WATER RESOURCES RESEARCH, VOL. 30, NO. 9, PAGES 2637–2646, SEPTEMBER 1994

Optimal design of water distribution networks

Gideon Eiger
Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Haifa

Uri Shamir
Faculty of Civil Engineering, Technion–Israel Institute of Technology, Haifa

Aharon Ben-Tal
Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Haifa

Abstract. Optimal design of a water distribution network is formulated as a two-stage decomposition model. The master (outer) problem is nonsmooth and nonconvex, while the inner problem is linear. A semi-infinite linear dual problem is presented, and an equivalent finite linear problem is developed. The overall design problem is solved globally by a branch and bound algorithm, using nonsmooth optimization and duality theory. The algorithm stops with a solution and a global bound, such that the difference between this bound and the true global optimum is within a prescribed tolerance. The algorithm has been programmed and applied to a number of examples from the literature. The results demonstrate its superiority over previous methods.

Introduction

A water distribution network is a system of hydraulic elements (pipes, pumps, valves, reservoirs) which are connected together to convey given quantities of water, within prescribed pressures, from sources to consumers.

Such a system can be described as a graph in which the nodes represent the sources, consumption points, and control elements, and the links represent the connecting pipes (Figure 1).

The overall planning process of water distribution networks consists of three phases: layout, design, and operation. Although each phase is dependent on the others, they can be formulated and solved as separate problems. The complete planning process is then carried out by iterating on these three phases. In this work we address the design phase, which has been investigated extensively for more than three decades by many researchers. Many different problem formulations and solution methods have been proposed and tested. Still, the following fact seems to be generally accepted: With existing optimization tools, it is not possible to solve the general optimal design problem with its full complexity [see *Walski*, 1987].

The design problem is difficult, mainly due to the following facts: (1) The problem contains discrete elements, for example, pumps, valves, and pipe segments. (2) Any formulation of the problem that is realistic enough to be useful is nonlinear and nonconvex. (3) Even a moderate size problem is of a rather high dimension.

For simplicity of presentation we consider here simple networks, which do not contain pumps and/or reservoirs. Note, however, that with an appropriate formulation, networks with all these elements have also been handled by the method presented in this work [*Eiger*, 1991]. Consider a looped water distribution network with a given topology and topography. Assume that one or more water sources in the network can supply water at a known (fixed) head. Assume, also, that at a given set of nodes there are known water demands, which must be supplied within a prescribed range of pressures.

The objective is to find the lowest cost network which can supply the demands under the given restrictions. Following the original formulation by *Alperovits and Shamir* [1977], which has been adopted in subsequent work [*Quindry et al.*, 1981; *Saphir*, 1983; *Fujiwara et al.*, 1987; *Fujiwara and Khang*, 1990] and restated in a more concise and complete form by *Kessler and Shamir* [1989], the decision variables are (1) lengths of all pipe segments in the network, making each link of one or more fixed diameter segments, where pipe diameters are taken from a given discrete set, and (2) flows (rates and directions) in all network links.

We use the same formulation as that of *Kessler and Shamir* [1989], but our solution method is entirely different. We set out to resolve the following difficulties involved with the design problem: (1) Because of the nonconvexity, one cannot be satisfied with a local solution and some global search is required. (2) A tight lower bound must be found in order to evaluate the quality of the solution found. (3) Since the optimization is nonsmooth, we must use a nonsmooth algorithm that can handle minimization of nonconvex functions.

Nonsmoothness is handled by the bundle-trust (BT) algorithm of Zowe and Schramm [*Schramm*, 1989]. BT is an algorithm which belongs to the category of bundle methods (see the appendix). For a discussion of nonsmooth optimization techniques, see *Shor* [1985], *Kiwiel* [1985], and *Zowe* [1986]. In order to obtain a bound on the objective value, we use duality theory. A dual problem, paired with the primal original problem, is found using ordinary Lagrangian duality. The dual is solved to give the required bound. The dual problem is a semi-infinite linear problem, that is, it has a

Paper number 94WR00623.
0043-1397/94/94WR-00623$05.00

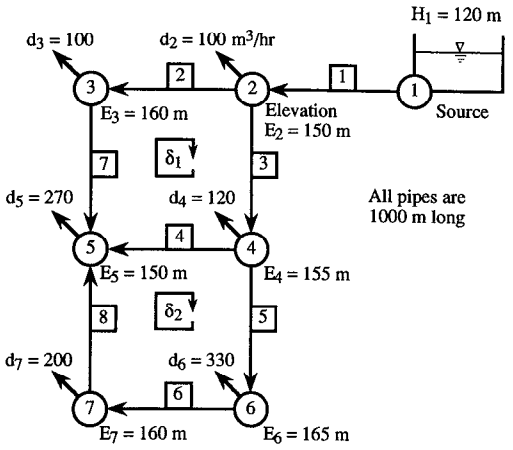

Figure 1. The two-loop network [*Alperovits and Shamir*, 1977].

linear objective function and infinitely many linear constraints and is not easy to solve effectively. We take advantage of the specific structure of the problem to show that the dual is equivalent to a finite linear problem, which can be solved effectively by any standard linear programming code.

The solution algorithm is a combination of a primal process (used to improve local solutions) and a dual process (used to tighten the bound). The two processes are combined in a branch and bound type algorithm. The algorithm stops when the gap between the best found value and the global lower bound is within a stated tolerance. Of course, the running time is dependent on that tolerance.

Review of Previous Research

Numerous solution methods have been developed to solve the network optimization problem, using different formulations. They range from rules of thumb, based on engineering experience and insight, through heuristics used for specific network types, to optimization by mathematical programming techniques. The last category includes linear, nonlinear, dynamic and mixed integer programming. Some of these, which lead up to the approach presented here, are reviewed briefly.

The linear programming gradient (LPG) method [*Alperovits and Shamir*, 1977] was the beginning of a new approach, in which the optimization problem is solved (locally) by iterating between two stages. In the first stage the flows are given, and the optimal pipe segments and node pressures are found by solving a linear problem. In the second stage the flows are modified, using an approximation of the "gradient" of the objective function. *Alperovits and Shamir* [1977] and subsequent work based on the same formulation [*Quindry et al.*, 1981; *Saphir*, 1983; *Fujiwara et al.*, 1987; *Kessler and Shamir*, 1989; *Fujiwara and Khang*, 1990] ignored the fact that the gradient does not always exist or considered this fact to be insignificant. In most cases a suboptimal yet feasible solution could be found. LPG can be applied for looped networks which include pumps and/or reservoirs.

Kessler and Shamir [1989] used a graph theory formulation, adding insight to the problem and reformulating the model in matrix notations. *Fujiwara and Khang* [1990] made some effort toward a global search, but no bound was produced to evaluate the gap between the solution and the global optimal value. (Readers interested in previous works are referred to *Walski* [1985], *Kessler and Shamir* [1989], *Goulter et al.* [1986], *Charles Howard and Associates, Ltd.* [1986], *Karmeli et al.* [1968], *Kessler* [1988], *Morgan and Goulter* [1985], *Quindry et al.* [1979], and *Rowell and Barnes* [1982].)

Problem Formulation and Theoretical Notes

Problem Formulation

Consider the design problem presented in the introduction. To meet space limitations, we present here the model for networks of pipes only, with no pumps, boosters, or valves. The method has been developed and tested for the full problem [*Eiger*, 1991]. Following *Kessler and Shamir* [1989], we formulate the problem as follows:

$$\min c^T x: A(q)x \le b$$
$$x \ge 0 \tag{1}$$
$$q \in Q$$

where

- q vector of flows in network links;
- Q set of all flow vectors which maintain continuity of flow at all network nodes;
- x vector of lengths of all pipe segments (allowing each link to be made of segments of all possible diameters);
- $A(q)$ matrix which is dependent on q and represents the constraints which result from physical laws of energy conservation and length restrictions;
- c vector of pipe segment unit length costs;
- b right-hand side vector.

A full, detailed description of the model is given by *Kessler and Shamir* [1989]. Since we are using the same model we shall not repeat it here, except for those parts which are needed for completeness and clarity of presentation.

The constraints $A(q)x \le b$ are of three types:

$$A^1(q)x = b \tag{2}$$

$$A^2(q)x \le \Delta h_p \tag{3}$$

$$A^3 x = a \tag{4}$$

The number of equations in (2) is equal to the number of basic loops in the network, where a basic loop set is a minimal (not necessarily unique) set of loops, such that all other loops can be represented as a linear combination of them [*Kessler and Shamir*, 1989]. The ith constraint in (2) states that the hydraulic energy loss on the ith basic loop is equal to the head difference between its ends; $b = 0$ for closed loops. It can be shown [*Kessler and Shamir*, 1989] that imposing such a constraint on all basic loops ensures this condition for any other loop. The block (3) has one inequality for each path, over which the allowed loss of hydraulic energy, $(\Delta h_p)_i$, is restricted. The block (4) has one equation for each link in the network. The ith equation states that the sum of lengths of the pipe segments that make up the

ith link must be equal to the given length of that link a_i. The total number of constraints in (2), (3), and (4) is denoted n.

The specific form of the terms (2) and (3) is obtained using the Hazen-Williams equation, which relates the head loss over a unit length of pipe to flow and pipe properties. Let d be the diameter of the pipe and q the flow. Denote by $J(d, q)$ the head loss over a unit length of that pipe, in the direction of the flow, given by the Hazen-Williams equation,

$$J(d, q) = \alpha(\text{chw}^{-1.852})d^{-4.87}(q^{1.852}) \tag{5}$$

where α is a coefficient which depends on the units used and chw is the Hazen-Williams coefficient, which is a measure of pipe smoothness.

Let R be the directed incidence matrix of links on nodes. Assume that a direction has been chosen arbitrarily for each link.

$$\begin{aligned} R_{ij} &= 1 \quad \text{if link } j \text{ is directed toward node } i \\ R_{ij} &= -1 \quad \text{if link } j \text{ is directed away from node } i \\ R_{ij} &= 0 \quad \text{if link } j \text{ is not adjacent to node } i \end{aligned} \tag{6}$$

Let w be a vector such that its ith component w_i is the demand for water at node i, and define the set of flows in all links of the network which maintain continuity at nodes:

$$Q = \{q: Rq = w\} \tag{7}$$

Q is an unbounded set. However, we shall use a bounded subset, contained in Q, with the following box definition:

$$Q = \{q: Rq = w,\ q_i^{\min} \le q_i \le q_i^{\max}\ \forall i\} \tag{8}$$

where $q^{\min}$ and $q^{\max}$ are parameters whose values are determined heuristically. We shall return to this point later.

The model (1)–(4) is smooth (although nonconvex), but because of its high dimension, it cannot be conveniently solved. Instead, we use the following formulation:

$$\min_{q \in Q} \varphi(q) = \min_{x \ge 0} c^T x: A(q)x \le b \tag{9}$$

The inner problem

$$\min_{x \ge 0} c^T x: A(q)x \le b$$

is linear in x for any $q \in Q$, and can be solved by a linear programming code. The objective function in the master (outer) problem in (9) is recognized as an optimal value function. It follows that the outer problem can be nonsmooth [*Clarke*, 1983]. Nonconvexity of both models (1)–(4) and (9) follows from the fact that the left-hand side functions in blocks (2) and (3) are nonconvex in (x, q) space.

Dual Formulation

By solving either (1)–(4) or (9) we can find, at best, a local optimal solution, which (due to nonconvexity) is usually different from the global optimum and depends on the starting point q^0. This calls for a global search, producing an improving sequence of local solutions. In order to evaluate the quality of a current solution and to have a stopping criterion, we must produce a global lower bound on the objective value. Such a bound can be computed, according to duality theory, by solving a dual problem, paired with the primal, original problem. To be useful, the bound should be tight.

By standard Lagrangian duality we get the following dual problem paired with the full original problem (1)–(4):

$$\max_{y \in Y} -b^T y: A(q)^T y + c \ge 0\ \ \forall q \in Q \tag{10}$$

where y is the vector of Lagrange multipliers and Y is a set of vectors y, defined as follows: For i such that the ith constraint in (2)–(4) is an equation, y_i is not restricted in sign; for i such that the ith constraint in (1)–(4) is an inequality, $y_i \ge 0$.

Problem (10) is a semi-infinite linear problem, that is, it has a linear objective function and infinitely many linear constraints (one finite block of constraints for each q in the infinite set Q). At first glance, it is not clear how (10) can be solved effectively. However, $A(q)$ has a special structure, which gives rise to some very useful mathematical properties.

Solving the Dual Problem

The statement of theorem 1 is as follows: The rows $A_j(q)$ of $A(q)$ in (10) are continuous functions of q, and Q is a box given by (8). The infinite linear problem (10) is therefore equivalent to the following finite-constrained linear problem:

$$\max_{y \in Y} -b^T y: A_j(q^k)^T y + c_j \ge 0 \quad j = 1, \cdots, n;\ k = 1, 2 \tag{11}$$

where

$$q^1 = q^{\min} \qquad q^2 = q^{\max}$$

The proof is as follows: The left-hand side of the jth constraint in (11) has the following functional form:

$$v_j(q) = k_j \text{ sign } (q_j)|q_j|^{1.852} w_j(y) + c_j \tag{12}$$

where $k_j > 0$, $w_j(y)$ is a linear function of y which does not depend on the flow q, and q_j is a component of the vector q. It is easy to show that $v_j(q)$ is a monotonic function of q_j (although the direction of monotonicity may not be the same for all constraints). As q_j is bounded by $q_j^{\min} \le q_j \le q_j^{\max}$ the minimum of the monotonic function $v_j(q)$ is at one of the bounds. It is therefore sufficient to have in (11) the constraints formulated at the bounds only, instead of the infinite set in (10).

Reducing the Problem Dimension

The dimension of the outer problem in (9) can be reduced significantly by using an affine transformation, which leads to a change of variables. Let L be the directed incidence matrix of links on a set of basic loops. Assume that each basic loop in the set has been assigned an arbitrary but fixed positive direction, and assume the same for each link. A link is assumed to be positively (negatively) directed in a loop if it belongs to the loop and their mutual positive directions agree (do not agree). The matrix L is defined as follows:

$$\begin{aligned} L_{ij} &= 1 \quad \text{if link } j \text{ belongs to loop } i, \text{ positively directed} \\ L_{ij} &= -1 \quad \text{if link } j \text{ belongs to loop } i, \text{ negatively directed} \\ L_{ij} &= 0 \quad \text{if link } j \text{ does not belong to loop } i \end{aligned} \tag{13}$$

Let e be the cardinality of the set of links (also the dimension of q), and u be the cardinality of a basic loop set. Consider the definition of Q in the form (7), i.e., $Q = \{q: Rq = w\}$. From linear algebra theory, any vector $q \in Q$ can be written uniquely as a sum of a solution of the linear equation system $Rq = w$ (say q^0) and some vector from the null space of the matrix R (say $\bar{q}$). It is easy to verify that the null space of R is given by

$$\bar{q} \in \text{Null } R \Leftrightarrow \bar{q} = L^T\delta \qquad \text{for some } \delta \in R^u. \tag{14}$$

We have

$$q^0 \in Q \Rightarrow q(\delta) = q^0 + L^T\delta \in Q \qquad \forall\, \delta \in R^u. \tag{15}$$

It follows that the next set equality is valid: If and only if $q^0 \in R^e$ and $Rq^0 = w$, then

$$Q = Q' = \{q(\delta) \in R^e:\ q(\delta) = q^0 + L^T\delta, \text{ taking all } \delta \in R^u\} \tag{16}$$

Note that a change of δ can be interpreted as a simultaneous change in flow, independently for each basic loop. Changing the ith component of δ is equivalent to decreasing or increasing (depending on the direction of change) the flows in all links which belong to the ith basic loop by an equal amount. This technique was first proposed by *Alperovits and Shamir* [1977].

The above results enable work in the dimension u, which is typically much smaller than e. In fact, a box Δ is used as the solution space instead of a box Q.

Calculation of Subgradients

To use the BT algorithm, one has to supply one (arbitrary) subgradient of the objective function at each iteration. A discussion of the theoretical aspects of calculating subgradients, with specific reference to the model (9), is given by *Ben-Tal et al.* [1994]. Here we cite only the essential results.

The nondifferentiable problem we solve is

$$\inf_{\delta \in R^u} \varphi(\delta) \tag{17}$$

where δ is the vector of discharge changes in the u basic loops and $\varphi(\delta)$ is the optimal value of the internal problem

$$\inf C^T x \tag{18}$$

subject to

$$L\bar{I}J(q^0 + L^T\delta)x = 0 \tag{19}$$

$$P\bar{I}J(q^0 + L^T\delta)x \le \Delta h_p \tag{20}$$

$$\bar{I}x = a \tag{21}$$

$$x \ge 0 \tag{22}$$

where $\bar{I}$ is a matrix which represents the arrangement of the pipe segments within the links, J is a vector of the hydraulic gradients in the pipes, given by (5), and P is the path matrix which represents the arrangement of the links within the paths from a reference node. Equations (19), (20), and (21) are the specific forms of (2), (3), and (4), respectively.

Suppose we have solved the inner problem for some δ^0, and we have the primal solution x_0 and the corresponding dual solution. *Ben-Tal et al.* [1992, equation (4.6)] show that a subgradient $g \in \partial\varphi(\delta^0)$ can be calculated from:

$$g = \sum_{i \in u} \mu_{0i}\{\nabla_\delta[L\bar{I}J(q^0 + L^T\delta^0)]_i\}^T x_0 + \sum_{i \in I(\delta^0, x_0)} \lambda_{0i}\{\nabla_\delta[P\bar{I}J(q^0 + L^T\delta^0)]_i\}^T x_0 \tag{23}$$

where

x_0	optimal solution of the inner problem;
μ_0	optimal dual subvector associated with (19);
λ_0	optimal dual subvector associated with (20);
∇_δ	derivative with respect to δ;
$I(\delta^0, x_0)$	active subset of constraints in (20).

The mathematical conditions under which (23) holds are specified by *Ben-Tal et al.* [1992]. It is, however, not practical to check at every iteration whether they are satisfied. Since there is no known parallel formula for other situations, when the conditions are not met, the best strategy anyhow is to use (23). If the subgradient given by it is not correct at some point during the iterations (which could happen, but infrequently), then the resulting local direction may not be the best, and this will be compensated for by subsequent good directions at (most) points where the conditions are met. The use of (23) has proven itself in practice [*Eiger*, 1991].

Branch and Bound Type Algorithm

Reducing the Duality Gap

From standard duality theory, it is known that a bound on the primal optimal value is obtained by solving an appropriate dual problem. When the primal is not convex, as in our case, there will usually be a gap between the primal and dual optimal values, called the duality gap, and it can be too large to be helpful. We next show how a reduction of the duality gap can be obtained.

We state a theoretical result on which the method is based. Rigorous treatment and proofs are given by *Ben-Tal et al.* [1994], where more general results were obtained.

Recall that the variable q of the outer problem in (9) is allowed to vary inside the box Q (8). The dependency of the optimal value of (9) on Q is emphasized in the following notation:

Denote by P_Q the primal problem (9) and by D_Q its dual (10). Let q^* be a point in the optimal solution set of (9). Define the following function (using notation of the previous section):

$$d(\lambda) = \sup_{y \in Y} (-b^T y:\ A(q)^T y + c \ge 0,\ \forall\, q:\ \|q - q^*\| \le \lambda,\ \lambda \ge 0) \tag{24}$$

With the above notations we state the following results: We first state theorem 2:

$$d(\lambda) \text{ is continuous at } \lambda = 0 \tag{25}$$

This result follows from lemmas 2–5 given by *Ben-Tal et al.* [1994] and the note following the proof of lemma 5 there.

The statement of theorem 3 is as follows: Let $\{Q_i,\ i \in I\}$ be a partition of Q. Then

$$\min P_Q \geq \min_{i \in I} (\max D_{Q_i}) \geq \max D_Q \tag{26}$$

and if $d(\lambda)$ is continuous at $\lambda = 0$ and $A_{ij}(q)$ is a continuous function on Q for all i, j, then there exists a partition (Q_i^{ε}) with radius $\lambda > 0$, such that for any $\varepsilon > 0$ the duality gap $\min (P_Q) - \min_i (\max (D_{Q_i^{\varepsilon}}))$ is smaller than ε.

The proof of (26) is as follows. By weak duality for the pair (P_{Q_i}) and (D_{Q_i}):

$$\min (P_Q) = \min_{i \in I} \{\min (P_{Q_i})\} \geq \min_{i \in I} \{\max (D_{Q_i})\} \tag{27}$$

Hence the left inequality in (26) holds. Let $L(q, x, y)$ be the Lagrangian function of (1). Then

$$\max (D_{Q_i}) = \max_{y \geq 0} h_i(y) \tag{28}$$

where

$$h_i(y) = \min_{q \in Q_i, x} L(q, x, y)$$

However, for every y,

$$h_i(y) \geq \min_{q \in Q, x} L(q, x, y) \tag{29}$$

because $Q_i \subset Q$. Hence for every $i \in I$

$$\max (D_{Q_i}) \geq \max_{y \geq 0} \min_{q \in Q, x} L(q, x, y) = \max (D_Q) \tag{30}$$

which proves the right inequality in (26). For proof of the second part of theorem 3, see theorem 2 of *Ben-Tal et al.* [1994].

Algorithm

A branch and bound algorithm was developed to solve the design problem globally. In order to reduce computation time, we use the following heuristics:

Smallest reasonable box. While from a mathematical point of view, the set Q (or Δ) is unbounded, from practical considerations it is desirable to restrict it to be as small as possible. This will help to keep down the computation time for the global search. Thus we seek the smallest region (Δ^0) in which the vector q should be allowed to change around an initial q^0, such that we can still be sure that the optimal vector was not excluded. No analytic method was found for this, so the following heuristic rule is used: The bounds on the flow changes in each loop are taken as those minimum (negative) and maximum values which just bring the largest flow in a pipe belonging to the loop to reach zero. This prevents a situation wherein all flows in a loop are in the same direction, which, in the absence of a pump, is physically infeasible.

Need for outer iterations. The time to solve a problem is strongly dependent on the size of the box Δ^0 and the target gap. The overall efficiency is improved greatly by the following strategy. Start with a rather large Δ^0 and large target relative gap (say 10%). After solving the problem with these parameters, repeat the process (next outer iteration) after reducing the box and the target relative gap (increased precision). The best way to change the parameters must be found by trial and observation of the running times.

Starting point q^0. The starting point can be taken from previous trials or just by engineering considerations. Although the algorithm is not very sensitive to the starting point, it is clearly better to have it as close as possible to the optimal solution. For the chosen q^0, Δ^0 is calculated. By definition, $\delta = 0$ is the image of q^0 in the transformed space. Thus, as a rule, δ^0 is taken to be equal to 0.

The algorithm is detailed next in a symbolic form, using the following notation:

q^0	best known flow vector (or any other feasible flow);
ε	required relative gap (percent) between primal and dual optimal solutions;
m	number of parts in each partition;
cmax	length of branch, used as a criterion for starting a new local search;
f^*	optimal value of the primal problem after the latest local search;
δ^*	optimal solution of the primal problem after the latest local search;
$\Delta^0(q^0)$	bounded subset of the solution space, chosen such that the global optimal solution is included;
Bound	best global lower bound;
fmin	best primal objective value;
Solution	best optimal solution (q^*);
Gap	minimal relative gap (percent) found.

The conceptual algorithm proceeds as follows: The input consists of network data, q_0, ε, m, and cmax. Step 0 (initialization) is as follows:

$\ell = 0,\ c = 0,\ \delta = 0,\ \Delta = \Delta^0(q^0)$

$\Delta^{\ell} = \Delta$

local search on Δ^{ℓ} with δ as a starting point

$f = f^*,\ \delta = \delta^*.$

Step 1 (main) is as follows:

partition Δ^{ℓ} into m parts Δ_i^{ℓ}, $i = 1, \cdots, m$

solve dual D on each part Δ_i^{ℓ}

record dual optimal solutions g_i^{ℓ}, $i = 1, \cdots, m$

find $i^{\ell *} \in \underset{i}{\operatorname{argmin}}\ g_i^{\ell}$.

Step 2 (check branch or bound) is as follows:

if $100((f - g_{i^{\ell *}}^{\ell})/f) \leq \varepsilon$ then (bound) go to 3

else (branch) go to 4.

Step 3 (bound) is as follows:

$c = 0$

if $\ell = 0$ then (end) go to 5

else: $g_{i^{\ell-1}}^{\ell-1} = g_{i^{\ell *}}^{\ell}$

$\ell = \ell - 1$

find $i^{\ell *} \in \underset{i}{\operatorname{argmin}}\ g_i^{\ell}$

go to 2

endelse.

Table 1. Two-Loop Network: Solutions for Different Gap Sizes

Run	$f$min	B	Gap, %	D	α	Time, min	tloc, %
1	402,352.06	400,743.17	0.40	204	1.00	0.90	40.46
2	404,676.45	400,743.17	0.97	162	1.00	0.62	32.71
3	404,676.45	398,618.83	1.50	144	1.00	0.58	35.29
4	408,889.23	400,743.17	1.99	120	1.00	0.46	32.70
5	408,889.23	399,052.19	2.41	105	1.00	0.38	27.73
6	408,889.23	397,584.00	2.76	93	1.00	0.34	29.68
7	408,889.23	395,144.65	3.36	84	1.00	0.32	32.82
8	408,889.23	392,828.60	3.93	78	1.00	0.31	34.45
9	408,889.23	388,661.18	4.95	75	1.00	0.31	36.72

Here, tloc denotes the percent of time spent in local search out of the total running time.

Step 4 (branch) is as follows:

$c = c + 1$

$\ell = \ell + 1$

$\Delta^{\ell} = \Delta^{\ell}_{i^{\ell-1}*}$

if $c >$ cmax then:

$\hat{\delta}$ = center point of Δ^{ℓ}

local search with $\hat{\delta}$ as a starting point.

if $f^* < f$ then:

$f = f^*$, $\delta = \delta^*$

$c = 0$.

endif

go to 1.

Step 5 (end) is as follows:

Bound = $g^{\ell}_{i^{\ell}*}$

$f$min = f

Gap = 100 $((f - \text{Bound})/f)$

Solution = $q(\delta)$

print output.

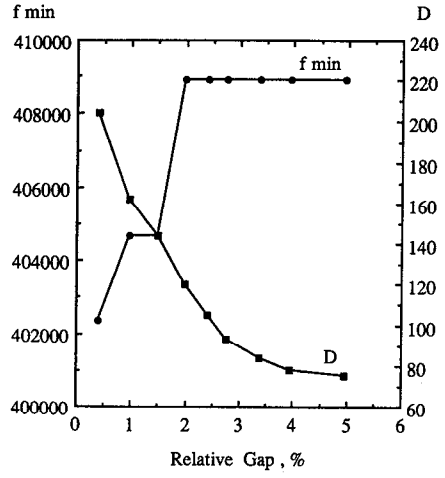

Figure 2. Optimal cost ($f$min) and number of times the dual problem is solved (D) versus the allowed relative gap (percent) for the two-loop network.

The main properties of the algorithm are as follows: (1) For all $\varepsilon > 0$, we have a "bound" condition after a finite number of "branch" steps. (2) If at some iteration $[(f - f\text{min})/f\text{min}] \times 100 >$ Gap, then at some later iteration we shall necessarily have $f^* < f$ after the local search. (3) From properties 1 and 2 it follows that the algorithm must stop. (4) The algorithm stops if and only if Gap $\leq \varepsilon$, except when the specified maximum number of iterations or the maximum running time has been exceeded.

Numerical Results

The method was tested on four example problems, which were taken from the literature. Due to space limitation, we have presented in this paper the method for networks of pipes only and, therefore, detailed results will be presented for only two examples. They are the two-loop network [*Alperovits and Shamir*, 1977; *Kessler and Shamir*, 1989] and the Hanoi network [*Fujiwara and Khang*, 1990].

Less detailed results are given for two additional networks, which include pumps and reservoirs, and more than a single loading condition. Extension of the methodology to cover these additions is detailed by *Eiger* [1991]. The two additional examples are the complex two-loop network [*Alperovits and Shamir*, 1977; *Kessler and Shamir*, 1989] and the real network [*Alperovits and Shamir*, 1977].

The complex two-loop network is similar to the two-loop network except that it is formulated and solved for two different loading conditions simultaneously. In addition, one pump and one reservoir are included. The real network has

Table 2. Optimal Solution of the Two-Loop Network

Link	Initial Flow q^0, m^3/h	Optimal Flow q^*, m^3/h	Diameter, inches	Length, m
1	1120.00	1120.00	18.00	1000.00
2	−30.00	373.34	10.00	761.98
			12.00	238.02
3	1050.50	646.66	16.00	1000.00
4	505.50	0.97	1.00	1000.00
5	425.00	525.69	14.00	371.14
			16.00	628.86
6	100.00	200.69	8.00	10.95
			10.00	989.05
7	−130.50	273.34	8.00	78.14
			10.00	921.86
8	−100.00	0.69	1.00	1000.00

1 inch equals 2.54 cm.

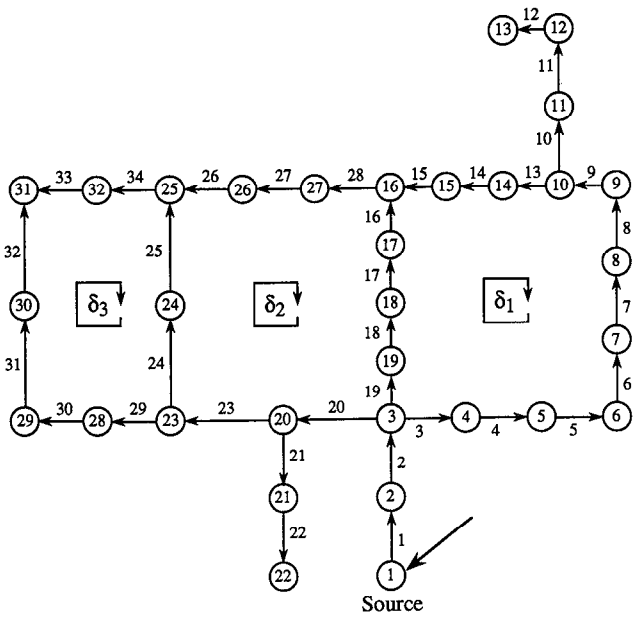

Figure 3. The Hanoi network [*Fujiwara and Khang*, 1990].

a much higher dimension, with two pumps and a reservoir. The overall results for these last two networks will be cited below.

All runs were made on a SUN Sparc 4 workstation in FORTRAN. For the solution of the linear problems we used a software package by *CPLEX Optimization, Inc.*, 1991], which was found to be most efficient. BT [*Schramm*, 1989] was used as the nonsmooth routine.

The results are presented in Tables 1–5, using the following additional notation:

gap% actual gap at the end of the run (or the outer iteration), equal to $(f\text{min} - B)/f\text{min}$ (100);
D total number of times the dual problem was solved, to the end of the run (or the outer iteration);
time total running time in minutes;
tloc% percent of time spent in local search out of the total running time.

Two-Loop Network

The network is shown in Figure 1. The data are those of *Alperovits and Shamir* [1977]. Because the network is small and simple it can be solved in a very short time and there is no need for outer iterations.

The problem dimensions are as follows: x, the maximal number of pipe segments, is 88; q, the number of links, is 8; u, the number of basic loops, is 2; and the number of constraints in the primal problem is 16.

Table 1 contains the results of a sequence of runs. In the first run the target relative gap was set at 0.5%, and was then increased by 0.5% in each run, to 5% in run 9. Each solution required only one outer iteration.

Figure 2 shows the optimal cost ($f$min) and the number of iterations (D) versus gap (percent). The behavior of D is typical, as was verified by other examples: a sharp drop which then tapers off. Note that solving dual problems consumes well over half of the running time. The solution appears in Table 2. Link numbers are according to Figure 1. Negative flow means that the flow is opposite to the positive direction. The same solution was found (with negligible differences) whenever the target gap was small enough (≤0.5%).

The best value of the objective function, 402,352.06, was found with the smallest gap. The bound is 400,743.17, 1609 lower, which is a difference of only 0.4%. The best value previously obtained has been 415,271, by *Fujiwara and Khang* [1990] (see also *Kessler and Shamir* [1989, p. 1476]).

Hanoi Network

The network is shown in Figure 3. The data are given by *Fujiwara and Khang* [1990]. The problem dimensions are as follows: x, the maximal number of pipe segments, is 204; q, the number of links, is 34; u, the number of basic loops, is 3; and the number of constraints in the primal problem is 68. Many runs were made, with the number of the outer iterations ranging from 1 to 16. A solution with target gap less than 0.5% was obtained in about 117 CPU minutes on a SUN 4 workstation.

Table 3 contains the results of a sequence of runs: each with just one outer iteration. In the first run the target relative gap was 0.5%, and it was then increased by 0.5% in each successive run. Figure 4 shows the optimal cost ($f$min) and the number of iterations (D) versus gap (percent). The solution appears in Table 4. The same solution was found (with negligible differences) whenever the target gap was small enough (≤0.5%).

The best value found was 6,026,660.26, only 0.4% higher than the bound. *Fujiwara and Khang* [1990] report a value of 5,562,000. We have found that their solution is not feasible, because not all minimum head requirements are met.

Complex Two-Loop Network

This is an expansion of the two-loop network [*Alperovits and Shamir*, 1977; *Kessler and Shamir*, 1989]. It has a

Table 3. Hanoi Network: Solution for Different Gap Sizes

Run	$f$min	B	Gap, %	D	α	Time, min	tloc, %
1	6,026,660.26	6,001,973.67	0.41	7854	1.00	117.02	11.09
2	6,026,661.75	5,965,154.14	1.02	3999	1.00	59.82	11.82
3	6,026,804.90	5,934,273.46	1.54	2697	1.00	41.10	12.85
4	6,031,028.66	5,908,630.69	2.03	1995	1.00	30.22	13.32
5	6,036,715.74	5,883,283.93	2.54	1608	1.00	24.70	14.13
6	6,027,830.17	5,846,480.53	3.01	1089	1.00	17.70	19.87
7	6,032,858.78	5,816,703.47	3.58	843	1.00	14.43	24.11
8	6,032,858.78	5,789,435.93	4.03	687	1.00	12.44	28.24
9	6,026,916.93	5,724,806.53	5.01	612	1.00	10.84	27.08

Table 4. Optimal Solution of the Hanoi Network

Link	Initial Flow q^0, m^3/h	Optimal Flow q^*, m^3/h	Diameter, inches	Length, m
1	19940.00	19940.00	40.00	100.00
2	19050.00	19050.00	40.00	1350.00
3	8705.00	7740.00	40.00	900.00
4	8575.00	7610.68	40.00	1150.00
5	7850.00	6885.68	40.00	1450.00
6	6845.00	5880.68	40.00	450.00
7	5495.00	4530.68	40.00	850.00
8	4945.00	3980.68	40.00	850.00
9	4420.00	3455.68	30.00	641.17
			40.00	158.83
10	2000.00	2000.00	30.00	950.00
11	1500.00	1500.00	24.00	1198.96
			30.00	1.04
12	940.00	940.00	24.00	3500.00
13	1895.00	930.68	16.00	800.00
14	1280.00	315.68	12.00	500.00
15	1000.00	35.68	12.00	550.00
16	1105.00	284.08	12.00	2730.00
17	1970.00	1149.08	16.00	634.13
			20.00	1115.87
18	3315.00	2494.00	24.00	800.00
19	3375.00	2554.08	24.00	400.00
20	6120.00	7905.25	40.00	2200.00
21	1415.00	1415.00	16.00	514.12
			20.00	985.88
22	485.00	485.00	12.00	500.00
23	3430.00	5215.25	40.00	2650.00
24	1320.00	3573.06	30.00	1230.00
25	500.00	2753.00	30.00	1300.00
26	525.00	−1260.25	20.00	850.00
27	1425.00	−360.25	12.00	7.28
			16.00	292.72
28	1795.00	9.75	12.00	750.00
29	1065.00	597.19	16.00	1500.00
30	775.00	307.19	12.00	2000.00
31	415.00	−52.81	12.00	1600.00
32	55.00	−412.81	16.00	150.00
33	50.00	517.81	16.00	632.92
			20.00	227.08
34	855.00	1322.81	24.00	950.00

1 inch equals 2.54 cm.

reservoir attached to node 7 via a pipe and a pump on pipe 1. It operates under two loading conditions, one representing daytime, when the demands are high and both reservoirs are emptying, and the other when there are no demands (nighttime) and the reservoir at node 1 provides through the pump the water required to fill the reservoir connected to node 7. All data are from *Alperovits and Shamir* [1977].

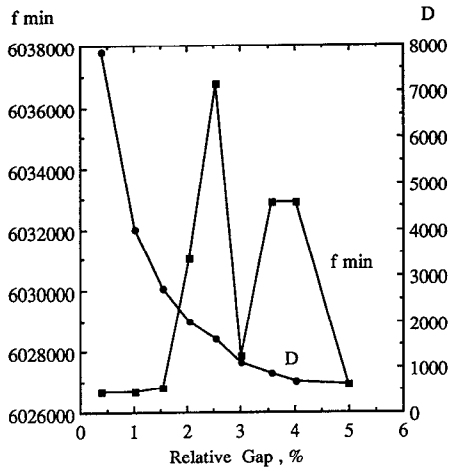

Figure 4. Optimal cost ($f$min) and number of times the dual problem is solved (D) versus the allowed relative gap (percent) for the Hanoi network.

Table 5 gives the values of the objective function obtained in 10 runs, in which the relative gap was changed between 10% and 0.3%. The optimal cost varies in these runs between 291,079.34 and 295,622.32 (a difference of only 1.53%) while the number of iterations increases substantially with the decrease in the gap required.

The dimensions of the problem are as follows: x, the maximal number of pipe segments, is 154; q, the number of links, is 22; u, the number of basic loops, is 4; and the number of constraints in the primal problem is 110. The best cost obtained by *Alperovits and Shamir* [1977, p. 895] was 299,851.

Real Network

The real network is taken from *Alperovits and Shamir* [1977]. It has 52 nodes, 68 pipes, two pumps, and one reservoir. It was solved for one loading condition (condition 1 of *Alperovits and Shamir* [1977]). The number of basic loops is 15, and there are 14 minimum head constraints. All data are as given by *Alperovits and Shamir* [1977], except that only one loading is considered. The dimensions of the problem are as follows: x, the maximal number of pipe segments, is 340; q, the number of links, is 68; u, the number of basic loops, is 15; and the number of constraints in the primal problem is 267. The best optimal solution was obtained with the following parameters: initial gap, 25%; factor for reduction of gap, 0.8; local criterion for branching, 8; and

Table 5. Complex Two-Loop Network: Solutions for Different Gap Sizes

Run	$f$min	B	Gap, %	D	α	Time, min	tloc, %
1	295,622.32	264,663.79	10.47	93	1.00	1.49	46.54
2	295,622.32	271,370.91	8.20	138	0.49	1.96	35.53
3	295,622.32	277,077.00	6.27	177	0.43	2.35	29.63
4	295,622.32	281,300.93	4.84	243	0.38	3.01	23.15
5	295,622.32	284,321.59	3.82	372	0.33	4.25	16.36
6	295,622.32	286,668.61	3.03	621	0.29	6.63	10.49
7	295,622.32	288,476.16	2.42	939	0.26	9.68	7.19
8	295,622.32	289,927.27	1.93	1476	0.23	14.68	4.74
9	291,079.34	289,927.27	0.40	1683	0.21	16.91	5.39
10	291,079.34	290,215.41	0.30	1764	0.09	17.62	5.17

stopping gap, 1.7%. This optimal cost is 721,320.57, obtained after 24,972 iterations, with a relative gap of 5.59%.

Summary

The method presented has performed well for the example problems, and we feel that it may have some potential to become a practical design tool. Of course, a lot of work should be done in order to adapt it to more realistic networks. Important realistic aspects which were not treated in this work are, for example, water quality constraints and reliability. Adding such aspects will, no doubt, complicate this already difficult problem.

Appendix: Nonsmooth Optimization

If one or more functions in an optimization problem are not continuously differentiable, we have to use a nonsmooth method to solve it. For most of the nonsmooth methods, the nondifferentiable function must be locally Lipschitz [*Zowe*, 1986]. The local Lipschitz continuity implies that the functions are differentiable almost everywhere. There are several differences between the nondifferentiable and differentiable cases.

1. Since the gradient of a nonsmooth function does not exist everywhere, we have to replace the gradient by the generalized gradient [*Clarke*, 1983]:

$$\partial f(x) = \text{conv}\,\{g \mid \exists \text{ a sequence } (x_i)_{i \in N} \ni \lim_{i \to \infty} x_i = x,\ f \text{ differentiable at } x_i,\ i \in N, \text{ and } \lim_{i \to \infty} \nabla F(x_i) = g\}, \tag{A1}$$

which is a well-defined, nonempty, convex, and compact subset of R^n. If f is convex, the generalized gradient coincides with the subdifferential:

$$\partial f(x) = \{g \in R^N \mid f(y) \geq f(x) + \langle g,\ y - x\rangle \ \forall\, y \in R^N\}. \tag{A2}$$

2. A necessary optimality condition for an unconstrained minimum of f is $O \in \partial f(x)$. This condition is also sufficient for convex f. For the constrained case, one can formulate rules similar to the Karush-Kuhn-Tucker rules.

3. Classical gradient methods can fail for nonsmooth problems. Bundle methods have been developed for solution of nonsmooth optimization problems. Their main features are as follows: (1) Subgradients g_i evaluated in preceding iterations at points y_i in the bundle are stored. (2) This information is weighed. For convex f one investigates the linearization errors,

$$\alpha(x_k,\ y_i) := \alpha_k^i := f(x_k) - [f(y_i) + \langle g_i,\ x_k - y_i\rangle] \qquad g_i \in \partial f(y_i) \tag{A3}$$

at the current iterate x_k, which gives a measure of the distance of g_i from $\partial f(x_k)$. (3) Search direction $d_k = -z_k$ is computed, where z_k is a convex combination of the subgradients in the bundle. The larger the weight α_i^k of a subgradient g_k, the less should be its influence on d_k. (4) Line search is performed along d_k to compute the next iterate and enrich the subgradient information; two different kinds of steps are possible.

Serious step

$$x_{k+1} := y_{k+1} := x_k + td_k,\ t > 0$$

if a sufficient decrease of the function value is achieved at x_{k+1}

Null step

$$x_{k+1} := y_{k+1}$$

if the decrease is too small, but $g_{k+1} \in \partial f(y_{k+1})$ in $y_{k+1} := x_{k+1} d_k,\ t > 0$, enriches the subgradient information significantly.

Bundle–Trust Region Methods

These methods combine some features of the trust region methods, from differential optimization, with the bundle concept: The line search of bundle methods is replaced by the trust region strategy.

For simplicity, assume that f is convex (for the nonconvex case, see *Schramm* [1989]). The stored subgradients are used to define the cutting plane model $\hat{f}_k$ of f around x_k. With the linearization errors $\alpha_i^k := f(x_k) - [f(y_i) + \langle g_i,\ x_k - y_i\rangle]$ we can write

$$\hat{f}_k(x_k + d) = f(x_k) + \max_{i \in J_k} \{-\alpha_i^k + \langle g_i,\ d\rangle\} \tag{A4}$$

where J_k is the set of indices of the subgradients in the bundle. To compute the next iterate, the trust region approach consists of minimizing the model $\hat{f}_k$ on a ball around the current iterate x_k, i.e., solving the trust region (TR) problem:

$$\text{minimize } \hat{f}_k(x_k + d)$$

subject to

$$\tfrac{1}{2}\|d\|^2 \leq \rho_k. \tag{A5}$$

with a given $\rho_k > 0$.

The solution d_k of TR gives the next trial point $x_k + d_k$. Then ρ_k is adapted for the next step. However, since TR is a problem with a quadratic constraint, we prefer to investigate

$$\text{minimize } \hat{f}_k(x_k + d) + (\tfrac{1}{2} t_k)\|d\|^2 \tag{A6}$$

which can be written as a quadratic programming problem in $(v,\ d)$ (the bundle-trust (BT) problem),

$$\text{minimize } v + (\tfrac{1}{2} t_k)\|d\|^2$$

subject to

$$v \geq \alpha_i^k + \langle g_i,\ d\rangle \qquad i \in J_k \tag{A7}$$

Then we modify t_k instead of ρ_k. To reach a serious or null step as in bundle methods, we substitute the line search of bundle methods by a trust region approach. During this so-called inner iteration we adapt t_k.

Bundle Trust Algorithm: The Convex Case

Let $\lambda_i^k,\ i \in J_k$, be a solution of the dual problem of BT (problem BT_d):

$$\text{minimize } \frac{1}{2}\left\|\sum_{i\in J_k}\lambda_i g_i\right\|^2 + (1/t_k)\sum_{i\in J_k}\lambda_i\alpha_i^k$$

subject to

$$\lambda_i \geq 0 \qquad i\in J_k,$$
$$\sum_{i\in J_k}\lambda_i = 1 \tag{A8}$$

Choose a starting point $x_1 \in R^n$ and parameters $T > 0$, $0 < m_1 < m_2 < 1$, $0 < m_3 < 1$, and $\varepsilon \geq 0$ and an upper bound $J_{\max} \geq 3$ for $|J_k|$. Step 0 is to compute $f(x_1)$, $g_1 \in \partial f(x_1)$ and put $y_1 = x_1$, $J_1 = \{1\}$ and $k = 1$. Step 1, the inner iteration, is to compute x_{k+1} and g_{k+1} such that the serious or null step criterion is satisfied, or determine that x_k is nearly ε optimal (in which case we stop). For step 2, if $|J_k| = J_{\max}$ then go to step 3; otherwise put $J = J_k$ and go to step 4. For step 3, reset by choosing $J \in J_k$ with $|J| \leq J_{\max} - 2$ and max $\{i | i \in J_k, a_i^k = 0\} \in J$. Compute

$$z_k = -(1/t_k)d_k = \sum_{i\in J_k}\lambda_i^k g_i$$

$$\alpha_k = \sum_{i\in J_k}\lambda_i^k\alpha_i^k$$

Introduce an additional index $\bar{k}$, and define

$$g_{\bar{k}} = z_k,\ \alpha_{\bar{k}}^k = \alpha_k,\ J + J\cup\{\bar{k}\}.$$

Step 4 consists of updating as follows: If the outcome of the inner iteration is a serious step, then set

$$\alpha_i^{k+1} = \alpha_i^k + f(x_{k+1}) - f(x_k) - \langle g_i, d_k\rangle$$
$$i\in J,\ \alpha_{k+1}^{k+1} = 0.$$

If the outcome of the inner iteration is a null step, then set

$$\alpha_i^{k+1} = \alpha_i^k \qquad i\in J,\ a_{k+1}^{k+1} = \alpha(x_k, y_{k+1})$$
$$J_{k+1} = J\cup\{k+1\}$$

and go to step 1.

Acknowledgments. The authors thank the anonymous reviewers for comments and suggestions which have helped to modify and simplify the presentation of the algorithm. The research of A. Ben-Tal was supported by a grant from the United States–Israel Binational Science Foundation (BSF).

References

Alperovits, E., and U. Shamir, Design of optimal water distribution system, *Water Resour. Res.*, *13*(6), 885–900, 1977.

Ben-Tal, A., G. Eiger, J. Outrata, and J. Zowe, A nondifferential approach to decomposable optimization problems with an application to the design of water distribution networks, in *Advances in Optimization, Lect. Notes Econ. Math. Sys.*, vol. 382, edited by W. Jetti and D. Pallaschke, pp. 197–216, Springer-Verlag, New York, 1992.

Ben-Tal, A., G. Eiger, and Z. Gershovitz, Global minimization by reducing the duality gap, *Math. Programming*, *63*, 193–212, 1994.

Charles Howard & Associates, Ltd., Water distribution network analysis, *SDP10*, Victoria, B. C., Canada, April 1986.

Clarke, F. H., *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, 1983.

CPLEX Optimization, Inc., The CPLEX linear optimizer and callable library, Incline Village, Nev., 1991.

Eiger, G., Optimal design of water distribution networks (in Hebrew), D.Sc. thesis, 130 pp., Technion–Israel Inst. of Technol., Haifa, Israel, 1991.

Fujiwara, O., and D. B. Khang, A two-phase decomposition method for optimal design of looped water distribution networks, *Water Resour. Res.*, *26*(4), 539–549, 1990.

Fujiwara, O., B. Jenchaimahakoon, and N. C. P. Edirisinghe, A modified linear programming gradient method for optimal design of looped water distribution networks, *Water Resour. Res.*, *23*(6), 977–982, 1987.

Goulter, I. C., B. M. Lussier, and D. R. Morgan, Implications of head loss path choice in the optimization of water distribution network, *Water Resour. Res.*, *22*(5), 819–822, 1986.

Karmeli, D., Y. Gadish, and S. Meyers, Design of optimal distribution network, *J. Pipeline Div. Am. Soc. Civ. Eng.*, *94*(PL1), 1–10, 1968.

Kessler, A., Optimal design of water supply networks using graph theory (in Hebrew), D.Sc. thesis, Technion–Israel Inst. of Technol., Haifa, Israel, 1988.

Kessler, A., and U. Shamir, Analysis of the linear programming gradient method for optimal design of water supply networks, *Water Resour. Res.*, *25*(7), 1469–1480, 1989.

Kiwiel, K. C., *Methods of Descent for Nondifferentiable Optimization*, Springer-Verlag, New York, 1985.

Morgan, G. R., and I. C. Goulter, Optimal urban water distribution design, *Water Resour. Res.*, *21*(5), 642–652, 1985.

Quindry, G. E., E. D. Brill, J. C. Liebman, and A. R. Robinson, Comment on "Design of optimal water distribution system" by E. Alperovits and U. Shamir, *Water Resour. Res.*, *15*(6), 1651–1654, 1979.

Quindry, G. E., E. D. Brill, and J. C. Liebman, Optimization of looped water distribution systems, *J. Environ. Eng. Am. Soc. Civ. Eng.*, *107*(EE4), 665–679, 1981.

Rowell, W. F., and J. W. Barnes, Obtaining layout of water distribution network, *Proc. Am. Soc. Civ. Eng.*, *108*(HY1), 137–148, 1982.

Saphir, Y. N., Optimization of water distribution systems (in Hebrew), M.Sc. thesis, Technion–Israel Inst. of Technol., Haifa, Israel, 1983.

Schramm, H., Eine Kombination von Bundle und Trust-Region Verfharen zur Losung nichtdifferenierbarer Optimierungsprobleme, *Bayreuther Mat. Schr.*, no. 30, 1989.

Shor, N. Z., *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, New York, 1985.

Walski, T. M., State of the art—Pipe network optimization, research report, Civil Eng., U.S. Army Eng. Waterways Exp. Stn., Vicksburg, Miss., 1985.

Walski, T. M., Battle of the network models: Epilogue, *J. Water Resour. Plann. Manage. Div. Am. Soc. Civ. Eng.*, *113*(2), 191–203, 1987.

Zowe, J., Nondifferentiable optimization—A motivation and a short introduction into the subgradient and the bundle concept, in Computational Mathematical Programming, edited by K. Schittkovsky, *NATO ASI Ser., Ser. F*, 321–356, 1986.

A. Ben-Tal and G. Eiger, Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Technion City, Haifa 32000, Israel.

U. Shamir, Faculty of Civil Engineering, Technion–Israel Institute of Technology, Technion City, Haifa 32000, Israel. (e-mail: cvrusus@technion.technion.ac.il)

(Received April 30, 1993; revised December 13, 1993; accepted February 9, 1994.)