Proceedings of the Annual General Donald R. Keith Memorial Conference
West Point, New York, USA
May 3, 2018
A Regional Conference of the Society for Industrial and Systems Engineering

Design of a Nearest- Repair-Site Decision Support System for Water Utility Repair

Jose Soberanis-Chacon, Wisal A. Mohamed, Christopher F. Pertsch V, and Jomana Bashatah

Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA 22030, USA

Corresponding authors' Emails: jsoberan@gmu.edu, wmohame3@gmu.edu, cpertsch@gmu.edu, jbashata@gmu.edu

Author Note: Thank you: Mr. Christopher Coit (DC Water), Dr. Lance Sherry, Dr. Abbas Zaidi, Mrs. Paula Lewis, Dr. Kathryn Laskey, Dr. Leonard Adelman (GMU), and Mr. George Blaha (Raytheon)

Abstract: Water Utilities must maintain the functionality of water mains and their components even as the infrastructure ages. This must be accomplished without budget increases. For example, the water utility for the District of Columbia has experienced an increasing backlog of open work orders. An analysis of daily repair schedules identified two opportunities of improvement. First, the first-come/first-served assignment of repairs to crews did not consider the travel time between repair sites. Second, idle time occurs at the repair site when a crew wait for another function to complete their task (e.g. urban forestry). This paper describes a web-based tool that: (1) groups repair sites for a crew based on minimizing travel time, and (2) identifies nearby "quick" repairs that can be completed while the crew is waiting. A simulation analysis of these two features showed and 55% reduction in travel time, and a 6.2% reduction in backlogged work orders.

Keywords: Process Improvement, Process Engineering, Water Utility repair, Web Application, Python, Flask, Wrench Time Maximization, Travel Time minimization, Google Maps API

1. Introduction

1.1 DC Water Utility Workflow

DC Water's workflow process begins when a customer reports an incident or a damage to the infrastructure, which creates a work order. A crew is assigned to investigate the work order to determine if the work order is located on public property and the repair can be performed. The next step is to assign a priority number, which assess the severity of the work order from 1-5, five signaling a severe situation and one being relatively benign. Priority five work orders must be completed immediately due to water services being shut down for customers. Priority four work orders can be done within 24 hours, while priorities 1-3 are considered low priority and can be done at the foreman's discretion. All of these work orders go to the general foreman who assigns the work on a first come first serve basis to the three foremen.

Water Utilities are required to maintain the functionality of water mains and their components even as the infrastructure ages. This must be accomplished without budget increases. For example, the water utility for the District of Columbia has experienced an increasing backlog of open work orders due to an Automated Meter Reading replacement program, which created +33 additional work orders per week, and as a result, work orders that are opened for more than 90 days have increased 40%.

Analysis of the repair performance based on the first-come/first-served scheduling process identified the following opportunities:

(1) Repair crew experience unanticipated idle time at the repair site. For instance, main repairs can take up to seven hours to fix but has a great amount of wait time for the valve crew to shut off the valves. Another example is if a tree root is entangled within a pipe that needs repair, there is an additional wait time for the approval of urban forestry to remove the tree, along with other types of delays as illustrated in the timeline in Figure 1.

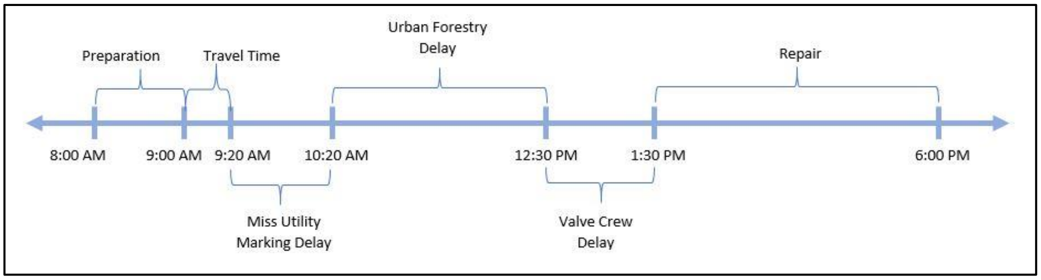

Figure 1. Timeline of Work order with delays

(2) Another issue created by the current random scheduling system is travel time between work orders. This is leading to work being spread around with no focus on distance between each work order thus increasing the travel time of the foreman form one location to another. The average travel time between work orders is 20 minutes. A travel time simulation was performed to reduce the driving time from one location to another by clustering work orders for the day based on close area proximity. Figure 2 shows how D.C. Water is currently scheduling their work order, each color representing a different work crew. There are three foremen known as Omegas. Each foreman is referred to as Omega 10, 20, or 30, and each foreman has three crews Ω-11, Ω-12, Ω-13, Ω-21, Ω-22, Ω-23, Ω-31, Ω-32, Ω-33. They are color coded on the map in figure 1 to illustrate how randomized the work order locations are with no consideration to travel time.

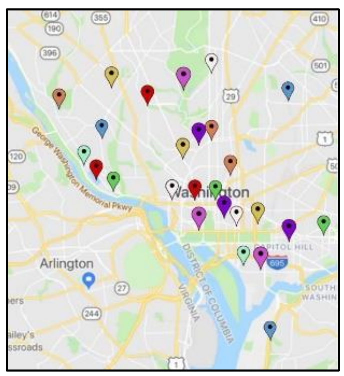

Figure 2. Current travel time between work orders shows randomness and longer travel time for crew foreman

Figure 3 illustrates how a crew's schedule would look if the schedules were based on work order proximity. The simulation showed that the estimated time between work orders dropped from 20 to 9 minutes. According to Pertsch V et al. in the paper titled 'Design of a Decision Support System for the Scheduling of a Workflow Process for a Water Utility Company', the simulation was also able to show statistically significant improvements using T-test with alpha 0.05. Number of closed work orders per year had a 2.8 % improvement and a 6.2% improvement in number of closed backlog work orders.

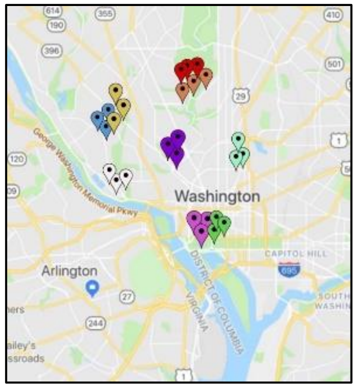

Figure 3. Grouped work orders based on close area proximity shows lesser travel time for each crew foreman

1.3 Solution

A web-based tool to reduce idle-time and travel time. The system was built to give a water utility company a data driven scheduling decision aid that shows the user (i.e. foremen) a list of nearby work orders according to an input location with the help of Google Maps, and details about the work orders such as mean time for completion, travel time, type, and description. The system has a filtering feature to give the foreman control over their schedule. The system will filter based on priority, type and work order number. *Castorea* can also be used by various utility companies that require on-site maintenance because it has a data-driven custom made scheduling system which increases productivity in an enterprise, and custom schedules tailored to the need of the customer. *Castorea* has a additonal feature built-in to the system called "Wrench Time Maximization system.", which shows a list of closest work orders that can be done during the crew's on-site wait time.

2. Methodology of Web Application

The tool, named *Castorea* which means beaver in Latin, is a multi-purpose application that allows water utility companies to increase productivity by reducing travel time between repairs and reducing on-site wait time Over time, this will lead to a higher output of completed work orders, a better maintained pipeline infrastructure which leads to less failures, all while increasing customer satisfaction.

To access the website, the company will be granted a unique login and password. This ensures that all potentially classified information cannot be viewable to anyone who has the website URL. Once logged in, the user will be redirected to the dashboard, where they can import the work order data, however the data must be in Excel (.xlsx) format in order for the website to read the file and extract its information, and store it within the website to be used when building the schedules or requesting nearby work. Along with those features, *Castorea* also allows the user to view all repair crew schedules and remove work orders upon completion. This website is also compatible for mobile devices.

Using Python 2.7, *Castorea* was built using the Flask web development tool and a variety of libraries such as Openpyxl to read the excel files and Google Maps Directions API to calculate the travel time between two addresses. Bootstrap was used in order to build the user interface in HTML code, along with JavaScript for sorting/filtering tables and decomposing the navigation bar items to one single dropdown menu in mobile mode. *Castorea* also comes built in with a user login infrastructure that grants the user access to the website with the proper credentials, which are stored in a database built using sqlite3.

3. Features of *Castorea*

Castorea has two features: (1) A Schedule Building Function and (2) A Wrench Time Maximization Function.

3.1 Schedule Building

In order to minimize travel time and balance complexity of work amongst repair crews, *Castorea* provides a scheduling system where once the first work order is selected, the system compares that address and outputs the top ten work orders nearest to

ISBN: 97819384961-2-7

that work order as seen in Figure 4. By default, the list displays from nearest to farthest work order, however the table can also be sorted by any of the attributes in the header when clicked.

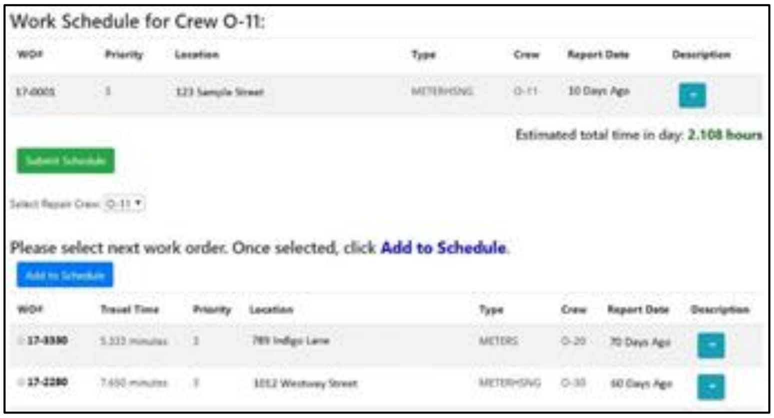

Figure 4. Schedule building page of Castorea after user selects first work order of the day. The system also provides estimated repair time as seen in the left top corner of the screen in green.

Once chosen, the system will continue looping through this process until the schedule is finished being built, and thus the user will submit the schedule which can then be viewable on the "View Schedule" page as seen in Figure 5. This page also gives the foremen a general idea of how much time will be spent to complete every work order. The estimated total time is calculated by taking the travel time between each work order and add the average repair time of every work order. Once a work order is completed, the user can select the work order and then click on the "Remove from Schedule" button.

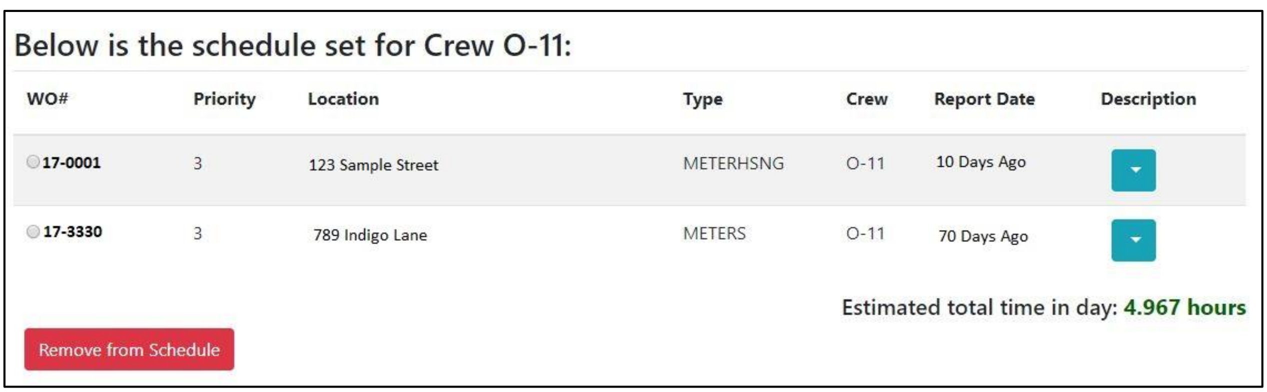

Figure 5. "View Schedule" page for repair crew O-11. Total time (bolded green) also takes travel time into consideration

3.2 Wrench Time Maximization Tool

Given the repair crew's current address, *Castorea* will follow a similar methodology as the scheduling tool, however along with displaying the top ten work orders, the user can also filter what work is displayed. The user can specify what type of work and also input approximate idle time, which will display work orders that can be done within the allotted idle time. This takes into consideration the time to get there, complete the repair, time to get back, as well as an additional 30-minute buffer for any unforeseen reasons.

Once the correct work order is selected, the user will be taken to a confirmation page as seen in Figure 6. If the wrong work order was selected, then the user can return to the previous list by pressing the "Return to List of Work Orders" button.

ISBN: 97819384961-2-7

Once confirmed, the work order is added to the repair crew's schedule, which can be seen on the "View Schedule" portion of the site.

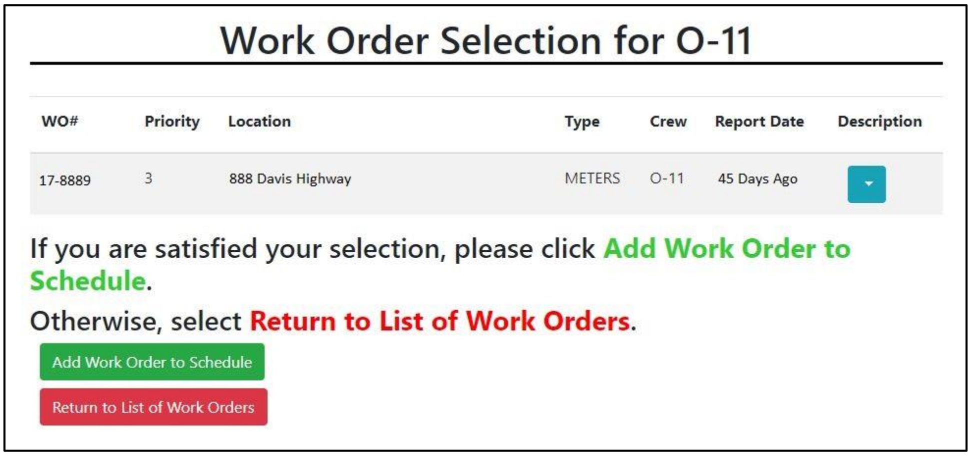

Figure 6. Wrench Time Maximization tool front page

4. Usability Testing

In order to evaluate how user friendly *Castorea's* interface is a usability test was performed. A within subject test was done with four foremen from D.C. Water where they performed three different tasks:

1. Setting Schedules
2. Wrench Time Maximization
3. Viewing Schedules

The usability goal requirements of the system were determined by the stakeholders. The requirements were:

1. The user shall be able to set a schedule with three work orders in under 3 minutes
2. The user shall be able find the closest work order and add it to the schedule in under 2 minutes
3. The user shall be able to find and view the schedules in under 20 seconds

The order of the tasks was randomized, and each user was timed for each task. After the tasks were done a questionnaire was provided in order to measure satisfaction and obtain feedback from the users. Table 1 shows the six questions, which are measured from 1 to 5 (unsatisfactory to satisfactory, unlikely to likely for question 6).

Table 1. Survey questions

1) How easy was it to make a schedule?	**4) Was it easy to sort work orders based on priorities?**
2) How easy was it to view the complete schedule?	**5) What is your overall satisfaction?**
3) Was the website easy to follow?	**6) Would you use this tool?**

4.1 Usability Test Results

The results showed that the average time to complete tasks 1,2 and 3 were 105.05, 77.75 and 5.25 seconds with a standard deviation of 55.53, 16.23 and 1.707 seconds respectively. For all of the tasks a P-Value larger than 0.05 was achieved indicating that they were all significant and passed all of the goal requirements. The users were asked to complete a 6-question survey to evaluate the overall satisfaction of the system. They were asked to evaluate their satisfaction with making and viewing schedules, navigating through the website, and sorting work orders. As seen in table 2, all the questions received a P-Value larger than 0.05 meaning that overall, they were satisfied with the website.

ISBN: 97819384961-2-7

Table 2. Survey Results

Question	Mean	SD	P-Value
1 (Scheduling)	4	1.15	0.9092
2 (Viewing Schedules)	4.5	0.577	0.9931
3 (Website Navigation)	4.25	0.95	0.9602
4 (Sorting Work Orders)	4	0.816	0.9541
5 (Overall Satisfaction)	4.75	0.5	0.997
6 (Would you use the tool?)	4.75	0.5	0.997

5. Simulation Analysis of Productivity Improvements

A simulation of analysis of productivity improvements using Castorea was conducted (Pertsch, Soberanis, Mohamed, Bashatah 2018). The simulation showed a 55% reduction in travel time. This resulted in a 6.2% reduction in backlog. Based on these results, DC Water are currently conducting field tests of the tool. Performance results will be available soon.

6. Business Case

A work order in District of Columbia water utility company cost ~$6,000. DC Water was able to complete 1600 work orders in 2017. Based on the simulation analysis of productivity improvement conducted (Pertsch, Soberanis, Mohamed, Bashatah 2018), an additional 496 work orders can be done per year, which costs ~$2,976,000, but if DC Water uses Castorea, they would only pay $3,461 per year. The aim is the expand to other water utility companies around the country. The cost projection in 10 years is $729,175 and the revenue projections are at $2,736,000. The profit in 10 years is $2,006,825, the return on investment is 274%. The breakeven is in the 3rd year.

7. References

Gazoni, E., & Clark, C. (2016, January 18). Openpyxl Documentation. Retrieved March 28, 2018, from https://media.readthedocs.org/pdf/openpyxl/latest/openpyxl.pdf

Kleint, J. (2009, September 11). Googlemaps – Google Maps and Local Search APIs in Python. Retrieved March 28, 2018, from http://py-googlemaps.sourceforge.net/

Pertsch V, C. Bashatah, J. Soberanis, J. Mohamed, W. (2018) Design of a Decision Support System for the Scheduling of a Workflow Process for a Water Utility Company, Proceeding of SIEDS, 2018. Char

Ronacher, A. (2008). Welcome to Jinja2. Retrieved March 28, 2018, from http://jinja.pocoo.org/docs/2.10/

Ronacher, A. (2010). Welcome to Flask. Retrieved March 28, 2018, from http://flask.pocoo.org/

Ronacher, A. (2014). Werkzeug. Retrieved March 28, 2018, from http://werkzeug.pocoo.org/