

Chapter 9

Optimizing the Design of Water Distribution Networks Using Mathematical Optimization

Cristiana Bragalli, Claudia D'Ambrosio, Jon Lee, Andrea Lodi
and Paolo Toth

1 Introduction

Decaying infrastructure in municipalities is becoming a problem of increasing importance as growing populations put increasing stress on all service systems. In tough economic times, renewing and maintaining infrastructure has become increasingly difficult. As an example, many municipal water networks were installed several decades ago and were designed to handle much smaller demand and additionally have decayed due to age.

For example, consider the case of Modena, a city northwest of Bologna in the Emilia-Romagna region of Italy. We can see in Fig. 9.1 how the population has increased by more than a factor of 2.5 over the last century. The Modena water distribution network comprises 4 reservoirs, 272 junctions, and 317 pipes. Its complexity can be gleaned from Fig. 9.2.

The aim is to replace all the pipes using the same network topology at minimum cost to achieve pressure demands at junctions of the network. Pipes are only available

The online version of this chapter (http://link.springer.com/book/10.1007%2F978-1-4939-1007-6_9) contains supplementary material, which is available to authorized users.

C. Bragalli (✉)
DISTART, University of Bologna, 40136 Bologna, Italy
e-mail: cristiana.bragalli@unibo.it

C. D'Ambrosio
CNRS LIX, École Polytechnique, 91128 Palaiseau, France
e-mail: dambrosio@lix.polytechnique.fr

J. Lee
IOE Department, University of Michigan, Ann Arbor, MI 48109-2117, USA
e-mail: jonxlee@umich.edu

A. Lodi · P. Toth
DEI, University of Bologna, 40136 Bologna, Italy
e-mail: andrea.lodi@unibo.it

P. Toth
e-mail: paolo.toth@unibo.it

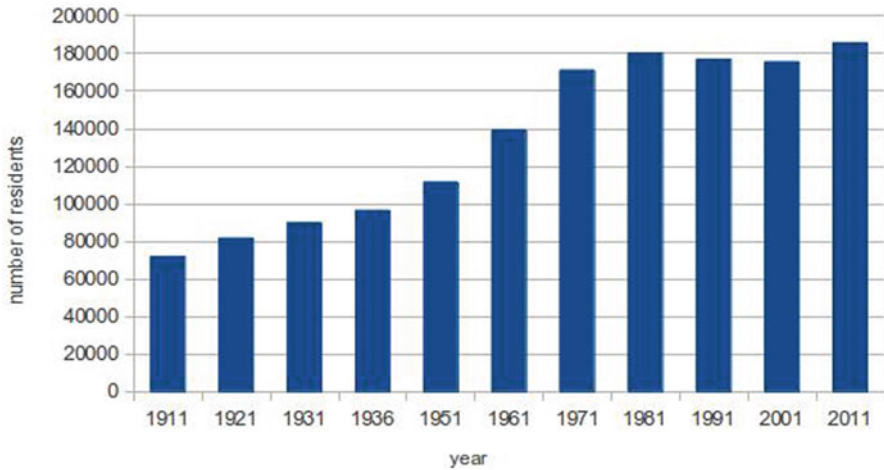


Fig. 9.1 Population of Modena



Fig. 9.2 Water distribution network of Modena

from commercial suppliers that produce pipes in a limited number of diameters. Reservoirs pressurize the network while most pressure is lost due to friction in pipes (some pressure is also lost at the junctions). Our goal is to model the problem using continuous variables as flow rates in the pipes, and pressures at the junctions; and discrete variables for the diameters of the pipes. Noting that pressure loss due to friction behaves nonlinearly, this puts us in the domain of mixed integer nonlinear programming (MINLP). Thus, we will try to obtain a model which is tractable by standard MINLP solvers.

In fluid dynamics, *hydraulic head* equates the energy in an incompressible fluid with the height of an equivalent static column of that fluid. The hydraulic head of a fluid is composed of pressure head and elevation head. The *pressure head* is the internal energy of a fluid due to the pressure exerted on its container and the *elevation head* is the energy given by the gravitational force acting on a column

of fluid, in this case water. As a convention, the units of hydraulic, pressure, and elevation head are meters, i.e., each unit represents the energy provided by a column of water of the height of 1 m. The water in its way from the reservoir to the rest of the junctions loses energy. This loss is called *head loss*. Head loss is divided into two main categories, “major losses” associated with energy loss per length of pipe, and “minor losses” associated with bends and relatively small obstructions. For our purposes, considering that we are interested in networks covering relatively large geographic areas, it suffices to ignore minor losses.

2 Nomenclature

In this section we introduce the sets and parameters, i.e., the data input of our problem.

2.1 Sets

The water network will be represented as a directed graph $G = (N, E)$ where, N is the set of nodes that represent the junctions and E is the set of pipes. Moreover, we define the set of reservoirs S as a subset of N . Finally, for ease of notation, we define $\delta_-(i)$ ($\delta_+(i)$) as the sets of pipes with an head (tail) at junction i .

2.2 Parameters

In the following, we introduce the notation for the data elements that we call parameters.

For each junction but the reservoirs $i \in N \setminus S$ we have:

- $\text{elev}(i)$ = physical elevation of junction i , i.e., the height of junction i ($[m]$).
- $\text{dem}(i)$ = water demand at junction i ($[m^3/s]$).
- $\text{ph}_{\min}(i)$ = lower bound on pressure head at junction i ($[m]$).
- $\text{ph}_{\max}(i)$ = upper bound on pressure head at junction i ($[m]$).

Moreover, for each reservoir $i \in S$, we have $h_s(i)$, i.e., the fixed hydraulic head at junction i .

Finally, for each pipe $e = (i, j) \in E$ the following parameters are needed:

- $l(e)$ = length of pipe e ($[m]$).
- $v_{\max}(e)$ = upper bound on the velocity of water in pipe e ($[m/s]$).
- $k(e)$ = physical constant depending on the roughness of pipe e .
- $\mathcal{D}(e, r)$ = r -th diameter that can be chosen for pipe e ($[m]$).
- $\mathcal{C}(e, r)$ = cost of the r -th diameter that can be chosen for pipe e ($[€/m]$).

Note that all the parameters described above are data input and are used to define constraints and objective function of the model.

3 The Mathematical Model for the Problem

We proceed further in describing the problem by specifying a formulation of it.

3.1 Decision Variables of the Problem

First, we specify the variables which are also the expected output of our problem:

- $Q(e)$ = flow in pipe e , for all $e \in E$ [m^3/s]. This is the volume of water which passes through pipe per unit time.
- $D(e)$ = diameter of pipe e , for all $e \in E$ [m].
- $H(i)$ = hydraulic head of junction i , for all $i \in N$ [m].

For modeling purposes, each pipe e has a nominal orientation, and negative flow corresponds to water flow in the direction opposite to the nominal orientation of the pipe.

The goals of the problem are to choose, for each edge, the diameter of the pipe that has to be installed in order to implement the drinkable water distribution network by minimizing the installation costs and satisfying physical and operational constraints.

3.2 Constraints in the Problem

Each reservoir has a fixed hydraulic head that is specified:

$$H(i) = h_s(i), \quad \text{for all } i \in S. \quad (9.1)$$

Typically, there are inequalities imposed that bound the velocity of the water in each pipe. Because the water flow is given by the product between the cross-sectional area of the pipe $\pi(D(e)/2)^2$ and the velocity, for simplicity the velocity bounds can be written as bounds on the flow:

$$-\frac{\pi}{4}v_{max}(e)D^2(e) \leq Q(e) \leq \frac{\pi}{4}v_{max}(e)D^2(e), \quad \text{for all } e \in E. \quad (9.2)$$

Notice how these equations are nonlinear in the diameter variables $D(e)$, but we can think of them as linear in the cross-sectional area $\pi(D(e)/2)^2$.

Next, we have the usual flow-conservation equations:

$$\sum_{e \in \delta_-(i)} Q(e) - \sum_{e \in \delta_+(i)} Q(e) = \text{dem}(i), \quad \text{for all } i \in N \setminus S, \quad (9.3)$$

where, $\text{dem}(i)$ is the demand for water at junction i , $\delta_-(i)$ denotes the set of pipes oriented *into* junction i , and $\delta_+(i)$ denotes the set of pipes oriented *out of* junction i . We only have these equations for junctions that are not reservoirs.

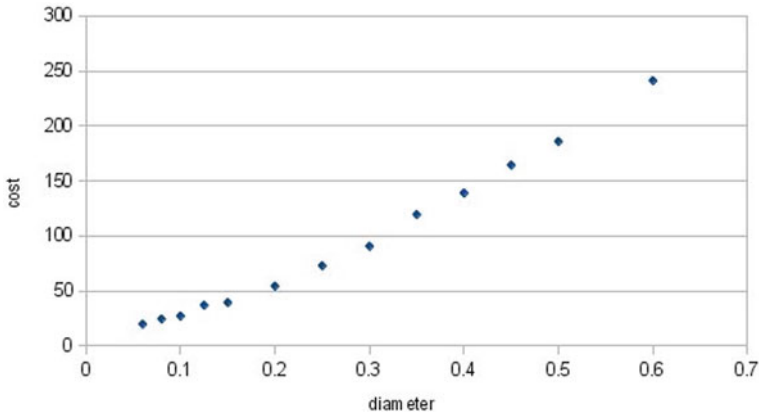


Fig. 9.3 Diameter set for Fossolo instance: diameter (m) and cost per meter ($\text{€}/m$)

Next, we impose upper and lower limits on the pressure head at each junction:

$$ph_{\min}(i) \leq H(i) - \text{elev}(i) \leq ph_{\max}(i), \quad \text{for all } i \in N \setminus S. \quad (9.4)$$

Head loss along each pipe is modeled via a nonlinear function of the diameter of the pipe and the flow in the pipe. Typically, one uses the so-called Hazen–Williams equation, which is an empirical formula relating the head loss caused by frictions to the physical properties of the pipe:

$$H(i) - H(j) = \frac{\text{sgn}(Q(e))|Q(e)|^{1.852} \cdot 10.7 \cdot l(e) \cdot k(e)^{-1.852}}{D(e)^{4.87}}, \quad \text{for all } e = (i, j) \in E, \quad (9.5)$$

where, we consider the pipe e to be oriented from junction i to junction j . Notice how following intuition head loss is proportional to the length $l(e)$ of the pipe e . The dependence on the flow $Q(e)$ and the diameter $D(e)$ is nonlinear.

For each pipe e , the available diameters belong to a discrete set of r_e elements. For $e \in E$:

$$d_{\min}(e) := \mathcal{D}(e, 1) < \mathcal{D}(e, 2) < \dots < \mathcal{D}(e, r_e) :=: d_{\max}(e),$$

and so, we have the equations:

$$D(e) \in \{\mathcal{D}(e, 1), \mathcal{D}(e, 2), \dots, \mathcal{D}(e, r_e)\}, \quad \text{for all } e \in E. \quad (9.6)$$

For each pipe $e \in E$, there is a cost function $C_e()$ having a discrete specification as a (typically rapidly) increasing function of diameter. That is, $\mathcal{C}(e, r) := C_e(\mathcal{D}(e, r))$, $r = 1, \dots, r_e$, where:

$$\mathcal{C}(e, 1) < \mathcal{C}(e, 2) < \dots < \mathcal{C}(e, r_e).$$

Our objective to be minimized is:

$$\sum_{e \in E} C_e(D(e)) l(e). \quad (9.7)$$

Our water-network optimization problem is now completely specified as the minimization of (9.7), subject to (9.1)–(9.6).

The mathematical model is reported below:

$$\begin{aligned} \min \quad & \sum_{e \in E} C_e(D(e)) l(e) \\ & H(i) = h_s(i), \quad \text{for all } i \in S \\ & -\frac{\pi}{4} v_{\max}(e) D^2(e) \leq Q(e) \leq \frac{\pi}{4} v_{\max}(e) D^2(e), \quad \text{for all } e \in E \\ & \sum_{e \in \delta_-(i)} Q(e) - \sum_{e \in \delta_+(i)} Q(e) = \text{dem}(i), \\ & \quad \text{for all } i \in N \setminus S \\ & \text{ph}_{\min}(i) \leq H(i) - \text{elev}(i) \leq \text{ph}_{\max}(i), \\ & \quad \text{for all } i \in N \setminus S \\ & H(i) - H(j) = \frac{\text{sgn}(Q(e)) |Q(e)|^{1.852} \cdot 10.7 \cdot l(e) \cdot k(e)^{-1.852}}{D(e)^{4.87}}, \\ & \quad \text{for all } e = (i, j) \in E \\ & D(e) \in \{\mathcal{D}(e, 1), \mathcal{D}(e, 2), \dots, \mathcal{D}(e, r_e)\}, \quad \text{for all } e \in E. \end{aligned}$$

4 Solver

Because of the presence of discrete choices (the pipe diameters) and intrinsic nonlinearity (the dependence of head loss on flow and on pipe diameter), we are in the realm of MINLP. This is a very active area of current research (see [8], for example). We chose to use the open-source code `Bonmin` (Basic Open-source Nonlinear Mixed INteger programming) as our MINLP solver (see [3]). This choice was guided by the facts that (i) it is accessible from the modeling language `AMPL` (see [5]), (ii) it is freely distributed under an open-source license that has fairly liberal terms, (iii) we are able to modify the source code in case that is needed, and (iv) the primary developer, Pierre Bonami, is a very good friend (so *he* can modify the source code for us, in case that would be helpful).

Among other algorithms, `Bonmin` includes an NLP-based branch-and-bound implementation (see [2]). `Bonmin` is primarily aimed at searching for a global optimum of a formulation having a convex relaxation. Our formulation is not convex,

in particular because of the nonlinear Eqs. 9.5, so our use of `Bonmin` is as a heuristic. We utilized several features of `Bonmin` that are designed to facilitate such a use. In fact, though these features have rather general value for nonconvex formulations, these features were implemented in `Bonmin` due to our needs for this project.

4.1 Decreasing Nonconvexity

Because `Bonmin` is really aimed at MINLP formulations having a convex relaxation, it is natural to expect that it will perform better as a heuristic on nonconvex formulations for which the nonconvexity is not too severe. We can soften nonconvexities by making a substitution of variables. Specifically, we can define the cross-sectional area variables:

$$A(e) := \pi(D(e)/2)^2, \quad \text{for all } e \in E.$$

Then, the nonlinear flow bounds (9.2) become linear flow bounds:

$$-v_{max}(e)A(e) \leq Q(e) \leq v_{max}A(e), \quad \text{for all } e \in E. \tag{9.8}$$

Moreover, the nonlinear head loss Eqs. (9.5) become:

$$H(i) - H(j) = \frac{\text{sgn}(Q(e))|Q(e)|^{1.852} \cdot 10.7 \cdot l(e) \cdot k(e)^{-1.852} \left(\frac{\pi}{4}\right)^{2.435}}{A(e)^{2.435}},$$

for all $e = (i, j) \in E$. (9.9)

So, we consider the cross-sectional area variables $A(e)$ to satisfy

$$A(e) \in \{\mathcal{A}(e, 1), \mathcal{A}(e, 2), \dots, \mathcal{A}(e, r_e)\}, \quad \text{for all } e \in E, \tag{9.10}$$

where,

$$\mathcal{A}(e, r) := \pi(D(e, r)/2)^2, \quad \text{for all } e \in E, r = 1, \dots, r_e.$$

In (9.9), $\text{sgn}(y)$ for any number y denotes the sign (+ or -) of y . Denoting $x = Q(e)$, the term $\text{sgn}(Q(e))|Q(e)|^{1.852}$ in (9.9) can be represented as the function $f(x)$, where $f(x) = x^p$ ($p = 1.852$) when x is nonnegative; and $f(x) = -(f(-x))$ when x is negative. In the next section, we will describe how we approximate $f(x)$ by a smooth function.

4.2 Smoothing

The MINLP code `Bonmin` makes use of the NLP code `Ipopt` (see [7]) to solve continuous relaxations. Like many codes for NLP, it is aimed at smooth formulations.

Unfortunately, the absolute value term in the head loss constraints (9.5) is nondifferentiable (at 0) but not so bad. Thus, to accommodate I_{PROPT} , we smoothed the nondifferentiability. In order to do that effectively, our main goal is not to provide a fully accurate approximation near 0 because it is well known that Hazen–Williams Eq. (9.5) is in itself a poor approximation of the real pressure loss for small values of the flow. Instead, we smooth away the mild nondifferentiability by defining the head loss equation in a piecewise manner, so as to have accurate evaluations of the function. We insure the smoothness by matching function values as well as first and second derivatives at the breakpoints. Fortunately, piecewise constraints can be modeled in AMPL (see §18.3 of [5]).

More precisely, let $f(x) = x^p$ ($p = 1.852$) when x is nonnegative, and $f(x) = -f(-x)$ when x is negative (x is standing in for $Q(e)$). This function misbehaves at 0 (the second derivative does not exist there). Choose a small positive δ , and replace f with a function g on $[-\delta, +\delta]$. Outside of the interval, we leave f alone. We will choose g to be of the following form: $g(x) = ax + bx^3 + cx^5$. In this way, we can choose a, b, c (uniquely) so that f and g agree in value, derivative, and second derivative at $x = |\delta|$. So we end up with a smooth-enough antisymmetric function. It agrees in value with f at 0 and outside $[-\delta, +\delta]$. It agrees with f in the first two derivatives outside of $[-\delta, +\delta]$.

Formally, it is easy to prove that:

Proposition 1 *The unique polynomial $g(x) = ax + bx^3 + cx^5$ having $f(x) = g(x)$, $f'(x) = g'(x)$ and $f''(x) = g''(x)$ at $x = |\delta|$ is:*

$$\begin{aligned} g(x) = & \left(\frac{3\delta^{p-5}}{8} + \frac{1}{8}(p-1)p\delta^{p-5} - \frac{3}{8}p\delta^{p-5} \right) x^5 \\ & + \left(-\frac{5\delta^{p-3}}{4} - \frac{1}{4}(p-1)p\delta^{p-3} + \frac{5}{4}p\delta^{p-3} \right) x^3 \\ & + \left(\frac{15\delta^{p-1}}{8} + \frac{1}{8}(p-1)p\delta^{p-1} - \frac{7}{8}p\delta^{p-1} \right) x. \end{aligned}$$

Proof Via simple calculation one simply has to equate: (i) $g(\delta) = a\delta + b\delta^3 + c\delta^5 = \delta^p = f(\delta)$, (ii) $g'(\delta) = a + 3b\delta^2 + 5c\delta^4 = p\delta^{p-1} = f'(\delta)$, and (iii) $g''(\delta) = 6b\delta + 20c\delta^3 = p(p-1)\delta^{p-2} = f''(\delta)$. This is now a square linear system in the a, b, c variables. We solve it (symbolically), using *Mathematica* (see [9]). Finally, we just observe that f and g are antisymmetric, so we have the same a, b, c for $x = -\delta$. \square

Figure 9.4, drawn for $\delta = 0.1$, shows that g provides a good approximation of f . Indeed the quintic curve fits very well on $(-\delta, +\delta)$, and of course it matches up to second order with the true function f at $\pm\delta$. This is all no surprise because we are operating in a small interval of 0, and the function that we approximate is not pathological. The NLP solver I_{PROPT} responds well to this technique, as does our MINLP solver *Bonmin*.

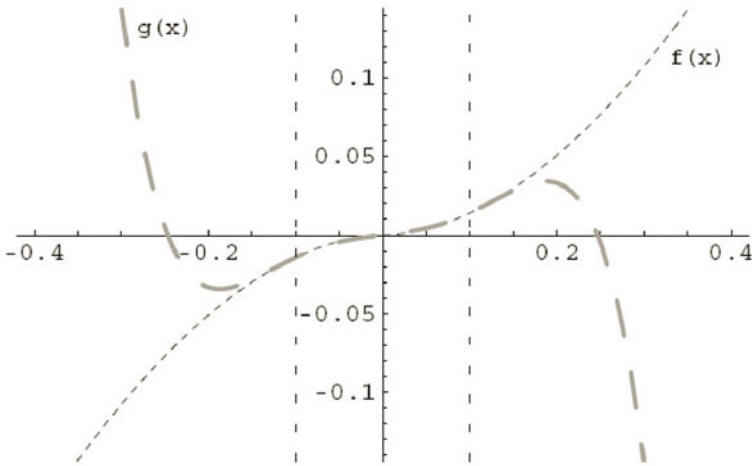


Fig. 9.4 Smoothing f near $x = 0$.

Our experience is that the inaccuracy in using this smoothed function is minimal compared to the other inaccuracies (e.g., numerical and modeling inaccuracies).

4.3 Continuous Cost Function

The NLP solver passes useful branching information to the MINLP solver by working with a good fully-continuous model. Because the cost data is truly discrete, we need to fit a continuous function to the data points. We use and advocate weighted fits to minimize relative error. For example, our least-squares fit for pipe e minimizes

$$\sum_{r=1}^{r_e} \left[1 - \left(\frac{\sum_{j=0}^{t_e} \beta(j, e) (\mathcal{A}(e, r)^2)^j}{C(e, r)} \right) \right]^2,$$

where, t_e is the desired degree and $\beta(j, e)$ are the coefficients of the polynomial¹ approximating C_e . Note that, for our purposes, a feature was added to Bonmin to accommodate one objective function for calculating lower bounds to guide the branch-and-bound search (for us, that is the fitted objective function) and another to calculate actual objective values to improve the incumbent solution (for us this just amounts to using the real objective function determined by the discrete cost data).

¹ Note that the least-square minimization is itself a nonconvex NLP that we solve to local optimality by using the open-source NLP solver Ipopt (see, [7]) which we use as an NLP solver throughout our work (see, §4.5).

4.4 SOS

It would be nice if `Bonmin` could directly handle branching for imposing constraints like (9.10), but at the present time, this is not the case. So, we simply define additional binary variables $X(e, r)$, where $X(e, r) = 1$ indicates that cross-sectional area $\mathcal{A}(e, r)$ is selected for pipe e ($r = 1, \dots, r_e$, $e \in E$). Then, we use the “SOS type-1” branching (see, [1]) that is available in `Bonmin`. As standard, we use `AMPL` suffixes to pass along the SOS information needed by the solver: `.sosno` (“SOS number”) is used to group variables into separate SOS constraints and `.ref` (“reference value”) is used to indicate the value symbolized by a variable. In this way, for $e \in E$, in `AMPL` we naturally set:

$$X(e, r).\text{sosno} := e, \text{ for } r = 1, \dots, r_e,$$

and

$$X(e, r).\text{ref} := \mathcal{A}(e, r), \text{ for } r = 1, \dots, r_e.$$

4.5 Heuristic Features of `Bonmin`

NLP-based branch-and-bound is only a heuristic for a formulation having a nonconvex continuous relaxation, because relaxations are not solved to global optimality. But `Bonmin` includes several options tailored to improve the quality of the solutions it provides in such a context.

First, the NLP-solver `Ippopt` may end up in different local optima when started from different starting points. The two options `num_resolve_at_root` and `num_resolve_at_node` allow for solving the root node or each node of the tree, respectively, with a user-specified number of different randomly-chosen starting points, saving the best solution found. Note that the function to generate a random starting point is very naive: it chooses a random point (uniformly) between the bounds provided for the variable.

Secondly, because the solution given by `Ippopt` does not truly give a lower bound, the user can adjust the fathoming rule to continue branching even if the solution value to the current node is worse than the best-known solution value. This is achieved by setting `allowable_gap`, `allowable_fraction_gap` and `cutoff_decr` to negative values.

5 Example

In this section, we describe a real-world instance as an example. The data are taken from a neighborhood of Bologna called Fossolo. We have 37 junctions (of which 1 reservoir, identified by the index 37), and 58 pipes. The topology of the network is represented in Fig. 9.5 and provided in detail in Table 9.3.

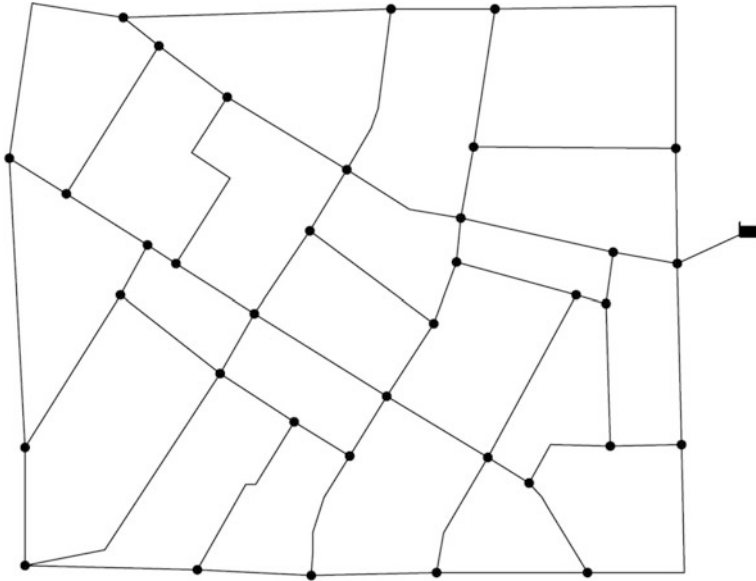


Fig. 9.5 Fossolo network

The hydraulic head at the reservoir is always fixed. In our example it is fixed to 121.0 m. The value of $v_{max}(e)$ is set to $1 \text{ m}^3/s$ and the roughness coefficient is set to 100 for all $e \in E$. The lower bound on the pressure head $ph_{min}(i)$ is equal to 40 m and the upper bound $ph_{max}(i)$ is equal to $121 - eval(i)$ meters for all $i \in N$. The values of the elevation and the demand for each junction besides the reservoir are reported in Table 9.1. The network topology and the length of the pipes are reported in Table 9.3.

Concerning the available diameters for each pipe, we have 13 different possibilities. Table 9.2 reports for each $e \in E$ the value of $(\mathcal{D}(e, r), \mathcal{C}(e, r))$ for all $r = 1, \dots, r_e$.

6 Solution Found by the MINLP Model

The solution found by Bonmin is depicted in Fig. 9.6 where the size of each diameter is proportional to the thickness of the arc. The diameters are expressed in meters, and the diameter is equal to 0.06 for the pipes without explicit number, i.e., the minimum diameter permissible for this data set.

The analysis of this solution shows a configuration in which the size of the selected diameters decreases from the reservoir toward the parts of the network farther away from the inlet point. This characteristic of the allocation of diameters to pipes plays

Table 9.1 Elevation (meters) and water demand (m^3/s) for instance Fossolo

i	$elev(i)$	$dem(i)$	i	$elev(i)$	$dem(i)$
1	65.15	0.00049	19	62.90	0.00188
2	64.40	0.00104	20	62.83	0.00093
3	63.35	0.00102	21	62.80	0.00096
4	62.50	0.00081	22	63.90	0.00097
5	61.24	0.00063	23	64.20	0.00086
6	65.40	0.00079	24	67.50	0.00067
7	67.90	0.00026	25	64.40	0.00077
8	66.50	0.00058	26	63.40	0.00169
9	66.00	0.00054	27	63.90	0.00142
10	64.17	0.00111	28	65.65	0.00030
11	63.70	0.00175	29	64.50	0.00062
12	62.64	0.00091	30	64.10	0.00054
13	61.90	0.00116	31	64.40	0.00090
14	62.60	0.00054	32	64.20	0.00103
15	63.50	0.00110	33	64.60	0.00077
16	64.30	0.00121	34	64.70	0.00074
17	65.50	0.00127	35	65.43	0.00116
18	64.10	0.00202	36	65.90	0.00047

Table 9.2 Diameter set (meters) and relative cost per meter (€/m) for instance Fossolo for each pipe

r	$\mathfrak{D}(e, r)$	$\mathfrak{C}(e, r)$
1	0.060	19.8
2	0.080	24.5
3	0.100	27.2
4	0.125	37.0
5	0.150	39.4
6	0.200	54.4
7	0.250	72.9
8	0.300	90.7
9	0.350	119.5
10	0.400	139.1
11	0.450	164.4
12	0.500	186.0
13	0.600	241.3

in favor of a correct hydraulic operation of the network and has a beneficial effect on water quality, see, e.g., the discussion in [10].

This characteristic could be noticed in the solution of MINLP for different instances, see Bragalli et al. [4] for details.

Note that the proposed solution is not guaranteed to be a global optimum of the problem. However, because of the intrinsic difficulty of the problem at hand, the proposed solution is, from a practical viewpoint, a very good quality feasible solution. Other approaches based on heuristic algorithms, mixed integer linear programming, or nonlinear programming models are not effective for medium/large instances. In this application, modeling the problem in the most natural way seems to be the most successful approach.

Table 9.3 Network topology and length of the pipes (meters) for instance Fossolo

$e = (i, j)$	i	j	$l(e)$	$e = (i, j)$	i	j	$l(e)$
1	1	17	132.76	30	15	22	80.82
2	17	2	374.68	31	22	7	340.97
3	2	3	119.74	32	5	13	77.39
4	3	4	312.72	33	13	14	112.37
5	4	5	289.09	34	14	20	37.34
6	5	6	336.33	35	20	15	108.85
7	6	7	135.81	36	15	16	182.82
8	7	24	201.26	37	16	29	136.02
9	24	8	132.53	38	29	30	56.7
10	8	28	144.66	39	30	9	124.08
11	28	9	175.72	40	17	18	234.6
12	9	36	112.17	41	12	13	203.83
13	36	1	210.74	42	19	20	248.05
14	1	31	75.41	43	14	21	65.19
15	31	10	181.42	44	21	6	210.09
16	10	11	146.96	45	21	22	147.57
17	11	19	162.69	46	22	23	103.8
18	19	12	99.64	47	24	23	210.95
19	12	4	52.98	48	23	25	75.08
20	2	18	162.97	49	26	27	180.29
21	18	10	83.96	50	28	29	149.05
22	10	32	49.82	51	29	33	215.05
23	32	27	78.5	52	32	33	144.44
24	27	16	99.27	53	33	34	34.74
25	16	25	82.29	54	31	34	59.93
26	25	8	147.49	55	34	35	165.67
27	3	11	197.32	56	30	35	119.97
28	11	26	83.3	57	35	36	83.17
29	26	15	113.8	58	37	1	1.0

7 Conclusions

More details concerning our methodology and results can be found in [4].

8 A Practical Exercise

In this section, we describe a small but interesting example taken from the literature, in particular it was introduced by Fujiwara and Khang [6]. The network is a simplified version of the water network of the city of Hanoi, Vietnam.

We have 32 junctions (of which 1 reservoir, identified by the index 1), and 34 pipes. The topology of the network is represented in Fig. 9.7.

The hydraulic head at the reservoir is always fixed. In our example it is fixed to 100.0 m. The roughness coefficient is set to 130 and the elevation of each junction is 0. The lower bound on the pressure head $ph_{min}(i)$ is equal to 30 m and the upper bound $ph_{max}(i)$ is equal to 100 m for all $i \in N$. The values of the demand for each

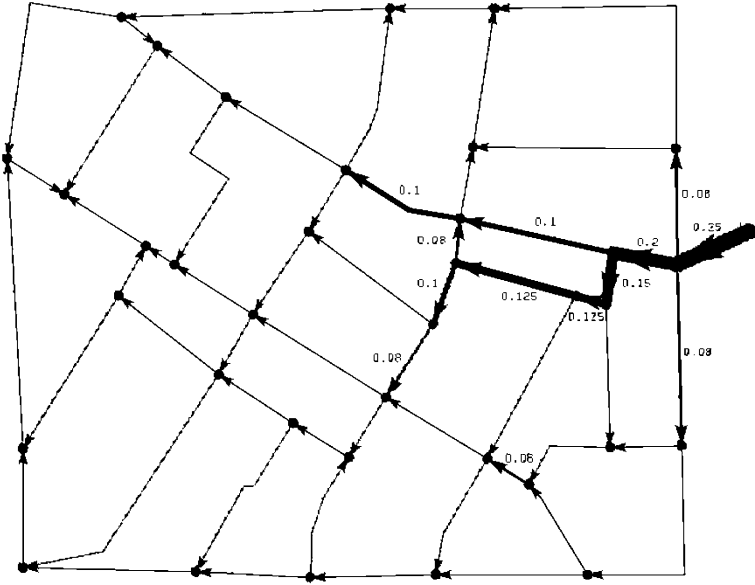


Fig. 9.6 Solution for Fossolo network: the size of each diameter is proportional to the thickness of the arc. The diameter (expressed in meters) is equal to 0.06 m for the pipes without explicit number

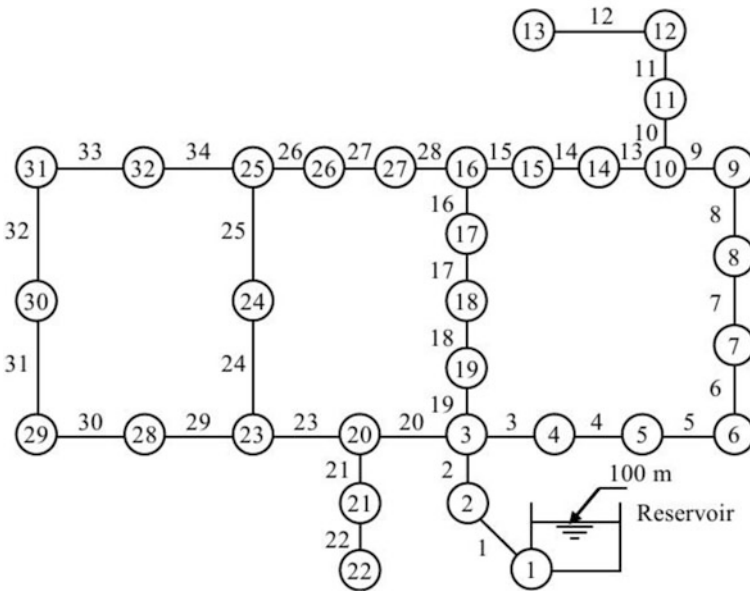


Fig. 9.7 Hanoi network

Table 9.4 Diameter set (meters) and relative cost per meter (€/m) for instance Hanoi for each pipe

r	$\mathfrak{D}(e, r)$	$\mathfrak{C}(e, r)$
1	0.3048	45.73
2	0.4064	70.40
3	0.5080	98.39
4	0.6096	129.33
5	0.7620	180.75
6	1.0160	278.28

Table 9.5 Water demand (m^3/s), topology, pipe length (m), and maximum speed of the water within the pipes (m^3/s) for instance Hanoi

i	$dem(i)$	$e = (i, j)$	i	j	$l(e)$	$v_{max}(e)$
1		1	2	100	7.0	
2	0.24722	2	3	1350	7.0	
3	0.23611	3	4	900	3.0	
4	0.03611	4	5	1150	3.0	
5	0.20139	5	6	1450	2.5	
6	0.27917	6	7	450	2.5	
7	0.375	7	8	850	2.0	
8	0.15278	8	9	850	2.0	
9	0.14583	9	10	800	2.0	
10	0.14583	10	11	950	2.0	
11	0.13889	11	12	1200	2.0	
12	0.15556	12	13	3500	2.0	
13	0.26111	13	14	800	2.0	
14	0.17083	14	15	500	2.0	
15	0.07778	15	16	550	2.0	
16	0.08611	16	17	2730	2.0	
17	0.24028	17	18	1750	2.0	
18	0.37361	18	19	800	3.5	
19	0.01667	19	3	400	3.5	
20	0.35417	20	3	20	2200	3.0
21	0.25833	21	20	21	1500	2.0
22	0.13472	22	21	22	500	2.0
23	0.29028	23	20	23	2650	2.0
24	0.22778	24	23	24	1230	3.0
25	0.04722	25	24	25	1300	2.0
26	0.25000	26	25	26	850	2.0
27	0.10278	27	26	27	300	2.0
28	0.08056	28	27	16	750	2.0
29	0.10000	29	23	28	1500	2.0
30	0.10000	30	28	29	2000	2.0
31	0.02917	31	29	30	1600	2.0
32	0.22361	32	30	31	150	2.0
		33	31	32	860	2.0
		34	32	25	950	2.0

junction besides the reservoir, the network topology, the upper bound on the speed of the water within the pipes, and the length of the pipes are reported in Table 9.5.

Concerning the available diameters for each pipe, we have 6 different possibilities. Table 9.4 reports for each $e \in E$ the value of $(\mathfrak{D}(e, r), \mathfrak{C}(e, r))$ for all $r = 1, \dots, r_e$.

References

1. Beale, E. M. L., & Tomlin, J. A. (1970). Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In J. Lawrence, editor, *Proc. of the 5th Int. Conf. on Operations Research*, pages 447–454
2. Bonami, P., Biegler, L. T., Conn, A. R., Cornuéjols, G., Grossmann, I. E., Laird, C. D., Lee, J., Lodi, A., Margot, F., Sawaya, N., & Wächter, A (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5:186–204
3. Bonmin. <https://projects.coin-or.org/Bonmin>, v. 1.0.
4. Bragalli, C., D'Ambrosio, C., Lee, J., Lodi, A., & Toth, P (2012). On the optimal design of water distribution networks: a practical minlp approach. *Optimization and Engineering*, 13:219–246
5. Fourer, R., Gay, D. M., & Kernighan., B. W. (2003). *A Modeling Language for Mathematical Programming*. Duxbury Press/Brooks/Cole Publishing Co., second edition
6. Fujiwara, O., & Khang, D. B. (1990). A two-phase decomposition method for optimal design of looped water distribution networks. *Water Resources Research*, 26: 539–549
7. Ipopt. <https://projects.coin-or.org/Ipopt>, v. 3.5.
8. Lee, J., & Leyffer, S. (2012). *Mixed Integer Nonlinear Programming*. The IMA Volumes in Mathematics and its Applications. Springer
9. Mathematica. www.wolfram.com/products/mathematica/index.html, v. 7.0.
10. Van Den Boomen, M., Van Mazijk, A., & Beuken, R. H. S. (2004). First evaluation of new design concepts for self-cleaning distribution networks. *Journal of Water Supply: Research and Technology AQUA*, 53(1), 43–50