A distributed real-time approach for mitigating CSO and flooding in urban drainage systems

Giuseppina Garofalo[b], Andrea Giordano[a,*], Patrizia Piro[b], Giandomenico Spezzano[a], Andrea Vinci[a]

[a] *CNR - National Research Council of Italy*
Institute for High Performance Computing and Networking (ICAR)
Via P. Bucci 41C - 87036 Rende(CS) - Italy
[b] *Department of Civil Engineering of the University of Calabria*
Via P. Bucci - 87036 Rende (CS), Italy

Abstract

In an urban environment, sewer flooding and combined sewer overflows (CSOs) are a potential risk to human life, economic assets and the environment. To mitigate such phenomena, real time control systems represent a valid and cost-effective solution. This paper proposes an urban drainage network equipped by sensors and a series of electronically movable gates controlled by a decentralized real-time system based on a gossip-based algorithm which exhibits good performance and fault tolerance properties. The proposal aims to exploit effectively the storage capacity of the urban drainage network so as to reduce flooding and CSO. The approach is validated by considering the urban drainage system of the city of Cosenza (Italy) and a set of extreme rainfall events as a testbed. Experiments are conducted by using a customized version of the SWMM simulation software and show that the CSO and local flooding volumes are significantly reduced.

Keywords: distributed real time systems, multi-agent systems, PID, Gossip-based algorithm, sewer system, flooding, combined sewer overflow

*Corresponding author
Email address: `giordano@icar.cnr.it` (Andrea Giordano)

1. Introduction

Climate change and exponential reduction of pervious surfaces in urban catchments have led to an increase in the frequency and magnitude of two undesired phenomena which negatively affect human life, economic assets and the environment: (i) local *flooding* and (ii) *combined sewer overflows* (CSOs)[1][2].

Urban flooding occurs when the *urban drainage system* (UDS) becomes drastically overloaded during extreme rainy events, causing untreated combined sewage and storm water to back up into basements and to overflow from manholes onto surface streets. This phenomenon is generally worsened by obstructions in conduits and manholes due to an infrequent maintenance.

CSO [3][4] takes place when the *wastewater treatment plant* (WWTP) is not able to treat the wastewater delivered by the UDS. Specifically, the sewage and wet weather flows that exceed the WWTP treatment capacity. Specifically, the sewage and wet weather flows are conveyed through the UDS to the WWTP until the maximum treatment capacity is reached. The exceedance of the water flows are discharged directly into the receiving water bodies, such as rivers or lakes, without receiving any treatment. As a consequence, CSO is one of the major contributors to water pollution experienced in rivers, lakes etc.

In accordance with the European Water Framework Directive (WFD) and the EU Flood Directive (2007/60/EC), measures need to be adopted to manage stormwater volumes efficiently by reducing CSOs and preventing urban areas from sewer flooding. For this purpose, offline storage facilities, which temporarily accumulate stormwater volumes, are widely used, even though they are often overly expensive due to the high construction and maintenance costs. In contrast, approaches aiming at temporarily accumulating stormwater volumes directly in the existing UDSs have been also developed thus avoiding large investments [5][6][7]. These approaches are supported by the fact that the UDSs are typically designed by taking into account a set of safety factors. In particular, conduits are intentionally designed to be larger than required in the case of typical network working conditions. Basically, the UDS is managed by a *real-time control* (RTC) system which requires the network to be embedded with sensors and actuators permitting the network to be real-time monitored and regulated so as to adapt to the different rainfall events [8][9]. In particular, physical layer information is collected by sensors and sent to the control component which elaborates an optimized actuation strategy so as to ensure the desired UDS working behaviour.

Many works in the literature focus on RTC based on a centralized approach. For instance, in [10] a sewer networks global optimal control (GOC) scheme with a two-level architecture has been designed. The upper level is composed of a central station, which computes flow set points, whereas the lower level is composed of local stations, which are used for monitoring, flow computation, data validation and feedback control. The real-time computer is dedicated to all RTC operations and supports a supervisory software, a GOC software, a non-linear hydrologic-hydraulic model and a non-linear programming algorithm. The site is controlled automatically under a flow set point computed by the GOC scheme. The optimization problem is defined by a multi-objective (cost) function and a set of equality and inequality constraints, based on the following control objectives: minimizing overflows, minimizing set point variations and maximizing the use of WWTP capacity. In [11] a multi-objective optimization genetic algorithm is proposed which is used to derive the Pareto optimal solutions, which can illustrate the whole trade-off relationships between objectives. In [5] a global optimal predictive real time control system has been implemented, which involves solution of a multi-objective optimisation problem. The control objectives are the minimisation of overflows, the maximization of the use of the treatment plant capacity, the minimization of accumulated volumes and, finally, the minimization of variations of the setpoints. The real time control system is implemented at a central station and uses flow monitoring and water level data, rainfall intensity data, radar rainfall images and 2 hours rain predictions. Set-points are translated into moveable gate positions at local stations by Programmable Logic Controllers (PLC). In [7] a global RTC of the sewer system in the city of Dresden, Germany, has been implemented, consisting of a local equipment (on site), an equipment for data transfer between control location and control center and an equipment in the control center. The control room operates with a process control system (PCS), also known as SCADA system (Supervisory Control and Data Acquisition), to monitor the urban drainage system. To allow for the realization of a global RTC, the existing PCS is provided with an additional control computer connected to the SCADA system. A RTC based on a neural optimal algorithm was applied on the wastewater plant collection system in Seattle, USA [12]. In the system a main control center is located at the treatment plant and a software which implements a SCADA system, is designed to monitor and control pump and regulator stations, including telemetry of real-time data on water levels, gate positions, etc. The RTC is developed to regulate in real-time an in-line storage in a combined sewer system. The neural-optimal control model integrates the dynamic hydraulic model with the optimization model used to minimize untreated overflows while maximizing through-flows to the wastewater treatment plant for a storm event. In [13] the potential of global water-quality based on RTC in the Lynetten catchment (Denmark) has been investigated. The integrated control of the catchment connects the control of the WWTP and catchment and structures, such as detention basins and pumping stations in the same platform. Such strategy uses the Dynamic Overflow Risk Assessment (DORA) approach to minimize CSO volumes, which aims at reducing overflow risk in the system by minimizing a global cost function through an optimization routine.

In those studies, as the control system is fully centralized, several issues arise due to the large amount of data to be read, managed and processed. Using a typical centralized monolithic approach, all sensory data are sent to a central unit that elaborates a suitable strategy based on a comprehensive network model thus producing commands for the actuation part. This approach tends to be very reliable, but it has some drawbacks: (i) it requires a complex mathematical model of the network (ii) all physical parts (sensors and actuators) need to be connected with and reachable by the remote central unit and (iii) a failure on one node (especially in the case of the central unit) can compromise the whole system.

Conversely, in this work the distributed real-time control (DRTC) system is adopted, exploiting a multi-agent paradigm and specifically a gossip-based algorithm. The UDS is equipped with electronically moveable gates and a set of water level sensors spread across the network. All the gates are locally controlled by *Proportional Integrative Derivative* (PID) controllers which are globally orchestrated by the mentioned gossip-based algorithm thus achieving an optimal hydrodynamic behaviour in terms of CSO and flooding reduction. The case study is the UDS of the city of Cosenza (Italy), which is modelled by using the StormWater Management Model (SWMM) simulation software. SWMM is an open-source computer model widely used by the hydraulic engineering community for simulation of hydrodynamic water and pollutant transport in sewer systems. It is provided by US EPA [14] and permits an accurate simulation of the hydrological and hydraulic behaviour of the UDS during both dry and wet weather conditions. SWMM simulation software has been customized in order to allow it to be integrated with an external real-time control module. Experiments, conducted using a set of 15 extreme rainfall events recorded in the years 2011-2015, show an substantial reduction of both CSO and flooding when the proposed approach is exploited.

2. Distributed control of urban drainage network

2.1. The structure of an urban drainage system

An *urban drainage system* (UDS) (see Figure 1) aims at collecting and delivering the combined wastewater (sewage and wet weather flows), coming from the urban catchment, to a wastewater treatment plant (WWTP). A UDS physically consists of a series of junctions, conduits, weirs and storage units [15], which, in most cases, forms a dendritic structure. Indeed, the network typically follows a tree structure, in which a series of sub-networks (branches), with smaller pipes, converges into a main network (trunk), consisting of larger collector conduits. Sequentially, each sub-network (branch) in turn follows a tree

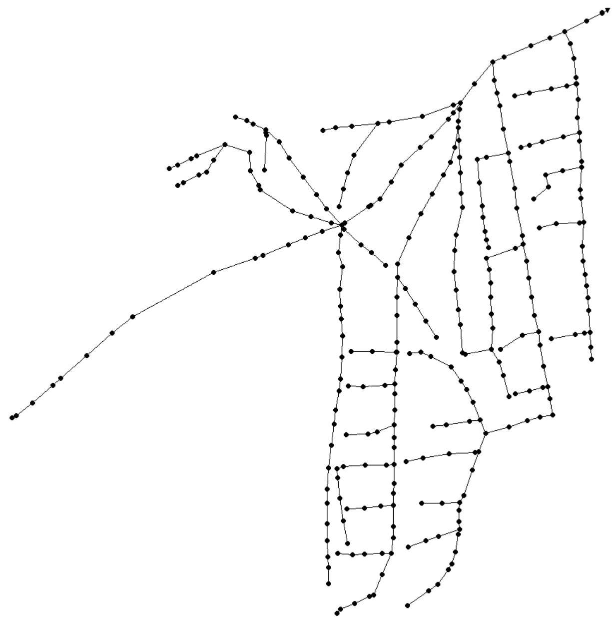


Figure 1: An example of urban drainage system

distribution. Finally, all conduits are linked to one final collector pipe, which ends with the *outfall* of the network, delivering the wastewater to the WWTP. Overflow structures are used to direct the combined wastewater volumes which exceed the capacity of the WWTP directly into receiving water bodies.

2.2. Drainage network instrumentation

In order to achieve the proposed goals, the network requires the following equipment: (i) *sensors*, one water level sensor per conduit and a flow sensor on the outfall; (ii) *computational nodes*, which can host and execute the distributed control algorithm [1] ; (iii) down-hinged *moveable gates*, which can be real-time regulated electronically.

[1]the computational nodes can be custom-built devices or single-board computers such as *Raspberry pi* or *Beagleboard* which can be effectively distributed inside the network because they have low energy consumption and small size.

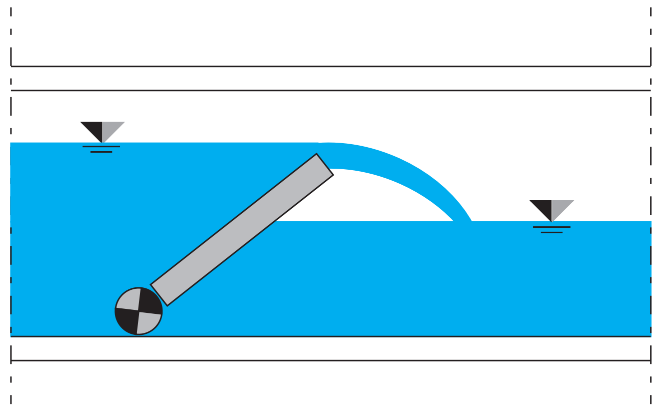


Figure 2: The down-hinged movable gate

The computational nodes dynamically regulate the gates according to the information acquired by the sensors in the neighbor areas. Each computational node has a partial view of the network as it can read only from the sensors, and actuate only on the gates, which are located in its spatial neighbourhood, i.e. the sensors/gates it can physically reach. In addition, each computational node communicates only with its neighbor peer nodes. The computational nodes are distributed throughout the network in order to cover all the points of interest, i.e. the sensors and the gates. The nodes read data from the sensors and collectively process the acquired information in order to trigger suitable actuations on the gates. The collective computation of the network of nodes supplies the gates with an "intelligent" behaviour.

The electronically moveable gates are made up of mobile plates rotating around a horizontal hinge placed on the bottom of the conduit[2], as shown in Figure 2. The gate is completely closed when the plate rotates in a perpendicular position with respect to the flow direction. Conversely, the gate is fully open when the plate is parallel to the flow. When the gate is closed, the opening area is null and no flow rate is delivered from the node. An intermediate position of

[2]The top-hinged gate was avoided to prevent high speed flow under the plate, which may re-suspend the sediment material at the bottom of the sewer pipes.

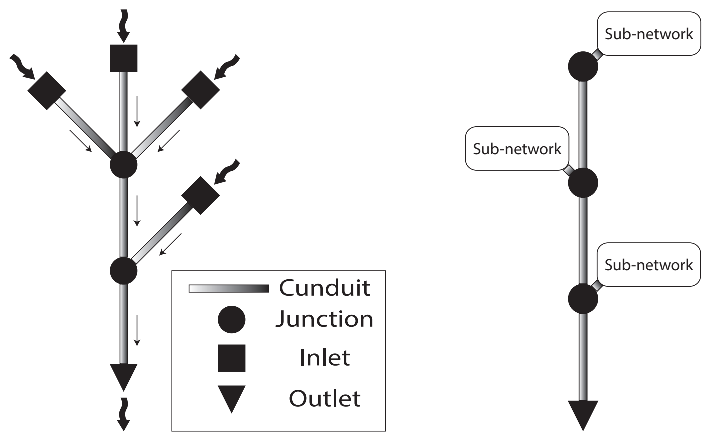
the gate corresponds to a partial opening degree. The gates allow utilization of the full storage capacity of the conduits by accumulating the excess stormwater volume in the less overloaded parts of the system [16].

2.3. Drainage network modelling

Our approach is focused on reducing: (i) flooding phenomenon and (ii) CSO problem. The flooding issue is tackled by balancing the water level throughout the conduits of the network so as to reduce flooding in the most overloaded conduits at any given moment. In other words, balancing the water level means that the underloaded conduits are triggered to store additional water in order to help the overloaded conduits when they are about to overflow. The CSO problem is addressed by extending this strategy, taking into account also the flow at the outfall as will be better clarified in Section 2.4.4.

Based upon the description reported in Section 2.1, a drainage network can be formally seen as a graph (V, E) of nodes $v \in V$ connected by edges $e \in E$. More specifically V comprises *Junctions* $j \in J$, *Inlets* $l \in L$ and *Outlets* $o \in O$. E is made up of *Conduits* $c \in C$. Junctions are just intersection points for conduits. Inlets are nodes where runoff enters into the system. Outlets are the points of the network where water is discharged into a river, lake, reservoir and so forth. Conduits are pipes of different cross-sectional shapes where the water flows [17].

In the following we introduce an enhanced version of the model used in our approach for achieving the proposed goal of balancing the conduits' water level. This refined model is based on some other features which are inherent to drainage networks. Firstly, in a typical urban scenario, the whole drainage watershed can be broken down into several, not connected, networks in which each network comprises only one outfall. In addition, each network is likely to be modelled by a tree structure. Indeed, we can see a network as a main channel and a set of sub-networks which are connected, through junctions, to different points of the main channel. Each of these sub-networks can be defined recursively in the same way. Finally, it can be assumed that inlets are located in the “leaves” of the tree. A very simple drainage network is outlined in Figure 3(a).



(a) A simple drainage network scheme

(b) Recursive definition of drainage network

Figure 3: Drainage network structure

On the basis of the previous considerations we formally define a drainage network as follows. Firstly, we define *Most Simple Drainage Network* ($MSDN = (c, l)$) as a network only made up of one conduit c ending with inlet node l. Then, we define a generic drainage network DN as either just an $MSDN$ or a couple (M, S) where M represents the main channel and S a set of DNs defined in the same way. A main channel M is an ordered set of conduits in which each conduit is linked with the next one through a junction. The last conduit optionally ends with an outlet node. Figure 3(b) shows graphically this recursive definition. Figure 4(a) and 4(b) show respectively: a case of a realistic network and the sub-networks, surrounded by dashed lines, as results from the above definition. We also define *Degree* of a DN as a function $Degree(DN)$ defined as:

$$Degree(DN) = \begin{cases} 0, & \text{if } DN \text{ is a } MSDN \\ 1 + max_{s \in S}(Degree(s)), & \text{with } DN = (M, S) \end{cases}$$

The recursive definition of a drainage network permits us to extend an optimization strategy conceived for simple networks (such as the one shown in Figure 3(a)) also for complex/general scenarios as will be better specified in the following. Basically, given the physical complex drainage network, we firstly generate a set of simpler logical networks, afterwards, we execute the "simple network" optimization strategy on each of these generated network and then we obtain the optimal behaviour to be actuated in the original network. For this purpose we firstly define, for a given network DN, the set $nets(DN)$ as follows:

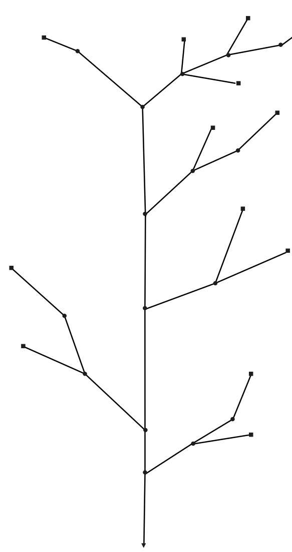
$$nets(DN) = \begin{cases} \emptyset, & \text{if } DN \text{ is a } MSDN \\ \{DN\} \cup \bigcup_{s_i \in S} nets(s_i), & \text{if } DN = (M, S) \end{cases}$$

Then, starting from $nets(DN) = \{dn_i\}$ with $dn_i = (M_i, S_i)$, we generate the set $GN = \{gn_i\}$, in which each $gn_i = (M'_i, S'_i)$ is given by the following formulas:

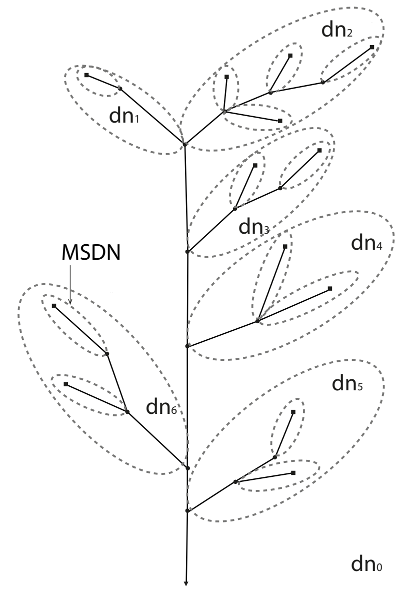
$$M'_i = \begin{cases} M_i, & \text{if } i = 0 \\ M_i \cup o_i, & \text{elsewhere} \end{cases}$$

$$S'_i = \{msdn_k = (c_k, l_k) : \forall dn_k \in S_i\}$$

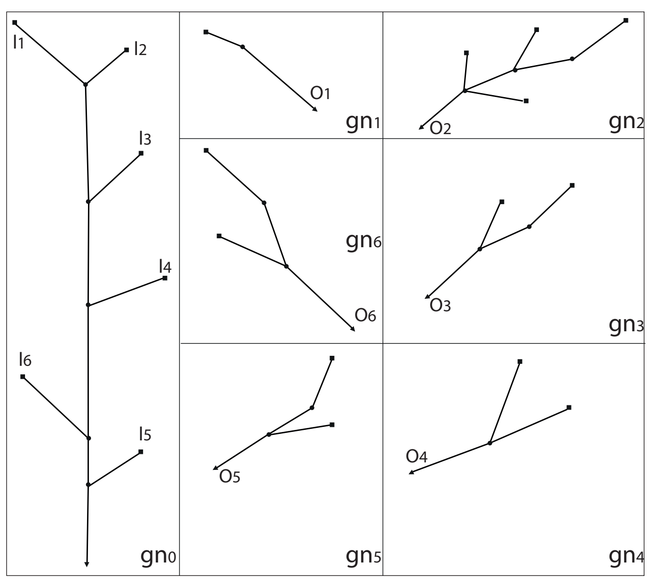
The intuitive idea concerns replacing the set of sub-networks $S_i = dn_k$ with a new set S'_i made up of only MSDNs. In particular, each $dn_k \in S_i$ corresponds to an $msdn_k = (c_k, l_k) \in S'_i$. M'_i is M_i plus o_i outlet node except for $i = 0$, since the top level sub-network already hosts an outlet node. As a result, starting from a complex network we generate a set of networks with 1 degree. Our approach relies on the assumption that the optimization algorithm, tailored for one degree networks, when executed simultaneously on all the generated networks permits us to find the global optimal behaviour for the original network.



(a) Network



(b) Sub-network



(c) Networks generated

Figure 4: Sub-networks in a realistic case

In other words, balancing the water level in all the generated networks results in a local flooding reduction but also concurs to the balance of the water level in the original network. This assumption holds only if the separated executions are kept consistently linked. Basically, it must be ensured that during the advancement of the execution the incoming flow of the generated inlet nodes l_k is set to be equal to the outcoming flow of the corresponding outlet nodes $o_{i=k}$ (i.e., the outlet nodes of the networks gn_k).

For instance, let's consider the network of Figure 4(a) and its set of generated networks $gn_{1..6}$, shown in Figure 4(c), where the balancing algorithm runs. The $i_{1..6}$ water levels, which correspond to the $o1..6$ water levels, are involved in both gn_0 balancing operation and $gn_{1..6}$ balancing operations at the same time. As a consequence, the balancing operations of all the generated networks, even if executed separately, results in a global balancing.

Practically, the drainage network control is achieved step-by-step. At each step the optimization algorithm runs upon each generated network separately thus producing actuation strategies tailored to the local hydraulic conditions. The resulting local actuations immediately change the status of the generated network in order to set up the local optimal hydraulic conditions, but also affect the neighbour generated networks, and gradually the whole network, because of the water flowing. Summarizing, the generated networks are kept linked with each other due to two different reasons. The first, described before, is essentially "logical" as the local optimization considers some parts of the network which are kept linked to parts of other generated networks. The second reason is related to the "physics" of the water network, i.e., the local hydraulic conditions smoothly affect the rest of the network by water flowing. As a consequence, the balance of each generated network contributes, over time, to an emergent behaviour which is the balance on the whole network. Dividing the network and balancing at generated network level, rather than balancing directly at whole network level, permits the local unbalances which cause flooding to be smoothed rapidly. Indeed, when a conduit is overcharged the exploitation of the residual capacity of a neighbour conduit is more convenient because the closer are the places the faster is the "transfer" of the water between these places. Anyway, the emergent global balance is also significant because it ensures that, since all the generated networks are kept equally loaded, each one is equally capable of storing additional water coming from an unexpected rainfall event. If, contrariwise, the generated networks are not kept balanced, the same rainfall event would cause flooding in the most loaded networks, while leaving the others partially unloaded.

2.4. The distributed control algorithm

The optimal behaviour we want to actuate on the drainage network consists in balancing the water level throughout the conduits of the network. In the following we firstly describe how to place the gates inside the network, afterwards, we details the multi-agent algorithm which is executed on each generated network.

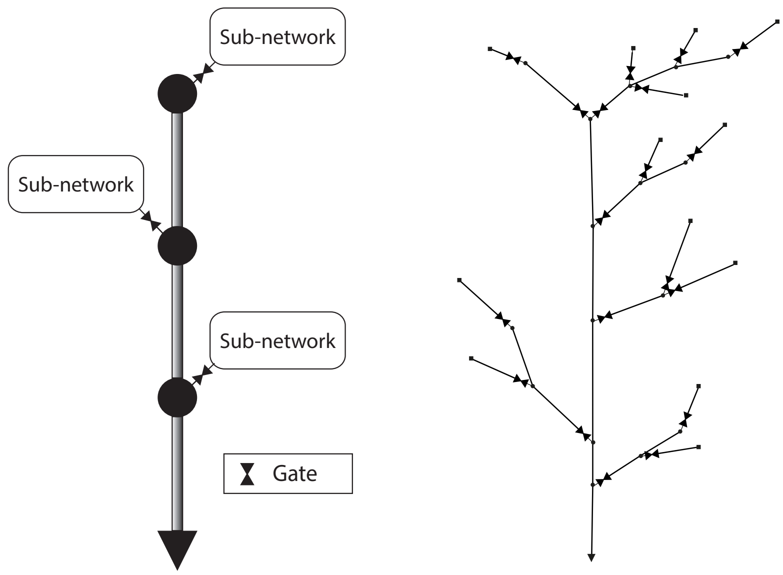
The gates are located at the points of the network where sub-networks are connected to the main channel. Figure 5(a) shows the logical places for inserting the gates, while Figure 5(b) shows the gates insertion in a case of a realistic network.

As said before, each computational node has a partial view of the network as it reads only from sensors located in its spatial neighbourhood, i.e. the sensors it can physically reach. In the same way, it can actuate only on its neighbour gates. On the basis of the previous considerations our proposal lies in using a *distributed agent-based* architecture [18]. The agent paradigm has several important characteristics:

Autonomy. Each agent is self-aware and has a self-behaviour. It perceives the environment, interacts with others and plans its execution autonomously.

Local views. No agent has a full global view of the whole environment but it behaves solely on the basis of local information.

Decentralization. There is no "master" agent controlling the others, but the system is made up of interacting "peer" agents.



(a) Logical (b) Realistic case

Figure 5: Gates positions

Through these basic features, multi-agent systems make it possible to obtain complex *emergent* behaviours based on the interactions among agents that have a simple behaviour. Examples of emergent behaviour could refer to the properties of adaptivity, fault tolerance, self-reconfiguration, etcetera. In general, we could talk about *swarm-intelligence* [19] when an “intelligent” behaviour emerges from interactions among simple entities.

In the case of drainage networks, the property of fault tolerance is particularly useful since the system needs to continue to operate properly even if unexpected conditions occur, such as obstructions and blockages, which may reduce the hydraulic capacity of the system.

Our proposal considers one agent per gate. Each *gate-agent* runs on one of the computational nodes covering the specific gate, it can perceive the local water level and communicate with the neighbouring gate-agents in order to elaborate a proper actuation strategy for its gate. Another agent, called *outfall* *agent*, is logically associated with the outlet node, it behaves the same as other agents except for the actuation part, indeed, it is not associated with any gate. Figure 6 gives an intuitive idea of the agents' role in the generated networks. For each generated network, the algorithm consists in real-time balancing the water level perceived by the agents. Given that the conduits have different sizes, the water level is normalized with respect to the height of the conduit.

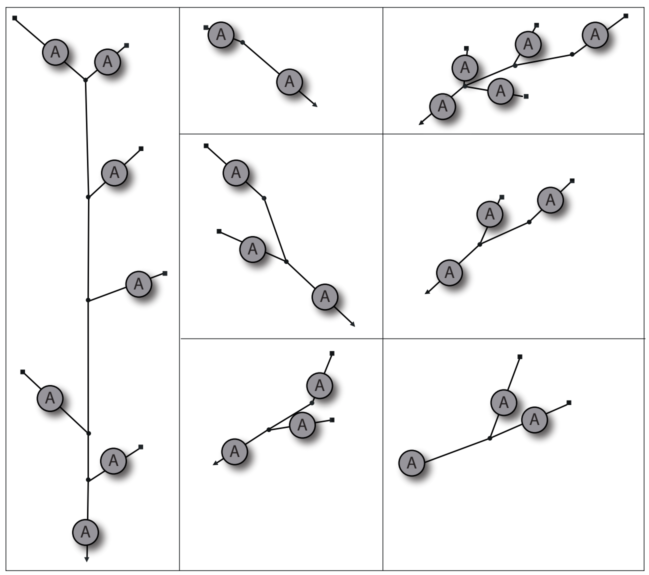


Figure 6: Agents in generated networks

This water balancing is achieved by means of agents continuously executing two tasks:

Task 1. figuring out collectively the average of the water level in the generated network.

Task 2. each agent triggers its specific gate in order to bring the water level closer to that average.

Given that an agent has no global knowledge of the network, i.e. it does not know the water level in each point of the network, *Task 1* is accomplished by exploiting a gossip-based algorithm summarized in Section 2.4.1. This kind of algorithm also supplies the previously mentioned fault-tolerance property. Anyway, even if we knew the optimal water level to set, we would not know how to tune the gate so as to achieve it. Indeed, the relationship between the actuation upon the gate (i.e. its opening degree) and the actual change of water level is determined by the structure of the whole network and the dynamics of the water flowing through the system, so it is very hard or even impossible to deduce a tractable mathematical model for it. For this reason *Task 2* is accomplished exploiting a *PID* controller as explained in Section 2.4.2.

2.4.1. Task 1: Gossip-based aggregation

In a Gossip-Based Algorithm [20] there are many nodes interconnected through a network. Each node possesses some numerical values and can exchange information only with a limited set of peer nodes (i.e. its neighbourhood). The goal of this kind of algorithm concerns estimating global aggregate values such as average, variance, maximum and so forth, despite only local communication being possible.

Basically, in the case of average aggregated value, each agent maintains its current measured value and its local average (initially set to the measured value). The algorithm consists in continuously exchanging local averages among neighbour nodes. Each time a node receives the average of a neighbour node, it updates its local average (just applying average operator). Values exchanges and local computations are done continuously for enough *step*s so as to ensure that each local average, computed at every node, converges to the actual global average (the algorithm convergence is proved in [20]).

In our approach, we run the gossip-based algorithm for computing the average of the degree level as measured by all the agents of a generated network. When the algorithm converges, the estimated average value is exploited by each gate-agent for tuning its gate so as to bring water levels closer to that average (*Task 2*). Afterwards, a new iteration begins, consisting in: measuring the new values of the water level, running the gossip-based algorithm until the convergence is reached again, using the computed value for the actuation part, and so on and so forth. Running this process continuously ensures the fault-tolerance property mentioned before because even if an unforeseen event dramatically changes some structural properties (as in the case of: obstructions, blockages, damages etc.) the algorithm is able to pass smoothly from the previous computed optimal value to the new one.

It is important to underline here that the convergence time of the gossip algorithm in a realistic scenario can be considered acceptable compared with the dynamic of the water levels. Indeed, as reported in [20], the difference between the maximum and minimum agent values decreases exponentially and this difference is reduced by several orders of magnitude after just a few cycles. For example, a normalized maximum-minimum difference equals to 10^2 is achieved after only 5-10 steps. As described before, each step consists of two operations: (i) the neighbour nodes exchange messages among themselves, and (ii) each node computes the new average value. The second operation is negligible from a computation time point of view. The first operation involves some network communications between computational nodes which, in the worst case, are no more than some kilometres distant from each other (recall that the gossip algorithm runs upon the generated networks). We can consider 100-500 milliseconds as an overestimated upper bound for the time needed for this kind of operation. As a consequence, 5 seconds can be considered an overestimated upper bound for the convergence of the gossip-based algorithm. In the experiments we found that for each conduit, the water level varies at most by $1-2\%$ for a 5 seconds time interval. For this reason, the convergence process, with respect the water level dynamics, can be considered fast enough not to introduce significant errors.

Obviously, there are other approaches to find the average of a given number of peer nodes. The gossip-based algorithm was chosen because it exhibits some useful properties. First of all, it is tolerant to communication failure and other kinds of unforeseen events as better explained in Section 2.4.3. In addition, differently from other approaches, there is not “master” node which knows the global average but each node stores the average value collectively computed so far. This kind of "distributed knowledge" is very important in the considered scenario because each gate must be controlled separately and continuously without being affected by communication failure or the need to synchronize with a master node.

2.4.2. Task 2: tuning gates through PID controllers

Once an agent knows the global water level through the previously described "gossip-based aggregation", there remains the problem of appropriately tuning its gate so as to reach that "desired" level.

This issue is addressed using the well-known controlling technique called Proportional Integral and Derivative (PID) control [21] which, indeed, can be used when you do not know an exact mathematical model of the system you want to control.

A PID controller is a control loop feedback mechanism where an error value is computed as the difference between a measured output of a process and the desired value (setpoint) (see Figure 7). The controller tries to minimize this error, appropriately tuning the actuator device.

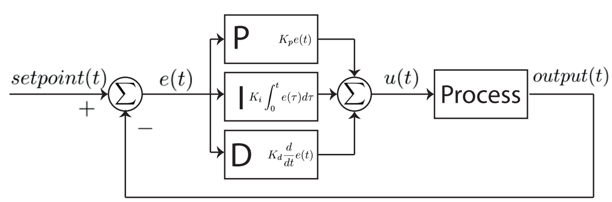


Figure 7: PID controller

The setting of the actuator device is determined by three effects, suitably tuned by three parameters: the proportional one (P), the integral one (I) and the derivative one (D). P is determined by present error, i.e. the absolute error computed at the current evaluation. D measures the foreseen error, i.e. the expected error in the next step, computed deriving the error signal, while I represents the integral effect, a measures of the historical behaviour of the error signal. The following equation defines a general time-continuous PID controller.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau \; + K_d \frac{d}{dt} e(t)$$

Where $e(t) = setpoint(t) - output(t)$; $u(t)$ is the controller output at time t, i.e. the actuation signal; K_p, K_i, K_d are three constants which refer respectively to the proportional, integral and derivative effects. These parameters are tuned adopting the well-known ZieglerNichols method [22].

In the case of this study, each gate of the drainage network is controlled by a PID implemented by the gate-agent. $u(t)$ represents the degree of opening of a gate, $output(t)$ is the actual water level of its related conduit and $setpoint(t)$ is the “desired” water level, i.e., the average computed by *Task 1*.

2.4.3. Adaptivity to common failures

In Section 2.4.1 we claimed our approach ensures adaptivity to unforeseen critical events and failures because it is based on the gossip-based algorithm. In this section we clarify how the system reacts when such common failure conditions occur, i.e., gate failure, conduit blockage, node disconnection and node shut down.

Gate failure. In the case of gate failure the mobile plate is blocked and so the gate-agent is unable to change its opening degree. In such a case, the gate-agent keeps working and collaborating with its neighbour nodes playing the gossip-based algorithm. In other words, the water level of this gate-agent keeps contributing to the evaluation of the local average. As a consequence, even though it is unable to change the water level directly, this water level tends anyway to the local average since the neighbour gate-agents change their water levels in order to reach the local average.

Conduit blockage. In this case a portion of the network can experience a sudden water level increase which can cause the flooding phenomenon. Anyway, as in the previous case, the gates located in the neighbourhood are triggered to change their behaviour so as to counterbalance this increase. In fact, the latter mechanism cannot work properly if the increase of water level is not correctly perceived. Indeed, as can be seen in Figure 8, an obstruction causes a part of a conduit to be overloaded while the other is underloaded. For this reason, it is important where the water level sensor is placed because if it is placed in the "underloaded" part the gate-agent would perceive a decreased water level instead of an increased one. In our approach, this issues is addressed by deploying more than one sensor per conduit and taking the maximum sensed value as the water level value for the conduit.

Node disconnection. When a node is disconnected from its neighbour nodes, the agent is not able to evaluate the neighbourhood average and so it can not actuate properly upon the gate. When this case occurs, in our approach, the gate is set to be fully open while the agent stops to actuate on the gate at all. The resulting behaviour would be the same as if we had one gate less with respect to the current configuration.

Node shut down. This is the worst case because neither can the gate be properly controlled nor the water level communicated to the neighbours node. As a consequence, there is no chance of attenuating a possible overloaded scenario occurring in the specific conduit, but the neighbourhood would react anyway preventing the propagation of the overloaded situation.

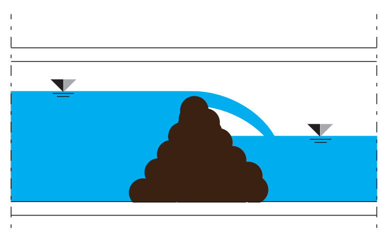


Figure 8: A blockage in a conduit

2.4.4. CSO Reduction

The approach described so far focuses only on reducing the flooding phenomenon. Anyway, the approach can be slightly enhanced so as to address the

CSO problem as well. As described in the introductory section, CSO occurs when the water flow on the outfall node exceeds the capacity of the treatment plant. For this reason the CSO issue can be tackled by controlling the water flow on the outfall node. Basically, we introduce an additional gate placed at the outfall and the outfall-agent is extended as to control the flow of the new inserted gate. Keeping the flow level under a certain desired value results unavoidably in increasing the water level which can cause local flooding. Fortunately, the algorithm described before is able to ensure a balanced water level throughout the whole network. In other words, when the water level on the outfall increases due to the flow control, the rest of the gates are triggered to store more water thus helping the outfall to decrease properly its local water level.

The latter consideration suggests that increasing the number of gates can results in further improving the CSO reduction. The experimental section (Section 3) will show that adding more gates can effectively improve the CSO reduction for the selected rainfall events.

2.5. SWMM

In this work the drainage network is simulated using the software StormWater Management Model (SWMM) provided by EPA [14], which is an open-source computer model for simulation of hydrodynamic water and pollutant transport in sewer systems. SWMM relies on a time stepped dynamic model, where the dynamic simulation is performed by numerically solving flow routing equations (dynamic wave) based on a fixed time step. The SWMM model also allows the modelling of water quality constituents, dry-weather pollutant build-up over different land uses in the contributing catchment and pollutant washoff from specific land uses during storm events. Figure 9 shows a snapshot of the software in execution.

The input data and parameters required for the SWMM model simulation of drainage flow hydrographs include physiographic characteristics of the catchment (e.g. the area and slope), physical characteristics of the sewer pipes (the

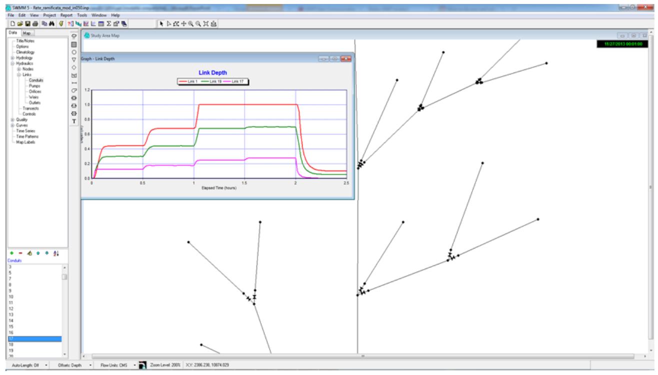


Figure 9: Snapshot of the SWMM software

diameter, length, slope and material) and the hydrological/hydraulic parameters such as the width of the subcatchments (i.e. the overland flow width).

The routing flow model used is the dynamic wave flow routing to predict non-steady flows through a general network of open channels, closed conduits and weirs. In contrast to simpler routing methods, this procedure can model such phenomena as backwater effects, flow reversals, pressurized flow, and entrance/exit energy losses [14]. The governing equations are the *Saint Venant equations* that are numerically solved using the modified *Euler method* (equivalent to a *2nd order Runge-Kutta method*)

2.5.1. SWMM Customization

Although SWMM permits some trivial real-time control of the parameters of the network (by defining some simple rules), for the purposes of the work, SWMM has been customized for permitting it to communicate in real time with a separate Java controller which implements the algorithm described in Section 2.3.

Figure 10 shows the architecture of the integration. SWMM has been enhanced by adding the *TCP connector* which is in charge of sending and receiving information to/from the external module during the advancement of a simulation.

More in detail, the connector enables the external module to select where

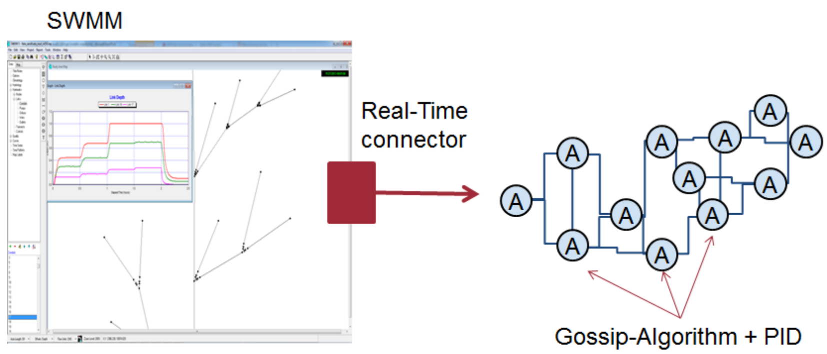


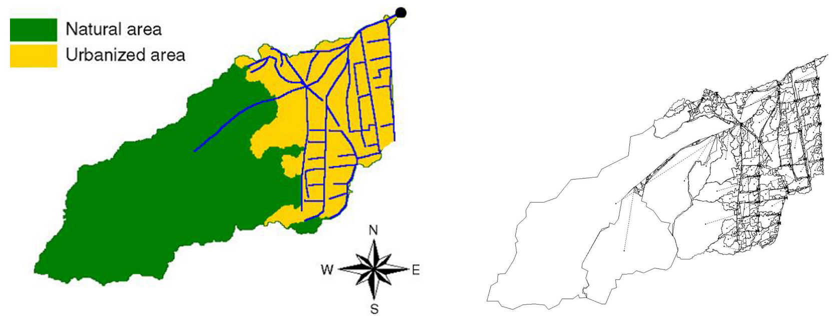
Figure 10: SWMM customization

the sensors are placed inside the network, i.e. it permits the physical variables which the external module is interested in to be chosen. Basically, at each step of SWMM simulation the values of the selected variables are collected and sent toward the external module which, in turn, replies to SWMM with all the actuation, i.e. the opening degree of the gates, as computed by the multi-agent algorithm.

3. Results and discussion

3.1. The case study: the sewer system of the city of Cosenza

In this study, the proposed approach is applied to an urban catchment in Cosenza, Italy, shown in Figure 11, which is densely populated (approximately 50,000 residents). The total surface of the catchment is 414 hectares, out of which 202 hectares are pervious, covered by vegetation, while the rest consists of paved areas. Buildings for residential housing and minor commercial and artisan enterprises are present on the catchment. Further details on the physical characteristics of the basin and the drainage system are reported in other publications [1]. Sewage and wet weather flows from the urban watershed, which do not exceed treatment capacity, are directly sent to the wastewater treatment plant (WWTP). The exceedance of wet weather flows is directly discharged, without receiving any treatment, into the Crati River. The dry weather flow of



(a) Urban watershed in the city of Cosenza

(b) SWMM model of Urban drainage network in the city of Cosenza

Figure 11: Cosenza, Italy

the urban catchment is approximately 0.23 m^3/s. The CSO occurs when the flow rate from the outfall of the drainage system is higher than 0.7 m^3/s as reported in [23][3] where other information about the untreated CSO features can be found.

The model used in this study was previously calibrated on the basis of several measurement campaigns [4]. Calibration parameters were: surface roughness of the impervious (N-Imperv) and pervious (N-Perv) catchment surfaces, and the depths of surface depressions on impervious (Dstore-Imperv) and pervious (Dstore-Perv) areas. The urban drainage network modelled in SWMM, as shown in Figure 11(b), consists of 324 conduits with different shapes and sizes, i.e., (i) circular and egg-shaped pipes with diameters varying from 0.3 to 1.5 m and (ii) policentric pipes with a maximum depth of 3.20 m. The slope of the pipes varies from 0.5% to 6%. The connection points between urban watershed surfaces (roads and street paving) and conduits are represented by catch basins which collect and deliver surface runoff into the sewer system. There are in total 326 nodes which represent the catch basins. The total number of subcatchments is 296.

The moveable gate is modelled as a transverse weir with the opening area equal to the conduit section area.

3.2. Experimental setup

In this study, different scenarios have been analyzed to evaluate the performance of the DRTC as a function of the number of the moveable gates, linked to actuators, and placed across the system. The scenario without DRTC, which corresponds to the actual UDS, is called *scenario 0*. The other scenarios are controlled by the DRTC and differ according to the number of secondary pipes equipped with moveable gates. The number of moveable gates is 91 for *scenario 1*, 107 for *scenario 2*, 214 for *scenario 3* and 322 for *scenario 4*. Scenario 1 corresponds to the gates' placement described in Section 2.3. In scenarios 2-4 an increasing number of gates is exploited which should result in a further CSO reduction as mentioned in Section 2.4.4. The response of the UDS for all these scenarios is modelled for 15 independent rainfall events recorded in the weather station in Cosenza (Italy) during the years 2010-2015 with a time resolution of 1 minute. The hydrological characteristics of the rainfall events selected are reported in table 1. Rainfall events were considered independent if they are separated by an inter-event time of 6 hours [24].

3.3. Experimental results

In the following section, the findings obtained from the DRTC applied to the case study are described and discussed. Firstly, results in terms of CSO and Flooding reductions are presented for scenarios 1 and 4, using the set of 15 rainfall events; afterwards, the behaviour of the controlled network is analysed in details for 3 representative rainfall events taking into account all the scenarios (0-4). Finally, consideration about the overall performance of the developed approach are drawn in the light of the study's goals. The CSO and the local flooding volumes, as given by the SWMM simulator, are reported in table 1 for the 15 rainfall events selected and for scenario 0 (without DRTC) used in this section as the reference scenario. The CSO volumes, computed as the sum of overflow spilled into the river, are fairly significant, ranging from 3489 m^3 to 66402 m^3. The local flooding volumes, calculated as the sum of spilled volume for each manhole of the UDS, range from 95 to 1007 m^3. In table 2, the CSO

Table 1: Hydrological and hydraulic characteristics of the selected rainfall events.

Event	**Date**	**htot (mm)**	**Iav (mm/h)**	**Qmax (m3/s)**	**Qav (m3/s)**	**CSO Vol (m3)**	**LF Vol (m3)**
1	22-Jan-11	31	1.29	2.06	0.64	23530	348
2	08-Oct-11	48.6	2.022	6.3	1.07	66402	933
3	05-Dec-11	54	2.25	4.85	1.32	53351	1007
4	01-Dec-13	31.4	2.76522	4.74	0.713	26790	366
5	23-Nov-13	46.6	3.33912	5.58	1.081	50216	795
6	21-Jan-14	31.8	1.32	2.8	0.69	17092	362
7	01-Feb-14	14.5	1.74228	2.8	0.83	11591	345
8	01-Feb-14	16.2	1.01298	2.98	0.6	8167	
9	24-Mar-14	38.6	1.60836	4.27	0.87	33309	563
10	30-Jan-15	42.8	1.78332	5.45	1	37039	680
11	31-Jan-15	4.6	1.53906	2.89	0.78	18478	565
12	31-Jan-15	35.38	2.24742	3.2	1.173	9442	
13	01-Feb-15	20.8	1.29738	3.57	0.64	14692	281
14	01-Feb-15	8.32	1.01568	2.98	0.49	3489	
15	22-Feb-15	18.91	0.78822	1.99	0.405	6080	95

htot	Total rainfall depth
Iav	Average rainfall intensity
Qmax	Maximum flow rate
Qav	Average flow rate
CSO Vol	Combined sewer overflow volume
LF Vol	Local flooding volume

Table 2: CSO and flooding reduction for scenarios 1 and 4

<table>
<tr><th rowspan="3">Event</th><th rowspan="3">Date</th><th colspan="4">CSO</th><th colspan="4">Flooding</th></tr>
<tr><th colspan="2">1</th><th colspan="2">4</th><th colspan="2">1</th><th colspan="2">4</th></tr>
<tr><th>(m3)</th><th>%</th><th>(m3)</th><th>%</th><th>(m3)</th><th>%</th><th>(m3)</th><th>%</th></tr>
<tr><td>1</td><td>22-Jan-11</td><td>16742</td><td>28.9</td><td>10965</td><td>53</td><td>0</td><td>100</td><td>320</td><td>8.1</td></tr>
<tr><td>2</td><td>08-Oct-11</td><td>64614</td><td>2.7</td><td>58030</td><td>13</td><td>0</td><td>100</td><td>808</td><td>13.4</td></tr>
<tr><td>3</td><td>05-Dec-11</td><td>43600</td><td>18.3</td><td>34244</td><td>36</td><td>0</td><td>100</td><td>971</td><td>3.6</td></tr>
<tr><td>4</td><td>01-Dec-13</td><td>20968</td><td>21.7</td><td>15582</td><td>42</td><td>0</td><td>100</td><td>333</td><td>9.0</td></tr>
<tr><td>5</td><td>23-Nov-13</td><td>39950</td><td>20.4</td><td>31785</td><td>37</td><td>0</td><td>100</td><td>7</td><td>99.1</td></tr>
<tr><td>6</td><td>21-Jan-14</td><td>8167</td><td>52.2</td><td>2617</td><td>85</td><td>0</td><td>100</td><td>356</td><td>1.7</td></tr>
<tr><td>7</td><td>01-Feb-14</td><td>7865</td><td>32.1</td><td>4718</td><td>59</td><td rowspan="2">0</td><td rowspan="2">100</td><td rowspan="2">324</td><td rowspan="2">6.1</td></tr>
<tr><td>8</td><td>01-Feb-14</td><td>5776</td><td>29.3</td><td>3596</td><td>56</td></tr>
<tr><td>9</td><td>24-Mar-14</td><td>25442</td><td>23.6</td><td>19497</td><td>41</td><td>0</td><td>100</td><td>518</td><td>8.0</td></tr>
<tr><td>10</td><td>30-Jan-15</td><td>30311</td><td>18.2</td><td>23962</td><td>35</td><td>0</td><td>100</td><td>664</td><td>2.4</td></tr>
<tr><td>11</td><td>31-Jan-15</td><td>13060</td><td>29.3</td><td>9223</td><td>50</td><td rowspan="2">0</td><td rowspan="2">100</td><td rowspan="2">532</td><td rowspan="2">5.8</td></tr>
<tr><td>12</td><td>31-Jan-15</td><td>8086</td><td>14.4</td><td>6629</td><td>30</td></tr>
<tr><td>13</td><td>01-Feb-15</td><td>10423</td><td>29.1</td><td>6859</td><td>53</td><td rowspan="2">0</td><td rowspan="2">100</td><td rowspan="2">265</td><td rowspan="2">5.7</td></tr>
<tr><td>14</td><td>01-Feb-15</td><td>1269</td><td>63.6</td><td>338</td><td>90</td></tr>
<tr><td>15</td><td>22-Feb-15</td><td>1035</td><td>83.0</td><td>44</td><td>99</td><td>0</td><td>100</td><td>90</td><td>5.3</td></tr>
</table>

reduction is computed for each rainfall event as the relative percentage difference between the CSO volume in the scenarios with DRTC and the reference scenario. In scenario 1, the CSO reduction varies from 2.7% to 83%, while in scenario 4 it varies from 13% to 99%, according to the rainfall events. The CSO drop is consistently higher in the scenario 4, demonstrating the beneficial effect provided by using a larger number of moveable gates.

Regarding local flooding, the results are reported in Table 2, as well. The local flooding reduction is computed as the relative percent difference between the total flooding volumes from the scenarios with DRTC and the reference scenario. The DRTC in scenario 1 utterly prevents the UDS from local flooding, through the temporary stormwater detention provided in the less overloaded

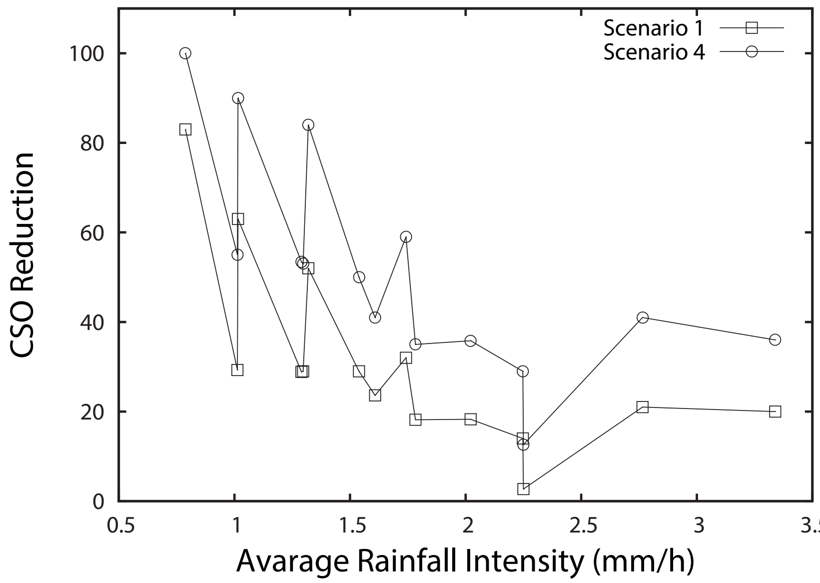


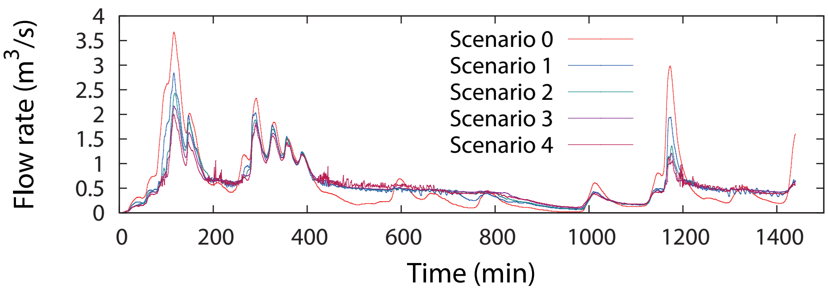
Figure 12: CSO reduction vs. rainfall event average intensity.

conduits. However, in the scenario 4 the risk of flooding is solely mitigated, with reductions, which vary from 2.4% to 13.4% for all the events, except for the event 23 November 2013, where a drop of 100% is obtained. As detailed later in this section, the reasons why scenario 4 offers a limited flooding reduction is strongly related to the high number of gates adopted.

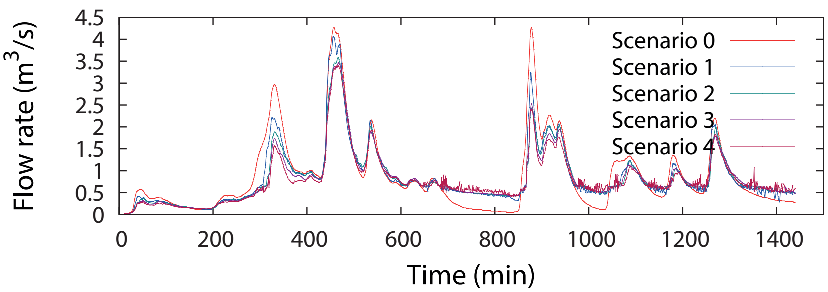
To investigate the effect of rainfall characteristics on the DRTC performance, the CSO reduction values for scenarios 1 and 4 are reported in Figure 12 as a function of the average rainfall intensity. As can be observed, the CSO reduction diminishes, as the average rainfall intensity increases. Specifically, for rainfall events with an average intensity less than 1.6 mm/h, the CSO reduction ranges from 100% to 40% for scenario 4 and from 85% to 25% for scenario 1. For events with an average intensity higher than 1.6 mm/h, the CSO reduction is around 30% on average, for scenario 4 and 10% on average, for scenario 1. These percentages correspond to the maximum storage capacity available in the different scenarios for detaining a portion of stormwater, in order to reduce CSOs, during rainy events. The trend observed is justified by the fact that very intense rainfall events such as, those on 11th October 2011 and 31st January 2015, with an average intensity of 2.2 and 2.24 mm/h, respectively, produce a large amount of runoff volume, which tends to utilize, almost completely, the whole capacity of the UDS. In such conditions, therefore, the likelihood of using the UDS as a temporary storage drastically decreases. Although the DRTC performance is affected by the hydrological characteristics of the events, the relative performance, computed as the difference between scenario 1 and scenario 4, turns out to be independent of rainfall characteristics. Indeed, the difference in the results between scenario 1 and scenario 4 are consistently around 20% and 30%, regardless of the average intensity.

In Figure 13, the time distribution of flow rate, discharged in the collector pipe, modelled with SWMM, is plotted for the five different scenarios investigated and three selected events: 24th March 2014, 30th January 2015, 1st February 2015[3]. The hydrographs, obtained for the reference scenario, exhibit a peak flow rate of 4.27, 5.45 and 3.57 m^3/s, respectively for the three events. Instead, the hydrographs show lower peak flow rates as the number of moveable gates increases. Indeed, the higher the number of moveable gates, the higher the effect of flow equalization provided by the DRTC. This is because the presence of moveable gates controlled by the DRTC enhances two major factors: (i) the storage of stormwater volume in the less overloaded pipes, with the results of abating the peak flow rates; (ii) the gradual release of the stored volumes to the outfall, once the rain stops. For instance, in the hydrograph obtained on 1st February 2015, reported in Figure 13(a), it is possible to observe that, comparing scenario 0 and scenario 4, the peak flow rate is reduced by around 40%. Furthermore, once the first event is over, the flow rate in scenario 0 rapidly decreases, tending to the minimum value. The flow rate in scenario 4, instead, diminishes more gradually, remaining fairly higher than the previous one. This is because once the rainfall event stops, the UDS in scenario 4 releases the

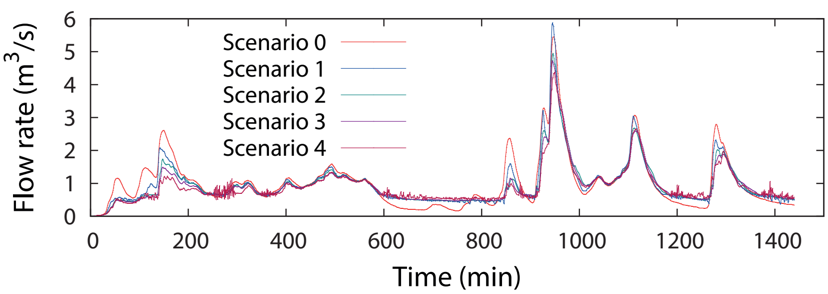
[3] There are two independent rainfall events on 1st February 2015. The experimental results are given considering both events



(a) 1st February 2015



(b) 24th March 2014



(c) 30th January 2015

Figure 13: Flow Rate vs. Time using Scenarios 0-4

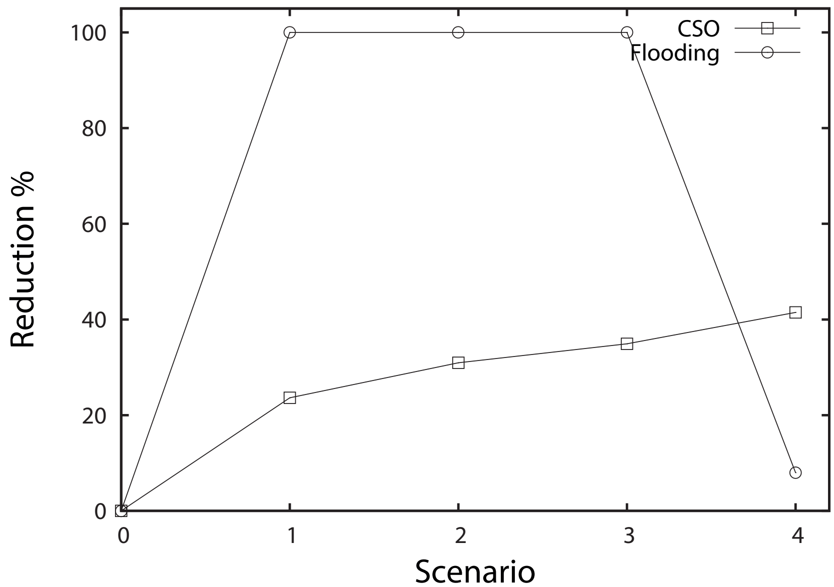


Figure 14: CSO and Flooding reduction vs. Scenario using the rainfall event of 24th March 2014.

volume previously stored during the rainfall event. The UDS empties almost completely (except for the sewage contribution) at time 1000 min, when the second event begins. A similar behavior can be observed in Figures 13(b) and 13(c) for 24th March 2014 and 30th January 2015. Moreover, as described in Section 3.1 occurs when flow rates are higher than 0.7 m^3/s. As can be observed from all the hydrographs reported in Figure 13, the DRTC tends, in fact, to maintain the flow rate in the collector pipe below this target value, with the beneficial consequence of reducing CSO.

Figures 14,15,16 show the reduction of local flooding and CSO as a function of the five scenarios for the events of 24th March 2014, 30th January 2015, 1st February 2015, respectively. In these figures, local flooding and CSO reductions are 0 for scenario 0, as it is the reference scenario. In Figure 14, it can be observed that for the event of 24th March 2014, scenario 1 reduces the CSO by 23.6%, with respect to scenario 0. As the number of gates increases, the reduction percentage rises, reaching a value of 41% for scenario 4. Regarding

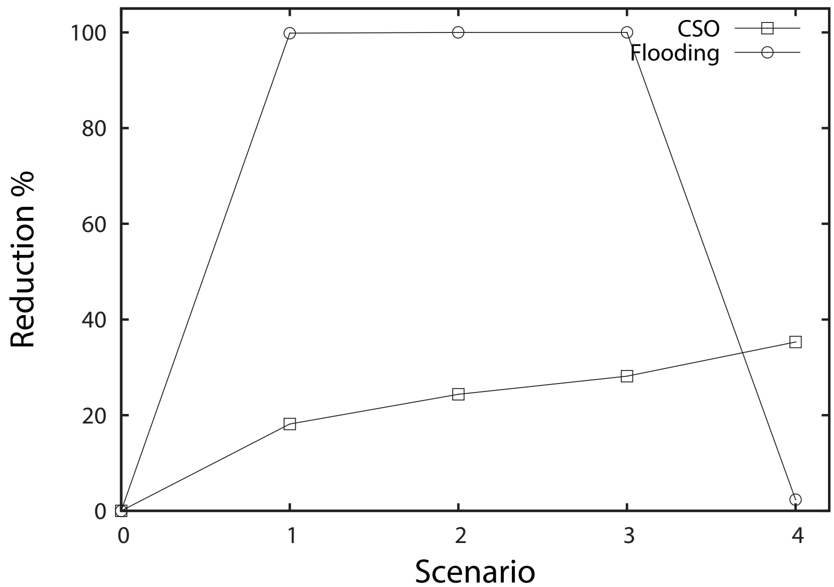


Figure 15: CSO and Flooding reduction vs. Scenario using the rainfall event of 30th January 2015.

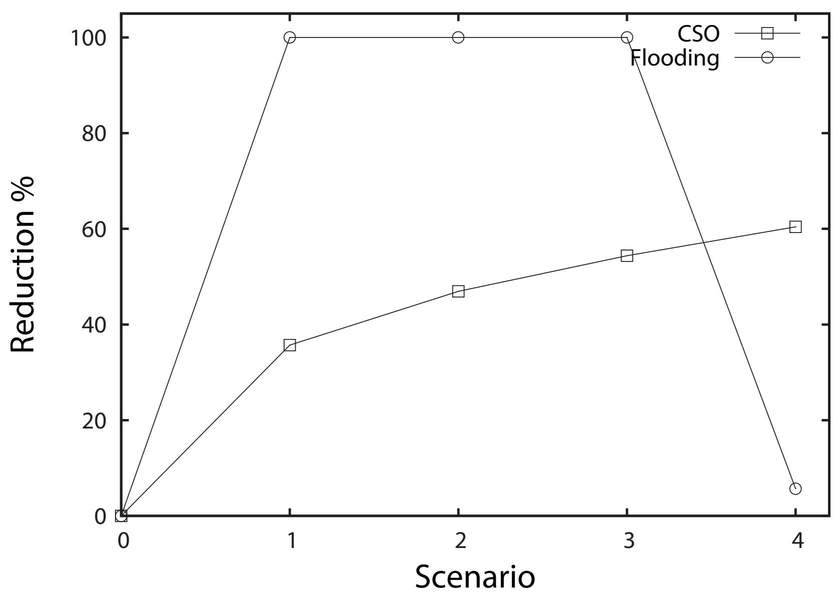


Figure 16: CSO and Flooding reduction vs. Scenario using the rainfall event of 1st February 2015.

local flooding, the reduction is 100% for all the controlled solutions, except for scenario 4. Similar results are obtained for the other two selected events (see Figures 15 and 16). In particular, the flooding reduction trend is basically the same as the previous case, while CSO reduction span from 18.2% (scenario 1) to 35% (scenario 4) for the event of 30th January 2015, and from 35.7% to 60% for the event of 1st February 2015.

Scenario 4 is the best choice for CSO reduction but it performs quite badly in terms of flooding reduction with respect to the other controlling scenarios. This behaviour is related to the high number of gates deployed in the network when scenario 4 is adopted. Indeed, this high number of gates is able to exploits all the possible storage capacity of the network in order to prevent the CSO. The latter is witnessed by the reduction of CSO in the experiment. Anyway, when the whole storage capacity is exploited, no additional water can be stored temporarily, and so a growth in incoming water flows produces unavoidable flooding phenomena. Conversely, in the other scenarios, there are portions of the network which are not instrumented by any gates. As a result, when the outfall is on the verge of experiencing CSO, there are less gates (with respect to scenario 4) that can act to store additional water volumes effectively in order to reduce the outfall water level. The latter behaviour is witnessed by a lower CSO reduction of scenarios 1-3 with respect to the scenario 4. Anyway, the partially unloaded network portions, not controlled by any gate, can offer a temporary storage capacity to additional water entry thus mitigating the flooding phenomena.

Summing up, these findings suggest that scenarios 2 and 3 are the most convenient solutions, since they offer the highest overall performance in terms of reduction of local flooding and perform well also with respect to the CSO reduction.

3.4. Comparison with other approaches

It is difficult to compare the performance of both centralized and distributed approaches among different case studies, since the results are site-specific and strongly determined by the characteristics of the rainfall events. Moreover, the advantages provided by our fully distributed approach offer some inherent properties such as adaptivity and fault tolerance as better detailed in Section 2.4.3. Anyway, a coarse-grained comparison with some well-known approaches in the literature is supplied in the following by considering the CSO volume reduction. For this purpose, the results obtained by our approach have been aggregated for the entire period of observation in which the distributed RTC for scenario 1 produces a CSO volume reduction of 29.5% and a local flooding reduction of 85.7% on average, while for scenario 4 it generates a CSO volume reduction of 50% and local flooding reduction of 14% on average.

In [25], the authors tested and designed a centralized real time control (RTC) strategy for the combined sewage system in the city of Kolding. The drainage area in the city of Kolding (Denmark) covers approximately 1300 ha. The same drainage system also manages the sewage water coming from an additional drainage area of 2000 ha. The drainage network is equipped with 16 control locations consisting in detention basins with capacities going up to 31 mm of rain, 7 of which are controlled by a gate. The strategy is based on a set of simple rules which check the degree of filling in each storage basin. The overall control strategy aims to have the same buffer volume in all the storage basins at the same time. The experimental part was carried out considering a set of rain events occurred in ten years (2000-2010) and shows a potential reduction up to 40% of the water volumes discharged into the Kolding River on an annual basis. This performance percentage is lower than the average value obtained with the decentralized approach presented in this study, which provides a higher CSO volume reduction of 50% for the scenario 4. It is worth noting that, differently from the latter study, no detention basins are exploited in our approach. Since structural changes of the existing network system are not required, our approach definitely provides a more convenient and feasible solution from an economic point of view. Furthermore, while our experiments focus on the most critical rain events occurred between 2011 and 2015, the approach described in [25] considers all the events from the selected period, even those with a limited intensity.

In [10] and [5] the authors describe a centralized RTC used in the Westerly sewer network in Quebec which covers 500 km^2 with a population of approximately 500,000 inhabitants. The sewer system is equipped with 5 electronically movable gates. The RTC system is designed with a two-level architecture. The upper level is composed of a central station, which computes flow set points, whereas the lower level is composed of local stations used for monitoring and feedback control. The central software computes every 5 minutes the optimal flow set points to be applied at the five control stations. The optimum is achieved by solving a non-linear multi-objective optimization problem with the goal of minimizing the overflows, maximizing the accumulated volumes and minimizing the variations of the setpoints. Experiments were carried out on seven rainfall events occurred between August 2013 and October 2014 showing a CSO reduction up to 87%. However, the rainfall events considered in this study have a total rainfall depth, varying from 4.3 mm to 18.3mm, much lower than that of the events considered in the present study, ranging from 8 to 54 mm. The authors of [10] also describe the behaviour of the system when a communication failure between local and central station occurs. In this case the local station applies a "default" behaviour in order to keep the system working. This kind of fault tolerance mechanism is, though, less efficient with respect to that of our approach. Indeed, as better detailed in section 2.4.3, in the case of local failure, suitable hydraulic conditions are guaranteed by the influence of the neighbour nodes.

In [12] a centralized RTC system is studied and simulated on the King County combined sewer system, Seattle, Washington, USA covering 26000 ha. The system is equipped with 11 pumps and 17 gates. The RTC system is based on a neural algorithm for minimizing CSO. Ten rainfall events have been used to train the neural network and 1 rainfall event has been used to test the approach, achieving a global CSO reduction of about 68%. Obviously, this single event result can not be compared with our aggregated CSO reduction value. Indeed, if we consider the results of the single events separately, we obtain better percentage of CSO reduction for some events such as: the event 6 (85%), the event

14 (90%) and the event 15 (99%).

As a final remark, it is important to underline that all of the reported approaches focus only on the CSO reduction while the proposed approach aims to reduce both flooding and CSO.

4. Conclusion

In this study, a decentralized real time control (DRTC), based on a multi-agent paradigm and specifically a gossip-based algorithm, has been developed and integrated with the hydrodynamic simulation model, SWMM. The DRTC proposed has been applied to the urban drainage system (UDS) in the city of Cosenza, Italy. The UDS, modelled in SWMM, is equipped with a series of moveable gates, functioning as actuators, and sensors, which monitor water level in each conduit. The findings show that the DRTC algorithm proposed was able to balance the hydraulic capacity of the conduits within the system by utilizing the storage capacity of the less overwhelmed conduits during intense rainfall events. In other words, the DRTC algorithm was able to control the water level within the UDS successfully, ensuring a full utilization of the actual storage capacity of the system. The findings clearly demonstrated that the DRTC produced beneficial effects on the management of the UDS by substantially mitigating the risk of flooding and CSO.

Future research will test the response of the DRTC not only to varying hydrological inputs, but also to changing boundary conditions, for example, sudden pipe breakage or increased clogging in the catch basins, to demonstrate the high adaptability of the algorithm to different, even unforeseen, situations. Furthermore, the actual costs of the physical implementation and the energy consumption required will be further investigated.

Acknowledgments

This work has been partially supported by RES-NOVAE - “Buildings, roads, networks, new virtuous targets for the Environment and Energy” project, funded

by the Italian Government (PON 04a2_E).

Vitae

Giuseppina Garofalo is a Research Fellow at the University of Calabria, Italy. She pursued her Ph.D. in Environmental Engineering Sciences at the University of Florida, USA in 2012. Her research interests are numerical and physical modelling of stormwater unit operations and sustainable water systems.

Andrea Giordano is a Researcher at the High Performance Computing and Networking Institute of the Italian National Research Council (ICAR-CNR) Rende, Italy. He pursued his Ph.D. in computer science and systems engineering at the University of Calabria, Italy in 2011. His research interests include multi-agent systems, swarm intelligence, parallel and distributed systems, parallel discrete event simulation.

Patrizia Piro is a Full Professor in Hydrology and Hydraulic Constructions at the University of Calabria, Italy. Her research interests are the management of combined sewer system overflows and urban flooding, monitoring and modelling urban drainage systems, sustainable water solutions, smart water grids.

Giandomenico Spezzano is a Research Director at the Institute of High Performance Computing and Networking of the Italian National Research Council (ICAR-CNR) Rende, Italy. He pursued his Ph.D. in computer science and systems engineering at the University of Calabria, Italy in 2011. His research interests include parallel architectures, grid computing, peer-to-peer computing, parallel and distributed data mining, pervasive computing and internet of things

Andrea Vinci is a Research Fellow at the High Performance Computing and Networking Institute of the Italian National Research Council (ICAR-CNR) Rende, Italy. Currently, He is also a Ph.D. candidate in computer science and systems engineering at the University of Calabria. His research interests include multi-agent systems, swarm intelligence, cyber-physical systems, internet of things and GP-GPU computing.

References

[1] P. Piro, "Il bacino sperimentale urbano del canale liguori nella città di cosenza. osservazioni sperimentali quali-quantitative nel periodo 1995–2003," *Bios*, 2007.

[2] P. Piro, M. Carbone, G. Garofalo, and J. Sansalone, "Size distribution of wet weather and dry weather particulate matter entrained in combined flows from an urbanizing sewershed," *Water, Air, and Soil Pollution*, vol. 206, no. 1-4, pp. 83–94, 2010.

[3] P. Piro, M. Carbone, and G. Garofalo, "Distributed vs. concentrated storage options for controlling cso volumes and pollutant loads," *Water Practice & Technology*, vol. 5, no. 3, 2010.

[4] P. Piro and M. Carbone, "A modelling approach to assessing variations of total suspended solids (tss) mass fluxes during storm events," *Hydrological Processes*, vol. 28, no. 4, pp. 2419–2426, 2014.

[5] M. Schütze, A. Campisano, H. Colas, W. Schilling, and P. A. Vanrolleghem, "Real time control of urban wastewater systemswhere do we stand today?," *Journal of hydrology*, vol. 299, no. 3, pp. 335–348, 2004.

[6] P. M. Bach, W. Rauch, P. S. Mikkelsen, D. T. McCarthy, and A. Deletic, "A critical review of integrated urban water modelling–urban drainage and beyond," *Environmental Modelling & Software*, vol. 54, pp. 88–107, 2014.

[7] T. Beeneken, V. Erbe, A. Messmer, C. Reder, R. Rohlfing, M. Scheer, M. Schuetze, B. Schumacher, M. Weilandt, and M. Weyand, "Real time

control (rtc) of urban drainage systems–a discussion of the additional efforts compared to conventionally operated systems," *Urban Water Journal*, vol. 10, no. 5, pp. 293–299, 2013.

[8] G. Dirckx, M. Schütze, S. Kroll, C. Thoeye, G. De Gueldre, and B. Van De Steene, "Rtc versus static solutions to mitigate csos impact," in *12nd International Conference on Urban Drainage, 2011b. Porto Alegre, Brazil*, 2011.

[9] S. Achleitner, M. Möderl, and W. Rauch, "City drain©–an open source approach for simulation of integrated urban drainage systems," *Environmental modelling & software*, vol. 22, no. 8, pp. 1184–1195, 2007.

[10] M. Pleau, H. Colas, P. Lavallée, G. Pelletier, and R. Bonin, "Global optimal real-time control of the quebec urban drainage system," *Environmental Modelling & Software*, vol. 20, no. 4, pp. 401–413, 2005.

[11] G. Fu, D. Butler, and S.-T. Khu, "Multiple objective optimal control of integrated urban wastewater systems," *Environmental Modelling & Software*, vol. 23, no. 2, pp. 225–234, 2008.

[12] S. Darsono and J. W. Labadie, "Neural-optimal control algorithm for real-time regulation of in-line storage in combined sewer systems," *Environmental Modelling & Software*, vol. 22, no. 9, pp. 1349–1361, 2007.

[13] L. Vezzaro and M. Grum, "A generalised dynamic overflow risk assessment (dora) for real time control of urban drainage systems," *Journal of Hydrology*, vol. 515, pp. 292–303, 2014.

[14] L. A. Rossman and W. Supply, *Storm water management model, quality assurance report: dynamic wave flow routing.* US Environmental Protection Agency, Office of Research and Development, National Research Management Research Laboratory, 2006.

[15] M. Moderl, M. Kleidorfer, and W. Rauch, "Influence of characteristics on combined sewer performance," *Water Science & Technology*, vol. 66, no. 5, pp. 1052–1060, 2012.

[16] M. Carbone, G. Garofalo, and P. Piro, "Decentralized real time control in combined sewer system by using smart objects," *Procedia Engineering*, vol. 89, pp. 473–478, 2014.

[17] F. Dötsch, J. Denzinger, H. Kasinger, and B. Bauer, "Decentralized real-time control of water distribution networks using self-organizing multi-agent systems," in *Self-Adaptive and Self-Organizing Systems (SASO), 2010 4th IEEE International Conference on*, pp. 223–232, IEEE, 2010.

[18] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[19] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford university press, 1999.

[20] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," *ACM Transactions on Computer Systems (TOCS)*, vol. 23, no. 3, pp. 219–252, 2005.

[21] K. J. Astrom, "Pid controllers: theory, design and tuning," *Instrument society of America*, 1995.

[22] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *trans. ASME*, vol. 64, no. 11, 1942.

[23] P. Piro, M. Carbone, G. Garofalo, and J. Sansalone, "Cso treatment strategy based on constituent index relationships in a highly urbanised catchment," *Water Science & Technology*, vol. 56, no. 12, pp. 85–91, 2007.

[24] B. J. Adams and F. Papa, "Urban stormwater management planning with analytical probabilistic models," *Canadian Journal of Civil Engineering*, vol. 28, no. 3, p. 545, 2001.

[25] N. H. Nielsen, C. Ravn, and N. Mølbye, "Implementation and design of a rtc strategy in the sewage system in kolding, denmark," *Proceedings of 7th Novatech*, 2010.