Predict-then-Optimise Strategies for Water Flow Control

Vincent Barbosa Vaz
The University of Melbourne, and the Australian Research Council OPTIMA ITTC, Australia

James Bailey
The University of Melbourne, Australia

Christopher Leckie
The University of Melbourne, Australia

Peter J. Stuckey
Monash University, and the Australian Research Council OPTIMA ITTC, Australia

Abstract

A pressure sewer system is a network of pump stations used to collect and manage sewage from individual properties that cannot be directly connected to the gravity driven sewer network due to the topography of the terrain. We consider a common scenario for a pressure sewer system, where individual sites collect sewage in a local tank, and then pump it into the gravity fed sewage network. Standard control systems simply wait until the local tank reaches (near) capacity and begin pumping out. Unfortunately such simple control usually leads to peaks in sewage flow in the morning and evening, corresponding to peak water usage in the properties. High peak flows require equalization basins or overflow systems, or larger capacity sewage treatment plants. In this paper we investigate combining prediction and optimisation to better manage peak sewage flows. We use simple prediction methods to generate realistic possible future scenarios, and then develop optimisation models to generate pumping plans that try to smooth out flows into the network. The solutions of these models create a policy for pumping out that is specialized to individual properties and which overall is able to substantially reduce peak flows.

2012 ACM Subject Classification Applied computing → Operations research; Mathematics of computing → Mixed discrete-continuous optimization

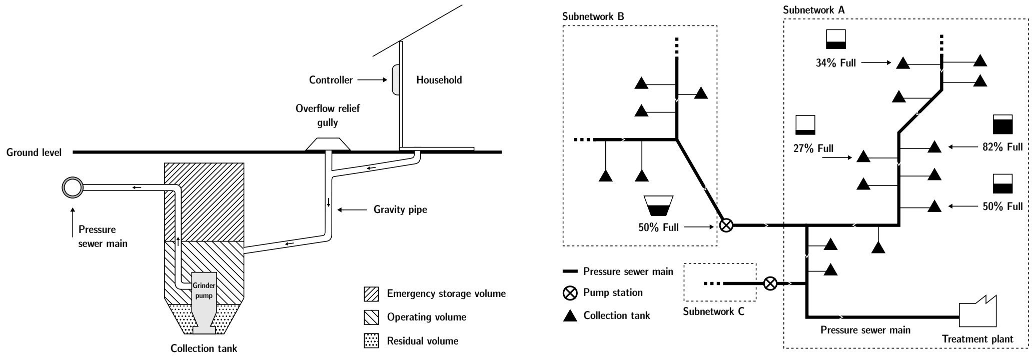
Keywords and phrases Water Flow Control, Optimization, Machine Learning

Digital Object Identifier 10.4230/LIPIcs.CP.2023.33

Acknowledgements This research was partially funded by the Australian Government through the Australian Research Council Industrial Transformation Training Centre in Optimisation Technologies, Integrated Methodologies, and Applications (OPTIMA), with industry partner South East Water, Project ID IC200100009.

1 Introduction

A Pressure Sewer System (**PSS**) is a network of pump stations used to collect and manage sewage from individual properties that cannot be directly connected to the classic gravity sewer network due to the topography of the terrain (gravity limitation). The sewerage gravitates to the pump station and is then pumped through a pressure main to a main sewer and on to wastewater treatment plants. This is illustrated in Figure 1a. A PSS is often composed of intermediate, bigger pump stations between sub parts of the whole network (Figure 1b). Pump stations collect household sewage from a sub part of the network and pump it to the main. Our focus in this paper is to optimise the operation of individual pump stations to balance the overall load at the treatment plant. In this paper, we only consider an existing network managed by our industry partner South East Water. The network that we consider is free of intermediate pump stations. Extending the approach to



(a) **(b)**

Figure 1 (a) Pressure sewer unit. (b) Pressure sewer network.

handle intermediate pump stations would require adding constraints to ensure the capacity of the pump station is not violated at any time.

We consider a classic scenario for PSS where volumes at the residential level are released without optimised control. When the collection tanks reach their capacity, water is released entirely into the network. This represents a considerable amount of sewage conveyed through the sewer network that must be handled by the treatment plant, resulting in stress on the treatment processes and increased capital costs of upsized pipe and pump networks. This simple control policy does not make good use of the network. Volumes can be retained in pressure sewer tanks at the residential level and selectively released in a way that can optimise the flows to provide network capacity increases for the current network, and improve operations of the downstream treatment plant.

Most of the water usage occurs in the morning and the evening peak loads, when people are home. This results in two, identifiable peaks of activity in the network. This sudden surge in activity represents a challenge for the water treatment processes that follow. Huge amounts of resources need to be deployed at the treatment plant at these times to cope with the volumes to be treated and the associated unpredictability. South East Water would like to flatten the input flow at the treatment plant to achieve greater plant operational efficiency. This is possible by leveraging the buffer capabilities of the collection tanks, assuming that each tank can be controlled individually.

Current operational strategy. The water industry has yet to integrate data-driven models and optimisation techniques to facilitate and control their processes in a more efficient and systematic way. Most water companies rely on operators' knowledge and experience to parameterise and control their network. In current operation each tank fills until it reaches its capacity (cut-in setpoint) and then fully (until cut-off setpoint) empties the tank. In a previous approach to tackling the problem of reducing maximum outflows on the network we investigated simply modifying the (cut-in) set points of the tanks to try to reduce homogeneous behaviour but this was not really successful. Without adjustment the tanks quickly reached a steady state where the usual morning and evening peak inflows again resulted in high outflow. Due to the location of the network in a holiday zone the set points needed to be adjusted frequently as usage behaviour fluctuated rapidly over weekends, summer etc.

In this paper we define an optimisation based approach to controlling the maximum outflow, by deciding in which time periods to empty the tanks. In order to flatten out peaks we need to have some idea of the future inflows into the tanks. Because patterns of water usage are quite distinct generating realistic time series of inflows is challenging with machine learning models. Hence we use simple historical sampling to generate plausible future inflows. We show how combining (simplistic) prediction models with an optimisation model to determine when to release sewage into the network from individual properties, can substantially reduce the peak flows in the network.

2 Problem Description

We model the problem of controlling a PSS system to reduce peak flows as a MIP. We discretize the control problem by working over a finite time horizon of n discrete time steps, which for our experiments are always 1 hour long. Note that this discretization is fine enough that none of our historical data has examples where a tank is filled from empty in under an hour. Finer discretization would allow some further reduction in peak outflows, but we expect that we capture most of the possible reduction using 1 hour discretization. The parameters for the water flow control model are shown in Table 1. In each time step we decide whether to empty the tank at a particular site. In the default control mechanism, when we decide to pump out of a tank, we empty it. This has the least wear and tear on the pumping and control mechanism.

Table 1 Parameters for the water flow control model.

Description	Parameter
Time horizon	$T = 1..n$
Set of tanks	S
Capacity of tank at site s (cut-in setpoint)	c_s
Minimum amount of water to be pumped out	m
Inflow to tank at site s during time period t	$i_{s,t}$
Tank level of site s tank at the beginning of the first time period	$l_{s,0}$

2.1 Direct Formulation

In this section, we propose a direct MIP formulation for the PSS problem, over a given set of tank sites S and time horizon T. The control systems for the tanks have simple functionality, we can (re-)set minimum and maximum tank levels, or initiate a pump out of the tank, but not control exactly how much volume is pumped out of the tank. This leads to the important decision we must make — *for each tank s at what times t should it be emptied?* $X_{s,t} \in \{0, 1\}$. The model makes use of auxiliary variables: $l_{s,t}$ the level of tank s at (the end of) time t; and $o_{s,t}$ the volume of water pumped out of tank s during time period t. The model is:

$$\text{minimize} \max_{t \in T} \sum_{s \in S} o_{s,t}$$

$$l_{s,t} = (l_{s,t-1} + i_{s,t}) * (1 - X_{s,t}), \qquad \forall s \in S, t \in T \qquad (1)$$

$$l_{s,t} \leq c_s, \qquad \forall s \in S, t \in T \qquad (2)$$

$$o_{s,t} = (l_{s,t-1} + i_{s,t}) * X_{s,t}, \qquad \forall s \in S, t \in T \qquad (3)$$

$$X_{s,t} = 1 \rightarrow o_{s,t} \geq m, \qquad \forall s \in S, t \in T \qquad (4)$$

$$X_{s,t} \in \{0, 1\}, \qquad \forall s \in S, t \in T$$

where Equation (1) computes the level $l_{s,t}$ in each tank s at the end of time period t (previous level plus inflows, unless emptied); Equation (2) ensures that each tank's level remains below tank capacity; Equation (3) computes the outflow $o_{s,t}$ from tank s at time t; and Equation (4) ensures that if the tank is emptied there is a minimum volume present (to prevent accelerated wear and tear on the infrastructure). The objective is to minimize maximum outflow across the period considered. Note that each of the constraints, and the objective are linear or easy to linearise.

2.2 Packing Formulation

The model above straightforwardly models the problem, but can become challenging to solve as the problem size grows. Next we instead consider the inflow to tank s at time t as a fixed amount of flow, we then decide when this should be pumped out. By precomputation we can then specify simple constraints to enforce that the capacity of tank is not exceeded. This leaves a packing problem, deciding when each chunk of water is pumped into the network.

The tank inflows are aggregated by hours and constitute the items of the problem. We require that the inflows are pumped out in order of inflow. The inflow at time t, $i_{s,t}$, must be pumped out, no earlier than t, and not later than $latest(s,t) = \min(\{t' - 1 \mid t \leq t' \leq n, \sum_{t \leq i \leq t'} i_{s,i} \geq c_s\} \cup \{n\})$ which would mean the tank capacity was exceeded, since no later flows can be pumped out before $i_{s,t}$. Note that we treat the starting tank level $l_{s,0}$ as an inflow at time 0, $i_{s,0} = l_{s,0}$. We can also define the last inflow that must be pumped out in the considered time period, $last(s) = \min\{t \mid \sum_{t \leq i \leq n} i_{s,i} \leq c_s\} - 1$. And for each tank and time define $below(s,t) = \max\{t' \mid t \leq t' < latest(s,t), \sum_{t < i \leq t'} i_{s,i} < m\}$ to be the latest time t' such that if the tank is emptied at time t the sum of inflows after it up to t' is too small to empty.

The new decision variables $p_{s,t,t'}$ determine the time t' when the inflow to s at time t is pumped out (including the original tank volume $t = 0$). The model is defined by:

$$\text{minimize} \max_{t \in T} \sum_{s \in S, t' \leq t} p_{s,t',t} \times i_{s,t}$$

$$\sum_{t \leq i \leq latest(s,t)} p_{s,t,i} = 1 \qquad s \in S, t \in 0..last(s) \tag{5}$$

$$p_{s,0,0} = 0 \qquad s \in S \tag{6}$$

$$\sum_{t \leq i \leq t'} p_{s,t,i} \geq p_{s,t+1,t'}, \qquad \forall s \in S, t \in 1..n, t' \in t+1..latest(s,t) \tag{7}$$

$$X_{s,t'} = 1 \rightarrow \sum_{t \leq i \leq t'} p_{s,t,i} \geq 1, \qquad s \in S, t \in T, t' \in t..latest(s,t) \tag{8}$$

$$\sum_{t' \leq t \leq latest(s,t')} p_{s,t',t} \geq 1 \rightarrow X_{s,t} = 1, \qquad s \in S, t \in T \tag{9}$$

$$X_{s,t} = 1 \rightarrow \sum_{t < i \leq below(s,t)} X_{s,i} \leq 0, \qquad \forall s \in S, t \in T \tag{10}$$

$$X_{s,t} \in \{0,1\}, \qquad \forall s \in S, t \in T$$

$$p_{s,t,t'} \in \{0,1\}, \qquad \forall s \in S, t \in 0..n, t' \in t..latest(s,t)$$

where Equation (5) enforces we pump out each inflow (before $last(s)$) exactly once; Equation (6) enforces that nothing is pumped out at time 0; Equation (7) enforces that the inflow to tank s at time t is pumped out no later than the time the inflow at time $t+1$ is pumped out, i.e. the inflows are pumped out in order; Equation (8) connects the emptied variables to the pump out variables by requiring that if tank s is emptied at time t' then each inflow before t' is pumped out by time t'; Equation (9) connects them in the other direction requiring that if any inflow is pumped out of tank s at time t then tank s is emptied at time t; and Equation (10) enforces that if tank s is emptied at time t then it is not emptied again until at least m units of flow have entered the tank. The objective minimizes maximum outflow, computed from the pumped out variables. Again the entire model is easy to linearise.

3 Predicted Water Usage Generation

Online optimisation problems can be augmented with a predictor that informs the model on future instances. Simple predictors can sample the inputs or sample the distribution of the inputs. More complex Machine Learning based predictors can learn from the distribution of the inputs as the online problem is being solved. Research [5, 1, 2, 3] has demonstrated that sampling the distribution of the inputs or providing estimates can significantly improve the quality of the solution. In practice, not all optimisation problems have access to the distribution of the inputs.

We wish to generate water usage predictions for each site. The collected water usage comes from diverse households exhibiting different behavioural patterns. Care must be taken when building a predictor to capture the seasonality and cycles in the data. Because of these properties, building an individual predictor for each site and time instance is unlikely to produce realistic distributions of water usage.

3.1 Historical Sampling

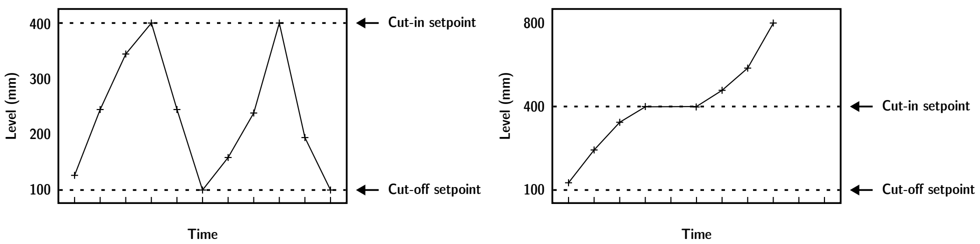
We use *historical sampling* as described by Bent and Van Hentenryck [4]. This generates samples from past subsequences in the historical data. Despite its simplicity, historical sampling captures the structure of the sequence while providing fast outcomes compared to ML techniques that require training during the online algorithm. We adapt the algorithm to sample historical data while preserving the structural periodic information of our samples. In particular, we "retrieve" a prediction sequence on inflows for tank s for times $T = 1..n$ from the historical data sequence S of inflows for tank s including $\bar{d}$ days of data, by randomly selecting starting position t' which is the start of some day (since our experiments always commence from the first hour of a day) in S, where $S[l..u]$ returns the slice of sequence S from index l to u inclusive.

Algorithm 1 Historical sampling

historicalSample(S, n)
  $\bar{d} \leftarrow |S|$ div 24
  $t' \leftarrow 24 \times RANDOM([0, \bar{d} - 1]) + 1$
  return $S[t'..t' + n - 1]$

4 Experimental Evaluation

In this section, we present the results of using proposed models on a set of benchmark instances. The models are implemented in MiniZinc [8], a high-level and solver-independent modelling language, allowing for fast experimentation across existing solvers (OR-tools,



(a) (b)

Figure 2 (a) Original level data available from the SCADA system, and (b) the reconstructed level (from where we compute inflow data)

Gurobi, CPLEX) without compromising on efficiency. All experiments were run on the same machine which has an Apple M1 Pro 3.22 GHz CPU with 10 cores and 16 GB of RAM. All approaches were given a time limit of one hour per instance. The solver used was Gurobi version 9.5.2.

4.1 Benchmark Instances

The data is provided by our industry partner, South East Water, and corresponds to pressure sewer readings. This catchment has been selected as it is of reasonable scale and is free of infiltration. The data is collected, through a series of scripts, from the SCADA server and corresponds to 3 years (2019-2021) of historical readings from approximately 4200 individual households. A range of attributes can be extracted from the readings; we focus on the water level and pump activation. The network is free of intermediate pump stations.

In order to make different decisions, we first need to rollback any previous decisions to obtain the original system inputs. The tanks are not equipped with water meters to measure the inflows but have level sensors. From the water level, we can derive the inflows to be the difference between two positive consecutive readings. This is illustrated in Figures 2(a) and 2(b). We generated inflow data for each site for each hour of the day in the periods used for instance generation.

We cluster similar sites using the k-means algorithm where distance is defined as the sum of absolute differences over their inflow data. We observe the average inflow for each cluster. We determined four identifiable water usage profiles amongst the households: two diurnal/bimodal (with a morning and afternoon peak) and two uni-modal usages (with just a a morning peak), at a time translation respectively. We combined the bimodal and uni-modal sites respectively. The average inflow for each cluster is shown in Figure 4 in the supplementary material.

We created 6 problem instances from our industry partner. These instances represent different levels of complexity. To ensure they are different we choose different kinds of flows. For each instance we pick $|S|$ sites to use, either from the unimodal clusters, the bimodal clusters, or the complete set of clusters. We choose a number of hours n to solve over and a uniform capacity C for each tank. For each instance we create 30 scenarios, corresponding to different date ranges for the actual flow, and generate different historical sampling predictions for each site in each of the scenarios. Details of the problem instances are shown in Table 2. We consider scenarios of 24 and 48 hours length, the 48 hour instances allow water to be kept overnight in the tanks in order to smooth the outflows. We also briefly explored longer scenarios of 1 week but they did not lead to much greater peak outflow reductions than 48 hour scenarios.

Table 2 Statistics of the 6 difference problem instances giving: the kinds water usages: unimodal (only using tanks that have a single peak in usage), bimodal (only using tanks that have bimodal water usage) and complete (using all types of tanks); number of tanks $|S|$; number of periods n; and the peak capacity of each tank C.

Inst.	Types of usage	$\lvert S\rvert$	n	C
***I*1**	unimodal	1250	24	500
***I*2**	unimodal	1250	48	500
***I*3**	bimodal	1250	24	500
***I*4**	bimodal	1250	48	500
***I*5**	complete	2500	24	500
***I*6**	complete	2500	48	500

4.2 Alternative approaches to controlling outflow

Because we only have a prediction of the future, the decisions made by the optimisation models may not be implementable with the actual inflows. Thus we implement out decisions as a "policy" for the tank to follow. If $X_{s,t} = 1$ then tank s will empty *only if* there is sufficient volume in the tank ($\geq m$), and if $X_{s,t} = 0$ then the tank will *still empty* if the level would reach capacity c_s. This means that the decisions always lead to operation of the tank within specification. We denote this approach as HS (historical sampling).

The current operational approach and baseline is that a tank s only empties when it reaches capacity. We can understand this as a set of decisions where $X_{s,t} = 0, \forall s \in S, t \in T$, since emptying only happens when capacity is reached. We denote this policy $[0, 0]$.

An alternate baseline strategy is to set $X_{s,t} = 1, \forall s \in S, t \in T$, this guarantees to empty each tank as soon as it has more than the minimal capacity. While unattractive in practice, since it induces significant wear and tear on the pumps, this may reduce peak outflows. We denote this policy $[1, 1]$.

We also consider a random policy by counting the average proportion of pump out periods $prop_s$ for each site s using the baseline $[0, 0]$ policy. We draw a random number in $[0..1)$ for each site, and try to pump out if it is below $prop_s$. The random policy is defined as $X_{s,t} = 1 \rightarrow RANDOM([0..1)) \leq prop_s, \forall s \in S, t \in T$.

Only using a single historical scenario is not robust, although since we are sampling for many tanks, certainly some of the variance of the problem is considered. We can make more robust plans by sampling for each tank s instead k scenarios, and computing the plan that leads to the minimum average maximum outflow across the scenarios (the deterministic equivalent). But the models are already slow, and this would substantially increases solve time. Instead we construct an artificial worst case scenario w, by in each time period defining the $i^w_{s,t} = \max\{i^i_{s,t} \mid i \in 1..k\}$, that is, the maximum inflow over all the k scenarios. This has the same size, and hence solving difficulty, as a single scenario. We denote this approach as WHS (worst case historical sampling) where we use $k = 7$.

4.3 Results

The direct formulation while asymptotically smaller, $O(|S||T|))$, is challenging to solve for the size of problem we tackle. On instances with 2500 sites it struggles to find solutions quickly. The packing formulation which is ostensibly $O(|S||T|^2)$ but since $latest(s, t) - t$ is either small, or there are many time periods where $i_{s,t} = 0$ which do not require any pumped out decision variables, the actual size grows as $O(|S||T|))$, and the constraints are much simpler and hence faster to solve. For the remainder of the paper, we only report results for

Table 3 (Maximum mean outflow/standard deviation of outflow) for each problem instance and method across 30 scenarios.

	[0,0]	[1,1]	random	clairvoyant	HS	WHS
I1	(13,793/223)	(14,444/170)	(10,858/204)	(8,596/169)	(**8,998**/**170**)	(9,514/173)
I2	(13,793/223)	(14,444/170)	(11,408/209)	(8,110/206)	(8,702/205)	(**8,592**/**202**)
I3	(12,176/254)	(12,425/217)	(11,156/236)	(10,859/212)	(11,387/221)	(**10,877**/**213**)
I4	(12,176/254)	(12,425/217)	(11,042/239)	(9,717/272)	(**10,382**/241)	(10,865/**237**)
I5	(24,438/303)	(25,165/273)	(19,578/284)	(18,446/261)	(**18,793**/276)	(20,309/**270**)
I6	(24,438/303)	(25,165/273)	(19,225/285)	(17,035/281)	(**17,314**/290)	(17,368/**287**)

the packing formulation.

Our proposed methods are compared against the current operational strategy [0,0] and an alternate base line [1,1]. In order to see how close to optimal we get we also compare against the *clairvoyant* approach which is running the optimisation model with the actual inflow data for the tested time period. This computes the minimal maximum outflow possible.

Figure 3 shows the behavior of the models running on instance I5. The clairvoyant approach illustrates that there is a significant reduction in peak outflow available if we make wise emptying decisions. The historical sampling approach actually gets quite close to the best solution on average but it is clear that the variance is large, often well over the clairvoyant solution. The worst case approach pays some penalty, it is unable to reduce the peak flow as well, but still is not that far from the clairvoyant solution, and its standard deviation is much smaller.

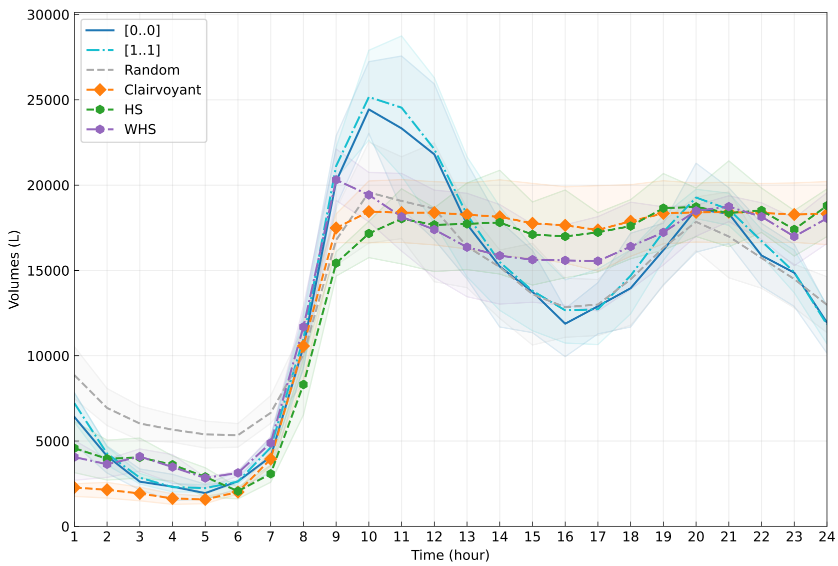


Figure 3 Mean total outflows in each hour of the day for different approaches applied to instance I5. The shaded regions show the 25% - 75% confidence interval, across the 30 scenarios.

Table 3 gives the summary results across the 6 instances. First note that the current baseline [0,0] is much better at reducing mean peak outflow then the alternative [1,1] of always pumping out when possible, but the standard deviation of the second method is much lower. The clairvoyant method shows that there is considerable reduction in peak available compared to the current baseline. The historical sampling optimisation approach

HS is able to capture much of the available reduction in peak outflow and while the standard deviation is larger than the clairvoyant approach it is not that much larger and considerably better than the current baseline. The worse case WHS approach also beats the baseline, and for some instances can be better than HS, its main strength is that usually reduces the standard deviation compared to HS. The random policy performs well for 24 period instances comparatively to 48 period instances. The HS and WHS approaches consistently perform better at reducing the peak flow when considering larger periods. The random policy is able to use the morning buffer capability of the tanks that is overlooked by the other approach for 24 period instances. For larger periods instances, the HS and WHS approaches systematically beat the random policy. This suggests a potential performance gain for CP based approach for larger period instances.

5 Conclusion

In this work, we introduced a novel practical problem from the water industry, controlling a pressure sewer system to reduce peak outflow. We proposed a direct MIP formulation and a packing formulation to solve practical scenarios. We provide an extensive experimental study on challenging and realistic instances of considerable size. The results show an optimisation model can significantly reduce the peak outflow of the system compared to the current operational approach.

So far we have only considered the most simple prediction approach, we plan to investigate forecasting models such as LSTM and LightGBM, to see whether they can produce realistic future flows. Ideally we would also extend the forecast to take into account parameters that affect the likely inflows, such as day of the week, season, and weather (the PSS we study is in a holiday zone, so inflow patterns change significantly on weekends, during summer, and when the weather is hot). It would be interesting to investigate Predict+Optimise approaches [6, 7] applied to the problem, but seeing that the predictions are for individual tanks and the objective results from considering all tanks simultaneously this appears challenging. While the simple optimisation approach we use here works we also plan to investigate decomposition approaches such as Benders or column generation, since the tanks are only weakly coupled by the objective. As future work, we plan to take into account more of the real features of the network such as the distance of tanks from the sewer treatment plant, the full topology of the network, and the inclusion of intermediate pump stations.

References

1. Russell Bent and Pascal Van Hentenryck. Regrets only! Online stochastic optimization under time constraints. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, page 501–506. AAAI Press, 2004.
2. Russell Bent and Pascal Van Hentenryck. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52:977–987, 2004. doi:10.1287/opre.1040.0124.
3. Russell Bent and Pascal Van Hentenryck. The value of consensus in online stochastic scheduling. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling, ICAPS 2004*, pages 219–226, 2004.
4. Russell Bent and Pascal Van Hentenryck. Online stochastic optimization without distributions. In *ICAPS 2005 - Proceedings of the 15th International Conference on Automated Planning and Scheduling*, pages 171–180, 2005.

5 Hyeong Soo Chang, Robert Givan, and Edwin K. P. Chong. On-line scheduling via sampling. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*, AIPS'00, page 62–71. AAAI Press, 2000.

6 Adam Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.

7 Jayanta Mandi, Emir Demirovic, Peter Stuckey, and Tias Guns. Smart predict-and-optimize for hard combinatorial optimization problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:1603–1610, 2020. doi:10.1609/aaai.v34i02.5521.

8 Nicholas Nethercote, Peter J. Stuckey, Ralph Becket, Sebastian Brand, Gregory J. Duck, and Guido Tack. Minizinc: Towards a standard CP modelling language. In Christian Bessière, editor, *Principles and Practice of Constraint Programming – CP 2007*, pages 529–543, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. doi:10.1007/978-3-540-74970-7_38.

A Clusters

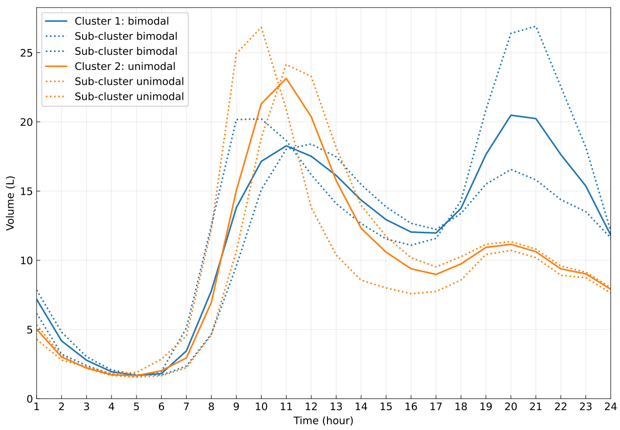


Figure 4 Inflows (averaged) for the two types of flow: unimodal and bimodal. Cluster 1 shows a diurnal water usage (morning and evening peak). Cluster 2 shows a unique morning peak. Each consists of 2 sub clusters where the peaks are time shifted.