RC23893 (W0602-210) February 28, 2006
Mathematics

IBM Research Report

An MINLP Solution Method for a Water Network Problem

Christiana Bragalli[1], Claudia D'Ambrosio[2], Jon Lee[3], Andrea Lodi[2,4], Paolo Toth[2]

[1]DISTART
University of Bologna
viale Risorgimento 2
40136 Bologna, Italy

[2]DEIS
University of Bologna
viale Risorgimento 2
40136 Bologna, Italy

[3]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

[4]Carried out in part while a Herman Goldstine Fellow at the IBM T. J. Watson Research Center

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

LIMITED DISTRIBUTION NOTICE: This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g, payment of royalties). Copies may be requested from IBM T. J. Watson Research Center , P. O. Box 218, Yorktown Heights, NY 10598 USA (email: reports@us.ibm.com). Some reports are available on the internet at http://domino.watson.ibm.com/library/CyberDig.nsf/home.

An MINLP solution method for a water network problem

Cristiana Bragalli[1], Claudia D'Ambrosio[2], Jon Lee[3], Andrea Lodi[2,4], and Paolo Toth[2]

[1] DISTART, University of Bologna, viale Risorgimento 2, 40136 Bologna, Italy
[2] DEIS, University of Bologna, viale Risorgimento 2, 40136 Bologna, Italy
[3] IBM TJ Watson Research Center, PO Box 218, Yorktown Heights, NY 10598, USA

February 28, 2006

Abstract. We propose a solution method for a water-network optimization problem using a nonconvex continuous NLP relaxation and an MINLP search. We report successful computational experience using available MINLP software on problems from the literature and on difficult real-world instances.

Introduction

The optimal design of a WDN (Water Distribution Network) consists, in its classical formulation, of the choice of a diameter for each pipe, while other design properties are considered to be fixed (e.g., the topology and pipe lengths). From a mathematical viewpoint, we can cast the optimal design problem of a WDN as a MINLP (Mixed Integer NonLinear Programming) problem in which the discrete variables select from a set of commercially-available diameters, water flows must respect the hydraulic constraints, and we seek to minimize the cost function which only depends on the selected diameters.

Recently there has been renewed interest in optimal WDN design, due to emerging issues related to water distribution systems; in particular, the gradual deterioration of network pipes and the need for a more rational use of water resources has lead to very costly renovation activities.

Approaches in the literature use various combinations of linearization and relaxation, which lead to MILP (Mixed Integer Linear Programming), NLP NonLinear Programming) and meta-heuristic algorithms. We survey these approaches in §3. In this paper we are interested to approaches exploiting mathematical-programming formulations, and we consider two cases.

The MILP approach to our problem relies on using piecewise-linear approximations. If tractable, solution of such a model would provide the global optimum of an approximation to the real system. If accurate models are desired for a large network, we are lead to using a large number of binary variables (to manage the

[4] Part of this research was carried out while Andrea Lodi was Herman Goldstine Fellow at the IBM T.J. Watson Research Center whose support is gratefully acknowledged

linear pieces). This tends to lead to a very poor relaxation and ultimately an intractable model.

With an MINLP approach, we are lead to a more natural model. Our view is that by accurately modeling the nonlinear phenomena, we will have a model that will provide an MINLP search with a good NLP relaxation. While foregoing any hope of verifying global optimality of the best solution encountered, we are able to find very good solutions to large real-world instances.

Our experiments were carried out using AMPL [9] as an interface to two MINLP codes. We are using Sven's Leyffer's code `MINLP_BB` [12], (available from the University of Dundee) as well as the new CMU/IBM open-source MINLP code `Bon-min` [2, 3] (to be available from COIN-OR [4]). Our modeling and solution methods are worked out with the target software in mind.

In Section §1, we formally set the notation for specifying instances of the problem. In §2, we describe the problem more fully, through a preliminary continuous model. In §3, we survey earlier approaches, and we describe an NLP model in which we make a smooth (approximate) relaxation of the preliminary model described in §2, so that we can apply methods of smooth optimization. In §4, so as to decrease the nonlinearity, we describe a reparameterization of pipes by (cross-sectional) area, rather than diameter. In §5, we describe how we incorporate binary variables for the purposes of then applying different MINLP codes. In §6, we describe the results of computational experiments.

1 Notation

The network is oriented for the sake of making a careful formulation, but flow on each pipe is not constrained in sign (i.e., it can be in either direction). The network consists of pipes (arcs) and junctions (nodes). In the optimization, the pipes are to have their diameters sized.

Sets:

E = set of pipes.
N = set of junctions.
ν = source junction (ν is a fixed element of N).
$\delta_+(i)$ = set of pipes with tail at junction i.
$\delta_-(i)$ = set of pipes with head at junction i.

Parameters:

$len(e)$ = length of pipe e ($e \in E$).
$k(e)$ = physical constant depending on pipe material ($e \in E$).
$dem(i)$ = demand at junction i ($i \in N$).
$elev(i)$ = physical elevation of junction i ($i \in N$).
$p_{min}(i)$ = minimum pressure at junction i ($i \in N$).
$p_{max}(i)$ = maximum pressure at junction i ($i \in N$).
$d_{min}(e)$ = minimum diameter of pipe e ($e \in E$).

$d_{max}(e) =$ maximum diameter of pipe e $(e \in E)$.
$v_{max}(e) =$ maximum speed of pipe e $(e \in E)$.

Pipes are only available from a discrete set of r_e diameters. For $e \in E$:

$$d_{min}(e) := \mathfrak{D}(e,1) < \mathfrak{D}(e,2) < \cdots < \mathfrak{D}(e,r_e) =: d_{max}(e).$$

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, $\mathfrak{C}(e,r) := C_e(\mathfrak{D}(e,r))$, $r = 1,\ldots,r_e$, where

$$\mathfrak{C}(e,1) < \mathfrak{C}(e,2) < \cdots < \mathfrak{C}(e,r_e).$$

2 A preliminary continuous model

In this section, we fully describe the problem, and at the time we develop a preliminary NLP relaxation.

Variables:

$Q(e) =$ flow in pipe e $(e \in E)$.
$D(e) =$ diameter of pipe e $(e \in E)$.
$H(i) =$ hydraulic head of junction i $(i \in N)$.

Simple bounds [Linear]:

$d_{min} \leq D(e) \leq d_{max} \quad (\forall e \in E)$.
$p_{min}(i) + elev(i) \leq H(i) \leq p_{max}(i) + elev(i) \quad (\forall\ i \in N)$.

Flow bounds (dependent on cross-sectional area of pipe) [Smooth but nonconvex]:

$$-\tfrac{\pi}{4} v_{max}(e) D^2(e) \leq Q(e) \leq \tfrac{\pi}{4} v_{max}(e) D^2(e) \quad (\forall\ e \in E).$$

Flow conservation [Linear]:

$$\textstyle\sum_{e \in \delta_-(i)} Q(e) - \sum_{e \in \delta_+(i)} Q(e) = dem(i) \quad (\forall\ i \in N \setminus \{\nu\}).$$

Head loss across links [Nonsmooth and nonconvex]:

$$H(i) - H(j) = sign(Q(e))|Q(e)|^{1.852} \cdot 10.7 \cdot len(e) \cdot k(e)^{-1.852}/D(e)^{4.87} \quad (\forall\ e = (i,j) \in E).$$

This constraint models friction loss in water pipes using the empirical Hazen-Williams equation. This is an accepted model for fully turbulent flow in *water* networks. Diameter is bounded away from 0, so the only nondifferentiability is when the flow is 0.

Objective to be minimized [Discrete]:

$\sum_{e \in E} C_e(D(e))\ len(e)$

Since we only have discretized cost data, within AMPL we are fitting a polynomial to the input discrete cost data to make a working continuous cost function $C_e()$.

We have experimented with different fits: l_1, l_2 and l_∞; with and without requiring that the fit under or over approximates the discrete points. Requiring an under approximation makes our formulation a true relaxation — in the sense that the global minimum of our relaxation is a lower bound on the discrete optimum. We use and advocate weighted fits to minimize relative error. For example, our least-squares fit for arc e minimizes

$$\sum_{r=1}^{r_e} \frac{1}{\mathfrak{C}(e,r)^2} \left[\mathfrak{C}(e,r) - \left(\sum_{j=0}^{t} \beta(j,e) \left(\frac{\pi}{4} \mathfrak{D}(e,r)^2 \right)^j \right) \right]^2$$

$$= \sum_{r=1}^{r_e} \left[1 - \left(\frac{\sum_{j=0}^{t} \beta(j,e) \left(\frac{\pi}{4} \mathfrak{D}(e,r)^2 \right)^j}{\mathfrak{C}(e,r)} \right) \right]^2$$

3 Models and algorithms

Optimal design of a WDN has already received considerable attention. Artina and Walker [1] linearize and use an MILP approach. Savic and Walters [14] and Cunha and Sousa [6] work within an accurate mathematical model, but they use meta-heuristic approaches for the optimization, and they work with the constraints by numerical simulation. Fujiwara and De Silva [10] employ a "split-pipe model" in which each pipe e is split into r_e stretches with unknown length where r_e is the number of possible choices of the diameter of pipe e and the variables become the length of the stretches. It is not difficult to see that models of this type have the disadvantage of allowing solutions with many changes in the diameter along the length of a pipe, and the hydraulic behavior of which is not accurately modeled. Using this type of model, they employ a meta-heuristic approach for the optimization, working with the constraints by numerical simulation. Eiger et al. [7] also work with a split-pipe model, but they use NLP methods for calculating a solution. Xu and Goulter [15] and Lansey and Mays [11] also employ an NLP approach, but they use an approximation of the split-pipe methodology (using just 2 discrete pipe sections).

In what follows, we develop an MINLP approach and compare it to the more standard MILP approach. The MILP approach has the advantage of correctly modeling the choices of discrete diameters with binary indicator variables $x_{e,r}$ representing the assignment of diameter $\mathfrak{D}(e,r)$ to arc e. In this way we can also easily incorporate costs for the chosen diameters. There is still the nonlinearity of the flow terms in the head-loss constraints. Piecewise-linear approximation of these nonlinear constraints is the standard MILP approach here. Unfortunately, the resulting MILPs are typically very difficult to solve. The difficulty of the MILP models is related to the fact that once the diameters have been fixed, the objective function is set, and a feasibility problem associated with the piecewise-linear approximation must be solved, without any guidance from the objective function. It turns out that linear-programming tools in such a context are not effective at all. Good feasible solutions to the models are not always obtainable for even networks of moderate size. Often one is lead to using very coarse piecewise-linear approximations to get some sort of solution, but these tend to not be accurate enough to be truly feasible. Indeed, especially with few linearization points, the MILP may (i) generate flows that are infeasible, and (ii) cut off some feasible (and potentially optimal) solutions. §6 includes some of these rather negative computational results with the MILP approach.

Instead, our preferred starting point is a fully-continuous nonconvex NLP model as described in §2. The main difficulty, besides giving up on global optimality, is to deal algorithmically with the absolute value term in the head-loss constraints. This term is nondifferentiable (at 0) but not badly. One possibility is to ignore the nondifferentiability issue, and just use a solver that will either get stuck or will handle it in its own way. This has the advantage of straightforward implementation from AMPL and access to many NLP solvers (e.g., via NEOS [13]). But since we ultimately wish to employ available MINLP solvers, and these solvers count on being given smooth NLP subproblems, we look for a more promising approach.

We suggest smoothing away the mild nondifferentiability as follows: 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 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 nice smooth-enough anti-symmetric function. It agrees in value with f at 0 and outside $[-\delta, +\delta]$. It agrees with f in the first two derivatives outside of $(-\delta, +\delta)$. Some simple calculations yields

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

Note that $f'(0) = 0$, while $g'(0)$ is slightly positive.

As can be seen in Figure 1, this seems to work pretty well on a micro level since the function f is not so bad near $x = 0$. In the figure, we have taken $\delta = 0.01$. Indeed the quintic curve fits very well in $(-\delta, +\delta)$, and of course it matches up to second order with the true function f at $\pm\delta$. This is all no surprise since we are operating in a small interval of 0, and the function that we approximate is not pathological. The NLP solvers that we have tested appear to respond well to this technique.

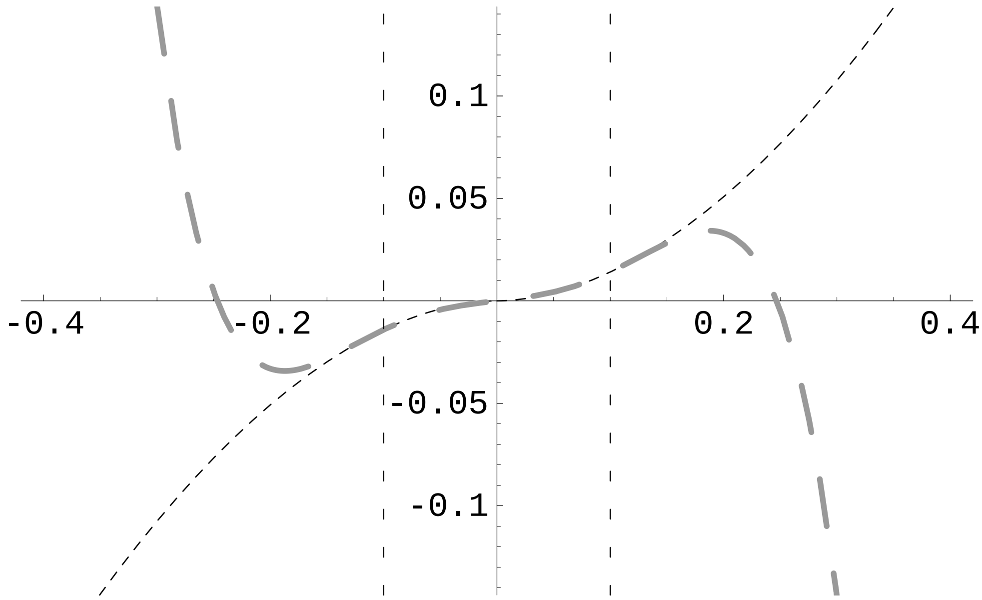

Fig. 1. Smoothing f near $x = 0$

Piecewise constraints can be modeled in AMPL (see §18.3 of [9]), so we have the advantage of then utilizing a variety of NLP solvers as well as a path to using `Bon-min` as well as `MINLP_BB`, both of which are interfaced with AMPL. There is already some accuracy lost in allowing the diameter variables to be continuous, and of course in that we will only find local optima of the NLP due to the nonconvexity of the model. Our experience is that the additional inaccuracy in using this smoothed function is minimal compared to the other inaccuracies mentioned.

4 Parameterizing by area rather than diameter

We can use variables

$A(e)$ = cross-sectional area of pipe e ($e \in E$),

rather than the diameter variables $D(e)$ ($e \in E$). This makes the model less nonlinear. In particular, we have the *now linear* flow bounds:

$$-v_{max}(e)A(e) \leq Q(e) \leq v_{max}(e)A(e) \quad (\forall\ e \in E),$$

the *still linear* simple bounds:

$$\tfrac{\pi}{4} d_{min}^2 \leq A(e) \leq \tfrac{\pi}{4} d_{max}^2 \quad (\forall e \in E),$$

and the *less nonlinear* head loss across links constraints:

$$H(i) - H(j) = sign(Q(e))|Q(e)|^{1.852} \cdot 10.7 \cdot len(e) \cdot k(e)^{-1.852} \left(\tfrac{\pi}{4}\right)^{2.435} / A(e)^{2.435}$$
$$(\forall\ e = (i,j) \in E).$$

Finally, there is the possibility that the cost function may be well modeled by a function that is nearly quadratic in diameter — this means *nearly linear* in area, which would be very nice.

We have tried out this area parameterization with different NLP solvers, and it seems to work well, presumably due to the fact that the model is less nonlinear.

5 Discretizing the diameters

With an eye toward using `Bon-min` as well as `MINLP_BB`, we discretized the diameters in a certain way. Specifically, we defined additional binary variables

$$X_{e,r}, \quad r = 1, \ldots, r_e - 1; \quad \forall e \in E.$$

These variables are used to represent diameter *increments*. That is, we have the linking equations

$$D(e) = \mathfrak{D}(e,1) + \sum_{r=2}^{r_e} \left(\mathfrak{D}(e,r) - \mathfrak{D}(e,r-1)\right) X_{e,r-1}, \quad \forall e \in E.$$

and

$$X_{e,r} \geq X_{e,r+1}, \quad \text{for } r = 1, \ldots, r_e - 2; \quad \forall e \in E.$$

The advantage of this incremental modeling is that branching $D(e) \leq \mathfrak{D}(e,r)$ vs $D(e) \geq \mathfrak{D}(e, r+1)$ can be realized by ordinary 0/1 branching on the single binary variable $X_{e,r}$, without requiring any special solver handling of so-called SOS of Type 1.

If we wish to work with the area parameterization instead (see §4), we employ precisely the same discretization. That is, we keep the same 0/1 variables, but we employ the *still linear* linking equations:

$$A(e) \;=\; \frac{\pi}{4}\left(\mathfrak{D}^2(e,1) + \sum_{r=2}^{r_e} \left(\mathfrak{D}^2(e,r) - \mathfrak{D}^2(e,r-1)\right) X_{e,r-1}\right), \quad \forall e \;\in\; E.$$

6 Some computational results

The area parameterization seems to be better behaved than the diameter one, so we confine our reported experimental results to the area parameterization. For convenience, we define the discrete areas $\mathfrak{A}(e,r) := \frac{\pi}{4}\mathfrak{D}(e,r)^2$, for $r = 1,\ldots,r_e$.

For the computational results, for approximating the cost function (see §2), we used rather high-degree polynomials, l_2 approximation, and we required that the fitted curve be a lower approximation of the discrete points.

We created an AMPL model that first fits the cost function, and then solves the continuous problem instances using a variety of NLP solvers (notably, we experimented with the Dundee solver `filterSQP` and the open-source COIN-OR solver `Ipopt`). This seems to give decent local minima without any special starting points needed. On all of our data sets, `filterSQP` and `Ipopt`, using the AMPL interface, have been able to find good local optimua rather easily.

We first solve the NLP relaxation to get continuous areas $A(e)$. Then, toward using the MINLP solvers `MINLP_BB` and `Bon-min`, we are setting branching priorities as follows. If $A(e)$ is between say $\mathfrak{A}(e,r')$ and $\mathfrak{A}(e,r'+1)$, then we let

$$prio(X_{e,r}) := 100.5 - |r' - r + 0.5|,$$

so that $prio(X_{e,r'}) = prio(X_{e,r'+1}) = 100$, $prio(X_{e,r'-1}) = prio(X_{e,r'+2}) = 99$, $prio(X_{e,r'-2}) = prio(X_{e,r'+3}) = 98, \ldots$

Our data sets are `shamir`, `hanoi` and `foss`. For `foss`, we have three variations: `foss_poly_0`, `foss_iron`, and `foss_poly_1`. Summary statistics and computational results using MINLP for the data sets are in Table 1. For comparison, Table 2 contains results using an MILP model.

For the small network `shamir`, we obtain an MINLP solution equal to the previously best known (and almost certainly optimal) one.

For `hanoi`, which is a significantly harder problem, we also perform well. We obtain an MINLP solution that is only slightly worse that the best known one. Previously computed solution values that we know of are:

6.073×10^6, Savic and Walters [14];
6.056×10^6, Cunha and Sousa [6];
6.327×10^6, MILP (see Table 2).

In particular, we do significantly better than the solution that we obtained by MILP. Possibly the value of our NLP relaxation can be compared (somewhat favorably) with the "split-pipe" designs obtained in the literature:

6.319×10^6, Fujiwara and Khang [10];
6.027×10^6, Eiger et al. [7].

The `foss` data is from a real problem of the Fossolo neighborhood of Bologna. In Figure 2, we have a diagram of the Fossolo network made with EPANET 2.0 [8]. EPANET is free software distributed by the US Environmental Protection Agency. It is commonly used to model the hydraulic and water quality behavior of water distribution piping systems.

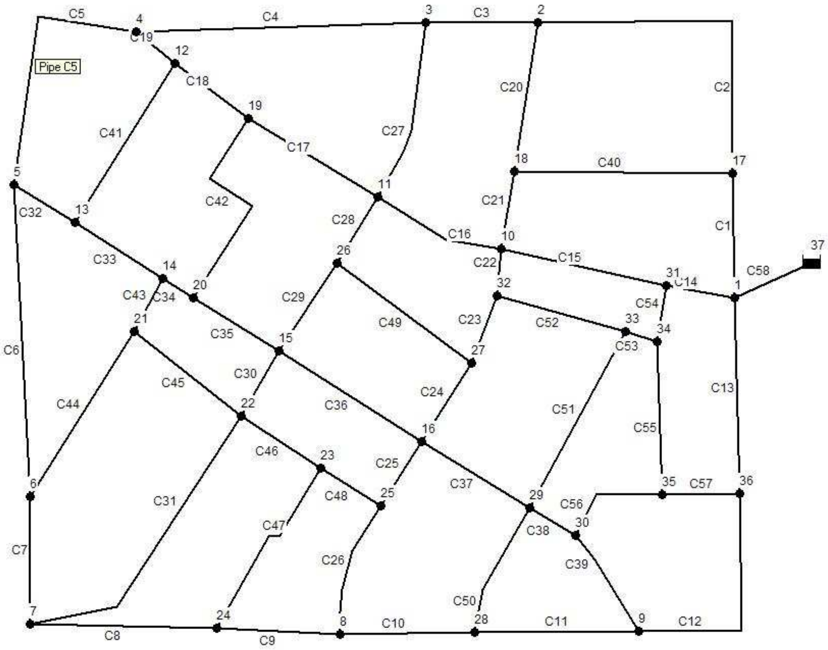

Fig. 2. Fossolo network

We have three concrete instances for this network. Instance `foss_poly_0` consists of the original data provided to us for this network. The pipe material for that instance is polyethylene. Our solution compares quite favorably with the solution obtained using MILP. Not only is the objective value poor for the solution obtained by MILP, the piecewise-linear approximation is very coarse, and so the solution obtained can not really be considered as feasible. Instance `foss_iron` is for the same network, but with almost twice as many choices of pipe diameters and with the material being cast iron. Instance `foss_poly_1`, a polyethylene instance, is a much harder instance than the other two, with even more choices for the pipe diameters. Note that for instance `foss_poly_1`, there is a larger relative discrepancy between the value of the continuous optimum and the value of the MINLP solution that we were able to find. This suggests that there is a good possibility that we may be able to obtain a significantly better MINLP solution for this instance.

We note that the MILP model is entirely too difficult to work with for the `foss_iron` and `foss_poly_1` data sets.

The cost data for `foss_poly_0` is out of date, and so the solution values can not be directly compared to that of `foss_poly_1` and `foss_iron`, which can be reasonably compared. The value of the solution that we obtained for `foss_poly_1` is much lower than for `foss_iron`. At first this seems surprising, but this is explained by comparing the costs of the varying diameters of pipe. We see in Figure 3 that for small diameters, polyethylene is much cheaper than cast iron, and we note that the data is such that there are feasible solutions with very low flows. Although polyethylene is generally a much cheaper material than cast iron, its life is rather limited, and so cast iron is strongly preferred as a long-term solution.

The MINLP results were obtained under the following computing environment: Windows XP, Pentium M, 1.70 GHz, 1 GB RAM. The instance `shamir` required just a few seconds, and each of the other instances took 3-4 minutes for the solutions obtained. The MILP results were obtained with `CPLEX` 9.0.3 [5] under the following computing environment: Windows XP, Pentium IV, 1.70 GHz, 512 MB RAM. The MILP run times for `shamir` (resp., `hanoi`, `foss_poly_0`) were 262 (91,730, 176,960) seconds.

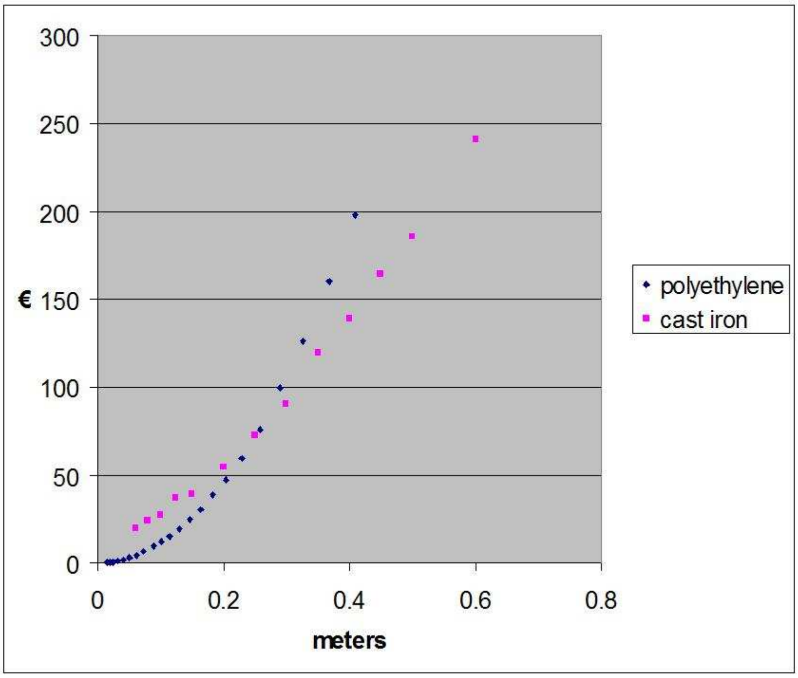

Fig. 3. Cast iron vs. polyethylene

We have experimented with restricting the range of discrete diameters to ones nearby the diameters chosen in the continuous optimum; this seems to be a very useful approach for difficult instances like `foss_poly_1`.

Network	# junctions	# pipes	# diameters	NLP (fitted obj)	MINLP (fitted obj)	MINLP (actual obj)	Previously best known
`shamir`	7	8	14	425,103.06	443,295.95	419,000.00	419,000.00
`hanoi`	32	34	6	6,013,430.03	6,109,620.90	6,109,620.90	6,056,000.00
`foss_poly_0`	37	58	7	35,403.19	36,503.44	36,503.44	(46,533.38)*
`foss_iron`	37	58	13	178,829.52	180,373.35	178,673.70	—
`foss_poly_1`	37	58	22	27,827.06	31,442.21	31,178.89	—

Table 1. Computational results for the MINLP model

Network	Best MILP solution	LP solution	Lower bound	Gap (%)	# nodes	# nodes remaining	# lineariz-ation points
`shamir`	419,000.0	307,897.7	419,000.0	0.00	35,901	0	15
`hanoi`	6,327,613.3	5,508,664.4	6,117,905.6	3.31	4,532,718	2,592,716	7
`foss_poly_0`	(46,533.4)*	33,882.7	34,851.8	25.10	1,845,254	1,299,426	3

Table 2. Computational results for MILP model

*piecewise-linearization is too coarse for us to rely on this solution as being truly feasible to the MINLP as discussed in detail in Section 3

7 Conclusions

We are able to get good solutions to practical instances of water-network optimization problems, with very low development time. We attribute our success to:

1. The availability of software for finding good solutions to MINLP problems.
2. The easy interface to such software via the modeling language AMPL.
3. The natural framework of MINLP allows for an easy-to-develop and close model of the real system — to some extent we give up on a MILP model that seeks a globally-optimal solution, so that we can get a close MINLP model of the system which is tractable for finding good local solutions.
4. Smoothing mild nonlinearities (of the head-loss constraints) makes for good behavior of typical codes that solve the NLP subproblems.
5. Reparameterizing (by cross-sectional area rather than diameter) leads to a less nonlinear and more convex model.
6. Modeling discrete choices (of pipes) by (cross-sectional area) increments, and then setting appropriate branching priorities, enables us to mimic SOS branching while using only simple single-variable branching of the MINLP solvers.

Our belief is that much of this wisdom (omitting the parenthetical remarks above) applies to other instances of optimization problems with significant discrete and nonlinear aspects.

Acknowledgments

We thank Roger Fletcher and Sven Leyffer for giving us access to MINLP_BB. We thanks Andreas Wächter for helping us get set up to use MINLP_BB and Bon-min. We thank Andreas for also working with us on tuning Ipopt to handle some nasty subproblems. Finally, we thank Pierre Bonami for making some accommodations to Bon-min to handle nonconvexities.

References

1. S. Artina and J. Walker, *Sull'uso della programmazione a valori misti nel dimensionamento di costo minimo di reti in pressione*, Atti dell'Accademia delle Scienze dell'Istituto di Bologna, Anno 271, Serie III, Tomo X, 1983.
2. P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, *An algorithmic framework for convex mixed integer nonlinear programs*, Tech. report, IBM Research Report RC23771, 2005.
3. P. Bonami and J. Lee, Bob-min *users' manual*, Tech. report, March 2006.
4. COIN-OR, www.coin-or.org.
5. CPLEX, www.ilog.com/products/cplex.
6. M. Cunha and J. Sousa, *Water distribution network design optimization: Simulated annealing approach*, J. Water Res. Plan. Manage. Div. Soc. Civ. Eng. **125(4)** (1999), 215–221.
7. G. Eiger, U. Shamir, and A. Ben-Tal, *Optimal design of water distribution networks*, Water Resources Research **30(9)** (1994), 2637–2646.
8. EPANET, www.epa.gov/ORD/NRMRL/wswrd/epanet.html.
9. R. Fourer, D.M. Gay, and B.W. Kernighan, *AMPL: A modeling language for mathematical programming*, second ed., Duxbury Press/Brooks/Cole Publishing Co., 2003.
10. O. Fujiwara and D.B. Khang, *A two-phase decomposition method for optimal design of looped water distribution networks*, Water Resources Research **26(4)** (1990), 539–549.
11. K.E. Lansey and L.W. Mays, *Optimization model for water distribution system design*, Journal of Hydraulic Engineering **115(10)** (1989), 1401–1418.
12. S. Leyffer, *User manual for* MINLP_BB, Tech. report, University of Dundee, April 1998; revised, March 1999.
13. NEOS, www-neos.mcs.anl.gov/neos.
14. D.A. Savic and G.A. Walters, *Genetic algorithms for the least-cost design of water distribution networks*, ASCE Journal of Water Resources Planning and Management **123(2)** (1997), 67–77.
15. C. Xu and I.C. Goulter, *Reliability-based optimal design of water distribution networks*, Journal of Water Resources Planning and Management **125(6)** (1999), 352–362.