OPTIMAL URBAN WATER DISTRIBUTION DESIGN

by

David Rhys Morgan

A thesis
presented to the University of Manitoba
in partial fulfillment of the
requirements for the degree of
MSc in Civil Engineering
in
Department of Civil Engineering

Winnipeg, Manitoba, 1983

OPTIMAL URBAN WATER DISTRIBUTION DESIGN

by

David Rhys Morgan

A thesis submitted to the Faculty of Graduate Studies of
the University of Manitoba in partial fulfillment of the requirements
of the degree of

MASTER OF SCIENCE

ABSTRACT

A model for the least cost layout and design of an urban looped water distribution has been developed. The technique couples the Hardy Cross network solving technique with a linear programming based pipe diameter modification method. An algorithm is used to assign different weights to pipes in the system according to the effect that their modification will have on the pressure at each node. A large number of demand patterns can be considered simultaneuosly during the optimization. Three design examples, including an expansion of an existing system are analysed and the results compared to previous studies.

ACKNOWLEDGEMENTS

The author would like to thank Dr. I.C. Goulter for his ideas and guidance in the development of this thesis. Other people who assisted were, A.J. Kettler for insights into the problem and F.K. Tai for his advice in the many aspects of computer operation needed during this project. This research was supported in part by a Natural Science and Engineering Research Council Postgraduate Scholarship to the author.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter I

INTRODUCTION

This thesis studies the problem of optimizing a looped urban water distribution network. This problem involved finding the least cost network which has the flexibility to perform adequately under various adverse conditions, such as pipes breakage and fire flow conditions.

The American Water Works Association (AWWA) estimates that $15 billion will be invested in water facilities in the coming decades. In fact, a recent issue of the Journal of the AWWA is dedicated to economic considerations in water supply and distribution. The Economic Research Committee of the AWWA has found that the problem of optimal design of looped networks is virtually unsolved(Lauria [1983]). It was these same concerns which prompted this study into optimal design of looped networks.

There is also a shortage of research which pertains to uncertainties occurring in water distribution operation. Fire flows are uncertain in their quantity and location , and pipe breakage is an additional uncertaintity. For many communities it is these uncertaintities, rather than steady state maximum day flows, which are the determining factors in system design.

The problem of designing a looped system is directly linked to uncertainties in the design and the operation of the system. A looped network cannot be optimized under steady state conditions and remain looped, since the network will be driven to branched system. The following thesis will study the problem of urban water distribution system optimization by addressing the uncertainties in design and operation of these systems.

In this method the Hardy Cross netwrk solving technique is coupled with a linear programming based pipe diameter modification technique. The results from each technique are passed to the other in an iterative process until an optimal solution is converged upon. The problem of flexibility in the design is dealt with explicitly by considering a large number of adverse conditions during the optimization. An algorithm is developed which assigns different weights to each pipe in the system according to how its modification effects the pressure at each node. The method is then tested in three design examples. The first example demonstrates the shortcomings of using a single steady state demand pattern in the design of a looped system. The second example considers the same system, however, many demand patterns are considered during the design. A final example considers the expansion of an existing and the results are compared to previous studies. It should be noted that the solutions obtained by this model may not be the true mathematical optimal, however, for the practical problem this model has obtained less expensive solutions more efficiently than previuos models.

Chapter II

OBJECTIVES

This study investigates new methods of designing the layout and pipe sizes of urban water distribution systems while considering many of the rigorous criteria described below:

1. The system must deliver a certain flow at a specified pressure to any node in the system when one of the key pipes in the system is not functioning. Therefore at least two independent and adequate paths from a source to each node must be provided.
2. The system must deliver severe fire flow demands at adequate pressures. While these fire flow demands occur infrequently at different nodes in the system, they may, however, may be the limiting factor in the design of systems.
3. Combinations of 1) and 2)
4. The methods should be efficient enough to be applied to large systems.
5. The techniques used to solve the problem should be widely available.
6. The method should be applicable to expansion of existing systems as well as design of new systems.

7. The method should incorporate a realistic cost function, preferably using standard unit costs as given by suppliers, i.e. cost per length.

Chapter III

LITERATURE REVIEW

An overview of systems analysis of water distribution networks was given by de Neufville et al. [1971]. In this overview systems analysis was studied as being an integrated part of the design process. The steps of the process were described by "five fundamental steps: (1) Definition of objectives; (2) formulation of measures of effectiveness; (3) generation of alternatives; (4) evaluation; and (5) selection."[1] It was stressed that of the five process steps only evaluation could be done by systems analysis while the other four are dependent upon the judgement of the engineer.

Many of the previous techniques used in water distribution system optimization were based upon a technique developed by Karmeli, Gadish and Meyers [1968]. In this paper linear programming was used to optimize branched water distribution systems in which the flow in each section was predetermined. Since this technique was the basis for much of the subsequent research it is reviewed.

---

[1] de Neufville et al. "Systems Analysis of Water Distribution Networks. _Journal of the Sanitary Engineering Division_. December,1971. pg. 825.

The loss of energy as described by the Hazen-Williams formula was used in the form:

$$J=1.13\times10^{12}\ Q^{1.852}\ C^{-1.852}\ D^{-4.87} \qquad \text{Eq.(1)}$$

where

J = friction loss in metres per kilometre
Q = flow in cubic metres per hour
D = pipe diameter in millimetres
C = friction factor

There are G(n) different diameter pipes to be considered in each section (n). It is clear that since the flow in each section was known each pipe j had a specific head gradient which could be calculated using the Hazen-Williams formula. Thus, by multiplying the head gradient by the length of each pipe (X_{nj}) and summing for all G(n), the total friction loss in each section was calculated.

The cost of the pipe in each section n was equal to the sum of the lengths of each pipe j times the unit cost of that pipe (k_{nj}). The objective function was therefore written

$$\text{Minimize } Z = \sum_{n=1}^{N} \sum_{j=1}^{G(n)} X_{nj} k_{nj} \qquad \text{Eq.(2)}$$

An important constraint upon the system was that the pressure at each node must always be maintained above the minimum allowable pressure. This was done by ensuring that the sum of pressure losses in the pipe sections along the path between the reservoir and the particular node is equal to or less than the allowable pressure difference.

$$\sum_{n \in p}^{N} \sum_{j=1}^{G(n)} X_{nj} J_{nj} \leq E_0 - E_n \quad \forall n \qquad \text{Eq.}(3)$$

where

p = the set of paths from 0 to n.

E_n = the minimum allowable pressure at demand point n.

E_0 = the reservoir pressure.

The total length of pipe in each section must equal the length of that section. A length constraint was therefore needed.

$$\sum_{j=1}^{G(n)} X_{nj} = l_n \qquad \forall n \qquad \text{Eq.}(4)$$

Using this formulation and selecting a group of eligible pipe diameters for each section, the least cost solution can be obtained.

More recently, Bhave [1979] has indicated that since at most two different diameters are choosen in any one link the number of eligible pipe diameters could be reduced. He devised a method of selecting these two pipe diameters before the optimization proceeded.

Alperovits and Shamir [1978] expanded upon Karmeli et al. [1968] to include looped systems. The flow in each section was initially assumed and an additional constraint was needed to ensure that the summation of headlosses around each loop must be zero. This kept the problem hydraulically consistent during the iterative solution procedure.

The dual variables associated with the loop constraints were used in a gradient technique to adjust the flows in a manner which will reduce the overall cost. Using an iterative process the problem was rerun with the new flows until the solution was believed to have converged upon optimality.

In this paper Alperovits and Shamir noted the importance of designing for multiple demand patterns and presented a method of expanding the problem to consider multiple design loadings. Unfortunately the number of constraints increases rapidly with the number of demand patterns, making it impossible to design for all the severe possibilities which might occur within urban water distribution systems.

In their discussion of Alperovits and Shamir [1978], Quindry et al. [1979] noted a deficiency in the original formulation and provided the corrective measures necessary for the application of the gradient technique. This new technique took into account the dual solution of the path constraints. While this correction lead to a cheaper solution, it did however reduce to a solution in which a branched layout, based on the paths chosen for headloss constraints, formed a primary network with minimum allowable pipe diameters spanning the primary links.

It can also be shown that a cheaper solution is obtained if different paths are chosen for headloss constraints(see Appendix A). The cause of the dependence of the solution on choice of the paths in this formulation was due to an implicit assumption. Under this assumption it was assumed that a change in the resistance (pipe diameter) in a section of designated path has a one to one relationship with the change in pressure at the end of that path. This is true in a branched system because all of the flow going to the end node passes through the designated sections. In a looped network, however, there is more than one path to the end node. Consequently the effect of changing the resistance in one of these paths has less effect on the head at the end node than than it would if it was the only path. This issue will be discussed in more detail during the development of the model described in this thesis.

The linear programming approach was used in a different manner by Schaake and Lai [1969] to optimize looped water distribution systems. The decision variable in this model was derived from the Hazen-Williams formula and was proportional to the flow in each section. Hydraulic consistency was maintained by assuming an initial head at each node rather than the initial flows in the links. A continuity constraint was used to insure that the summation of flow at each node was zero. The cost function was not linear but was assumed to be a concave polynomial. While this polynomial function was not directly usable in linear programming, piecewise linearization was utilized to permit inclusion of the values in the model. A method to consider multiple loadings (up to three) was also considered. The expansion of the New York City water distribution system was used as an example to develop and demonstrate the model. There are however a number of shortcomings to this procedure. Since the head of each node was fixed it was obvious that the solution is far from optimal. Variations in the initial head pattern may provide a more 'optimal' answer.

Quindry et al. [1981] expanded upon this method by adding a gradient technique using the dual solution to adjust the head at each node. A technique was then developed to solve for the optimal flows, update the pressures, and re-iterate until no major improvement was realized. This method resulted in a less expensive alternative than Schaake and Lai's solution (63.5 million vs. 78 million). However this is far greater than the solution obtained by Gessler [1982] (41.8 million) or by the model developed in this thesis (38.8 million) (see section 5.4).

Morgan and Goulter [1982] used two linked linear programming models to determine the least cost layout and design of a water distribution system. A redundancy constraint was added to Schaake and Lai's model to ensure that at least two pipes connect every node. In this method a minimum pipe size was not required and uneconomical pipes could by eliminated from the layout. Since one linear program changed heads given flows and the other linear program changed flows for given heads the two could be interfaced to work toward a better solution. Theoretically multiple loadings could be considered, although like the previous models the additional multiple loading constraints made the problem too cumbersome to solve effectively.

Kettler and Goulter [1983] have addressed a different aspect of reliability. By analysing pipe breakage data the statistical probabilities of failures per kilometre for different diameter pipe was calculated. The probability of failure of pipes in the paths from the demand points to the source were constrained to increase the reliability of the system. This study pointed out that small diameter pipes, although lowering the cost of the system, decrease the reliability of the system.

Kally [1972] used a linear programming approach similar to Karmeli et al. [1968]. Initially a design was specified and flows and heads were solved by using the Hardy Cross method. In this model the decision was the length of pipe changed from the present diameter to either a larger or smaller diameter. If there was excess pressure at some nodes then certain pipes would be changed to smaller diameters. It was assumed that changing a length of pipe from one diameter to the next would result in a proportional (linear) head change at a junction. It was noted that in a looped system this linearity would apply only approximatley. In a branched system, however, it would be accurate. The new design was then resolved by the Hardy Cross analysis. This iterative procedure was continued for several iterations until a satisfactory solution was reached. While the results from two examples were given, one for a branched system and one for a looped system, the procedure was difficult to duplicate due to inadequate descriptions of the model and the formulation.

Another operations research technique, dynamic programming has also been used to design water distribution systems and other hydraulic networks (Mays et al.[1976]). This method has the advantage of being a nonlinear logical optimization method. A more realistic cost function (actually cost tables) can therefore be used. Since the method requires the problem to be serialized into discrete stages it is particularly appropriate for pipelines (Liang [1971],Martin [1980]) and branched networks such as irrigation systems (Yang et al.[1975]). This requirement however makes dynamic programming extremely difficult to adapt to a looped water distribution system.

Methods other than linear and dynamic programming have also been used in many studies. One of the earlier attempts at optimizing a looped water distributon system was by Jacoby [1968]. The problem was defined subject to the laws of hydraulic consistency. The pipe cost function was represented by a nonlinear continuous cubic function and pumping costs were also included. A merit function was used to move toward optimality in which a cost penalty was assigned to any constraint violation. The flows and pipe diameters were then changed to reduce the sum of the cost of the system and penalty costs. This model recognized the probability of reaching local minima and therefore suggested the use of many starting points.

Deb and Sakar[1971](and Deb[1974,1976]) have studied many aspects of optimal hydraulic network design using differrent techniques. In one paper (Deb and Sakar [1971]), a new method was developed based on an equivalent diameter concept from which the minimum cost of pipe was obtained for a particular pressure surface and reservoir height. The optimal pressure profile was found for a particular network and the cost was empirically determined as a function of pipe size , service reservoir height and horse power of pumps. An example with a single source and a steady state condition was given. There was no reference to the incorporation of multiple sources or demand patterns into the design procedure.

In a later paper Deb [1974] developed another method to achieve least cost design of branched pipe networks. The method could be used with the help of a desk calculator. In a third paper looped systems were again analysed(Deb[1976]). In this paper it was recognized that without suitable constraints the system would deteriorate to a branched system. To overcome this problem minimum pipes were specified. The location and elevation of the single reservoir was considered along with energy costs during the optimization procedure.

The optimization method used in all three papers by Deb (Deb and Sakar[1971],Deb[1974,1976]) analysed the derivative of the cost function with respect to the change in diameter, while satisfying constraints which represent the minimum flows and pressures at the nodes. Since the derivative was needed it was assumed that the cost function is continuous.The nearest commercially available pipe diameters were selected following the analysis. The continuous function used was first derived by Linaweaver [1964] for large scale pipe and tunnels. Unfortunately, it is unlikely that the cost function for small discrete pipes will follow this continuous polynomial function.

Shamir [1974] proposed a method of optimizing both design and operation of water distribution systems. A newly developed optmization technique was combined with a previous flow solving model (Shamir and Howard [1968]) to produce a new procedure. The objective function was constructed to select the optimal changes in an existing system or an existing design of a new system. Decision variables associated with operation (pump and valve settings) as well as design variables (pipe diameters and pump capacities) were considered for up to five different demand patterns. The problem was solved using a generalized reduced gradient method which used Lagrangian multipliers to construct the gradient vectors. These gradient vectors were used to 'step' towards an optimal solution while maintaining the hydraulic consistancy. A large network (20 nodes, 50 pipes) was solved to illustrate this method.

Watanatada [1973] used a more complex quadratic function to define the costs. The nonlinear function representing the system, along with constraints, was defined and solved using the Lagrangian function. Both capital and pumping costs were considered during the solving of the system for a single steady state demand pattern. Minimum diameters were specified to prevent the system from deteriorating to a branched system. A major strength of this method is that it was powerful enough to be used on a large system (50 nodes).

Rasmusen [1976] has used a heuristic technique to converge upon an optimal solution. The technique operated in a two stage fashion. The hydraulic flow problem was solved initially and then the pipe diameters were modified to reduce the cost. These two problems were linked and solved iteratively until the optimal solution was found. Within the procedure itself nodes with pressures below the minimum allowable were labelled critical nodes. All pipes in which flow may travel from the source to this node were labelled critical links. These critical links have a potential for saving energy by a reduction in pressure losses. Noncritical pipes could be reduced to save capital costs. The relationship between energy costs and capital costs dictated the degree to which pipe diameters were reduced or increased. The fact that a number of paths from source to node have an important effect on the pressure at the critical node is an important concept in designing looped water distribution systems. This concept which has been recognised by Rasmusen has been neglected in many other studies (Bhave [1978], Alperovits and Shamir [1978]). While only a single source and a single steady state demand pattern was considered during the design, Rasmusen indicated that further work would enable the model to consider multi-source networks.

Bhave [1978] suggested a simple technique which can be used without the aid of a computer. The procedure found the optimal spanning tree,which were then termed primary links, with redundant pipes connecting the end nodes to create looping. While energy costs were considered,only a single source and a single steady state demand pattern were used.

Gessler [1982] has used a complete enumeration approach to arrive at an optimal solution. Alternative designs were generated and tested. The first test calculated the cost of the new alternative and checked it against the best previous cost. If it failed this test, i.e. was more expensive, then a new alternative was generated. If it passed this test then the network was solved to check if it performed adequately. If this second test was failed then the alternative was discarded. If the test was passed then this alternative became the new best soluton. New alternatives were generated and this was continued until the choices were exhausted.

This method is very time consuming and would presently be impracticable on a system of any complexity. The method, however, was made more usable by the procedure of "grouping". By grouping certain pipes into sets (at the discretion of the design engineer) and specifying that all pipes in each set must be the same size, the number of alternatives was greatly reduced. Although this technique ensures that the solution is a global optimum for the specified groupings , there is no guarentee that the groupings are optimal. Thus the true global optimum may not be reached.

This technique has been applied successfully to the New York City water supply problem. The solution is the best published to date being over thirty per cent cheaper than the solution generated by the Quindry et al. [1981] method. While this method is applicable to multi-source and multi-demand pattern design, the time requirements which are already quite extensive increase very rapidily with the size of the system.

The problem of designing the layout as well as the size of the components has been addressed in recent years. Mays et al. [1976] incorporated the layout and design of sewer systems into one model. A heuristic technique using discrete differiential dynamic programming (DDDP) was used in a conjunctive form with two models to form a screening model. One of these models simultaneously considers system layout and design, while the other uses the layouts of the first model to compute the optimal design.

Martin [1980] used dynamic programming to optimize a water conveyance system. In this study the layout and component size of a single path large scale system was considered. A combination of facilities (pipelines, open canals and pumping stations) were considered during the optimization. Both of these dynamic programming methods, i.e. Mays et al. [1976] and Martin [1980], were limited to serial systems and are unsuitable for looped water distribution systems.

Rowell and Barnes [1982] have addressed the problem of obtaining a layout for municipal water distribution systems directly. A two level hierachically integrated system of models was developed. An economical tree layout was first developed then redundant pipes were added. A "rule of thumb" method, which estimates pipe capacity using just the diameter, and neglecting the hydraulic gradient, was used in the model. The performance of the system was not tested by a network solver and hydraulic consistency is neglected during the optimization.

The major shortcoming of these present optimization methods is they tend to design inflexible systems. Templeman [1982] in his discussion of Quindry et al. [1981] stated

> "Optimization tends to remove redundancy, and any spare capacity which is not immediately required by the design demand pattern is optimized out. Thus, all flexibility is removed. The optimization process is not at fault here, it merely extrapolates the design process to its logical limits. Such faults are inherent in the design process itself which does not directly incorporate resilience, flexibility, and reliability into the the design process."[2]

Templeman continued that the design criteria must include the flexibility to serve firefighting demands at all nodes. Most of the papers mentioned consider only one demand pattern and thus tend to design branched systems with minimum size redundant pipes connecting the end nodes. Two papers

---

[2] Templeman,A.B. , "Discussion of Looped Water Distribution Systems" , Journal of the Environmental Engineering Division,ASCE. pg. 599, June , 1982.

Alperovits and Shamir [1978] and Quindry et al. [1981] realized that flexibility is obtained by considering more than one demand pattern (two to three patterns were considered). This however falls far short of considering a fire flow at each node in the system.

Chapter IV

MODEL DEVELOPMENT

4.1 GENERAL DESCRIPTION

To begin the optimization procedure an initial layout and design for the system is assumed. For a given demand pattern the flows in all pipes and the pressures at all nodes is then determined. Of the many methods available (see Holloway and Chaudhry [1983]) to solve this problem the Hardy Cross technique appears most suitable due to its simplicity and general widespread acceptance. The solution provided by the Hardy Cross Method is passed to a pipe design/modification procedure. If the pressures at some demand or diversion points are below minimum it is necessary to replace sections of pipe with pipe of larger diameter. This procedure determines which sections of pipe should be replaced in order to raise the pressure to the minimum at the least cost.

Conversely,if the pressures are above the minimum allowable then some section of pipe may be replaced by pipes of smaller diameter. Again the decision is which section of pipe should be replaced in order to bring about the greatest saving while maintaining minimum pressure requirements elsewhere. In general , replacement of designated pipes (designated pipes are the pipe diameters which have been initially assumed or selected as replacement pipes in previous iterations) with both smaller and larger diameters is needed since some areas may be underdesigned while other areas may be overdesigned. An overall reduction in cost may also be obtained by increasing the diameter of one pipe, therefore allowing a number of other pipes to be replaced by pipes of smaller diameter. This new configuration of pipe sizes is passed back to the network solver to calculate true flows and pressures. The process is repeated iteratively until an optimal solution is reached. This process is described in more detail in later sections.

4.2 SINGLE DEMAND PATTERN

The procedure used in multiple demand pattern design is an extension of that used for single demand pattern design. The common parts of the formulation will therefore be described in this section and the differences will be explained in the section on multiple demand patterns.

Since the powerful simplex algorithm allows complex problems to be solved efficiently, linear programming is used to select the optimal replacement sections. In order to formulate a linear programming model the terms must have linear relationships. In this model the cost function is constructed by assuming that the cost of replacing a length of one diameter of pipe with the same length of another diameter is a linear function of the length of pipe replaced. The basis of this assumption is described as follows. The unit cost of pipe is usually defined in cost per length (.ie dollars per metre). Expressing the unit cost of a pipe as C_d for a designated pipe of the **dth** diameter and C_r as the unit cost of a replacement pipe of **rth** diameter, then the unit cost of replacing pipe d with pipe r is a constant.

$$K_{dr} = C_r - C_d \qquad \text{Eq. (5)}$$

Using the definition of cost given in the above equation the objective function can be written

$$\text{Minimize } Z = \sum_j K_{jdr} X_{jdr} + K_{jds} X_{jds} \qquad \text{Eq. (6)}$$

where

K_{jdr} = the unit cost of changing a pipe of the d^{th} diameter to a larger diameter pipe r.

$$K_{jdr} > 0$$

K_{jds} = is the cost of changing a pipe of the d^{th} diameter to a smaller diameter pipe s

$$K_{jds} < 0$$

X_{jdr} & X_{jds} = the decision variable; the length of pipe d replaced by pipe r or s.

In this part of the procedure the pressure must be constrained to maintain a feasible solution. If the flow in each section remains constant, as in a branched system , then every metre of pipe replaced will cause a proportional change in pressure across the section. For example an X_{jdr} value of 2 metres will cause twice as much change in pressure across the section as an X_{jdr} value of 1 metre. This relationship can be derived from the Hazen-Williams formula

$$J=1.13 \times 10^{12} Q^{1.852} C^{-1.852} D^{-4.87} \qquad \text{Eq (7)}$$

where

J = hydraulic gradient in m/km

Q = flow in liters/second

C = Hazen-Williams friction coefficient

D = diameter in metres

The change in pressure head (ΔP)caused by replacing a section of diameter d with a section of diameter r is

$$\Delta P_{jdr}=G_{jdr}X_{jdr} \qquad \text{Eq. (8)}$$

where

$G_{jdr} = J_{jr} - J_{jd}$

J_{jd} = the hydraulic gradient in metres per kilometre for pipe diameter d in link j

All other terms are as previously described.

The change in pressure at a particular node is the sum of all changes in the flow links between that node and the source which has a fixed head. The change in pressure at each node must satisfy the minimum allowable pressure at that node. The constraint to ensure this condition can be expressed as

$$\sum_{j \in p} G_{jdr}X_{jdr} + G_{jds}X_{jds} \geq H_i - h_i \quad \text{for all } i \qquad \text{Eq.(9)}$$

where

G_{jdr} & G_{jds} = the change in pressure caused by replacing the designated pipe d with a larger pipe r or smaller pipe s

H_i = the minimum allowable pressure head

h_i = the existing pressure

All other terms are as previously described.

If the initial solution is not close to the optimal the model may try to replace more of a designated pipe than is available. It is obviously unrealistic to allow a pipe length greater than that of the link length to be replaced. The following constraints are therefore needed.

$$X_{jdr} \leq L_j \qquad \text{Eq. (10)}$$

$$X_{jds} \leq L_j \qquad \text{Eq. (11)}$$

where L_j is the length of link j.

If the designated pipe in the link is made up of two diameters (see Figure 1) then the length of pipe available to be replaced is less than the length of the link. The new constraints are

$$X_{jdr} \leq l_{1j} \qquad \text{Eq. (12)}$$

$$X_{jds} \leq l_{2j} \qquad \text{Eq. (13)}$$

where

l_{1j} = the length of designated pipe of smaller diameter

l_{2j} = the length of designated pipe pipe of larger diameter

and

$l_{1j}+l_{2j}=L_j$ (see Figure 1)

All other terms are as previously described.

If the system is branched it can be solved directly with the linear programming model as formulated. With a looped system, however, additional factors must be taken into account. If there are several paths to the demand node i from a source (or sources) with a fixed head then all paths to that node must be considered . It should be recognized that a change to one link, on one of these several paths, may not have the same importance to the change in pressure at node i as a change in another link in the same path or in one of the other paths. This condition is explained in reference to Figure 2 which shows a simple looped system. It can be seen that changes in links 1 and 4 will have a direct

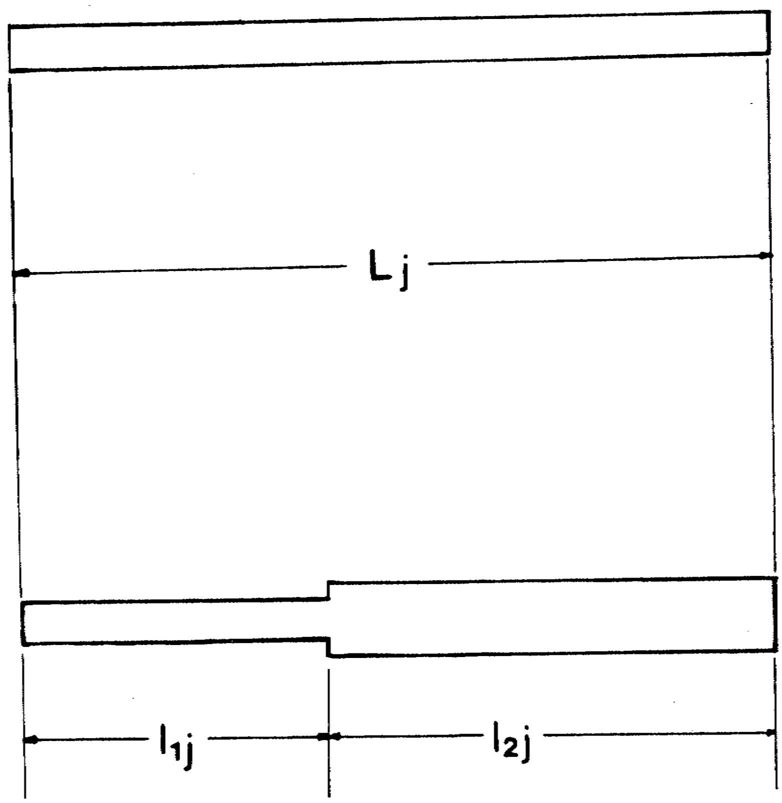

Figure 1: Definition of Length Constraints

one to one effect on the pressure at node i. The weighting attributed to link j with regard to its effect on node i is $W_{ij}=1$.

The weightings given to links 2 and 3 must be less than one since links 2 and 3 do not handle all the flow between the reservoir and the outlet separately. The method used to weight these links is to calculate the percentage of flow drawn from node i that passes through link j. For example if 65% of the flow is passing through link 2 and 35% is through link 3 then the weightings would be $W_{i2} = 0.65$ and $W_{i3}=0.35$.

Consequently, Equation 9 can be modified to consider many paths.

$$\sum_{j \in P} W_{ij} G_{jdr} X_{jdr} + W_{ij} G_{jds} X_{jds} \geq H_i - h_i \quad \text{for all } i \qquad \text{Eq.(14)}$$

where P = the set of paths from node i to a fixed head source.(each link is counted only once)

All other terms as previously described.

With a looped system, however, the linearity assumption, expressed by Equation 14, only holds approximately for the changes in pressure, since the flows in the pipes do not remain constant as pipes are changed throughout the network. This is in direct contrast to the situation which exists

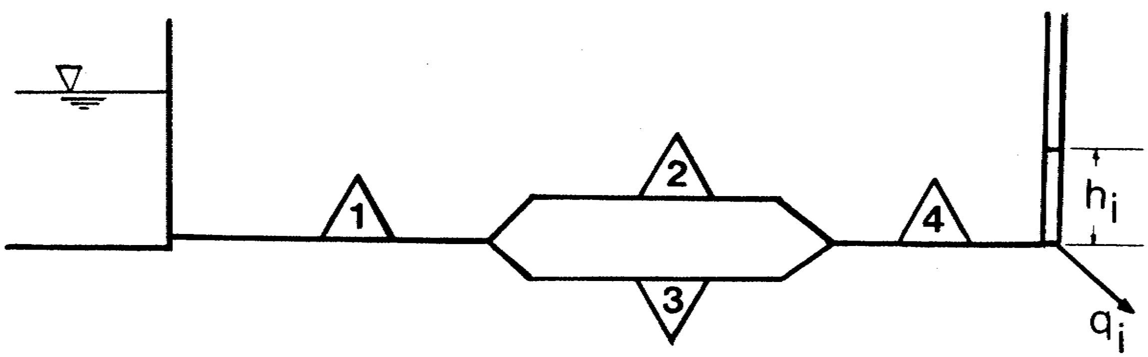

Figure 2: Simple Looped System

with branched networks. A feedback to the network solver (see Figure 3) is therefore needed to calculate the true pressures and flows. This process is repeated iteratively until an "optimal" solution is converged upon. This convergence is recognized when the change in cost becomes insignificant (0.01%) and none of the length constraints are binding.

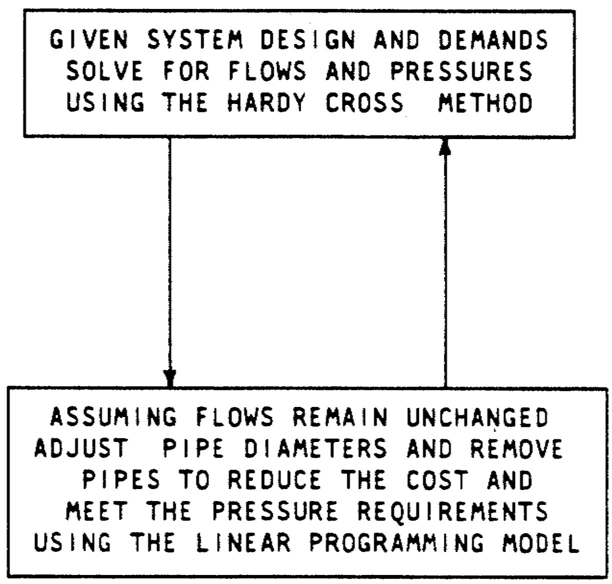

Figure 3: Schematic of Model

4.3 MULTIPLE DEMAND PATTERNS

If a system is designed for one situation , then the most economical layout will be a branched system. This is due to economies of scale , where one large pipe is a less expensive method of transporting water than two smaller pipes. However, if a single pipe fails then no flow can reach the nodes downstream of this failure. This problem is alleviated by adding redundant pipes to ensure looping. These pipes have small diameters, usually the minimum available diameter, and there is no guarantee that the secondary path can deliver water at an adequate pressure in case of a primary link failure.

A solution to this problem is to design this system while considering all the worst case scenarios; fire flows and pipes breaks. One method of doing this is to add one set of constraints for each demand pattern. Formulations for multiple demand patterns are described by Schaake and Lai [1969], Alperovits and Shamir [1978] and Quindry et al. [1981]. The problem with these formulations is that the additional constraints needed to consider multiple demand patterns increase rapidly making the problem computationally impractical. If forty constraints were needed for one demand pattern then 120 would be needed for three demand patterns.

Many of these constraints can, however, be eliminated because they are non-binding. From linear programming theory (Hillier and Lieberman [1980], p. 68-117) it is known the maximum number of non-zero variables in the solution equals the number of binding constraints. Equation (6)(p.23) states that there are two variables for each link in the system, one representing the length of pipe to be replaced by a larger diameter pipe , the other representing the length of pipe to be replaced by a smaller diameter pipe. If pressures need to be increased then the larger diameter pipe will replace the designated diameter, and if the pressures can be reduced a smaller diameter pipe will replace the designated diameter. Therefore, only one of these variables will ever be non-zero in the solution. In many cases if there is no change in the pipes of that link, both variables will be zero. Therefore the maximum number of pressure constraints needed (independent of the number of demand patterns) is equal to the number of links (N).

A further condition of this model is that when the optimal solution is found none of the length constraints will be binding.They are, however, needed initially if the first assumption is far from optimal. Since the number of these constraints is not dependent on the number of demand patterns considered they are left intact for the multiple demand patterns situation.

The pressure constraints used must be selected before the linear programming stage proceeds. The Hardy Cross stage of the procedure first solves each of the demand patterns for the pressures and flows. The amount by which the actual pressure is below (or above) minimum pressure is calculated.

$$a_{it} = H_i - h_{it} \text{ for all } i \text{ and } t \qquad \text{Eq.(15)}$$

a_{it} = the amount the actual head is below the minimum allowable head at node i demand pattern t

Starting with the largest value of a_{it} the pressure constraints are constructed.

$$\sum_{j \in P} W_{ijt} G_{jtdr} X_{jdr} + W_{ijt} G_{jtds} X_{jds} \geq H_i - h_{it} \quad \forall\, i \qquad \text{Eq.(16)}$$

where

$G_{jtdr} = J_{jtr} - J_{jtd}$

J_{jtd} = the hydraulic gradient in metres per kilometre for pipe d in link j when the flow in the pipe is Q_{jt} for demand pattern t.

All other terms have previously been described.

This construction of pressure constraints continues until N constraints have been constructed.

One aspect of this technique is the interaction between two similar methods working together to achieve optimality.

The well accepted Hardy Cross method solves the flows and pressures in an iterative manner which uses the previous solution to arrive at a better solution. The linear programming method developed modifies the previous solution to arrive at a better solution. This iterative process has been used with linear programming in previous studies by Alperovits and Shamir [1978], Quindry et al. [1981] and Kally [1972]. In the two former studies the design variable is the diameter or length of a certain diameter selected as opposed to the length of diameter to be modified used in the latter study and in this thesis. In the studies by Alperovits and Shamir[1978],and Quindry et al.[1981] therefore the simplex tableau from the previous iteration must be saved in some form to maintain efficiency. If the previous tableau is not saved, many redundant simplex iterations must be carried out at each step. This condition requires a complex simplex computer package in order to save and modify this tableau during the optimization. In Kally's technique and the one described in this thesis the previous solution is described by the zero simplex tableau. This makes it possible to use a straightfoward simplex package, while still maintaining the efficiency of the method.

4.4 WEIGHTING ALGORITHM

The weight assigned to each link j in the pressure constraint equation (Equation 16,p.33) for each node i and demand pattern t is denoted by W_{ijt}. Since i and t remain constant throughout the weighting procedure for each equation, they are omitted for clarity in this example.

$$W_{ijt} = W_j = (Q_j/I_m) \times w_m \qquad \text{Eq.(17)}$$

where

Q_j = the flow in link j

I_m = the sum of the inflow to node m

w_m = the weight of the node immediately downstream of link j.

The weight of the node m is calculated as

$$w_m = \sum_{j \in B} W_j \qquad \text{Eq.(18)}$$

where

B = the set of all outflow links from the node m.

The procedure begins at the node being constrained with m=i and w_m=1.0 . The algorithm used to calculate these weights is demonstrated by application to the simple network shown in Figure 4. Initially this network is solved for a given demand pattern and pipe configuration using the Hardy Cross technique. The Hardy Cross solution provides the flows in each pipe and the pressure at each node. Since this weighting algorithm requires only the flows in each pipe, the pipe diameters and the demands and pressures at each node are not shown in Figure 5. Only the flows in each link which are obtained from this Hardy Cross Solution, are shown in this figure. In this example the equation (Equation 16, p.33) constrains the pressure at node 5. The procedure begins immediately upstream of this node. The total flow entering this node is calculated from the summation of all inflow, i.e. links 5 and 7 (160+190=350) and the weights of links 5 and 7 are calculated from this total.

$$W_5 = (160/350) \times 1 = .46 \qquad \text{Eq.(19)}$$

$$W_7 = (190/350) \times 1 = .54 \qquad \text{Eq.(20)}$$

These weights are then assigned to the nodes upstream of the links, .i.e nodes 4 and 6. For example node 4 is assigned the weight 0.46 from the single outflow link from that node. Similarly node 6 is assigned a weight of 0.54. The process then continues upstream. Under this formulation link 8 contributes 100% of the flow to node 6 via node 7. Link 8 has its weight calculated as follows

$$W_8 = 1x(.54) = .54 \qquad \text{Eq.(21)}$$

Link 3 contributes 50% of the flow to node 4, and is therefore assigned a weight calculated thus

$$W_3 = (.5)x(.46) = .23 \qquad \text{Eq.(22)}$$

Likewise

$$W_6 = (.5)x(.46) = .23 \qquad \text{Eq.(23)}$$

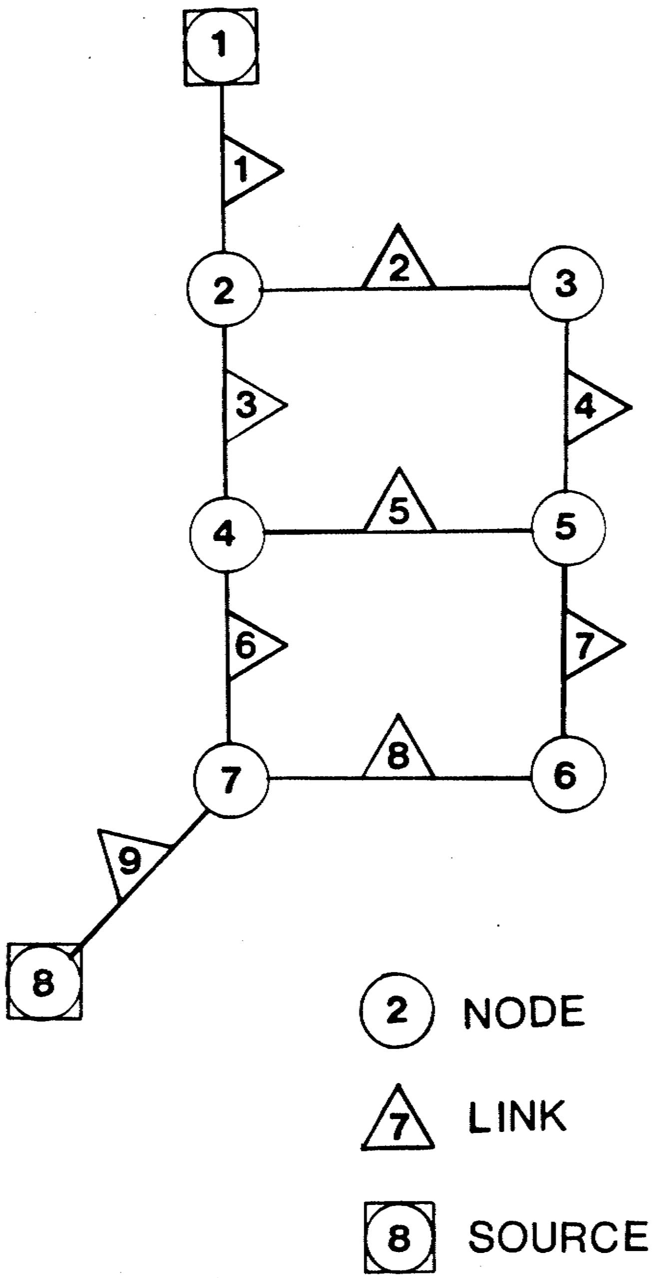

Figure 4: Network for Weighting Algorithm

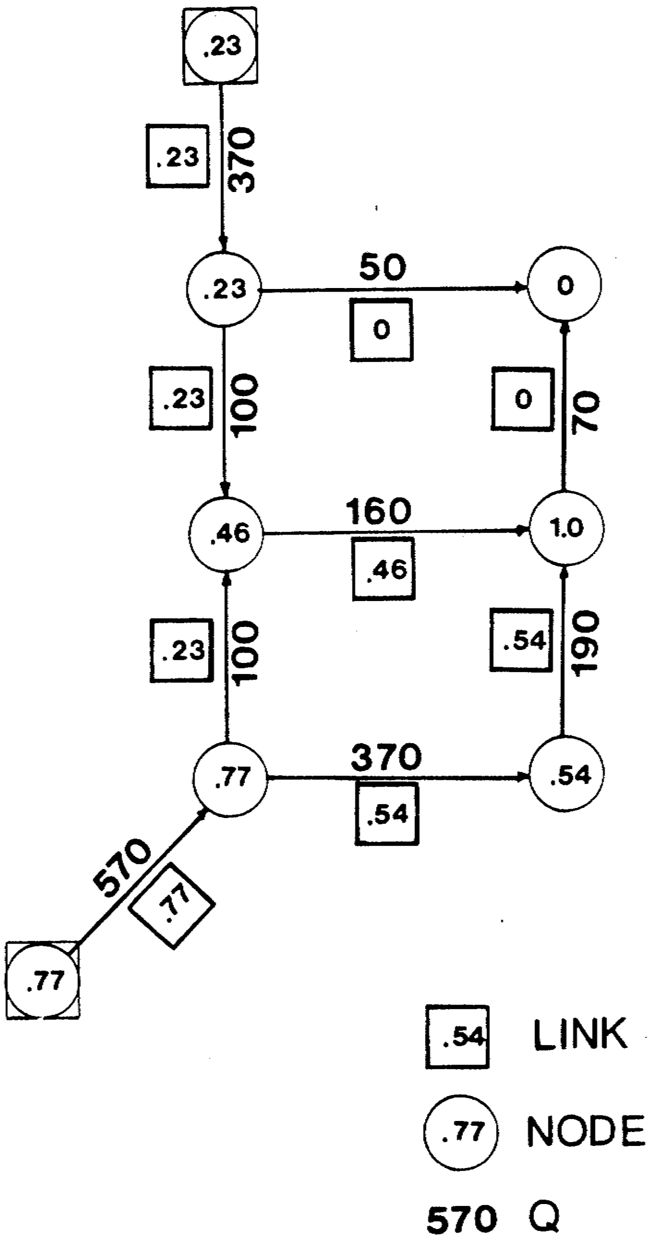

Figure 5: Flows and Weights of Network

It should be noted that the flow to node 5 from node 7 follows two distinct paths. The weighting at node 7 is therefore

$$w_7 = (.23)+(.54) = .77 \qquad \text{Eq.(24)}$$

The weights for each link and node are shown in Figure 5. Links 2 and 4 have weights of zero since none of the flow going to node 5 passes through these links.

Chapter V

USING THE MODEL IN DESIGN

5.1 INTRODUCTION

Three types of examples are illustrated in the following sections. The first example ,a large network with two reservoirs , is designed using a single demand pattern. Using one demand pattern is the standard procedure in present optimization methods. This example will demonstrate the basic approach used in the model while illustrating the shortcomings of single demand pattern design.

The second example uses the same network but with consideration for multiple demand patterns. This method will create a reliable and flexible water distribution system while eliminating pipes deemed uneconomical.

The third example uses a real problem encountered in the expansion of the New York City Water Supply Tunnels in 1969. Since this problem has been studied many times (Schaake and Lai [1969], Quindry et al. [1981] and Gessler [1982]), its solution allows the model to be compared in terms of performance and efficiency with respect to previous models. A listing of the computer program used in these examples is shown in Appendix B.

5.2 SINGLE DEMAND PATTERN DESIGN

The example network shown in figure 6 has 20 nodes and 37 possible pipe locations. The costs per metre of the available pipe sizes are shown in Table 1. The system lengths and connections are shown in Table 2 and the initial pipe size assumptions are shown in Table 3. The demands, minimum allowable pressures and initial pressure assumption for the system are shown in Table 4.

The results of the first computer run are shown in Table 5, where the final column lists the maximum weighting given to each pipe during the construction of the constraint equations. Many of the pipes have been reduced to the sub-minimum diameter and then automatically eliminated. The criteria for eliminating these pipes is that if the entire section of pipe is designated sub-minimum and the maximum weighting is less than 0.5 it is removed. The reason for its removal is that a small pipe is uneconomical due to economies of scale. A low maximum weighting also indicates that since only a small proportion of flow to any node passes through this pipe then its removal will have little effect on system performance. It can be seen that although the system tends towards branching, it is not entirely branched since many of the weightings are less than one. With a single demand pattern the most economical system should theoretically be a branched one. The model is there-

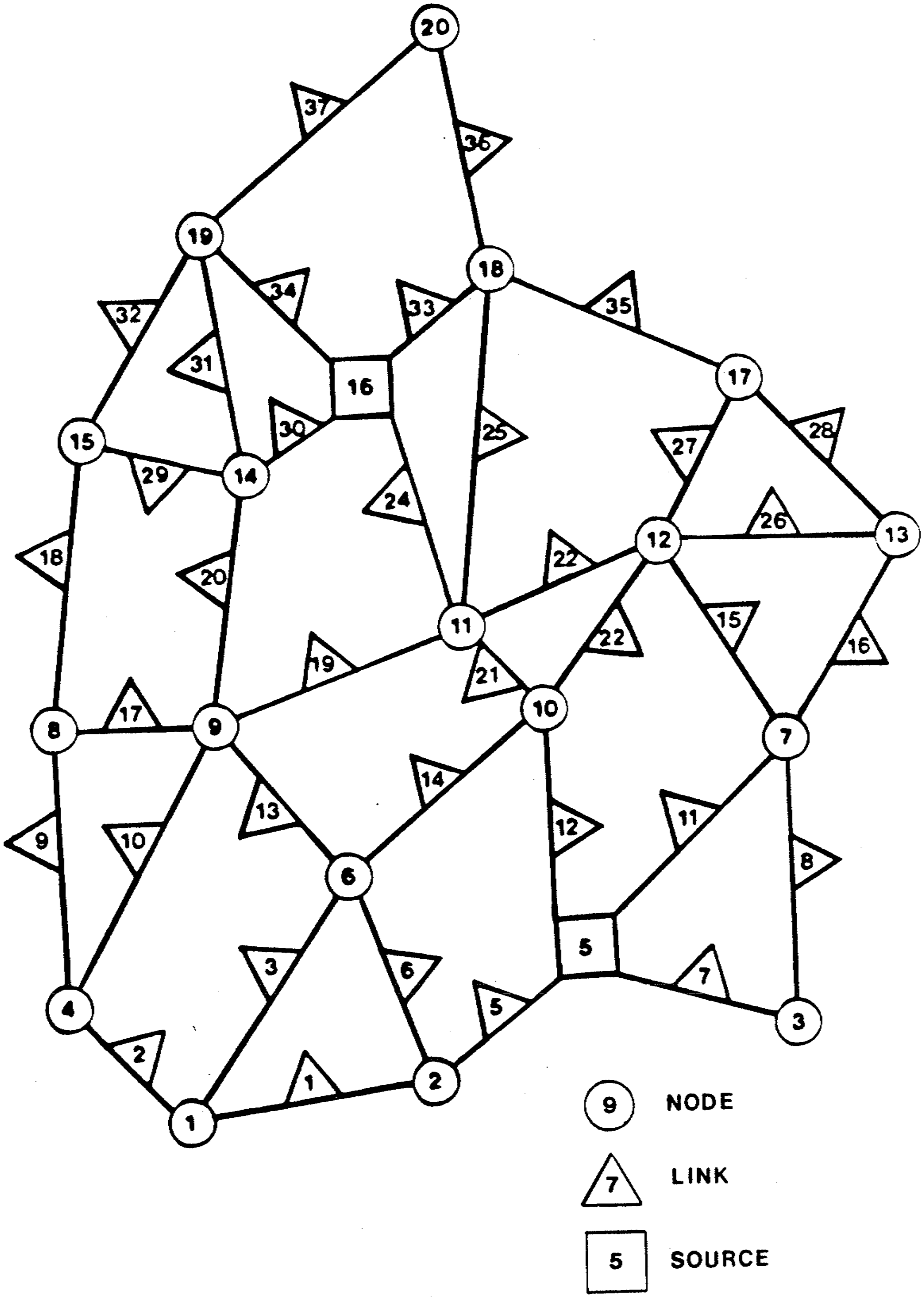

Figure 6: Possible Pipe Locations for System in Single and Multiple Demand Pattern Design

TABLE 1

Cost per Metre of Different Diameter Pipes

Diameter (metres)	Cost ($/m)
0.125	58.00
0.150	62.00
0.200	71.70
0.250	88.90
0.300	112.30
0.350	138.70
0.400	169.00
0.450	207.00
0.500	248.00
0.550	297.00
0.600	347.00
0.650	405.00
0.700	470.00

TABLE 2

Pipe Data for System

Pipe	Connecting Nodes From	To	Length (metres)
1	1	2	760.00
2	1	4	520.00
3	1	6	890.00
4	2	3	1120.00
5	2	5	610.00
6	2	6	680.00
7	3	5	680.00
8	3	7	870.00
9	4	8	860.00
10	4	9	980.00
11	5	7	890.00
12	5	10	750.00
13	6	9	620.00
14	6	10	800.00
15	7	12	730.00
16	7	13	680.00
17	8	9	480.00
18	8	15	860.00
19	9	11	800.00
20	9	14	770.00
21	10	11	350.00
22	10	12	620.00
23	11	12	670.00
24	11	16	790.00
25	11	18	1150.00
26	12	13	750.00
27	12	17	550.00
28	13	17	700.00
29	14	15	500.00
30	14	16	450.00
31	14	19	750.00
32	15	19	720.00
33	16	18	540.00
34	16	19	700.00
35	17	18	850.00
36	18	20	750.00
37	19	20	970.00

TABLE 3

Initial Pipe Size Assumption

Pipe	Diameter (m)	Length (m)
1	0.400	760.00
2	0.400	520.00
3	0.400	890.00
4	0.400	1120.00
5	0.400	610.00
6	0.400	680.00
7	0.400	680.00
8	0.400	870.00
9	0.400	860.00
10	0.400	980.00
11	0.400	890.00
12	0.400	750.00
13	0.400	620.00
14	0.400	800.00
15	0.400	730.00
16	0.400	680.00
17	0.400	480.00
18	0.400	860.00
19	0.400	800.00
20	0.400	770.00
21	0.400	350.00
22	0.400	620.00
23	0.400	670.00
24	0.400	790.00
25	0.400	1150.00
26	0.400	750.00
27	0.400	550.00
28	0.400	700.00
29	0.400	500.00
30	0.400	450.00
31	0.400	750.00
32	0.400	720.00
33	0.400	540.00
34	0.400	700.00
35	0.400	850.00
36	0.400	750.00
37	0.400	970.00

Initial Cost= $ 4,590,040.00

TABLE 4

Initial Pressure Assumptions

Node	Minimum Head (m)	Demand (l/s)	Initial Head (m)
1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

fore neglecting some aspect of reality. As stated in the model development (Chapter 4) the flow in each pipe is assumed fixed during the solution of the linear programming substage. The flow is then corrected during the Hardy Cross procedure. Since it is assumed that the flow is fixed , replacing a small pipe with the sub-minimum pipe diameter (125mm) will cause a severe pressure drop across that link. Since in reality only a minority of the flow going to the downstream node passes through this pipe it is easily rerouted causing a much smaller pressure drop. Thus the reduction and removal of this pipe becomes economical. In addition since the cost function only compares the difference in cost between two pipes it fails to take into account the basic fixed cost of putting any pipe in that position initially i.e. trenching, transport etc.. The pipe with the lowest weighting is therefore removed and the computer program is run again. Four of these computer runs, in which the pipe with the lowest maximum weighting was removed each time, were needed to arrive at an optimal solution. An AMDAHL 580 computer was used and 22.25 seconds CPU time was needed to obtain these results.

The final configuration is shown in Figure 7 with the pipe sizes given in Table 6. As expected the cost has been minimized. However as Templeman[1982] stated the flexibily has been optimized out of the system. This can be clearly seen in Figure 7, since the system has deteriorated into two

TABLE 5

Results of First Run for Single Demand Pattern Design

Pipe	Diameter (m)	Length (m)	Diameter (m)	Length (m)	Maximumg Weighting
1	0.200	37.75	0.250	722.25	1.000
2	0.200	520.00	-	-	1.000
3	-	-	-	-	-
4	-	-	-	-	-
5	0.300	610.00	-	-	1.000
6	0.150	418.38	0.200	261.62	1.000
7	0.125	103.15	0.150	576.85	1.000
8	-	-	-	-	-
9	-	-	-	-	-
10	-	-	-	-	-
11	0.200	442.59	0.250	447.41	1.000
12	0.200	648.56	0.250	101.44	1.000
13	-	-	-	-	-
14	-	-	-	-	-
15	-	-	-	-	-
16	0.200	680.00	-	-	1.000
17	0.200	480.00	-	-	0.679
18	0.150	860.00	-	-	0.321
19	0.200	800.00	-	-	0.661
20	0.150	770.00	-	-	0.339
21	-	-	-	-	-
22	0.200	620.00	-	-	1.000
24	0.250	790.00	-	-	0.661
25	-	-	-	-	-
26	-	-	-	-	-
27	-	-	-	-	-
28	-	-	-	-	-
29	0.200	500.00	-	-	1.000
30	0.250	291.45	0.300	158.55	1.000
31	-	-	-	-	-
32	-	-	-	-	-
33	0.250	540.00	-	-	1.000
34	0.150	398.18	0.200	301.82	1.000
35	0.150	181.01	0.200	668.99	1.000
36	0.150	240.60	0.200	509.40	1.000
37	-	-	-	-	-

Cost = $ 1,026,979

separate branched systems. A pipe failure anywhere in the system will the cause water supply to be stopped in some part of the system. Clearly this configuration is not suitable for an urban water distribution system, therefore the criteria by which the system is designed must be changed.

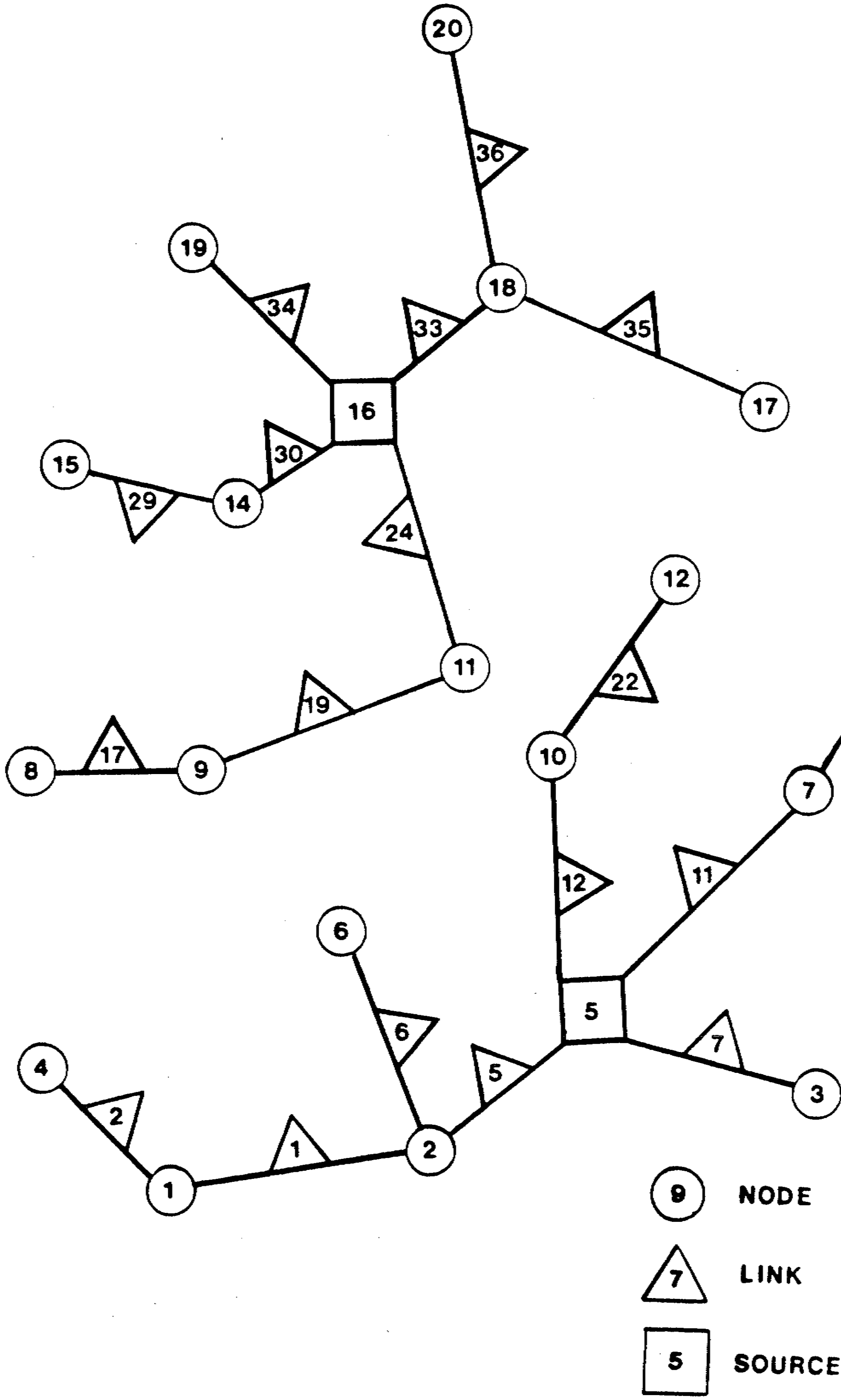

Figure 7: Single Demand Pattern Design Layout

TABLE 6

Results of Final Run for Single Demand Pattern Design

Pipe	Diameter (m)	Length (m)	Diameter (m)	Length (m)	Maximum Weighting
1	0.200	38.75	0.250	721.25	1.000
2	0.200	520.00	-	-	1.000
3	-	-	-	-	-
4	-	-	-	-	-
5	0.300	610.00	-	-	1.000
6	0.150	418.78	0.200	261.22	1.000
7	0.125	103.15	0.150	576.85	1.000
8	-	-	-	-	-
9	-	-	-	-	-
10	-	-	-	-	-
11	0.200	442.70	0.250	447.30	1.000
12	0.200	650.53	0.250	99.47	1.000
13	-	-	-	-	-
14	-	-	-	-	-
15	-	-	-	-	-
16	0.200	680.00	-	-	1.000
17	0.200	322.15	0.250	157.85	1.000
18	-	-	-	-	-
19	0.250	800.00	-	-	1.000
20	-	-	-	-	-
21	-	-	-	-	-
22	0.200	620.00	-	-	1.000
23	-	-	-	-	-
24	0.300	790.00	-	-	1.000
25	-	-	-	-	-
26	-	-	-	-	-
27	-	-	-	-	-
28	-	-	-	-	-
29	0.200	500.00	-	-	1.000
30	0.200	400.23	0.250	49.77	1.000
31	-	-	-	-	-
32	-	-	-	-	-
33	0.250	540.00	-	-	1.000
34	0.150	398.18	0.200	301.82	1.000
35	0.150	181.01	0.200	668.99	1.000
36	0.150	240.60	0.200	509.40	1.000
37	-	-	-	-	-

Cost = $ 950,230

5.3 MULTIPLE DEMAND PATTERN DESIGN

In this design example the pipe costs and system configuration remain the same as the previous example(see Tables 1 & 2). The initial assumption for pipe sizes also remains the same as in Table 3. This system is, however, designed for many situations simultaneously. The criteria of flexibility selected was that the system must give adequate pressure when one of each possible pipes is out of order and a fire flow is present at the worst possible location. This worst location will vary with the pipe that is assumed broken and should be picked by the design engineer. For this example the location and magnitude of each of these fire flows is given in Table 7. From this table and Table 4 , thirty seven different demand patterns are generated (see Appendix C). It should be noted that the demand is not evenly distributed throughout the system and the fire flow demands vary in magnitude depending upon their location. These patterns are used, as described in section 4.2, to construct the pressure constraints.

The results of the first run are shown in Table 8 (An example of the output for this type of computer run is given in Appendix C). The final column gives the maximum weighting assigned to each pipe during the construction of the constraints. The same criteria as stated in the previous example is used to remove pipes, i.e. a small pipe with the lowest maximum weighting is removed first. This low maximum weighting indicates that removal of this pipe will do the least damage to the pressure profiles. The program is then rerun and if the solution is cheaper this result is the new best result. This process continues until the cost increases or until all the maximum weightings are greater than 0.5. The final results are shown in Figure 8 and Table 9. To obtain these results six computer runs requiring 6 minutes and 4.6 seconds of CPU time were needed.

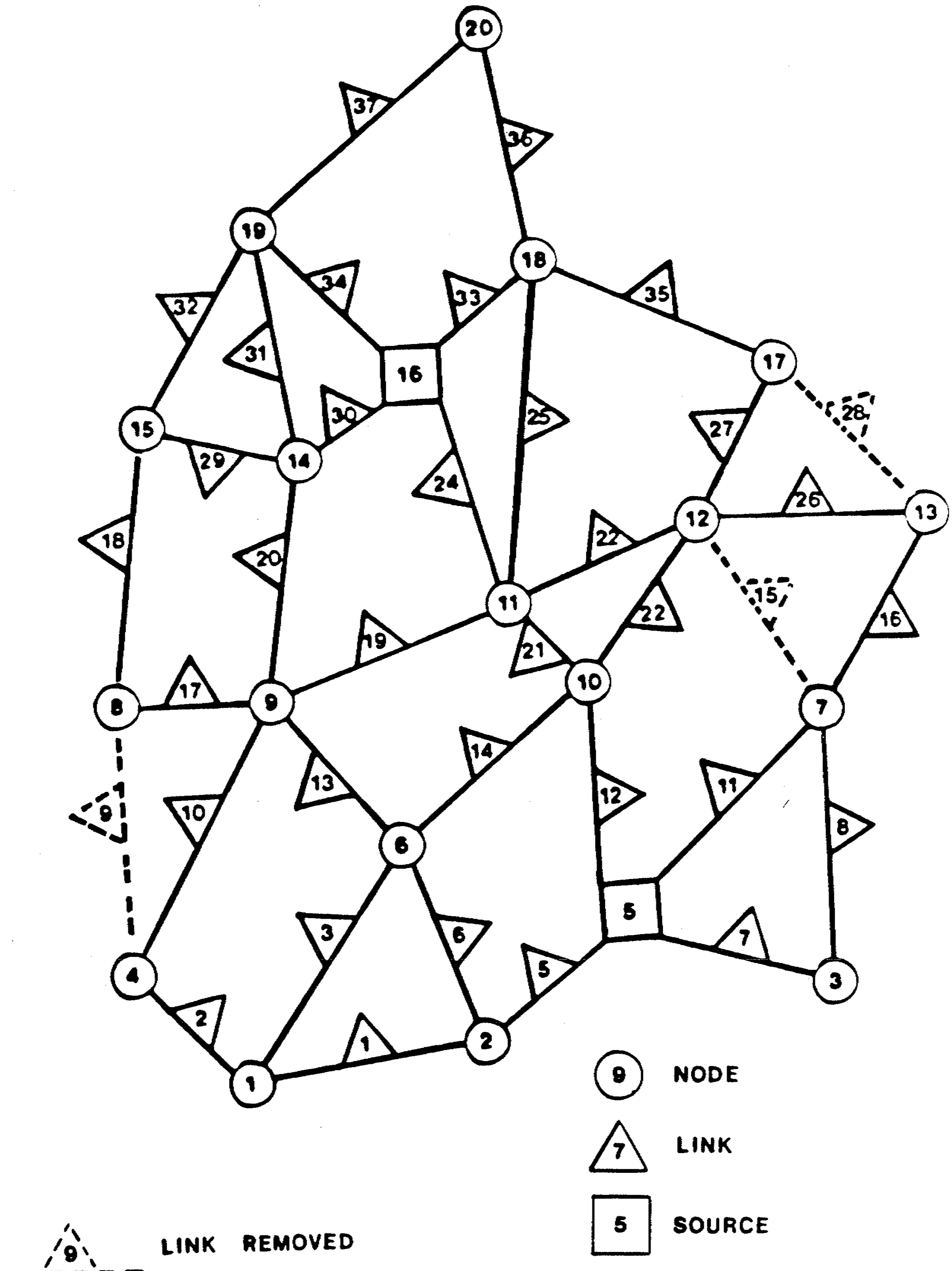

Figure 8: Multiple Demand Pattern Design Layout

TABLE 7

Pipe Breaks and Fire Flows for Each Demand Pattern Design

Demand Pattern	Pipe Broken	Fire Flows at Node	Fire Flows Demand (l/s)
1	1	1	70.0
2	2	4	70.0
3	3	1	70.0
4	4	3	70.0
5	5	2	70.0
6	6	6	90.0
7	7	3	70.0
8	8	7	70.0
9	9	4	70.0
10	10	4	70.0
11	11	7	90.0
12	12	10	90.0
13	13	9	90.0
14	14	6	90.0
15	15	12	50.0
16	16	13	70.0
17	17	8	70.0
18	18	8	70.0
19	19	9	90.0
20	20	9	90.0
21	21	11	100.0
22	22	12	50.0
23	23	12	50.0
24	24	11	100.0
25	25	11	100.0
26	26	13	70.0
27	27	17	70.0
28	28	13	70.0
29	29	15	70.0
30	30	14	120.0
31	31	19	120.0
32	32	15	70.0
33	33	18	120.0
34	34	19	120.0
35	35	17	70.0
36	36	20	120.0
37	37	20	120.0

TABLE 8

Results of First Run of Multiple Demand Pattern Design

Pipe	Diameter (m)	Length (m)	Diameter (m)	Length (m)	Maximum Weighting
1	0.200	350.98	0.250	409.02	1.000
2	0.150	250.77	0.200	269.23	0.703
3	0.200	753.42	0.250	136.58	0.894
4	0.200	1120.00	-	-	0.550
5	0.300	610.00	-	-	0.892
6	0.200	680.00	-	-	0.777
7	0.250	680.00	-	-	0.771
8	0.150	130.92	0.200	739.08	0.867
9	0.125	89.63	0.150	770.37	0.297
10	0.200	980.00	-	-	0.759
11	0.200	890.00	-	-	0.383
12	0.350	577.84	0.400	172.16	1.000
13	0.200	620.00	-	-	0.440
14	0.300	800.00	-	-	0.885
15	-	-	-	-	-
16	0.150	507.91	0.200	172.09	0.611
17	0.150	123.33	0.200	356.67	0.754
18	0.200	860.00	-	-	0.748
19	0.200	800.00	-	-	0.338
20	0.200	677.52	0.250	92.48	0.720
21	0.200	350.00	-	-	0.483
22	0.200	620.00	-	-	1.000
23	0.200	600.07	0.250	69.93	1.000
24	0.250	790.00	-	-	0.653
25	0.150	118.84	0.200	1031.16	0.612
26	0.150	123.94	0.200	626.06	0.686
27	0.150	150.86	0.200	399.14	0.759
28	0.150	700.00	-	-	0.389
29	0.200	500.00	-	-	0.952
30	0.250	314.92	0.300	135.08	0.955
31	0.200	647.03	0.250	102.97	0.643
32	0.200	720.00	-	-	1.000
33	0.250	540.00	-	-	0.848
34	0.300	606.71	0.350	93.29	1.000
35	0.150	466.31	0.200	383.69	0.808
36	0.200	546.60	0.250	203.40	1.000
37	0.200	569.24	0.250	400.76	1.000

Cost = $ 2,074,762

TABLE 9

Results of Final Run of Multiple Demand Pattern Design

Pipe	Diameter (m)	Length (m)	Diameter (m)	Length (m)	Maximum Weighting
1	0.250	760.00	-	-	1.000
2	0.150	112.99	0.200	407.01	1.000
3	0.250	796.70	0.300	93.30	1.000
4	-	-	-	-	-
5	0.300	370.60	0.350	239.40	1.000
6	0.200	680.00	-	-	0.725
7	0.200	473.58	0.250	206.42	1.000
8	0.200	314.96	0.250	555.04	1.000
9	-	-	-	-	-
10	0.200	519.99	0.250	460.01	1.000
11	0.250	890.00	-	-	1.000
12	0.400	750.00	-	-	1.000
13	0.250	620.00	-	-	0.797
14	0.350	540.84	0.400	259.16	1.000
15	-	-	-	-	-
16	0.150	98.14	0.200	581.86	1.000
17	0.150	41.82	0.200	438.18	1.000
18	0.200	173.27	0.250	686.73	1.000
19	-	-	-	-	-
20	0.200	770.00	-	-	0.889
21	-	-	-	-	-
22	0.200	35.54	0.250	584.46	1.000
23	0.150	345.08	0.200	324.92	0.663
24	0.200	336.57	0.250	453.43	0.553
25	0.200	1150.00	-	-	0.588
26	0.200	750.00	-	-	1.000
27	0.150	98.81	0.200	451.19	1.000
28	-	-	-	-	-
29	0.200	500.00	-	-	1.000
30	0.250	6.36	0.300	443.64	0.973
31	0.150	81.59	0.200	668.41	0.562
32	0.200	713.83	0.250	6.17	1.000
33	0.250	540.00	-	-	1.000
34	0.300	700.00	-	-	1.000
35	0.150	39.23	0.200	810.77	1.000
36	0.200	538.20	0.250	211.80	1.000
37	0.200	625.46	0.250	344.54	1.000

Cost = $ 1,950,698

5.4 NEW YORK CITY WATER SUPPLY TUNNELS EXPANSION

This example demonstrates how the technique can be applied to expansion of an existing system and compares the results to previous studies. The imperial system of measurements was used to facilitate comparison with the previous studies. The cost and layout in the New York City problem are described in Figure 9 and Tables 10 and 11. The proposed method of expansion is the same as in the previous studies (Schaake and Lai[1969],Quindry et al.[1981] and Gessler[1982]), i.e. to reinforce the system by constructing tunnels parallel to the existing tunnels. The existing pipe locations, lengths and diameters are shown in Table 10. The corresponding parallel reinforcing pipes are shown in column 2 of this table. In the initial assumption all reinforcing tunnels were assumed to be 84 inches in diameter.

The final results, shown in Table 12 and Figure 10, indicate that only a few pipes are needed to arrive at a least cost solution of 38.9 million. Two computer runs and 14.37 seconds CPU time were needed to obtain these results. This solution has discrete pipe diameters which span an entire link length in some links and are 'split' diameters in others. To compare this solution to that of Gessler[1982], which used only discrete pipes across the entire length of the links, the split pipes were replaced by a single diameter equal to the major portion of the present solution (see

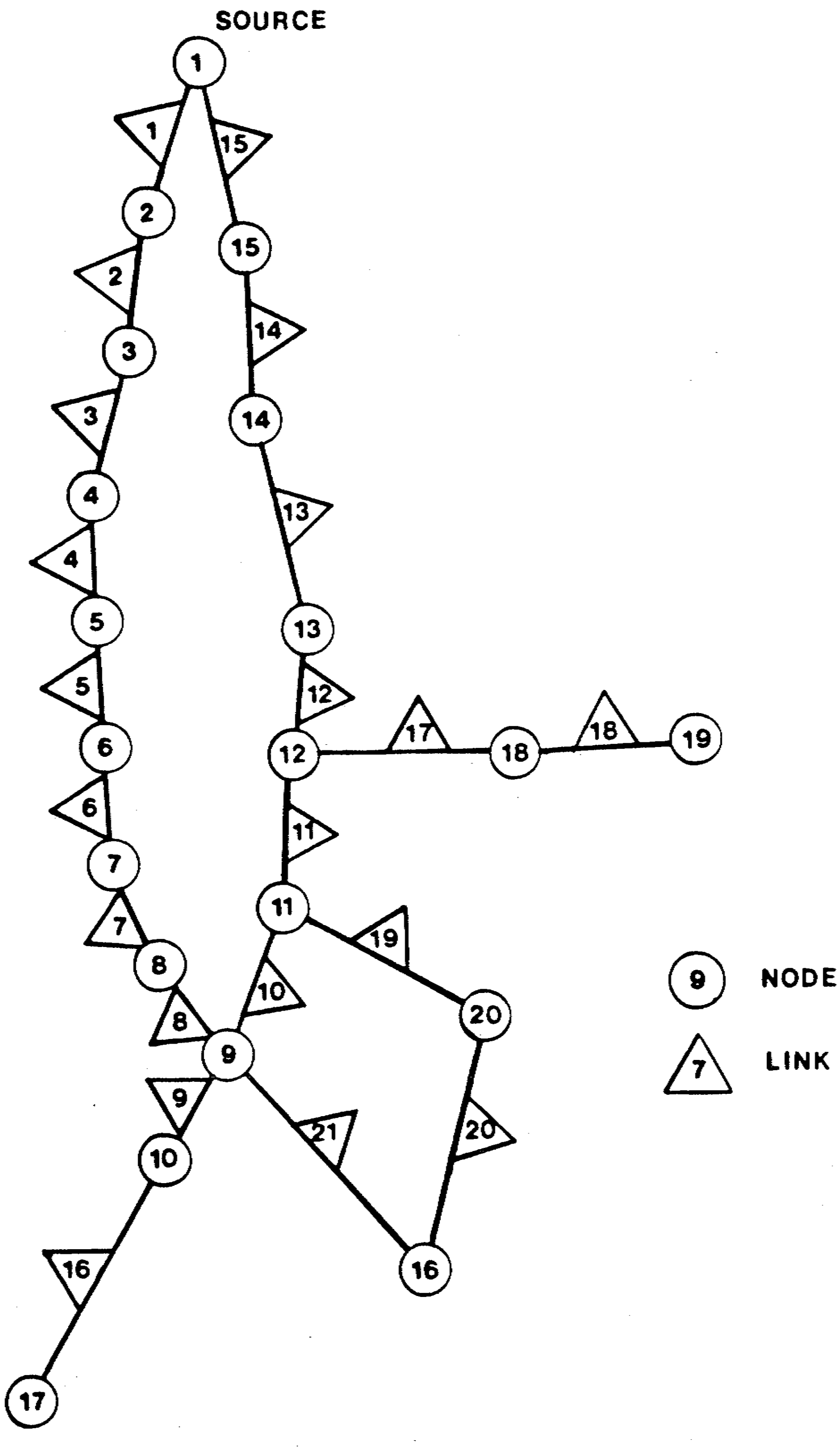

Figure 9: New York City Water Supply Tunnels

TABLE 10

Tunnel Costs for New York City

Diameter (inches)	Cost ($/ft)
36	93.50
48	134.00
60	176.00
72	221.00
84	267.00
96	316.00
108	365.00
120	417.00
132	469.00
144	522.00
156	577.00
168	632.00
180	689.00
192	746.00
204	804.00

TABLE 11

New York City Pipe Data

Pipe Existing	Pipe Reinforcing	Connecting Nodes From	Connecting Nodes To	Length (feet)	Existing Diameter (inches)
1	22	1	2	11600.	180
2	23	2	3	19800.	180
3	24	3	4	7300.	180
4	25	4	5	8300.	180
5	26	5	6	8600.	180
6	27	6	7	19100.	180
7	28	7	8	9600.	132
8	29	8	9	12500.	132
9	30	9	10	9600.	180
10	31	11	9	11200.	204
11	32	12	11	14500.	204
12	33	12	13	12200.	204
13	34	13	14	24100.	204
14	35	14	15	21100.	204
15	36	1	15	15500.	204
16	37	10	17	26400.	72
17	38	12	18	31200.	72
18	39	18	19	24000.	60
19	40	11	20	14400.	60
20	41	16	20	38400.	60
21	42	9	16	26400.	72

table 13). This configuration was tested using the Hardy Cross technique and the pressures were found to be adequate. The costs of these two solutions i.e. the split pipe solution and the discrete pipe solution, are compared with the previous studies in table 14. It can be seen that even the 'discrete' pipe solution, used for comparison with Gessler's solution, is less expensive than the previous solutions. Gessler's method, although it was a complete enumeration technique, failed to obtain this solution. This method's (decribed earlier in this thesis (p.18)) major shortcoming was that to reduce the number of combinations available pipes must be grouped together. In this problem pipes 7 and 8 are considered to be a single pipe, therefore a reinforcing pipe was added to both of these links. The solution of in this thesis is less expensive because only a reinforcing pipe parrellel to link 7 was used. A more complete record of the input and output datafor the New York City Water Supply Tunnels problem, including demands and pressures at each node, is shown in Appendix D.

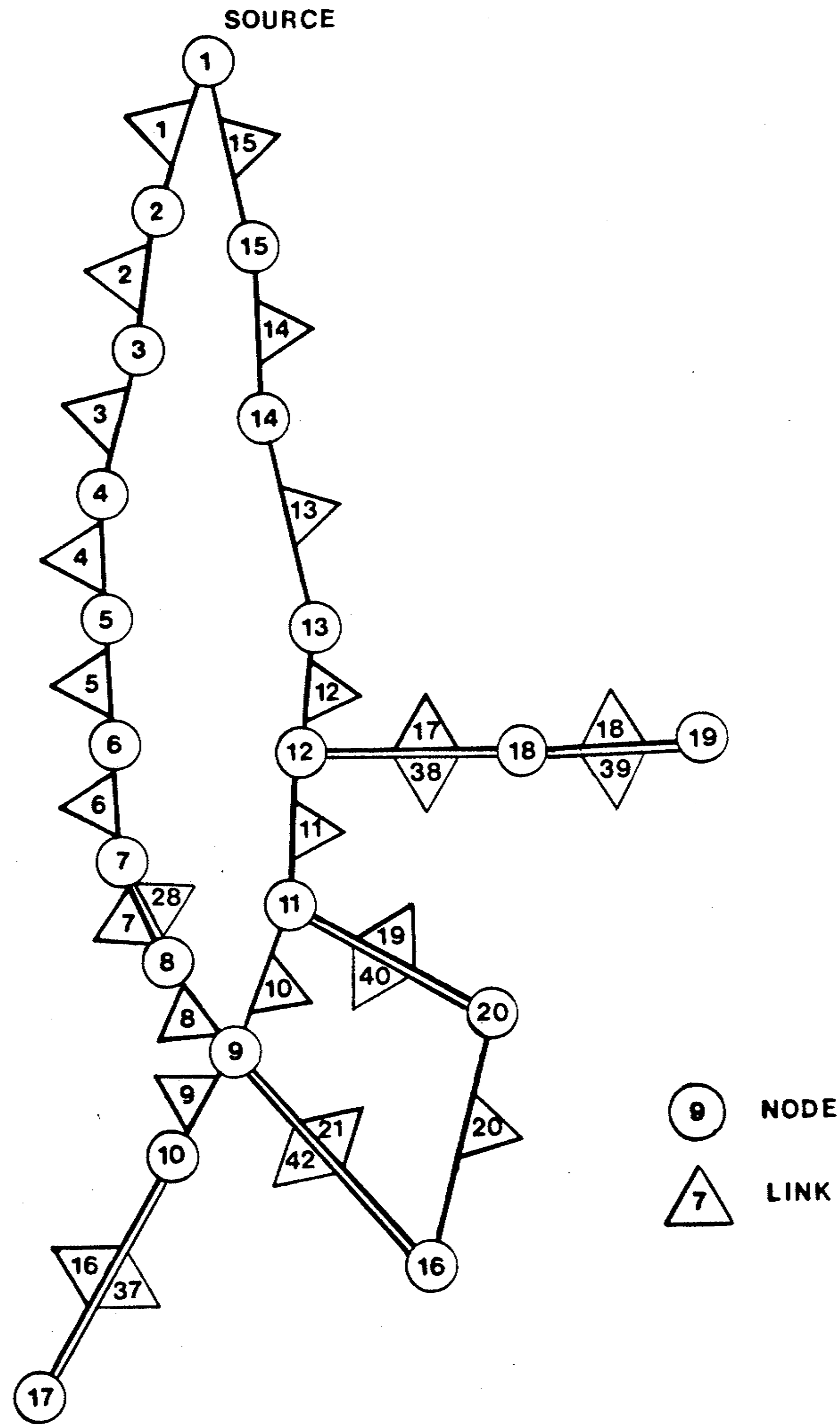

Figure 10: New York City Water Supply Tunnels Final Solution

TABLE 12

Split Pipe Solution for New York City Expansion

Pipe	Diameter (inches)	Length (feet)	Diameter (inches)	Length (feet)
28	144	9600.00	-	-
37	96	23432.08	108	2967.92
38(	96	31200.00	-	-
39	72	57.99	84	23942.01
40	48	4527.76	60	9872.24
42	72	3492.89	84	22907.11

Cost = $ 38.9 million

TABLE 13

Discrete Pipe Solution for New York City Expansion

Pipe	Diameter (inches)	Length (feet)	Diameter (inches)	Length (feet)
28	144	9600.00	-	-
37	96	26400.00	-	-
38	96	31200.00	-	-
39	84	24000.00	-	-
40	60	14400.00	-	-
42	84	26400.00	-	-

Cost = $ 39.2 million

TABLE 14

Comparison with Previous Studies

Schaake & Lai [1969]	78.1 million
Quindry et al. [1981]	63.6 million
Gessler [1982]	41.8 million
Split Pipe Solution	38.9 million
Discrete Pipe Solution	39.2 million

Chapter VI

CONCLUSIONS

A method of optimizing looped urban water distribution systems was developed. Many of the problems encountered in this type of optimization were addressed and the major conclusions are listed below.

1. Using a single demand pattern the loops are optimized out of the system thereby creating an inflexible system which will not perform adequately during an unexpected pipe failure.
2. A model has been developed which can consider many demand patterns simultaneously during the optimization procedure. Fire flow demands and pipe breakages were considered in each of the demand patterns.
3. Two widely available methods, the Hardy Cross network solver and a simple linear programming package were joined to produce the optimization technique.
4. The efficient simplex method of solving linear programming problems makes the technique applicable to large systems.
5. The technique produced a less expensive solution of the expansion of an existing system than had been obtained in all previous studies of that system.

6. The method incorporates a realistic cost function which uses standard unit costs (cost per length).

Appendix A

ANOMALIES IN ALPEROVITS AND SHAMIR'S TECHNIQUE

Since the method developed by Alperovits and Shamir strongly influenced this work, reasons requiring it to be modified are explained in detail in this appendix. In their method a single path was selected, presumably by the engineer, from each demand point to the source. The test system used to demonstrate their technique is shown in Figure 11. There are three different paths which can be used to constrain the pressure at node 5. The path choosen by Alperovits and Shamir (1-3-4) produced the result shown in Table 15. However, if an alternative path is choosen (1-2-7) then a less expensive solution, shown in Table 16, is arrived at. From these results it can be inferred that the method may choose larger pipes along the path which is selected, and reduces, to a minimum diameter, the pipes which happen to be off the paths. A different choice of paths will give a different least cost solution. Unfortunately it is not possible a priori to determine which path will lead to the least expensive solution. This is the main reason why this technique must be modified to consider all paths from the source(s) to each demand point.

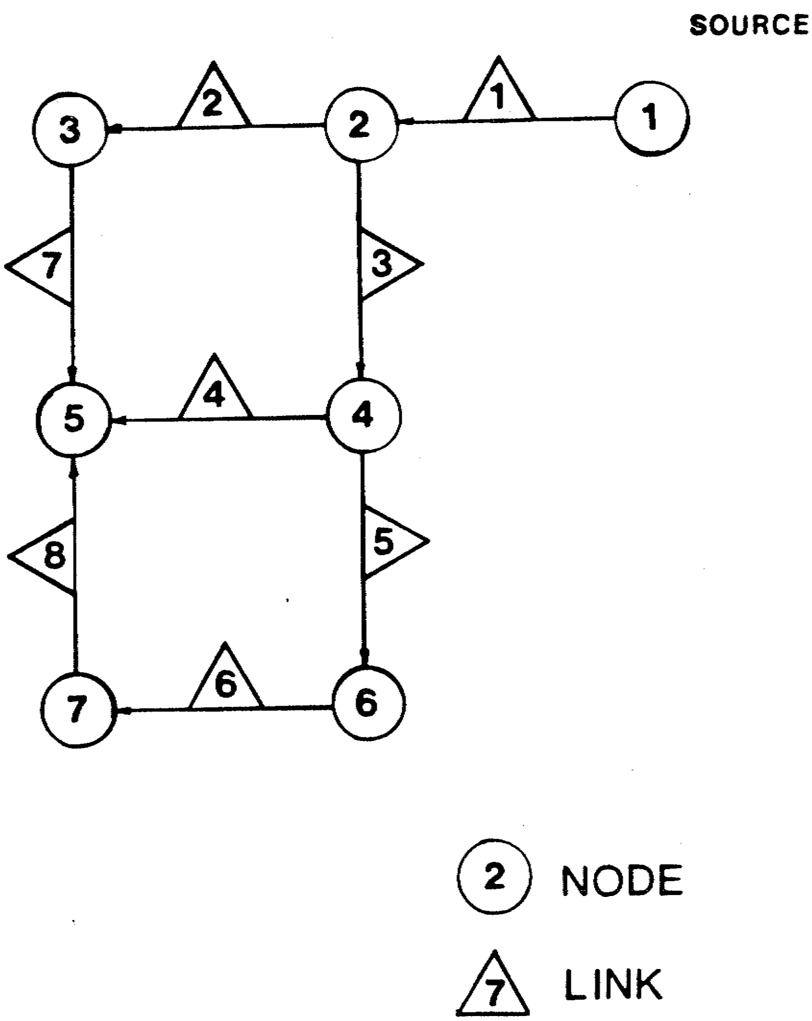

Figure 11: Example Used by Alperovits and Shamir

TABLE 15

Results Using Path 1-3-4

Pipe	Diameter (inches)	Length (feet)	Diameter (inches)	Length (feet)
1	18	783.43	20	216.57
2	6	666.60	8	333.40
3	18	1000.00	-	-
4	8	511.07	10	488.93
5	16	1000.0	-	-
6	10	902.84	12	97.16
7	1	1000.00	-	-
8	1	1000.00	-	-

Cost = $443,079

TABLE 16

Results Using Path 1-2-7

Pipe	Diameter (inches)	Length (feet)	Diameter (inches)	Length (feet)
1	18	889.54	20	110.46
2	10	653.57	12	346.43
3	16	1000.00	-	-
4	1	1000.00	-	-
5	16	1000.0	-	-
6	10	902.72	12	97.27
7	8	50.78	10	949.22
8	1	1000.00	-	-

Cost =$421,948

Appendix B

LISTING OF COMPUTER PROGRAM

```
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C:
C:                    VARIABLE LIST
C:
C: AAA(NL,I)           DIFFERENCE BETWEEN ACTUAL PRESSURE AND MINIMUN
C:                     ALLOWABLE PRESSURE AT NODE I IN DEMAND PATTERN NL
C: ALP(NC,NV)          SIMPLEX COEFFICIENT MATRIX
C: BIG                 LARGE CONSTANT (10**10)
C: BLP(NC)             RIGHT HAND SIDE OF CONSTRAINT MATRIX
C: BREAK(NL,J)         0 IF PIPE J IN DEMAND PATTERN NL IN SERVICE
C:                     1 IF PIPE J IN DEMAND PATTERN NL OUT OF SERVICE
C: CHOICE(I)           NUMBER OF PIPES CONNECTING NODE I
C: CLP(NV)             COST FUNCTION IN SIMPLEX ALGORITHM
C: CONDUC(J)           CONDUCTIVITY OF LINK J
C: COSPIP(ID)          COST PER METRE OF PIPE OF DIAMETER NUMBER ID
C: COS1                TEMPORARY COST PER METRE VALUE
C: DELH                CHANGE OF PRESSURE TO MOVE TOWARD SOLUTION OF
C:                      FLOW AND PRESSURE PROBLEM
C: DELQ                AVERAGE ERROR OF FLOW IN HARDY CROSS SOLUTION
C: DEMAND
C: DENOM               DENOMINATER IN PRESSURE CHANGE CALCULATION IN
C:                      HARDY CROSS SOLUTION
C: DEQUIV              EQUIVALENT DIAMETER OF SPLIT PIPES
C: DEXP                HAZEN WILLIAMS EXPONENT = 2.63
C: DIAM(ID)            DIAMETER OF PIPE OF DIAMETER NUMBER ID
C: DIV1                DIVISOR OF COST FUNCTION TO MAKE VALUES FALL
C:                      WITHIN A CERTAIN RANGE FOR SIMPLEX PACKAGE
C: DIV2(N2)            DIVISOR OF CONSTRAINT MATRIX VALUES
C: DQEX                EXPONENT IN HAZEN WILLIAMS EQUATION (-DEXP/QEXP)
C: DSOL(NC)            DUAL SOLUTION OF SIMPLEX SOLUTION
C: DO
C: D1                  TEMPORARY DIAMETER VALUE
C: D2                  TEMPORARY DIAMETER VALUE
C: D3                  TEMPORARY DIAMETER VALUE
C: EPSOL               CHANGE IN COST FROM ONE ITERATION TO THE NEXT
C: EQUA                NUMBER OF EQUALITY CONSTRAINTS IN LP FORMULATION = 0
C: EXIST(J)            DIAMETER NUMBER OF EXISTING PIPE
C: FACTOR(K,J)         WEIGHTING FACTOR FOR CONSTRAINT K AND PIPE J
C: FLOW(NL,J)          FLOW IN PIPE J DURING DEMAND PATTERN NL
C: F1                  TEMPORARY FLOW
C: HAZEN               HAZEN-WILLIAMS COEFFICIENT
C: HCON                NUMBER OF PRESSURE CONSTRAINTS
C: HDIFF               PRESSURE DROP ACCROSS A PIPE
C: I                   NODE NUMBER
C: ID                  DIAMETER NUMBER
C: IDPT1(NV)           DIAMETER OF REPLACEMENT PIPE OF VARIABLE NV
C: IDPT2(NV)           DIAMETER OF DESIGNATED PIPE
C: IER                 ERROR NUMBER FOR SIMPLEX PACKAGE
C: II                  COUNTER FOR READ DO LOOPS
C: IK(K)               POINTER TO NODE I WHOSE PRESSURE IS
C:                      CONSTRAINED BY CONSTRAINT K
C: INEQ                NUMBER OF INEQUALITY CONSTRAINTS
C: INFLOW(NL,I)        TOTAL INFLOW INTO NODE I DEMAND PATTERN NL
C: INHEAD(NL,I)        ACTUAL PRESSURE AT NODE I DEMAND PATTERN NL
```

```
C: IPT(NC)              POINTER TO NODE OF WHICH THE PRESSURE IS CONSTRAINED
C:                       BY CONSTRAINT NC
C: ISUM                 VALUE TO CHECK IF TWO CONSTRAINTS HAVE IDENTICAL
C:                       PATHS
C: IT                   ITERATION COUNTER FOR OPTIMIZATION PROCEDURE
C: ITEMP                TEMPORY VALUE (INTEGER)
C: ITER(NL)             NUMBER OF HARDY CROSS ITERATIONS
C: IW                   WORK VECTOR FOR SIMPLEX PACKAGE
C: I2                   TEMPORARY NODE NUMBER USED TO TRACE FLOW FROM
C:                       CONSTRAINED NODE TO THE SOURCE(S)
C: J                    PIPE NUMBER
C: JJ                   COUNTER FOR READING IN PIPE DATA
C: JNV(NV)              PIPE NUMBER ASSOCIATED WITH VARIABLE NV
C: JPT(NC)              LINK ASSOCIATED WITH LENGTH CONTRAINT NC
C: J1                   HYDRAULIC GRADIENT
C: J2                   HYDRAULIC GRADIENT
C: K                    PRESSURE CONSTRAINT NUMBER
C: KA                   CONSTANT MUST EQUAL FIRST DIMENSION OF ALP(NC,NV)
C: KEND                 NUMBER OF PRESSURE CONSTRAINTS
C: K1                   CONSTRAINT NUMBER
C: L                    COUNTER FOR LINKS CONNECTING A NODE
C: LEN(1,J)             LENGTH OF SECTION 1 OF LINK J
C: LENGTH(J)            LENGTH OF SECTION J
C: LENTES               IF =0 THEN NO LENGTH CONSTRAINTS ARE BINDING
C:                         =1 THEN A LENGTH CONSTRAINT IS BINDING
C: LL                   LENGTH OF LINK IN 1000 UNITS
C: LOADS                TOTAL NUMBER OF DEMAND PATTERNS
C: L1                   LENGTH OF SECTION 1
C: L2                   LENGTH OF SECTION 2
C: MAXFAC(J)            MAXIMUM WIEGHTING FACTOR FOR LINK J
C: MINHED(NL,I)         MINIMUM ALLOWABLE PRESSURE FOR NODE I
C:                       DEMAND PATTERN J
C: N                    LINK END NUMBER 1 OR 2
C: NC                   CONSTRAINT NUMBER
C: ND(2,J)              DIAMETER NUMBER FOR SECTION 2 OF LINK J
C: NDIAM                NUMBER OF DIFFERENT DIAMETERS
C: ND1                  DIAMETER NUMBER FOR SECTION 1
C: ND2                  DIAMETER NUMBER FOR SECTION 2
C: NFACT(I)             WEIGHT OF NODE I
C: NL                   DEMAND PATTERN NUMBER
C: NLK(K)               POINTER TO DEMAND PATTERN ASSOCIATED WITH PRESSURE
C:                       CONSTRAINT K
C: NLPT(NC)             POINTER TO DEMAND PATTERN ASSOCIATED WITH
C:                       CONTRAINT K
C: NN                   COUNTER FOR READING IN DEMAND PATTERNS
C: NODE(2,J)            NODE NUMBER CONNECTING END 2 OF LINK J
C: NODES                NUMBER OF NODES
C: NTYPE                TYPE OF PROGRAM TO RUN 0 = HARDY CROSS ANALYSIS
C:                                             1 = OPTIMIZATION
C: NUMCON               TOTAL NUMBER OF CONSTRAINTS
C: NUMVAR               TOTAL NUMBER OF VARIABLES
C: NV                   VARIABLE NUMBER
C: PATH(K,J)             IF 1 THEN LINK J IS ON ONE OF THE PATHS OF
                          CONSTRAINT K, OTHERWISE = 0
```

```
C: PENALT             VALUE OF PENALTY COST (SHOULD BE ABOUT 3X MAXIMUM
C:                     PIPE COST)
C: PIPES              TOTAL NUMBER OF LINKS
C: PSOL(NV)           PRIMAL SOLUTION
C: PT                 POINTER USED TO KEEP TRACK OF POSITION WHEN
C:                     KEEPING TRACK OF PATHS FROM NODE TO SOURCE(S)
C: PT1                    "                "              "
C: PT2                    "                "              "
C: QEXP               EXPONENT USED IN THE HAZEN WILLIAMS FORMULA=0.54
C: QNEG               -0.54
C: RW                 WORK VECTOR FOR SIMPLEX PACKAGE
C: S                  SIMPLEX SOLUTION
C: SELECT(I,L)        LINK NUMBERS OF LINKS CONNECTED TO NODE I
C: SIGN1,SIGN2,SIGN3  VARIABLES TO CHANGE SIGN
C: SMALL              LOWEST AAA VALUE
C: SOURCE(NL,I)       RESERVOIR IN DEMAND PATTERN NL AT NODE I
C: SUMNOD             TOTAL DEMAND IN ALL NODES
C: SUPPLY(NL,I)       DEMAND CALCULATED BY HARDY CROSS METHOD
C: TEMP               TEMPORARY VALUES
C: TOTAL1(IT)         SYSTEM COST CALCULATED FROM SUBTRACTING  S
C:                     FROM PREVIOUS TOTAL2
C: TOTAL2(IT)         SYSTEM COST AS CALCULATED BY SUMMING PIPE COSTS
C: UNITS              COEFFICIENT IN HAZEN WILLIAMS FORMULA WHICH
C:                     CHANGES WITH UNITS
C: WORKN(PT)          WORK VECTOR FOR WEIGTHING LINKS
C:
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
      REAL*8 DIAM(23),D1,D2,UNITS,HAZEN,LENGTH(49),LEN(2,49)
     $       ,DEXP,QEXP,DQEX,L1,L2,LL,DEQUIV,QNEG,J1,J2,TEMP
     $       ,COSPIP(23)
      REAL DEMAND(40,30),CONDUC(49),SUPPLY(40,30),SUMNOD
     $     ,DELQ,HDIFF,INHEAD(40,30),TOTAL2(90),DIV1,DIV2(120)
     $     ,FLOW(40,49),AAA(40,30),MINHED(40,30),BLP(120)
     $     ,BIG,SMALL,F1,CLP(153),ALP(120,153)
     $     ,TOTAL1(90),PSOL(153),DSOL(120),RW(25000)
     $     ,FACTOR(40,49),MAXFAC(49)
      REAL NFACT(30),INFLOW(40,49),PENALT
      INTEGER PT,PT1,PT2,WORKN(40),EXIST(49)
      INTEGER SOURCE(40,30),NODE(2,49),SELECT(30,10),CHOICE(30),ND(2,49)
     $        ,NL,I,J,L,LOADS,NODES,PIPES,BREAK(40,49),SIGN1,SIGN2,SIGN3
     $        ,K,NLK(49),IK(49),N,I2,ND1,ND2,PATH(40,49)
     $        ,KEND,K1,ISUM,NV,NC,ID
     $    ,IDPT1(153),JNV(153),NUMVAR,NUMCON,IPT(120),NLPT(80),IDPT2(153)
     $        ,IER,IW(1000),INEQ,EQUA,ITER(40)
      INTEGER JPT(120),IT,NDIAM,HCON
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C:                                                                  :
C:                          READ IN DATA                            :
C:                                                                  :
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      IT=1
      REWIND 9
      READ(9,*) NTYPE
```

```
      READ(9,*)  UNITS
      READ(9,*)  HAZEN
      READ(9,*)  PENALT
      READ(9,*)  NDIAM
      IF(NTYPE.NE.1)  WRITE(6,32)
   32 FORMAT(///20X,'**** HARDY CROSS ANALYSIS ONLY ***')
      IF(NTYPE.EQ.1)  WRITE(6,34)
   34 FORMAT(///20X,'***** NEWORK OPTIMIZER *****')
      WRITE(6,36)  UNITS,HAZEN,NDIAM
   36 FORMAT(//25X,'UNITS=',F13.7,//25X,'HAZEN-WILLIAMS C=',F7.0,
     $        //25X,'NUMBER OF CANDIDATE DIAMETERS',I5)
      WRITE(6,39)
   39 FORMAT(//20X,'         DIAMETER       COST',/)
      DO 44 II=1,NDIAM
         READ(9,*)  ID,DIAM(ID),COSPIP(ID)
         WRITE(6,43)  ID,DIAM(ID),COSPIP(ID)
   43    FORMAT(17X,I5,F13.3,F12.2)
   44 CONTINUE
C
C                              READ IN SYSTEM LAYOUT DATA
      REWIND 10
      READ(10,*)  PIPES,NODES
      WRITE(6,50)  PIPES,NODES
   50 FORMAT(//12X,'NUMBER OF LINKS',I5,5X,'NUMBER OF NODES',I5)
      WRITE(6,52)
   52 FORMAT(//15X,' PIPE  FROM    TO        LENGTH',/)
      DO 57 JJ=1,PIPES
         READ(10,*)  J,NODE(1,J),NODE(2,J),LENGTH(J)
         WRITE(6,56)  J,NODE(1,J),NODE(2,J),LENGTH(J)
   56    FORMAT(13X,3I6,F14.0)
   57 CONTINUE
C
      REWIND 11
      WRITE(6,61)
   61 FORMAT('1',/23X,'INITIAL ASSUMPTION')
      WRITE(6,63)
   63 FORMAT(//3X,' PIPE    DIAMETER    LENGTH    DIAMETER    LENGTH',
     $        '      EXISTING',/)
      TOTAL2(1)=0
      DO 82 JJ=1,PIPES
         READ(11,*)  J,ND(1,J),LEN(1,J),ND(2,J),LEN(2,J),EXIST(J)
         D1=DIAM(ND(1,J))
         D2=0
         D3=0
         IF(EXIST(J).NE.0)  D3=DIAM(EXIST(J))
         IF(LEN(2,J).NE.0)  D2=DIAM(ND(2,J))
         IF(LEN(2,J).EQ.0.AND.ND(1,J).EQ.1)  D1=0
         WRITE(6,75)  J,D1,LEN(1,J),D2,LEN(2,J),D3
   75 FORMAT(I6,6X,F7.3,F12.2,F10.3,F10.2,F10.3)
      IF(EXIST(J).NE.0)  GO TO 82
      IF(D1.EQ.0)  GO TO 82
C                              CALCULATE INITIAL COST
      TOTAL2(1)=TOTAL2(1)+LEN(1,J)*COSPIP(ND(1,J))
      IF(LEN(2,J).EQ.0)  GO TO 82
```

```
      TOTAL2(1)=TOTAL2(1)+LEN(2,J)*COSPIP(ND(2,J))
   82 CONTINUE
      WRITE(6,84) TOTAL2(1)
   84 FORMAT(//25X,'INITIAL COST=',F12.0)
C
      WRITE(6,87)
   87 FORMAT('1')
      REWIND 12
      READ(12,*) LOADS
      READ(12,*) NODES
      WRITE(6,92) LOADS,NODES
   92 FORMAT(/20X,'NUMBER OF LOADING CONDITIONS',I5,
     $       /20X,'NUMBER OF NODES',I5)
      WRITE(6,61)
      WRITE(6,96)
   96 FORMAT(//15X,'NODE   MINHEAD     DEMAND       INHEAD',/)
      DO 105 NN=1,LOADS
         READ(12,*) NL
         WRITE(6,634) NL
         DO 104 II=1,NODES
            READ(12,*) I,MINHED(NL,I),DEMAND(NL,I),INHEAD(NL,I)
            WRITE(6,103) I,MINHED(NL,I),DEMAND(NL,I),INHEAD(NL,I)
  103       FORMAT(13X,I5,F10.2,F12.1,F13.2)
  104    CONTINUE
  105 CONTINUE
      DO 110 NL=1,LOADS
         DO 109 I=1,NODES
            BREAK(NL,I)=0
  109    CONTINUE
  110 CONTINUE
      WRITE(6,112)
  112 FORMAT(///10X,'PIPES OUT OF SERVICE IN EACH CONDITION')
      REWIND 14
  114 READ(14,*) NL
      IF(NL.EQ.0) GO TO 124
      WRITE(6,117) NL
  117 FORMAT(/3X,'LOAD',I6)
  118 READ(14,*) J
      IF(J.LE.0) GO TO 114
      WRITE(6,121) J
  121 FORMAT(10X,'PIPE'I5)
      BREAK(NL,J)=1
      GO TO 118
  124 CONTINUE
      WRITE(6,87)
      IF(NTYPE.EQ.1)WRITE(6,127)
  127 FORMAT(//5X,'ITERATION',10X,'COST 1',10X,'COST 2',/)
C
C                                   CALCULATE THE NUMBER OF POSSIBLE
C                                   PIPES CONNECTING EACH NODE
      LENTES=1
  132 CONTINUE
      DO 146 I=1,NODES
         DO 137 NL=1,LOADS
```

```
            SOURCE(NL,I)=0
            IF(DEMAND(NL,I).LT.0)  SOURCE(NL,I)=1
  137     CONTINUE
C
          L=0
          DO 144 J=1,PIPES
             IF(NODE(1,J).NE.I .AND. NODE(2,J).NE.I)  GO TO 144
             L=L+1
             SELECT(I,L)=J
  144     CONTINUE
          CHOICE(I)=L
  146 CONTINUE
C
C                               HARDY CROSS FLOW BALANCING METHOD
C
      QEXP=0.54
      DEXP=2.63
      DQEX=(-1.0D0)*DEXP/QEXP
      QNEG=(-1.0D0)*QEXP
C     WRITE(6,155)  QEXP,DEXP,DQEX,QNEG
C 155 FORMAT(//3X,'N=',F12.7,//3X,'M=',F12.7,//3X,'-M/N=',F12.7,
C    $         //3X,'-N=',F12.7)
C     WRITE(6,158)
C 158 FORMAT(//3X,'PIPE      CONDUCTIVITY')
      DO 179 J=1,PIPES
         CONDUC(J)=0
         IF(ND(1,J).EQ.1.AND.LEN(2,J).EQ.0)  GO TO 176
         L1=LEN(1,J)/1000.00
         L2=LEN(2,J)/1000.00
         LL=LENGTH(J)/1000.00
         IF(DIAM(ND(1,J)).EQ.0)  GO TO 176
         IF(LEN(2,J).EQ.0)  GO TO 175
         D1=DIAM(ND(1,J))
         D2=DIAM(ND(2,J))
         DEQUIV=(D1**DQEX*L1+D2**DQEX*L2)**QNEG
C        WRITE(6,*)  DEQUIV
         CONDUC(J)=UNITS*HAZEN*DEQUIV
C        CONDUC(J)=UNITS*HAZEN*((DIAM(ND(1,J))**DQEX)*L1
C    $   (DIAM(ND(2,J))**DQEX)*L2)**(-1.0D0/QEXP)
         GO TO 176
  175    CONDUC(J)=UNITS*HAZEN*DIAM(ND(1,J))**DEXP/LL**QEXP
  176    CONTINUE
C        WRITE(6,178)  J,CONDUC(J)
C 178    FORMAT(I5,F15.2)
  179 CONTINUE
C
      DO 239 NL=1,LOADS
C        WRITE(6,634)  NL
       ITER(NL)=0
  184    DELQ=0
         DO 217 I=1,NODES
            SUPPLY(NL,I)=0
            SUMNOD=0
            DENOM=0
```

```
            ITEMP=CHOICE(I)
            DO 208 L=1,ITEMP
            J=SELECT(I,L)
               IF(BREAK(NL,J).EQ.1) GO TO 208
               SIGN1=1
               IF(NODE(2,J).EQ.I) GO TO 197
C              IF(NODE(1,J).NE.I) GO TO 208
               SIGN1=-1
  197          HDIFF=INHEAD(NL,NODE(1,J))-INHEAD(NL,NODE(2,J))
               SIGN2=1
               IF(HDIFF.LT.0) SIGN2=-1
               HDIFF=ABS(HDIFF)
               FLOW(NL,J)=CONDUC(J)*(HDIFF**QEXP)*SIGN2
               SUPPLY(NL,I)=SUPPLY(NL,I) + FLOW(NL,J)*SIGN1
               SUMNOD=SUMNOD+ABS(FLOW(NL,J))
               IF(HDIFF.EQ.0) GO TO 208
               DENOM = DENOM + FLOW(NL,J)*SIGN2/HDIFF
C             WRITE(6,207) NL,I,J,HDIFF,FLOW(NL,J),SUPPLY(NL,I),DENOM
C 207          FORMAT(3X,'7001',3I5,F8.1,2F8.0,F8.1)
  208       CONTINUE
            IF(SOURCE(NL,I).NE.0) GO TO 217
            IF(SUMNOD.EQ.0)SUMNOD=0.1
            IF(DENOM.EQ.0) DENOM=0.1
            DELQ=DELQ+ABS(DEMAND(NL,I)-SUPPLY(NL,I))/SUMNOD
            DELH=(DEMAND(NL,I)-SUPPLY(NL,I))/(QEXP*DENOM)
            INHEAD(NL,I)=INHEAD(NL,I)-DELH
C           WRITE(6,216) DELQ,DELH,INHEAD(NL,I)
C 216       FORMAT(3X,'7002',F8.0,F8.2,F8.2)
  217    CONTINUE
C        WRITE(6,219) DELQ
C 219    FORMAT(3X,'DELQ=',F10.4)
         ITER(NL)=ITER(NL)+1
         IF(ITER(NL).GT.500) GO TO 223
         IF(DELQ.GT.0.01) GO TO 184
  223    CONTINUE
C        WRITE(6,225) ITER(NL)
C 225    FORMAT(/3X,'NUMBER OF ITERATIONS',I5)
         DO 238 I=1,NODES
            ITEMP=CHOICE(I)
            INFLOW(NL,I)=0
            DO 237 L=1,ITEMP
               J=SELECT(I,L)
               SIGN1=1
               IF(NODE(2,J).EQ.I) GO TO 234
               SIGN1=-1
  234          F1=FLOW(NL,J) *SIGN1
               IF(F1.LE.0) GO TO 237
               INFLOW(NL,I)=INFLOW(NL,I)+F1
  237       CONTINUE
  238    CONTINUE
  239 CONTINUE
      IF(NTYPE.NE.1) GO TO 580
      DO 243 NL=1,LOADS
         IF(ITER(NL).GE.3) GO TO 248
```

```
  243 CONTINUE
      GO TO 580
C                               CALCULATE CHANGE IN COST
C                               IF LESS THAN 0.01 % AND LENGTH
C                               CONSTRAINTS NONBINDING
  248 CONTINUE
      EPSOL=ABS(TOTAL2(IT)-TOTAL2(IT-1))/TOTAL2(IT)
      IF(EPSOL.LT.0.0001.AND.LENTES.EQ.0) GO TO 580
      IF(IT.GE.65) GO TO 580
C                               FIND PATHS OF BINDING CONSTRAINTS
      DO 258 J=1,PIPES
         DO 257 K=1,PIPES
            PATH(K,J)=0
            FACTOR(K,J)=0
  257    CONTINUE
  258 CONTINUE
      BIG=10**10
      DO 265 I=1,NODES
         DO 264 NL=1,LOADS
            AAA(NL,I)=INHEAD(NL,I)-MINHED(NL,I)
            IF(SOURCE(NL,I).EQ.1)AAA(NL,I)=BIG
  264    CONTINUE
  265 CONTINUE
C
      K=0
  268 K=K+1
      IF(K.GT.PIPES) GO TO 355
C                               SEARCH FOR LOWEST PRESSURE
      SMALL=BIG
      DO 279 I=1,NODES
         DO 278 NL=1,LOADS
            IF(AAA(NL,I).GE.SMALL) GO TO 278
            SMALL=AAA(NL,I)
            NLK(K)=NL
            IK(K)=I
  278    CONTINUE
  279 CONTINUE
      IF(SMALL.EQ.BIG) GO TO 355
      IF(SMALL.GT.10) GO TO 355
C                               TRACE PATH TO A SOURCE
      DO 285 I=1,NODES
         NFACT(I)=0
  285 CONTINUE
      NL=NLK(K)
      I=IK(K)
      NFACT(I)=1
      PT1=1
      PT2=1
C
C                               SEARCH FOR PATHS FROM NODE TO SOURCE(S)
  293 L=0
  294 L=L+1
      IF(L.GT.CHOICE(I)) GO TO 327
      J=SELECT(I,L)
```

```
      IF(ND(1,J).EQ.1.AND.LEN(2,J).EQ.0) GO TO 294
      IF(BREAK(NL,J).EQ.1) GO TO 294
      N=1
      SIGN1=1
      IF(NODE(N,J).NE.I) GO TO 304
      N=2
      SIGN1=-1
  304 I2=NODE(N,J)
      F1=FLOW(NL,J)*SIGN1
      IF(F1.LE.0.01) GO TO 294
      PATH(K,J)=1
      TEMP=ABS(F1/INFLOW(NL,I))*NFACT(I)
      NFACT(I2)=NFACT(I2)+(TEMP-FACTOR(K,J))
      FACTOR(K,J)=TEMP
      PT=PT1
  312 PT=PT+1
      IF(PT.GT.40) PT=1
      IF(PT.EQ.(PT2+1)) GO TO 320
      IF(PT1.EQ.1.AND.PT2.EQ.1) GO TO 320
      IF(PT.EQ.1.AND.PT2.EQ.40) GO TO 320
      IF(WORKN(PT).EQ.I2) GO TO 294
      GO TO 312
C
  320 CONTINUE
      PT2=PT2+1
      IF(PT2.GT.40)PT2=1
      WORKN(PT2)=I2
C     WRITE(6,325) PT1,PT2,WORKN(PT2),I2
C 325 FORMAT(7X,'4001',4I7)
      GO TO 294
  327 PT1=PT1+1
      IF(PT1.GT.40) PT1=1
      I=WORKN(PT1)
      IF(PT1.EQ.(PT2+1)) GO TO 334
      IF(PT1.EQ.1.AND.PT2.EQ.40) GO TO 334
      IF(SOURCE(NL,I).EQ.1) GO TO 327
      GO TO 293
  334 CONTINUE
C                                CHECK IF PATH HAS BEEN TRACED ALREADY
      AAA(NL,IK(K))=BIG
      KEND=K-1
      IF(KEND.EQ.0) GO TO 268
      DO 353 K1=1,KEND
         ISUM=0
         DO 344 J=1,PIPES
            ISUM=PATH(K,J)-PATH(K1,J)
            IF(ISUM.NE.0) GO TO 353
  344    CONTINUE
C                                IF IDENTICAL PATH HAS BEEN TRACED DON'T
C                                SAVE THIS ONE
         DO 350 J=1,PIPES
            PATH(K,J)=0
            FACTOR(K,J)=0
  350    CONTINUE
```

```
         K=K-1
         GO TO 268
  353 CONTINUE
      GO TO 268
  355 HCON=K-1
      DO 359 K=1,HCON
C        WRITE(6,358)  (FACTOR(K,J),J=1,PIPES)
C 358    FORMAT(15F7.3)
  359 CONTINUE
C     IF(IT.GT.3)STOP
C
C                       COST FUNCTION
      NV=0
      DO 394 J=1,PIPES
         IF(LEN(2,J).EQ.O.AND.ND(1,J).EQ.1)  GO TO 394
         COS1=COSPIP(ND(1,J))
         IF(EXIST(J).EQ.ND(1,J))GO TO 394
         IF(LEN(2,J).EQ.O)  GO TO 380
         NV=NV+1
         CLP(NV)=COS1-COSPIP(ND(2,J))
         IDPT1(NV)=ND(1,J)
         IDPT2(NV)=ND(2,J)
         JNV(NV)=J
         NV=NV+1
         CLP(NV)=COSPIP(ND(2,J))-COS1
         IDPT1(NV)=ND(2,J)
         IDPT2(NV)=ND(1,J)
         JNV(NV)=J
         GO TO 394
  380    ID=ND(1,J)-1
         IF(ID.LT.1)  GO TO 387
         NV=NV+1
         CLP(NV)=COSPIP(ID)-COS1
         IDPT1(NV)=ID
         IDPT2(NV)=ND(1,J)
         JNV(NV)=J
  387    ID=ND(1,J)+1
         IF(ID.GT.NDIAM)  GO TO 394
         NV=NV+1
         CLP(NV)=COSPIP(ID)-COS1
         IDPT1(NV)=ID
         IDPT2(NV)=ND(1,J)
         JNV(NV)=J
  394 CONTINUE
      NUMPIP=NV
      DO 399 I=1,PIPES
         NV=NV+1
         CLP(NV)=PENALT
  399 CONTINUE
      NUMVAR=NV
      DO 403  NV=1,NUMVAR
         CLP(NV)=CLP(NV)*(-1)
  403 CONTINUE
C     WRITE(6,405)  (CLP(NV),NV=1,NUMVAR)
```

```
C 405 FORMAT(7F9.2)
C                             CONSTRUCT HEADLOSS CONSTRAINTS
C                             VERSION MAY 23,83
      NUMCON=HCON+PIPES*2
      DO 413 NV=1,NUMVAR
         DO 412 NC=1,NUMCON
            ALP(NC,NV)=0
  412    CONTINUE
  413 CONTINUE

      NC=0
      IF(HCON.EQ.0) GO TO 444
      DO 443 K=1,HCON
         NL=NLK(K)
         I=IK(K)
         NC=NC+1
         NV=0
         TEST=0
         DO 435 NV=1,NUMPIP
            J=JNV(NV)
            IF(PATH(K,J).NE.1) GO TO 435
            IF(EXIST(J).NE.0) GO TO 435
            F1=ABS(FLOW(NL,J))
  428       CONTINUE
            J1=(F1/(HAZEN*UNITS))**(1/QEXP)*DIAM(IDPT1(NV))
     $          **DQEX
            J2=(F1/(HAZEN*UNITS))**(1/QEXP)*DIAM(IDPT2(NV))
     $          **DQEX
            ALP(NC,NV)=(J1-J2)*FACTOR(K,J)
            IF(ALP(NC,NV).GE.TEST) TEST=ALP(NC,NV)
  435    CONTINUE
         NV=NUMPIP+K
         ALP(NC,NV)=(-0.001)*TEST
         BLP(NC)=(INHEAD(NL,I)-MINHED(NL,I))
C        WRITE(6,405) (ALP(NC,NV),NV=1,NUMVAR)
C        WRITE(6,405) BLP(NC)
         NLPT(NC)=NL
         IPT(NC)=I
  443 CONTINUE
  444 CONTINUE
C                                        LENGTH CONTRAINTS
      DO 463 NV=1,NUMPIP
         IF(CLP(NV).LT.0) GO TO 456
         J=JNV(NV)
         NC=NC+1
         JPT(NC)=J
         ALP(NC,NV)=1
         IF(LEN(2,J).EQ.0) TEMP=0
         IF(LEN(2,J).NE.0) TEMP=LEN(1,J)
         BLP(NC)=(LENGTH(J)-TEMP)/1000
         GO TO 463
  456    J=JNV(NV)
         NC=NC+1
         JPT(NC)=J
```

```
         ALP(NC,NV)=1
         IF(LEN(2,J).EQ.0)  TEMP=0
         IF(LEN(2,J).NE.0)  TEMP=LEN(2,J)
         BLP(NC)=(LENGTH(J)-TEMP)/1000
  463 CONTINUE
      NUMCON=NC
C
C                              ADJUST COEFFICIENTS TO ORDER OF MAGNITUDE
C                              OF .01 TO 10
      DIV1=0
      DO 472 NV=1,NUMVAR
         TEST=ABS(CLP(NV))/10
         IF(TEST.GT.DIV1)  DIV1=TEST
  472 CONTINUE
C
      DO 476 NV=1,NUMVAR
         CLP(NV)=CLP(NV)/DIV1
  476 CONTINUE
C
C     WRITE(6,405)  (CLP(NV),NV=1,NUMVAR)
      DO 493 NC=1,NUMCON
         DIV2(NC)=0
         DO 484 NV=1,NUMVAR
            TEST=ABS(ALP(NC,NV))/10
            IF(TEST.GT.DIV2(NC))  DIV2(NC)=TEST
  484    CONTINUE
C
         DO 489 NV=1,NUMVAR
            ALP(NC,NV)=ALP(NC,NV)/DIV2(NC)
C           IF(ALP(NC,NV).LT.0.001)  ALP(NC,NV)=0
  489    CONTINUE
         BLP(NC)=BLP(NC)/DIV2(NC)
C        WRITE(6,405)  (ALP(NC,NV),NV=1,NUMVAR)
C        WRITE(6,405)  BLP(NC)
  493 CONTINUE
      INEQ=NUMCON
C     WRITE(6,496)  INEQ,NUMVAR
C 496 FORMAT(3X,'NUM INEQ=',I5,3X,'NUMVAR=',I5)
      KA=120
      EQUA=0
C
      CALL ZX3LP(ALP,KA,BLP,CLP,NUMVAR,INEQ,EQUA,S,PSOL,DSOL,RW,IW,IER)
C
      IT=IT+1
C                              CALCULATE CHANGE IN COST
      TOTAL1(IT)=TOTAL2(IT-1)-S*1000 *DIV1
C     DO 507  NV=1,NUMVAR
C        WRITE(6,*)  NV,JNV(NV),PSOL(NV)
C 507 CONTINUE
C                              COMPUTE THE CHANGE TO THE NEW SYSTEM
      NV=0
      DO 542 J=1,PIPES
         IF(LEN(2,J).EQ.0.AND.ND(1,J).EQ.1)  GO TO 542
         IF(EXIST(J).NE.0)  GO TO 542
```

```
      IF(LEN(2,J).EQ.0) GO TO 525
      NV=NV+1
      LEN(1,J)=LEN(1,J)+PSOL(NV)*1000
      LEN(2,J)=LEN(2,J)-PSOL(NV)*1000
      NV=NV+1
      LEN(1,J)=LEN(1,J)-PSOL(NV)*1000
      LEN(2,J)=LEN(2,J)+PSOL(NV)*1000
      IF(LEN(1,J).LT.0.01) LEN(1,J)=0
      IF(LEN(2,J).LT.0.01) LEN(2,J)=0
      IF(LEN(1,J).GT.LENGTH(J)) LEN(1,J)=LENGTH(J)
      IF(LEN(2,J).GT.LENGTH(J)) LEN(2,J)=LENGTH(J)
      GO TO 542
525   IF(ND(1,J).LE.1) GO TO 534
      NV=NV+1
      IF(PSOL(NV).EQ.0) GO TO 534
      ND(2,J)=ND(1,J)
      ND(1,J)=ND(2,J)-1
      LEN(2,J)=LEN(1,J)-PSOL(NV)*1000
      LEN(1,J)=PSOL(NV)*1000
      IF(LEN(2,J).LT.0.01) LEN(2,J)=0
      IF(LEN(1,J).GT.LENGTH(J)) LEN(1,J)=LENGTH(J)
534   IF(ND(1,J).GE.NDIAM) GO TO 542
      NV=NV+1
      IF(PSOL(NV).EQ.0) GO TO 542
      ND(2,J)=ND(1,J)+1
      LEN(2,J)=LEN(2,J)+PSOL(NV)*1000
      LEN(1,J)=LEN(1,J)-PSOL(NV)*1000
      IF(LEN(1,J).LT.0.01) LEN(1,J)=0
      IF(LEN(2,J).GT.LENGTH(J)) LEN(2,J)=LENGTH(J)
542 CONTINUE
    NTEMP=HCON+1
C                              CHECK TO SEE IF LENGTH CONSTRAINT ARE
C                              BINDING
    SUM=0
    LENTES=0
    DO 550 NC=NTEMP,NUMCON
      SUM=SUM+DSOL(NC)
550 CONTINUE
    IF(SUM.NE.0) LENTES=1
C                              CALCULATE MAXIMUM WEIGHTING FOR EACH
C                              LINK
    DO 559 J=1,PIPES
      MAXFAC(J)=0
      DO 558 K=1,HCON
        IF(FACTOR(K,J).GT.MAXFAC(J)) MAXFAC(J)=FACTOR(K,J)
558   CONTINUE
559 CONTINUE
C                              CALCULATE TRUE COST AND REMOVE SUB-
C                              MINIMUM PIPES WITH LOW WEIGHTING
    DO 575 J=1,PIPES
      IF(LEN(2,J).EQ.0.AND.ND(1,J).EQ.1.AND.MAXFAC(J).LT.0.5)
   $   GO TO 575
      IF(LEN(1,J).GE.0.01) GO TO 570
      LEN(1,J)=LEN(2,J)
```

```
      LEN(2,J)=0
      ND(1,J)=ND(2,J)
      ND(2,J)=ND(1,J)+1
570   COS1=COSPIP(ND(1,J))
      IF(ND(1,J).EQ.EXIST(J))  COS1=0
      TOTAL2(IT)=TOTAL2(IT)+LEN(1,J)*COS1
      IF(LEN(2,J).EQ.0)  GO TO 575
      TOTAL2(IT)=TOTAL2(IT)+LEN(2,J)*COSPIP(ND(2,J))
575 CONTINUE
    WRITE(6,577)   IT,TOTAL1(IT),TOTAL2(IT)
577 FORMAT(5X,I5,6X,F15.0,F15.0)
    GO TO 132
C                             WRITE PRIMAL SOLUTION
580 WRITE(6,87)
    WRITE(6,582)
582 FORMAT(//30X,'FINAL RESULTS')
    IF(NTYPE.NE.1)  GO TO 623
    WRITE(6,585)
585 FORMAT(//3X,' PIPE    DIAMETER    LENGTH    DIAMETER    LENGTH',
   $              '  EXISTING    MAX %',/)
    REWIND11
    DO 599 J=1,PIPES
      WRITE(11,590)  J,ND(1,J),LEN(1,J),ND(2,J),LEN(2,J),EXIST(J)
590   FORMAT(2I8,F12.2,I8,F12.2,I8)
      IF(LEN(2,J).EQ.0.AND.ND(1,J).EQ.1)  GO TO 599
      D1=DIAM(ND(1,J))
      D2=0
      D3=0
      IF(EXIST(J).NE.0)D3=DIAM(EXIST(J))
      IF(LEN(2,J).NE.0)  D2=DIAM(ND(2,J))
      WRITE(6,598)  J,D1,LEN(1,J),D2,LEN(2,J),D3,MAXFAC(J)
598 FORMAT(I6,6X,F7.3,F12.2,F10.3,F10.2,2F10.3)
599 CONTINUE
C                             WRITE DUAL CONSTRAINTS
    WRITE(6,602)
602 FORMAT(///25X,'DUAL SOLUTION',/)
    WRITE(6,604)
604 FORMAT(///15X,'CONSTRAINTS   LOAD',8X,'NODE',12X,'DUAL SOL.')
    DO 610 NC=1,HCON
      DSOL(NC)=DSOL(NC)*DIV1
      DSOL(NC)=DSOL(NC)/DIV2(NC)
      WRITE(6,609)  NC,NLPT(NC),IPT(NC),DSOL(NC)
609   FORMAT(15X,I5,I11,3X,I9,3X,5X,F15.8)
610 CONTINUE
      WRITE(6,612)
612   FORMAT('1',///15X,'CONSTRAINT      LINK        DUAL SOL.')
      NTEMP=HCON+1
      DO 619  NC=NTEMP,NUMCON
      DSOL(NC)=DSOL(NC)*DIV1
      DSOL(NC)=DSOL(NC)/DIV2(NC)
      WRITE(6,618)  NC,JPT(NC),DSOL(NC)
618   FORMAT(15X,I5,3X,6X,I3,6X,F10.2)
619 CONTINUE
C
```

```
C                                  WRITE PRESSURES FOR EACH DEMAND PATTERN
      WRITE(6,87)
  623  WRITE(6,624)
  624 FORMAT(///15X,'CORRECTED PRESSURES AND FLOWS')
      REWIND 12
      WRITE(12,628) LOADS
      WRITE(12,628) NODES
  628 FORMAT(I7)
      WRITE(6,630)
  630 FORMAT(//15X,'NODE   DEMAND   SUPPLY     MINHED     INHEAD')
      DO 641 NL=1,LOADS
         WRITE(6,634) NL
         WRITE(12,628) NL
  634 FORMAT(/25X,'LOAD',I5)
         DO 640 I=1,NODES
            WRITE(6,639) I,DEMAND(NL,I),SUPPLY(NL,I),MINHED(NL,I)
     $                    ,INHEAD(NL,I)
            WRITE(12,103) I,MINHED(NL,I),DEMAND(NL,I),INHEAD(NL,I)
  639 FORMAT(12X,I5,2X,2F9.0,2F10.2)
  640    CONTINUE
  641 CONTINUE
C
C                                  WRITE PIPE FLOWS AND HYDRAULIC GRADIENTS
      WRITE(6,645)
  645 FORMAT(//15X,'PIPE           FLOWS          HGL')
      DO 653 NL=1,LOADS
         WRITE(6,634) NL
         DO 652 J=1,PIPES
         J1=(INHEAD(NL,NODE(1,J))-INHEAD(NL,NODE(2,J)))/LENGTH(J)*1000
            WRITE(6,651) J,FLOW(NL,J),J1
  651 FORMAT(12X,I5,F15.0,F15.6)
  652    CONTINUE
  653 CONTINUE
C      IF(IT.LE.15)GO TO 132
      STOP
      END
```

Appendix C

OUTPUT FOR MULTIPLE DEMAND PATTERN DESIGN EXAMPLE

***** NEWORK OPTIMIZER *****

UNITS= 24.0000000

HAZEN-WILLIAMS C= 130.

NUMBER OF CANDIDATE DIAMETERS 13

	DIAMETER	COST
1	0.125	58.00
2	0.150	62.00
3	0.200	71.70
4	0.250	88.90
5	0.300	112.30
6	0.350	138.70
7	0.400	169.00
8	0.450	207.00
9	0.500	248.00
10	0.550	297.00
11	0.600	347.00
12	0.650	405.00
13	0.700	470.00

NUMBER OF LINKS 37 NUMBER OF NODES 20

PIPE	FROM	TO	LENGTH
1	1	2	760.
2	1	4	520.
3	1	6	890.
4	2	3	1120.
5	2	5	610.
6	2	6	680.
7	3	5	680.
8	3	7	870.
9	4	8	860.
10	4	9	980.
11	5	7	890.
12	5	10	750.
13	6	9	620.
14	6	10	800.
15	7	12	730.
16	7	13	680.
17	8	9	480.
18	8	15	860.
19	9	11	800.

20	9	14	770.
21	10	11	350.
22	10	12	620.
23	11	12	670.
24	11	16	790.
25	11	18	1150.
26	12	13	750.
27	12	17	550.
28	13	17	700.
29	14	15	500.
30	14	16	450.
31	14	19	750.
32	15	19	720.
33	16	18	540.
34	16	19	700.
35	17	18	850.
36	18	20	750.
37	19	20	970.

1

INITIAL ASSUMPTION

PIPE	DIAMETER	LENGTH	DIAMETER	LENGTH	EXISTING
1	0.400	760.00	0.0	0.0	0.0
2	0.400	520.00	0.0	0.0	0.0
3	0.400	890.00	0.0	0.0	0.0
4	0.400	1120.00	0.0	0.0	0.0
5	0.400	610.00	0.0	0.0	0.0
6	0.400	680.00	0.0	0.0	0.0
7	0.400	680.00	0.0	0.0	0.0
8	0.400	870.00	0.0	0.0	0.0
9	0.400	860.00	0.0	0.0	0.0
10	0.400	980.00	0.0	0.0	0.0
11	0.400	890.00	0.0	0.0	0.0
12	0.400	750.00	0.0	0.0	0.0
13	0.400	620.00	0.0	0.0	0.0
14	0.400	800.00	0.0	0.0	0.0
15	0.400	730.00	0.0	0.0	0.0
16	0.400	680.00	0.0	0.0	0.0
17	0.400	480.00	0.0	0.0	0.0
18	0.400	860.00	0.0	0.0	0.0
19	0.400	800.00	0.0	0.0	0.0
20	0.400	770.00	0.0	0.0	0.0
21	0.400	350.00	0.0	0.0	0.0
22	0.400	620.00	0.0	0.0	0.0
23	0.400	670.00	0.0	0.0	0.0
24	0.400	790.00	0.0	0.0	0.0
25	0.400	1150.00	0.0	0.0	0.0
26	0.400	750.00	0.0	0.0	0.0
27	0.400	550.00	0.0	0.0	0.0
28	0.400	700.00	0.0	0.0	0.0
29	0.400	500.00	0.0	0.0	0.0
30	0.400	450.00	0.0	0.0	0.0

31	0.400	750.00	0.0	0.0	0.0
32	0.400	720.00	0.0	0.0	0.0
33	0.400	540.00	0.0	0.0	0.0
34	0.400	700.00	0.0	0.0	0.0
35	0.400	850.00	0.0	0.0	0.0
36	0.400	750.00	0.0	0.0	0.0
37	0.400	970.00	0.0	0.0	0.0

INITIAL COST= 4590040.

1

NUMBER OF LOADING CONDITIONS 37
NUMBER OF NODES 20

1

INITIAL ASSUMPTION

NODE	MINHEAD	DEMAND	INHEAD
	LOAD 1		
1	75.00	235.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00
	LOAD 2		
1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	235.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00

11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 3

1	75.00	235.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 4

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	215.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

	LOAD	5	
1	75.00	165.0	80.00
2	74.00	290.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

	LOAD	6	
1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	230.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

	LOAD	7	
1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	215.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00

9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 8

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	265.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 9

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	235.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00

19	70.00	185.0	90.00
20	67.00	160.0	70.00
	LOAD 10		
1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	235.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00
	LOAD 11		
1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	265.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00
	LOAD 12		
1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00

7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	250.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 13

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	230.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 14

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	230.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00

17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00
	LOAD	15	
1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	210.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00
	LOAD	16	
1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	260.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00
	LOAD	17	
1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00

5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	250.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 18

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	250.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 19

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	230.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00

15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 20

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	230.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 21

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	270.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 22

1	75.00	165.0	80.00
2	74.00	220.0	90.00

3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	210.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 23

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	210.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 24

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	270.0	93.00
12	70.00	160.0	85.00

13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 25

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	270.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 26

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	260.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 27

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	235.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 28

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	260.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 29

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00

11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	220.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 30

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	320.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 31

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	305.0	90.00
20	67.00	160.0	70.00

LOAD 32

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	220.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 33

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	260.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 34

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00

9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	305.0	90.00
20	67.00	160.0	70.00

LOAD 35

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	235.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	160.0	70.00

LOAD 36

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00

19	70.00	185.0	90.00
20	67.00	280.0	70.00

LOAD 37

1	75.00	165.0	80.00
2	74.00	220.0	90.00
3	73.00	145.0	90.00
4	72.00	165.0	70.00
5	102.00	-800.0	102.00
6	73.00	140.0	80.00
7	67.00	175.0	90.00
8	72.00	180.0	70.00
9	70.00	140.0	75.00
10	69.00	160.0	90.00
11	71.00	170.0	93.00
12	70.00	160.0	85.00
13	64.00	190.0	80.00
14	73.00	200.0	90.00
15	73.00	150.0	80.00
16	96.00	-800.0	96.00
17	67.00	165.0	80.00
18	70.00	140.0	90.00
19	70.00	185.0	90.00
20	67.00	280.0	70.00

PIPES OUT OF SERVICE IN EACH CONDITION

LOAD 1
PIPE 1

LOAD 2
PIPE 2

LOAD 3
PIPE 3

LOAD 4
PIPE 4

LOAD 5
PIPE 5

LOAD 6
PIPE 6

LOAD 7
PIPE 7

LOAD 8
PIPE 8

LOAD 9

```
        PIPE     9

LOAD     10
        PIPE    10

LOAD     11
        PIPE    11

LOAD     12
        PIPE    12

LOAD     13
        PIPE    13

LOAD     14
        PIPE    14

LOAD     15
        PIPE    15

LOAD     16
        PIPE    16

LOAD     17
        PIPE    17

LOAD     18
        PIPE    18

LOAD     19
        PIPE    19

LOAD     20
        PIPE    20

LOAD     21
        PIPE    21

LOAD     22
        PIPE    22

LOAD     23
        PIPE    23

LOAD     24
        PIPE    24

LOAD     25
        PIPE    25

LOAD     26
        PIPE    26

LOAD     27
```

PIPE 27

LOAD 28
PIPE 28

LOAD 29
PIPE 29

LOAD 30
PIPE 30

LOAD 31
PIPE 31

LOAD 32
PIPE 32

LOAD 33
PIPE 33

LOAD 34
PIPE 34

LOAD 35
PIPE 35

LOAD 36
PIPE 36

LOAD 37
PIPE 37

1

ITERATION	COST 1	COST 2
2	3767093.	3767064.
3	3050041.	3050042.
4	2414499.	2414501.
5	2133649.	2133649.
6	2101664.	2101664.
7	2069679.	2175942.
8	2113005.	2113006.
9	2110889.	2110887.
10	2108599.	2066259.
11	2068086.	2068084.
12	2068111.	2068115.
13	2070268.	2070266.
14	2072391.	2072391.
15	2074021.	2074025.
16	2075025.	2075023.
17	2075265.	2075268.
18	2075058.	2075055.
19	2074888.	2074886.

20	2074646.	2074649.
21	2074458.	2074457.
22	2074589.	2074594.
23	2074763.	2074762.

1

FINAL RESULTS

PIPE	DIAMETER	LENGTH	DIAMETER	LENGTH	EXISTING	MAX %
1	0.200	350.98	0.250	409.02	0.0	1.000
2	0.150	250.77	0.200	269.23	0.0	0.703
3	0.200	753.42	0.250	136.58	0.0	0.894
4	0.200	1120.00	0.0	0.0	0.0	0.550
5	0.300	610.00	0.0	0.0	0.0	0.892
6	0.200	680.00	0.0	0.0	0.0	0.777
7	0.250	680.00	0.0	0.0	0.0	0.771
8	0.150	130.92	0.200	739.08	0.0	0.867
9	0.125	89.63	0.150	770.37	0.0	0.297
10	0.200	980.00	0.0	0.0	0.0	0.759
11	0.200	890.00	0.0	0.0	0.0	0.383
12	0.350	577.84	0.400	172.16	0.0	1.000
13	0.200	620.00	0.0	0.0	0.0	0.440
14	0.300	800.00	0.0	0.0	0.0	0.885
16	0.150	507.91	0.200	172.09	0.0	0.611
17	0.150	123.33	0.200	356.67	0.0	0.754
18	0.200	860.00	0.0	0.0	0.0	0.748
19	0.200	800.00	0.0	0.0	0.0	0.338
20	0.200	677.52	0.250	92.48	0.0	0.720
21	0.200	350.00	0.0	0.0	0.0	0.483
22	0.200	620.00	0.0	0.0	0.0	1.000
23	0.200	600.07	0.250	69.93	0.0	1.000
24	0.250	790.00	0.0	0.0	0.0	0.653
25	0.150	118.84	0.200	1031.16	0.0	0.612
26	0.150	123.94	0.200	626.06	0.0	0.686
27	0.150	150.86	0.200	399.14	0.0	0.759
28	0.150	700.00	0.0	0.0	0.0	0.389
29	0.200	500.00	0.0	0.0	0.0	0.952
30	0.250	314.92	0.300	135.08	0.0	0.955
31	0.200	647.03	0.250	102.97	0.0	0.643
32	0.200	720.00	0.0	0.0	0.0	1.000
33	0.250	540.00	0.0	0.0	0.0	0.848
34	0.300	606.71	0.350	93.29	0.0	1.000
35	0.150	466.31	0.200	383.69	0.0	0.808
36	0.200	546.60	0.250	203.40	0.0	1.000
37	0.200	569.24	0.250	400.76	0.0	1.000

DUAL SOLUTION

CONSTRAINTS	LOAD	NODE	DUAL SOL.
1	12	1	0.79868931
2	12	4	3.87035275
3	34	15	0.08093327
4	10	4	0.19100928
5	11	7	0.13621670
6	17	8	0.73620003
7	26	13	0.40397751
8	27	17	0.39013046
9	18	8	0.30809450
10	37	20	0.87695795
11	16	13	0.25092107
12	35	17	0.22638708
13	30	14	2.39629555
14	36	20	0.80073583
15	1	1	0.72842526
16	33	18	0.36467516
17	5	1	2.36910343
18	30	15	0.09528291
19	22	12	0.76084524
20	2	4	0.0
21	12	12	0.0
22	30	8	0.0
23	12	6	0.0
24	34	8	0.0
25	12	8	0.0
26	5	2	0.0
27	34	19	0.0
28	12	17	0.0
29	22	17	0.0
30	30	4	0.0
31	5	4	0.0
32	23	12	0.0
33	1	4	0.0
34	33	17	0.0
35	12	10	0.0
36	12	9	0.0
37	29	15	0.0

1

CONSTRAINT	LINK	DUAL SOL.
38	1	0.0
39	1	0.0
40	2	0.0
41	2	0.0
42	3	0.0
43	3	0.0
44	4	0.0
45	4	0.0
46	5	0.0
47	5	0.0

48	6	0.0
49	6	0.0
50	7	0.0
51	7	0.0
52	8	0.0
53	8	0.0
54	9	0.0
55	9	0.0
56	10	0.0
57	10	0.0
58	11	0.0
59	11	0.0
60	12	0.0
61	12	0.0
62	13	0.0
63	13	0.0
64	14	0.0
65	14	0.0
66	16	0.0
67	16	0.0
68	17	0.0
69	17	0.0
70	18	0.0
71	18	0.0
72	19	0.0
73	19	0.0
74	20	0.0
75	20	0.0
76	21	0.0
77	21	0.0
78	22	0.0
79	22	0.0
80	23	0.0
81	23	0.0
82	24	0.0
83	24	0.0
84	25	0.0
85	25	0.0
86	26	0.0
87	26	0.0
88	27	0.0
89	27	0.0
90	28	0.0
91	28	0.0
92	29	0.0
93	29	0.0
94	30	0.0
95	30	0.0
96	31	0.0
97	31	0.0
98	32	0.0
99	32	0.0
100	33	0.0
101	33	0.0

102	34	0.0
103	34	0.0
104	35	0.0
105	35	0.0
106	36	0.0
107	36	0.0
108	37	0.0
109	37	0.0

1

CORRECTED PRESSURES AND FLOWS

NODE	DEMAND	SUPPLY	MINHED	INHEAD
	LOAD	1		
1	235.	235.	75.00	75.01
2	220.	220.	74.00	97.10
3	145.	145.	73.00	97.03
4	165.	165.	72.00	75.44
5	-800.	-1635.	102.00	102.00
6	140.	140.	73.00	88.73
7	175.	175.	67.00	91.19
8	180.	180.	72.00	79.54
9	140.	140.	70.00	83.80
10	160.	160.	69.00	92.71
11	170.	170.	71.00	89.11
12	160.	160.	70.00	84.26
13	190.	190.	64.00	81.05
14	200.	200.	73.00	88.60
15	150.	150.	73.00	84.61
16	-800.	-1443.	96.00	96.00
17	165.	165.	67.00	81.15
18	140.	140.	70.00	89.22
19	185.	185.	70.00	89.61
20	160.	160.	67.00	87.52
	LOAD	2		
1	165.	165.	75.00	91.66
2	220.	220.	74.00	95.81
3	145.	145.	73.00	96.07
4	235.	235.	72.00	72.09
5	-800.	-1654.	102.00	102.00
6	140.	140.	73.00	91.86
7	175.	175.	67.00	90.89
8	180.	180.	72.00	78.98
9	140.	140.	70.00	83.99
10	160.	160.	69.00	93.99
11	170.	170.	71.00	89.54
12	160.	160.	70.00	84.87
13	190.	190.	64.00	81.46
14	200.	200.	73.00	88.61

15	150.	150.	73.00	84.47
16	-800.	-1425.	96.00	96.00
17	165.	165.	67.00	81.60
18	140.	140.	70.00	89.54
19	185.	185.	70.00	89.63
20	160.	160.	67.00	87.70
	LOAD	3		
1	235.	235.	75.00	81.50
2	220.	220.	74.00	94.24
3	145.	145.	73.00	95.29
4	165.	165.	72.00	80.97
5	-800.	-1720.	102.00	102.00
6	140.	140.	73.00	92.28
7	175.	175.	67.00	90.61
8	180.	180.	72.00	82.22
9	140.	140.	70.00	86.30
10	160.	160.	69.00	94.25
11	170.	170.	71.00	89.98
12	160.	160.	70.00	85.17
13	190.	190.	64.00	81.66
14	200.	200.	73.00	89.43
15	150.	150.	73.00	85.88
16	-800.	-1360.	96.00	96.00
17	165.	165.	67.00	81.85
18	140.	140.	70.00	89.91
19	185.	185.	70.00	90.16
20	160.	160.	67.00	88.14
	LOAD	4		
1	165.	165.	75.00	88.92
2	220.	220.	74.00	94.76
3	215.	215.	73.00	94.34
4	165.	165.	72.00	84.22
5	-800.	-1744.	102.00	102.00
6	140.	140.	73.00	91.39
7	175.	175.	67.00	90.16
8	180.	180.	72.00	84.08
9	140.	140.	70.00	87.14
10	160.	160.	69.00	93.89
11	170.	170.	71.00	90.03
12	160.	160.	70.00	85.02
13	190.	190.	64.00	81.48
14	200.	200.	73.00	89.81
15	150.	150.	73.00	86.68
16	-800.	-1335.	96.00	96.00
17	165.	165.	67.00	81.71
18	140.	140.	70.00	89.96
19	185.	185.	70.00	90.42
20	160.	160.	67.00	88.29
	LOAD	5		
1	165.	165.	75.00	75.04
2	290.	290.	74.00	75.37

3	145.	145.	73.00	89.73
4	165.	165.	72.00	74.96
5	-800.	-1535.	102.00	102.00
6	140.	140.	73.00	81.38
7	175.	175.	67.00	87.35
8	180.	180.	72.00	77.72
9	140.	140.	70.00	80.72
10	160.	160.	69.00	89.93
11	170.	170.	71.00	87.44
12	160.	160.	70.00	82.05
13	190.	190.	64.00	78.67
14	200.	200.	73.00	87.77
15	150.	150.	73.00	83.60
16	-800.	-1543.	96.00	96.00
17	165.	165.	67.00	78.99
18	140.	140.	70.00	88.27
19	185.	185.	70.00	89.08
20	160.	160.	67.00	86.75
	LOAD	6		
1	165.	165.	75.00	87.10
2	220.	220.	74.00	96.42
3	145.	145.	73.00	96.46
4	165.	165.	72.00	82.68
5	-800.	-1700.	102.00	102.00
6	230.	230.	73.00	87.49
7	175.	175.	67.00	90.94
8	180.	180.	72.00	82.66
9	140.	140.	70.00	85.43
10	160.	160.	69.00	92.33
11	170.	170.	71.00	89.25
12	160.	160.	70.00	84.17
13	190.	190.	64.00	80.95
14	200.	200.	73.00	89.26
15	150.	150.	73.00	85.91
16	-800.	-1400.	96.00	96.00
17	165.	165.	67.00	81.08
18	140.	140.	70.00	89.35
19	185.	185.	70.00	90.04
20	160.	160.	67.00	87.78
	LOAD	7		
1	165.	165.	75.00	86.73
2	220.	220.	74.00	91.34
3	215.	215.	73.00	77.95
4	165.	165.	72.00	83.04
5	-800.	-1684.	102.00	102.00
6	140.	140.	73.00	89.75
7	175.	175.	67.00	79.06
8	180.	180.	72.00	83.04
9	140.	140.	70.00	86.14
10	160.	160.	69.00	92.95
11	170.	169.	71.00	89.23
12	160.	160.	70.00	82.73

13	190.	190.	64.00	76.87
14	200.	200.	73.00	89.44
15	150.	150.	73.00	86.12
16	-800.	-1394.	96.00	96.00
17	165.	165.	67.00	78.64
18	140.	139.	70.00	89.24
19	185.	185.	70.00	90.12
20	160.	160.	67.00	87.75
	LOAD	8		
1	165.	165.	75.00	89.31
2	220.	220.	74.00	95.79
3	145.	145.	73.00	98.04
4	165.	165.	72.00	84.23
5	-800.	-1730.	102.00	102.00
6	140.	139.	73.00	91.44
7	265.	265.	67.00	75.31
8	180.	180.	72.00	84.02
9	140.	140.	70.00	87.00
10	160.	160.	69.00	93.60
11	170.	170.	71.00	89.43
12	160.	160.	70.00	82.27
13	190.	190.	64.00	74.86
14	200.	200.	73.00	89.74
15	150.	150.	73.00	86.60
16	-800.	-1369.	96.00	96.00
17	165.	165.	67.00	77.76
18	140.	140.	70.00	89.39
19	185.	185.	70.00	90.32
20	160.	160.	67.00	87.92
	LOAD	9		
1	165.	165.	75.00	87.98
2	220.	220.	74.00	94.95
3	145.	145.	73.00	95.61
4	235.	235.	72.00	80.41
5	-800.	-1714.	102.00	102.00
6	140.	140.	73.00	90.99
7	175.	175.	67.00	90.71
8	180.	180.	72.00	83.07
9	140.	140.	70.00	86.10
10	160.	160.	69.00	93.69
11	170.	170.	71.00	89.79
12	160.	160.	70.00	84.90
13	190.	190.	64.00	81.47
14	200.	200.	73.00	89.47
15	150.	150.	73.00	86.16
16	-800.	-1364.	96.00	96.00
17	165.	165.	67.00	81.65
18	140.	140.	70.00	89.78
19	185.	185.	70.00	90.20
20	160.	160.	67.00	88.09
	LOAD	10		

1	165.	165.	75.00	86.15
2	220.	220.	74.00	94.62
3	145.	145.	73.00	95.46
4	235.	235.	72.00	71.99
5	-800.	-1729.	102.00	102.00
6	140.	140.	73.00	90.93
7	175.	175.	67.00	90.67
8	180.	180.	72.00	80.92
9	140.	140.	70.00	87.57
10	160.	160.	69.00	93.72
11	170.	170.	71.00	90.05
12	160.	160.	70.00	85.04
13	190.	190.	64.00	81.58
14	200.	200.	73.00	89.62
15	150.	150.	73.00	85.59
16	-800.	-1350.	96.00	96.00
17	165.	165.	67.00	81.77
18	140.	140.	70.00	89.95
19	185.	185.	70.00	90.19
20	160.	160.	67.00	88.18
	LOAD	11		
1	165.	165.	75.00	87.96
2	220.	220.	74.00	93.70
3	145.	145.	73.00	92.75
4	165.	165.	72.00	83.44
5	-800.	-1682.	102.00	102.00
6	140.	139.	73.00	90.40
7	265.	265.	67.00	66.99
8	180.	180.	72.00	83.37
9	140.	140.	70.00	86.32
10	160.	160.	69.00	92.88
11	170.	169.	71.00	88.74
12	160.	160.	70.00	79.70
13	190.	190.	64.00	68.38
14	200.	200.	73.00	89.48
15	150.	150.	73.00	86.23
16	-800.	-1416.	96.00	96.00
17	165.	165.	67.00	74.41
18	140.	139.	70.00	88.74
19	185.	185.	70.00	90.10
20	160.	160.	67.00	87.45
	LOAD	12		
1	165.	165.	75.00	74.86
2	220.	220.	74.00	89.58
3	145.	145.	73.00	92.88
4	165.	165.	72.00	71.90
5	-800.	-1209.	102.00	102.00
6	140.	141.	73.00	73.96
7	175.	175.	67.00	86.88
8	180.	180.	72.00	73.06
9	140.	141.	70.00	74.15
10	250.	250.	69.00	72.57

11	170.	170.	71.00	76.43
12	160.	160.	70.00	70.21
13	190.	190.	64.00	69.11
14	200.	200.	73.00	85.92
15	150.	150.	73.00	81.14
16	-800.	-1892.	96.00	96.00
17	165.	165.	67.00	69.20
18	140.	140.	70.00	84.48
19	185.	185.	70.00	87.65
20	160.	160.	67.00	83.82
	LOAD	13		
1	165.	165.	75.00	89.01
2	220.	220.	74.00	95.62
3	145.	145.	73.00	95.96
4	165.	165.	72.00	79.57
5	-800.	-1633.	102.00	102.00
6	140.	140.	73.00	93.36
7	175.	175.	67.00	90.84
8	180.	180.	72.00	79.30
9	230.	230.	70.00	80.47
10	160.	160.	69.00	94.59
11	170.	170.	71.00	89.25
12	160.	160.	70.00	84.90
13	190.	190.	64.00	81.44
14	200.	200.	73.00	87.95
15	150.	150.	73.00	84.19
16	-800.	-1467.	96.00	96.00
17	165.	165.	67.00	81.58
18	140.	140.	70.00	89.30
19	185.	185.	70.00	89.36
20	160.	160.	67.00	87.44
	LOAD	14		
1	165.	166.	75.00	81.17
2	220.	220.	74.00	92.02
3	145.	145.	73.00	94.44
4	165.	165.	72.00	78.42
5	-800.	-1673.	102.00	102.00
6	230.	231.	73.00	81.14
7	175.	175.	67.00	90.35
8	180.	180.	72.00	79.02
9	140.	140.	70.00	81.29
10	160.	160.	69.00	96.80
11	170.	170.	71.00	89.97
12	160.	160.	70.00	85.92
13	190.	190.	64.00	82.11
14	200.	200.	73.00	88.11
15	150.	150.	73.00	84.22
16	-800.	-1428.	96.00	96.00
17	165.	165.	67.00	82.32
18	140.	140.	70.00	89.86
19	185.	185.	70.00	89.48
20	160.	160.	67.00	87.78

	LOAD	15		
1	165.	165.	75.00	89.06
2	220.	220.	74.00	95.20
3	145.	145.	73.00	95.69
4	165.	165.	72.00	84.19
5	-800.	-1705.	102.00	102.00
6	140.	140.	73.00	91.36
7	175.	175.	67.00	90.47
8	180.	180.	72.00	84.02
9	140.	140.	70.00	87.03
10	160.	160.	69.00	93.69
11	170.	170.	71.00	89.69
12	210.	210.	70.00	83.32
13	190.	190.	64.00	80.30
14	200.	200.	73.00	89.77
15	150.	150.	73.00	86.63
16	-800.	-1353.	96.00	96.00
17	165.	165.	67.00	80.48
18	140.	140.	70.00	89.68
19	185.	185.	70.00	90.36
20	160.	160.	67.00	88.10

	LOAD	16		
1	165.	165.	75.00	88.81
2	220.	220.	74.00	95.38
3	145.	145.	73.00	96.88
4	165.	165.	72.00	83.76
5	-800.	-1656.	102.00	102.00
6	140.	140.	73.00	90.87
7	175.	175.	67.00	95.66
8	180.	180.	72.00	83.59
9	140.	140.	70.00	86.48
10	160.	160.	69.00	92.94
11	170.	170.	71.00	88.56
12	160.	160.	70.00	78.31
13	260.	260.	64.00	64.01
14	200.	200.	73.00	89.53
15	150.	150.	73.00	86.32
16	-800.	-1422.	96.00	96.00
17	165.	165.	67.00	72.48
18	140.	140.	70.00	88.56
19	185.	185.	70.00	90.11
20	160.	160.	67.00	87.35

	LOAD	17		
1	165.	165.	75.00	88.50
2	220.	220.	74.00	95.15
3	145.	145.	73.00	95.72
4	165.	165.	72.00	81.46
5	-800.	-1689.	102.00	102.00
6	140.	139.	73.00	91.52
7	175.	175.	67.00	90.81
8	250.	250.	72.00	72.00

9	140.	140.	70.00	87.76
10	160.	160.	69.00	93.97
11	170.	170.	71.00	90.13
12	160.	160.	70.00	85.17
13	190.	190.	64.00	81.70
14	200.	200.	73.00	89.04
15	150.	150.	73.00	82.85
16	-800.	-1389.	96.00	96.00
17	165.	165.	67.00	81.88
18	140.	140.	70.00	89.93
19	185.	185.	70.00	89.61
20	160.	160.	67.00	87.89
	LOAD	18		
1	165.	165.	75.00	87.39
2	220.	220.	74.00	94.72
3	145.	145.	73.00	95.49
4	165.	165.	72.00	79.27
5	-800.	-1743.	102.00	102.00
6	140.	140.	73.00	90.37
7	175.	175.	67.00	90.59
8	250.	250.	72.00	72.00
9	140.	140.	70.00	83.93
10	160.	160.	69.00	93.37
11	170.	170.	71.00	89.43
12	160.	160.	70.00	84.59
13	190.	190.	64.00	81.21
14	200.	200.	73.00	90.00
15	150.	150.	73.00	88.82
16	-800.	-1336.	96.00	96.00
17	165.	165.	67.00	81.38
18	140.	140.	70.00	89.55
19	185.	185.	70.00	90.80
20	160.	160.	67.00	88.22
	LOAD	19		
1	165.	165.	75.00	87.46
2	220.	220.	74.00	94.71
3	145.	145.	73.00	95.52
4	165.	166.	72.00	79.97
5	-800.	-1723.	102.00	102.00
6	140.	140.	73.00	90.21
7	175.	175.	67.00	90.80
8	180.	180.	72.00	79.87
9	230.	230.	70.00	81.34
10	160.	160.	69.00	93.73
11	170.	170.	71.00	91.15
12	160.	160.	70.00	85.56
13	190.	190.	64.00	82.02
14	200.	200.	73.00	88.24
15	150.	150.	73.00	84.55
16	-800.	-1378.	96.00	96.00
17	165.	165.	67.00	82.23
18	140.	140.	70.00	90.36

19	185.	185.	70.00	89.63
20	160.	160.	67.00	88.11
	LOAD	20		
1	165.	165.	75.00	87.35
2	220.	220.	74.00	94.62
3	145.	145.	73.00	95.43
4	165.	165.	72.00	80.29
5	-800.	-1763.	102.00	102.00
6	140.	139.	73.00	89.86
7	175.	175.	67.00	90.50
8	180.	180.	72.00	80.29
9	230.	230.	70.00	81.51
10	160.	160.	69.00	93.08
11	170.	170.	71.00	88.97
12	160.	160.	70.00	84.25
13	190.	190.	64.00	80.93
14	200.	199.	73.00	90.66
15	150.	150.	73.00	85.99
16	-800.	-1335.	96.00	96.00
17	165.	165.	67.00	81.08
18	140.	140.	70.00	89.29
19	185.	184.	70.00	90.68
20	160.	160.	67.00	88.02
	LOAD	21		
1	165.	165.	75.00	89.20
2	220.	220.	74.00	95.36
3	145.	145.	73.00	95.77
4	165.	165.	72.00	83.21
5	-800.	-1602.	102.00	102.00
6	140.	140.	73.00	92.09
7	175.	175.	67.00	90.44
8	180.	180.	72.00	82.96
9	140.	140.	70.00	85.49
10	160.	160.	69.00	95.49
11	270.	270.	71.00	85.66
12	160.	160.	70.00	83.20
13	190.	190.	64.00	80.02
14	200.	200.	73.00	89.22
15	150.	150.	73.00	85.94
16	-800.	-1507.	96.00	96.00
17	165.	165.	67.00	80.10
18	140.	140.	70.00	88.03
19	185.	185.	70.00	89.88
20	160.	160.	67.00	86.93
	LOAD	22		
1	165.	165.	75.00	89.50
2	220.	220.	74.00	95.33
3	145.	145.	73.00	95.45
4	165.	165.	72.00	84.15
5	-800.	-1627.	102.00	102.00
6	140.	140.	73.00	92.36

7	175.	175.	67.00	88.02
8	180.	180.	72.00	83.89
9	140.	140.	70.00	86.81
10	160.	160.	69.00	95.55
11	170.	169.	71.00	88.26
12	210.	210.	70.00	70.03
13	190.	190.	64.00	69.26
14	200.	200.	73.00	89.61
15	150.	150.	73.00	86.45
16	-800.	-1432.	96.00	96.00
17	165.	165.	67.00	69.44
18	140.	139.	70.00	88.26
19	185.	185.	70.00	90.13
20	160.	160.	67.00	87.17
	LOAD	23		
1	165.	165.	75.00	88.98
2	220.	220.	74.00	95.04
3	145.	145.	73.00	95.34
4	165.	165.	72.00	84.43
5	-800.	-1755.	102.00	102.00
6	140.	140.	73.00	91.18
7	175.	175.	67.00	88.56
8	180.	180.	72.00	84.28
9	140.	140.	70.00	87.45
10	160.	160.	69.00	93.28
11	170.	170.	71.00	91.39
12	210.	210.	70.00	73.22
13	190.	190.	64.00	72.11
14	200.	200.	73.00	89.90
15	150.	150.	73.00	86.78
16	-800.	-1304.	96.00	96.00
17	165.	165.	67.00	72.36
18	140.	140.	70.00	89.79
19	185.	185.	70.00	90.44
20	160.	160.	67.00	88.20
	LOAD	24		
1	165.	165.	75.00	87.06
2	220.	220.	74.00	94.38
3	145.	145.	73.00	95.18
4	165.	165.	72.00	80.88
5	-800.	-1857.	102.00	102.00
6	140.	140.	73.00	89.12
7	175.	175.	67.00	89.67
8	180.	180.	72.00	80.80
9	140.	140.	70.00	82.79
10	160.	160.	69.00	91.51
11	270.	270.	71.00	82.60
12	160.	160.	70.00	80.33
13	190.	190.	64.00	77.69
14	200.	200.	73.00	88.38
15	150.	150.	73.00	84.78
16	-800.	-1253.	96.00	96.00

17	165.	165.	67.00	77.76
18	140.	140.	70.00	86.96
19	185.	185.	70.00	89.29
20	160.	160.	67.00	86.03
	LOAD	25		
1	165.	165.	75.00	88.83
2	220.	220.	74.00	95.12
3	145.	145.	73.00	95.67
4	165.	165.	72.00	83.81
5	-800.	-1719.	102.00	102.00
6	140.	140.	73.00	91.11
7	175.	175.	67.00	90.61
8	180.	180.	72.00	83.65
9	140.	140.	70.00	86.52
10	160.	160.	69.00	93.45
11	270.	270.	71.00	88.38
12	160.	160.	70.00	84.16
13	190.	190.	64.00	80.92
14	200.	200.	73.00	89.61
15	150.	150.	73.00	86.42
16	-800.	-1389.	96.00	96.00
17	165.	165.	67.00	81.09
18	140.	140.	70.00	89.65
19	185.	185.	70.00	90.27
20	160.	160.	67.00	88.05
	LOAD	26		
1	165.	165.	75.00	89.10
2	220.	220.	74.00	95.03
3	145.	145.	73.00	95.15
4	165.	165.	72.00	84.31
5	-800.	-1735.	102.00	102.00
6	140.	140.	73.00	91.54
7	175.	175.	67.00	86.85
8	180.	180.	72.00	84.13
9	140.	140.	70.00	87.20
10	160.	160.	69.00	94.03
11	170.	170.	71.00	90.06
12	160.	160.	70.00	85.84
13	260.	260.	64.00	64.00
14	200.	200.	73.00	89.82
15	150.	150.	73.00	86.69
16	-800.	-1343.	96.00	96.00
17	165.	165.	67.00	76.57
18	140.	140.	70.00	89.65
19	185.	185.	70.00	90.39
20	160.	160.	67.00	88.10
	LOAD	27		
1	165.	165.	75.00	89.13
2	220.	220.	74.00	95.18
3	145.	145.	73.00	95.51
4	165.	165.	72.00	84.25

5	-800.	-1705.	102.00	102.00
6	140.	140.	73.00	91.53
7	175.	175.	67.00	89.31
8	180.	180.	72.00	84.06
9	140.	140.	70.00	87.11
10	160.	160.	69.00	93.97
11	170.	170.	71.00	89.84
12	160.	160.	70.00	85.69
13	190.	190.	64.00	75.38
14	200.	200.	73.00	89.74
15	150.	150.	73.00	86.60
16	-800.	-1374.	96.00	96.00
17	235.	235.	67.00	67.00
18	140.	140.	70.00	88.92
19	185.	185.	70.00	90.27
20	160.	160.	67.00	87.63
	LOAD	28		
1	165.	165.	75.00	89.01
2	220.	220.	74.00	95.11
3	145.	145.	73.00	95.45
4	165.	165.	72.00	84.17
5	-800.	-1726.	102.00	102.00
6	140.	140.	73.00	91.33
7	175.	175.	67.00	89.13
8	180.	180.	72.00	84.01
9	140.	140.	70.00	87.02
10	160.	160.	69.00	93.67
11	170.	169.	71.00	89.70
12	160.	160.	70.00	83.28
13	260.	260.	64.00	74.65
14	200.	200.	73.00	89.77
15	150.	150.	73.00	86.62
16	-800.	-1353.	96.00	96.00
17	165.	165.	67.00	80.94
18	140.	139.	70.00	89.70
19	185.	185.	70.00	90.36
20	160.	160.	67.00	88.12
	LOAD	29		
1	165.	165.	75.00	88.14
2	220.	220.	74.00	94.95
3	145.	145.	73.00	95.61
4	165.	165.	72.00	81.38
5	-800.	-1718.	102.00	102.00
6	140.	140.	73.00	90.88
7	175.	175.	67.00	90.68
8	180.	180.	72.00	77.72
9	140.	140.	70.00	85.41
10	160.	160.	69.00	93.61
11	170.	170.	71.00	89.64
12	160.	160.	70.00	84.78
13	190.	190.	64.00	81.37
14	200.	200.	73.00	90.22

15	220.	220.	73.00	77.66
16	-800.	-1360.	96.00	96.00
17	165.	165.	67.00	81.54
18	140.	139.	70.00	89.63
19	185.	185.	70.00	89.87
20	160.	160.	67.00	87.86
	LOAD	30		
1	165.	165.	75.00	85.24
2	220.	220.	74.00	93.91
3	145.	145.	73.00	95.06
4	165.	165.	72.00	74.78
5	-800.	-1847.	102.00	102.00
6	140.	140.	73.00	88.18
7	175.	175.	67.00	90.08
8	180.	180.	72.00	72.33
9	140.	140.	70.00	76.39
10	160.	160.	69.00	92.10
11	170.	170.	71.00	87.45
12	160.	160.	70.00	83.01
13	190.	190.	64.00	79.76
14	320.	320.	73.00	73.01
15	150.	149.	73.00	73.03
16	-800.	-1282.	96.00	96.00
17	165.	165.	67.00	79.84
18	140.	140.	70.00	87.55
19	185.	185.	70.00	84.23
20	160.	160.	67.00	83.71
	LOAD	31		
1	165.	165.	75.00	88.96
2	220.	220.	74.00	95.20
3	145.	145.	73.00	95.74
4	165.	165.	72.00	83.77
5	-800.	-1693.	102.00	102.00
6	140.	140.	73.00	91.37
7	175.	175.	67.00	90.77
8	180.	180.	72.00	83.43
9	140.	140.	70.00	86.65
10	160.	160.	69.00	93.85
11	170.	170.	71.00	89.83
12	160.	160.	70.00	84.96
13	190.	190.	64.00	81.52
14	200.	200.	73.00	88.79
15	150.	150.	73.00	85.64
16	-800.	-1435.	96.00	96.00
17	165.	165.	67.00	81.68
18	140.	140.	70.00	89.68
19	305.	305.	70.00	88.94
20	160.	160.	67.00	87.42
	LOAD	32		
1	165.	165.	75.00	88.09
2	220.	220.	74.00	94.93

3	145.	145.	73.00	95.60
4	165.	165.	72.00	81.35
5	-800.	-1720.	102.00	102.00
6	140.	140.	73.00	90.81
7	175.	175.	67.00	90.69
8	180.	180.	72.00	78.40
9	140.	140.	70.00	85.04
10	160.	160.	69.00	93.60
11	170.	170.	71.00	89.71
12	160.	160.	70.00	84.83
13	190.	190.	64.00	81.42
14	200.	200.	73.00	88.17
15	220.	220.	73.00	78.42
16	-800.	-1360.	96.00	96.00
17	165.	165.	67.00	81.60
18	140.	140.	70.00	89.78
19	185.	185.	70.00	91.30
20	160.	160.	67.00	88.56
	LOAD	33		
1	165.	165.	75.00	87.62
2	220.	220.	74.00	94.54
3	145.	145.	73.00	95.10
4	165.	165.	72.00	81.91
5	-800.	-1837.	102.00	102.00
6	140.	139.	73.00	89.75
7	175.	175.	67.00	88.68
8	180.	180.	72.00	81.60
9	140.	140.	70.00	84.41
10	160.	160.	69.00	92.03
11	170.	170.	71.00	85.51
12	160.	160.	70.00	78.58
13	190.	190.	64.00	73.18
14	200.	199.	73.00	87.64
15	150.	150.	73.00	84.22
16	-800.	-1290.	96.00	96.00
17	165.	165.	67.00	70.46
18	260.	260.	70.00	70.02
19	185.	184.	70.00	87.61
20	160.	160.	67.00	71.88
	LOAD	34		
1	165.	165.	75.00	86.24
2	220.	220.	74.00	94.24
3	145.	145.	73.00	95.18
4	165.	165.	72.00	77.28
5	-800.	-1821.	102.00	102.00
6	140.	140.	73.00	89.04
7	175.	175.	67.00	90.03
8	180.	180.	72.00	73.03
9	140.	140.	70.00	80.27
10	160.	160.	69.00	92.34
11	170.	170.	71.00	87.21
12	160.	160.	70.00	82.73

13	190.	190.	64.00	79.30
14	200.	200.	73.00	80.63
15	150.	150.	73.00	72.98
16	-800.	-1309.	96.00	96.00
17	165.	165.	67.00	79.30
18	140.	140.	70.00	85.66
19	305.	305.	70.00	71.87
20	160.	160.	67.00	73.37

LOAD 35

1	165.	165.	75.00	88.79
2	220.	220.	74.00	94.97
3	145.	145.	73.00	95.30
4	165.	165.	72.00	84.04
5	-800.	-1762.	102.00	102.00
6	140.	139.	73.00	91.00
7	175.	175.	67.00	88.56
8	180.	180.	72.00	83.91
9	140.	140.	70.00	86.87
10	160.	160.	69.00	93.19
11	170.	170.	71.00	89.46
12	160.	160.	70.00	79.04
13	190.	190.	64.00	72.19
14	200.	200.	73.00	89.80
15	150.	150.	73.00	86.65
16	-800.	-1316.	96.00	96.00
17	235.	235.	67.00	67.01
18	140.	140.	70.00	90.94
19	185.	185.	70.00	90.51
20	160.	160.	67.00	88.84

LOAD 36

1	165.	165.	75.00	88.94
2	220.	220.	74.00	95.22
3	145.	145.	73.00	95.77
4	165.	165.	72.00	83.51
5	-800.	-1681.	102.00	102.00
6	140.	139.	73.00	91.43
7	175.	175.	67.00	90.92
8	180.	180.	72.00	82.96
9	140.	140.	70.00	86.44
10	160.	160.	69.00	94.04
11	170.	170.	71.00	90.53
12	160.	160.	70.00	85.53
13	190.	190.	64.00	82.11
14	200.	199.	73.00	88.00
15	150.	150.	73.00	84.78
16	-800.	-1447.	96.00	96.00
17	165.	165.	67.00	82.38
18	140.	140.	70.00	91.23
19	185.	185.	70.00	87.58
20	280.	280.	67.00	67.01

LOAD 37

1	165.	165.	75.00	89.01
2	220.	220.	74.00	95.16
3	145.	145.	73.00	95.66
4	165.	165.	72.00	84.24
5	-800.	-1716.	102.00	102.00
6	140.	140.	73.00	91.25
7	175.	175.	67.00	90.37
8	180.	180.	72.00	84.17
9	140.	140.	70.00	86.94
10	160.	160.	69.00	93.53
11	170.	170.	71.00	88.76
12	160.	160.	70.00	83.74
13	190.	190.	64.00	79.90
14	200.	200.	73.00	90.24
15	150.	150.	73.00	87.21
16	-800.	-1413.	96.00	96.00
17	165.	165.	67.00	79.89
18	140.	140.	70.00	84.96
19	185.	185.	70.00	91.58
20	280.	280.	67.00	67.00

PIPE	FLOWS	HGL
	LOAD 1	
1	0.	-29.054561
2	-25.	-0.817900
3	-210.	-15.405959
4	10.	0.058774
5	-405.	-8.039005
6	176.	12.309063
7	-238.	-7.308264
8	103.	6.709886
9	-46.	-4.765018
10	-144.	-8.532528
11	174.	12.142944
12	817.	12.388855
13	139.	7.942126
14	-313.	-4.977894
15	0.	9.495252
16	102.	14.921458
17	-108.	-8.883254
18	-118.	-5.898214
19	-126.	-6.636105
20	-127.	-6.235554
21	159.	10.278713
22	185.	13.624376
23	137.	7.238155
24	-262.	-8.720495
25	-11.	-0.097484
26	80.	4.286743
27	83.	5.651634
28	-8.	-0.152370
29	139.	7.985748

30	-410.	-16.437107
31	-56.	-1.346883
32	-129.	-6.948662
33	319.	12.550156
34	451.	9.123622
35	-90.	-9.494198
36	78.	2.270996
37	81.	2.158552
	LOAD 2	
1	-144.	-5.449737
2	0.	37.646162
3	-21.	-0.222950
4	-21.	-0.233378
5	-460.	-10.153924
6	117.	5.799080
7	-262.	-8.724280
8	97.	5.945982
9	-61.	-8.015105
10	-174.	-12.149313
11	177.	12.478106
12	755.	10.684469
13	179.	12.690489
14	-223.	-2.654896
15	0.	8.249863
16	99.	13.875557
17	-117.	-10.444450
18	-123.	-6.376488
19	-129.	-6.927490
20	-125.	-5.993236
21	179.	12.714364
22	193.	14.700908
23	134.	6.961993
24	-253.	-8.181492
25	-1.	-0.000146
26	82.	4.550639
27	85.	5.940191
28	-9.	-0.208391
29	142.	8.288696
30	-410.	-16.423509
31	-56.	-1.363363
32	-131.	-7.176208
33	311.	11.968909
34	451.	9.097225
35	-89.	-9.331539
36	82.	2.454936
37	78.	1.996243
	LOAD 3	
1	-263.	-16.757021
2	28.	1.014680
3	0.	-12.113250
4	-44.	-0.941072
5	-519.	-12.728932

6	80.	2.874330
7	-280.	-9.868599
8	91.	5.374198
9	-24.	-1.451324
10	-113.	-5.438871
11	179.	12.793483
12	741.	10.334920
13	154.	9.642472
14	-214.	-2.460003
15	0.	7.460419
16	96.	13.170871
17	-105.	-8.504073
18	-99.	-4.256741
19	-103.	-4.602413
20	-101.	-4.063832
21	175.	12.184012
22	193.	14.646961
23	137.	7.189122
24	-244.	-7.614677
25	9.	0.060929
26	84.	4.680115
27	86.	6.030690
28	-11.	-0.276010
29	131.	7.100616
30	-385.	-14.596388
31	-47.	-0.968709
32	-118.	-5.940056
33	301.	11.269746
34	430.	8.345490
35	-90.	-9.486497
36	80.	2.361491
37	80.	2.077256

LOAD 4

1	-173.	-7.683784
2	91.	9.032880
3	-83.	-2.772488
4	0.	0.376252
5	-500.	-11.867786
6	107.	4.959062
7	-301.	-11.265811
8	86.	4.801345
9	8.	0.171307
10	-82.	-2.973159
11	183.	13.301035
12	759.	10.809021
13	128.	6.856315
14	-244.	-3.130932
15	0.	7.037312
16	94.	12.769699
17	-90.	-6.377125
18	-82.	-3.024345
19	-91.	-3.614426
20	-93.	-3.476547

21	166.	11.040344
22	190.	14.303860
23	139.	7.469063
24	-243.	-7.558084
25	8.	0.059443
26	84.	4.728210
27	86.	6.026362
28	-12.	-0.330941
29	122.	6.274048
30	-372.	-13.745524
31	-42.	-0.803446
32	-110.	-5.193901
33	300.	11.183788
34	420.	7.975573
35	-91.	-9.706367
36	78.	2.230977
37	82.	2.195441

LOAD 5

1	-37.	-0.433069
2	10.	0.156227
3	-139.	-7.130912
4	-180.	-12.827478
5	0.	-43.660523
6	-147.	-8.849088
7	-388.	-18.038447
8	64.	2.745530
9	-37.	-3.210538
10	-118.	-5.876767
11	205.	16.466016
12	942.	16.093933
13	47.	1.078255
14	-473.	-10.681362
15	0.	7.259839
16	94.	12.757761
17	-89.	-6.246185
18	-128.	-6.841793
19	-143.	-8.406448
20	-157.	-9.164825
21	130.	7.109855
22	179.	12.716109
23	145.	8.053042
24	-295.	-10.834049
25	-33.	-0.716619
26	82.	4.500793
27	82.	5.555337
28	-14.	-0.457371
29	142.	8.342285
30	-434.	-18.282539
31	-65.	-1.747803
32	-136.	-7.613881
33	343.	14.323680
34	471.	9.880415
35	-97.	-10.911865

36	74.	2.016947
37	86.	2.403306
	LOAD 6	
1	-222.	-12.263047
2	88.	8.505278
3	-31.	-0.440310
4	-8.	-0.036975
5	-435.	-9.147994
6	0.	13.129470
7	-253.	-8.145388
8	100.	6.346323
9	2.	0.014815
10	-79.	-2.811837
11	177.	12.427152
12	836.	12.899801
13	87.	3.321002
14	-347.	-6.041832
15	0.	9.267206
16	102.	14.685844
17	-85.	-5.767377
18	-93.	-3.770997
19	-105.	-4.774513
20	-113.	-4.964873
21	146.	8.779646
22	182.	13.145767
23	140.	7.578357
24	-259.	-8.541426
25	-10.	-0.082371
26	80.	4.295085
27	83.	5.619673
28	-9.	-0.186419
29	126.	6.696472
30	-390.	-14.987522
31	-49.	-1.042643
32	-116.	-5.736415
33	316.	12.320370
34	435.	8.517718
35	-91.	-9.721231
36	75.	2.091349
37	85.	2.328979
	LOAD 7	
1	-152.	-6.064706
2	80.	7.093283
3	-93.	-3.392595
4	173.	11.956610
5	-616.	-17.475366
6	72.	2.337893
7	0.	-35.369671
8	-42.	-1.274354
9	-1.	-0.002910
10	-84.	-3.165872
11	262.	25.778301

12	806.	12.063110
13	117.	5.815100
14	-278.	-4.003010
15	0.	-5.028795
16	45.	3.215251
17	-91.	-6.458441
18	-90.	-3.579393
19	-94.	-3.861046
20	-104.	-4.281933
21	162.	10.625523
22	206.	16.490862
23	161.	9.709555
24	-260.	-8.564894
25	-3.	-0.004299
26	110.	7.809855
27	96.	7.427396
28	-35.	-2.531891
29	126.	6.637726
30	-384.	-14.573364
31	-45.	-0.907511
32	-114.	-5.554856
33	319.	12.520967
34	432.	8.396258
35	-104.	-12.465192
36	73.	1.979390
37	87.	2.441737

LOAD 8

1	-183.	-8.537212
2	95.	9.761077
3	-77.	-2.400533
4	-66.	-2.003670
5	-460.	-10.174235
6	123.	6.399716
7	-211.	-5.826726
8	0.	26.125625
9	9.	0.245879
10	-79.	-2.822347
11	284.	29.990413
12	774.	11.203532
13	131.	7.171508
14	-225.	-2.694302
15	0.	-9.534256
16	19.	0.666450
17	-89.	-6.202825
18	-82.	-3.003852
19	-83.	-3.043175
20	-94.	-3.566455
21	172.	11.906390
22	217.	18.272277
23	169.	10.688918
24	-256.	-8.316310
25	6.	0.036780
26	125.	9.884257

27	101.	8.205566
28	-46.	-4.143045
29	122.	6.280426
30	-375.	-13.907233
31	-41.	-0.768311
32	-110.	-5.161730
33	315.	12.244783
34	424.	8.117174
35	-109.	-13.685105
36	72.	1.953084
37	88.	2.469045

LOAD 9

1	-190.	-9.172219
2	118.	14.563722
3	-93.	-3.389046
4	-34.	-0.591087
5	-493.	-11.558308
6	117.	5.815641
7	-273.	-9.394926
8	94.	5.636790
9	0.	-3.097818
10	-117.	-5.812478
11	179.	12.688266
12	769.	11.074849
13	138.	7.892190
14	-254.	-3.373833
15	0.	7.958357
16	97.	13.580031
17	-90.	-6.316884
18	-90.	-3.598022
19	-103.	-4.615231
20	-105.	-4.372574
21	166.	11.142970
22	190.	14.187130
23	138.	7.307434
24	-248.	-7.855920
25	3.	0.011146
26	82.	4.566427
27	85.	5.907593
28	-10.	-0.250920
29	125.	6.609375
30	-383.	-14.514398
31	-47.	-0.973348
32	-115.	-5.603748
33	305.	11.516656
34	429.	8.287811
35	-90.	-9.567458
36	78.	2.254557
37	82.	2.173654

LOAD 10

1	-211.	-11.150099
2	165.	27.223499

3	-119.	-5.370048
4	-39.	-0.749016
5	-505.	-12.096593
6	113.	5.433430
7	-276.	-9.617682
8	93.	5.507090
9	-70.	-10.384706
10	0.	-15.892714
11	179.	12.731676
12	768.	11.033488
13	113.	5.420488
14	-259.	-3.498173
15	0.	7.716537
16	97.	13.363154
17	-137.	-13.841693
18	-113.	-5.429574
19	-84.	-3.107929
20	-80.	-2.663828
21	161.	10.493992
22	188.	14.014755
23	140.	7.486941
24	-242.	-7.529131
25	10.	0.086272
26	83.	4.605164
27	85.	5.932090
28	-11.	-0.273176
29	140.	8.051453
30	-379.	-14.184909
31	-41.	-0.760966
32	-123.	-6.383959
33	300.	11.198567
34	429.	8.303550
35	-90.	-9.623162
36	80.	2.367147
37	80.	2.072269

LOAD 11

1	-171.	-7.552498
2	89.	8.680373
3	-83.	-2.750817
4	41.	0.849533
5	-539.	-13.612116
6	106.	4.840694
7	-333.	-13.610099
8	230.	29.601332
9	5.	0.081280
10	-81.	-2.936336
11	0.	39.334861
12	809.	12.164062
13	125.	6.587761
14	-242.	-3.090019
15	0.	-17.403778
16	-35.	-2.047191
17	-88.	-6.140645

18	-87.	-3.326398
19	-82.	-3.019485
20	-102.	-4.108043
21	172.	11.830967
22	236.	21.258422
23	192.	13.491616
24	-270.	-9.194792
25	-2.	-0.002468
26	157.	15.083557
27	111.	9.619917
28	-68.	-8.602448
29	124.	6.500000
30	-383.	-14.480625
31	-43.	-0.820699
32	-112.	-5.368784
33	331.	13.446384
34	432.	8.429653
35	-122.	-16.862560
36	67.	1.716248
37	92.	2.729357

LOAD 12

1	-285.	-19.361918
2	71.	5.699451
3	48.	1.014538
4	-81.	-2.945709
5	-669.	-20.361728
6	246.	22.967641
7	-331.	-13.413912
8	105.	6.897253
9	-23.	-1.350261
10	-71.	-2.295000
11	209.	16.991090
12	0.	39.244080
13	-24.	-0.303748
14	178.	1.743011
15	0.	22.831308
16	139.	26.122194
17	-51.	-2.266407
18	-152.	-9.397746
19	-80.	-2.855339
20	-206.	-15.285779
21	-166.	-11.048584
22	93.	3.799783
23	157.	9.287865
24	-461.	-24.767159
25	-112.	-6.993846
26	45.	1.461650
27	45.	1.838129
28	-7.	-0.121809
29	153.	9.551727
30	-485.	-22.400614
31	-75.	-2.310954
32	-149.	-9.040387

33	425.	21.339134
34	522.	11.924373
35	-127.	-17.972663
36	47.	0.877726
37	113.	3.952954

LOAD 13

1	-185.	-8.695542
2	133.	18.143228
3	-113.	-4.887407
4	-24.	-0.305217
5	-467.	-10.465778
6	87.	3.321794
7	-265.	-8.885709
8	96.	5.876756
9	11.	0.321286
10	-43.	-0.914048
11	177.	12.533775
12	723.	9.876139
13	0.	20.787910
14	-166.	-1.544800
15	0.	8.140439
16	98.	13.827425
17	-54.	-2.441820
18	-116.	-5.693320
19	-165.	-10.973225
20	-162.	-9.717867
21	197.	15.273612
22	200.	15.629799
23	129.	6.484644
24	-259.	-8.547935
25	-7.	-0.046015
26	83.	4.613505
27	86.	6.048418
28	-9.	-0.190713
29	135.	7.517151
30	-429.	-17.885980
31	-67.	-1.878662
32	-131.	-7.177183
33	317.	12.407317
34	461.	9.485277
35	-88.	-9.087363
36	82.	2.477132
37	78.	1.977429

LOAD 14

1	-241.	-14.268735
2	68.	5.286760
3	8.	0.032472
4	-69.	-2.166339
5	-595.	-16.364001
6	202.	15.989909
7	-299.	-11.111383
8	85.	4.700549

9	-16.	-0.689733
10	-81.	-2.920672
11	182.	13.084514
12	598.	6.939880
13	-20.	-0.229103
14	0.	-19.562836
15	0.	6.079791
16	92.	12.121358
17	-77.	-4.727268
18	-120.	-6.045213
19	-164.	-10.848312
20	-154.	-8.855963
21	225.	19.513070
22	213.	17.546057
23	124.	6.043255
24	-244.	-7.638588
25	11.	0.094857
26	87.	5.072367
27	90.	6.531677
28	-11.	-0.302647
29	137.	7.778595
30	-425.	-17.542318
31	-66.	-1.833008
32	-133.	-7.311185
33	303.	11.376981
34	456.	9.313267
35	-86.	-8.861551
36	87.	2.762756
37	73.	1.748815

LOAD 15

1	-178.	-8.080272
2	93.	9.357159
3	-80.	-2.589588
4	-29.	-0.439603
5	-484.	-11.150373
6	115.	5.641578
7	-271.	-9.278488
8	97.	5.998511
9	8.	0.199961
10	-80.	-2.901365
11	181.	12.952895
12	770.	11.082092
13	129.	6.979222
14	-234.	-2.908039
15	0.	9.793802
16	103.	14.955745
17	-89.	-6.281885
18	-82.	-3.029562
19	-87.	-3.319397
20	-94.	-3.549412
21	169.	11.422947
22	207.	16.719326
23	159.	9.504404

24	-250.	-7.986836
25	3.	0.013083
26	77.	4.027242
27	79.	5.161715
28	-10.	-0.259269
29	122.	6.285858
30	-374.	-13.849046
31	-42.	-0.791768
32	-110.	-5.189938
33	307.	11.712307
34	422.	8.054635
35	-96.	-10.813940
36	75.	2.093811
37	85.	2.326556

LOAD 16

1	-184.	-8.643060
2	95.	9.706820
3	-76.	-2.315153
4	-53.	-1.339844
5	-477.	-10.858329
6	126.	6.629764
7	-242.	-7.533758
8	44.	1.400827
9	8.	0.196554
10	-79.	-2.774749
11	131.	7.125477
12	806.	12.080444
13	130.	7.078675
14	-220.	-2.589359
15	0.	23.764352
16	0.	46.540361
17	-87.	-6.017272
18	-85.	-3.176685
19	-76.	-2.601776
20	-100.	-3.962133
21	177.	12.510986
22	250.	23.595675
23	205.	15.299213
24	-273.	-9.416682
25	-1.	0.000013
26	178.	19.065959
27	117.	10.596397
28	-82.	-12.102073
29	123.	6.414368
30	-381.	-14.377238
31	-42.	-0.773112
32	-111.	-5.259747
33	336.	13.776285
34	432.	8.414176
35	-130.	-18.915854
36	65.	1.616842
37	95.	2.847322

	LOAD 17	
1	-185.	-8.749470
2	113.	13.532815
3	-93.	-3.397704
4	-32.	-0.509875
5	-486.	-11.231619
6	112.	5.331825
7	-270.	-9.235629
8	94.	5.648856
9	72.	11.006448
10	-124.	-6.427562
11	178.	12.578351
12	755.	10.700928
13	120.	6.067756
14	-241.	-3.064041
15	0.	7.721846
16	97.	13.393492
17	0.	-32.842827
18	-178.	-12.617493
19	-81.	-2.966156
20	-62.	-1.657481
21	165.	10.972334
22	190.	14.203201
23	139.	7.411444
24	-240.	-7.425332
25	15.	0.177413
26	83.	4.627502
27	86.	5.983970
28	-10.	-0.256348
29	176.	12.379547
30	-397.	-15.472616
31	-41.	-0.766663
32	-152.	-9.395515
33	301.	11.240811
34	451.	9.125257
35	-90.	-9.473913
36	87.	2.723816
37	73.	1.778577

	LOAD 18	
1	-195.	-9.645161
2	122.	15.613791
3	-92.	-3.343732
4	-37.	-0.687490
5	-502.	-11.935250
6	123.	6.403530
7	-276.	-9.574284
8	94.	5.626003
9	62.	8.448508
10	-105.	-4.756554
11	179.	12.814760
12	786.	11.505920
13	160.	10.376887
14	-269.	-3.756828

15	0.	8.226661
16	98.	13.802584
17	-188.	-24.848207
18	0.	-19.557243
19	-128.	-6.868210
20	-144.	-7.878252
21	167.	11.270185
22	189.	14.163159
23	137.	7.218796
24	-256.	-8.321525
25	-12.	-0.108908
26	82.	4.507060
27	84.	5.841814
28	-10.	-0.238996
29	72.	2.348328
30	-366.	-13.338487
31	-50.	-1.072408
32	-78.	-2.747875
33	311.	11.942150
34	404.	7.425733
35	-90.	-9.617453
36	69.	1.775167
37	91.	2.661982

LOAD 19

1	-194.	-9.540016
2	117.	14.412660
3	-88.	-3.081435
4	-38.	-0.722354
5	-502.	-11.945230
6	126.	6.629316
7	-275.	-9.525815
8	92.	5.432182
9	6.	0.119569
10	-54.	-1.401707
11	178.	12.588261
12	768.	11.030558
13	190.	14.295787
14	-293.	-4.402008
15	0.	7.170272
16	95.	12.910663
17	-61.	-3.076045
18	-113.	-5.451185
19	0.	-12.261505
20	-155.	-8.952133
21	133.	7.359401
22	182.	13.169246
23	148.	8.342003
24	-217.	-6.137606
25	32.	0.692391
26	84.	4.726603
27	86.	6.057850
28	-11.	-0.304478
29	133.	7.363251

30	-421.	-17.255046
31	-67.	-1.865987
32	-130.	-7.057105
33	289.	10.453627
34	451.	9.093257
35	-90.	-9.558482
36	91.	2.999552
37	69.	1.576642

LOAD 20

1	-194.	-9.557624
2	114.	13.579677
3	-84.	-2.820339
4	-38.	-0.725760
5	-506.	-12.105748
6	129.	6.990724
7	-277.	-9.664199
8	94.	5.659573
9	-1.	-0.001118
10	-51.	-1.242813
11	180.	12.916273
12	800.	11.895426
13	184.	13.473511
14	-279.	-4.020786
15	0.	8.563107
16	99.	14.083144
17	-55.	-2.535407
18	-126.	-6.629447
19	-151.	-9.325428
20	0.	-11.882574
21	171.	11.742467
22	190.	14.233841
23	135.	7.037490
24	-265.	-8.900548
25	-20.	-0.282646
26	81.	4.433960
27	84.	5.764354
28	-10.	-0.221536
29	151.	9.330505
30	-344.	-11.871541
31	-7.	-0.027181
32	-125.	-6.507831
33	317.	12.419241
34	409.	7.602583
35	-91.	-9.659478
36	67.	1.703328
37	93.	2.744411

LOAD 21

1	-178.	-8.113199
2	104.	11.502251
3	-91.	-3.255968
4	-26.	-0.362369
5	-477.	-10.883194

6	106.	4.806204
7	-269.	-9.166022
8	98.	6.124141
9	10.	0.293572
10	-71.	-2.326701
11	181.	12.989773
12	674.	8.674744
13	162.	10.643251
14	-287.	-4.251137
15	0.	9.913468
16	104.	15.324222
17	-81.	-5.276330
18	-89.	-3.461119
19	-19.	-0.206261
20	-111.	-4.838503
21	0.	28.099191
22	227.	19.825277
23	95.	3.667097
24	-326.	-13.089588
25	-58.	-2.061953
26	79.	4.244853
27	83.	5.637429
28	-7.	-0.118648
29	125.	6.563446
30	-391.	-15.066969
31	-45.	-0.882996
32	-113.	-5.477736
33	348.	14.758386
34	441.	8.739842
35	-89.	-9.327985
36	62.	1.463643
37	98.	3.040589

	LOAD 22	
1	-173.	-7.665895
2	98.	10.290733
3	-90.	-3.215113
4	-14.	-0.114659
5	-479.	-10.941327
6	100.	4.359750
7	-277.	-9.626164
8	118.	8.542922
9	10.	0.305832
10	-78.	-2.716142
11	200.	15.705769
12	671.	8.594076
13	148.	8.952922
14	-278.	-3.991604
15	0.	24.649382
16	143.	27.589641
17	-88.	-6.093407
18	-82.	-2.986571
19	-62.	-1.811066
20	-95.	-3.642154

21	233.	20.843549
22	0.	41.171978
23	280.	27.211021
24	-279.	-9.798480
25	-1.	0.000902
26	37.	1.022542
27	34.	1.062123
28	-10.	-0.261056
29	123.	6.321686
30	-379.	-14.189317
31	-39.	-0.685689
32	-109.	-5.104319
33	343.	14.336734
34	431.	8.387037
35	-142.	-22.134723
36	62.	1.454122
37	98.	3.053095

LOAD 23

1	-176.	-7.974745
2	90.	8.751209
3	-78.	-2.468306
4	-22.	-0.268132
5	-490.	-11.407321
6	116.	5.682373
7	-279.	-9.791408
8	112.	7.800047
9	8.	0.172761
10	-83.	-3.081357
11	196.	15.105841
12	790.	11.627584
13	119.	6.012406
14	-222.	-2.627239
15	0.	21.013924
16	133.	24.182780
17	-92.	-6.600634
18	-81.	-2.908307
19	-107.	-4.922543
20	-88.	-3.178485
21	113.	5.404140
22	296.	32.360766
23	0.	27.122725
24	-211.	-5.838148
25	47.	1.392928
26	45.	1.472168
27	41.	1.552318
28	-12.	-0.357644
29	122.	6.229187
30	-370.	-13.561639
31	-40.	-0.729289
32	-109.	-5.085500
33	305.	11.507416
34	419.	7.936816
35	-136.	-20.498980

36	75.	2.110372
37	85.	2.310322

LOAD 24

1	-195.	-9.632572
2	106.	11.889032
3	-75.	-2.309786
4	-38.	-0.713839
5	-514.	-12.488093
6	137.	7.742713
7	-283.	-10.026820
8	100.	6.333239
9	5.	0.087401
10	-65.	-1.947473
11	187.	13.851860
12	873.	13.992737
13	159.	10.208844
14	-237.	-2.985287
15	0.	12.796083
16	112.	17.626774
17	-71.	-4.132684
18	-104.	-4.624762
19	21.	0.236435
20	-138.	-7.266830
21	260.	25.448172
22	216.	18.023780
23	91.	3.384900
24	0.	-16.963814
25	-80.	-3.793189
26	72.	3.526754
27	75.	4.678456
28	-6.	-0.102735
29	131.	7.203705
30	-417.	-16.926236
31	-53.	-1.214864
32	-122.	-6.268056
33	373.	16.739344
34	463.	9.579511
35	-96.	-10.827296
36	56.	1.234375
37	104.	3.360174

LOAD 25

1	-180.	-8.268758
2	94.	9.666267
3	-80.	-2.554887
4	-31.	-0.495611
5	-487.	-11.282274
6	118.	5.897657
7	-272.	-9.304563
8	96.	5.824122
9	8.	0.182165
10	-78.	-2.766948
11	179.	12.802347

12	782.	11.401510
13	133.	7.401128
14	-235.	-2.926826
15	0.	8.826582
16	100.	14.241589
17	-87.	-5.975564
18	-85.	-3.222532
19	-72.	-2.328644
20	-100.	-4.017560
21	192.	14.477844
22	195.	14.978003
23	127.	6.297188
24	-277.	-9.643516
25	0.	-1.099389
26	80.	4.321167
27	82.	5.579307
28	-10.	-0.246081
29	123.	6.380829
30	-379.	-14.195048
31	-45.	-0.880310
32	-112.	-5.348121
33	308.	11.766815
34	426.	8.182199
35	-93.	-10.061215
36	76.	2.129089
37	84.	2.292122

LOAD 26

1	-174.	-7.805473
2	92.	9.219478
3	-83.	-2.742879
4	-14.	-0.108215
5	-490.	-11.422604
6	109.	5.133820
7	-283.	-10.068512
8	125.	9.539707
9	8.	0.201966
10	-81.	-2.953354
11	209.	17.018127
12	753.	10.631694
13	129.	7.001618
14	-243.	-3.106270
15	0.	1.394256
16	159.	33.610108
17	-90.	-6.391621
18	-82.	-2.969626
19	-90.	-3.578396
20	-92.	-3.396666
21	168.	11.323722
22	182.	13.209952
23	127.	6.308758
24	-242.	-7.515282
25	23.	0.362390
26	0.	29.116089

27	150.	16.844011
28	-101.	-17.961230
29	122.	6.259064
30	-372.	-13.743015
31	-41.	-0.759277
32	-110.	-5.137486
33	308.	11.766335
34	421.	8.021284
35	-116.	-15.381559
36	74.	2.063090
37	85.	2.356948

LOAD 27

1	-176.	-7.951837
2	93.	9.391902
3	-82.	-2.692422
4	-24.	-0.299999
5	-484.	-11.183292
6	112.	5.363442
7	-275.	-9.537955
8	107.	7.126995
9	9.	0.222335
10	-81.	-2.920439
11	190.	14.254264
12	755.	10.700907
13	131.	7.125830
14	-240.	-3.054085
15	0.	4.957643
16	122.	20.493137
17	-90.	-6.360912
18	-81.	-2.955308
19	-88.	-3.413811
20	-92.	-3.417097
21	172.	11.800668
22	184.	13.354345
23	126.	6.193224
24	-247.	-7.792297
25	35.	0.801325
26	150.	13.755005
27	0.	33.990895
28	81.	11.969626
29	122.	6.285675
30	-375.	-13.901774
31	-39.	-0.700480
32	-109.	-5.094719
33	327.	13.106367
34	426.	8.186340
35	-154.	-25.791680
36	68.	1.723348
37	92.	2.721145

LOAD 28

1	-177.	-8.025882
2	93.	9.299498

3	-81.	-2.609562
4	-24.	-0.307669
5	-487.	-11.299108
6	114.	5.554648
7	-277.	-9.629216
8	108.	7.263201
9	8.	0.192296
10	-81.	-2.910536
11	192.	14.457137
12	770.	11.100769
13	129.	6.945038
14	-235.	-2.930050
15	0.	8.017595
16	124.	21.298689
17	-89.	-6.286875
18	-83.	-3.038841
19	-87.	-3.345985
20	-94.	-3.559737
21	168.	11.351929
22	207.	16.764708
23	159.	9.583499
24	-250.	-7.973104
25	2.	0.004140
26	136.	11.507019
27	71.	4.260670
28	0.	-8.981280
29	122.	6.290588
30	-374.	-13.854540
31	-42.	-0.795898
32	-110.	-5.197525
33	307.	11.673171
34	422.	8.053741
35	-94.	-10.305355
36	75.	2.105876
37	85.	2.314743

LOAD 29

1	-188.	-8.967791
2	111.	12.997114
3	-88.	-3.077372
4	-34.	-0.585188
5	-493.	-11.551203
6	119.	5.995088
7	-273.	-9.398270
8	94.	5.663010
9	43.	4.251116
10	-97.	-4.115389
11	179.	12.716452
12	774.	11.182719
13	147.	8.813354
14	-256.	-3.419819
15	0.	8.083249
16	98.	13.695616
17	-148.	-16.018836

18	11.	0.075265
19	-111.	-5.280247
20	-127.	-6.243243
21	168.	11.359820
22	190.	14.244153
23	137.	7.246923
24	-251.	-8.054400
25	1.	0.002216
26	82.	4.549662
27	85.	5.900269
28	-10.	-0.238713
29	0.	25.122131
30	-359.	-12.844170
31	32.	0.471781
32	-209.	-16.954486
33	309.	11.788008
34	442.	8.762447
35	-90.	-9.527103
36	80.	2.368958
37	80.	2.070649

LOAD 30

1	-214.	-11.407230
2	140.	20.123379
3	-91.	-3.299790
4	-46.	-1.024546
5	-531.	-13.262364
6	143.	8.430414
7	-285.	-10.209633
8	95.	5.720187
9	35.	2.842074
10	-59.	-1.643481
11	184.	13.392262
12	846.	13.201925
13	222.	19.016709
14	-310.	-4.901600
15	0.	9.692571
16	103.	15.176526
17	-105.	-8.447488
18	-41.	-0.812513
19	-187.	-13.834248
20	105.	4.389815
21	183.	13.269261
22	193.	14.666527
23	131.	6.640306
24	-295.	-10.817323
25	-10.	-0.084985
26	80.	4.325948
27	84.	5.761136
28	-6.	-0.108337
29	-9.	-0.048248
30	0.	-51.096090
31	-205.	-14.964315
32	-199.	-15.554322

33	359.	15.644356
34	628.	16.814292
35	-88.	-9.076897
36	122.	5.118612
37	38.	0.532894

LOAD 31

1	-179.	-8.203968
2	96.	9.984442
3	-82.	-2.700909
4	-31.	-0.480924
5	-484.	-11.147621
6	115.	5.634128
7	-270.	-9.207961
8	95.	5.709137
9	12.	0.401732
10	-81.	-2.931400
11	178.	12.616138
12	762.	10.869141
13	136.	7.617655
14	-242.	-3.099251
15	0.	7.957082
16	98.	13.606756
17	-92.	-6.704712
18	-75.	-2.571088
19	-96.	-3.986282
20	-82.	-2.782173
21	169.	11.466631
22	191.	14.330932
23	137.	7.271428
24	-247.	-7.804021
25	13.	0.134755
26	83.	4.591899
27	85.	5.963801
28	-10.	-0.234048
29	122.	6.298798
30	-405.	-16.026510
31	0.	-0.202209
32	-103.	-4.584800
33	307.	11.703972
34	476.	10.086103
35	-89.	-9.408210
36	91.	3.013468
37	69.	1.566983

LOAD 32

1	-188.	-8.997967
2	111.	12.970029
3	-88.	-3.046409
4	-34.	-0.596523
5	-494.	-11.584823
6	120.	6.069340
7	-273.	-9.409759
8	94.	5.643858

9	38.	3.428561
10	-93.	-3.764281
11	179.	12.706508
12	774.	11.196167
13	151.	9.301167
14	-259.	-3.495960
15	0.	8.031181
16	98.	13.632157
17	-137.	-13.828246
18	-5.	-0.017920
19	-117.	-5.836391
20	-101.	-4.063138
21	166.	11.126796
22	189.	14.152305
23	137.	7.283658
24	-250.	-7.963929
25	-9.	-0.062853
26	82.	4.542806
27	85.	5.873413
28	-10.	-0.252468
29	225.	19.501526
30	-423.	-17.404446
31	-103.	-4.177429
32	0.	-17.894215
33	305.	11.517080
34	382.	6.712755
35	-90.	-9.626716
36	66.	1.631897
37	94.	2.829090

LOAD 33

1	-189.	-9.104116
2	101.	10.988588
3	-77.	-2.388343
4	-31.	-0.495352
5	-508.	-12.223341
6	130.	7.049269
7	-285.	-10.149182
8	109.	7.382588
9	11.	0.360427
10	-75.	-2.555022
11	195.	14.971118
12	849.	13.296326
13	145.	8.606080
14	-232.	-2.846870
15	0.	13.831841
16	129.	22.783594
17	-86.	-5.862268
18	-83.	-3.040828
19	-54.	-1.367779
20	-103.	-4.185922
21	220.	18.625837
22	238.	21.692411
23	166.	10.343660

24	-329.	-13.280110
25	159.	13.468960
26	105.	7.194132
27	139.	14.757025
28	44.	3.886806
29	128.	6.843872
30	-438.	-18.583001
31	8.	0.041219
32	-105.	-4.709752
33	0.	48.112205
34	523.	11.990378
35	19.	0.520809
36	-82.	-2.477661
37	242.	16.215547

LOAD 34

1	-205.	-10.530211
2	129.	17.223446
3	-89.	-3.152877
4	-41.	-0.842244
5	-519.	-12.721252
6	136.	7.642499
7	-283.	-10.024486
8	96.	5.923129
9	47.	4.942765
10	-83.	-3.050824
11	184.	13.449182
12	835.	12.874674
13	189.	14.149106
14	-283.	-4.126072
15	0.	9.996325
16	106.	15.785397
17	-143.	-15.084553
18	10.	0.058179
19	-145.	-8.667889
20	-31.	-0.466602
21	193.	14.682835
22	199.	15.501748
23	131.	6.674764
24	-299.	-11.132909
25	46.	1.347603
26	83.	4.582336
27	88.	6.240845
28	-2.	-0.006125
29	198.	15.299805
30	-609.	-34.155613
31	180.	11.680379
32	57.	1.542197
33	401.	19.156929
34	0.	34.471872
35	-79.	-7.476250
36	228.	16.381836
37	-68.	-1.545558

	LOAD 35	
1	-178.	-8.136047
2	92.	9.142391
3	-78.	-2.480873
4	-23.	-0.294794
5	-492.	-11.516709
6	117.	5.846203
7	-280.	-9.845621
8	112.	7.757411
9	7.	0.145899
10	-80.	-2.892662
11	196.	15.105584
12	794.	11.743551
13	126.	6.656794
14	-227.	-2.741184
15	0.	13.035437
16	133.	24.071570
17	-88.	-6.167253
18	-85.	-3.179772
19	-85.	-3.231888
20	-97.	-3.796763
21	163.	10.670428
22	245.	22.826090
23	207.	15.548547
24	-255.	-8.281408
25	-45.	-1.289580
26	120.	9.137065
27	172.	21.875166
28	63.	7.397919
29	122.	6.298370
30	-373.	-13.787367
31	-47.	-0.956604
32	-112.	-5.370331
33	272.	9.369066
34	416.	7.838375
35	0.	-28.155159
36	88.	2.797750
37	72.	1.722418

	LOAD 36	
1	-180.	-8.261530
2	99.	10.445932
3	-84.	-2.804154
4	-31.	-0.495447
5	-483.	-11.121756
6	114.	5.563332
7	-269.	-9.160838
8	93.	5.571089
9	15.	0.634695
10	-82.	-2.995721
11	177.	12.445188
12	752.	10.616109
13	140.	8.051251
14	-249.	-3.256569

15	0.	7.394451
16	95.	12.955138
17	-97.	-7.253424
18	-68.	-2.121220
19	-109.	-5.110283
20	-69.	-2.025931
21	157.	10.025155
22	186.	13.729169
23	139.	7.467583
24	-232.	-6.925172
25	-30.	-0.608414
26	82.	4.548726
27	83.	5.711670
28	-13.	-0.385895
29	124.	6.434723
30	-428.	-17.775879
31	35.	0.563822
32	-94.	-3.881243
33	264.	8.835574
34	524.	12.031446
35	-94.	-10.405148
36	0.	32.291646
37	280.	21.204054

LOAD 37

1	-178.	-8.097839
2	92.	9.175110
3	-79.	-2.519740
4	-29.	-0.441442
5	-485.	-11.209006
6	116.	5.752631
7	-272.	-9.328057
8	98.	6.080364
9	5.	0.080073
10	-78.	-2.758088
11	181.	13.070782
12	778.	11.294210
13	129.	6.952741
14	-232.	-2.848282
15	0.	9.082052
16	104.	15.388466
17	-85.	-5.774562
18	-90.	-3.539329
19	-71.	-2.278767
20	-104.	-4.284827
21	185.	13.618033
22	201.	15.793930
23	140.	7.501380
24	-269.	-9.160720
25	74.	3.304496
26	88.	5.112345
27	93.	6.989163
28	2.	0.013973
29	120.	6.054565

30	-358.	-12.801480
31	-65.	-1.787130
32	-120.	-6.066153
33	415.	20.439148
34	370.	6.314741
35	-70.	-5.964463
36	280.	23.950846
37	0.	25.340161

Appendix D

NEW YORK CITY WATER SUPPLY TUNNELS OUTPUT

**** HARDY CROSS ANALYSIS ONLY ***

UNITS= 0.0000151

HAZEN-WILLIAMS C= 100.

NUMBER OF CANDIDATE DIAMETERS 15

	DIAMETER	COST
1	36.000	93.50
2	48.000	134.00
3	60.000	176.00
4	72.000	221.00
5	84.000	267.00
6	96.000	316.00
7	108.000	365.00
8	120.000	417.00
9	132.000	469.00
10	144.000	522.00
11	156.000	577.00
12	168.000	632.00
13	180.000	689.00
14	192.000	746.00
15	204.000	804.00

NUMBER OF LINKS 42 NUMBER OF NODES 20

PIPE	FROM	TO	LENGTH
1	1	2	11600.
2	2	3	19800.
3	3	4	7300.
4	4	5	8300.
5	5	6	8600.
6	6	7	19100.
7	7	8	9600.
8	8	9	12500.
9	9	10	9600.
10	11	9	11200.
11	12	11	14500.
12	12	13	12200.
13	13	14	24100.
14	14	15	21100.
15	1	15	15500.
16	10	17	26400.
17	12	18	31200.
18	18	19	24000.
19	11	20	14400.

20	16	20	38400.
21	9	16	26400.
22	1	2	11600.
23	2	3	19800.
24	3	4	7300.
25	4	5	8300.
26	5	6	8600.
27	6	7	19100.
28	7	8	9600.
29	8	9	12500.
30	9	10	9600.
31	11	9	11200.
32	12	11	14500.
33	12	13	12200.
34	13	14	24100.
35	14	15	21100.
36	1	15	15500.
37	10	17	26400.
38	12	18	31200.
39	18	19	24000.
40	11	20	14400.
41	16	20	38400.
42	9	16	26400.

1

INITIAL ASSUMPTION

PIPE	DIAMETER	LENGTH	DIAMETER	LENGTH	EXISTING
1	180.000	11600.00	0.0	0.0	180.000
2	180.000	19800.00	0.0	0.0	180.000
3	180.000	7300.00	0.0	0.0	180.000
4	180.000	8300.00	0.0	0.0	180.000
5	180.000	8600.00	0.0	0.0	180.000
6	180.000	19100.00	0.0	0.0	180.000
7	132.000	9600.00	0.0	0.0	132.000
8	132.000	12500.00	0.0	0.0	132.000
9	180.000	9600.00	0.0	0.0	180.000
10	204.000	11200.00	0.0	0.0	204.000
11	204.000	14500.00	0.0	0.0	204.000
12	204.000	12200.00	0.0	0.0	204.000
13	204.000	24100.00	0.0	0.0	204.000
14	204.000	21100.00	0.0	0.0	204.000
15	204.000	15500.00	0.0	0.0	204.000
16	72.000	26400.00	0.0	0.0	72.000
17	72.000	31200.00	0.0	0.0	72.000
18	60.000	24000.00	0.0	0.0	60.000
19	60.000	14400.00	0.0	0.0	60.000
20	60.000	38400.00	0.0	0.0	60.000
21	72.000	26400.00	0.0	0.0	72.000
22	0.0	11600.00	0.0	0.0	0.0
23	0.0	19800.00	0.0	0.0	0.0
24	0.0	7300.00	0.0	0.0	0.0
25	0.0	8300.00	0.0	0.0	0.0

26	0.0	8600.00	0.0	0.0	0.0
27	0.0	19100.00	0.0	0.0	0.0
28	144.000	9600.00	0.0	0.0	0.0
29	0.0	12500.00	0.0	0.0	0.0
30	0.0	9600.00	0.0	0.0	0.0
31	0.0	11200.00	0.0	0.0	0.0
32	0.0	14500.00	0.0	0.0	0.0
33	0.0	12200.00	0.0	0.0	0.0
34	0.0	24100.00	0.0	0.0	0.0
35	0.0	21100.00	0.0	0.0	0.0
36	0.0	15500.00	0.0	0.0	0.0
37	96.000	26400.00	0.0	0.0	0.0
38	96.000	31200.00	0.0	0.0	0.0
39	84.000	24000.00	0.0	0.0	0.0
40	60.000	14400.00	0.0	0.0	0.0
41	0.0	38400.00	0.0	0.0	0.0
42	84.000	26400.00	0.0	0.0	0.0

INITIAL COST= 39204000.

1

NUMBER OF LOADING CONDITIONS 1
NUMBER OF NODES 20

1

INITIAL ASSUMPTION

NODE	MINHEAD	DEMAND	INHEAD
	LOAD	1	
1	300.00	-2020.0	300.00
2	255.00	92.4	294.19
3	255.00	92.4	286.11
4	255.00	88.2	283.74
5	255.00	88.2	281.64
6	255.00	88.2	280.01
7	255.00	88.2	277.44
8	255.00	88.2	276.59
9	255.00	170.0	273.68
10	255.00	1.0	273.65
11	255.00	170.0	273.81
12	255.00	117.1	275.09
13	255.00	117.1	278.06
14	255.00	92.4	285.54
15	255.00	92.4	293.31
16	260.00	170.0	261.55
17	272.80	57.5	272.77
18	255.00	117.0	261.14
19	255.00	117.1	255.01
20	255.00	170.0	258.02

PIPES OUT OF SERVICE IN EACH CONDITION

1

1

FINAL RESULTS

CORRECTED PRESSURES AND FLOWS

NODE	DEMAND	SUPPLY	MINHED	INHEAD
	LOAD	1		
1	-2020.	-2019.	300.00	300.00
2	92.	92.	255.00	294.19
3	92.	92.	255.00	286.11
4	88.	88.	255.00	283.74
5	88.	88.	255.00	281.64
6	88.	88.	255.00	280.01
7	88.	88.	255.00	277.44
8	88.	88.	255.00	276.59
9	170.	171.	255.00	273.68
10	1.	2.	255.00	273.65
11	170.	171.	255.00	273.81
12	117.	117.	255.00	275.09
13	117.	117.	255.00	278.06
14	92.	92.	255.00	285.54
15	92.	92.	255.00	293.31
16	170.	170.	260.00	261.55
17	58.	58.	272.80	272.78
18	117.	117.	255.00	261.14
19	117.	117.	255.00	255.01
20	170.	170.	255.00	258.02

PIPE	FLOWS	HGL
	LOAD 1	
1	885.	0.500762
2	792.	0.408245
3	700.	0.324439
4	612.	0.252788
5	524.	0.189436
6	435.	0.134571
7	154.	0.088679
8	259.	0.232520
9	59.	0.003281
10	160.	0.011575
11	481.	0.088143
12	-833.	-0.243200
13	-950.	-0.310322

14	-1042.	-0.368548
15	1135.	0.431357
16	18.	0.033218
17	75.	0.447122
18	34.	0.255432
19	75.	1.096785
20	20.	0.091966
21	76.	0.459567
22	0.	0.500762
23	0.	0.408245
24	0.	0.324439
25	0.	0.252788
26	0.	0.189436
27	0.	0.134571
28	193.	0.088679
29	0.	0.232520
30	0.	0.003281
31	0.	0.011575
32	0.	0.088143
33	0.	-0.243200
34	0.	-0.310322
35	0.	-0.368548
36	0.	0.431357
37	39.	0.033218
38	159.	0.447122
39	83.	0.255432
40	75.	1.096785
41	0.	0.091966
42	114.	0.459567

Appendix E

RUNNING THE PROGRAM ON THE AMDAHL 580

The following JCL is needed to run the program on the University of Manitoba's AMDAHL 580.

```
// JOB ',,,T=8M,L=10','MORGAN',CLASS=1
// EXEC FORTXCLG,PARM='SIZE=312K'
//FORT.SYSIN DD *
            .
            .
            .
        Program
            .
            .
            .
//GO.FT09F001 DD DSN=MORGAN.COST.DATA,
//  DISP=SHR
//GO.FT10F001 DD DSN=MORGAN.SYSTEM.DATA,
//  DISP=SHR
//GO.FT11F001 DD DSN=MORGAN.PIPES.DATA,
//  DISP=SHR
//GO.FT12F001 DD DSN=MORGAN.DEMAND.DATA,
//  DISP=SHR
//GO.FT14F001 DD DSN=MORGAN.BREAK.DATA,
//  DISP=SHR
```

The data sets assigned input/output units above contain all the input data. Data sets 'PIPES' and 'DEMAND' are updated each time the program is run. To delete a pipe which is considered uneconomical due to its weighting 'manually', use 'FETCH DA=PIPES.DATA'. When the data et has been fetched change the left most diameter number to 1 and change the left most length to the total length. The right hand diameter number and length should be 0.

The following pages are examples of how the data is entered into the data sets. The data set 'DEMAND.DATA' contains the data for single demand pattern design to save space. The data set 'BREAK.DATA' contains the data for multiple demand patterns design.

COST.DATA

1
24.00
130
1000
13

1	0.125	58.0
2	0.150	62.0
3	0.200	71.7
4	0.250	88.9
5	0.300	112.3
6	0.350	138.7
7	0.400	169.0
8	0.450	207.0
9	0.500	248.0
10	0.550	297.0
11	0.600	347.0
12	0.650	405.0
13	0.700	470.0

SYSTEM.DATA

37 20

1	1	2	760
2	1	4	520
3	1	6	890
4	2	3	1120
5	2	5	610
6	2	6	680
7	3	5	680
8	3	7	870
9	4	8	860
10	4	9	980
11	5	7	890
12	5	10	750
13	6	9	620
14	6	10	800
15	7	12	730
16	7	13	680
17	8	9	480
18	8	15	860
19	9	11	800
20	9	14	770
21	10	11	350
22	10	12	620
23	11	12	670
24	11	16	790
25	11	18	1150
26	12	13	750
27	12	17	550
28	13	17	700
29	14	15	500
30	14	16	450
31	14	19	750
32	15	19	720
33	16	18	540
34	16	19	700
35	17	18	850
36	18	20	750
37	19	20	970

PIPE.DATA

1	7	760	0	0	0
2	7	520	0	0	0
3	7	890	0	0	0
4	7	1120	0	0	0
5	7	610	0	0	0
6	7	680	0	0	0
7	7	680	0	0	0
8	7	870	0	0	0
9	7	860	0	0	0
10	7	980	0	0	0
11	7	890	0	0	0
12	7	750	0	0	0
13	7	620	0	0	0
14	7	800	0	0	0
15	7	730	0	0	0
16	7	680	0	0	0
17	7	480	0	0	0
18	7	860	0	0	0
19	7	800	0	0	0
20	7	770	0	0	0
21	7	350	0	0	0
22	7	620	0	0	0
23	7	670	0	0	0
24	7	790	0	0	0
25	7	1150	0	0	0
26	7	750	0	0	0
27	7	550	0	0	0
28	7	700	0	0	0
29	7	500	0	0	0
30	7	450	0	0	0
31	7	750	0	0	0
32	7	720	0	0	0
33	7	540	0	0	0
34	7	700	0	0	0
35	7	850	0	0	0
36	7	750	0	0	0
37	7	970	0	0	0

DEMAND.DATA

```
1
20
   1
      1        75.0      165.0      80.0
      2        74.0      220.0      90.0
      3        73.0      145.0      90.0
      4        72.0      165.0      70.0
      5       102.0     -800.0     102.0
      6        73.0      140.0      80.0
      7        67.0      175.0      90.0
      8        72.0      180.0      70.0
      9        70.0      140.0      75.0
     10        69.0      160.0      90.0
     11        71.0      170.0      93.0
     12        70.0      160.0      85.0
     13        64.0      190.0      80.0
     14        73.0      200.0      90.0
     15        73.0      150.0      80.0
     16        96.0     -800.0      96.0
     17        67.0      165.0      80.0
     18        70.0      140.0      90.0
     19        70.0      185.0      90.0
     20        67.0      160.0      70.0
```

BREAK.DATA

```
1
1
0
2
2
0
3
3
0
4
4
0
5
5
0
6
6
0
7
7
0
8
8
0
9
9
0
10
10
0
11
11
0
12
12
0
13
13
0
14
14
0
15
15
0
16
16
0
17
17
0
18
18
```

0
19
19
0
20
20
0
21
21
0
22
22
0
23
23
0
24
24
0
25
25
0
26
26
0
27
27
0
28
28
0
29
29
0
30
30
0
31
31
0
32
32
0
33
33
0
34
34
0
35
35
0
36
36

0
37
37
0
0

REFERENCES

1. Alperovits,E and U. Shamir. "Design of Optimal Water Distribution Systems", Water Resources Research . Vol. 13, No. 6, December, 1977. pp. 885-900.

2. Bhave, Pramod R.. "Noncomputer Optimization of Single Source Networks" Journal of the Environmental Engineering Division,ASCE. Vol. 104, No. EE4, August,1978. pp.799-812.

3. Bhave, Pramod R.. "Selecting Pipe Sizes in Network Optimization by LP", Journal of the Hydraulics Division,ASCE. Vol. 105, No. HY7, August 1979. pp 1019-1025.

4. Deb, Arun K. and A. K. Sakar. "Optimization in Design of Hydraulic Network", Journal of the Sanitary Engineering Divsion,ASCE. Vol. 97, No. SA2, April, 1971. pp. 141-159.

5. Deb, Arun K.. "Least Cost Design of Branched Pipe Network System", Journal of the Environmental Engineering Division,ASCE. Vol. 100, No. EE4, August 1974. pp. 821-835.

6. Deb, Arun K.. "Optimization of Water Distribution Network Systems", Journal of the Environmental Engineering Division,ASCE. Vol. 102, No. EE4, August 1976. pp. 837-851.

7. de Neufville, Richard, John Schaake,Jr. and Joseph H. Stafford . "Systems Analysis of Water Distribution Networks", Journal of the Sanitary Engineering Division,ASCE. Vol. 97, No. SA6, December, 1971. pp. 825-842.

8. Gessler, Johannes. "Optimization of Pipe Networks", Proceedings of the Ninth International Symposium on Urban Hydrology, Hydraulics and Sediment Control. Lexington, Kentucky. July 1982. pp. 165-171

9. Hillier, Frederick S., and Gerald J. Lieberman. Introduction to Operations Research. Holden-Day, Inc.,Oakland, 1980.

10. Holloway, M.B. and M.H. Chaudhry. "Comparsion of Pipe-Network-Analysis Computer Programs", Proceedings of the 6th Canadian Hydrotechnical Conference. Vol. 2, The Canadian Society for Civil Engineering, June 2, Ottawa, Ontario, 1983. pp. 543-558.

11. Jacoby, Shmuel L. S.. "Design of Optimal Hydraulic Networks" Journal of the Hydraulics Division,ASCE. Vol. 94, No. HY3, May, 1968. pp.641-661.

12. Kally, Elisha. "Computerized Planning of the Least Cost Water Distribution Network", Water and Sewage Works. 1972, pp. 121-127.

13. Karmeli, David , Y. Gaddish and S. Meyers. "Design of Optimal Water Distribtion Networks", Journal of the Pipeline Division, ASCE. Vol. 94, No. PL1, October, 1968. pp. 1-9.

14. Kettler, A.J. and I.C. Goulter. "Reliability Consideration in the Least Cost Design of Looped Water Distribution Systems", Proceedings of the Tenth International Symposium on Urban Hydrology, Hydraulics and Sediment Control. Lexington, Kentucky, July 25-28, 1983. pp. 305-312.

15. Lauria, Donald T.. "Research Needs for Capacity Planning", Journal of the American Water Works Association. January, 1983

16. Liang, Tung. "Design Conduit System By Dynamic Programming", Journal of the Hydraulics Division,ASCE. Vol. 97, No. HY3, March, 1971. pp. 383-393.

17. Linaweaver, F. P. and C. Scott Clark. "Costs of Water Transmission", Journal of the American Water Works Association. Vol. 56, No. 12, December, 1964. pp. 1549-1560.

18. Martin, Quentin W.. "Optimal Design of Water Conveyance Systems", Journal of the Hydraulics Division, ASCE. Vol. 106, No. HY9, September 1980. pp. 1415-1432.

19. Mays, Larry W., and Harry G. Wenzel,Jr. and Jon C. Liebman. "Model for Layout and Design of Sewer Systems", Journal of the Water Resources Planning and Management Division,ASCE. Vol. 102, No. WR2, November, 1976. pp. 385-405.

20. Morgan, D.R. and I. C. Goulter. "Least Cost Layout and Design of Looped Water Distribution Systems", Proceedings of the Ninth International Symposium on Urban Hydrology, Hydraulics and Sediment Control. Lexington, Kentucky, July 27-30, 1982. pp. 65-72.

21. Pomeroy, Richard D.. "Flow Velocities in Pipelines", Journal of the Hydraulics Division. Vol. 109, No. HY7, August, 1983. pp.1108-1117.

22. Quindry, Gerald E., E. Downey Brill,Jr. and Jon C. Liebman. Water Distribution System Design Criteria. Final Report. Department of Civil Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, 61801. March,1979.

23. Quindry G., E. D. Brill, J. Liebman and A. Robinson. Comments on "Design of Optimal Water Distribution System" by Alperovits and Shamir, Water Resources Research. Vol. 15, December, 1979.

24. Quindry, Gerald E., E. Downey Brill and Jon C. Liebman. "Optimization of Looped Water Distribution Systems", Journal of the Environmental Engineering Division,ASCE. Vol. 107, No. EE4, August, 1981. pp. 665-679.

25. Rasmusen, Hans Jorgen. "Simplified Optimization of Water Supply Systems", Journal of the Environmental Engineering Engineering Division,ASCE. Vol. 102, No. EE2, April, 1976, pp. 313-327.

26. Rowell, William F. and J. Wesley Barnes. "Obtaining Layout of Water Distribution Systems", Journal of the Hydraulics Division, ASCE. Vol. 108, No. HY1, January, 1982. pp. 137-148.

27. Schaake, John C. and Dennis Lai. Linear Programming and Dynamic Programming Application to Water Distribution Network Design. Hydrodynamics Laboratory, Department of Civil Engineering, Massachusetts Institute of Technology. Report No. 116, July 1969.

28. Shamir, Uri and Charles D. D. Howard . "Water Distribution System Analysis", Journal of the Hydraulics Division, ASCE. Vol. 94, No. HY1. January, 1968. pp. 219-234.

29. Shamir, Uri. "Optimal Design and Operation of Water Distribution Systems" Water Resources Research. Vol. 10, No. 1, February, 1974. pp. 27-36.

30. Templeman, Andrew B.. "Discussion of 'Optimization of Looped Water Distribution Systems' by Quindry et al." Journal of the Environmental Engineering Division, ASCE. Vol. 108, No. EE3, June, 1982. pp. 599-602.

31. Watanatada, Thawat. "Least Cost Design of Water Distribution Systems", Journal of the Hydraulics Division,ASCE. Vol. 99, No. HY9, September, 1973. pp. 1497-1513.

32. Yang, Kwang-Ping, Tung Liang and I-Pai Wu. "Design of Conduit System with Diverging Branches", Journal of the Hydraulics Division, ASCE. Vol. 101, No. HY1, January, 1975. pp. 167-188.