GLOBAL OPTIMIZATION OF PIPE NETWORKS BY THE INTERVAL ANALYSIS APPROACH: THE BELGIUM NETWORK CASE

J. FRÉDÉRIC BONNANS, GRÉGOIRE SPIERS, AND JEAN-LÉOPOLD VIE

Abstract. We show that global optimization techniques, based on interval analysis and constraint propagation, succeed in solving the classical problem of optimization of the Belgium gas network.

Keywords: Pipe networks, global optimization, constraint programming, interval analysis.

1. Introduction

We consider in this paper gas network optimization problems which are based on the hypothesis of a stationary flow. Although this approach neglects importants effects (e.g. variations of consumption during the day) it is widely used since it it gives a reasonable approximation allowing to design future networks. The model is detailed in the next section. Let us just say that the variables are the pressures at the nodes or vertices of the network, and the flow on the edges. Both are bounded and subject to the Weymouth equation that links the pressure at end points of an edge with its flow and diameter.

Several types of problems can be considered in this framework.

- Operations problem 1: both the topology and diameters of pipes are fixed, and input and outputs are fixed. Minimize the energy used by the compressors.
- Operations problem 2: same as before, except that inputs and outputs may vary between certain bounds, and are available at given prices. Minimize the sum of cost of energy used by the compressors, and of the net revenue due to input and output flows. Note that this problem is meaningful even if the network does not include compressors.
- Static design: Fix the topology and diameters of the network so as to minimize the sum of investment and operations cost (possibly with various operations conditions in order to take into account e.g. seasonality).
- Investment planning: plan which investments should be done (at minimum cost) each year in order to take into account an increasing consumption. This includes the possibility of “doubling diameters”.

These problems have no analytic solutions. An exception is the static design for a gunbarrel sytem, for which the optimal diameter, number and location of compressors, and inlet-outlet pressures of these compressors can be computed in a

Published as Inria Research report RR-7796, November 2011. The authors thank Daniel de Wolf for useful remarks concerning the Belgium gas network model.

simple way, see André and Bonnans [2]. Sometimes dynamic programming techniques can be used, typically when the network has no loop, or perhaps a small number of them, or if the loops are of local nature; see Carter [9].

Except in these situations the problem is hard and various heuristic approaches have been suggested. As noticed by Maugis [16], for given input-output flows (with zero sum), and a network without compressors, one can solve simultaneously Kirchhoff's law for the flows and Weymouth's equation, by minimizing some strictly convex flow potential (squares of pressures are then interpreted as dual variables). The resulting flow may help designing a starting point for a local search algorithm. For instance, for solving gas transmission problems, de Wolf and Smeers [24] first minimize the potential under input-output flow bounds, and use the resulting flow as a starting point for a certain extension of the simplex method to the case of piecewise linear constraints (about the latter see de Wolf, Janssens de Bisthoven and Smeers [10]).

Another possibility is to make a linear perturbation of the potential e.g. in order to take into account the influence of compressors, and then to use this point as the starting point of a nonlinear local solver; see Babonneau, Nesterov and Vial [4].

Many optimization procedures include also the optimization of pipe diameters.

The DC (difference of convex functions approach, see e.g. Horst and Tuy [14]) is based on the fact that it is often natural to write the cost and constraints as differences of convex functions. Since concave functions are often easily underestimated by affine functions this gives the possibility of solving an optimization by a branch and bound approach. Zhang and Zhu [25] use a bilevel approach, simplifying the lower level problem by conjugate duality. Genetic algorithms are used in Abebe and Solomatine [1], Surry and Radcliffe [22], and Van Vuuren [23]. Hansen, Madsen, and Nielsen [11] use a trust-region successive linear programming approach. André, Bonnans and Cornibert [3] start with a convex relaxation and then use a local search heuristic that can be viewed as an uncomplete branch and bound method. Manojlovic, M. Arsenovic and Pajovic [15]. apply the successive-approximation (i.e. Hardy Cross) method for determining the optimal hydraulic solution of a gas-pipeline network.

In this paper we will consider the application to the gas network problem of the global optimization technique based on the combination of interval analysis with constraint propagation. These techniques have been well-established for a long time, see Messine [17, 18, 19], Carrizosa, Hansen and Messine [8], Hansen, Lagouanelle, and Messine [13], and the reference book Hansen and Walster [12].

Recently a patent [20, 21] was registered whose object is precisely to use the combination of interval analysis and constraint propagation in order to solve gas network optimization problems. However, the efficiency of the method is not established in [20, 21]. The aim of this paper is to show that this approach is effective when applied to the case of the Belgium network. This is a small network to which various local approaches have been applied, especially by Bakhouya and de Wolf [5, 6, 7]. The network has no loop, and hence, as we have stated before, one could find the global solution by a dynamic programming approach. Yet it is of interest to solve it by the interval analysis approach, which in principle works as well in the case of looped networks.

The paper is organized as follows. Section 2 briefly states the class of problems we are interested in. We then recall the approach of global optimization by the interval arithmetic approach in section 3. Constraint propagation is discussed in section 4. Numerical results are displayed in section 5. We conclude the paper in section 6.

2. Gas network optimization problems

The network is described by a set of nodes $\mathcal{N}$ and (directed) arcs $\mathcal{A} \subset \mathcal{N} \times \mathcal{N}$, and the arcs are partitioned into *compressive arcs* $\mathcal{A}_C$ and *passive arcs* $\mathcal{A}_P$. The equations to be satisfied are: Kirchhoff's law for the flow

$$\sum_{j:ij\in\mathcal{A}} f_{ij} - \sum_{j:ji\in\mathcal{A}} f_{ji} - s_i = 0, \quad i \in \mathcal{N}, \tag{1}$$

the compressors law

$$f_{ij}\left(k_1\left(\frac{\pi_i}{\pi_j}\right)^{k_3} - k_2\right) - P_{ij} = 0, \quad ij \in \mathcal{A}_C, \tag{2}$$

where P_{ij} is the amount of power used by the compressor, Weymouth's equation

$$\operatorname{sign}(f_{ij})\frac{f_{ij}^2}{C_{ij}^2} - \pi_i + \pi_j = 0, \quad ij \in \mathcal{A}_P, \tag{3}$$

as well as bound constraints

$$\underline{\pi_i} \le \pi_i \le \overline{\pi_i}, \quad i \in \mathcal{N}, \tag{4}$$

$$\underline{f_{ij}} \le f_{ij} \le \overline{f_{ij}}, \quad ij \in \mathcal{A}_C. \tag{5}$$

The cost function to be minimized is

$$\sum_{i\in\mathcal{N}} c_i s_i + \sum_{ij\in\mathcal{A}_C} c_{ij} P_{ij}. \tag{6}$$

Here c_i is the purchasing cost per unit, and c_{ij} is the energy cost.

3. Interval analysis and its application to global optimization

3.1. **Inclusion functions.** Let $\bar{\mathbb{R}} := \mathbb{R} \cup \{+\infty\} \cup \{+\infty\}$ denote the extended real line. In order to compute the global minimum of a function $f : \mathbb{R}^n \to \mathbb{R}$ over a set $K \subset \mathbb{R}^n$, i.e., to solve the optimization problem

$$\operatorname*{Min}_x f(x); \quad x \in K, \tag{P_K}$$

it is useful to be able to compute some bounds on the image by f of subsets of $\mathbb{R}^n$. Let $\mathcal{P}_B^n$ denote the set of *boxes*, i.e., subsets of $\bar{\mathbb{R}}^n$ of the form $\Pi_{i=1}^n[\alpha_i, \beta_i]$. We say that $F : \mathcal{P}_B^n \to \bar{\mathbb{R}}$ is an *inclusion function* for f over X if

$$x \in X \Rightarrow f(x) \in F(X), \quad \text{for all } X \in \mathcal{P}_B^n. \tag{7}$$

In other words, F provides upper and lower bounds of f over X. We assume that the feasible set K is defined by finitely many inequalities of the form

$$K = \{x \in \mathbb{R}^n;\ \ g_i(x) \in [a_i, b_i],\ i = 1, \dots, p\}. \tag{8}$$

We will denote by F_i some inclusion functions for g_i, $i = 1$ to p. We have the following *feasibility test* for problem (P_K):

$$\begin{cases} \text{The set } K \text{ is empty if } K \subset X \in \mathcal{P}_B^n, \text{ and} \\ F_i(X) \cap [a_j, b_j] = \emptyset, \quad \text{for some } 1 \le j \le p. \end{cases} \tag{9}$$

So the *basic operation* to be performed on problem (P_K) are

$$\left\{\begin{array}{ll} \text{(i)} & \text{Input } f,\, K. \text{ Choose } X \in \mathcal{P}_B^n \text{ containing } K. \\ \text{(ii)} & \text{Perform the feasibility test and compute } F(X). \\ \text{(iii)} & \text{Compute, if possible, a point } x_K \in K \text{ and } f(x_K). \end{array}\right. \tag{10}$$

The third step is typically performed by computing a point in K close to the center of X. In case of success this gives an upper bound of the value of (P_K). We will give more details later about our implementation.

3.2. **Interval analysis.** Interval analysis is an effective way to implement the computation of inclusion functions. We have to distinguish the case of unary and binary operators.

Unary operators are in practice the usual nonlinear functions such as logarithm, exponential, trigonometric functions, absolute value, etc. If such an operator say h is nondecreasing (resp. nonincreasing) we have that, denoting in the sequel the inclusion function as the function itself:

$$f([\alpha, \beta]) = [f(\alpha), f(\beta)] \quad \text{resp. } [f(\beta), f(\alpha)]. \tag{11}$$

In the case of e.g. the sine or absolute value function, the operator is piecewise monotone and it is also easy to give an exact expression of the image of an interval.

Binary operators involve mainly the four arithmetic operations and some function as the maximum. For the latter we have that,

$$\max([\alpha, \beta], [\alpha', \beta']) = [\max(\alpha, \alpha'), \max(\beta, \beta')]. \tag{12}$$

For the addition we have that

$$[\alpha, \beta] + [\alpha', \beta'] = [\alpha + \alpha', \beta + \beta']. \tag{13}$$

The subtraction is equivalent to the addition of the opposite of the second term:

$$[\alpha, \beta] - [\alpha', \beta'] = [\alpha, \beta] + [-\beta', -\alpha'] = [\alpha - \beta', \beta - \alpha']. \tag{14}$$

The multiplication has a slightly more complicated expression since signs enter into account:

$$[\alpha, \beta] * [\alpha', \beta'] = [\min(\ell), \max(\ell)] \quad \text{where } \ell := \{\alpha\alpha', \alpha\beta', \beta\alpha', \beta\beta'\}. \tag{15}$$

Finally the division is nothing more than the product with the inverse of the second term, and (assuming for the sake of simplicity that α, β are nonzero)

$$[\alpha, \beta]^{-1} = \left\{\begin{array}{ll} [\beta^{-1}, \alpha^{-1}] & \text{if } \alpha, \beta \text{ are nonzero and of same sign,} \\ [-\infty, +\infty] & \text{otherwise.} \end{array}\right. \tag{16}$$

When evaluating an expression we may associate it with a tree of computations of unary or binary operators and we apply the previous expressions inductively.

3.3. **Branching.** The basic idea is as follows. Let K be contained in the box $X = \Pi_{i=1}^n [\alpha_i, \beta_i]$. By *branching on component* j, $1 \leq j \leq n$, with parameter $\gamma \in (\alpha_i, \beta_i)$, we mean considering the two boxes

$$X_1 := \{x \in X;\ x_j \in [\alpha_j, \gamma]\}; \quad X_2 := \{x \in X;\ x_j \in [\gamma, \beta_j]\}, \tag{17}$$

the sets $K_i := K \cap X_i$, $i = 1, 2$, and the "subproblems" (P_{K_1}) and (P_{K_2}). By induction a list of pairs $L = \{(K_i, X_i),\ i \in I\}$ is generated, where the index set I is finite, the K_i are of the form $K \cap X_i$, and $X_i \in \mathcal{P}_B^n$. We can describe the resulting algorithm as follows, where θ denotes the upper bound of $\min\{f(x); x \in K\}$, and X is a box containing K.

Branch and bound algorithm

(1) Data: f, $K_1 := K$, $X_1 := X$.
(2) **Initialization** $L := \{(K_1, X_1)\}$; $\theta := +\infty$, $k := 0$, $I := \{1\}$.
(3) $k := k + 1$. Choose $i \in I$.
(4) **Branching**
Branch (K_i, X_i) into say (K', X') and (K'', X'').
Eliminate (K_i, X_i) from L.
Compute θ', $\theta'' \in \mathbb{R} \cup \{+\infty\}$, upper bounds of
$\min\{f(x); x \in K'\}$ and $\min\{f(x); x \in K'\}$, resp.
(5) **Elimination**
Set $\theta := \min(\theta, \theta', \theta'')$.
Remove from L any pair (K_j, X_j), $j \in I$, such that $\theta < \min(F(X_j))$.
(6) **Inclusion**
If $\min(F(X')) \leq \theta$, include (K', X') in L.
If $\min(F(X'')) \leq \theta$, include (K'', X'') in L.

In practice the update of the list L is performed by sorting the elements by say increasing order of $\min(F(X_i))$, so that the elimination step is immediate. The computational costs consist in evaluating $F(\cdot)$ and searching for a feasible point.

4. Constraint propagation

Consider a constraint of the form

$$\varphi(x) \in [a, b]. \tag{18}$$

Using the interval analysis for φ, we may sometimes reduce the interval $[a, b]$. Consider first the case when we can eliminate a component x_k, that is, denoting as usual by x_{-k} the vector of components of x_k except for the kth one, write

$$\varphi(x) = c \quad \Leftrightarrow \quad x_k = \psi(x_{-k}, c). \tag{19}$$

Denote by $X(x_{-k})$ the box in which x_{-k} is included. If F_ψ is an inclusion function for ψ, we deduce that

$$x_k \in F_\psi(X(x_{-k}), [a, b]). \tag{20}$$

We may then propagate the interval reduction (if any) to some other constraints where the kth component of x enters.

In our implementation we will content ourself with the simpler but effective back-propagation approach. That is, each constraint is evaluated in the interval analysis approach, by considering a tree representation of the formula, starting from the leaves and finishing by the root. Once this is done, we propagate backward the interval estimate of subexpression, taking into account the properties of the operators. For instance, if (18) is written as

$$\varphi_1(x) + \varphi_1(x) \in [a, b], \tag{21}$$

then in the course of evaluating φ we have obtained intervals say $[\alpha_i, \beta_i]$ for $\varphi_i(x)$, $i = 1, 2$, and (18) implies then

$$\varphi(x) \in [\alpha_1 + \alpha_2, \beta_1 + \beta_2] \cap [a, b]. \tag{22}$$

So we update if necessary

$$a := \max(a, \alpha_1 + \alpha_2); \quad b := \min(b, \beta_1 + \beta_2). \tag{23}$$

And then we have that for e.g. $\varphi_1(x)$:

$$\varphi_1(x) \in [\alpha_1, \beta_1] \cap ([a, b] - [\alpha_2, \beta_2]) , \tag{24}$$

which possibly allows to improve the interval estimate for $\varphi_1(x)$; and we can back propagate these estimates along the formula tree. We observe that the amount of additional computations is quite small. Of course the interval reductions depend on the order in which evaluation of functions is made.

The constraint propagation and branch and bound approaches can be combined in a quite natural way: when branching over component j, all interval estimates for subexpressions are inherited form the previous step, and all expressions in which x_j is involved may potentially held reducing the intervals of their subexpressions; these improvements may in turn be used for the remaining expressions.

In our implementation we propagate several times at each iteration in order to remove inconsistent values. We arbitrarily choose to repeat the propagation 10 times.

Finding a feasible value at each iteration is a difficult but key step. The idea we are using consists in progressively reducing the domains of the variables. We divide by half the domain of a variable and then propagate the constraints to remove inconsistent values with this new interval. We iterate this process until all the domains have a negligible size.

$\epsilon = 0.001$
while $\exists\ x \in [a, b]$ with $b - a > \epsilon$ **do**
    $[a, b] = \left[\frac{a+b}{2} - \frac{b-a}{4}, \frac{a+b}{2} + \frac{b-a}{4}\right]$
    Propagate the constraints (10 times).
end while

We did not find this kind of procedure in the literature. Of course we have no guarantee of finding a (near) feasible point. However it proved to be quite effective in our example, since it provided a feasible point at each visited node.

5. Numerical results

In order to test the algorithm, we apply it to the Belgium gas network. The goal is to minimize the energy used by the two compressors of the network subject to constraints on pressure and flow at each node of the network. This network as well as all the constraints of the problem are fully described in [5]. The problem has 60 variables: 2 compressor powers, 6 input and 9 output flows, 21 flows on arcs, pressure at the 22 nodes.

5.1. **Algorithm and heuristics.** For the particular constraints of this example we build the corresponding tree for each constraint. For example the tree corresponding to the compressor constraint (2) is represented on figure 1.

Concerning the exploration tree, several heuristics can be used and we notice that the speed of the algorithm is greatly influence by their choice. We used the *best first search*: at each step we select the node which has the best feasible value. The branching step is made by splitting in two the interval of the variable that has the largest domain.

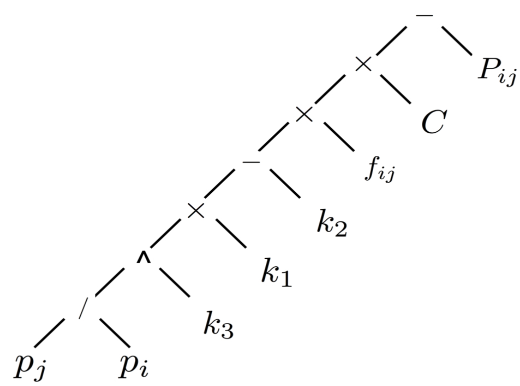

FIGURE 1. Tree representing the compressor constraint. C is a constant used for the unit conversion.

5.2. **Solution.** The power used by each compressor are the two variables that appear in the objective function and therefore these are the variables we are branching on. Let $\delta = 0.1$ be the precision on the final intervals. The best feasible value returned by the algorithm is

$$z_1^* = 6832.7 \text{ keuro per day}$$

This result was reached after having visited 48 nodes, which took 344 seconds on a PC.

At the end of the algorithm some nodes of the exploration tree have not been removed. These are the boxes in which the global optimum lies. We get at the end, still with a precision of 0.1, three boxes which are:

$$I_1 = \begin{cases} [5368.85, 5368.93] \\ [780.487, 780.547] \end{cases} \quad I_2 = \begin{cases} [5368.85, 5368.93] \\ [780.547, 780.607] \end{cases} \quad I_3 = \begin{cases} [5368.93, 5369.01] \\ [780.475, 780.541] \end{cases}$$

The three domains are close to each other. In fact they even form a connex domain. Therefore, we have a very precise idea of the location of the global optimum.

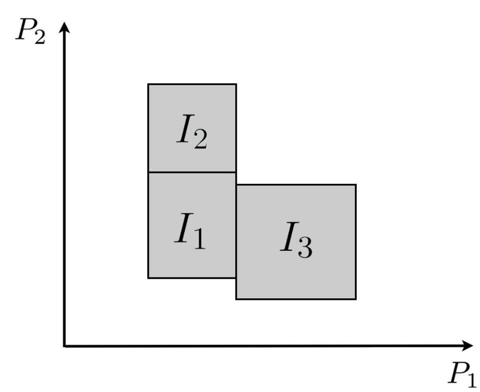

FIGURE 2. Visual representation of the three remaining boxes.

5.3. **Comparison.** The result obtained in [5] is

$$z_2^* = 6831.892 \text{ keuro per day}$$

which was only known to be a local optimum.

We notice that we have reached almost the same value for the criteria. Since the remaining boxes correspond to small domain and since the result in [5] is a local minimum, we can claim that that this local optimum is either the global optimum, or is very close.

5.4. **Behaviour of the algorithm.** The procedure for searching a feasible point happened to be successful for all visited nodes. In order to give a more precise description of the behavior of the algorithm in practice, we look at the number of nodes that are reached in the exploration tree depending on the precision δ that is required on the result. These values are represented on figure 3. This shows a smooth and sharp increase of the running time of the algorithm when the precision becomes better.

We have also represented on figure 4 the evolution of the value of the objective function for the best feasible solution regarding the precision we set up. The improvement of the criteria turn out to be linear with the precision and as a result there is a clear convergence of the algorithm toward the global optimum.

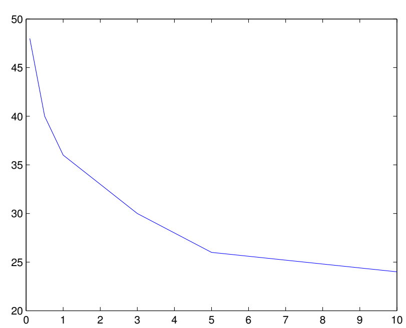

FIGURE 3. Number of nodes for several precision values.

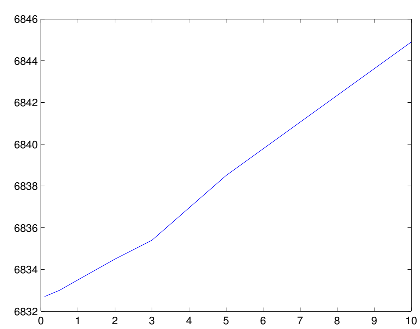

FIGURE 4. Best criteria for several precision values.

The following table gives the values of graph 3 and 4.

Precision δ	Number of nodes	Best criteria z^*
10	24	6844.9
5	26	6838.5
3	30	6835.4
2	33	6834.5
1	36	6833.5
0.5	40	6833.0
0.1	48	6832.7

Table of values used in figures 3 and 4.

We observe that the number of visited nodes remains quite small. Since the number of variables is quite large, this is a sign of the effectiveness of the constraint propagation procedure. This is confirmed by figures 5 and 6, in which for each visited node, we display the volume of compressor variables, and the volume of the box for all variables. We observe that both quantities decrease quite approximately as geometric sequences.

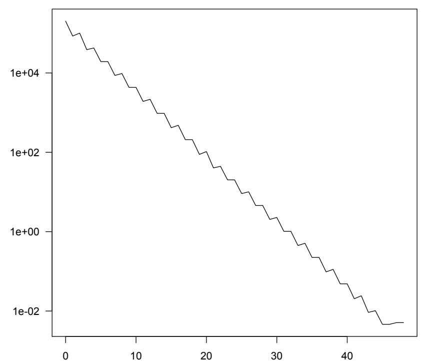

FIGURE 5. Volume of compressor variables function of visited node.

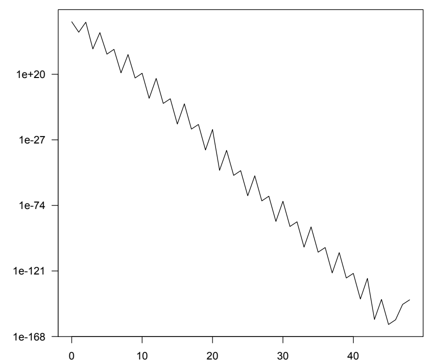

FIGURE 6. Volume of boxes function of visited node.

6. CONCLUSION

It would be of interest of course to deal with larger problems, without doubt with more elaborated algorithms. For instance we could have parameterized the flow over the network by the input and output flows (and flows on looping arcs whenever they are present), in order to reduce the number of optimization variables. We could also express the differences of squares of pressure as function of the flow along a spanning tree (adding of course compatibility relations for loops, if any). We also observe that we could try to refine the bounds on the variables by solving a convexified problem. The extensions of affine arithmetic in [18] could also enhance the results. In any case we hope that our results will convince researchers in the field that there is room for improvement with this kind of methods for solving nonlinear network problems.

REFERENCES

[1] A.J. Abebe and D.P. Solomatine. Application of global optimization to the design of pipe networks. In *Proc. 3rd International Conference on Hydroinformatics, Copenhagen*, pages 989–996, Copenhagen, Denmark, 1998.
[2] J. André and J.F. Bonnans. Optimal structure of gas transmission trunklines. *Optimization and Engineering*, 12(1-2):175–198, 2011.

[3] M. André, J.F. Bonnans, and L. Cornibert. Planning reinforcement on gas transportation networks with optimization methods. *European J. Operational Research*, 197(3):1019–1027, 2009.
[4] F. Babonneau, Y. Nesterov, and J.-P. Vial. Design and operations of gas transmission networks. *Operations Research*, 2011. To appear.
[5] B. Bakhouya and D. De Wolf. The gas transmission problem when the merchant and the distribution functions are disconnected. Technical report, Université du Littoral, Dunkerque, 2007.
[6] B. Bakhouya and D. De Wolf. Optimal dimensioning of pipe networks: the new situation when the distribution and the transportation functions are disconnected. Technical report, Université du Littoral, Dunkerque, 2008.
[7] B. Bakhouya and D. De Wolf. Solving gas transmission problems by taking compressors into account. Technical report, Université du Littoral, Dunkerque, 2008.
[8] Emilio Carrizosa, Pierre Hansen, and Frédéric Messine. Improving interval analysis bounds by translations. *J. Global Optim.*, 29(2):157–172, 2004.
[9] Carter. Pipeline optimization: dynamic programming after 30 years. In *Proc. 30th PSIG Annual Meeting*, 1998.
[10] D. de Wolf, O. Janssens de Bisthoven, and Y. Smeers. The simplex algorithm extended to piecewise linearly constrained problems ii: An application to the gas transmission problem. Technical Report 1991003, CORE, Louvain la Neuve, 1991.
[11] Claus Thorp Hansen, Kaj Madsen, and Hans Bruun Nielsen. Optimization of pipe networks. *Math. Programming*, 52(1, (Ser. B)):45–58, 1991. Applications of mathematical programming (Tokyo, 1988).
[12] E. Hansen and G.W. Walster. *Global optimization using interval analysis*, volume 264 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker Inc., New York, 2004. Second edition.
[13] Pierre Hansen, Jean-Louis Lagouanelle, and Frédéric Messine. Comparison between Baumann and admissible simplex forms in interval analysis. *J. Global Optim.*, 37(2):215–228, 2007.
[14] Reiner Horst and Hoang Tuy. *Global optimization*. Springer-Verlag, Berlin, second edition, 1993. Deterministic approaches.
[15] V. Manojlovic, M. Arsenovic, and V. Pajovic. Optimized design of a gas-distribution pipeline network. *Applied Energy*, 48(3):217–224, 1994.
[16] J.J. Maugis. Etude de réseaux de transport et de distribution de fluide. *RAIRO Operations Research*, 11:243–248, 1977.
[17] F. Messine. *Méthodes d'Optimisation Globale basées sur l'Analyse d'Intervalle pour la Résolution des Problèmes avec Contraintes*. PhD thesis, LIMA-IRIT-ENSEEIHT-INPT, Toulouse, 1997.
[18] Frédéric Messine. Extensions of affine arithmetic: application to unconstrained global optimization. *J.UCS*, 8(11):992–1015 (electronic), 2002.
[19] Frederic Messine. Deterministic global optimization using interval constraint propagation techniques. *RAIRO Oper. Res.*, 38(4):277–293, 2004.
[20] E. Peureux, B. Casoetto, T. Pillay, and M. Benoît. Procédé d'optimisation automatique d'un réseau de transport de gaz naturel, April 2007. European patent.
[21] E. Peureux, B. Casoetto, T. Pillay, and M. Benoît. Method for the automatic optimization of a natural gas transport network, April 2009. US patent.
[22] P.D. Surry and N.J. Radcliffe. The COMOGA method: Constrained optimisation by multi-objective genetic algorithms. *Control and Cybernetics*, 26, 1997.
[23] Van Vuuren. Application of genetic algorithms: Determination of the optimal pipe diameters. *Water S.AA*, 28(2):217–223, 2002.
[24] Daniel De Wolf and Yves Smeers. The gas transmission problem solved by an extension of the simplex algorithm. *Management Sciences*, 46(11):1454–1465, 2000.
[25] Jianzhong Zhang and Detong Zhu. A bilevel programming method for pipe network optimization. *SIAM J. Optim.*, 6(3):838–857, 1996.

J.F. Bonnans, INRIA-Saclay and Centre de Mathématiques Appliquées, Ecole Polytechnique, 91128 Palaiseau, France
E-mail address: Frederic.Bonnans@inria.fr

G. Spiers, Ecole Polytechnique, 91128 Palaiseau, France
E-mail address: gregoire.spiers@polytechnique.org

J.-L. Vie, Ecole Polytechnique, 91128 Palaiseau, France
E-mail address: jean-leopold.vie@polytechnique.org