Chapter 6

EFFECTIVE RELAXATION AND PARTITIONING SCHEMES FOR SOLVING WATER DISTRIBUTION NETWORK DESIGN PROBLEMS TO GLOBAL OPTIMALITY

6.1 Introduction

The design of water distribution systems has consistently received a great deal of attention because of its importance to society. Even so, many of the existing pipe networks in older urbanized areas function at severely reduced levels of efficiency and in some cases, are inadequate with respect to meeting the required pressure and flow demands. The investments associated with the installation, expansion and maintenance of water distribution systems are very high, and account for the largest proportion in municipal maintenance budgets. An important component in this process of designing a cost effective water distribution system or extending a preexisting network is to design the sizes of the various pipes that are capable of satisfying the flow demand, in addition to satisfying the minimum pressure head and hydraulic redundancy requirements. However, this least cost pipe design problem is a hard nonconvex optimization problem having a number of local optima, and has hence proven difficult to solve. A number of research efforts over the last two decades have focused on solving this problem, most of them generating improved local optimum solutions for several standard test problems from the literature, with no adequate lower bounds to evaluate the prescribed solutions. Three notable exceptions discussed below that are capable of providing solutions within a proven tolerance of a global optimum are the methods due to are Eiger et al. (1994), Sherali and

Smith (1995), and Sherali et al. (1998). Our development in the present chapter is a further enhancement of these three procedures.

The most general water distribution design problem is to build new networks, or to replace old and deteriorating sections of existing networks with new configurations having adequate capacity, while integrating network reliability and redundancy issues, network expansion and pipe sizing decisions, and multi-period economic analysis, all within a holistic framework. Traditionally, most of the work on the design of water distribution networks has focused on developing optimization procedures for the least cost pipe sizing problem. Sherali and Smith (1993) show how this important subproblem can be embedded in an integrated pipe-reliability-and-cost, and network optimization approach, that provides replacement recommendations along with the design and sizing of expanded and replaced sections of the network. Comprehensive reviews of work on designing water distribution systems are given in Lansey and Mays (1989). The techniques employed range from traditional nonlinear Hardy-Cross solver and Newton-Raphson methods, to more complicated hierarchical decomposition methods based on linear programming approximations. The use of stochastic optimization techniques such as simulated annealing and genetic algorithms have also been employed with considerable success.

The global optimization method addressed in this chapter uses a branch-and-bound approach based on a transformation of the pipe network optimization problem into the space of certain new design variables. The concave-convex nature of the nonlinear flow conservation constraints in the transformed space is used to develop tight linear relaxations via suitable polyhedral outer approximations and the generation of additional RLT constraints. This procedure provides an enhancement of Sherali et al.'s (1998) polyhedral outer approximation scheme, as well as provides an alternative RLT application for deriving a lower bounding problem than that proposed by Sherali

and Smith (1995). An upper bounding heuristic is also used to actively search for improved pipe designs. A partitioning scheme is designed to reduce the gap from optimality, inducing an infinite convergence to a global optimum. Four alternative branching strategies are developed and tested, and a particular maximal spanning tree-based branching variable choice procedure is proposed for implementation. These enhancements are shown to substantially reduce computational effort while determining proven global optimal solutions to standard test problems available in the literature. The algorithm is applied to two standard network design problems and one network expansion problem from the literature for which global optimal solutions that lie at least within 10^{-4} % of optimality are obtained. This results in an improved incumbent solution over that previously reported in the literature for each of these problems, and also provides a tight global lower bound for the New York network test problem, in particular, for the first time.

The remainder of this chapter is organized as follows. Section 6.2 presents the network optimization model, and Section 6.3 derives the proposed linear programming lower bounding problem and the upper bounding heuristic. The branch-and-bound algorithm is described in Section 6.4. Section 6.5 presents extensive computational results on test problems from the literature, as well as for the new Blacksburg network. Finally, Section 6.6 concludes the chapter with a discussion on possible algorithmic variants and further enhancements.

6.2. Model Formulation

Consider a distribution network $G'(N', A')$ comprised of a set of reservoirs or supply nodes and a set of consumption or demand nodes. Let these nodes be collectively identified by the index set $N' = \{1, 2, ..., n\}$, where the set of source nodes is denoted by $S' \subset N'$ and the set of demand nodes is denoted by $D' \subset N'$ such that $N' = S' \cup D'$. Associate with each node a quantity b_i that represents

the net water supply rate or demand rate corresponding to node i in the index set N'. We will assume that $b_i > 0$ for $i \in S$' and $b_i \leq 0$ for $i \in D$'. To ensure feasibility, we assume that the total supply rate is at least equal to the total demand rate.

In standard terminology, *sections* of pipes are defined to be short (6.20"-30") lengths of pipe that are used to physically construct a pipeline. A *segment* is defined to be a length of pipe having constant properties of diameter, roughness and annualized cost, perhaps composed of many sections. *Links* are defined as a collection of segments between two nodes, the lengths of which add up to the required length of the pipeline between the nodes.

For each pipe (new or existing) that connects certain designated node pairs i and j, where $i, j \in N$', $i < j$, we create a (notationally) directed arc $(i, j) \in A$'. For each $(i, j) \in A$', let L_{ij} denote the pipe length corresponding to a connection in the network between the nodes i and j. If we are working on the more general problem of expanding an existing network, the problem becomes one of designing new connections as well as constructing parallel pipe links that need to be installed between certain specified node pairs i and j, similar to the consideration of Loganathan et al. (1995). Let $A_f \subseteq A$' denote existing pipes that will remain fixed in design, and let $A_r \subseteq A$' denote existing pipes for which a parallel replacement connection needs to be designed. The flow between the node pairs i and j associated with the links in A_r will then be carried by the previously existing pipe link, as well as by the newly designed parallel pipe link. (When the existing pipe is being *replaced*, we simply treat this case similar to that of designing a new connection between the corresponding nodes, and accordingly absorb this within the arc set $A' - (A_f \cup A_r)$.) For the p^{th} such node pair $(i, j) \in A_r$, $p = 1, \ldots, |A_r|$, in order to avoid multi-arcs, we represent the situation by creating two new contiguous arcs by creating a dummy node $n+p$, along with arcs $(i, n + p)$ and $(n + p, j)$. We set the demand value b_{n+p} for the dummy node to be zero. The lengths of the arcs $(i, n + p)$ and $(n + p, j)$ are each set

equal to $L_{ij}/2$, and the diameters of the segments in these pipes are fixed at their existing values. It is assumed that the "installation" cost for such pre-existing pipes is zero. Under this scheme, the link to be *newly designed* is represented by the (original) arc (i, j) that runs parallel to the arcs $(i, n + p)$ and $(n + p, j)$. Such a procedure adds $|A_r|$ nodes and $2|A_r|$ arcs to the network G', effectively creating an expanded network $G(N, A)$, where $N = N' \cup \{n + p: p = 1, \ldots, |A_r|\}$, $A = A' \cup \{(i, n + p), (n + p, j): (i, j) \in A_r\}$. The revised set of demand nodes D associated with this network G are suitably updated to include the newly created (zero-demand) nodes, so that $D = D' \cup \{n + p: p = 1, \ldots, |A_r|\}$. Since no new source nodes are added, we have $S = S'$. For notational convenience, we will denote the set of arcs in A that are to be newly designed as P, while $A - P$ will represent the existing links in the network. We will assume that each link that needs to be designed is constructed from segments of lengths having standard available diameters, chosen from the set $\{d_k, k = 1, ..., K\}$. Also, let us denote by c_k the cost per unit length for a pipe of diameter d_k.

Associated with each link connecting node pairs $(i, j) \in A$ is the decision variable q_{ij} that represents the flow rate (m^3/hr). Note that this variable may be nonnegative or negative, thus permitting flow in either direction. A positive flow value means that flow is along the specified conventional direction of the arc.

Our next set of decision variables relate to the lengths of segments having different standard diameters chosen from the set of available diameters, that comprise each link of the network. Let x_{ijk} denote the length of segment of diameter k in the link $(i, j) \in A$, and let x_{ij} be the vector having components $(x_{ijk}, k = 1, .., K)$. We assume that the variable x_{ijk} is fixed at a value $\tilde{x}_{ijk}$ for all arcs $(i, j) \in A\text{-}P$, $\forall k = 1, \ldots, K$. Now, let us consider the energy heads at the various nodes $i \in N$ in the network. For each node $i \in S$ let E_i denote the ground elevation of node $i \in N$ and let H_i denote the

established head above E_i. Additionally, for the source nodes $i \in S$, let F_i denote the fixed maximum available energy head, and suppose that there is an opportunity to further raise this head by an amount H_{si} at an annualized cost $c_{si} > 0$ per unit energy head, as suggested by Rowell and Barnes (1982). Correspondingly, for each demand node $i \in D$, suppose that there is the requirement that at a flow equilibrium, the established head ($H_i + E_i$) at this node lies in the interval $[H_{iL}, H_{iU}]$ where $H_{iL} < H_{iU}$. For any dummy node $n+p$ that is formed by the conjunction of the arcs $(i, n+p)$ and $(n+p, j)$ $\forall (i, j) \in A_f$, the node elevation E_{n+p} is set equal to $(E_i + E_j)/2$, and the pressure bounding interval for $(H_{n+p} + E_{n+p})$ is set equal to $[\min(H_{iL} + E_i, H_{jL} + E_j), \max(H_{iU} + E_i, H_{jU} + E_j)]$.

The pressure loss (or head loss) in a pipe due to friction, given by $[(H_i+E_i)-(H_j+E_j)]$ for a link (i, j) depends on the pipe characteristics such as diameter, roughness, length and the water flow rate through the pipe. The frictional head loss in a segment of pipe under smooth flow conditions is approximately described by the empirical Hazen-William equation as follows (see Alperovits and Shamir (1977) and Walski (1984)), where the sign depends on the direction of flow. That is, for a link (i, j), the frictional loss is taken to be positive if computed along the direction of flow and negative if opposite to it.

$$\Phi (q, C_{HW}, d, x) = (1.52)\, 10^4 \,\text{sign}(q)\, |q/C_{HW}|^{1.852} d^{-4.87}\, x \qquad (6.2.1)$$

where

Φ = pressure head loss assuming smooth flow conditions in a given pipe segment

q = water flow rate in the pipe (m^3/hr)

C_{HW} = Hazen Williams coefficient based on roughness and diameter

d = pipe diameter (in centimeters)

x = pipe length (in meters).

Remark 1: Note that if pipe flow data is recorded in other frequently used units such as gallons per minute (gpm) or cubic-feet per second (cfs), we would either need to appropriately convert the dimensions to SI units or adjust the coefficients in the head loss equation (6.2.1). For example, Fujiwara and Khang (1990) analyze the New York network assuming flow to be in cfs, pipe length in feet, and pipe diameter in inches. They use a discharge coefficient value of 1.85 (instead of 1.852, which used here) and a head loss coefficient value of 851500 (instead of 1.52×10^4, which is used here). To avoid conversion errors, and for comparative purposes, it might be preferable to adjust the coefficients in (6.2.1) according to the units used to present pipe data, rather than convert the latter to appropriate SI units and then adopt Equation (6.2.1).

For our model, the head loss in a pipe that has several potential segments of varying diameter and roughness is computed as follows

$$\Phi_{ij}(q_{ij}, x_{ij}) = \sum_{k=1}^{K} \Phi(q_{ij}, C_{HW(ijk)}, d_k, x_{ijk}), \text{ where } x_{ij} \equiv (x_{ijk}, k = 1,..., K). \qquad (6.2.2)$$

The flow values q_{ij} associated with each link are assumed to lie between some analytically determined minimum and maximum bounds $q\min_{ij}$ and $q\max_{ij}$, that may appropriately be of either sign. We define the hyperrectangle restricting the flows q as $\Omega = \{q: q\min \le q \le q\max\}$, where the notation $q$min and $q$max with the subscripts dropped denotes the corresponding vectors of lower and upper bounds. Sherali et al. (1998) discuss procedures for determining these bounds on the flows from the network configuration using logical arguments (without making any *a priori* assumption on the nature of the optimal flow distribution).

The network optimization problem NOP, restricted on Ω, can now be formulated as follows.

NOP (Ω):

Minimize $$\sum_{ij \in P} \sum_{k=1}^{K} c_k x_{ijk} + \sum_{i \in S} c_{si} H_{si} \qquad (6.2.3a)$$

subject to

$$\Phi_{ij}(q_{ij}, x_{ij}) = (H_i + E_i) - (H_j + E_j) \qquad \forall (i, j) \in A \qquad (6.2.3b)$$

$$\sum_{k=1}^{K} x_{ijk} = L_{ij} \qquad \forall \in (i, j) \in P \qquad (6.2.3c)$$

$$\sum_{j \in FS(i)} q_{ij} - \sum_{j \in RS(i)} q_{ji} = b_i \quad \forall i \in D \qquad (6.2.3d)$$

$$\sum_{j \in FS(i)} q_{ij} - \sum_{j \in RS(i)} q_{ji} \leq b_i \quad \forall i \in S \qquad (6.2.3e)$$

$$q\min_{ij} \leq q_{ij} \leq q\max_{ij} \qquad \forall (i, j) \in A \qquad (6.2.3f)$$

$$H_i + E_i \leq F_i + H_{si} \qquad \forall i \in S \qquad (6.2.3g)$$

$$H_{iL} \leq H_i + E_i \leq H_{iU} \qquad \forall i \in D \qquad (6.2.3h)$$

$$H_{si} \geq 0 \qquad \forall i \in S \qquad (6.2.3i)$$

$$x_{ijk} \geq 0 \qquad \forall (i, j) \in P, k = 1, \ldots, K$$

$$x_{ijk} = \tilde{x}_{ijk} \qquad \forall (i, j) \in A - P, k = 1, \ldots, K \qquad (6.2.3j)$$

The objective function, Equation (6.2.3a), denotes the total cost of the pipes and the cost of the additional head generated at each source node. Constraints (6.2.3b) are the conservation of energy equations, and along with Constraints (6.2.3h), ensure that the hydraulic energy loss over each chain in the network is such that the minimum head requirements (H_{iL}) are met for each demand node. It may be noted that Constraints (6.2.3b) implicitly enforce that the hydraulic energy loss in each loop in the network is zero. The link length constraints are represented by Equation (6.2.3c). Equations

(6.2.3d) and (6.2.3e) enforce conservation of flow at all nodes, and Equation (6.2.3f) bounds the flow value in each link to lie in a specified valid or implied interval. These bounds that define Ω will be suitably modified during the course of the algorithm for solving Problem NOP. Constraints (6.2.3g) and (6.2.3h) represent restrictions on the maximum variable head at each source node, and the head requirements at each demand node, respectively. Finally, Constraints (6.2.3i) and (6.2.3j) enforce logical nonnegativity restrictions. Note that the variables x_{ijk} are assumed to be fixed for arcs $(i, j) \in A - P$ since the pipe diameters of segments in these arcs are pre-specified. Note that by enforcing that pre-existing pipes be either replaced or retained without any parallel connections, Problem NOP reduces to the network design problem of Sherali et al. (1998).

Our principal set of decision variables are the lengths x_{ijk} of the different segments comprising each link $(i, j) \in P$, and the additional head H_{si} to be developed at each source node $i \in S$. The resulting heads H_i at each node $i \in N$ (above the elevation E_i of the node) and the flows q_{ij} in the links $(i, j) \in A$ are also problem variables that happen to be governed by the foregoing design variables.

6.3. Lower and Upper Bounding Problems

The frictional head loss expression in the Constraints (6.2.3b) cause NOP(Ω) to become nonlinear and nonconvex. We take advantage of the monotone nature of these constraints to transform the problem NOP(Ω) into a set of newly defined variables, and accordingly, develop suitable relaxations for the flow conservation constraints that turn out to be nonlinear in the projected space of these new decision variables.

Since the relations derived subsequently hold for each link, the subscripts defining the links will be dropped for convenience. Equation (6.2.2) which appears in Constraints (6.2.3b) can be

written as follows using Equation (6.2.1), for any link having a flow q and a length segment vector $x = (x_k, k = 1, \ldots, K)$.

$$\Phi(q, x) = \sum_{k=1}^{K} \text{sign}(q) \mid q \mid^{1.852} \ (1.52)10^4 (C_{HW(k)})^{-1.852} d_k^{-4.87} x_k . \tag{6.3.1}$$

Denoting

$$v(q) \equiv \text{sign}(q) \mid q \mid^{1.852}, \tag{6.3.2}$$

Equation (6.3.1) can be rewritten as follows,

$$\Phi(q, x) = \sum_{k=1}^{K} v(q) \alpha_k x_k , \tag{6.3.3}$$

where, $\alpha_k \equiv (1.52)\ 10^4\ (C_{HW(k)})^{-1.852}\ d_k^{-4.87}$ (6.3.4)

By the monotonicity of $v(q)$ we can represent its value for any q as some convex combination of its minimum and maximum values $v(q\text{min})$ and $v(q\text{max})$, henceforth denoted as $v\text{min}$ and $v\text{max}$, respectively.

$$v(q) = \lambda\ v\text{min} + (1-\lambda)\ v\text{max}, \quad \text{for some } 0 \le \lambda \le 1. \tag{6.3.5}$$

Using the representation (6.3.5) in Equation (6.3.4) and rearranging terms, we get,

$$\Phi(q, x) = \sum_{k=1}^{K} v\min\ \alpha_k (\lambda x_k) + \sum_{k=1}^{K} v\max\ \alpha_k (1-\lambda) x_k . \tag{6.3.6}$$

We now define our new decision variables as

$$x^1_k = \lambda x_k \text{ and } x^2_k = (1-\lambda) x_k , \tag{6.3.7}$$

so that,

$$x_k = x^1_k + x^2_k. \tag{6.3.8}$$

Note that for the existing pipes, x_k is fixed at a value $\tilde{x}_k$ (some possibly zero) for all k. Equation (6.3.6) can now be rewritten in terms of the new decision variables as

$$\Phi(q,x) = \sum_{k=1}^{K} (v\min)\,\alpha_k x_k^1 + \sum_{k=1}^{K} (v\max)\,\alpha_k x_k^2 . \tag{6.3.9}$$

Note that in the space of the new decision variables, x^1, x^2 and λ, we have linearized the energy conservation constraints by substituting (6.3.9) on the left-hand side of (6.2.3b), but at the expense of introducing nonlinearity elsewhere in the problem. Specifically, this nonlinearlity arises in the two sets of relationships. First it occurs in the nonlinear representation (6.3.7) along with the linear relationship (6.3.8). Second, the flow q (for each generic link) is now given via (6.3.2) and (6.3.5) by the following function $q(\lambda)$,

$$q(\lambda) = \text{sign}[\lambda v\min+(1-\lambda)\, v\max]\,\big|\,\lambda\, v\min+(1-\lambda)\, v\max\,\big|^{1/1.852}. \tag{6.3.10}$$

Using the foregoing transformations, we therefore obtain an alternative equivalent representation of the Problem NOP(Ω). We will now construct relaxations for the nonlinear relationships (6.3.7) and (6.3.10) in order to derive lower bounding linear programs.

As far as relating x^1 and x^2 to λ is concerned, we use an aggregate relationship based on (6.2.3c) and (6.3.7) by including the constraints

$$\sum_{k=1}^{K} x_{ijk}^1 = \lambda_{ij} L_{ij} \qquad \forall\,(i,j) \in P. \tag{6.3.11}$$

Then, in the proposed relaxation, we substitute out x_k from the problem using (6.3.8) and omit the relationship (6.3.7) (but include its aggregate representation (6.3.11). Note that the symmetric relationship $\sum_{k=1}^{K} x_{ijk}^2 = (1-\lambda_{ij})L_{ij} \forall (i,j) \in P$ is implied by (6.3.11) and (6.2.3c) after substituting (6.3.8) into (6.2.3c).

Next, for the nonlinear relationship (6.3.10), we replace this by several implied linear supporting inequalities to construct a polyhedral relaxation. First of all, by examining q as a function

of λ over its positive and negative ranges as given by (6.3.10), it can be readily verified that the derivative (or slope) of $q(\lambda)$, denoted by $q'(\lambda)$, is given by

$$q'(\lambda) = \frac{(v\min - v\max)}{1.852} \mid \lambda v\min + (1-\lambda)v\max \mid^{\frac{-0.852}{1.852}}, \text{ for } \lambda \in [0,1], \lambda v\min + (1-\lambda)v\max \neq 0.$$

Note that for λ such that $\lambda v\min + (1\text{-}\lambda)v\max = 0$, $q'(\lambda)$ is undefined (negative infinity). Moreover, the function $q(\lambda)$ takes different forms depending on the bounds on the flow values in a link. We will separately treat the following three situations:

(1) $q\max > 0$ and $q\min < 0$ (concave-convex function)

(6.2) $q\max > 0$ and $q\min \geq 0$ (concave function)

(6.3) $q\max \leq 0$ and $q\min < 0$ (convex function)

(1) Concave-Convex Function ($q\max > 0$ and $q\min < 0$):

In its most general form, the function is clearly concave-convex as depicted in Figure 6.1. Let $\bar{\lambda}$ be such that if it exists, the tangential support to $q(\lambda)$ at $\lambda = \bar{\lambda}$ passes through the coordinate (1, $q\min$) in the (λ, q) space (see Figure 6.2). Similarly, let $\tilde{\lambda}$ be such that if it exists, the tangential support to $q(\lambda)$ at $\lambda = \tilde{\lambda}$ passes through the coordinate (0, $q\max$) in the (λ, q) space. These two situations are represented by Equations (6.3.12) and (6.3.13), respectively, as given below, which have a solution if an only if the corresponding entities $\bar{\lambda}$ and $\tilde{\lambda}$ exist.

$$q\min - q(\bar{\lambda}) - (1-\bar{\lambda})q'(\bar{\lambda}) = 0 \qquad \text{for } 0 < \bar{\lambda} < 1, \qquad (6.3.12)$$

$$q\max - q(\tilde{\lambda}) + \tilde{\lambda}q'(\tilde{\lambda}) = 0 \qquad \text{for } 0 < \tilde{\lambda} < 1. \qquad (6.3.13)$$

We now consider three subcases.

Case (a):$\bar{\lambda}$ and $\tilde{\lambda}$ exist. As a relaxed representation of $q(\lambda)$ in this case, we will use supporting hyperplanes as shown in Figure 6.2. The points $\bar{\lambda}$ and $\tilde{\lambda}$ in Figure 6.2 can be obtained by using a bisection search procedure for determining solutions to Equations (6.3.12) and (6.3.13), respectively. Then, in the relaxation of the transformed problem NOP(Ω), instead of using (6.3.10), we employ the set of inequalities given by Equations (6.3.14) and (6.3.15) below.

$$q \le q(\hat{\lambda}) + (\lambda - \hat{\lambda})q'(\hat{\lambda}) \text{ for } \hat{\lambda} = 0, \frac{\bar{\lambda}}{2}, \text{ and } \bar{\lambda}. \tag{6.3.14}$$

$$q \ge q(\hat{\lambda}) + (\lambda - \hat{\lambda})q'(\hat{\lambda}) \text{ for } \hat{\lambda} = \tilde{\lambda}, \frac{(1+\tilde{\lambda})}{2}, \text{ and } 1. \tag{6.3.15}$$

Case (b): $\bar{\lambda}$ exists, but $\tilde{\lambda}$ does not. In this case, the function is concave, except for a small convex portion ($q \le 0$) as depicted in Figure 6.3 such that $\bar{\lambda}$ can be obtained as before using (6.3.12), but $\tilde{\lambda}$ cannot, i.e., a solution $\tilde{\lambda}$ does not exist in (6.3.13). As a relaxed representation of $q(\lambda)$ in this case, we will use supporting hyperplanes as shown in Figure 6.4. The inequalities that represent these supporting hyperplanes are specified by (6.3.14) and (6.3.16) stated below. Note that (6.3.16) is valid for the convex segment since $\tilde{\lambda}$ does not exist, and that for $q(\lambda) \ge 0$, the function is concave, and therefore this function lies above its corresponding convex envelope, which in turn lies above the linear function of λ given by (6.3.16) over [0, 1].

$$q \ge \lambda\, q\text{min} + (1\text{-}\lambda)\, q\text{max} \tag{6.3.16}$$

Case (c): $\tilde{\lambda}$ exists, but $\bar{\lambda}$ does not. This situation is symmetric to the previous case, except that. This situation is symmetrical to the previous case, except that the function is almost convex, except for a small concave portion ($q \ge 0$), such that $\tilde{\lambda}$ exists, but $\bar{\lambda}$ cannot be obtained using (6.3.12). The supporting hyperplane inequalities used in this case are given by (6.3.15) and (6.3.17), where the validity of the latter follows arguments similar to that used for (6.3.16).

$$q \leq \lambda\, q\min + (1-\lambda)\, q\max \qquad (6.3.17)$$

(2) Concave Function ($q\max > 0$ and $q\min \geq 0$):

Here, we utilize the fact that the function lies above its convex envelope, and that it lies below its supporting hyperplanes, to construct the required bounding functions (6.3.16) and (6.3.18) below, respectively. Note that in (6.3.18), as illustrated in Figure 6.5, we have used five points equal to 0, 0.25, 0.5, 0.8 and 0.9 to determine the tangential supports based on the shape of the function. It may be noted that at $\lambda = 1$, if $q\min = 0$, the tangent has infinite slope. Hence for simplicity and numerical stability, we take $\lambda = 0.9$ at the upper end in general for the concave case. The proposed relaxation then uses (6.3.16) and (6.3.18) in lieu of (6.3.10) within the transformed problem NOP(Ω).

$$q \leq q(\hat{\lambda}) + (\lambda - \hat{\lambda})q'(\hat{\lambda}) \text{ for } \hat{\lambda} = 0, 0.25, 0.5, 0.8, \text{ and } 0.9. \qquad (6.3.18)$$

(3) Convex Function ($q\max \leq 0$ and $q\min < 0$):

Similar to the foregoing case, using the properties of convex functions, we utilize inequalities (6.3.17) and (6.3.19) given below as a relaxation for $q(\lambda)$.

$$q \geq q(\hat{\lambda}) + (\lambda - \hat{\lambda})q'(\hat{\lambda}) \text{ for } \hat{\lambda} = 0.1, 0.2, 0.5, 0.75, \text{ and } 1. \qquad (6.3.19)$$

Note that, except for cases 1(b) and 1(c) of concave-convex functions, six facetial supports are generated, while only 4 facetial supports are generated for the former cases. While we have chosen the points of tangency of these supports by a process of trial and error, it may be a good idea to generate additional supporting hyperplanes in order to increase the accuracy of the polyhedral outer approximation where ever possible, attempting to select points that minimize the area created by the intersection of the corresponding halfspaces.

The above analysis leads to the following lower bounding problem in the space of the variables x^1, x^2, λ , q, H_S and H.

LB(Ω):

Minimize $$\sum_{ij\in P}\sum_{k=1}^{K} c_k (x_{ijk}^1 + x_{ijk}^2) + \sum_{i\in S} c_{si} H_{si} \qquad (6.3.20a)$$

subject to

$$\sum_{k=1}^{K} (v\min_{ij})\alpha_{ijk} x_{ijk}^1 + \sum_{k=1}^{K} (v\max_{ij})\alpha_{ijk} x_{ijk}^2 = (H_i + E_i) - (H_j + E_j)\ \forall (i,j)\in A \qquad (6.3.20b)$$

$$\sum_{k=1}^{K} x_{ijk}^1 + \sum_{k=1}^{K} x_{ijk}^2 = L_{ij}\ \ \forall (i,j)\in P \qquad (6.3.20c)$$

$$\sum_{k=1}^{K} x_{ijk}^1 = \lambda_{ij} L_{ij}\ \forall (i,j)\in P$$

$$x_{ijk}^1 = \lambda_{ij}\tilde{x}_{ijk}\ \forall (i,j)\in A-P, k=1,...,K$$

$$x_{ijk}^2 = (1-\lambda_{ij})\tilde{x}_{ijk}\ \forall (i,j)\in A-P, k=1,...,K \qquad (6.3.20d)$$

$$\sum_{j\in FS(i)} q_{ij} - \sum_{j\in RS(i)} q_{ji} = b_i\ \ \forall i\in D \qquad (6.3.20e)$$

$$\sum_{j\in FS(i)} q_{ij} - \sum_{j\in RS(i)} q_{ji} \le b_i\ \ \forall i\in S \qquad (6.3.20f)$$

$$q_{ij} \le \lambda_{ij} q\min_{ij} + (1-\lambda_{ij}) q\max_{ij}\ \forall (i,j)\ni q\min_{ij} < 0, q\max_{ij} > 0, \bar{\lambda}_{ij} \text{ does not exist} \qquad (6.3.20g)$$

$$q_{ij} \le q_{ij}(\hat{\lambda}_{ij}) + (\lambda_{ij} - \hat{\lambda}_{ij}) q_{ij}^{/}(\hat{\lambda}_{ij}) \text{ for } \hat{\lambda}_{ij} = 0, \frac{\bar{\lambda}_{ij}}{2}, \bar{\lambda}_{ij}\ \forall (i,j)\ni q\min_{ij} < 0, q\max_{ij} > 0, \bar{\lambda}_{ij} \text{ exists} \qquad (6.3.20h)$$

$$q_{ij} \ge q_{ij}(\hat{\lambda}_{ij}) + (\lambda_{ij} - \hat{\lambda}_{ij}) q_{ij}^{/}(\hat{\lambda}_{ij}) \text{ for } \hat{\lambda} = \tilde{\lambda}_{ij}, (\frac{1+\tilde{\lambda}_{ij}}{2}), 1\ \forall (i,j)\ni q\min_{ij} < 0, q\max_{ij} > 0, \tilde{\lambda}_{ij} \text{ exists} \qquad (6.3.20i)$$

$$q_{ij} \ge \lambda_{ij} q\min_{ij} + (1-\lambda_{ij}) q\max_{ij}\ \forall (i,j)\ni q\min_{ij} < 0, q\max_{ij} > 0, \tilde{\lambda}_{ij} \text{ does not exist} \qquad (6.3.20j)$$

$$q_{ij} \geq \lambda_{ij} q\min_{ij} + (1-\lambda_{ij}) q_{ij} \;\forall (i,j) \ni q\min_{ij} \geq 0 \qquad (6.3.20k)$$

$$q_{ij} \leq \lambda_{ij} q\min_{ij} + (1-\lambda_{ij}) q\max_{ij} \;\forall (i,j) \ni q\max_{ij} \leq 0 \qquad (6.3.20l)$$

$$q_{ij} \leq q_{ij}(\hat{\lambda}_{ij}) + (\lambda_{ij} - \hat{\lambda}_{ij}) q'_{ij}(\hat{\lambda}_{ij}) \text{ for } \hat{\lambda}_{ij} = 0, 0.25, 0.5, 0.8, 0.9 \;\forall (i,j) \ni q\min_{ij} \geq 0 \qquad (6.3.20m)$$

$$q_{ij} \geq q_{ij}(\hat{\lambda}_{ij}) + (\lambda_{ij} - \hat{\lambda}_{ij}) q'_{ij}(\hat{\lambda}_{ij}) \text{ for } \hat{\lambda}_{ij} = 0.1, 0.2, 0.5, 0.75, 1 \;\forall (i,j) \ni q\max_{ij} \leq 0 \qquad (6.3.20n)$$

$$H_i + E_i \leq F_i + H_{si} \qquad \forall\, i \in S \qquad (6.3.21o)$$

$$H_{iL} \leq H_i + E_i \leq H_{iU} \qquad \forall\, i \in D \qquad (6.3.21p)$$

$$H_{Si} \geq 0 \qquad \forall\, i \in S \qquad (6.3.20q)$$

$$q\min_{ij} \leq q_{ij} \leq q\max_{ij} \qquad \forall\, (i,j) \in A \qquad (6.3.20r)$$

$$x^1_{ijk},\, x^2_{ijk} \geq 0 \qquad \forall\, (i,j) \in P,\, k = 1, \ldots, K \qquad (6.3.20s)$$

$$0 \leq \lambda_{ij} \leq 1 \qquad \forall\, (i,j) \in A. \qquad (6.3.20t)$$

For computing upper bounds on the least cost pipe sizing problem NOP(Ω), we employ the following heuristic based on the flow conserving solution q produced by the lower bounding problem LB(Ω) for any given Ω. Observe that if we fix the flow as obtained for this problem within the Problem NOP(Ω), the resulting formulation, denoted ILP(Ω), is a linear programming problem. An upper bounding completion to this fixed flow, if it exists, can therefore be obtained by solving ILP(Ω).

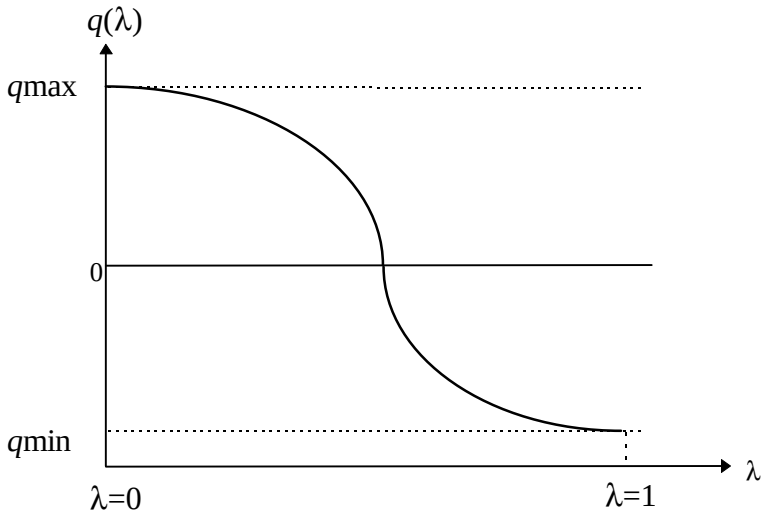

Figure 6.1. Plot of $q(\lambda)$ for Concave-Convex Case (a).

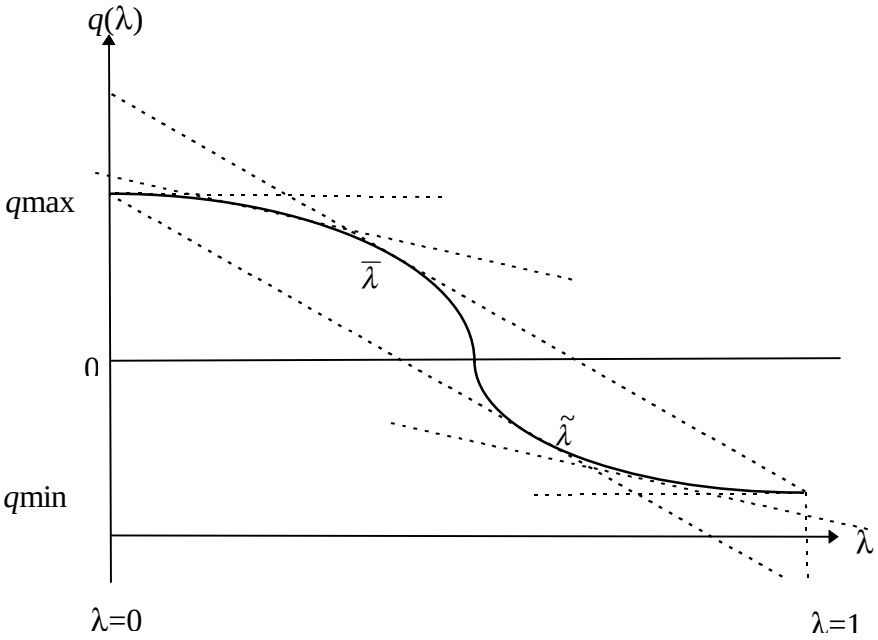

Figure 6.2. Relaxation of $q(\lambda)$ for Concave-Convex Case (a).

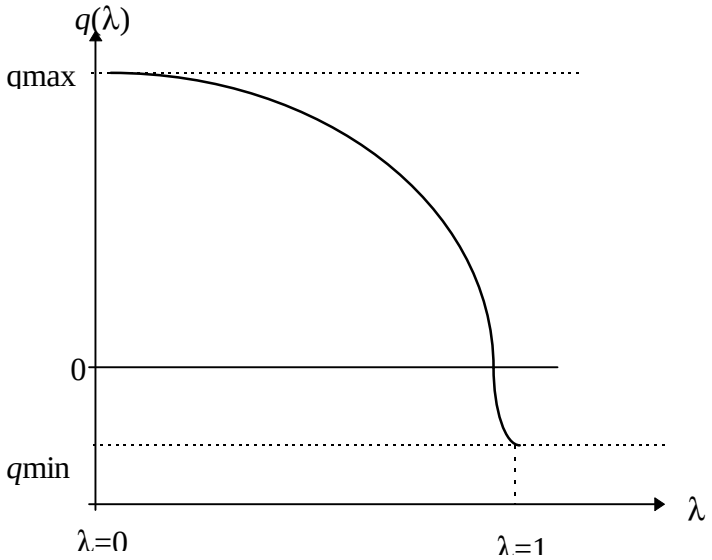

Figure 6.3. Plot of $q(\lambda)$ for Concave-Convex Case (b).

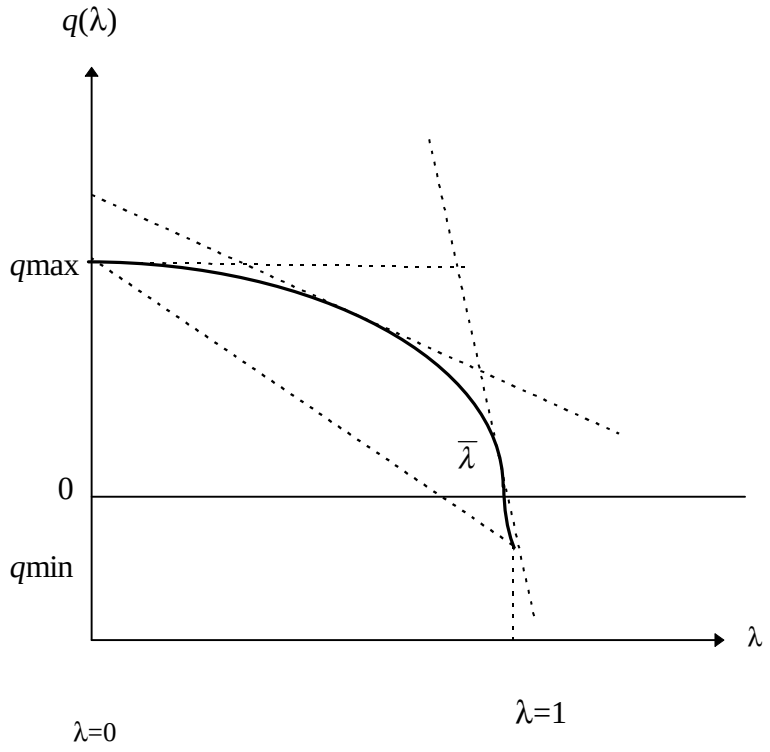

Figure 6.4. Relaxation of $q(\lambda)$ for Concave-Convex Case (b).

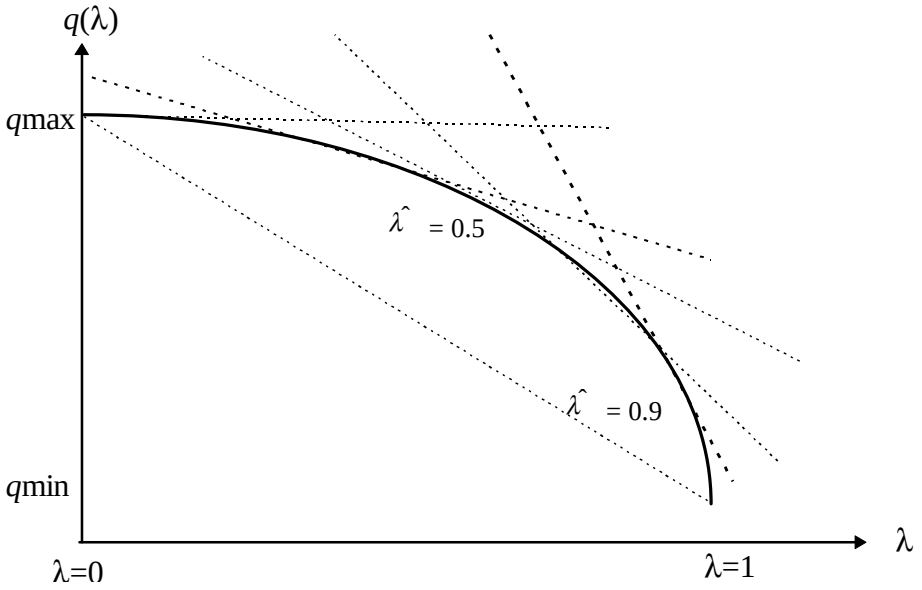

Figure 6.5. Relaxation of $q(\lambda)$ for the Concave Function.

6.4. A Branch-and-Bound Algorithm

We embed LB(Ω) and ILP(Ω) in a branch-and-bound procedure to solve NOP(Ω) globally to any specified percentage tolerance of optimality. Each branch-and-bound node principally differs in the specification of the hyperrectangle Ω. The hyperrectangle associated with node t of the branch-and-bound tree at the main iteration or stage S of the procedure is denoted below by $\Omega^{S,\,t} = \{q\colon q\text{min}^{S,\,t} \le q \le q\text{max}^{S,\,t}\}$. In our implementation of the branch-and-bound procedure, we successively partition the hyperrectangle defined by the initial bounds $\Omega^{1,\,1} \equiv \Omega$ on the flow variables into smaller and smaller hyperrectangles. At any stage S of the branch-and-bound algorithm, we have a set of active or nonfathomed nodes denoted as T_S. We select an active node t^* in T_S that has the least lower bound, breaking ties arbitrarily, and we partition the hyperrectangle associated with this node according to a suitable **branching variable selection strategy**. Three such strategies are discussed below. The selection of a branching variable according to these strategies ensures

convergence of the overall procedure to a global optimum for NOP(Ω) using the general theory discussed in Sherali, et al. (1998). This process continues by solving the bounding problems for the resulting two node subproblems, and then fathoming nodes as permissible based on this analysis. Whenever the set of active nodes is empty, the process terminates.

Branching Variable Selection Strategy 1 : The choice of a branching variable according to this rule attemps to reduce the discrepancy between the nonlinear constraints (6.2.3b) of NOP(Ω) and its relaxed version used in the lower bounding problem LB($\Omega^{S,\ t}$).

Let (q, x^1, x^2, H, H_S, λ) be the optimal solution obtained for problem LB($\Omega^{S,\ t}$). Define

$$\Delta H_{ij} = (H_i + E_i) - (H_j + E_j) \qquad \forall\ (i, j) \in A \qquad (6.4.1)$$

$$\Delta \overline{H}_{ij} = \text{sign}\ (q_{ij})\ |\ q_{ij}\ |^{1.852}\ \sum_{k=1}^{K} \alpha_{ijk}\ x_{ijk} \qquad \forall\ (i, j) \in A \qquad (6.4.2)$$

where x_{ijk} is given by Equation (6.3.8) for arcs $(i, j) \in P$, and are assumed to be fixed values for all other arcs. The solution to LB($\Omega^{S,\ t}$) is feasible to NOP($\Omega^{S,\ t}$) if $\Delta H_{ij} = \Delta \overline{H}_{ij}\ \forall\ (i, j) \in A$. Note that if $q\text{min}_{ij} = q\text{max}_{ij}$ for any arc (i, j), then we must have $\Delta H_{ij} = \Delta \overline{H}_{ij}$. Hence, discrepant arcs with respect to the latter equality must have positive flow interval widths.

In our first branching strategy, for each $(i, j) \in A$, we compute the following discrepancy in the head loss constraint:

$$\delta_{ij} = |\ \Delta H_{ij} - \Delta \overline{H}_{ij}\ | \qquad \forall\ (i, j) \in A. \qquad (6.4.3)$$

Let

$$\Delta = \max\ \{\delta_{ij}\ :\ (i, j) \in A\}, \qquad (6.4.4)$$

and define the set of arcs for which δ_{ij} is relatively close to Δ as $I = \{(i, j): \delta_{ij} \geq 0.9\Delta\}$.

Note that if $\Delta = 0$, then, as discussed above, we have achieved an optimal solution to the node subproblem, and hence, the corresponding node can be fathomed. Otherwise, the branching variable choice is given by

$$(r, s) \in \text{argmax } \{q\max_{rs} - q\min_{rs} : (r, s) \in I\}. \quad (6.4.5)$$

We then partition the interval $[q\min_{rs}, q\max_{rs}]$ into two sub-intervals as follows:

(*i*) If $q\max_{rs} > 0$ and $q\min_{rs} < 0$, then the two sub-intervals are taken as $[q\min_{rs}, 0]$ and $[0, q\max_{rs}]$.

(ii) Else, the sub-intervals are taken as $[q\min_{rs}, (q\min_{rs}+q\max_{rs})/2]$ and $[(q\min_{rs}+q\max_{rs})/2, q\max_{rs}]$.

Branching Variable Selection Strategy 2 : The lower bounds obtained via the problem LB(•) are very sensitive to the lengths of the flow intervals. Our second strategy attempts to initially reduce the length of the largest such interval by choosing the branching variable (r, s) according to

$$(r, s) \in \text{argmax } \{q\max_{ij} - q\min_{ij} : (i, j) \in A\}. \quad (6.4.6)$$

The intervals for q_{rs} in the two node subproblems are then taken as $[q\min_{rs}, (q\min_{rs}+q\max_{rs})/2]$ and $[(q\min_{rs}+q\max_{rs})/2, q\max_{rs}]$. When a substantial improvement in the global lower bound is not obtained for any pair of successive stages, *i*.e., if for stages S and $S+1$, and corresponding global lower bounds GLB_S and GLB_{S+1}, if

$$GLB_S \geq 0.8GLB_{S+1}, \quad (6.4.7)$$

we adopt branching variable selection Strategy 1 at Stage S+1. This method allows us to initially focus on reducing large flow intervals that might be significantly deteriorating the quality of the global lower bound, and then attempt to reduce the error in estimating the head losses.

Branching Variable Selection Strategy 3: Our third strategy utilizes the fact that the polyhedral approximations in the (q, λ) space are less exact when the flow is distant from both of its bounds. Hence this scheme selects the branching variable (r, s) according to

$$(r, s) \in \text{argmax } \{\min\{q\max_{ij} - \hat{q}_{ij} \ , \ \hat{q}_{ij} \ - q\min_{ij}\} : (i, j) \in A\}. \quad (6.4.8)$$

The intervals for q_{rs} in the two node subproblems are then taken as $[q\min_{rs}, \hat{q}_{rs}]$ and $[\hat{q}_{rs}, q\max_{rs}]$. However, it was found that such a strategy was useful only in the initial stages of branching, because the flow values tend to be driven close to either their lower or upper bounding interval end-points as the algorithm progresses, thereby making the process stall using this strategy. Hence, when a substantial improvement in the global lower bound is not obtained for any pair of successive stages, *i*.e., if for stages S and S+1, and corresponding global lower bounds GLB_S and GLB_{S+1}, if

$$GLB_S \geq 0.9GLB_{S+1}, \quad (6.4.9)$$

we adopt branching variable selection Strategy 1 at Stage S+1, similar to the consideration of Strategy 2.

Tree based Approach for Reducing the Candidate Set of Branching Variables: Using any of the three foregoing branching variable selection strategies, we can reduce the number of possible candidates for selecting a branching variable to be a set of independent arcs in A. To see this, suppose that at the beginning of the branch-and-bound procedure, we construct a maximal spanning tree B for the distribution network via Kruskal's (1956) procedure, using arc weights ($q\max_{ij}$ -

$q\min_{ij}$) $\forall$ $(i, j) \in A$. (The supply nodes are connected to a dummy sink via slack arcs having a large weight for this purpose, in order to balance supply and demand and obtain an equality flow conservation system.) The remaining arcs $\{A\text{-}B\}$in the network are designated as non-tree arcs and form the set from which the branching variables are selected. Note that such a spanning tree yields a valid basis for the underlying network flow problem, and given the flows on the independent nonbasic arcs, the corresponding flows on the dependent basic arcs are uniquely determined. Hence, only the nonbasic arcs are selected for flow partitioning intervals. Upon fixing the flow bounds for the set of nonbasic arcs, the flow bounds for the basic arcs are updated using the representation of the dependent basic variables in terms of the independent nonbasic variables (see Bazaraa, et al., 1990). Let ρ be the arc-to-node ratio of the network. Since the number of arcs in tree B is $|N|$ -1, the number of choices for the branching variable reduces to $|N|(\rho\text{-}1)+1$. Since most water distribution networks are almost "tree-like", *i*.e., have a value of ρ close to 1, the reduction in the number of choices for the branching variable is substantial.

Given the flow bounds on the nonbasic arcs, the actual basic flow bounds are updated using the following recursive tree traversal procedure starting from the ends of the tree. In this process, for any basic arc $(i, j) \in B$, we can obtain.

$$q\max_{ij} = \min\{q\max_{ij}, -b_j - \sum_{\substack{k \in RS(j) \\ k \neq i}} q\min_{kj} + \sum_{\substack{k \in FS(j) \\ k \neq i}} q\max_{jk}\} \tag{6.4.10}$$

$$q\min_{ij} = \max\{q\min_{ij}, -b_j - \sum_{\substack{k \in RS(j) \\ k \neq i}} q\max_{kj} + \sum_{\substack{k \in FS(j) \\ k \neq i}} q\min_{jk}\} \tag{6.4.11}$$

Hence, as the flow interval lengths for nonbasic arcs shrink to zero, so do the corresponding interval lengths for the basic arcs since the right-hand sides of (6.4.10) and (6.4.11) coincide in this case. Let us denote the maximal spanning tree based reduction procedure as **MSTR**.

Note that any of the foregoing three branching procedures can be used in concert with such a branching variable choice reduction procedure. Moreover, we would only need to calculate the values δ_{ij} in (6.4.3) for the nonbasic arcs in this procedure, as shown in Proposition 1.

Proposition 1. Under the maximal spanning tree based procedure (MSTR), we only need to ensure that the δ_{ij} values for the non-tree arcs reduce to zero in order to obtain an optimal solution to the node subproblem.

Proof: We first note that for any arc $(i, j)\in A$, when $q\min_{ij} = q\max_{ij}$, we have $\delta_{ij} = 0$ (Sherali, et al., 1998). Since $q\min_{ij} = q\max_{ij}$ $\forall(i, j)\in A\text{-}B$, implies that $q\min_{ij} = q\max_{ij}$ $\forall(i, j)\in B$ via (6.4.10) and (6.4.11), we have that $\delta_{ij} = 0$ $\forall(i, j)\in A$, and an optimal solution to the node subproblem is at hand. This completes the proof. □

Branch-and-Bound Algorithm

Step 0. Initialization Step

Select an optimality percentage tolerance $100\varepsilon\%$ $(0<\varepsilon<1)$, and limit the maximum head loss discrepancy to be within $\mu = 10^{-6}$. Set the stage counter $S = 1$, and let the set of active nodes be $T_1=\{1\}$. Determine a set of initial flow bounds as implied by the supply and demand rates and the given network connection structure. Denote the initial hyperrectangle as $\Omega^{1,1} \equiv \Omega$. Set $t^* = 1$ and solve the node zero relaxation problem $LB(\Omega^{1,1})$. If infeasible, then stop; the given problem is infeasible. Otherwise, find an optimal solution and initialize the global lower bound GLB_1 to this optimal value $LB_{1,1}$, say. If Δ defined by (6.4.4) less than μ, then stop; the solution to $LB(\Omega^{1,1})$ solves Problem $NOP(\Omega)$. Otherwise, determine the branching variable index (r, s) using any of the

strategies discussed above, and solve the corresponding upper bounding problem ILP($\Omega^{1,1}$). Initialize the best upper bound BUB to the optimal value of this problem, and record the incumbent solution, if this problem is feasible. If $LB_{1,1} \geq BUB(1-\varepsilon)$, then stop with the optimal solution to ILP($\Omega^{1,1}$) as the prescribed solution for Problem NOP(Ω).

Step 1. Partitioning Step

Having identified the active node (S, t^*) to be partitioned, and given the choice (r, s) of the branching variable, partition this node into two sub-nodes associated with the two hyperrectangles Ω^{S,t_1} and Ω^{S,t_2} that are identical to Ω^{S,t^*} except that the two respective interval restrictions on q_{rs} are modified according to the selected branching strategy. Solve LB(Ω^{S,t_1}). If this problem is infeasible, then fathom the corresponding node. Otherwise, find an optimal solution and denote its objective value by LB_{S,t_1}. If $\Delta < \mu$, where Δ is given by (6.4.4) for this solution, then fathom the corresponding node (after updating the incumbent solution) as this solution solves the corresponding node subproblem. If $\Delta \geq \mu$, solve ILP(Ω^{S,t_1}) to obtain an upper bound UB_{S,t_1}. If UB_{S,t_1}<BUB, then update BUB$\leftarrow UB_{S,t_1}$ along with the associated incumbent solution. Also, determine and store a branching variable index (r, s) if this node is to be later selected for further partitioning. Repeat this process for the node subproblem corresponding to t_2, similar to t_1. Update the set of active branch-and-bound nodes for stage S+1 to $T_{S+1} = (T_S - \{t^*\}) \cup \{t_1, t_2\} - \{t : LB_{S,t} \geq BUB(1-\varepsilon)\}$. (Note that $LB_{S+1,t}$ now refers to the lower bound found for the various active nodes $t \in S$+1.) Increment S by 1.

Step 2. Node Selection Step

If $T_S = \phi$ then stop; the incumbent solution is optimal (within the ε-tolerance). Else, select an active node (S, t^*) where $t^* \in \text{argmin}\{LB_{S,t}: t \in T_S\}$. Set $GLB_S = LB_{S,t^*}$. Note that this is the least lower bound over the active nodes at stage S. Return to Step 1.

Some computational results for this branch-and-bound procedure using (6.3.20) to generate the lower bounding problem is described in Section 6.6. Any valid lower bounding problem can be substituted in lieu of (6.3.20) in this branching procedure. In the next section, we describe the derivation of an alternative tighter lower bounding linear program that can be embedded in such a branch-and-bound framework to develop a possibly more efficient solution procedure.

6.5. A Reformulation-Linearization Technique for Computing Enhanced Lower Bounds

In the foregoing section, we have presented a general branch-and-bound algorithmic framework for solving Problem NOP(Ω) in which any suitable lower bounding scheme can be inserted in lieu of that described in Section 6.2. As an alternative to (6.3.20), we now describe an enhanced lower bounding procedure that is motivated by the Reformulation-Linearization Technique (RLT) of Sherali and Tuncbilek (1992) for solving polynomial programming problems. Our purpose here is to study the tradeoff between a quicker versus a more involved, but stronger, lower bounding procedure with respect to the overall effort for solving the problem. To construct such a lower bounding problem, we generate a set of additional constraints using the RLT concept as follows.

Reformulation Step. The following quadratic valid constraints are generated based on the products of the stated pairs of inequalities (written in the form $\{\bullet\} \geq 0$), or based on products of equations with variables.

(a) Using the pipe length constraints in (6.3.20c), generate the equality product constraints

$$\left(\sum_{k=1}^{K} x_{ijk}^{1} + \sum_{k=1}^{K} x_{ijk}^{2} \right) q_{ij} = L_{ij} q_{ij} \ \forall (i, j) \in P.$$

(b) Multiply each inequality in (6.3.20g), …, (6.3.20n) with each corresponding variable $x_{ijk}, \forall (i, j) \in P, k = 1, \ldots, K$.

Linearization Step. Linearize the resulting product constraints by substituting $y_{ijk} = q_{ij} x_{ijk}, \forall (i, j) \in P, k = 1, .., K$, and by using equations (6.3.7) and (6.3.8).

This produces the following linear programming lower bounding problem, as a further relaxation of NOP(Ω), and will be referred to as RLT(Ω) based on the hyperrectangle $\Omega = \{q: q\min_{ij} \le q_{ij} \le q\max_{ij}, \forall (i, j) \in A\}$. Here, for sake of notational convenience, we let $H \in \mathbb{H}$ denote the pressure head bounds for the nodes in the network, as specified by (6.3.20o)-(6.3.20q).

Also, for any fixed value $\hat{\lambda}, 0 < \hat{\lambda} < 1,$ such that $q'(\hat{\lambda})$ exists, let

$$\overline{q}(\hat{\lambda}) = q(\hat{\lambda}) + (1 - \hat{\lambda}) q'(\hat{\lambda}), \text{ and} \qquad (6.5.1)$$

$$\underline{q}(\hat{\lambda}) = q(\hat{\lambda}) - \hat{\lambda} q'(\hat{\lambda}) \qquad (6.5.2)$$

denote the values of q estimated at $\lambda = 1$ and $\lambda = 0$, respectively, by the tangential first-order approximation to $q(\lambda)$ constructed at $\hat{\lambda}$. The reformulation steps (a) and (b) respectively generate the constraints (6.5.3e), and (6.5.3i), (6.5.3k), (6.5.3m), (6.5.3o), (6.5.3q), (6.5.3s), (6.5.3u), (6.5.3w) upon using the substitutions (6.5.1) and (6.5.2). These constraints are listed below in sequential order, with the RLT constraint preceded by the one used to generate it.

RLT(Ω):

Minimize: $$\sum_{ij \in P} \sum_{k=1}^{K} c_k (x_{ijk}^{1} + x_{ijk}^{2}) + \sum_{i \in S} c_{si} H_{si} \qquad (6.5.3a)$$

subject to

$$\sum_{k=1}^{K}(v\min_{ij})\alpha_{ijk}x_{ijk}^{1}+\sum_{k=1}^{K}(v\max_{ij})\alpha_{ijk}x_{ijk}^{2}=(H_i+E_i)-(H_j+E_j)\ \forall(i,j)\in A \tag{6.5.3b}$$

$$\sum_{k=1}^{K}x_{ijk}^{1}+\sum_{k=1}^{K}x_{ijk}^{2}=L_{ij}\ \forall(i,j)\in P \tag{6.5.3c}$$

$$\sum_{k=1}^{K}x_{ijk}^{1}=\lambda_{ij}L_{ij}\forall(i,j)\in P$$

$$x_{ijk}^{1}=\lambda_{ij}\tilde{x}_{ijk}\forall(i,j)\in A-P,k=1,...,K,$$

$$x_{ijk}^{2}=(1-\lambda_{ij})\tilde{x}_{ijk}\forall(i,j)\in A-P,k=1,...,K \tag{6.5.3d}$$

$$\sum_{k=1}^{K}y_{ijk}=L_{ij}q_{ij}\ \forall(i,j)\in P,$$

$$y_{ijk}=q_{ij}\tilde{x}_{ijk}\ \forall(i,j)\in A-P,k=1,...,K \tag{6.5.3e}$$

$$\sum_{j\in FS(i)}q_{ij}-\sum_{j\in RS(i)}q_{ji}=b_i\ \ \forall i\in D \tag{6.5.3f}$$

$$\sum_{j\in FS(i)}q_{ij}-\sum_{j\in RS(i)}q_{ji}\le b_i\ \ \forall i\in S \tag{6.5.3g}$$

$$q_{ij}\le q_{ij}(\hat{\lambda}_{ij})+(\lambda_{ij}-\hat{\lambda}_{ij})q_{ij}'(\hat{\lambda}_{ij})\text{ for }\hat{\lambda}_{ij}=0,\frac{\bar{\lambda}_{ij}}{2},\bar{\lambda}_{ij}\ \forall(i,j)\ni q\min_{ij}<0,q\max_{ij}>0,\bar{\lambda}_{ij}\text{ exists} \tag{6.5.3h}$$

$$y_{ijk}\le\bar{q}_{ij}(\hat{\lambda}_{ij})x_{ijk}^{1}+\underline{q}_{ij}(\hat{\lambda}_{ij})x_{ijk}^{2}\text{ for }\hat{\lambda}_{ij}=0,\frac{\bar{\lambda}_{ij}}{2},\bar{\lambda}_{ij}\ \forall(i,j)\ni q\min_{ij}<0,q\max_{ij}>0,\bar{\lambda}_{ij}\text{ exists},k=1,...,K \tag{6.5.3i}$$

$$q_{ij}\le\lambda_{ij}q\min_{ij}+(1-\lambda_{ij})q\max_{ij}\ \forall(i,j)\ni q\min_{ij}<0,q\max_{ij}>0,\bar{\lambda}_{ij}\text{ does not exist} \tag{6.5.3j}$$

$$y_{ijk}\le q\min_{ij}x_{ijk}^{1}+q\max_{ij}x_{ijk}^{2}\forall(i,j)\ni q\min_{ij}<0,q\max_{ij}>0,\bar{\lambda}_{ij}\text{ does not exist},k=1,...,K \tag{6.5.3k}$$

$$q_{ij} \geq q_{ij}(\hat{\lambda}_{ij}) + (\lambda_{ij} - \hat{\lambda}_{ij}) q'_{ij}(\hat{\lambda}_{ij}) \text{ for } \hat{\lambda} = \tilde{\lambda}_{ij}, (\frac{1+\tilde{\lambda}_{ij}}{2}), 1 \; \forall (i, j) \ni q\min_{ij} < 0, q\max_{ij} > 0, \tilde{\lambda}_{ij} \text{ exists} \tag{6.5.3l}$$

$$y_{ijk} \geq \bar{q}_{ij}(\hat{\lambda}_{ij}) x^1_{ijk} + \underline{q}_{ij}(\hat{\lambda}_{ij}) x^2_{ijk} \text{ for } \hat{\lambda} = \tilde{\lambda}_{ij}, (\frac{1+\tilde{\lambda}_{ij}}{2}), 1 \; \forall (i, j) \ni q\min_{ij} < 0, q\max_{ij} > 0, \tilde{\lambda}_{ij} \text{ exists}, k = 1, \ldots, K \tag{6.5.3m}$$

$$q_{ij} \geq \lambda_{ij} q\min_{ij} + (1 - \lambda_{ij}) q\max_{ij} \; \forall (i, j) \ni q\min_{ij} < 0, q\max_{ij} > 0, \tilde{\lambda}_{ij} \text{ does not exist} \tag{6.5.3n}$$

$$y_{ijk} \geq q\min_{ij} x^1_{ijk} + q\max_{ij} x^2_{ijk} \forall (i, j) \ni q\min_{ij} < 0, q\max_{ij} > 0, \tilde{\lambda}_{ij} \text{ does not exist}, k = 1, \ldots, K \tag{6.5.3o}$$

$$q_{ij} \geq \lambda_{ij} q\min_{ij} + (1 - \lambda_{ij}) q_{ij} \; \forall (i, j) \ni q\min_{ij} \geq 0 \tag{6.5.3p}$$

$$y_{ijk} \geq q\min_{ij} x^1_{ijk} + q\max_{ij} x^2_{ijk} \forall (i, j) \ni q\min_{ij} \geq 0, k = 1, \ldots, K \tag{6.5.3q}$$

$$q_{ij} \leq \lambda_{ij} q\min_{ij} + (1 - \lambda_{ij}) q\max_{ij} \; \forall (i, j) \ni q\max_{ij} \leq 0 \tag{6.5.3r}$$

$$y_{ijk} \leq q\min_{ij} x^1_{ijk} + q\max_{ij} x^2_{ijk} \forall \; \forall (i, j) \ni q\max_{ij} \leq 0, k = 1, \ldots, K \tag{6.5.3s}$$

$$q_{ij} \leq q_{ij}(\hat{\lambda}_{ij}) + (\lambda_{ij} - \hat{\lambda}_{ij}) q'_{ij}(\hat{\lambda}_{ij}) \text{ for } \hat{\lambda}_{ij} = 0, 0.25, 0.5, 0.8, 0.9 \; \forall (i, j) \ni q\min_{ij} \geq 0 \tag{6.5.3t}$$

$$y_{ijk} \leq \bar{q}_{ij}(\hat{\lambda}_{ij}) x^1_{ijk} + \underline{q}_{ij}(\hat{\lambda}_{ij}) x^2_{ijk} \text{ for } \hat{\lambda}_{ij} = 0, 0.25, 0.5, 0.8, 0.9 \; \forall (i, j) \ni q\min_{ij} \geq 0, k = 1, \ldots, K \tag{6.5.3u}$$

$$q_{ij} \geq q_{ij}(\hat{\lambda}_{ij}) + (\lambda_{ij} - \hat{\lambda}_{ij}) q'_{ij}(\hat{\lambda}_{ij}) \text{ for } \hat{\lambda}_{ij} = 0.1, 0.2, 0.5, 0.75, 1 \; \forall (i, j) \ni q\max_{ij} \leq 0 \tag{6.5.3v}$$

$$y_{ijk} \geq \bar{q}_{ij}(\hat{\lambda}_{ij})x^1_{ijk} + \underline{q}_{ij}(\hat{\lambda}_{ij})x^2_{ijk} \text{ for } \hat{\lambda}_{ij} = 0.1,0.2,0.5,0.75,1 \; \forall (i,j) \ni q\max_{ij} \leq 0, k = 1,...,K \quad (6.5.3w)$$

$$x^1_{ijk}, x^2_{ijk}, y_{ijk} \geq 0 \qquad \forall (i,j) \in P,\ k=1, \ldots, K \quad (6.5.3x)$$

$$0 \leq \lambda_{ij} \leq 1 \qquad \forall (i,j) \in A \quad (6.5.3y)$$

$$H \in \mathbb{H},\ q \in \Omega. \quad (6.5.3z)$$

Remark 2. It is possible to generate RLT constraints by also multiplying the inequalities in (6.3.20s) by the bound factor inequalities $q\min_{ij} \leq q_{ij} \leq q\max_{ij}$, $\forall (i,j) \in P$, as shown below.

$$x_{ijk} * (q\max_{ij} - q_{ij}) \geq 0 \Rightarrow y_{ijk} \leq q\max_{ij}\, x^1_{ijk} + q\max_{ij}\, x^2_{ijk} \;\forall (i,j) \in P.$$

$$x_{ijk} * (q_{ij} - q\min_{ij}) \geq 0 \Rightarrow y_{ijk} \geq q\min_{ij}\, x^1_{ijk} + q\min_{ij}\, x^2_{ijk} \;\forall (i,j) \in P.$$

However, these constraints are implied by (stronger) inequalities generated by the foregoing RLT process. This is clearly evident for the cases of convex and concave $q(\lambda)$, and cases (b) and (c) for concave-convex $q(\lambda)$. We can see that these constraints are also implied for the concave-convex case (a) via equations (6.5.3h) and (6.5.3m), since by (6.3.12) and (6.3.13), (6.5.1), and (6.5.2), $\underline{q}(\bar{\lambda}) = q\min$, and $\bar{q}(\tilde{\lambda}) = q\max$.

Note that if n_s is the number of supporting hyperplanes used to develop the polyhedral outer approximation for each arc flow, the RLT formulation has $(n_s|A|K + |A|)$ constraints more than the original lower bounding problem. In the next section, we present some computational experience for the branch-and-bound procedure using the RLT enhanced lower bounding problem, as well as for the original lower bounding formulation discussed in Section 6.3.

6.5. Computational Experience

In this section, we apply the proposed branch-and-bound algorithm to three standard test problems from the literature, and a newly generated Blacksburg network. The algorithms were implemented on a SUN SPARC 10 UNIX workstation, using the CPLEX 6.0 callable library to

solve the linear programming problems. The computer code was written in C++. The algorithm was implemented for the three branching variable selection strategies discussed in Section 6.3, using the tree-based branching variable choice reduction procedure MSTR. Various optimality tolerances ε between 0.1 and 10^{-6} are used, with the higher tolerance values being used wherever computationally feasible. In addition, we also implemented the algorithm using a lesser number of supporting hyperplanes (four) for constructing the lower bounding linear programs, with and without RLT enhancements. The data, computational results and the best design values for each of these four networks are presented sequentially below, in order of problem size. Illustrations of the network configurations for these four test problems are displayed in Figures 6-9, respectively.

TEST PROBLEM 1: TWO LOOP NETWORK

Two Loop Network : This is a single source test problem originally presented by Alperovits and Shamir (1977). Data for this problem are presented in Tables 6.1 and 6.2, and in Figure 6.4. The Hazen William's coefficient C_{HW} is 130 for all links. The flow bounds in Table 6.1 were logically determined as previously stated in Section 6.3. The set of commercially available diameters d (inches) was taken as {1, 2, 3, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24}, with the corresponding costs per unit length (meter) being {2, 5, 8, 11, 16, 23, 32, 50, 60, 90, 130, 170, 300, 550}. Note that the original Alperovits and Shamir (1977) test problem excludes certain pipe diameters, whereas several authors have later solved this problem by including all the possible aforementioned diameters. To enable a comparison, as well as from practical viewpoint, we permit the selection of all commercially available pipe diameters. Tables 6.1c-6.1f summarize the results obtained using different optimality tolerances for the different branching variable strategies, lower bounding schemes, and number of supporting hyperplanes.

Fujiwara et al. (1990) report a best heuristic solution having an objective function value of $415,271, while Loganathan et al. (1990) found a solution value of $412,931. Eiger et al. (1994) report a solution with an objective value of $402, 352 and a global lower bound of $400,703 for an optimality tolerance of 0.5%. However, their solution contains some violations in the flow conservation constraints. In fact, note that the best global lower bound of $403,214 obtained by our algorithm is greater than Eiger et al.'s near feasible solution of $402,352. The heuristic of Loganathan et al. (1995) yields a solution having a total cost of $403,657. Sherali et. al (1998) obtain a solution with an objective value of $403,390, which is slightly better than the previous solution. Earlier, Sherali and Smith (1995) had recently obtained a global lower bound of $403,385 on this problem, along with a feasible solution of $403,386, which is $1 within global optimality. Their algorithm, when implemented on the same computer and using CPLEX 2.0 to solve the LP relaxations, enumerated only 49 nodes, but consumed 342 CPU seconds due to the size of their lower bounding problem. The optimum solution presented in Table 6.6.1g is obtained using our branch-and-bound procedure with the lower bounding problem (6.3.20), four supporting hyperplanes per arc, the tree-based based branching scheme in concert with Strategy 1, and with both ε and μ being fixed at 10^{-6} (see Table 6.1e). This solution is the most accurate one reported in the literature, and has an objective value that lies within 0.2$ of global optimality, and was derived while consuming only 12 CPU seconds.

With respect to the efficiency of branching schemes, it is evident from Tables 6.1c and 6.1d that the Strategies 1 and 3 performed better than Strategy 2. Hence, only these two strategies were used for further testing on this test problem. It can be observed from Tables 6.1e-6.1f that the computational efficiency of the procedure in terms of CPU time expended is governed by the size of the lower bounding problem. While the larger, tighter relaxations consume more time to solved, their

benefit is that they result in a proportional reduction in the number of branch-and-bound nodes being enumerated. For the present test problem, this compromise is not favorable. On the other hand, Tables 6.1c, 6.1e, 6.1d, and 6.1f indicate that the use of the MSTR procedure significantly reduces computational effort. On the average, a reduction in computation time by a factor of 5 can be observed for the non-RLT formulation and by a factor of 3 for the RLT-enhanced formulation. Based on comparative results for Strategies 1 and 3, it can be seen that on the average, Strategy 3 outperforms Strategy 1. Hence, for the remainder of the tests in this Section, Strategy 3 is used throughout in tandem with the MSTR procedure.

TEST PROBLEM #2: HANOI NETWORK

The Hanoi network is a single source network consisting of three basic loops, thirty two nodes and thirty four links. All the nodes are located at the same elevation. The Hazen William's coefficient C_{HW} is 130 for all links. The pressure head for all demand nodes is restricted to lie within the interval [30, 100]. The data for this problem appears in Fujiwara and Khang (1990) and is reproduced here for convenience. In the network layout shown in Figure 6.6, the directions of some of the arcs have been reversed from as they appear in the previous papers, to be consistent with our definition of an arc in a network. The set of commercially available diameters d is {12, 16, 20, 24, 30, 40} and the corresponding cost per unit length in $/m is {45.73, 70.40, 98.39, 129.33, 180.74, 278.28}. The flow bounds given in Table 6.2a are heuristically determined as described in Sherali et al. (1998). The details of the least cost pipe design are presented in Table 6.2e, along with the optimum set of flow values and pressure heads.

Eiger et al. (1994) report a solution having an objective function value of $6,026,660 for this problem using an optimality tolerance of 0.5%. However, their solution contains some violations in the flow conservation constraints, as shown by Sherali et al., (1998), who obtained a

solution having an objective value of $6,058,976, which is the best solution previously reported in the literature. The best solution found by our algorithm has an objective value of $6,055,542 which is significantly better than the values reported in the literature for this test problem. This solution was obtained using the lower bounding problem (6.3.20), the MSTR procedure in concert with branching strategy 3, and four supporting hyperplanes, using a value of 10^{-6} for ε and for μ, and expending 4 minutes of CPU time. This solution is within 10^{-4} % of optimality (within 6$ of global optimality) as verified by our global lower bound, and is the best feasible solution reported in the literature for this problem.

It can be observed from Tables 6.2c and 6.2d that the introduction of additional supporting hyperplanes results in a marginal increase in CPU time and a reduction in the number of nodes enumerated, similar to the results obtained for the two loop network. Likewise, the introduction of the RLT enhanced lower bounding scheme has a similar effect as for the foregoing test problem. However, it was observed that in the absence of the MSTR procedure, the RLT-enhanced procedure enumerated relatively lesser number of nodes while also consuming lesser CPU time when compared with the non-RLT based scheme. We will comment more on this behavior at the end of this section.

TEST PROBLEM 3: NEW YORK NETWORK

The New York test network configuration is displayed in Figure 6.8, along with the parallel pipe representation. The additional dummy nodes having zero demand that are constructed are depicted in the figure with their indices in bold lettering. The original network has 20 nodes and 21 arcs, while the expanded network (shown in Figure 6.8) has 26 nodes and 33 arcs. The data for the original network is presented in Tables 6.3a and 6.3b. The data for the expanded network was generated using the procedure described in Section 6.2. The set of available pipe diameters is chosen in increments of 4 inches. The Hazen-Williams coefficient C_{HW} equals 100 for all links. The nodes are

all assumed to be at the same elevation. Since there is only a single source node, the initial flow bounds for the arcs are simply set equal to $\pm b_{i*}$, where $i*$ is the source node index. It may be possible to devise tighter flow bounds based on the discussion in Section 6.3,this was not done for the purpose of testing the algorithm here. The computational results obtained are presented in Tables 6.3c and 6.3d. The minimum cost design along with the corresponding flow values are presented in Tables 6.3e and 6.3f. The coefficients used in the head-loss equation are the same as that used in Fujiwara and Khang (1990) and Loganathan et al. (1985). The flow rate exponent value was set equal to 1.85, while the head loss coefficient was set equal to 851500 to conform with the flow rates being measured in cubic-feet per second, the pipe diameters in inches, and the head losses in feet.

The New York test network problem was first solved using parallel links by Schaake and Lai (1969) and they obtained a solution having an objective value of $\$77.61(10^6)$. Fujiwara and Khang (1990) used their two-phase approach to solve this problem, but the solution presented by them was infeasible. Quindry, et al. (1981) obtained a solution having a total cost of $\$63.581(10^6)$, while Gessler (1982), Bhave (1985), and Morgan and Goulter (1985) obtained solutions having costs of $\$41.2(10^6)$, $\$40.18(10^6)$, and $\$39.018(10^6)$, respectively. Loganathan et al. (1995) used a simulated annealing based heuristic procedure to further improve the objective value to $\$38.04(10^6)$. All these approaches are heuristic in nature and simply seek to determine (at best) local optimal solutions, providing no indication of a competitive global lower bound on the optimum value. Using the non-RLT enhanced procedure, we were able to obtain such a global lower bound of value 37878580.86 with the corresponding upper bounding solution having a cost of 37878581.21 The minimum cost design corresponding to this solution is presented in Table 6.3e. This is the best solution reported thus far in the literature and lies within 10^{-6}% (or within 0.4$) of optimality. Note that this result is obtained by considering 4 inch pipe increments for the set of available pipe diameters. All the results

presented above since 1985 use 12 inch pipe diameter increments. Hence, for the sake of comparison, the procedure (six supports, MSTR, but no RLT) was run using 12 inch pipe increments and a global lower bound value of 38,067,895 along with a corresponding feasible solution of 38,067,935, after solving 2467 linear programs. It was observed that while the optimal pipe diameters coincided with that obtained by Loganathan et al. (1995), the corresponding pipe segment lengths were different. This is due to the fact that the head loss values obtained by Loganathan et al. (1995) have a feasibility tolerance of 0.1, while the results presented in this chapter were obtained using higher precision, using a feasibility tolerance value of 10^{-6}. Consequently, the objective cost corresponding to the optimal solution is higher than that obtained by Loganathan et al. (1995), and represents a relatively more accurate estimate of the actual solution.

It can be seen from Tables 6.3c and 6.3d that the computational times are significantly higher for this test case, as compared with the computational efforts for the previous two test problems. One important reason for this difference is that no logical test based schemes were used to generate tight initial flow interval bounds. In fact the initial feasible solution was itself near-optimal, but was polished to the final solution only toward the tail end of the branching procedure (see Tables 6.3c and 6.3d). The results for this test problem also differ from the previous two in that the introduction of additional hyperplanes results in an improvement in the computational effort, both in terms of number of nodes enumerated and the CPU time expense. However, the RLT-based scheme, while enumerating fewer nodes, continued to expend more CPU time.

TEST PROBLEM 4: BLACKSBURG TEST NETWORK

Figure 6.9 depicts a network representation of a newly designed subdivision of the water distribution system in the town of Blacksburg, Virginia. The network data for this problem was acquired from the public works department of the town, along with other problem parameters such

as pressure requirements, locations of fire hydrants, cost factors, pipe quality that is reflected via the associated C_{HW} value, and demand requirements. The unit pipe costs used in this problem represent real-life values and include installation costs as well. The Blacksburg network test problem deals with the task of pipe replacement.. The link and node data for the network are presented in Tables 6.4a and 6.4b. The set of pipes whose diameters are fixed is listed in Table 6.4a. A Hazen-Williams coefficient value of 120 was used for all the links. Expression (6.2.1) was used to compute the head losses. The flow rates were converted using double precision from gallons per minute (gpm) into units of m^3/hr, the pipe diameters were specified in centimeters, and the head losses in meters.

The computational results for the Blacksburg network are similar to that for the Hanoi network in that the introduction of additional supporting hyperplanes, or the use of RLT results in a decrease in the number of nodes enumerated, but an increase in the CPU time expense. The best solution obtained for this test problem has an objective value of 577066.7, along with a best global lower bound of 577066.3 derived using the RLT procedure. This is within 10^{-4}% or within 0.4$ of optimality. The mimimum cost design results is presented in Table 6.4e, along with the optimal pressure head values for the nodes. The flow values for any link can be computed from these results using the head loss equation (6.2.1).

The computational results for the four test problems studied thus far clearly show that using the maximal spanning tree reduction procedure (MSTR) results in a superior computational performance. In fact, for the two larger problems, the non-tree based procedure was unable to obtain good quality feasible solutions within the time limit imposed. Furthermore, the use six versus four hyperplanes in the polyhedral approximation of the flow relationships improved the relative performance for two of the larger test problems, while for the other two test problems, it resulted in only a marginally greater CPU time expense. However, in the cases where the introduction of

additional supports was beneficial, a significant reduction in computational time was observed. Hence, we recommend the use of at least the six prescribed supporting hyperplanes in the lower bounding problem, and suggest experimenting with additional supports.

It was generally observed that the RLT enhanced scheme yielded much tighter lower bounds, and resulted in fewer branch-and-bound nodes being enumerated as compared with the non-RLT scheme. With regard to overall computational performance, the benefits of using such tighter lower bounding formulations were more pronounced as the size of the problems increased. Tables 6.4f and 6.4g illustrate this phenomenon. In particular, it can be seen from Table 6.4g that for the Hanoi network, the RLT scheme enumerates significantly fewer nodes and consumes lesser CPU time when compared with the non-RLT based formulation. It is possible that for problems having several more independent variables than those analyzed in this chapter, such RLT-enhanced procedures may be beneficial.

Overall, we suggest the use of the non RLT- enhanced formulation that uses additional supporting hyperplanes, as well as the maximal spanning tree based procedure, for solving problems of sizes similar to the test cases used in this chapter. For larger problems having more independent variables, it might be a good idea to also test the RLT-enhanced procedure that uses additional supporting hyperplanes in order to investigate its competitiveness with the former approach.

6.7. Summary and Conclusions

In this chapter, we have proposed an improved method to compute global lower bounds for a water distribution network design problem. The lower bounding scheme takes advantage of the monotonicity and the convex-concave nature of the nonlinear constraints to develop tight linear programming relaxations. This lower bounding formulation is further enhanced by introducing additional RLT related constraints. The flow-conserving nature of this relaxation also permits the

design of an accompanying upper bounding heuristic for obtaining good quality network designs. Several branching strategies are tested in concert with a maximal spanning tree based branching variable choice reduction procedure.

A new test problem dealing with the water distribution system in Blacksburg, Virginia, is introduced to the literature in this chapter. Results obtained on three other standard test problems from the literature demonstrate the efficacy of the proposed methodology. Improved solutions are reported for each of these problems, significantly so for the two larger cases of the Hanoi and the New York test networks for which solutions proved to within 10^{-4}% of optimality are derived for the first time in the literature. Further enhancements in algorithmic efficiency can be achieved by including a more effective preprocessor to deduce valid, tighter initial bounds on the flow variables. The algorithm can also benefit by computing sharper upper bounds by using some local optimization scheme, rather than simply evaluating the flow solution produced by the lower bounding problem. The application of efficient schemes such as those described in Sherali and Smith (1997) to obtain tight flow bounds or upper bounds for each node subproblem is more critical in the case of the network design problems having several independent variables. Such problems can also benefit via the construction of tighter lower bounding problems through the use of an additional, suitable number of supporting hyperplanes in the approximation of the flow relationships, once the direction of flow in any link is determined, as well as through the proposed RLT constructs. Such investigations and further computational tests are proposed for further research.

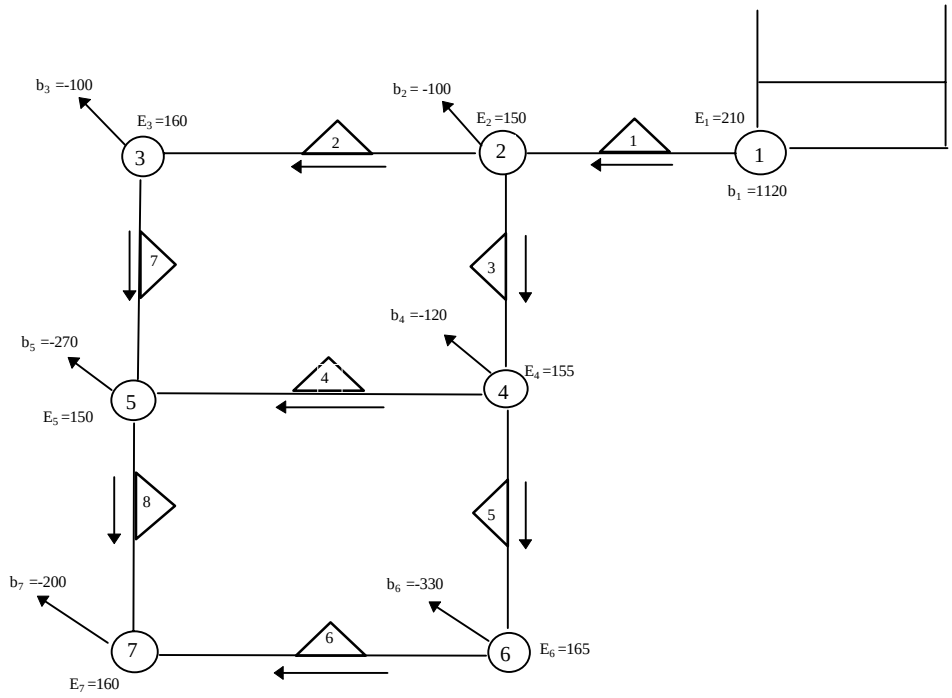

Figure 6.6. Two Loop Network Configuration.

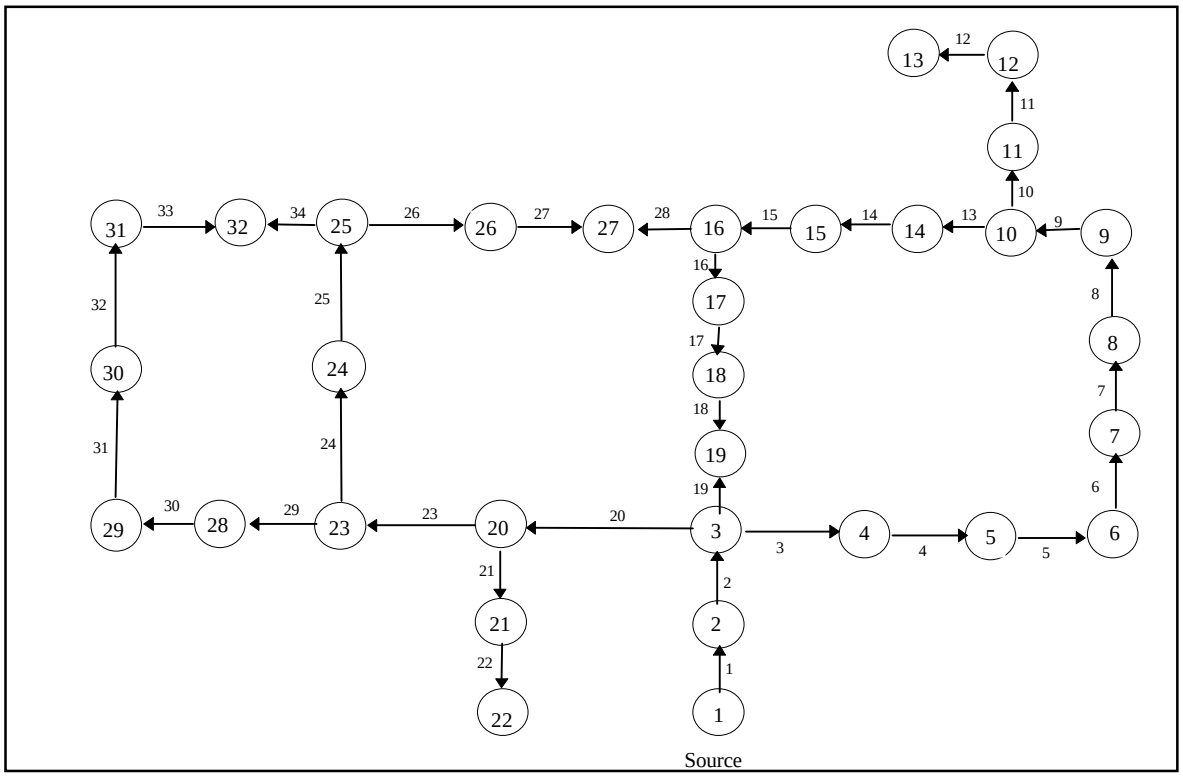

Figure 6.7. Hanoi Network Configuration.

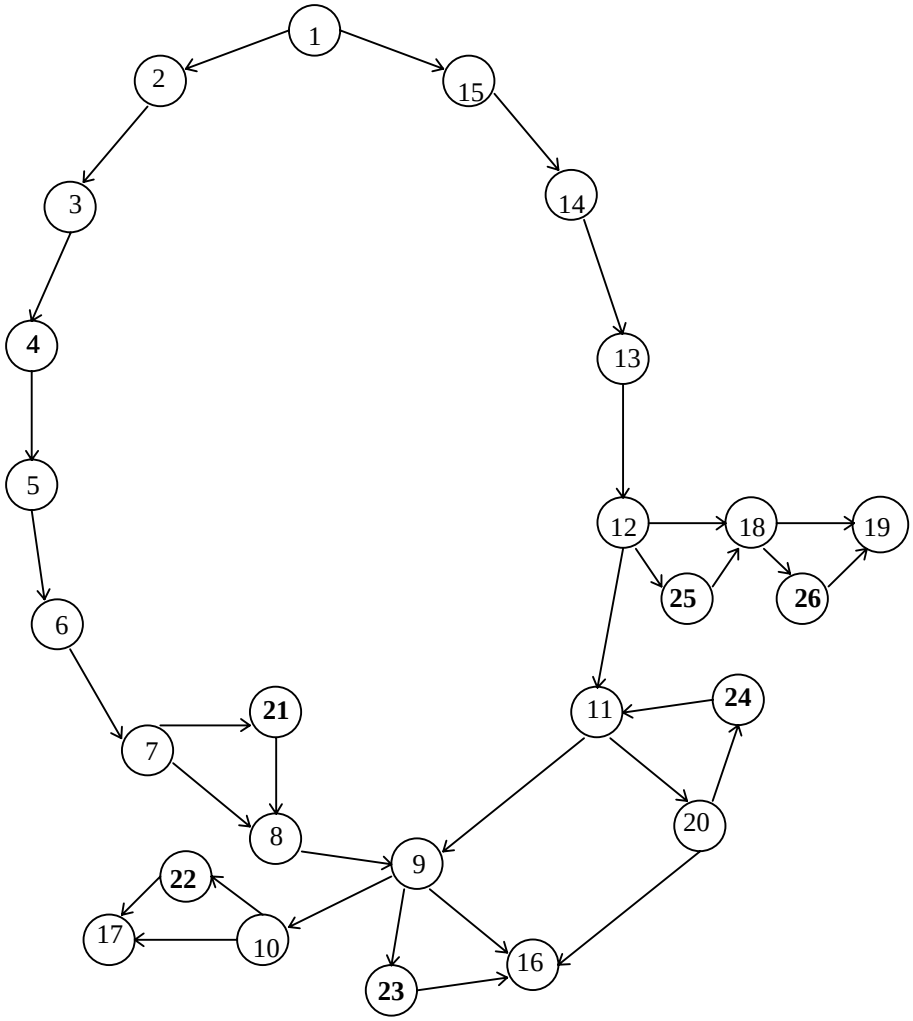

Figure 6.8. The New York Test Network (including Parallel Pipes).

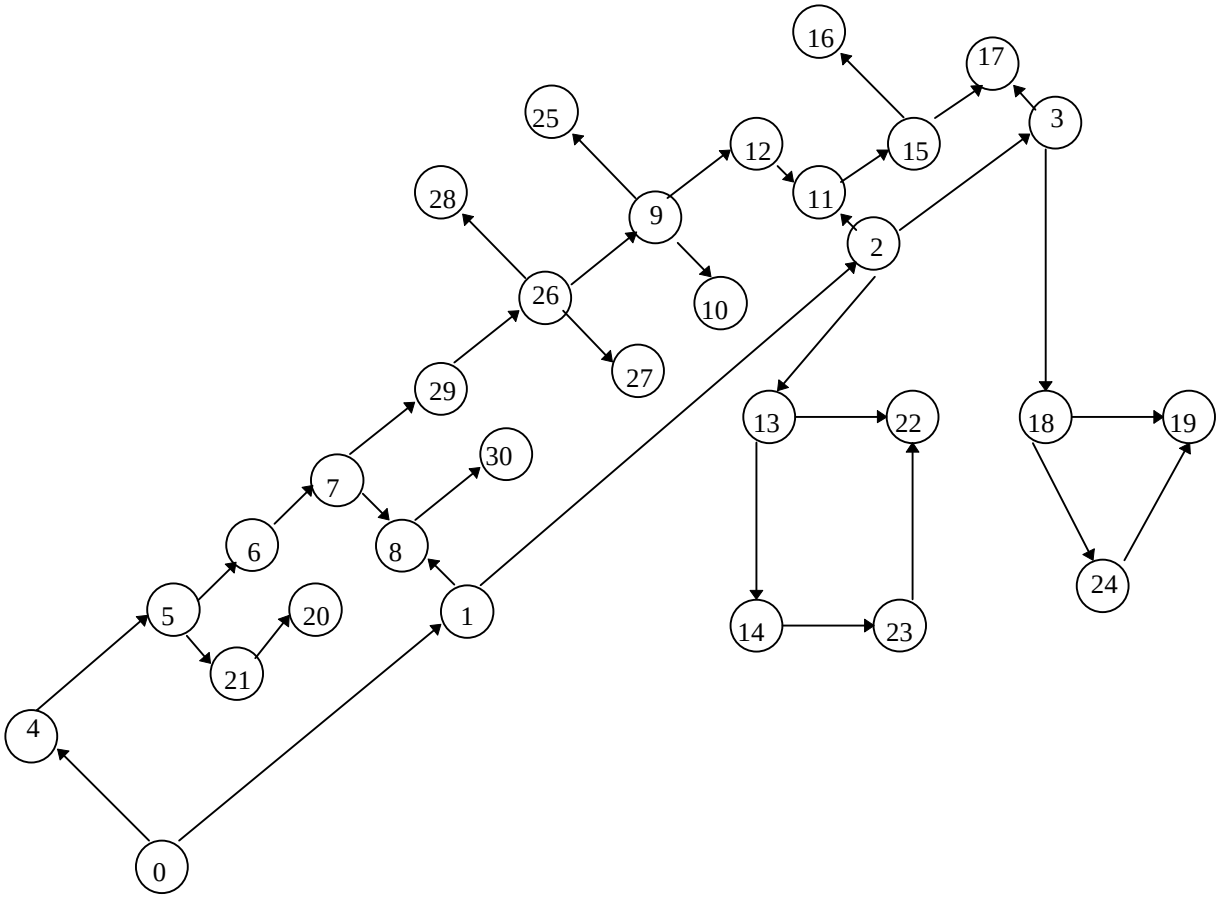

Figure 6.9. Blacksburg Network Configuration.

Table 6.1a. Arc Data for the Two Loop Network

Link Index	Arcs	Length (m)	$q$min (m^3/hr)	$q$max (m^3/hr)
1	(1, 2)	1000	1120	1120
2	(6.2, 3)	1000	0	1020
3	(6.2, 4)	1000	0	1020
4	(6.4, 5)	1000	-650	900
5	(6.4, 6)	1000	-120	900
6	(6, 7)	1000	-450	570
7	(6.3, 5)	1000	-100	920
8	(6.5, 7)	1000	-370	650

Table 6.1b. Node Data for the Two Loop Network.

Node	Elevation (m)	Pressure Bounds on H_i (m)	Supply or Demand (m^3/hr)	H_{iL} (m)	H_{iU} (m)
1	210	-	1120	*N*/A	*N*/A
2	150	[30, 60]	-100	180	210
3	160	[30, 50]	-100	190	210
4	155	[30, 55]	-120	185	210
5	150	[30, 60]	-270	180	210
6	165	[30, 45]	-330	195	210
7	160	[30, 50]	-200	190	210

Table 6.1c. Computational results for the two-loop network using no RLT or MSTR, for various ε values.

Branching Strategy	n_s	ε	Global Lower Bound	Global Upper Bound	# LP Solved	CPU Time
1	4	0.05	389837.029	405880.014	535	36.87 sec
2			389836.855	405919.744	553	37.05 sec
3			390550.315	410309.305	505	32.58 sec
1	4	0.01	399498.974	403514.243	887	60.98 sec
2			402488.708	403524.775	943	65.28 sec
3			399423.260	403499.261	771	62.11 sec
1	6	0.05	392045.838	406672.043	535	56.36 sec
2			392045.775	406711.677	567	58.82 sec
3			394263.402	406672.043	505	505 sec
1	6	0.01	400482.238	403516.952	793	83.82 sec
2			400423.051	403527.352	845	87.03 sec
3			400508.586	403570.969	663	63.48 sec

Table 6.1d. Computational results for the two-loop network using RLT but no MSTR, for various ε values.

Branching Strategy	n_s	ε	Global Lower Bound	Global Upper Bound	# LP Solved	CPU Time
1	4	0.05	395567.003	405836.941	413	88.83 sec
2			395539.944	405881.218	455	94.92 sec
3			397376.652	413213.138	451	90.76
1	4	0.01	399527.969	403516.178	521	109.13 sec
2			403133.502	403523.650	585	118.05 sec
3			399590.862	403557.453	517	116.67 sec
1	6	0.05	397870.438	412963.496	429	127.54 sec
2			397870.763	412736.802	473	132.61 sec
3			398182.793	412963.496	491	137.84 sec
1	6	0.01	400909.596	403531.560	481	139.95 sec
2			400957.078	403535.463	561	158.30 sec
3			399657.338	403591.425	527	151.48 sec

Table 6.1e. Computational results for the Two Loop Network using MSTR but no RLT, for various ε values.

Branching Strategy	n_s	ε	Global Lower Bound	Global Upper Bound	# LP Solved	CPU Time (sec)
1	4	10^{-2}	402787.55	423741.858	77	6.31
3			402253.84	416956.497	81	6.82
1	4	10^{-3}	403302.157	405436.186	109	8.92
3			403231.776	406515.798	97	7.94
1	4	10^{-6}	403385.238	403385.418	147	12.10
3			403385.1	403385.401	141	11.64
1	6	10^{-2}	403283.316	405256.724	97	12.11
3			403307.675	407329.796	99	12.43
1	6	10^{-3}	403367.562	403732.5	105	12.77
3			403340.551	403364.363	111	13.85
1	6	10^{-6}	403385.238	403385.418	139	16.97
3			403385.093	403385.408	147	18.05

Table 6.1f. Computational Results for the Two Loop Network using both RLT and MSTR, for various ε values.

Branching Strategy	n_s	ε	Global Lower Bound	Global Upper Bound	# LP Solved	CPU Time (sec)
1	4	10^{-2}	403304.123	404092	83	26.21
3			403333.05	403808.618	75	23.96
1	4	10^{-3}	403374.374	403573.12	91	28.86
3			403374.482	403670.503	81	26.62
1	4	10^{-6}	403382.571	403406.737	105	33.82
3			403382.141	403406.736	99	33.25
1	6	10^{-2}	403283.35	405256.724	75	32.59
3			403273.562	404763.839	67	30.05
1	6	10^{-3}	403367.57	403732.5	85	37.27
3			403372.287	403534.916	77	35.91
1	6	10^{-6}	403385.238	403385.418	119	49.40
3			403385.12	403385.4	105	45.84

Table 6.1g. Optimum design for the two loop network ($\varepsilon = 10^{-6}$).

Arc #	Segments having Length (m)	Flow (m^3/hr)	Diameter (inches)	Head Loss (m)
1	1000	1120.0	18	6.749242
2	795.408540	368.331588	10	11.982436
2	204.591460	368.331588	12	1.268322
3	1000	651.668412	16	4.393325
4	999.998776	0.975686	1	18.857409
4	0.001224	0.975686	2	0.000001
5	310.354145	530.692726	14	1.786092
5	689.645855	530.692726	16	2.071341
6	11.139149	200.692726	8	0.161574
6	988.860851	200.692726	10	4.838426
7	98.493861	268.331588	8	2.446478
7	901.506139	268.331588	10	7.553516
8	1000	-0.692726	1	-10.000004

Node (*i*)	H_{iL} (m)	$H_i + E_i$ (m)	H_{iU} (m)
1	*n*/a	210.00	*n*/a
2	180.00	203. 250758	210.00
3	190.00	190.00	210.00
4	185.00	198. 857433	210.00
5	180.00	180.000005	210.00
6	195.00	195.00	210.00
7	190.00	190.00	210.00

Table 6.2a. Arc data for the Hanoi network.

Link Index	Link Length (m)	*q*min (m^3/hr)	*q*max m^3/hr)	Link Length (m)	Link Length (m)	*q*min (m^3/hr)	*q*max (m^3/hr)
1	100	19940	19940	18	800	-5280	-2210
2	1350	19050	19050	19	400	2270	5340
3	900	4885	9455	20	2200	6475	8255
4	1150	4755	9325	21	1500	1415	1415
5	1450	4030	8600	22	500	485	485
6	450	3025	7595	23	2650	3785	5565
7	850	1675	6245	24	1230	1265	4230
8	850	1125	5695	25	1300	445	3410
9	800	600	5170	26	850	-3240	4400
10	950	2000	2000	27	300	-4140	3500
11	1200	1500	1500	28	750	-3130	4510
12	3500	940	940	29	1500	290	1920
13	800	-1925	2645	30	2000	0	1630
14	500	-2540	2030	31	1600	-360	1270
15	550	-2820	1750	32	150	-720	910
16	2730	-3070	0	33	860	-825	805
17	1750	-3935	-865	34	950	0	1630

Table 6.2b. Node Data for the Hanoi network.

Node	Supply or Demand (m^3/hr)	Node	Supply or Demand (m^3/hr)
1	19940	17	-865
2	-890	18	-1345
3	-850	19	-60
4	-130	20	-1275
5	-725	21	-930
6	-1005	22	-485
7	-1350	23	-1045
8	-550	24	-820
9	-525	25	-170
10	-525	26	-900
11	-500	27	-370
12	-560	28	-290
13	-940	29	-360
14	-615	30	-360
15	-280	31	-105
16	-310	32	-805

Table 6.2c. Computational results for the Hanoi Network using MSTR but no RLT, for various ε values.

n_s	ε	Global Lower Bound	Global Upper Bound	# LP Solved	CPU Time
4	0.05	5811286.509	6115328.661	27	6.76 sec
6		5835535.829	6138561.381	23	8.82 sec
4	0.01	6008396.288	6064585.193	83	20.56 sec
6		6006719.502	6064234.239	81	30.78 sec
4	0.001	6050065.657	6055652.912	203	51.09 sec
6		6050860.518	6056671.749	167	65.18 sec
4	10^{-6}	605536.43	605542.48	1533	245.44 sec
6		605536.31	605542.37	1075	552.82 sec

Table 6.2d. Computational results for the two-loop network using both MSTR and RLT, for various ε values.

n_s	ε	Global Lower Bound	Global Upper Bound	# LP Solved	CPU Time
4	0.05	5821341.703	6116155.285	9	10.54 sec
6		5843154.511	6125405.317	7	12.51 sec
4	0.01	6010377.119	6067704.889	39	43.87 sec
6		6004302.126	6063931.308	39	69.85 sec
4	0.001	6049649.086	6055652.912	143	161.23 sec
6		6050692.558	6056562.717	115	201.54 sec
4	10^{-6}	6055536.35	6055542.37	1500	21 min
6		6055536.332	6055542.369	757	38 min

Table 6.2e. Optimal design, flows, and pressure heads for the Hanoi network.

Node #	Head (m)	Node #	Head (m)	Node #	Head (m)	Node #	Head (m)
1	100	9	40.478285	17	32.862888	25	34.992832
2	97.140177	10	39.193423	18	49.793513	26	31.169824
3	61.663134	11	37.634118	19	58.928623	27	30.000000
4	56.958704	12	34.206805	20	50.517657	28	38.778997
5	51.127922	13	30.000000	21	35.159782	29	30.000000
6	44.986041	14	33.639216	22	30.000000	30	30.286252
7	43.548644	15	32.087264	23	44.359177	31	30.571084
8	41.839479	16	30.295907	24	38.664555	32	32.838570

Arc #	Dia	Length (m)	Flow (m^3/hr)	Head Loss (m)	Arc #	Dia	Length (m)	Flow (m^3/hr)	Head Loss (m)
1	40	100	19940	2.859823	19	24	400	2403.036233	2.734511
2	40	1350	19050	35.477043	20	40	2200	7831.762902	11.145477
3	40	900	7965.200865	4.704430	21	16	491.360368	1415	9.074382
4	40	1150	7835.200865	5.830782	21	20	1008.639632	1415	6.283494
5	40	1450	7110.200865	6.141881	22	12	500	485	5.159782
6	40	450	6105.200865	1.437398	23	40	2650	5141.762902	6.158480
7	40	850	4755.200865	1.709164	24	30	1230	3501.070442	5.694622
8	40	850	4205.200865	1.361194	25	30	1300	2681.070442	3.671724
9	30	74.233811	3680.200865	0.376960	26	20	850	1186.762902	3.823008
9	40	725.766189	3680.200865	0.907902	27	12	299.999150	286.762902	1.169823
10	30	950	2000	1.559305	27	16	0.000850	286.762902	0.000001
11	24	1200	1500	3.427313	28	12	750	83.237098	0.295907
12	24	3500	940	4.206805	29	16	1500	595.692460	5.580180
13	16	253.562337	1155.200865	3.216195	30	12	1999.999879	305.692460	8.778997
13	20	546.437663	1155.200865	2.338011	30	16	0.000121	305.692460	0
14	16	500	540.200865	1.551952	31	12	1600	-54.307540	-0.286252
15	12	550	260.200865	1.791357	32	16	150	-414.307540	-0.284832
16	12	2730	-133.036233	-2.566981	33	16	748.166501	-519.307540	-2.158642
17	16	1750	-998.036233	-16.930625	33	20	111.833499	-519.307540	-0.108844
18	20	419.803213	-2343.036233	-6.654887	34	24	950	1324.307540	2.154262
18	24	380.196787	-2343.036233	-2.480223					

Table 6.3a. Arc Data for the New York Network (without the parallel pipes).

Link Index	Link Length (ft)	Link Index	Link Length (ft)
1	3538	12	3721
2	6039	13	7350.5
3	2226.5	14	6435.5
4	2531.5	15	4727.5
5	2623	16	8052
6	5825.5	17	9516
7	2928	18	7320
8	3812.5	19	4392
9	2928	20	11712
10	3416	21	8052
11	4422.5		

Table 6.3b. Node Data for the New York Network.

Node	Supply or Demand (ft^3/sec)	Minimum Head (ft)	Node	Supply or Demand (ft^3/sec)	Minimum Head (ft)
1	2017.5	300.0	11	170.0	255.0
2	92.4	255.0	12	117.1	255.0
3	92.4	255.0	13	117.1	255.0
4	88.2	255.0	14	92.4	255.0
5	88.2	255.0	15	92.4	255.0
6	88.2	255.0	16	170.0	260.0
7	88.2	255.0	17	57.5	272.0
8	88.2	255.0	18	117.1	255.0
9	170.0	255.0	19	117.1	255.0
10	1.0	255.0	20	170.0	255.0

Table 6.3c. Computational results for the New York network using MSTR but no RLT, for various ε values.

n_s	ε	Global Lower Bound	Global Upper Bound	# LP Solved	CPU Time
4	10^{-2}	37878432	37878600	20663	87 min
6		37878430	37878619	13673	70 min
4	10^{-4}	37878432	37878600	20663	87 min
6		37878430	37878619	13673	70 min
4	10^{-8}	37878580.86	37878581.21	24137	89 min
6		37878580.83	37878581.21	16171	85 min

Table 6.3d. Computational results for the New York network using both MSTR and RLT, for various ε values.

n_s	ε	Global Lower Bound	Global Upper Bound	# LP Solved	CPU Time
4	10^{-2}	37876879	37879227	14235	266 min
6		37876198	37880733	6781	225 min
4	10^{-4}	37876879	37879227	14235	266 min
6		37876317	37879229	6822	234 min
4	10^{-8}	37878580.86	37878581.21	17695	341 min
6		37878580.83	37878581.21	9913	287 min

Table 6.3e. Optimal design for the New York network.

Link Index	Dia (inches)	Segment Length (ft)	Flow (gpm)	New Cost ($)	Head Loss (ft)
1	180	11600	879.302755	7987090.046769	5.729565
2	180	19800	786.902755	13633136.459141	7.963917
3	180	7300	694.502755	5026358.391501	2.330383
4	180	8300	606.302755	5714900.636912	2.060921
5	180	8600	518.102755	5921463.310536	1.596524
6	180	19100	429.902755	13151156.887353	2.510590
71	132	4800	201.677399	2249808.727816	0.704444
72	132	4800	201.677399	2249808.727816	0.704444
7	112	2408.899652	140.025356	920962.065953	0.400677
7	116	7191.100348	140.025356	2871547.079089	1.008212
8	132	12500	253.502755	5858876.895354	2.800706
9	180	9600	58.500000	6610005.555947	0.031515
10	204	11200	131.124553	9006418.334910	0.088965
11	204	14500	484.997245	11660095.165731	1.295005
12	204	12200	836.297245	9810562.829098	2.985471
13	204	24100	953.397245	19379882.309939	7.515528
14	204	21100	1045.797245	16967448.827374	7.808112
15	204	15500	1138.197245	12464239.659919	6.708414
161	72	13200	17.039827	2917822.789095	0.383495
162	72	13200	17.039827	2917822.789095	0.383495
16	100	26399.999870	40.460173	8769939.752862	0.766990
16	104	0.000130	40.460173	0.045337	0.000000
171	72	15600	69.403955	3448336.023476	6.090610
172	72	15600	69.403955	3448336.023476	6.090610
17	96	0.000063	164.796045	0.019895	0.000000
17	100	31199.999937	164.796045	10364474.283491	12.181220
181	60	12000	38.897986	2115833.464170	3.900627
182	60	12000	38.897986	2115833.464170	3.900627
18	76	9745.580952	78.202014	2303611.685506	3.646560
18	80	14254.419048	78.202014	3590655.926313	4.154695
191	60	7200	64.220140	1269500.078502	5.917147
192	60	7200	64.220140	1269500.078502	5.917147
19	72	0.195798	119.652552	43.280596	0.000209
19	76	14399.804202	119.652552	3403753.700483	11.834085
20	60	38400	13.872692	6770667.085344	1.853176
211	72	13200	80.619266	2917822.789095	6.799253
212	72	13200	80.619266	2917822.789095	6.799253
21	68	10595.684537	75.508042	2181889.521376	6.386791
21	72	15804.315463	75.508042	3493499.380606	7.211714

Total Cost	**Existing Cost**	**New Cost**
217,700,927	179,822,346	37,878,581

Node Index (*i*)	H_i+E_i (ft)	Node Index (*i*)	H_i+E_i (ft)
1	300.000000	14	285.483473
2	294.270435	15	293.291586
3	286.306518	16	260.000000
4	283.976135	17	272.800000
5	281.915215	18	262.801255
6	280.318691	19	255.000000
7	277.808101	20	261.853176
8	276.399212	21	277.103656
9	273.598505	22	273.183495
10	273.566990	23	268.891865
11	273.687470	24	258.900627
12	274.982475	25	267.770323
13	277.967946	26	266.799253

Table 6.4a. Arc data for the Blacksburg network.

Arc	Length (ft)	Fix Dia (inches)	Arc	Length (ft)	Fix Dia (inches)	Arc	Length (ft)	Fix Dia (inches)
(0, 1)	1363	-	(6, 7)	95	-	(15, 17)	1009	-
(0, 4)	194	-	(7, 8)	419	-	(18, 19)	408	-
(1, 2)	1832	-	(7, 29)	208	-	(18, 24)	1181	-
(1, 8)	151	-	(8, 30)	110	-	(21, 20)	113	-
(2, 3)	888	-	(9, 10)	451	6	(22, 13)	701	6
(2, 11)	155	-	(9, 12)	59	-	(23, 22)	351	6
(2, 13)	309	-	(9, 25)	416	6	(24, 19)	967	6
(3, 17)	699	-	(11, 15)	303	12	(26, 9)	271	-
(3, 18)	1151	-	(12, 11)	823	-	(26, 27)	317	6
(4, 5)	1098	-	(13, 14)	766	8	(26, 28)	424	6
(5, 6)	578	-	(14, 23)	382	-	(29, 26)	730	6
(5, 21)	611	6	(15, 16)	758	10			-

Table 6.4b. Node data for the Blacksburg network.

Node Index	Supply or Demand (gpm)	Elevation (ft)	Node Index	Supply or Demand (gpm)	Elevation (ft)
0	1548.63	2163	16	-52.11	2149
1	-52.11	2141	17	-10.38	2109
2	-50.58	2132	18	-103.65	2144
3	-25.77	2121	19	-52.11	2149.5
4	-13.84	2153.5	20	-10.96	2140
5	-53.65	2141.5	21	-51.35	2141.5
6	-51.73	2129	22	-11.73	2144
7	-200.58	2127	23	-51.54	2156.5
8	-11.35	2127	24	-102.50	2178
9	-10.77	2109.5	25	-51.54	2118
10	-52.11	2121	26	-52.50	2099.5
11	-100.77	2139	27	-50.96	2102
12	-27.11	2110	28	-25.58	2098.5
13	-100.77	2136.5	29	-41.92	2120
14	-25.77	2143	30	-51.35	2123
15	-51.54	2144.5			

Table 6.4c. Computational results for the Blacksburg network using MSTR, but no RLT, for various ε values.

n_s	ε	Global Lower Bound	Global Upper Bound	# LP Solved	CPU Time
4	10^{-2}	577010	577266	3279	10 min
6		576598	577170	2851	11 min
4	10^{-4}	577022	577080	4215	15 min
6		577026	577073	3850	17 min
4	10^{-6}	577066.1	577066.7	5331	18 min
6		577066.3	577066.7	4909	22 min

Table 6.4d. Computational results for the Blacksburg network using MSTR and RLT, for various ε values.

n_s	ε	Global Lower Bound	Global Upper Bound	# LP Solved	CPU Time
4	10^{-2}	576736	577306	2673	48 min
6		575081	577281	2321	81 min
4	10^{-4}	577029	577082	2814	74 min
6		577024	577081	3035	127 min
4	10^{-6}	577066.1	577066.7	4465	103 min
6		577066.3	577066.7	4011	142 min

Table 6.4e Optimal design for the Blacksburg network.

Arc	Dia (inches)	Length (ft)	Arc	Dia (inches)	Length (ft)
13,14	8.0	766	18,24	10.0	0.001887
23,22	6.0	351	18,24	12.0	1180.998113
26,28	6.0	424	7,29	10.0	208
14,23	4.0	63.645446	9,25	6.0	416
14,23	6.0	318.354554	9,10	6.0	451
9,12	10.0	59	26,9	6.0	92.889470
2,3	16.0	888	26,9	8.0	178.110530
3,18	16.0	1151	22,13	6.0	701
12,11	12.0	823	0,4	16.0	194
11,15	12.0	303	21,20	2.0	12.848676
15,16	10.0	758	21,20	3.0	100.151324
0,1	24.0	1363	1,8	3.0	43.747654
2,11	12.0	69.689856	1,8	4.0	107.252346
2,11	16.0	85.310144	5,6	12.0	577.999064
2,13	10.0	309	5,6	16.0	0.000936
24,19	6.0	967	26,27	6.0	317
8,30	4.0	0.000108	29,26	6.0	730
8,30	6.0	109.999892	4,5	16.0	1098
1,2	20.0	13.727890	5,21	6.0	611
1,2	24.0	1818.272110	3,17	1.0	698.988391
15,17	3.0	714.434858	3,17	2.0	0.011609
15,17	4.0	294.565142	6,7	12.0	95
18,19	10.0	408	7,8	4.0	419

Node #	Head (ft)	Node #	Head (ft)
0	184.660000	16	93.308922
1	168.722421	17	46.160000
2	134.430362	18	90.189925
3	129.949917	19	80.263327
4	185.105409	20	46.160000
5	148.536939	21	72.661969
6	99.580936	22	66.147108
7	94.038161	23	46.160000
8	51.706685	24	46.160000
9	97.880803	25	53.029402
10	46.160000	26	75.785923
11	112.829253	27	46.160000
12	102.771162	28	66.662345
13	104.838176	29	97.379870
14	80.826940	30	46.160000
15	103.426382		

Table 6.4f. Comparative results between RLT and non-RLT based bounding procedures (using MSTR) for $\varepsilon = 10^{-3}$.

Network	Nodes (RLT)/Nodes (No RLT)	Time (RLT) / Time (No RLT)
Two Loop	97/99 = 0.98	33/8 = 12.5
Hanoi	115/167 = 0.69	167/51 = 3.27
New York	6871/13673 = 0.50	255/71 = 3.59
Blacksburg	2734/3445 = 0.81	51/12 = 4.25

Table 6.4g. Comparative results between RLT and non-RLT based bounding Procedures (without using MSTR) for $\varepsilon = 10^{-3}$.

Network	Nodes (RLT)/Nodes (No RLT)	Time (RLT) / Time (No RLT)
Two Loop	521/887 = 0.59	109/61 = 1.79
Hanoi	2905/15681 = 0.19	49/65 = 0.75