Computational Logic in Hydroinformatics: the Case of the Optimal Location of Isolation Valves

Marco Gavanelli, Maddalena Nonato, and Andrea Peano

EnDiF, Università degli Studi di Ferrara
via G. Saragat 1 – 44122, Ferrara, Italy
{marco.gavanelli, maddalena.nonato, andrea.peano}@unife.it

1 Hydroinformatics

By the marriage of hydraulic engineering and computer science, a new scientific discipline was recently spawn, sometimes called *Hydroinformatics*. It aims to address problems in the hydraulic engineering literature through computational methods, and the interest it gets is witnessed by the many large conferences (such as the International Conference on Hydroinformatics) and well renowned journals (like the Journal of Hydroinformatics) that are focussed on these issues.

In many cases, the problems have constraints and objective functions; often hydraulic engineers approach these problems with genetic algorithms, because they have wide applicability, they interface easily with hydraulic simulators, they are easy to understand even for non-experts, and they are already available in the tools widely used by hydraulic engineers, such as MatLab.

For example, hydraulic engineers are interested to design a new water distribution network: given the topology of the network, they want to define the size of each pipe, satisfying a number of constraints, such as minimum pressure in each pipe. This problem has been addressed by means of genetic algorithms, possibly combined with linear programming [13,12,5]. Also, pure nonlinear programming optimisation approaches were developed [3].

Another interesting problem is the following: suppose that a terrorist poisons the drinking water in a hydraulic network; how can we effectively react?

First, we should place a set of sensors to detect as soon as possible the presence of the poison. But sensors are expensive, and the budget is limited, so the number of sensors cannot be very high: how do we place the sensors in such a way that we are able to detect the presence of poison in most (if not all) the possible contamination scenarios?

One way is to simulate (through a hydraulic simulator, e.g., Epanet [17]) the spreading of the contaminant in the network, and run a genetic algorithm that decides where the sensors should be placed in the network. Each individual in the population carries in its genes the information on the positioning of the sensors; in order to compute its fitness, we can run the hydraulic simulator for each of the contamination scenarios, and then count in how many scenarios the contaminant is detected (or have more objective functions [11]). But each hydraulic simulation takes some seconds (for the size of a real network), and scenarios can be many, so the computing time grows very quickly. Other approaches use greedy algorithms, exploiting some features of this problem, such as *submodularity* [14], or mixed integer linear programming approaches [18,2].

After placing the sensors, one should decide how to react to contamination (in case it is detected). Of course, one could close immediately the entire network, but it is not always feasible. For example, one cannot close the water to the fire department (what happens if the terrorist poisons the water network and lights a fire?). Another option is to have a reaction plan: depending on which sensor raised the alarm, one can close network patches (to deviate the poison to uninhabited zones) or open hydrants (to expel poisoned water); but this can only be done by sending teams of workers on the network to close valves and open hydrants. So, we have a scheduling/routing problem [8].

We strongly believe that the hydroinformatics community would benefit from computational logic approaches, so we recently started applying computation logics to one of the problems that hydraulic engineers in our department pointed to us. Before explaining our approaches, we define the problem.

2 The Isolation Valves Location Problem

A water distribution network can be thought of as a labelled indirected graph (see Figure 1), in which the edges represent the pipes in the network, and nodes are their junctions. The labels associated to the pipes represent the users demands, expressed in *litres per second* (l/s). There is at least one special node that represents the source of water (node T in Figure 1).

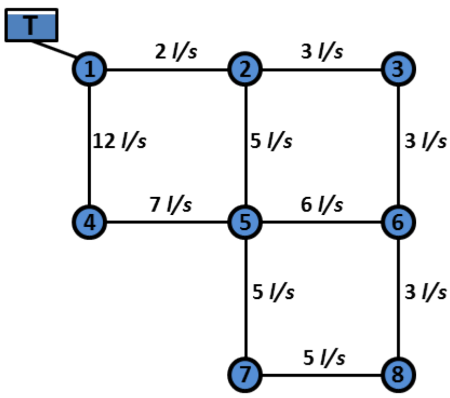

Fig. 1: An example of water distribution network

Any pipe of the network may get damaged; in this case a set of technicians is sent to fix the damaged pipe. First, they isolate a portion of the network by closing a set of *isolation valves*, then they fix the pipe and finally re-open the valves. During these works, the users that take water from the isolated pipes are without water.

So, in order to minimize service disruption, one should have a large number of valves (ideally, one at each end of each pipe). Unfortunately, isolation valves have installation and (even more) maintenance costs, so their number is limited. Thus, deciding an optimal placement of the available valves becomes a combinatorial optimisation problem. There are different proposals for the objective function to be optimized: one often used by hydraulic engineers [10] is the minimisation of the worst isolation case.

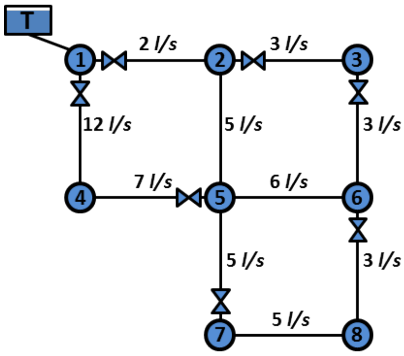

Fig. 2: A possible placement of 7 valves

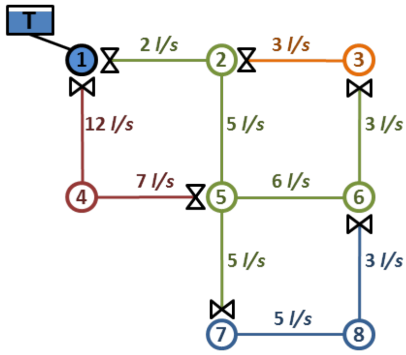

Fig. 3: The sectorization given the placement in Fig. 2

Figure 2 shows a possible placement when there are 7 valves available. The valves partition the network in the so-called *sectors* (Figure 3). Intuitively, each sector represents the minimal portion of the network that gets isolated when one of its pipes is de-watered. For instance, if the broken pipe is the one connecting nodes 2 and 3 (let us call it $p_{2,3}$), as shown in Figure 4, only one pipe is isolated, and the service disruption is only the user demand of this pipe, namely $3l/s$.

However, if the broken pipe is in the green sector (Figure 5), the service disruption is not only the demand of the sector itself ($21l/s$), because also the users in the blue sector ($8l/s$) and the orange sector ($3l/s$) are without water, and the total disruption is $21 + 8 + 3 = 32l/s$. This effect is called *unintended isolation*; due to this effect, the problem is much harder than the classical *graph partitioning problem*.

3 Computational Approaches

Hydraulic engineers often attack optimisation problems through Genetic Algorithms (GAs); in particular, multi-objective GAs have been proposed to address

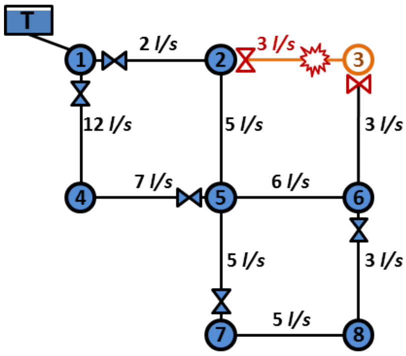

Fig. 4: A small isolation case

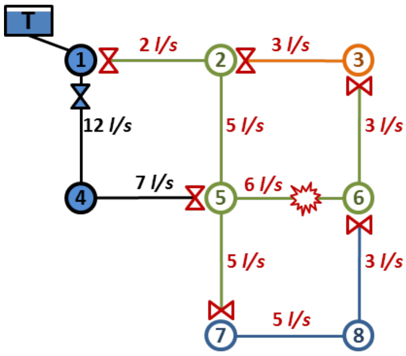

Fig. 5: Isolation case with some unintended isolations

the valves location problem [6,10]. Such algorithms compute the (near)Pareto-front of two objectives to minimise: the value of the undelivered demand, and the cost of the solutions (in terms of number of valves as well as their costs).

Of course, these works cannot guarantee to provide the exact optimal solution. The first exact algorithms was developed with computational logic tools, more precisely in Constraint Logic Programming on Finite Domains (CLP(FD)), and is based on a two-player game model [4]. In this formulation, player A chooses the positioning of the available valves, then player B selects the worst isolation case; finally, A closes the valves that isolate the selected pipe. The resulting minmax algorithm was enriched of several pruning procedures, in order to reduce the search space. The CLP(FD) program was tested on the Apulian water distribution network considered in the hydroinformatics literture [10], and it improved the state-of-the-art in hydraulic engineering for this problem, providing solutions with lower undelivered demand.

Then, the problem was also modelled in ASP (Answer Set Programming [1,15,9]). A preliminary overview is drawn in [7]. A first intuitive encoding follows closely the problem definition; it is based on a predicate `valve/2`; the atom

```
valve(A,B)
```

is true if there is a valve on pipe $p_{A,B}$, close to node A. Also, a valve can be closed or open; its state depends on which pipe is broken, so we have a predicate `closed_valve/2`, where

```
closed_valve(valve(A,B),broken(C,D))
```

is true iff the valve (A, B) is closed when the pipe $p_{C,D}$ is broken. Depending on which pipe is broken, we can then easily compute which pipes are reached by water; the atom

```
reached(pipe(A,B),broken(C,D))
```

is true if pipe $p_{A,B}$ is reached by water when the damaged pipe is $p_{C,D}$. In this way, one can compute the satisfied (and unsatisfied) water demand for each possible break situation, and is able to minimize the service disruption in the worst isolation case.

The problem with this encoding is that it has to compute if a pipe is reached by water for each possible *pipe* that can be broken. On the other hand, this is not always necessary; for example, in Figure 2, if the broken pipe is $p_{1,4}$ we have exactly the same isolation that we have when the broken pipe is $p_{4,5}$. In the game example, there is no point for the second player to try to damage all the possible *pipes*, but only one pipe for each *sector*.

So, we developed more efficient encodings (that get expanded in smaller ground programs) that rely on the concept of *sector*; we have a predicate `sector` that associates each pipe with its sector:

```
sector(pipe(A,B),S).
```

On the other hand, this introduces a large number of symmetries, because sectors names are interchangeable. So, one can try to introduce symmetry breaking constraints, that help improve the solution process.

Finally, we investigated the use of bilevel Mixed Integer Linear Programming (MILP) [16].

4 Comparison

The good news is that, so far, Computational Logic is still the state-of-the-art for the valve location problem. The winner is still the CLP(FD) formulation, developed in ECL^{i}PS^e; from the web page of the Valve Location Problem in CLP(FD) one can download the source code as well as the benchmark instance. Mixed Integer Linear Programming takes second place in the considered instances.

ASP formulations are close to MILP, but they still suffer from the fact that there is a large number of symmetries. We tried to reduce the number of symmetries through symmetry breaking constraints, and they provided a significant improvement. However, we were expecting a larger improvement: for example, by reducing the search space to one half, one does not get a speedup of 2, possibly because the symmetry breaking constraints partially disrupt the clever search strategies implemented in current ASP solvers.

But everybody in Computational Logic can contribute to change this ranking! The problem was accepted as benchmark for the 4th Answer Set Programming Competition, so ASP solvers will compete on an encoding of the problem, while everybody can compete in the Model&Solve competition, through any formulation, in any computational logic language or even with non logic-based tools.

References

1. Chitta Baral. *Knowledge representation, reasoning and declarative problem solving.* Cambridge University Press, 2003.
2. J. Berry, W.E. Hart, C.A. Phillips, and J.G Uber. A general integer-programming-based framework for sensor placement in municipal water networks. In *Proceedings of World Water and Environment Resources Conference*, 2004.
3. Cristiana Bragalli, Claudia DAmbrosio, Jon Lee, Andrea Lodi, and Paolo Toth. On the optimal design of water distribution networks: a practical minlp approach. *Optimization and Engineering*, 13:219–246, 2012.
4. Massimiliano Cattafi, Marco Gavanelli, Maddalena Nonato, Stefano Alvisi, and Marco Franchini. Optimal placement of valves in a water distribution network with CLP(FD). *Theory and Practice of Logic Programming*, 11(4-5):731–747, 2011.
5. E. Creaco and M. Franchini. Fast network multi-objective design algorithm combined with an a-posteriori procedure for reliability evaluation under various operational scenarios. *Urban Water Journal*, 9(6):385–399, 2012.
6. Enrico Creaco, Marco Franchini, and Stefano Alvisi. Optimal placement of isolation valves in water distribution systems based on valve cost and weighted average demand shortfall. *Water Resources Management*, 24:4317–4338, 2010. 10.1007/s11269-010-9661-5.
7. Marco Gavanelli, Maddalena Nonato, Andrea Peano, Stefano Alvisi, and Marco Franchini. An ASP approach for the valves positioning optimization in a water distribution system. In Francesca Lisi, editor, *9th Italian Convention on Computational Logic (CILC 2012), Rome, Italy*, volume 857 of *CEUR workshop proceedings*, pages 134–148, 2012.
8. Marco Gavanelli, Maddalena Nonato, Andrea Peano, Stefano Alvisi, and Marco Franchini. Genetic algorithms for scheduling devices operation in a water distribution system in response to contamination events. In Jin-Kao Hao and Martin Middendorf, editors, *Evolutionary Computation in Combinatorial Optimization*, volume 7245 of *Lecture Notes in Computer Science*, pages 124–135. Springer Berlin / Heidelberg, 2012. 10.1007/978-3-642-29124-1_11.
9. Michael Gelfond. Answer sets. In *Handbook of Knowledge Representation*, chapter 7. Elsevier, 2007.
10. Orazio Giustolisi and Dragan A. Savić. Identification of segments and optimal isolation valve system design in water distribution networks. *Urban Water Journal*, 7(1):1–15, 2010.
11. M. Guidorzi, M. Franchini, and S. Alvisi. A multi-objective approach for detecting and responding to accidental and intentional contamination events in water distribution systems. *Urban Water*, 6(2):115–135, 2009.

12. Ali Haghighi, HosseinM.V. Samani, and ZeinabM.V. Samani. Ga-ilp method for optimization of water distribution networks. *Water Resources Management*, 25:1791–1808, 2011.
13. A. Krapivka and A. Ostfeld. Coupled genetic algorithm linear programming scheme for least cost design of water distribution systems. *Journal of Water Resources Planning and Management Division*, 135(6):556–568, 2009.
14. Andreas Krause and Carlos Guestrin. Optimizing sensing: From water to the web. *Computer*, 42:38–45, 2009.
15. Nicola Leone. Logic programming and nonmonotonic reasoning: From theory to systems and applications. In Chitta Baral, Gerhard Brewka, and John Schlipf, editors, *Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'07)*, volume 4483 of *Lecture Notes in Computer Science*. Springer, 2007.
16. Andrea Peano, Maddalena Nonato, Marco Gavanelli, Stefano Alvisi, and Marco Franchini. A bilevel mixed integer linear programming model for valves location in water distribution systems. In *3rd Student Conference on Operational Research (SCOR 2012) Proceedings*, Dagstuhl OpenAccess Series in Informatics (OASIcs), 2012.
17. L.A Rossman. *EPANET 2 users manual.* National Risk Management Research Laboratory, Office of research and development, U.S. Environmental Protection Agency, USA.
18. J.P. Watson, H.J. Greenberg, and W.E. Hart. A multiple-objective analysis of sensor placement optimization in water networks. In *Proceedings of World Water and Environment Resources Conference*, 2004.