Wastewater systems planned maintenance scheduling using multi-objective optimisation

Sabrina Draude[1] sd406@exeter.ac.uk
Edward Keedwell[1] E.C.Keedwell@exeter.ac.uk
Zoran Kapelan[1] Z.Kapelan@exeter.ac.uk
Rebecca Hiscock[2] Rebecca.Hiscock@dwrcymru.com

[1]College of Engineering, Mathematics and Physical Sciences, University of Exeter, EX4 4QF, UK

[2]DCWW, Fortran Road, Cardiff, CF3 0LT, UK

ABSTRACT

Water utilities (WU) in the UK are responsible for providing sewerage disposal for customers across the country. Blockages within sewer networks can be disruptive for customers, damaging to the local environment and costly to rectify. Effective scheduling of preventative maintenance (PM) is an important for WU to prevent blockages, reduce costs and protect the environment. In this paper, we describe a novel multi-objective optimisation methodology to the scheduling of PM applied to a case study in Swansea, Wales. The results of real-world trials demonstrate that solutions generated by the proposed method achieve a 13.8% increase in jobs completed for compared to the standard approach used in November 2019.

CCS CONCEPTS

Applied computing→ Multi-criterion optimisation and decision making → *planning for stochastic actions*

KEYWORDS

Wastewater systems maintenance, multi-objective optimization

ACM reference format

S. Draude et al, 2021. Wastewater systems planned maintenance scheduling using multi-objective optimisation. In *proceedings of GECCO. ACM '21, New York, NY, USA, 2 pages.* https://doi.org/10.1145/3449726.3459535

1. INTRODUCTION

To produce a practical, optimal maintenance plan for a WU it is necessary to combine two problems: maintenance planning and workforce vehicle routing.

Maintenance activities are generally categorised into two subsets, reactive maintenance (RM) and PM. RM is where sewer maintenance is performed once an asset has failed and a blockage has formed. This method can incur high costs, customer dissatisfaction and environmental damage. PM is performed before the blockage has fully formed, allowing the asset to maintain its function, reduce its deterioration rate and extend its lifecycle. However, due to limited resources and cost constraints faced by WU, all PM cannot be attended to as soon as required. Therefore, the most critical maintenance operations need to be prioritised and maintenance scheduling optimisation models have been developed to support the management of maintenance crews. The stated problem is an integrated maintenance scheduling and vehicle routing problem. The objective is to meet the needs of the highest priority jobs as soon as possible within scheduling constraints. The scheduling methodology has been created using the multi-objective optimization library platypus to aid solving the problem [1]. Here we propose a dynamic scheduling approach combined with the optimisation of crew routes:

Task 1: Decide the priority of pipes needed to be maintained in the near future according to the current network status and environmental conditions. To decide job priority score, the potential factors affecting job priority are entered into a GIS decision making (DM) model. The model assigns a job priority score based on pipe characteristics derived from a blockage likelihood machine-learning model [2], the population density around the asset, the distance between the asset and a watercourse, whether the incident is repeated, and whether the asset has ever experienced a DG5 (A property that has experienced indoor flooding due to inadequacy of the public sewer network). Add all this data to a routing and scheduling database.

Task 2: Construct crew travel routes using a routing model, which minimize distance and cost but maximize the number of jobs and priority of jobs completed daily. This is achieved through more efficient route plans whilst adhering to company specific time constraints.

Previous trials conducted in Cardiff, Wales maximised the priority score and minimised the total cost objectives. The trials resulted in an average of an excessive 63 miles of daily travelling between jobs. Therefore, a third objective of minimizing travel time was added. As a comparison, the Swansea trials with the new objective, resulted in an average of 17.8 miles travelled per day, allowing more jobs to be completed and reduced travel costs.

2. METHOD

This problem is similar to the well-known periodic vehicle routing problem. However, there are a number of distinguishable characteristics. Firstly, presently over the monthly period, not all PM jobs are completed within Swansea. The proposed methodology will aim to complete all jobs. Secondly, currently the maintenance crew is given limited scheduling guidance. This method uses DM to understand which pipes the WU considers to be a blockage risk and utilize that into a priority-scheduling model. Thirdly the proposed multi-objective optimization approach can identify Pareto-optimal maintenance plans and help understand the trade-offs between the three objectives. Finally, the case study demonstrates the applicability of the proposed methodology and the real world implications of it. A simplified outline of the objective function used is listed below:

Step 1: Create a matrix of distances and durations between all the jobs (J) and the crew Homebase (HB) using the routing model.

Step 2: Calculate the cost of each of the J in the JC for the initial population using the costing database and cost model. (*Job type + Crew cost per hour + material costs + after hour costs*).

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459535

Step 3: Create a dictionary for each job available for the month with details on job cost, job priority score, job postcode, job length, if the job is a DG5 event, job latitude and job longitude sourced from the database. Then place all the individual dictionaries into a single job dictionary of dictionaries.

Step 4: Run the *MOF* using the three objectives and three constraints. The *MOF* iterates through *days_list* (the amount of scheduling days required by the user) in a one day function (*one_day_function*). Start with the following steps for *i* in *days_list*:

Step 4.1: While T_{dd} is less than *Max_time* add J_i from the *initial_job_chromosome* to the *one_day_chromosome*.

Step 4.2: Calculate the total cost T_C of J_i by adding the *JC (*sourced from the job dictionary*)* and the travel cost between *Ji* and the previous location (J_{i-1} or *HB*) or to the *HB* if T_{dd} is greater than *Max_time*. The travel cost and travel duration (T_{di}) is calculated by sourcing the travel distance in miles between J_i and the previous location (or last location if it's the last job) by using the distance matrix in step 1. Miles * £1.20 gives the Jt_i. Add the (*Tc*) to the total day cost *(TDC).* Add the job duration (Jd_i) sourced from the job dictionary and the travel duration (td_i) to the total duration for the day (*tdd*) Add the jps_i to the total priority score for the day (*tps*). Add the *one_day _chromosome* to the *total_chromosome*

Step 4.3: If the *tdd* is longer than the *jobmax*, add a penalty cost of £50 to the *tdc*. If the *tdd* is longer than JobMax + 30 min leeway add a travel constraint violation (CV_{tr}) If the T_d between any job exceeds 5 miles, add a £50 penalty to the T_{dc}. If the td_i exceeds 20% of the *Tdd* add a travel constraint CV_{tc}. If the job numbers are duplicated in the *JC*, add a job duplicate constraint CV_{dj}.

Step 4.4: Add J_i to job counter (*job_counter += 1*) and return to step 4.1 with the next *J*. If J_i does not satisfy the while constraint go to step 4.5.

Step 4.5: Return the objectives: T_{dc}, T_{ps}, T_{dd} and all the *CV* and add them to the overall objective cost (O_c), priority score (O_p), total distance (O_d), and *CV* in the *MOF.*

Step 5: If the loop has reached the end of the days available to schedule in *days_list*, return the overall objectives and *CV's*. If not, then *days_list* += 1 and return to step 4.

The parameter tuning for this case study are listed below:

Parameters	Values
Initial population size	100
Number of generations	50,000
Crossover rate	0.6
Mutation rate	0.2

For this case study, the methodology uses NSGA-II. SPEA-II was also tested, but the statistical analysis demonstrated that there was a statistically significant difference at the 5% level, with NSGA2 performing better. A similar multi-objective routing problem solved by Jemai et al, 2012 [3] also found that NSGA2 outperformed SPEA2. In this case study the optimizer runs for 50,000 iterations, as previous experiments showed the hypervolume score tailed off at this point. The methodology outputs a Pareto front of the best non-dominated solutions. Multi-criteria decision analysis ensures that the user is given the solution that best fits their criteria, through the ranking of the options available. The highest ranked option creates a scheduling plan. The schedules produced from the case study covered the first 11 working days which enabled the completion of all PM jobs available; therefore, the first 11 working days in the human derived November 2019 schedule is used as a comparison.

3. PRELIMINARY RESULTS

This study was undertaken for the month of November 2020, in Swansea, South Wales. The results show a 13.79% increase in jobs completed over the month period for 2020, compared with the jobs completed in 2019. The results also show that jobs with a higher priority score are targeted in 2020. The schedules in 2020 were shown to be more efficient, with 7.2 miles travelled less each day for the crews. The objective performance of the Pareto set of scheduling options for the first 3-day schedule is shown in figure 1. The figure shows the difference in the number of solutions between 1 iteration and 50,000 iterations and the improvement of the results over the iterations. Figure 1 shows a clear progression towards the utopia point (minimal distance, minimal cost and maximised summed job priority score). The 2019 human-derived schedule is included as a reference.

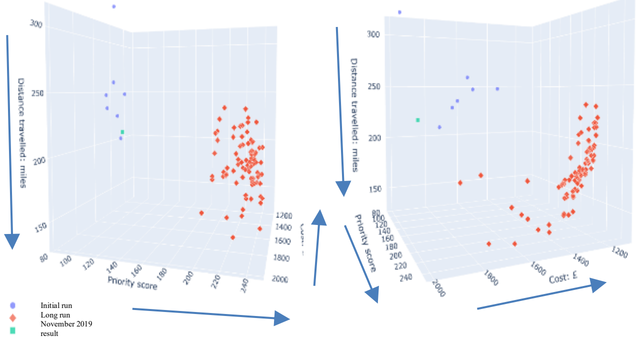

Figure 1. Pareto front differences between the initial random population and the long run (50,000 iterations) and the November 2019 schedule. The preferred direction of objective performance is indicated by arrows on each axis label.

4. Conclusions

This work has proposed a novel EA approach to the problem of real-world PM scheduling and routing for sewer maintenance. The approach generates schedules that improve significantly on those developed manually in 2019. The methodology could be adapted to schedule according to predicted weather events. In a first flush event, intensive rainfall after a period of dry weather causes an excess of debris in the sewer system. This in turn increases the likelihood of a blockage accumulating on a defect [4]. By adding in this additional factor, the methodology can be further improved. As a final point, this DM model could be implemented in blockage maintenance plans for other WU globally.

REFERENCES

[1] Hadka, D., Herman, J., Reed, P. and Keller, K., 2015. An open source framework for many-objective robust decision making. Environmental Modelling & Software, 74, pp.114-129.

[2] Bailey, J., Harris, E., Keedwell, E., Djordjevic, S. and Kapelan, Z., 2016. Developing Decision Tree Models to Create a Predictive Blockage Likelihood Model for Real-World Wastewater Networks. Procedia Engineering, 154, pp.1209-1216.

[3] Jemai, J., Zekri, M. and Mellouli, K., 2012, April. An NSGA-II algorithm for the green vehicle routing problem. In European Conference on Evolutionary Computation in Combinatorial Optimization (pp. 37-48). Springer, Berlin.

[4] Draude, S., Keedwell, E., Hiscock, R. and Kapelan, Z., 2019. A statistical analysis on the effect of preceding dry weather on sewer blockages in South Wales. Water Science and Technology, 80(12), pp.2381-2391