OPERATIONS RESEARCH CENTER

working paper

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

WORST-CASE ANALYSIS OF NETWORK DESIGN PROBLEM HEURISTICS

by

Richard T. Wong

OR 085-78 December 1978

This research was supported, in part, by the U. S. Department of Transportation under Contract DOT-TSC-1058, Transportation Advanced Research Program (TARP).

ABSTRACT

The Optimal Network problem (as defined by Scott [16]) consists of selecting a subset of arcs that minimizes the sum of the shortest paths between all nodes subject to a budget constraint. This paper considers the worst-case behavior of heuristics for this problem. Let n be the number of nodes in the network and ε be a constant between 0 and 1. For a general class of Optimal Network Problems, we show that the question of finding a solution which is always less than $n^{1-\varepsilon}$ times the optimal solution is NP-complete. This indicates that all polynomial-time heuristics for the problem most probably have poor worst-case performance. An upper bound for worst-case heuristic performance of 2n times the optimal solution is also derived. For a restricted version of the Optimal Network problem we describe a procedure whose maximum percentage of error is bounded by a constant.

"Worst-Case Analysis of Network Design Problem Heuristics"

by

Richard T. Wong

1. Introduction

This paper discusses the "optimal" network problem which can be described in the following way: select a subset of arcs in a network so that the total weighted sum of the shortest paths in the network is minimized subject to the constraint that the total cost of the arcs selected does not exceed a given budget. More formally, the optimal network problem can be formulated as the following mixed integer programming problem:

$$\text{Minimize} \quad \sum_{(i,j)\varepsilon A} \sum_{(k,\ell)\varepsilon(D\times D)} c_{ij} x_{ij}^{k\ell}$$

$$\text{subject to} \quad \sum_j x_{ij}^{k\ell} - \sum_q x_{qi}^{k\ell} = \begin{cases} r_{k\ell} & \text{if} \quad i = k \\ -r_{k\ell} & \text{if} \quad i = \ell \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij}^{k\ell} \leq r_{k\ell} y_{ij}$$

$$\sum_{(i,j)\varepsilon A} d_{ij} y_{ij} \leq B$$

$$x_{ij}^{k\ell} \geq 0 \qquad (i,j)\varepsilon A \text{ and } (k,\ell)\varepsilon(D\times D)$$

$$y_{ij} = 0 \text{ or } 1 \quad (i,j)\varepsilon A$$

where the decision variables are $x_{ij}^{k\ell}$, the amount of commodity (k,ℓ) routed on arc (i,j), and y_{ij}, a binary variable indicating whether or not arc (i,j) is to be constructed. Let D be the set of nodes and A be the set of possible arcs (undirected). Define $r_{k\ell}$ to be the amount of commodity (k,ℓ) that must be routed, d_{ij} to be the construction cost of arc (i,j) and c_{ij} to be the per unit routing cost of arc (i,j). Let B be the construction budget. All data d_{ij} and c_{ij} are assumed to be nonnegative. For technical purposes (and without any real loss of generality), we assume that all c_{ij} and d_{ij} are integer valued and that all problems under consideration have an optimal solution greater than zero.

This type of network design problem has potential uses in designing air, rail or highway transportation networks. Although such systems are usually much more complex than the above problem, this model could be useful in screening network configurations for more detailed study [4].

Previous work done on the optimal network problem has indicated that it is a very difficult optimization problem. Johnson, Lenstra, and Rinooy Kan have shown that the optimal network problem is NP-complete [8], which means that there is very probably no efficient method for solving problems of this type. Computational studies by several authors [1,3,4,7] using branch and bound techniques have shown that for optimal network problems with more than 50 or 75 arcs, solution times are prohibitive. So suboptimal heuristic methods appear to be the only methods available for generating solutions to large-scale network design models. Scott [16] and Dionne and Florian [4] have proposed heuristics for the optimal network problem. In the next section we review some of these procedures (for a more complete survey of the optimal network problem and related design models see Wong [17]).

An important question that arises in using heuristic techniques is the accuracy of the answers generated. One technique for evaluating heuristics is to analyze their worst-case performance. That is, we compute the maximum possible percentage deviation from the optimal solution when using the heuristic. This type of analysis is conservative in that only the worst possible error is computed, but can be useful in terms of evaluating performance guarantees for heuristics. Many researchers have analyzed heuristics for various combinatorial problems in terms of their worst-case error performance. See Garey and Johnson [6] for a survey of these results.

In this paper we analyze the worst-case behavior of a wide class of optimal network problem heuristics. The next section reviews some past work in designing such heuristics. Also some examples are given which demonstrate worst-case behavior for some of these procedures. The third section contains our main results which show that even finding an approximate optimal network solution is NP-complete. These results indicate that all polynomial-time heuristics for the optimal network problem probably have poor worst-case error bounds. The fourth section describes a particular heuristic algorithm whose worst-case error ratio for a restricted version of the optimal network problem is bounded by a constant that does not depend on the size of the input problem. The last section provides a summary and overview of the paper's results.

We should note that most of the previous work in this area (see [1,4,7,16]) dealt with a restricted version of the optimal network problem where all required flows $r_{k\ell}$ were one and every arc routing cost c_{ij} was equal to its construction cost d_{ij}. In this paper, unless otherwise noted, we assume that all required flows $r_{k\ell}$ are one <u>but</u> that an arc routing cost may be different from its construction cost.

2. Previous Work in Optimal Network Problem Heuristics

Scott [16] and Dionne and Florian [4] have presented some optimal network problem heuristics which we consider here.

The first heuristic that we review is due to Dionne and Florian and was stated as follows:

(H1) 1) Construct the minimal cost spanning tree (using the construction costs d_{ij} as the arc costs) as the initial network configuration.

2) As long as the budget constraint is not violated, add to the network configuration the arc whose construction cost is the least of all arcs not yet included in the network design.

Note that if the minimal cost spanning tree is infeasible because of the construction budget constraint then the problem is infeasible.

Dionne and Florian also presented another heuristic that is a modified version of one described by Scott. It has the following description:

(H2) 0) Let M be the set of arcs in the current network design. For $k \varepsilon M$, define $Q_k(M)$ as the increase in the total routing cost if arc k is deleted from M.

1) Initialize M so it contains all arcs in the network.

2) Find k* such that

$$L_{k*}(M) = \frac{Q_{k*}(M)}{d_{k*}} = \underset{k \varepsilon M}{\text{MIN}} \frac{Q_k(M)}{d_k} ,$$

where d_k is the construction cost of arc k. If $L_{k*}(M) = \infty$, then the removal of any link will disconnect the network and computation should be restarted using heuristic H1. Otherwise, delete arc (k*) from M and continue with step 3.

3) If $\sum_{k \varepsilon M} d_k > B$, i.e., the current network exceeds the construction budget, go to step 2; otherwise continue with step 4.

4) If $B - \sum_{k \varepsilon M} d_k \geq 0$, then introduce as many arcs as possible so that the routing cost decrease is maximized and the budget constraint is satisfied. END

The quantity $L_k(M)$ can be viewed as the normalized "loss" due to deleting arc k. At each iteration we delete the arc whose loss is the minimum of all arcs; the process continues until a feasible solution is reached. This procedure is related to the "greedy" heuristic that has been studied previously [2].

Dionne and Florian performed computational tests to compare both heuristics. H2 performed noticeably better than H1. In fact, for many test problems H2 was able to find the optimal solution.

Now we consider the worst-case performance for these heuristics. Let us define the following terms:

$V_h(\cdot)$ = the value of the solution computed by heuristic h for problem $(\cdot)$.

$V(\cdot)$ = the optimal solution value for problem $(\cdot)$.

$S(n)$ = the set of optimal network problems containing n nodes.

$$R_h(n) = \max_{s \in S(n)} \frac{V_h(s)}{V(s)} .$$

$R_h(n)$ is the worst possible error ratio when heuristic h is applied to optimal network problems consisting of n nodes. The goal of our worst-case performance analysis is to compute $R_h(n)$.

We show that for both of the above heuristics, the worst-case error ratio essentially behaves as a linear function of n, the number of nodes in the network. Therefore the error ratio is unbounded as the size of the network increases.

Consider the following canonical example depicted in Figure 1. Let t_1 and t_2 represent a subnetwork consisting of Z nodes. Figure 2 contains a diagram of this subnetwork. Any arc connected to t_1 or t_2 is considered to be connected to the center node in the corresponding subnetwork.

The label associated with each arc in Figure 1 denotes the arc's routing cost and the construction cost respectively. The construction budget B is 13.

Using heuristic H2, we start with all arcs in the network. Then we drop arc (t_1,t_2). Next, we drop arc (t_1,b) or (t_2,b) (the analysis is the same regardless of which arc is deleted). This leaves us with the following network depicted in Figure 3. Recalling that all required flows r_{ij} are equal to one, we compute the cost of the above solution as

$$V_{H2} = 8Z^4 + 16Z^3 + 4Z^2 + 4Z + 2.$$

Figure 4 depicts the optimal solution to the above problem. The optimal solution has

$$V = 8Z^3 + 8Z^2 + 12Z + 6.$$

The total number of nodes in the network is 2Z + 2.

$$R_{H2}(2Z+2) \geq \frac{8Z^4 + 16Z^3 + 4Z^2 + 4Z + 2}{8Z^3 + 8Z^2 + 12Z + 6}$$

$$R_{H2}(2Z+2) \geq Z \quad \text{for} \quad Z \geq 1.$$

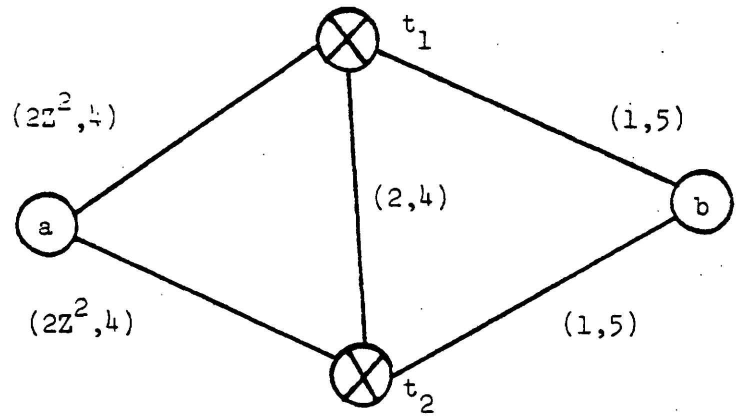

Figure 1: Optimal Network Problem Example for Heuristic H2.

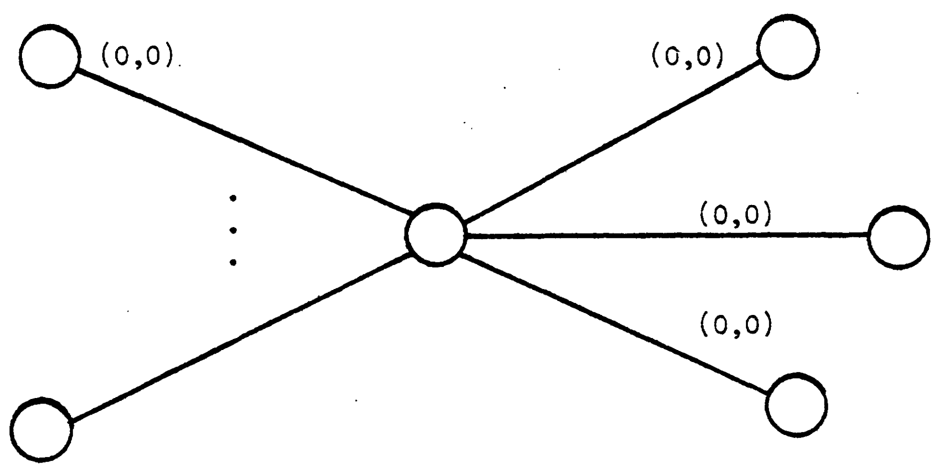

Figure 2: Star Network Representing a Node.

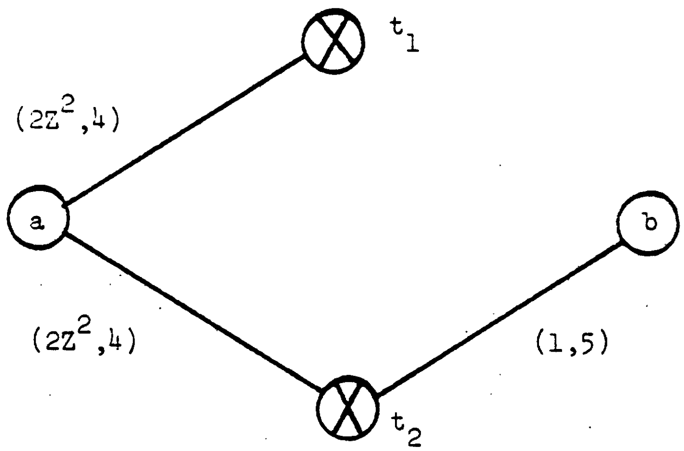

Figure 3: Solution Computed by Heuristic H2 for the Example.

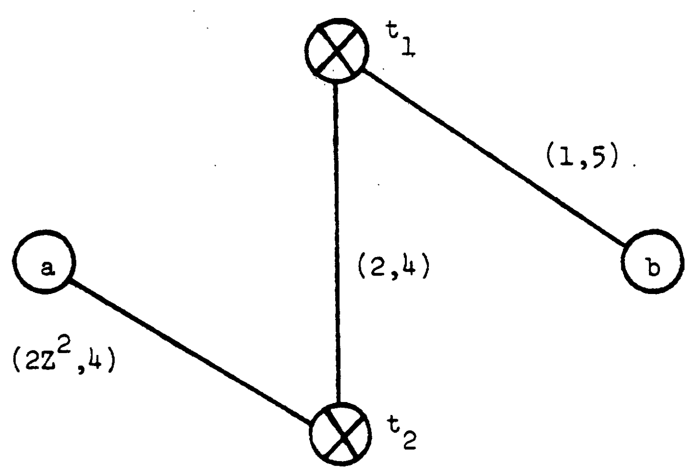

Figure 4: Optimal Solution for Optimal Network Example.

This implies

$$R_{H2}(n) \geq \frac{1}{2} n - 1 \quad \text{for} \quad n = 6,8,10,\ldots$$

So our example shows that the worst-case error ratio for H2 must be at least linear since our canonical example exhibits such behavior for an infinite number of network sizes.

Heuristic H1 behaves similarly. Consider the canonical example represented by Figure 5. Let the budget B be 25. An analysis that closely follows the one given above tells us that

$$R_{H1}(n) \geq \frac{1}{2} n - 1 \quad \text{for} \quad n = 6,8,10,\ldots$$

So the worst-case error ratio for H1 must also be at least linear.

The above results lead us to question if there are optimal network heuristics whose worst-case behavior is better than the ones given above. The next section gives a result which indicates that all "reasonable" heuristics must probably perform nearly as badly in terms of worst-case error margins. Also we show that the worst-error ratio for the above heuristics is no worse than a linear function of network size. So the examples given above show essentially the worst possible behavior of heuristics H1 and H2.

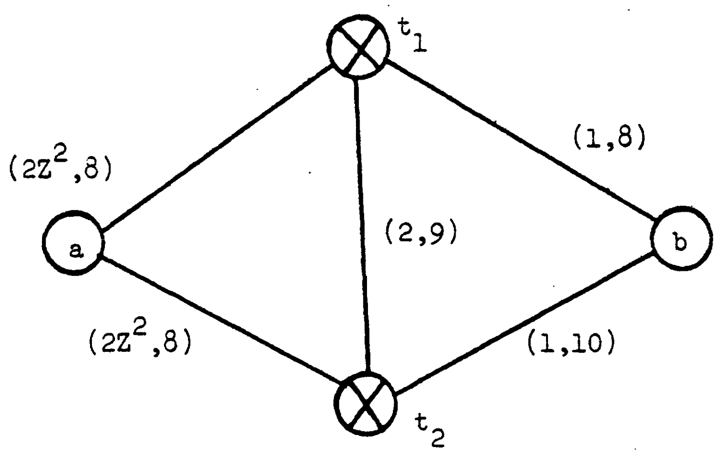

Figure 5: Optimal Network Problem Example for Heuristic H1.

3. Two Theorems on the Accuracy of Optimal Network Problem Heuristics

The first result that we consider concerns the class of polynomial-time heuristics for optimal network problems, that is, the set of all optimal network design heuristics whose worst-case computation time is a polynomial function of the problem input size. As we stated previously, Johnson, Lenstra and Rinooy Kan [8] showed that the optimal network problem is NP-complete. Next we show that the problem of finding an optimal network design heuristic whose worst-case error ratio is less than $n^{1-\varepsilon}$, where n is the number of nodes in the network and ε is between 0 and 1, is also NP-complete. So finding a polynomial-time optimal network design heuristic that is always "close" to the optimal solution is as hard as finding a polynomial-time procedure that is always optimal. Sahni and Gonzales [15] demonstrated similar results for the traveling salesman problem (without the triangle inequality restriction), the multicommodity network flow problem and other combinatorial problems. Garey and Johnson [5] derived a related result for the graph-coloring problem.

Our first result can be stated in the following terms:

Definition: The approximate optimal network problem is the following: let ε be any fixed positive constant between 0 and 1, for any optimal network problem s find a solution whose value is less than or equal to $n^{1-\varepsilon}V(s)$, where n is the number of nodes in the problem s.

Theorem 1: The approximate optimal network problem is NP-complete.

Proof: Since the optimal network problem belongs to NP (see [8,9,10]), the approximate optimal network problem must also belong to NP. Now we show that if the approximate problem could be solved in polynomial-time, that is, if there existed a polynomial-time heuristic h* and a constant ε, $0 < \varepsilon < 1$, such that $R_{h*}(n) < n^{1-\varepsilon}$ for all n, then all of the NP-complete problems could be solved in polynomial-time.

Let us define a useful auxillary problem. The Steiner tree problem [9] has the following description: given a network (D,A) with node set D and arc set A and the data i) $\{d_{ij}\}_{(i,j)\varepsilon A}$, the set of arc construction costs, ii) B, the construction budget, and iii) S, a set of nodes which is a subset of D, determine if there is a subtree of the network whose construction cost is less than the given budget B with the property that all nodes in S are connected by the subtree. Karp [9] has shown that the Steiner tree problem is NP-complete.

We next demonstrate that if the heuristic h* defined above exists, then the Steiner tree problem could be solved in polynomial-time. It would then follow [9,10] that every NP-complete problem could be solved in polynomial-time.

Given any Steiner tree problem, transform it into an approximate optimal network problem in the following way: replace each node in the set S by a subnetwork of the type pictured in Figure 2. Each of these subnetworks should have M^k nodes, where M is the number of nodes in the original Steiner tree problem and k is an integer constant that will be specified later. All routing and construction costs for arcs in the subnetwork should be zero.

Attach a special node T to the Steiner problem network. Every "special" arc between T and the set of nodes D has a construction cost of zero and routing cost of one. Every arc between T and a node in S, which is represented by a star network corresponding to Figure 2, is connnected to the center of the star network. All arcs originally in the Steiner problem network have zero routing cost and retain their original construction costs.

Figures 6 and 7 illustrate such a transformation. S' is the set (D-S). The arc labels in the original Steiner tree problem network are the arc construction costs. The arc labels in the modified optimal network problem indicate the arc routing and construction costs.

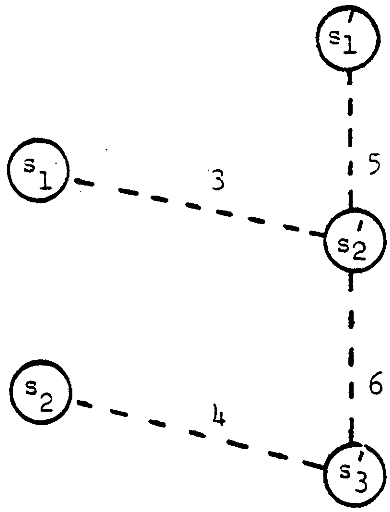

Figure 6: Example of a Steiner Network Problem (Before the Transformation).

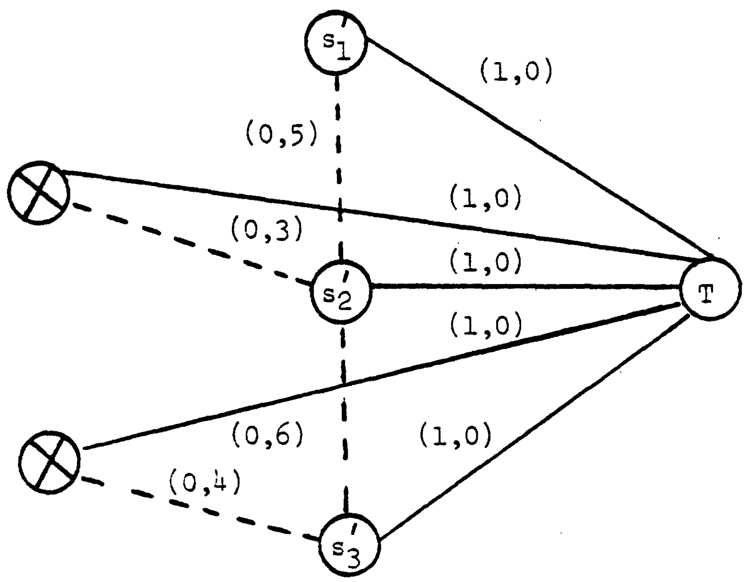

Figure 7: Example of a Steiner Network Problem (after the Transformation).

The construction budget for the optimal network problem is the same as the Steiner problem budget. As we have assumed throughout this paper, all required flows in the optimal network problem are equal to one.

It is important to note that this transformation to create an optimal network problem from a Steiner tree problem is a polynomially-time bounded procedure for any finite value of the parameter k. Also note that the size n of the optimal network problem created by our transformation is at most $(M^{k+1}+1)$ nodes.

Now if one of the special arcs is utilized in the optimal network design to connect two nodes that are in S,

$$\text{routing cost} \geq 4M^{2k}.$$

If all nodes in S are connected with arcs from the original Steiner tree problem,

$$\text{routing cost}^{\dagger} \leq 2M^{k+2}, \quad k \geq 3 \text{ and } M \geq 4.$$

---

† Let $RC(N_1,N_2)$ represent the cost of routing between every pair of nodes in the set $(N_1 \times N_2)$. Then we can say total routing cost = $RC(S,S) + 2RC(S,S') + RC(S',S') + 2RC(S,\{T\}) + 2RC(S',\{T\})$, where the factors of 2 are a result of the symmetry of the required flows in the network. Since all arcs from the original Steiner tree problem have routing cost zero, $RC(S,S) = 0$. We can always utilize the special arcs connecting T to the rest of the network so we have $RC(S,S') \leq \frac{M^{k+2}}{2}$, $RC(S',S') \leq 4M^2$, $RC(S,\{T\}) \leq M^{k+1}$ and $RC(S',\{T\}) \leq M$. Therefore, total routing cost $\leq M^{k+2} + 4M^2 + 2M^{k+1} + 2M \leq 2M^{k+2}$, $k \geq 3$ and $M \geq 4$.

Now suppose there is a polynomial-time heuristic h* for the optimal network problem such that for some $0 < \varepsilon < 1$

$$R_{h*}(n) < n^{1-\varepsilon} \quad \text{for all } n \geq 1.$$

Since there exists a $k \geq 3$ such that $\frac{k-2}{k+2} \geq 1 - \varepsilon$ we have

$$R_{h*}(n) < n^{1-\varepsilon} < n^{\frac{k-2}{k+2}} \quad \text{for some } k \geq 3.$$

Next we examine the implications of the above statement on the class of optimal network problems consisting of our transformed Steiner problems. Note that $n \leq M^{k+1} + 1$, where M is the number of nodes in the original Steiner problem. Therefore, <u>for this class of optimal network problems</u>

$$R_{h*}(n) < n^{\frac{k-2}{k+2}} \leq (M^{k+1}+1)^{\frac{k-2}{k+2}} ;$$

for $M \geq 4$,

$$(M^{k+1}+1)^{\frac{k-2}{k+2}} < (M^{k+2})^{\frac{k-2}{k+2}} = M^{k-2} ;$$

and,

$$R_{h*}(n) < M^{k-2} \quad M \geq 4,\ k \geq 3.$$

The above inequality implies that for $M \geq 4$ the Steiner tree problem could be solved in polynomial-time by first using our polynomial-time transformation to create an optimal network problem and then applying the heuristic h* to it. The existence of a subtree satisfying the conditions of the Steiner problem could be verified by examining whether the heuristic gave a routing cost solution that was less than $4M^{2k}$.

Since the finite number of cases where $M < 4$ will not effect the polynomial-time bound of this procedure, the above inequality implies that the Steiner tree problem could be solved in polynomial-time.

Finding a heuristic h* as defined above is equivalent to solving an NP-complete problem, so we can say that the approximate optimal network problem is also NP-complete. □

We have seen that all polynomial-time bounded heuristics most probably have a worst-case error ratio that grows almost linearly with the size of the network, or at a faster rate. Next we see that for reasonable heuristics the error ratio grows no faster than linearly with the size of the network.

Before presenting this result we introduce some additional notation. Let T be any spanning tree of a network and arbitrarily choose a node R with degree one from T and designate it as the root node. A node f is the father of node N if f lies on the (unique) path in T between N and R and if there is an arc in T that connects f and N. Node s is the son of node f if f is its father. Let w_i be the number of nodes which are descendants of node i (i.e., nodes other than i whose path to R in T must pass through i). Des(N) is the set of nodes which are descendants of N.

Figure 8 contains an example illustrating these definitions. Node 1 is the root node. In this example node 2 is the father of node 5. Also $w_2 = 5$ and $w_6 = 0$.

Theorem 2: For optimal network problems whose routing costs satisfy the triangle inequality, any heuristic h which always produces a feasible solution will have a worst-case error ratio

$$R_h(n) \leq 2n \qquad \text{for all } n,$$

where n is the number of nodes in the input network.

Proof: We will show that

$$\frac{\text{routing cost of any spanning tree network}}{\text{routing cost of the complete network}} \leq 2n.$$

(The complete network contains every arc in A, the set of all possible arcs. Note that A may not have an arc for every pair of nodes in the network.) The theorem immediately follows from this fact since the above ratio is greater than or equal to $R_h(n)$ for any heuristic h which always produces a feasible solution.

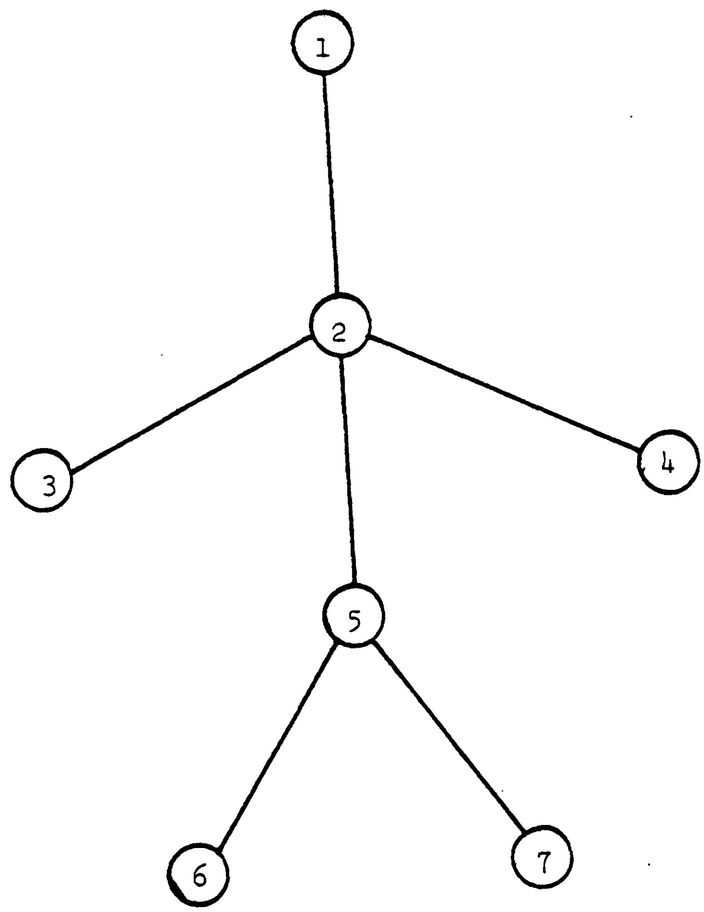

Figure 8: Example of a Tree with Root Node 1.

Let T be any spanning tree for an optimal network problem and C denote the complete network. Let RC(T) represent the routing cost of network design T and n be the total number of nodes in s. Consider an arc (i,j) belonging to T (since we are dealing with undirected arcs, assume that for any arc (i,j) in T, i is the father of j). Its contribution to the total routing cost is $S(i,j) = 2(w_j+1)(n - (w_j+1))c_{ij}$ (that is, the number of origin-destination pairs whose travel path passes through arc (i,j) multiplied by the routing cost of arc (i,j)).

Therefore,

$$RC(T) = \sum_{(i,j)\varepsilon T} S(i,j).$$

For the routing cost of the complete network, let a_{ij} be the minimum routing cost between nodes i and j on the complete network. Since all required flows are one we have

$$RC(C) = \sum_{(i,j)\varepsilon(D\times D)} a_{ij}.$$

Let us define the following quantity

$$C(i,j) = \sum_{k\varepsilon Des(j)\cup\{j\}} 2(a_{jk}+a_{ki}) \geq 2(w_j+1)c_{ij} \qquad (i,j)\varepsilon T$$

where the inequality follows from the triangle inequality for the routing costs and symmetry of the routing costs (since the arcs are undirected).

Therefore,

$$\frac{S(i,j)}{C(i,j)} \leq \frac{2(w_j+1)(n - (w_j+1))c_{ij}}{2(w_j+1)c_{ij}} \leq n \qquad (i,j)\varepsilon T.$$

Combining these inequalities for all $(i,j)\varepsilon T$ we have

$$\frac{\sum_{(i,j)\varepsilon T} S(i,j)}{\sum_{(i,j)\varepsilon T} C(i,j)} \leq n$$

and since $\sum_{(i,j)\varepsilon T} S(i,j) = RC(T)$

$$\frac{RC(T)}{\sum_{(i,j)\varepsilon T} C(i,j)} \leq n.$$

Next we show that $\sum_{(i,j)\varepsilon T} C(i,j) \leq 2RC(C)$ and thus complete the proof.

We argue that each arc cost term a_{st} appears in at most two expressions of the form $C(i,j)$ (without loss of generality assume that node t is a descendent of node s). a_{st} appears in the expression $C(i,j)$ only if

1) j equals s. Recall that since i must be the father of j, i must be the father of s.

2) i equal s and t belongs to $Des(j) \cup \{j\}$.

The first situation can only happen once since node s must have a unique father. The second situation can only occur once since if it happened twice, for example, with $C(i,j_1)$ and $C(i,j_2)$, $j_1 \neq j_2$, then between s and t there would be two distinct paths in the tree T.

Since $RC(C) = \sum_{(s,t)\varepsilon D\times D} a_{st}$ and the term a_{st} occurs in at most two terms of the form $C(i,j)$ we have

$$\sum_{(i,j)\varepsilon T} C(i,j) \leq 2RC(C)$$

Therefore,

$$\frac{RC(T)}{2RC(C)} \leq n$$

or

$$\frac{RC(T)}{RC(C)} \leq 2n. \qquad \square$$

Notice that the optimal network problem used in the proof of Theorem 1 had routing costs which satisfy the triangle inequality. Therefore Theorem 1 also holds if we impose the triangle inequality for the routing costs of the optimal network problem.

With these two theorems we have demonstrated probable lower and upper bounds on the worst-case error ratio for all reasonable polynomial-time heuristics for the optimal network problem with the triangle inequality for all routing costs.

The above results can also be extended to situations in which the required flows $r_{k\ell}$ are not necessarily equal to one. Suppose that all the $r_{k\ell}$ are positive integers such that

$$\max_{i,j,k,\ell} \frac{r_{k\ell}}{r_{ij}} \leq n^P \quad \text{for some } P \geq 3. \text{ Then}$$

Theorem 1 is modified by changing the worst-case error ratio from $n^{1-\varepsilon}$ to n^{P-2}. Theorem 2 is modified by changing the upper bound of 2n to $2n^{P+1}$. The proofs of such generalizations are straightforward modifications of the ones given above and will not be given here.

4. A Heuristic for a Special Case of the Optimal Network Problem

In this section we consider a special case of the optimal network problem where all construction costs d_{ij} are one. The budget constraint for this type of problem essentially limits the number of arcs allowed in the optimal network design. We will not have to assume that the triangle inequality holds for the routing costs. Johnson, Lenstra, and Rinooy Kan [8] have also shown that this restricted problem is NP-complete.

With these new restrictions on the problem, the result of Theorem 1 is no longer valid. We will describe a polynomial-time heuristic h whose worst-case error ratio

$$R_h(n) \leq 2 \quad \text{for all } n.$$

Let TREE(i) be the tree network of minimum routing cost paths between node i and every other node in the network. COST(i) is the sum of the minimum routing costs from node i to every other node in the network.

Our third heuristic can be defined as:

(H3) 1) Find i such that

$$COST(i) = \min_{j \in D} COST(j).$$

2) TREE(i) is the proposed network configuration.

Theorem 3: For optimal network problems having all construction costs equal to one

$$R_{H3}(n) \leq 2 \quad \text{for all } n.$$

Proof: We demonstrate this result by proving the stronger fact that

$$\frac{V_{H3}(s)}{RC(C)} \leq 2, \quad \text{for all } s,$$

where $V_{H3}(s)$ is the value of the solution computed by heuristic H3 for optimal network problem s and RC(C) is the routing cost (and solution cost) of the complete network (i.e., the network with all arcs in A).

As before define a_{ij} to be the minimum routing cost between nodes i and j on the complete network. Therefore,

$$\mathrm{COST}(i) = \sum_{j \varepsilon D} a_{ij}.$$

The routing cost for connecting node $j \neq i$ to all other nodes in the problem using the network TREE(i) is at most $(n-2)a_{ij}$ + COST(i). So

$$V_{H3}(s) \leq n \cdot \mathrm{COST}(i) + (n-2) \sum_{j \neq i} a_{ij},$$

and

$$V_{H3}(s) \leq (2n-2)\ \mathrm{COST}(i).$$

For the complete network we have

$$\mathrm{RC}(C) = \sum_{j \varepsilon D} \mathrm{COST}(j).$$

Since $\mathrm{COST}(i) \leq \mathrm{COST}(j)$ for all j,

$$\mathrm{RC}(C) \geq n \cdot \mathrm{COST}(i).$$

This implies

$$\frac{V_{H3}(s)}{\mathrm{RC}(C)} \leq \frac{(2n-2)\ \mathrm{COST}(i)}{n \cdot \mathrm{COST}(i)} \leq 2. \qquad \Box$$

Note that heuristic H3 has a polynomially bounded computation time so that it is possible to have a polynomial-time approximation procedure for a restricted class of optimal network problems whose worst-case error ratio is bounded by a constant. Theorem 1 shows that it is unlikely that such a heuristic exists for a broader class of network design problems.

We believe that combining some local improvement heuristic (perhaps one which added arcs in a "greedy" manner) with H3 could lead to a useful optimal network problem heuristic. It would be necessary to perform additional worst-case analyses or some computational tests in order to verify this conjecture.

5. Conclusions

The results of this paper indicate some unusual aspects concerning the complexity of the optimal network problem. Theorem 1 shows that even getting "close" to the optimal solution is an NP-complete problem. So, in a sense, this network design problem is more difficult than many other NP-complete problems. Similar results of this nature have been developed by Sahni and Gonzalez [15] and Garey and Johnson [6].

Theorem 1 also applies to other discrete network design problems such as the one treated by Leblanc [13] and Morlok and Leblanc [14]. This problem is similar to the optimal network problem except that more complex routing costs and strategies are allowed. So a variety of network design problems appear to be inherently very difficult.

For optimal network problems where the routing costs (c_{ij}'s) satisfy the triangle inequality, we have an even stronger result. A strengthened version of theorem 1 along with theorem 2, implies that the upper and lower bounds on the worst-case behavior of all reasonable optimal network heuristics (i.e., polynomial-time heuristics that always produce a feasible solution) must be very close unless P = NP.

In addition, we explored the relation between various problem parameters and heuristic accuracy. By allowing the required flows (r_{ij}'s) to assume different values we were again able to obtain probable (unless P=NP) upper and lower bounds on the worst-case behavior of reasonable heuristics. We also saw that by restricting all the construction costs (d_{ij}'s) to be equal, it is then possible to find heuristics whose worst-case error is bounded by a constant independent of problem size.

Although most optimal network heuristics probably have a bad worst-case error, there may be some heuristics whose "average" case behavior is quite good. In Section 2 we saw that heuristics used by Dionne and Florian [3] can be very inaccurate in terms of worst-case error even though computational tests have indicated that their relative margins of error are usually quite small. Many heuristics, especially ones for complicated real world problems (such as telephone network optimization), also appear to behave in a similar way. An interesting area of future work would be to explore probabilistic analyses of optimal network heuristics. See Karp [11,12] for some examples of probabilistic analyses for various combinatorial problems.

Acknowledgments

I am indebted to Professor Thomas L. Magnanti of MIT for his encouragement and suggestions concerning this paper. Paulo Villela of MIT also provided useful comments. This work was supported in part by the Department of Transportation Advanced Research Program (TARP) contract No. DOT-TSC-1058.

REFERENCES

[1] Boyce, D. E., Farhi, A., and Weischedel, R., "Optimal Network Problem: A Branch-and-Bound Algorithm," Environment and Planning, Vol. 5, pp. 519-533, 1973.

[2] Cornuejols, G., Fisher, M. L., and Nemhauser, G. L., "An Analysis of Heuristics and Relaxations for the Uncapacitated Location Problem," Man. Sci., Vol. 23, pp. 789-810, 1977.

[3] Dionne, R., "Une Analyse Théorique et numérique du Problème du Choix Optimal d'un Réseau de Transport sans congestion," Publication No. 198, Départment d'informatique, Université de Montreal, October 1974.

[4] Dionne R., and Florian, M., "Exact and Approximate Algorithms for Optimal Network Design," Publication No. 41, Centre de Recherche Sur les Transports, Université de Montreal, 1977.

[5] Garey, M. R., and Johnson, D. S., "The Complexity of Near-Optimal Graph Coloring," J. of Assoc. for Com. Mach., Vol. 23, pp. 43-49, 1976.

[6] Garey, M. R. and Johnson, D. S., "Approximation Algorithms for Combinatorial Problems: An Annotated Bibliography," in Algorithms and Complexity (J. F. Traub, ed.), Academic Press, New York, 1976.

[7] Hoang, H. H., "A Computational Approach to the Selection of an Optimal Network," Man. Sci., Vol. 19, pp. 488-498, 1973.

[8] Johnson, D. S., Lenstra, J. K., Rinnooy Kan, A. H. G., "The Complexity of the Network Design Problem," (to appear in Networks).

[9] Karp, R. M., "Reducibility Among Combinatorial Problems," in Complexity of Computer Computations, (R. E. Miller and J. W. Thatcher, eds.), Plenum Press, New York, pp. 85-104, 1972.

[10] Karp, R. M., "On the Computational Complexity of Combinatorial Problems," Networks, Vol. 5, pp. 45-68, 1975.

[11] Karp, R. M., "The Probabilistic Analysis of Some Combinatorial Search Algorithms," in Algorithms and Complexity, (J. F. Traub, ed.), Academic Press, New York, 1976.

[12] Karp, R. M., "Probabilistic Analyses of Partitioning Algorithms for the Traveling Salesman Problem in the Plane," Math. of OR, Vol. 2, pp. 209-224, 1977.

[13] LeBlanc, L. J., "An Algorithm for the Discrete Network Design Problem," Trans. Sci., Vol. 9, No. 3, pp. 283-287, August 1975.

[14] Morlok, E. K. and LeBlanc, L. J., "A Marginal Analysis Technique for Determining Improvements to an Urban Road Network," Paper presented at ORSA/TIMS National Meeting, Las Vegas, NV, Fall 1975.

[15] Sahni, S., and Gonzalez, T., "P-Complete Approximation Problems," J. of Assoc. for Com. Mach., Vol. 23, pp. 555-565, 1976.

[16] Scott, A. J., "The Optimal Network Problem: Some Computational Procedures," Trans. Res., Vol. 3, pp. 201-210, 1969.

[17] Wong, R., "A Survey of Network Design Problems," Working Paper ORC 053-76, M.I.T. Operations Research Center, 1976.