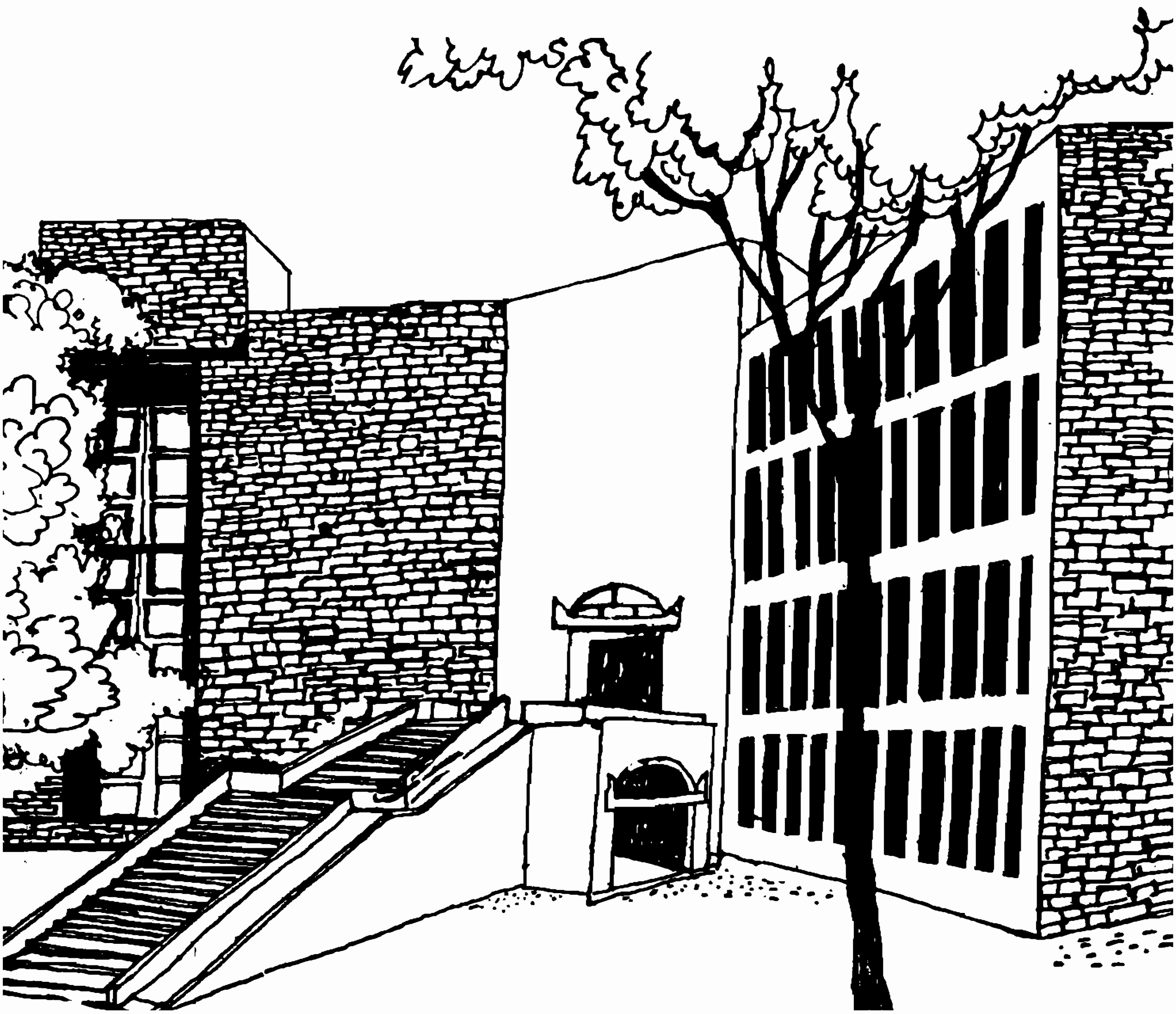
विद्याविनियोगाद्विकासः
IIM
AHMEDABAD

Working Paper



AN EXACT ALGORITHM FOR THE UNCAPACITATED NETWORK DESIGN PROBLEM

By

Trilochan S_stry

W.P. No.1412
November 1997

The main objective of the working paper series of the IIMA is to help faculty members to test out their research findings at the pre-publication stage.

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD - 380 015
INDIA

Abstract

We describe an $O(n^2 2^K + n 3^K)$ algorithm for the uncapacitated network design problem where K is the number of commodities, and n the number of nodes in the graph.

Key words and phrases: network design, algorithm

1 Introduction

In this paper, we study the uncapacitated multi commodity network design problem, which can be described as follows. Consider an undirected graph $G = (N, A)$, with node set N, arc set A, and origin destination pairs s_k, t_k, with demand of 1 unit between every pair for $k = 1, \ldots, K$. Capacity can be purchased on each arc $(i, j) \in A$ at cost $w_{ij} \geq 0$. Flow costs are assumed to be zero. The objective is to minimise the total cost while satisfying demand between every origin destination pair. The Steiner Tree problem, which is known to be NP-complete, is a special case of this problem in which all commodities have a common origin. Balakrishnan, Magnanti and Wong (1989) have studied the uncapacitated network design problem and solved large instances using a dual ascent based procedure. They also provide detailed references to the literature on this problem. Several agloriths for the Steiner Tree problem are known. Hakimi (1971) described a spanning tree enumeration algorithm. Balakrishnan and Patel (1987) describe an enumeration algorithm based on an implicit degree constrained formulation of the problem. Dreyfus and Wagner (1971) and Levin (1971) describe a dynamic programming based algorithm. Hwang, Richards and Winter (1992) provide d·tailed references to the literature on Steiner trees. The uncapacitated multi nmodity network design (UMC) problem can be formulated as follows.

oblem *UMC*

$$\text{Min} \sum_{(i,j)\in A} w_{ij} y_{ij}$$

subject to:

$$\sum_j (x_{ji}^k - x_{ij}^k) = \begin{cases} -1 & \text{if } i = s_k \\ 1 & \text{if } i = t_k \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ij} \geq x_{ij}^k + x_{ji}^k$$

$$x, y \geq 0; y \in \{0, 1\}.$$

Let $m = |A|$ and $n = |N|$ denote the number of arcs and nodes repectively, and let $\mathcal{K} = \{1, \ldots, K\}$ be the set of commodities. The arcs are undirected and have symmetric cost, i.e., $w_{ij} = w_{ji}$. The flow variables x_{ij}^k are directed and have zero flow cost. In the next section we characterise the optimal solutions and describe a $O(n^2 2^K + n3^K)$ algorithm to solve the problem.

2 An Exact Algorithm

It is possible to convert the undirected arc set to a set of directed arcs by using the following transformation. We replace each undirected arc (i, j) with five directed arcs, (i, i'), (j, i'), (i', j'), (j', i) and (j', j). Arc (i', j') has cost w_{ij} and the other four arcs have zero costs. For ease of exposition, we consider the directed version of the problem. The following result enables us to confine our search for optimal solutions to $0 - 1$ flows.

Lemma 1 *There is an optimal solution in which each commodity k flows on exactly one path between s_k and t_k.*

Proof
If a commodity flows on two paths, we can redirect all flow from one path to the other without increasing cost since capacity is unrestricted.

□

Hereafter, we only consider optimal solutions with each commodity k having exactly one path from s_k to t_k and $0-1$ flows for each commodity on any arc. Let $Q \subset \mathcal{K}$ denote a subset of commodities. Let $a(i, j)$ denote the shortest distance from node i to node j using w_{ij} as arc costs. In the algorithm, we implicitly allow the fixed charge variables y_{ij} to take any nonnegative integer value. In any feasible solution, we say that an arc (i, j) *shares at most q* commodities if $y_{ij} \geq 2$ whenever $x_{ij}^k = 1$ for $q + 1$ or more commodities.

The algorithm finds $\pi_i(Q)$, the minimum cost of sending one unit of flow from nodes $s_k : k \in Q$ to node i, using a generalisation of Dijkstra's shortest path algorithm for graphs with nonnegative arc costs. Similarly, it finds $\tau_i(Q)$, the minimum cost of sending one unit of flow from node i to nodes $t_k : k \in Q$. Thus, $\pi_i(Q) + \tau_i(Q)$ is the minimum cost of sending one unit from s_k to t_k for all $k \in Q$ if all commodities flow through node i. By varying Q over all subsets of $\mathcal{K}$, the algorithm finds the optimal solution.

Algorithm Multi Path

Let $A(i)$ denote the set of arcs adjacent to node i

Initialise

$pred(i,k) \longleftarrow s_k$ and $succ(i,k) \longleftarrow t_k$ for all $i \in N$, $k \in \mathcal{K}$

$\pi_i(\Phi) = \tau_i(\Phi) = 0$, $\pi_i(\{s_k\}) = a(s_k . i)$, $\tau_i(\{t_k\}) = a(i, t_k)$ for all $i \in N$

$OPT \longleftarrow \sum_{k \in \mathcal{K}} a(s_k, t_k)$

for $q = 2$ to K **do**

begin

for all subsets $Q \subset \mathcal{K}$ such that $|Q| = q$ **do**

begin

$\pi_i(Q) = \sum_{k \in Q} a(s_k, i)$, $\tau_i(Q) = \sum_{k \in Q} a(i, t_k)$

$\pi_i(Q) = \min\ \{\pi_i(Q_1) + \pi_i(Q_2) : Q_1 \cap Q_2 = \Phi,\ Q_1 \cup Q_2 = Q\}$

$\tau_i(Q) = \min\ \{\tau_i(Q_1) + \tau_i(Q_2) : Q_1 \cap Q_2 = \Phi,\ Q_1 \cup Q_2 = Q\}$

$S \longleftarrow \Phi$, $T \longleftarrow \Phi$

while $|S| < n$ or $|T| < n$ **do**

begin

choose $i \in \overline{S}$ such that $\pi_i(Q) = \min\ \{\pi_j(Q) : j \in \overline{S}\}$

$S \longleftarrow S \cup \{i\}$, $\overline{S} \longleftarrow \overline{S} - \{i\}$

for $j \in A(i)$ **if** $\pi_j(Q) > \pi_i(Q) + w_{ij}$ **then**

  begin

  $\pi_j(Q) = \pi_i(Q) + w_{ij}$

  $pred(j,k) \longleftarrow i$ for all $k \in Q$

  end

choose $i \in \overline{T}$ such that $\tau_i(Q) = \min\ \{\tau_j(Q) : j \in \overline{S}\}$

$T \longleftarrow T \cup \{i\}$, $\overline{T} \longleftarrow \overline{T} - \{i\}$

for $j \in A(i)$ **if** $\tau_j(Q) > \tau_i(Q) + w_{ij}$ **then**

  begin

  $\tau_j(Q) = \tau_i(Q) + w_{ij}$

  $succ(j,k) \longleftarrow i$ for all $k \in Q$

  end

end{*while*}

$OPT(Q) = \min\ \{\pi_i(Q) + \tau_i(Q) : i \in N\}$

$OPT(Q) = \min\ \{OPT(Q_1) + OPT(Q_2) : Q_1 \cap Q_2 = \Phi,\ Q_1 \cup Q_2 = Q\}$

end

end

Let the iterations in which q varies from 2 to K be the *outer iterations*. Let the iterations in which we choose all subsets Q of cardinality $|Q| = q$ be the *intermediate iterations,* and those in which we vary the cardinality of the sets S and T, the *inner iterations.*

Theorem 1 *Algorithm multi path solves the uncapacitated multi commodity network design problem UMC.*

Proof

To establish the result, we use the following inductive hypotheses on the cardinality of the sets Q, S and T. For a given value of q, and set $Q : |Q| = q$, and some set S or T in an inner iteration,

(i) $\pi_i(Q)$ is the mimimum cost of reaching node i from nodes $s_k : k \in Q$, and $\tau_i(Q)$ is the mimimum cost of reaching nodes $t_k : k \in Q$ from node i.

(ii) $\pi_j(Q)$ ($\tau_j(Q)$) for any node $j \notin S$ ($j \notin T$) is the mimimum cost of reaching node j (nodes $t_k : k \in Q$) from nodes $s_k : k \in Q$, (from node j) either using only nodes in S (nodes in T) as intermediate nodes, or the shortest paths from $s_k : k \in Q$ (from j) to node j (to $t_k : k \in Q$).

(iii) At the start of any intermediate iteration, and before the start of the inner iterations, $\pi_i(Q)$ ($\tau_i(Q)$) equals the minimum cost of reaching node i (nodes $t_k : k \in Q$) from origins $s_k : k \in Q$ (from node i) if at most $q - 1$ commodities share an arc.

(iv) At the end of any outer iteration with $|Q| = q$, $OPT(Q)$ equals the minimum cost of sending flow from s_k to t_k for $k \in Q$.

Hypotheses (i), (ii), (iii) and (iv) are valid for $q = 1$ and $|Q| = 1$ immediately after initialization. We first show that hypotheses (iii) is valid. At the start of the intermediate loop we set $\pi_i(Q) = \sum_{k \in Q} a(s_k, i)$ and $\pi_i(Q) = \min\ \{\pi_i(Q_1) + \pi_i(Q_2) : Q_1 \cap Q_2 = \Phi,\ Q_1 \cup Q_2 = Q\}$. This can be re-written as

$$\pi_i(Q) = \min\ \{\pi_i(Q_1)+\pi_i(Q_2) : Q_1 \cap Q_2 = \Phi,\ Q_1 \cup Q_2 = Q,\ 1 \leq |Q_1| \leq q-1,\ \sum_{k \in Q} a(s_k, i)\}.$$

By inductive hypothesis (i), $\pi_i(Q_1)$ and $\pi_i(Q_2)$ are the minimum costs of reaching node from nodes $s_k : k \in Q_1$ and nodes $s_k : k \in Q_2$. Since $|Q_1| \leq q - 1$, it follows that $\pi_i(Q)$ is the minimum cost of reaching node i if at most $q - 1$ commodities share an arc.

Consider hypotheses (i) and (ii). In the inner loop we choose a node $i \in \overline{S}$. Suppose we reach node i from some node $j \notin S$. From the way we choose node i, it follows that $\pi_j(Q) \geq \pi_i(Q)$. Since $w_{ij} \geq 0$, it follows that $\pi_j(i, Q) \geq \pi_j(Q) \geq \pi_i(Q)$, where $\pi_j(i, Q)$ is the minimum cost of reaching node i via node $j \notin S$. Hence, the cost of reaching node i from any node $j \notin S$ is at least as much as $\pi_i(Q)$. From hypothesis (ii), $\pi_i(Q)$ is the minimum cost of reaching node i using only nodes in S or the shortest paths from origins $s_k : k \in Q$ to node i. Hence, $\pi_i(Q)$ is the minimum cost of reaching node i, establishing hypothesis (i).

Consider hypothesis (ii). From hypothesis (iii), the minimum cost of reaching node $j \notin S$ with at most $q-1$ commodities sharing an arc is $\pi_j(Q)$. The cost of reaching node $j \notin S$ from nodes in S may go down after we shift node i to S. But this can only happen if we all Q commodities flow from i to j on arc (i, j). Since the algorithm updates the value of $\pi_j(Q)$, this establishes hypothesis (ii).

Thus, for a given set S, we obtain the optimum values of $\pi_i(Q)$ at the end of the inner iterations. The same arguments establish that hypotheses (i), (ii), and (iii) are valid for $\tau_i(Q)$.

Now consider hypothesis (iv). There are two cases to consider. In the first case, the optimum way of sending flow from s_k to t_k for $k \in Q$ involves some arc (i, j) that has Q commodities on it. In that case, min $\{\pi_i(Q) + \tau_i(Q) : i \in N\}$ gives the minimum cost. In the second case, any arc has at most $q - 1$ commodities on it in the optimum solution. Therefore, min $\{OPT(Q_1) + OPT(Q_2) : Q_1 \cap Q_2 = \Phi,\ Q_1 \cup Q_2 = Q,\ 1 \leq |Q_1| \leq q - 1\}$ is the minimum cost. The result follows.

□

Lemma 2 *Algorithm multi path takes $O(n^2 2^K + n3^K)$ iterations.*

Proof

For given q and Q, it is clear that there are $O(n^2)$ inner iterations. At the start of the intermediate loop we update the $\pi_i(Q)$ values over all nodes $i \in N$ and all subsets of Q. This takes $O(n2^q)$ time. Therefore, to complete all intermediate iterations for a given value of q takes $O({}^K C_q n^2 + {}^K C_q n 2^q)$

iterations. The total number of iterations is therefore

$$O(\sum_{q=1}^{K}({}^K C_q n^2 + {}^K C_q 2^q n)).$$

Since $(1+1)^K = \sum_{q=1}^{K} {}^K C_q$ and $(1+2)^K = \sum_{q=1}^{K} {}^K C_q 2^q$, it follows that the algorithm takes $O(n^2 2^K + n 3^K)$ iterations.

3 References

Balakrishnan, A., T.L.Magnanti and R.T.Wong (1989) "A dual ascent procedure for large-scale uncapacitated network Design", *Operations Research* 37, 716-740.

Balakrishnan, A. and N.R.Patel (1987) "Problem reduction methods and a tree generation algorithm for the Steiner network problem", *Networks*, 17, 65-85.

Dreyfus, S.E., and R.A.Wagner (1971) "The Steiner problem in graphs", *Networks* 1, 195-207.

Hakimi, S.L., (1971) "Steiner's problem in graphs and its implications", *Networks* 1, 113-133.

Hwang, F.K., D.S.Richards and P.Winter, (1992) "The Steiner tree problem", *Annals of Discrete Math.* 53, North-Holland.

Levin, A.Y., Algorithm for the shortest connection of a group of vertices, *Sov. Math. Dokl.* 12, 1477-1481.