Optimisation of a fuzzy logic-based local real-time control system for mitigation of sewer flooding using genetic algorithms

S. R. Mounce, W. Shepherd, S. Ostojin, M. Abdel-Aal, A. N. A. Schellart, J. D. Shucksmith and S. J. Tait

ABSTRACT

Urban flooding damages properties, causes economic losses and can seriously threaten public health. An innovative, fuzzy logic (FL)-based, local autonomous real-time control (RTC) approach for mitigating this hazard utilising the existing spare capacity in urban drainage networks has been developed. The default parameters for the control algorithm, which uses water level-based data, were derived based on domain expert knowledge and optimised by linking the control algorithm programmatically to a hydrodynamic sewer network model. This paper describes a novel genetic algorithm (GA) optimisation of the FL membership functions (MFs) for the developed control algorithm. In order to provide the GA with strong training and test scenarios, the compiled rainfall time series based on recorded rainfall and incorporating multiple events were used in the optimisation. Both decimal and integer GA optimisations were carried out. The integer optimisation was shown to perform better on unseen events than the decimal version with considerably reduced computational run time. The optimised FL MFs result in an average 25% decrease in the flood volume compared to those selected by experts for unseen rainfall events. This distributed, autonomous control using GA optimisation offers significant benefits over traditional RTC approaches for flood risk management.

Key words | fuzzy logic, genetic algorithm, real-time control, sewer flooding, urban flood risk

S. R. Mounce (corresponding author)
W. Shepherd
M. Abdel-Aal
A. N. A. Schellart
J. D. Shucksmith
S. J. Tait
Pennine Water Group, Department of Civil and Structural Engineering,
University of Sheffield,
Sheffield S1 3JD,
UK
E-mail: *s.r.mounce@sheffield.ac.uk*

S. Ostojin
Environmental Monitoring Solutions,
Unit 7, President Buildings, Savile Street East, Sheffield S4 7UQ,
UK

INTRODUCTION

Climate change, demographics and economic change all cause considerable impact on the performance of urban drainage networks. Climate change is generally accepted to influence future rainfall patterns and is expected to increase the occurrence of extreme rainfall events (IPCC 2014). However, wide regional variations are predicted for the UK (Sanderson 2010). Currently, climate change models work on a larger space and time scales than urban drainage systems. Gooré Bi *et al.* (2017) provide an overview of the recent work on downscaling the predictions of rainfall data for climate change impact studies in urban areas. The validity of many downscaling methods has, however, not yet been tested in urban areas, and there is little published information available on their effectiveness to predict rainfall–runoff at smaller urban (1–10 km^2) scales. Hence, any solutions to deal with future urban drainage flooding will need to be intrinsically flexible and adaptable and cannot rely on rainfall and runoff predictions at small urban scales.

Wastewater service providers are under renewed pressure to improve their sewer network performance through innovation and asset optimisation. One such area for performance improvement is urban flooding. Currently, the vast majority of sewers and piped drainage networks are passive systems, with operators having little or no

doi: 10.2166/hydro.2019.058

control of the system during rainfall and any resulting pollution and/or flooding events. The size, complexity and varied elevation of sewer networks mean that local urban flooding events can occur while there is still the significant storage capacity available within the sewer network.

Real-time control (RTC) systems have been utilised to regulate the stormwater flow and hence reduce the urban flood risk. Pleau *et al.* (2001) and Fuchs & Beeneken (2005) have described RTC system deployments using large centralised control systems. However, the drivers for these systems tend to be focused on solving existing system-wide water quality problems caused by overflows rather than local flooding issues. The RTC of sewer systems has often developed slowly apart from large cities in economically advanced countries (see, e.g. Schutze *et al.* 2005). Latest approaches have proposed the more distributed RTC of urban drainage systems to locally manage flooding and overflow (Garofalo *et al.* 2017).

Local, autonomous RTC systems have the potential to optimise the performance of urban drainage networks at relatively low cost through retrofitting. The RTC algorithms can adapt to different climatic conditions, and there is the potential for the hardware to be relocated as circumstances change. Abdel-Aal *et al.* (2017) describe a novel system using intelligent fuzzy logic (FL)-based RTC which has been designed and developed to take advantage of the local unused storage capacity that is present in the upper parts of many sewerage networks, thus attenuating the flow at flood-threatened downstream locations.

This paper presents and tests a new genetic algorithm (GA)-based methodology for optimisation of the local RTC system, described by Abdel-Aal *et al.* (2017). Rather than using design rainfall events, the compiled rainfall time series based on recorded rainfall and incorporating multiple events were used as an input to the hydrodynamic model which is used in the optimisation.

BACKGROUND

The RTC of sewer networks utilises current and predicted states of flows and wastewater levels in a network to adjust a control strategy (Vanrolleghem *et al.* 2005; Vezzaro & Grum 2014). In such a way, it is possible to redirect flows in order to, for example, reduce combined sewer overflow (CSO) discharge loads (environmental impact) by actively controlling the system (e.g. using gates and pumps). RTC offers benefits for system operators, the environment and for customers, subject to charges (Beeneken *et al.* 2013). Villeneuve *et al.* (2000) describe the essential components of an RTC system as sensors, automated gates and some form of strategy. Campisano *et al.* (2013) give a comprehensive review of the field. Lund *et al.* (2018) give a state-of-the-art review in the context of smart real-time water management.

Highly complex, centralised RTC strategies are vulnerable to sensor or network failure, and they also come with large investment costs (and fixed infrastructure); hence, they are less adaptable in the future, if the local climate does not behave as predicted (Radhakrishnan *et al.* 2018). Along with limited robustness and capacity for graceful degradation in performance, there is also a significant cost involved in the maintenance and in the expensive provision of several layers of fail-safe mechanisms necessary within such RTC systems (Frier *et al.* 2014).

Much of the recent RTC research for sewer systems is focused on model-based optimisation techniques, which are used at the higher layers in the control hierarchy (Mollerup *et al.* 2016). Experiences from related fields (such as the plantwide control of wastewater treatment plants) suggest that it is preferable for system operators to focus on lower layer controllers to obtain high resilience (Mollerup *et al.* 2016). This approach favours decentralised solutions which feature fail-safe abilities, ensuring that system performance is not compromised in the event of sensor/power/communication failure. Decentralised control promotes robustness and decreases maintenance costs. García *et al.* (2015) reviewed modelling and the RTC of urban drainage systems and the potential of rule-based and fuzzy control in this context. Most ‘simple’ control theories (such as PID systems), routinely used in industrial plant control applications, assume that the behaviour of the system is linear and transfer functions are known and fixed. In contrast, fuzzy control systems do not require either of those two assumptions and have been shown to provide superior performance in some control plant applications (Natsheh & Buragga 2010). Urban drainage networks are dynamic systems where the inputs are variable and non-linear; hence, FL control approaches are more applicable.

Work on providing the active control of the flow pattern in urban drainage systems has been progressed in recent years, with the aim of utilising infrastructure in a more intelligent way. This work is now moving from academia into practical applications. Seggelke *et al.* (2013) presented a practical implementation of RTC for the combined sewer system in the German coastal city of Wilhelmshaven. Linking together both sewer flow modelling and wastewater treatment modelling enabled the derivation of a set of IF-THEN fuzzy rules for integrated control. Kroll *et al.* (2018) also presented an automated design of RTC strategies for combined sewer systems and with implementation and testing on five case studies. Pumping station optimisation, for on/off states, has been explored within the sewer system described as a directed graph and then attempting an optimal set of on/off states over a set of pumps in order to minimise environmental damage caused by CSO events during storm conditions (van Nooijen & Kolechkina 2013).

The potential for data-driven approaches within urban drainage system management is receiving increased interest, as the practical constraints and scientific challenges of using conventional hydrodynamic modelling approaches become more apparent (Solomatine & Ostfeld 2008). Data requirements for hydrodynamic model calibration impose a significant cost burden on urban drainage system operators, and significant uncertainties remain even after model verification (Sriwastava *et al.* 2018). Techniques for the time-series analysis of urban drainage data (Branisavljević *et al.* 2010) and for autocalibration of urban storm water runoff models using multiple rainfall events within the network employing GAs (Pierro *et al.* 2006; Barco *et al.* 2008) have sought to address such issues. Data-driven approaches are especially suited to local RTC applications as, once the control systems are trained or optimised, these systems can be autonomous rather than be managed from a central control site. This approach removes a significant proportion of the cost and complexity of traditional RTC approaches and allows a faster response to network conditions.

In this work, an RTC system makes use of frequently measured water levels to control a gate by using fuzzy logic (FL), and the objective is to minimise local flooding by maximising the use of storage in the network during rainfall events. The use of local flow level sensing avoids the need for spatial and temporal characterisation of rainfall via an expensive real-time radar or rain gauge systems and the large communication and computational resources required for rainfall–runoff and hydrodynamic network models needed for large centralised RTC solutions. After initial development and testing in a full-scale laboratory environment (Abdel-Aal *et al.* 2017), complete pilot systems have been manufactured and installed in Coimbra, Portugal and Toulouse, France to conduct a field testing programme (although this paper does not use these sites).

The performance of any FL-based control system is a function of the rules and membership functions (MFs) which capture expert knowledge for the system operation. It may, however, be beneficial to carry out further optimisation of the control algorithm parameters, in particular, the MFs. A genetic algorithm (GA) is one suitable technique for such optimisation. GAs are highly parallel, mathematical algorithms based on the mechanics of natural selection and genetics that transform a set (population) of mathematical objects (typically strings of ones and zeros in the form of genes) into a new population over multiple generations. GAs and other evolutionary algorithms have successfully been applied to many complex engineering optimisation problems. They have been shown to provide highly effective solutions for hydrological applications (Nicklow *et al.* 2010).

The initial work to develop a control algorithm for an RTC gate defined a rule base and manually set the vertices of the membership functions (MFs) (which were predetermined in shape and number) for the FL controller. This control algorithm was first tested in a modelling study (Shepherd *et al.* 2016) and subsequently in a laboratory system (Abdel-Aal *et al.* 2017). An initial pilot study of GA optimisation was explored which would allow the RTC MFs to be automatically tuned for sites with different characteristics or enable it to be re-tuned in case of changes within a sewer network or in the prevailing climate (Shepherd *et al.* 2017). This pilot study used a design rainfall event, with a return period of 5 years (20% annual exceedance probability) and the duration of 120 min (Reed *et al.* 1999) on a test network. This event was selected for the GA optimisation because it resulted in a total flood volume larger than the available storage, thus ensuring that the objective function had a suitable target. The optimisation resulted in a flood volume reduction of between 2 and 25% when compared to the flood volumes resulting from the unoptimised expert MFs.

In the present work, the GA optimisation approach is developed significantly, including the application of both integer (Deep *et al.* 2009) and decimal input vector optimisation (to explore issues of speed and generalisation), and with metrics describing gate movements. Furthermore, results of optimising on the combined time series of real rainfall events are presented and performance on completely unseen events assessed. The combined rainfall time series were used as these are a significant improvement in the use of a single design event because they provide a range of different and realistic characteristics, such as the rate of rise in the flow, and also provide the environment for the storage volume to be emptied between each event. The parameter space being optimised is also explored via a Monte Carlo exposition to give some understanding of the solution space.

METHODS

The RTC system considered in this work aims to prevent or minimise flooding at a downstream location by autonomously closing a gate in order to mobilise upstream storage. Water-level data are input into the FL control algorithm, the outputs of which are processed to determine the timing and magnitude of the gate movements. A GA is utilised to optimise the FL MFs with the objective of minimising flood volumes. The GA interfaces with an SWMM hydrodynamic sewerage network model to calculate the flood volumes. Figure 1 provides a summary flow chart of this methodology and its components.

Sewer network modelling

Urban catchments are physically complex systems, and mathematical models describe the rainfall–runoff and in-pipe hydraulic processes, hence incorporating various features of a hydrological and hydrodynamic simulation albeit under a certain level of simplification. In this study, a small UK catchment of combined sewers was used which included a known flood location and excess in-pipe capacity that could be used to store the flow volume and hence reduce the flood risk. The network upstream of the flooding location drains an area of just under 39 ha, 17 ha of which is classed as impermeable. The network is modelled in the hydrodynamic sewer network software SWMM (Rossman 2015), utilising a network of subcatchments, nodes and links in order to route dry weather wastewater flow rainfall–runoff through the combined sewer network. The area upstream of the flooding location

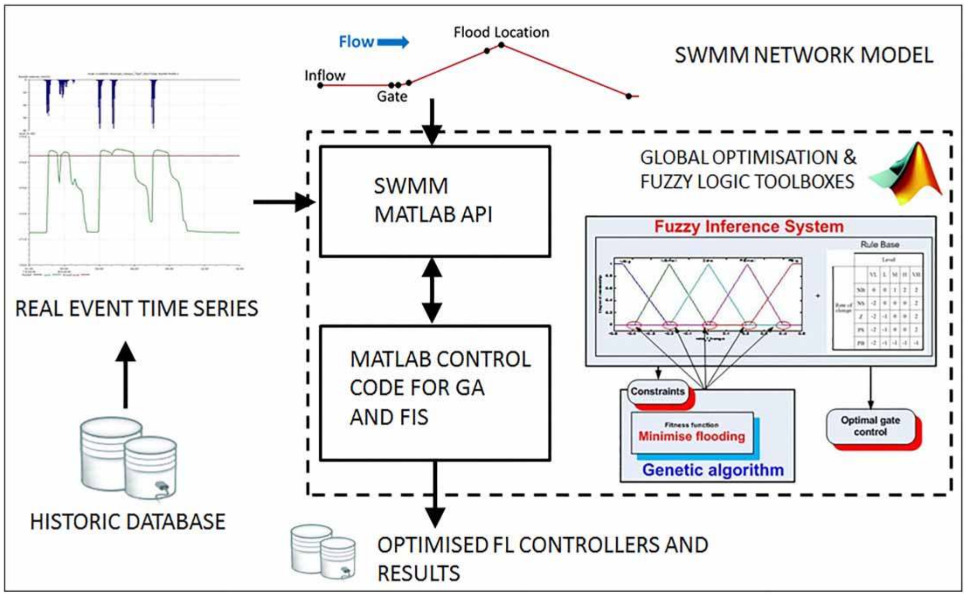

Figure 1 | Flow chart of methodology.

includes 31 subcatchments and 2.6 km of pipes with diameters in the range of 225–1,550 mm (mean 384 mm) and slopes in the range of 0.002–0.144 m/m (mean 0.053 m/m). In order to minimise model run times and hence the GA run time, the network was simplified, as shown in Figure 2, where dotted lines represent the parts of the network which have been removed. The simplified network has a total conduit length of 438 m and uses inflow hydrographs from previous model runs to represent the upstream catchments. The locations where the network has been cut were carefully selected to ensure that results in the simplified network did not differ from the full model.

Riaño Briceño *et al.* (2016) have developed an interface (API) 'MatSWMM' (https://github.com/water-systems/MatSWMM) which allows a SWMM simulation to be started and controlled programmatically. MatSWMM was therefore used to allow the MATLAB-based FL control to be applied. In order to minimise run times, the original model was cut at appropriate locations both the upstream of the storage location and the downstream of the flooding location. Flows from the upstream subcatchments were generated in the original model and saved as time-series data to be used as an inflow in the MatSWMM simulations.

Rather than using design rainfall events consisting of a single symmetrical rainfall profile, the compiled rainfall time series based on recorded rainfall for the catchment and incorporating multiple (non-conservative) events were used. The complete recorded time series were not used because they would be too long to use for the large number of simulations required by the GA optimisation process. These time series were created by manually assembling recorded rainfall data to form a number of discrete events. Each of these discrete events was large enough to cause flood volumes in the SWMM model that would not be easy to control, e.g. the flood volume was greater than the available hydraulic capacity. Multiple events were included in each time series to ensure that the optimisation target required the FL controller to reopen the gate quickly to be ready for any future rainfall which could also cause flooding. It was decided to base the time series on recorded, rather than design, rainfall because recorded rainfall exhibits a more natural variation in intensities and thus results in a range of rates of change in the runoff flow that would not be seen if the single design event was to be used. The discrete events in the time series were separated by a suitable minimum dry period, and this period was determined by running the simulations without control and using the expert hydraulic modelling judgment to estimate a short but realistic time for the storage to be emptied. The judgment for the separation of discrete events in the time series was based on the uncontrolled system being able to drain down between events, so the emptying time was a function of both the total flood volume and the shape of the recession limb.

Two of these rainfall time series were assembled, one for training (time series 1) and one for unseen testing (time series 2). Statistics of these compiled rainfall time series are presented in Table 1. The test catchment has an approximate time of concentration of 15 min, and the peak return period (Reed *et al.* 1999) for a 15-min duration is 5.4 years for time series 1 and 7.0 years for time series 2, while the return periods for the complete time series are 70.5 and 29.1 years, respectively. A SWMM model run of 18 h was required for each.

The water levels and calculated derivatives, used as an input to the FL during the GA optimisation, are generated by the SWMM hydrodynamic model. The SWMM simulation is run with a variable routing time step between 0.1 and 2 s. After each simulation time step, MatSWMM returns control to MATLAB, the FL is run at predetermined steps of 1, 2 or 5 min, if the FL is not run, results are stored in MATLAB and control returns to the SWMM. These FL

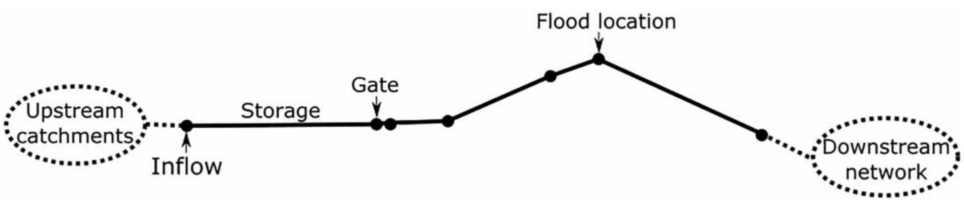

Figure 2 | Topology of the simplified urban drainage network model used in the GA.

Table 1 | Rainfall event datasets and baseline results for default FL input MFs

Name	Rainfall event duration (h)	Total rainfall (mm)	Peak intensity (mm/h)	Flood volume (m^3)				Default FL gate metrics number/total distance (m)		
				Gate inactive	Default FL 1 min	Default FL 2 min	Default FL 5 min	1 min	2 min	5 min
Time series 1	10.4	76	135	170.6	73.7	77.7	210.2	268/3.52	213/4.58	80/3.47
Time series 2	9.9	60	84	128.2	64.7	44.7	68.3	240/3.04	183/4.14	118/5.15
Design event (M5-120)	2	23	44	247.0	111.0	113.4	141.2	75/1.08	53/1.33	31/1.37

time step frequencies were selected to represent the likely periods to be used in real-world field deployments; shorter FL steps will provide a finer control of the gate, as long as the step is not too short to allow impacts of changes to be seen at the monitoring location, whereas longer FL steps reduce the number of communications to the base station and hence improve the battery life for the communications system. The FL outputs a target gate position, and MATLAB instructs the SWMM to move the gate to this position at a rate of 3.75 mm/s and the SWMM simulation resumes. This rate of gate movement was determined following a discussion with a manufacturer. The flow through the gate is computed by the SWMM, based on the area of its opening, its discharge coefficient and the head difference across the orifice. This 'virtual testing' modelling methodology and the sewer network used are described in more detail in Shepherd *et al.* (2016). Total gate movements and total distance moved by the gate were calculated to allow the additional assessment of a particular FL controller for any particular event. These gate metrics allow some measures of efficiency, in terms of wear on the gate and electricity usage to be calculated. The motor powering the gate will also have a duty cycle, so excessive gate movements may exceed the duty cycle of the motor. Note that although not part of the objective function, these are instructive as to how the controller is operating.

FL controller

The FL control algorithm uses water-level data provided by a local sensing network as input data, the FL rules implement expert knowledge, and the output adjusts the setting of the flow control gate. Level data are recorded at the downstream flood location and also upstream of the gate. The algorithm uses four sets of input data (level and calculated level derivatives at two locations), each has three MFs with triangular or trapezoidal shapes defined by 11 vertices in total (see Figure 3), seven of which are optimised by the GA. The default shapes of MFs were predetermined based on expert knowledge (an experienced hydraulic modeller with knowledge of multiple networks) since the goal is a technically feasible and sound solution. For example, the

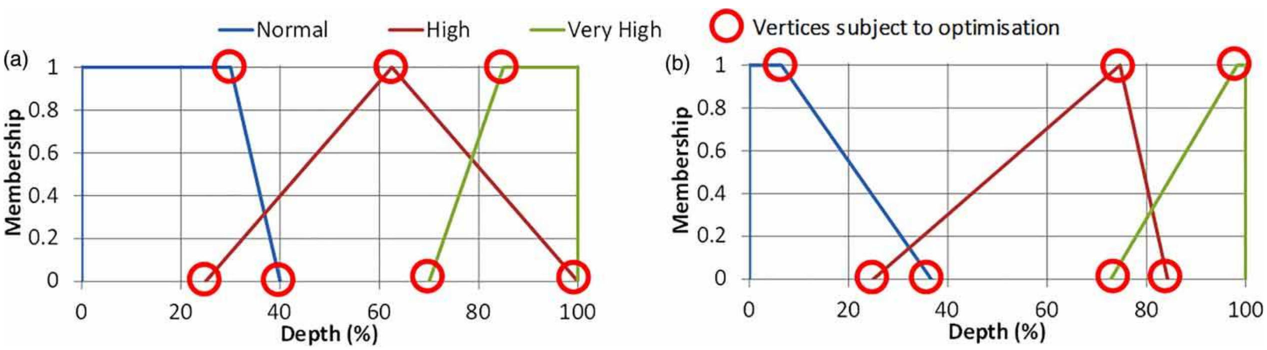

Figure 3 | Example input MFs for the level at the flood location. (a) Default and (b) optimised.

MFs for one of the level data inputs are Normal (N), High (H) or Very High (VH), and these MF labels give a textual description, e.g. VH represents a water-level range which may start from the pipe full flow and up to the onset of flooding. The output variable, change position (CP), has five MF labels, corresponding to changes in the gate position as follows: Small Open (SO), Big Open (BO), Small Close (SC), Big Close (BC) or Zero Change (Z). This output is used to adjust the gate by a given percentage which is a function of the FL step.

The FL rules (rule base) are expressed in the form of IF-THEN fuzzy rules written using expert knowledge (of the sewer network operation). This expert knowledge takes into account the expected response of drainage networks to the impact of rainfall events along with the understanding of when the gate should be activating – e.g. 'If rate of change at the gate is negative, and if level at the gate is High, and if rate of change at the flood location is Zero, and if level at the flood location is Normal, then control change is Big Open'. The FL algorithm was developed with the MATLAB FL toolbox, and it uses the Mamdani approach (Mamdani & Assilian 1975) and after defuzzification it provides a final output value for CP.

GA optimisation

The GA optimisation of the FL input MFs has been accomplished through a MATLAB script using the Global Optimisation toolbox. This allows the iterative running of a parameterised function of MFs (defined by the input vector) in the setting of gate positions and the calculation of the resulting flood volumes. MF positions were chosen as the decision variable for the GA, and as during manual tuning of the FL algorithm, it was found to be sensitive to changes in locations of the input MF vertices. After modifying the input vector, for each candidate solution at each iteration, the SWMM was used to calculate the flood volume for the input compiled rainfall time series. The objective of the GA is to minimise the flood volume by determining the optimum locations of the vertices of the MFs' relevant edges for the four sets of input data. The vertices which are optimised are highlighted for one of the level MFs in Figure 3, each input dataset has seven MF vertices to be optimised; hence, a total of 28 values are optimised. The objective function indicating the fitness of the parameter set is the flood volume from the target node for the input time series. Using a truncated network model prevents an assessment of flooding impacts in the whole catchment; however, in practice, any additional flooding upstream of the gate is prevented by the use of an overtopping weir. This weir allows flows to overtop the gate and thus limit upstream surcharge once the storage is full. Two versions of the GA were implemented, one integer only and one decimal in order to explore issues of speed and generalisation.

Three methodologies for initialisation of the GA starting point had been previously compared (Shepherd *et al.* 2017). These initialisations are (1) the default expert configuration, (2) a randomised configuration and (3) a pseudo-randomised configuration, where the default expert values have small perturbations applied. Figure 4 shows a flow diagram of the GA optimisation module for a randomised or pseudo-randomised start point. The starting point of the main GA optimisation for the randomised and pseudo-randomised sets is selected from the results of 10 generation mini-runs (population size 5), each starting from a different randomised or pseudo-randomised configuration (this latter involving small perturbations of the default expert starting points). Prior to running the FL, the randomised/pseudo-randomised values are first pre-sorted to maintain the MF shape and crossover (structure informed by the expert design). Appropriate lower/upper bounds, linear inequality constraints and tolerance checks are conducted in the next stage. For running the optimisation, the seeding of the random number generator needs consideration especially for repeatability. For every combination of the FL time step (1, 2 and 5 min) and initialisation methodology, three runs were conducted: two randomised (termed Shuffles 1 and 2) and one using the in-built MATLAB default (Mersenne Twister with seed 0). The GA stop criteria were based on having no improvement in the objective function for a number of generations (stall limit) or until the maximum number of generations is reached (see Supplementary Figure 1 for an example of results from each generation of this optimisation process). The GA parameters were set based on extensive empirical trials. For the decimal version, 25 generations with a population size of 200 were used with a stall limit of 5 generations. For

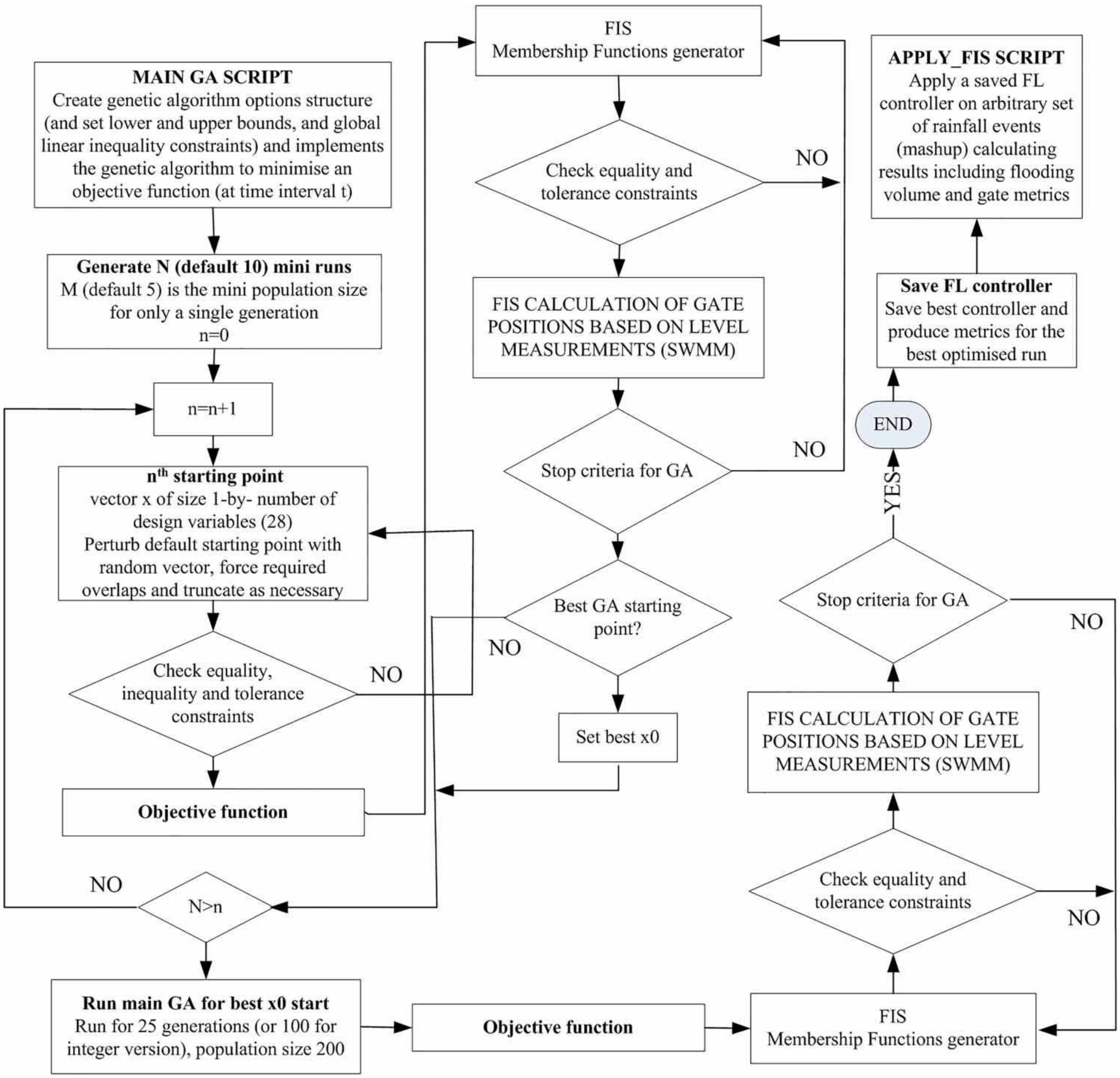

Figure 4 | Flow diagram for the GA optimisation module.

the integer version, 100 generations with a population size of 200 and a stall limit of 20 generations were used. A full simulation is run for each population member. These values were empirically discovered to be a useful compromise between run times and an improvement in the objective function. 'Arithmetic' (creating children that are the weighted arithmetic mean of two parents) and 'Adapt feasible' (randomly generating directions that are adaptive with respect to the last successful or unsuccessful generation and satisfying bounds and linear constraints) were selected, respectively, as the crossover and mutation functions for the decimal optimisation. The integer implementation uses special functions to enforce variables to be integers as described by Deep *et al.* (2009).

RESULTS AND DISCUSSION

The results of experiments using the assembled rainfall time series (based on recorded rainfall data) to explore the ability of the GA to improve the performance of the FL RTC system are now presented, along with some comparisons to the design rainfall used in Shepherd *et al.* (2017). The aim of the experiments is to optimise the FL MFs for a local RTC system installed in a particular sewer network but to avoid overfitting to the training events, i.e. a lack of generalisation, which is a common issue and the area of active research in the GA and the genetic programming field (Gonçalves & Silva 2011).

Optimisation and testing

Table 1 provides some overview statistics of the rainfall and baseline results for flooding without FL control (gate inactive) and for the non-optimised default FL for the two rainfall time series and the design event used in Shepherd *et al.* (2017). It can be seen that, when compared to the 'gate inactive' case, the default FL reduces flood volumes for all cases, except the 5-min FL time step for time series 1. It is interesting to note that in general, the shorter FL time steps result in lower flood volumes and also a smaller total distance moved by the gate during the time series. The exceptions are the 5-min FL time step gate movement for time series 1 which is the lowest and the 2-min FL step for time series 2 which has a significantly lower flood volume than either 1 min or 5 min. The former is due to the gate tending to stay in a closed position more of the time, hence smaller gate movements, but the flood volume is large. For the 2-min FL step, the gate movements are allowing the stored volume to drain more quickly and hence there is more storage available for the later runoff.

Table 2 presents results for the integer optimisation, and the same data for the decimal optimisation are included in Supplementary Table 1. Time series 1 is the training set, and time series 2 is the independent test set. In Table 2, the column header abbreviations are as follows: RNG is the random number generator used for seeding, Init. is the GA initialisation strategy, Stall is the number of generations after which the GA stalls, Gate # is the number of gate movements and Gate dist. is the total distance the gate moves. It can be seen that within the results for a single FL time step, the RNG and initialisation strategy impact both the flood volume and gate metrics, showing that in this relatively complex solution space, there are many local minima. For example, the 1-min FL time step training flood volumes vary from 23.5 to 48.2 m^3, with an average of 31.4 m^3, while the number of gate movements and the total distance moved vary by similar multiples, but there is no correlation between the gate movements or distance moved and the flood volume. In all cases, the integer optimisation stalled due to not improving the solution outside the tolerance for 20 generations.

Table 3 presents a summary of the overall results, including the mean percentage reduction in flood volumes from the default FL and no control option; results are presented for each FL time step; and the average of all time steps for both integer and decimal optimisation is reported. In Table 3, we see that the decimal optimisation provides the best reduction in the flood volume on the training set (mean of 59.7% flood volume reduction across all runs and time steps compared to 43.3% for the integer optimisation). However, the integer version provides better generalisation on the unseen time series (26.7% reduction in the flood volume compared to 23.2% for the decimal optimisation). Both integer and decimal GAs achieve the best flood volume reduction for the unseen time series on a 1-min FL time step and perform least well on the unseen time series for a 2-min time step. The decimal version gives the highest performance for the training dataset for a 5-min FL time step, although referring to Table 1, the default FL performed badly, resulting in flood volumes greater than if the gate was inactive; hence, there was the greatest scope for improvement. Overall, the integer version has the advantage of less overfitting and much faster run time. On a standard desktop PC (Intel® Core™ i7-3770 CPU, 32 GB RAM, Microsoft Windows 10) running MATLAB R2016b, the run time of the integer version is approximately 2.75 h compared to 9.75 h of an equivalent decimal run – i.e. around a 70% reduction.

The M5-120 design event, as used in Shepherd *et al.* (2017), was not expected to be representative as a training set, and this was the case when this was explored for a

Table 2 | Integer optimisation

				Training results			Test results		
FL time step (min)	**RNG**	**Init.**	**Stall**	**Flood volume (m^3)**	**Gate #**	**Gate dist. (m)**	**Flood volume (m^3)**	**Gate #**	**Gate dist. (m)**
1	Default	PseudoRandom	85	30.66	470	7.59	32.4	438	7.05
1	Default	Random	71	28.68	564	8.66	23.82	466	7.3
1	Default	Default	90	28.62	575	8.96	25.14	492	7.76
1	Shuffle 1	PseudoRandom	45	41.46	341	4.43	28.98	319	4.3
1	Shuffle 1	Random	79	23.7	476	7.89	21	373	5.9
1	Shuffle 1	Default	70	27.42	362	5.32	40.92	356	4.92
1	Shuffle 2	PseudoRandom	40	48.24	281	3.93	30.18	209	2.97
1	Shuffle 2	Random	66	23.52	391	6.17	24.12	329	5.27
1	Shuffle 2	Default	50	30.06	461	7	25.98	409	6.17
2	Default	PseudoRandom	62	43.38	286	5.99	38.22	224	4.39
2	Default	Random	21	64.8	245	6.17	44.22	199	5.28
2	Default	Default	77	51.3	212	5.45	38.34	141	3.48
2	Shuffle 1	PseudoRandom	38	37.56	205	4.81	39.36	160	3.75
2	Shuffle 1	Random	34	45.72	248	6.63	57.54	183	4.98
2	Shuffle 1	Default	80	52.38	238	5.26	43.98	197	4.09
2	Shuffle 2	PseudoRandom	45	44.04	251	5.93	32.22	172	3.85
2	Shuffle 2	Random	44	45.6	283	7.5	44.22	218	5.51
2	Shuffle 2	Default	71	46.74	226	5.58	43.44	185	4.35
5	Default	PseudoRandom	46	82.44	106	5.32	52.56	87	4.36
5	Default	Random	41	119.4	115	5.52	74.16	116	6.03
5	Default	Default	21	210.12	81	3.54	48.42	129	5.17
5	Shuffle 1	PseudoRandom	86	75.06	131	5.88	43.92	112	5.21
5	Shuffle 1	Random	50	79.98	133	6.39	72.78	110	5.1
5	Shuffle 1	Default	63	154.92	101	5.46	59.16	87	4.88
5	Shuffle 2	PseudoRandom	60	178.38	105	6.31	65.16	117	7.69
5	Shuffle 2	Random	21	200.58	97	4.05	52.2	115	4.61
5	Shuffle 2	Default	88	146.46	87	4.43	34.98	108	4.8

1-min FL time step. The subsequent optimised controller resulted in higher flood volumes than the default FL when both time series 1 and 2 were used as tests. The M5-120 event has also been used as a test event, and the integer optimisation provides the best reduction in the flood volume for the design event with a mean reduction of 4.3% across all runs and time steps, with the decimal optimisation providing only 2.7% reduction.

Table 4 summarises the gate movements across the optimisation type and time step (see Table 1 for the metrics for the default FL controller before optimisation). A general observation concludes that there are a greater number of gate movements for the smaller time steps, which is to be expected, although this is not reflected strongly in the total distance gates moved because the distance moved in each step is a function of the FL time step. Time series 1, the training event, tends to have a greater number of gate movements and greater distance moved, and this could indicate overfitting, but may also be a reflection of the differences in the total rainfall depth and the periods in the time during the event when rain is falling.

Table 3 | Summary results (flood reduction)

		Training results (time series 1)		Unseen test results (time series 2)	
FL time step (min)	GA type	% mean flood reduction from default FL	% mean flood reduction from no gate control	% mean flood reduction from default FL	% mean flood reduction from no gate control
1	Integer	57.4	81.6	56.6	78.1
2	Integer	38.4	71.9	5.1	66.9
5	Integer	34.4	19.1	18.2	56.4
All	Integer	43.3	57.4	26.7	67.2
1	Decimal	60.9	83.1	54.4	77.0
2	Decimal	53.9	79.0	1.6	65.7
5	Decimal	64.7	56.5	9.4	51.7
All	Decimal	59.7	72.9	23.2	65.5

Table 4 | Summary results (gate metrics)

		Training results (time series 1)		Test results (time series 2)	
FL time step (min)	GA type	Mean gate #	Mean gate dist. (m)	Mean gate #	Mean gate dist. (m)
1	Integer	435.7	6.66	376.8	5.74
2	Integer	243.8	5.92	186.6	4.41
5	Integer	106.2	5.21	109.0	5.32
All	Integer	261.9	5.93	224.1	5.15
1	Decimal	363.3	4.93	311.4	4.18
2	Decimal	247.3	5.82	195.1	4.61
5	Decimal	129.2	6.00	111.0	5.21
All	Decimal	246.6	5.58	205.9	4.67

Monte Carlo exploration of the problem space

A GA searches the solution space using an evolutionary approach whose settings and parameters generally set a fine balance between exploration and exploitation. Excessive exploration might waste time on solutions that are less likely to perform well in light of evolution already conducted. Excessive exploitation will result in becoming trapped in local maxima. A Monte Carlo exposition to look at a large range of parameter values was conducted on the 28 MF variables (corresponding to the GA input vector). A sort ensures that MF crossover and basic sense checks are carried out as for the GA. 30,000 runs (as a compromise between the proportion of solution space explored and run time) were carried out (the run time on a desktop PC was 3.5 days), and results were generated for both time series (Table 1) using a 1-min FL time step. Figure 5 provides histograms of all results for both time series using decimal inputs. For time series 1, the flood volumes range from 18.7 to 231.6 m^3, while for time series 2, the range is from 7.5 to 172.6 m^3. We can see by a reference to the 1-min time step test results in Table 2 that GA solutions are in the optimal region of the generated solutions. For time series 1, all of the trained flood volumes are within the bottom 2nd percentile, while all of the test results for time series 2 are within the bottom 43rd percentile of the Monte Carlo results. The average flood volumes are in the 0.3 and 28th percentiles, respectively. The flood volumes for the default FL were in the 51st and 73rd percentiles, respectively.

Example of controller operation

Prior to deploying a particular controller, the modeller will run a number of optimisations (for a particular FL time step) using a dataset containing real events for a particular hydraulic sewer model. The selection of the controller to be deployed can be based not only on minimum flood volumes (from training and test sets) but also by using the gate metrics. As illustrated in Figure 5, the solution space has multiple candidate solutions giving similar flood volume reduction; thus, a solution minimising both gate movements and flood volume can be selected.

In order to show the effect of the different control options, Figure 6 shows four cases for time series 2, firstly the control case where the gate is inactive, secondly the default expert control and finally the test cases for two optimised controllers. The *y*-axis in Figure 6 shows the proportional depth, where 0% is the pipe invert and 100% represents the ground surface, so any depth >100% indicates flooding. Figure 6(a) shows five periods when flooding occurs at the flood location, and at these times, the water depth upstream of the gate is at 22% or below, showing the potential for storage. Figure 6(b) shows that the default gate control reduces the amount of time flooding

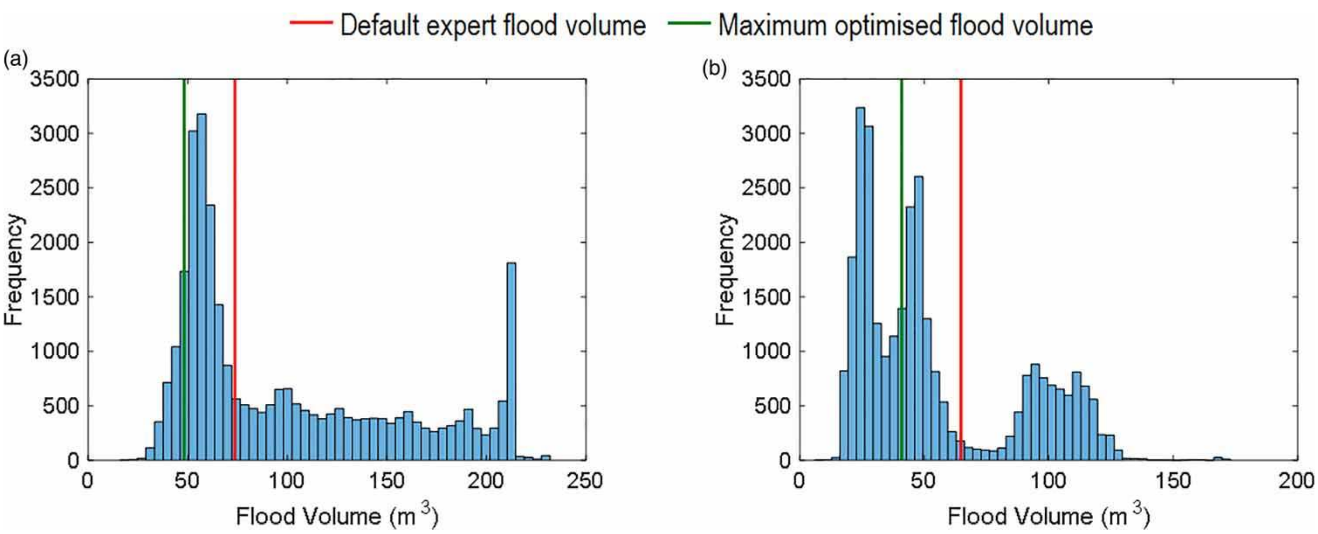

Figure 5 | Histogram of Monte Carlo runs for the 1-min time step, decimal MFs. (a) Time series 1 and (b) time series 2.

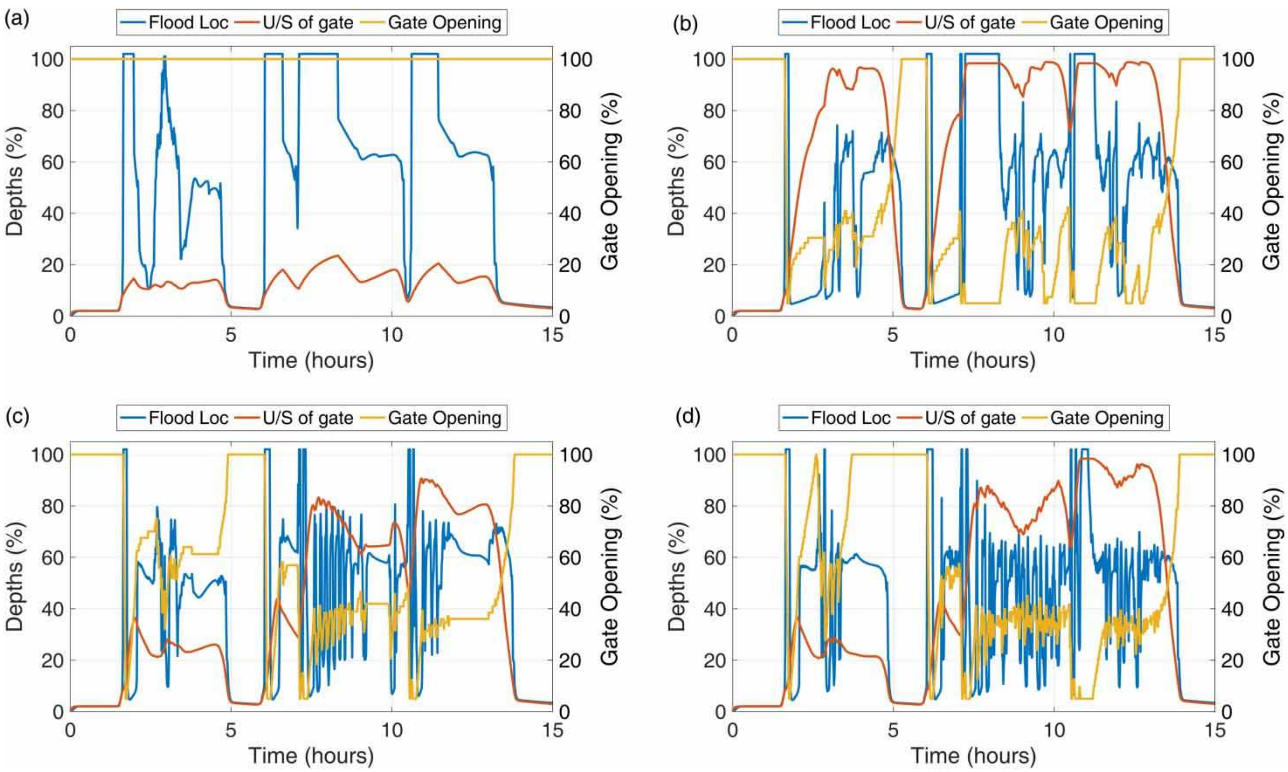

Figure 6 | Time series 2 results for control, default unoptimised and two optimised cases, FL run at a 1-min step. (a) Gate inactive (control), flood volume 128.2 m^3. (b) Default expert, unoptimised, flood volume 73.7 m^3, gate # 268, gate dist. 3.52 m. (c) Optimised, pseudorandom initialisation strategy, RNG shuffled, flood volume 22.62 m^3, gate # 239, gate dist. 3.44 m. (d) Optimised, random initialisation strategy, RNG shuffled, flood volume 21.42 m^3, gate # 381, gate dist. 5.27 m.

by utilising this storage. The remaining flooding periods are either because the gate has not responded quickly enough or because the storage has remained too full between events. Figure 6(c) shows that an optimised result can manage the storage better, and it further reduces the flooding volume and uses less storage while doing so. The remaining flooding is at the start of the four main flooding periods seen in the control case. The gate can be observed to operate more regularly and tends to operate with a higher opening percentage than the default expert case. Figure 6(d) shows an alternate optimisation which gives a very similar overall flood volume to Figure 6(c), but the gate is moving significantly more, with 142 more movements and moving an extra 1.83 m (53% further). These extra movements result in a much noisier depth profile, use more energy to move the gate and result in more wear to the gate seals, and so are undesirable.

Future work

The presented system has tackled the issue of flood reduction, but the future work could explore other issues, such as water quality improvement, e.g. by decreasing the frequency and volume of CSO discharges. Potential further development could include multi-objective optimisation (e.g. include the number of gate movements, total time storage is used, total flood duration and water quality aspects). The application of controller optimisation to different drainage networks (including the testing of optimisation in a live situation) is also planned, as is investigating how optimised MFs vary depending on the sewer network configuration and imposed rainfall. When considering future implementations of the system in new locations, results provided have demonstrated that in this relatively complex solution space, there are many local minima. This would suggest that the FL control algorithm is fairly transferable, in that many different settings of the algorithm should produce a reasonable flood reduction in potentially multiple networks. However, it is anticipated that to obtain the optimum FL control algorithm settings, it would be necessary to carry out GA optimisation using a hydraulic model of the new network. It would make an interesting future study to apply the FL controlled gate in other networks to understand how MFs might vary and whether the GA-optimised results are optimal among multiple networks, or if optimised MFs are network specific.

CONCLUSIONS

A GA software tool was coded to optimise an FL control system which uses local water-level sensing and a flow control gate to adjust the spatial distribution of the in-pipe water volume to reduce the local flood risk. The input MFs of the FL control algorithm are optimised using the outputs from a calibrated SWMM hydrodynamic model. A case study comprising training and test scenarios utilised the compiled rainfall time series based on recorded rainfall and incorporating multiple events was used in the optimisation. The average reduction in the flood volume for the GA-optimised input MFs when compared to no gate control was 66%. The GA also performs well compared to the expert (in sewer hydraulic modelling) defined MFs for unseen test rainfall events, resulting in an average 25% decrease in the flood volume (an average of 52% reduction for training events). Two GA-based approaches were tested. The integer-based GA optimisation performed better than the decimal version on unseen events, and the computational run time was significantly reduced. Both approaches operated significantly better on a 1-min FL control time step (an average of 56% flood reduction on unseen events), compared to a 2- or 5-min time step. The analysis showed that the increased performance in terms of flood volumes increases gate movements by an average of 28% and the total distance moved by the gate by an average of 34%. Key features include:

- Pioneering an autonomous, localised control technique for reducing the urban flood risk which optimises control rules based on virtual sensor data without human intervention.
- The GA-optimised FL approach is applicable to any hydraulically modelled network.
- The technique was applied on multiple (non-Gaussian) rainfall events and with performance demonstrated on unseen events (not used in optimisation).

The true potential of data-driven techniques is to distribute intelligent control and machine learning in a localised, autonomous manner rather than via centralised control. This paper contributes to defining such systems by developing an optimisation framework for a FL RTC control strategy using hydraulic models. Distributed methodologies offer significant benefits to the management of the large distributed infrastructure such as the flood risk for piped networks. These benefits include more efficient and cost-effective control solutions, faster analysis and response times, simpler more resilient control solutions and the growth of 'smart', self-learning and fixing networks, reducing the cost of infrastructure management by moving to proactive rather than reactive network management.

ACKNOWLEDGEMENTS

The CENTAUR project (www.shef.ac.uk/centaur) has received funding from the European Union's Horizon 2020 research and the innovation programme under grant agreement no. 641931. The contents of this paper reflect the view of the authors – the Executive Agency for Small and Medium-sized Enterprises (EASME) of the European Commission is not responsible for any use that may be made of the information it contains.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this paper is available online at https://dx.doi.org/10.2166/hydro.2019.058.

REFERENCES

Abdel-Aal, M., Shepherd, W., Ostojin, S., Mounce, S., Shucksmith, J., Schellart, A., Skipworth, P., Steinhardt, J. & Tait, S. 2017 Developing and testing a fuzzy logic algorithm to alleviate the risk of flooding by controlling a flow control device in a laboratory setting. In: *14th International Conference on Urban Drainage*, 2017, Prague, Czech Republic.

Barco, J., Wong, K. M. & Stenstrom, M. K. 2008 Automatic calibration of the US EPA SWMM model for a large urban catchment. *Journal of Hydraulic Engineering* **134** (4), 466–474.

Beeneken, T., Erbe, V., Messmer, A., Reder, C., Rohlfing, R., Scheer, M., Schetze, M., Schumacher, B., Weilandt, M. & Weyland, M. 2013 Real time control (RTC) of urban drainage systems – a discussion of the additional efforts compared to conventionally operated systems. *Water Science and Technology* **10** (5), 293–299.

Branisavljević, N., Prodanović, D. & Pavlović, D. 2010 Automatic, semi-automatic and manual validation of urban drainage data. *IWA Water Science and Technology* **62** (5), 1013–1021.

Campisano, A., Cabot Ple, J., Muschalla, D., Pleau, M. & Vanrolleghem, P. A. 2013 Potential and limitations of modern equipment for real time control of urban wastewater systems. *Water Science and Technology* **10** (5), 300–311.

Deep, K., Singh, K. P., Kansal, M. L. & Mohan, C. 2009 A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation* **212** (2), 505–518.

Frier, H., Basso, L., Kaltoft, J., Nygaard, M., Lynggaard-Jensen, A., Hansen, H. P. & Molby, N. 2014 Real-Time Monitoring, Modelling and Control of Combined Sewer Systems, Prepared End of Project Conference, Aarhus January 2014. Available from: http://www.preparedfp7.eu/viewer/file.aspx?FileInfoID = 465.

Fuchs, L. & Beeneken, T. 2005 Development and implementation of a real-time control strategy for the sewer, system of the city of Vienna. *Water Science & Technology* **52** (5), 187–194.

García, L., Barreiro-Gomez, J., Escobar, E., Téllez, D., Quijano, N. & Ocampo-Martinez, C. 2015 Modeling and real-time control of urban drainage systems: a review. *Advances in Water Resources* **85**, 120–132. doi:10.1016/j.advwatres.2015.08.007.

Garofalo, G., Giordano, A., Piro, P., Spezzano, G. & Vinci, A. 2017 A distributed real-time approach for mitigating CSO and flooding in urban drainage systems. *Journal of Network and Computer Applications* **78**, 30–42.

Gonçalves, I. & Silva, S. 2011 Experiments on controlling overfitting in genetic programming. In: *Proceedings of the 15th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence, EPIA*, Vol. 84.

Gooré Bi, E., Gachon, P., Vrac, M. & Monette, F. 2017 Which downscaled rainfall data for climate change impact studies in urban areas? Review of current approaches and trends. *Theoretical and Applied Climatology* **127**, 685–699. doi:10.1007/s00704-015-1656-y.

IPCC 2014 Climate Change 2014: Synthesis Report. In: Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change (Core Writing Team, R.K. Pachauri and L.A. Meyer (eds.)). IPCC, Geneva, Switzerland, 151 pp. Summary for Policy Makers. Available from: http://www.ipcc.ch/pdf/assessment-report/ar5/syr/AR5_SYR_FINAL_SPM.pdf (accessed 22 May 2018).

Kroll, S., Weemaes, M., Van Impe, J. F. M. & Willems, P. 2018 A methodology for the design of RTC strategies for combined sewer networks. *Water* **10** (11), 1675. pp. 1–23. doi:10.3390/w10111675.

Lund, N., Falk, A., Borup, M., Madsen, H. & Mikkelsen, P. 2018 Model predictive control of urban drainage systems: a review and perspective towards smart real-time water management. *Critical Reviews in Environmental Science and Technology*. doi:10.1080/10643389.2018.1455484.

Mamdani, E. H. & Assilian, S. 1975 An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies* **7**, 1–13.

Mollerup, A. L., Mikkelsen, P. S., Thornberg, D. & Sin, G. 2016 Controlling sewer systems – a critical review based on systems in three EU cities. *Urban Water Journal* **14** (4), 1–9. doi:10.1080/1573062X.2016.1148183.

Natsheh, E. & Buragga, K. A. 2010 Comparison between conventional and fuzzy logic PID controllers for controlling DC motors. *IJCSI International Journal of Computer Science Issues* **7** (5), 128–134.

Nicklow, J., Reed, P., Savic, D., Desselegne, T., Harrell, L., Chan-Hilton, A., Karamouz, M., Minsker, B., Ostfeld, A., Singh, A. & Zechman, E. 2010 State of the art for genetic algorithms and beyond in water resources planning and management. *Journal of Water Resources Planning and Management* **136** (4), 412–432.

Pleau, M., Pelletier, G., Colas, H., Lavallee, P. & Bonin, R. 2001 Global predictive real-time control of Quebec Urban Community's westerly sewer network. *Water Science & Technology* **43** (7), 123–130.

Pierro, F., Khu, S. T. & Savic, D. 2006 From single-objective to multiple-objective multiple-rainfall events automatic calibration of urban storm water runoff models using genetic algorithms. *Water Science and Technology* **54**, 57–64. doi:10.2166/wst.2006.609.

Radhakrishnan, M., Pathirana, A., Ashley, R. M., Gersonius, B. & Zevenbergen, C. 2018 Flexible adaptation planning for water sensitive cities. *Cities* **78**, 87–95. doi:10.1016/j.cities.2018.01.022.

Reed, D., Faulkner, D., Robson, A., Houghton-Carr, H. & Bayliss, A. 1999 *Flood Estimation Handbook*. Institute of Hydrology, Wallingford.

Riaño-Briceño, G., Barreiro-Gomez, J., Ramirez-Jaime, A., Quijano, N. & Ocampo-Martinez, C. 2016 MatSWMM – an open-source toolbox for designing real-time control of urban drainage systems. *Environmental Modelling & Software* **83**, doi:10.1016/j.envsoft.2016.05.009.

Rossman, L. A. 2015 Storm Water Management Model User's Manual Version 5.1. Available from: https://nepis.epa.gov/Exe/ZyPURL.cgi?Dockey=P100N3J6.txt.

Sanderson, M. 2010 Changes in the Frequency of Extreme Rainfall Events for Selected Towns and Cities. UK Met Office Report for OFWAT. Available from: https://www.ofwat.gov.uk/wp-content/uploads/2015/11/rpt_com_met_rainfall.pdf.

Schutze, M., Haas, U., Scheer, M. & Schilling, W. 2005 New guidelines for planning of real time control for urban drainage systems. In: *Proceedings of the 10th International Conference on Urban Drainage*, Copenhagen.

Seggelke, K., Lowe, R., Beeneken, T. & Fuchs, L. 2013 Implementation of an integrated realtime control system of sewer system and wastewater treatment plant in the city of Wilhelmshaven. *Water Science and Technology* **10** (5), 330–341.

Shepherd, W., Ostojin, S., Mounce, S. R., Skipworth, P. J. & Tait, S. 2016 CENTAUR: Real time flow control system for flood risk reduction. In: *CIWEM Urban Drainage Group Autumn Conference*, 2016, Blackpool, UK. doi:10.5281/zenodo.400969.

Shepherd, W., Mounce, S. R., Ostojin, S., Abdel-Aal, M., Schellart, A., Skipworth, P. J. & Tait, S. 2017 Optimising a fuzzy logic real-time control system for sewer flooding reduction using a genetic algorithm. In: *Proceedings of Computing and Control for the Water Industry (CCWI2017)*, 5–7 September 2017, Sheffield, UK.

Solomatine, D. P. & Ostfeld, A. 2008 Data-driven modelling: some past experiences and new approaches. *Journal of Hydroinformatics* **10** (1), 3.

Sriwastava, A. K., Tait, S., Schellart, A., Kroll, S., Dorpe, M. V., Assel, J. V. & Shucksmith, J. 2018 Quantifying uncertainty in simulation of sewer overflow volume. *Journal of Environmental Engineering* **144** (7), art. no. 04018050. doi:10.1061/(ASCE)EE.1943-7870.0001392.

van Nooijen, R. R. & Kolechkina, A. 2013 Speed of discrete optimisation solvers for real time sewer control. *Water Science and Technology* **10** (5), 354–363.

Vanrolleghem, P., Benedetti, L. & Meirlaen, J. 2005 Modelling and real-time control of the integrated urban wastewater system. *Environmental Modelling & Software* **20** (4), 427–442. http://dx.doi.org/10.1016/j.envsoft.2004.02.004.

Vezzaro, L. & Grum, M. 2014 A generalised dynamic overflow risk assessment (DORA) for real time control of urban drainage systems. *Journal of Hydrology* **515**, 292–303. http://dx.doi.org/10.1016/j.jhydrol.2014.05.019.

Villeneuve, E., Jolicoeur, N., Pleau, M., Marcoux, C., Field, R. & Stinson, M. 2000 The choice of a real-time control strategy for combined sewer overflow control. In: *Proceedings of Water Environment Federation Collection Systems Wet Weather Pollution Control: Looking Into Public, Private, and Industrial Issues Conference*, May 7–10, 2000, Rochester, NY.

First received 15 March 2019; accepted in revised form 30 November 2019. Available online 24 December 2019