "Cyclic scheduling in synchronous production lines," *IIE Trans.*, vol. 31, no. 8, pp. 709–719, 1999.
- [16] R. Wójcik, "Constraint Programming Approach to Designing Conflict-Free Schedules for Repetitive Manufacturing Processes," in *Digital Enterprise Technology*, P. P. F. Cunha and P. P. G. Maropoulos, Eds. Springer US, 2007, pp. 267–274.
- [17] J. C. Irastorza and R. H. Deane, "A Loading and Balancing Methodology for Job Shop Control," *E Trans. E Trans.*, vol. 6, no. 4, pp. 302–307, 1974.
- [18] S. Eilon, I. G. Chowdhury, and S. S. Serghiou, "Experiments with the S_{ix} rule in job shop scheduling," *Simulation*, vol. 24, no. 2, pp. 45–48, 1975.

- [19] S. A. Melnyk, K. C. Tan, D. R. Denzler, and L. Fredendall, "Evaluating variance control, order review/release and dispatching: a regression analysis," *Int. J. Prod. Res.*, vol. 32, no. 5, pp. 1045–1061, 1994.
- [20] S. Shimoyashiro, K. Tsoda, and H. Awane, "Input scheduling and load balance control for a job shop," *Int. J. Prod. Res. Int. J. Prod. Res.*, vol. 22, no. 4, pp. 597–605, 1984.
- [21] W. Bechte, "Theory and practice of load-oriented manufacturing control," *Int. J. Prod. Res. Int. J. Prod. Res.*, vol. 26, no. 3, pp. 375–395, 1988.
- [22] M. Land and G. Gaalman, "Workload control concepts in job shops A critical assessment," *Int. J. Prod. Econ.*, vol. 46–47, pp. 535–548, Dec. 1996.
- [23] K. R. Baker, "The effects of input control in a simple scheduling model," *J. Oper. Manag. J. Oper. Manag.*, vol. 4, no. 2, pp. 99–112, 1984.
- [24] S. A. Melnyk, G. L. Ragatz, and L. Fredendall, "Load smoothing by the planning and order review/release systems: A simulation experiment," *J. Oper. Manag. J. Oper. Manag.*, vol. 10, no. 4, pp. 512–523, 1991.
- [25] P. R. Philipoom and T. D. Fry, "Capacity-based order review/release strategies to improve manufacturing performance," *Int. J. Prod. Res. Int. J. Prod. Res.*, vol. 30, no. 11, pp. 2559–2572, 1992.
- [26] G. L. Ragatz and V. A. Mabert, "A framework for the study of due date management in job shops," *Int. J. Prod. Res.*, vol. 22, no. 4, p. 685, Aug. 1984.
- [27] D. J. Davis, H. V. Kher, and B. J. Wagner, "Influence of workload imbalances on the need for worker flexibility," *Comput. Ind. Eng.*, vol. 57, no. 1, pp. 319–329, Aug. 2009.
- [28] K. N. Bott and L. P. Ritzman, "Irregular workloads with MRP systems: Some causes and consequences," *J. Oper. Manag.*, vol. 3, no. 4, pp. 169–182, Aug. 1983.
- [29] Linda Hendry and Brian Kingsman, "A Decision Support System for Job Release in Make-to-order Companies," *Int. J. Oper. Prod. Manag.*, vol. 11, no. 6, 1991.
- [30] L. Onur and W. J. Fabrycky, "An Input/Output Control System For The Dynamic Job Shop," *IIE Trans. IIE Trans.*, vol. 19, no. 1, pp. 88–97, 1987.
- [31] S. Lingayat, J. Mittenthal, and R. M. O'Keefe, "An order release mechanism for a flexible flow system," *Int. J. Prod. Res.*, vol. 33, no. 5, p. 1241, 1995.
- [32] J. D. Wisner, "A review of the order release policy research," *Int. J. Oper. Prod. Manag.*, vol. 15, no. 6, pp. 25–40, 1995.
- [33] W. J. Hopp and M. L. Spearman, "Throughput of a constant work in process manufacturing line subject to failures," *Int. J. Prod. Res.*, vol. 29, no. 3, p. 635, Mar. 1991.
- [34] R. Cigolini, M. Perona, and A. Portioli, "Comparison of order review and release techniques in a dynamic and uncertain job shop environment," *Int. J. Prod. Res.*, vol. 36, no. 11, pp. 2931–2951, 1998.
- [35] S. V. Mehta and R. M. Uzsoy, "Predictable scheduling of a job shop subject to breakdowns," *IEEE Trans. Robot. Autom.*, vol. 14, no. 3, pp. 365–378, 1998.
- [36] J. J. Kanet and V. Sridharan, "Scheduling with Inserted Idle Time: Problem Taxonomy and Literature Review," 01-Feb-2000. [Online]. Available: <http://pubsonline.informs.org/doi/abs/10.1287/opre.48.1.99.12447>. [Accessed: 17-Nov-2013].
- [37] M. S. Akturk and E. Gorgulu, "Match-up scheduling under a machine breakdown," *Eur. J. Oper. Res.*, vol. 112, no. 1, pp. 81–97, Jan. 1999.

- [38] S. J. Mason, S. Jin, and C. M. Wessels, “Rescheduling strategies for minimizing total weighted tardiness in complex job shops,” *Int. J. Prod. Res.*, vol. 42, no. 3, pp. 613–628, 2004.

APPENDIX A – SIMULATION DATA TABLES

Cluster #	Simulation – 30 Replications (Values in Thousands)				Actual
	Average Total	Half-width	Minimum	Maximum	Total
1	375	9.5	324	440	379
2	406	6	370	447	425
3	439	4.2	417	482	449
4	303	7.5	269	351	300
5	315	4.2	286	340	315
6	314	3	299	330	315
7	298	3.2	275	313	293
8	368	4.2	339	388	351
9	436	5.8	396	455	434
10	415	7	373	445	420
11	283	5.2	253	316	286
12	416	6	388	448	420
13	397	7	356	429	404
14	478	10.5	421	536	489
15	464	9	419	499	484
16	473	7.4	437	511	489
17	392	6.6	360	437	428
18	522	10.2	482	582	550
19	453	10.3	398	497	470
20	389	7	350	424	402
21	243	6.2	212	288	251

Table 11 – Total Meters Input by the Base Simulation

Cluster #	Simulation - 30 Replications				Actual
	Average Total	Half-width	Minimum	Maximum	Total
1	1580	44.3	1390	1780	1646
2	2050	44.5	1820	2290	2215
3	2220	41.7	2000	2390	2389
4	1330	34.3	1170	1570	1388
5	1210	25.7	1090	1320	1270
6	2010	37.1	1840	2220	2133
7	2000	37.4	1790	2230	2080
8	1360	32.3	1190	1520	1369
9	2280	49.2	2040	2580	2435
10	2110	26.3	1980	2260	2283
11	2110	39.4	1840	2380	2241
12	1590	24.4	1500	1730	1609
13	2060	27	1950	2230	2021
14	2920	49.9	2620	3200	2899
15	2700	44.2	2540	3010	2732
16	3080	31.7	2910	3240	3009
17	3190	43.6	2880	3500	3265
18	3190	39.4	3020	3410	3197
19	2240	40.7	2020	2530	2250
20	2100	35.9	1860	2250	2112
21	1110	30.1	965	1320	1174

Table 12 – CSO Orders by Cluster for Base Simulation

Simulation	Measurement	Days In CSO Queue	Average Queue Length	Maximum Queue Length	B-Utilization	A-Utilization
Base						
	Average	1.21	146	967	0.91	0.88
	STD	0.33	41.8	187	0.03	0.03
	Half-Width	0.13	15.6	69.7	0.01	0.01
	Minimum	0.79	91.5	755	0.86	0.84
	Maximum	2.43	295	1580	0.97	0.93
STD8000						
	Average	1.10	137	889	0.94	0.90
	STD	0.46	58.6	208	0.03	0.04
	Half-Width	0.17	21.9	77.7	0.01	0.01
	Minimum	0.51	63.6	606	0.89	0.83
	Maximum	2.26	287	1490	1	0.98
STD4000						
	Average	1.13	141	947	0.94	0.90
	STD	0.41	52.8	198	0.02	0.03
	Half-Width	0.15	19.7	73.9	0.01	0.01
	Minimum	0.66	82.2	596	0.90	0.86
	Maximum	2.23	282	1510	0.98	0.96
STD2000						
	Average	1.21	150	928	0.95	0.92
	STD	0.5	62	253	0.02	0.02
	Half-Width	0.19	23	94	0.01	0.01
	Minimum	0.7	86	606	0.92	0.86
	Maximum	2.84	351	1700	0.99	0.96

Table 13 – Batch Size Variability Experiments Results

Simulation	Measurement	Days In CSO Queue	Average Queue Length	Maximum Queue Length	B-Utilization	A-Utilization
BaseORR						
	Average	1.67	212	1030	0.97	0.93
		1.21	146	967	0.91	0.88
	STD	0.68	88	242	0.03	0.03
		0.33	41.8	187	0.03	0.03
	Half-Width	0.26	32.8	90.3	0.01	0.01
		0.13	15.6	69.7	0.01	0.01
	Minimum	0.72	92.2	663	0.93	0.88
		0.79	91.5	755	0.86	0.84
	Maximum	3.43	439	1490	1.04	1.03
		2.43	295	1580	0.97	0.93
8000ORR						
	Average	1.22	152	861	0.96	0.92
		1.10	137	889	0.94	0.90
	STD	0.37	46.5	166	0.02	0.03
		0.46	58.6	208	0.03	0.04
	Half-Width	0.14	17.4	62	0.01	0.01
		0.17	21.9	77.7	0.01	0.01
	Minimum	0.7	87	580	0.92	0.86
		0.51	63.6	606	0.89	0.83
	Maximum	2.43	310	1240	1	0.97
		2.26	287	1490	1	0.98
4000ORR						
	Average	1.17	146	853	0.96	0.92
		1.13	141	947	0.94	0.90
	STD	0.4	50.8	194	0.02	0.03
		0.41	52.8	198	0.02	0.03
	Half-Width	0.15	19	72	0.01	0.01
		0.15	19.7	73.9	0.01	0.01
	Minimum	0.66	80.6	582	0.92	0.87
		0.66	82.2	596	0.90	0.86
	Maximum	2.41	307	1410	1	0.97
		2.23	282	1510	0.98	0.96

Table 14 – Output Comparison Original/ORR The values in bold are measurements for the ORR experiments. For comparison the values from the Base analogue are included immediately below the ORR values in standard text.

Simulation	Measurement	Days In CSO Queue	Average Queue Length	Maximum Queue Length	B-Utilization	A-Utilization
STD8000						
	Average	1.10	137	889	0.94	0.90
	STD	0.46	58.6	208	0.03	0.04
	Half-Width	0.17	21.9	77.7	0.01	0.01
	Minimum	0.51	63.6	606	0.89	0.83
	Maximum	2.26	287	1490	1	0.98
STD4000						
	Average	1.13	141	947	0.94	0.90
	STD	0.41	52.8	198	0.02	0.03
	Half-Width	0.15	19.7	73.9	0.01	0.01
	Minimum	0.66	82.2	596	0.90	0.86
	Maximum	2.23	282	1510	0.98	0.96
TRBase						
	Average	1.44	183	948	0.97	0.93
	STD	0.47	61.5	207	0.02	0.03
	Half-Width	0.18	23	77	.01	.01
	Minimum	0.81	102	567	0.93	0.88
	Maximum	2.76	357	1400	1.02	1
TRSTD4000						
	Average	1.43	183	976	0.97	0.93
	STD	0.48	63	269	0.02	0.03
	Half-Width	0.18	23.6	100	0.01	0.01
	Minimum	0.78	97	646	0.92	0.86
	Maximum	2.87	376	2030	1	1

Table 15 – Comparison results for Targeted Release Simulations

Simulation	Measurement	Days In CSO Queue	Average Queue Length	Maximum Queue Length	B-Utilization	A-Utilization
TRBase						
	Average	1.44	183	948	0.97	0.93
	STD	0.47	61.5	207	0.02	0.03
	Half-Width	0.18	23	77	0.01	0.01
	Minimum	0.81	102	567	0.93	0.88
	Maximum	2.76	357	1400	1.02	1
TRAdRes						
	Average	0.93	118	627	1	1
	STD	0.3	39.7	135	0.02	0.02
	Half-Width	0.11	14.8	50.6	0.01	0.01
	Minimum	0.55	67.7	454	0.97	0.97
	Maximum	1.79	226	997	1.03	1.03

Table 16 – Targeted Release Base vs. Targeted Release Adjusted Resources

Simulation	Measurement	Days In CSO Queue	Average Queue Length	Maximum Queue Length	B-Utilization	A-Utilization
Base						
	Average	1.21	146	967	0.91	0.88
	STD	0.33	41.8	187	0.03	0.03
	Half-Width	0.13	15.6	69.7	0.01	0.01
	Minimum	0.79	91.5	755	0.86	0.84
	Maximum	2.43	295	1580	0.97	0.93
BaseAdRes						
	Average	1.02	128	746	0.99	0.99
	STD	0.23	30	100	0.01	0.01
	Half-Width	0.09	11.2	37.2	0.005	0.005
	Minimum	0.65	79.3	586	0.96	0.96
	Maximum	1.57	199	1070	1.02	1.02

Table 17 – Base vs. Base Adjusted Resources

Arrivals	Duration (hrs)
0	7
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	839
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	671
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	839
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	671
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	671
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	671
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	839
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	671
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	671
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	839
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	671
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	839
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	671
ANINT(NORM(3.5e+004, 8.0e+003))	1
0	832

Table 18 – Cluster 1 Arrival Schedule

Day of the Week	Technicians Working
Monday	(7, 1)
Tuesday	(7, 1)
Wednesday	(7, 1)
Thursday	(7, 1)
Friday	0
Saturday	0
Sunday	0

Table 19 – B-Level Tech Schedule TRAdRes Scenario The number of technicians working is described by a normal distribution. The first number in the pair is the mean and the

Day of the Week	Technicians Working
Monday	(5, 1)
Tuesday	(5, 1)
Wednesday	(5, 1)
Thursday	(5, 1)
Friday	0
Saturday	0
Sunday	0

Table 20 – A-Level Tech Schedule TRAdRes Scenario The number of technicians working is described by a normal distribution. The first number in the pair is the mean and the