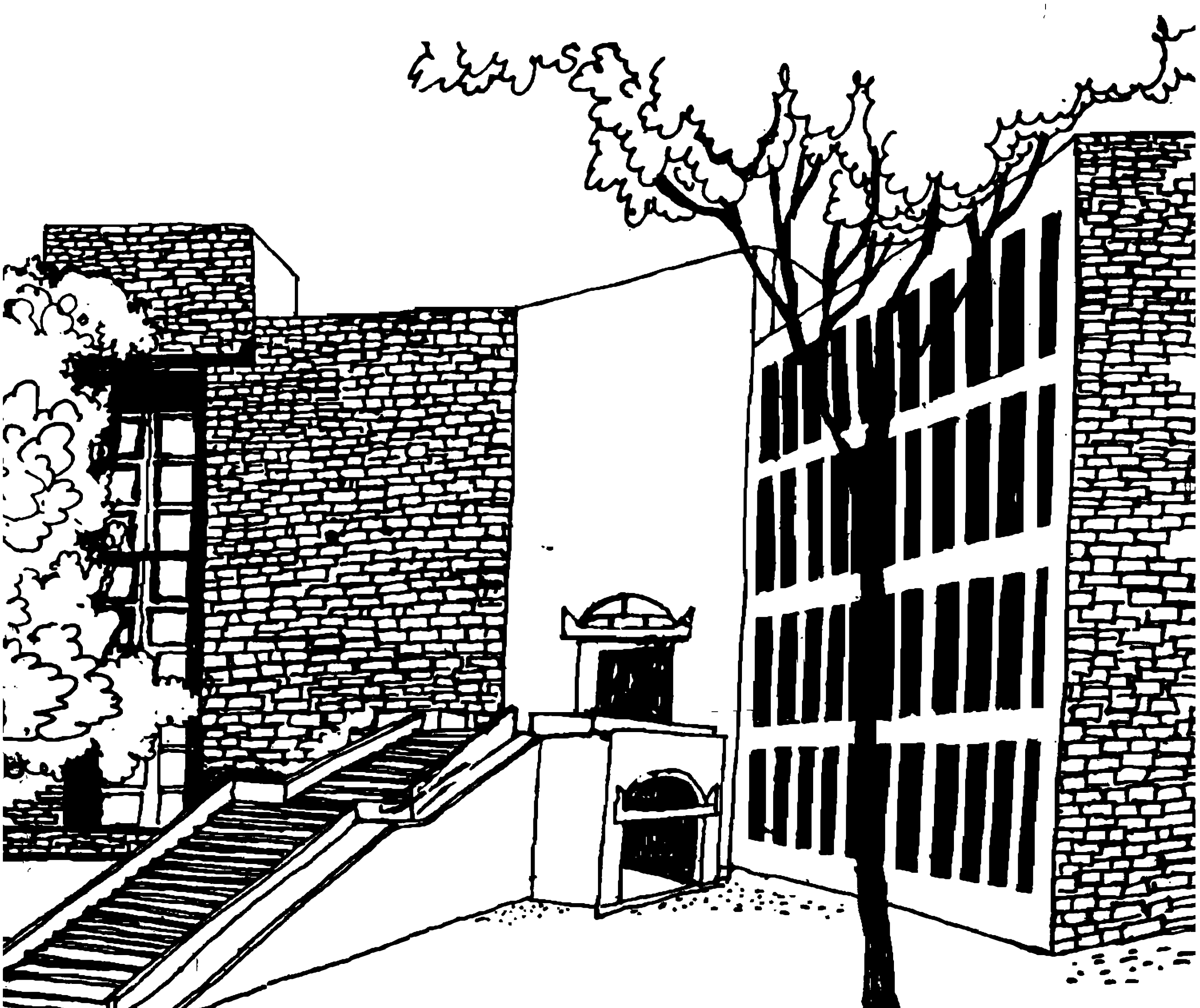
विद्याविनियोगाद्विकासः

IIM AHMEDABAD

Working Paper



TWO COMMODITY NETWORK DESIGN: THE CONVEX HULL

By

Trilochan Sastry

W.P. No.1411
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 study the uncapacitated and capacitated one facility versions of the two commodity network design problem. We characterize optimal solutions and show that we can restrict the search for optimal solutions to feasible solutions with at most one shared path. Using this characterization, we describe the convex hull of integer solutions to the uncapacitated problem using $O(m)$ variables and $O(n)$ constraints. We also describe how Dijkstra's shortest path algorithm can be used to solve the problem in a transformed graph with $O(n)$ nodes and $O(m)$ arcs. For the capacitated two commodity problem, we show that the problem can be solved either by using any standard shortest path algorithm or by the algorithm described for the uncapacitated case.

Key words and phrases: convex hull, network design. algorithm

1 Introduction

In this paper, we study the two commodity network design problem. We first consider the uncapacitated version of the problem. The multi commodity problem can be described as follows. Consider an undirected graph $G = (N, A)$, with node set N, arc set A, and origin destination pairs O_k, D_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. However, we show that if there are at most two commodities, the problem is easy and can be solved by a polynomial algorithm. Balakrishnan, Magnanti and Wong (1989) have studied the uncapacitated network design problem and solved large instances using a dual ascent based procedure. Hu(1963), Sakarovitch (1973), Seymour (1979) and Seymour (1980) have studied the two commodity flow problem. The uncapacitated multi commodity network design (UMC) problem can be formulated as follows.

Problem UMC

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

subject to:

$$\begin{aligned} \sum_{j}(x^k_{ji} - x^k_{ij}) &= \begin{cases} -1 & \text{if } i = O_k \\ 1 & \text{if } i = D_k \\ 0 & \text{otherwise} \end{cases} \\ y_{ij} &\geq x^k_{ij} + x^k_{ji} \\ x, y &\geq 0; y \in \{0,1\}. \end{aligned}$$

We designate the two commodity version of this problem as problem UTC. Let $m = |A|$ and $n = |N|$ denote the number of arcs and nodes repectively. The arcs are undirected and have symmetric cost, i.e., $w_{ij} = w_{ji}$. The flow variables x^k_{ij} are directed and have zero flow cost. In the next section we characterise the optimal solutions and describe a simple algorithm to solve the two commodity uncapacitated problem. In Section 3 we give an explicit reformulation for the problem in $O(m)$ variables and $O(n)$ constraints and show that it describes the convex hull of feasible integer solutions. In Section

4, we study the capacitated two commodity problem and show that it can either be solved by obtaining the shortest $O_k - D_k$ paths or that it reduces to the uncapacitated problem.

2 A Polynomial Algorithm

We characterise the optimal solutions and use this characterisation to obtain an algorithm to solve the problem. A *commodity k* path P_k connects O_k to D_k and has flow $x_{ij}^k + x_{ji}^k > 0$ for all arcs $(i,j) \in P_k$.

Lemma 1 *If (x, y) is an extreme point solution to UTC, then each commodity has one path and thus, $x_{ij}^k = 0$ or 1 for all arcs.*

Proof
Suppose commodity k has $m \geq 2$ paths $P_{k1} \ldots, P_{km}$ in some optimal solution. Let $\rho_{kq} = \min \{x_{ij}^k + x_{ji}^k : (i,j) \in P_{kq}\}$, $q = 1, \ldots, m$. By definition of a commodity path, $\rho_{kq} > 0$. We can re-route ρ_{k1} units of flow from path P_{k1} to path P_{k2} or ρ_{k2} units of flow from path P_{k2} to path P_{k1} to obtain two feasible solutions $(x(1), y(1))$ and $(x(2), y(2))$. Since

$$(x, y) = \frac{\rho_{k1}(x(2), y(2)) + \rho_{k2}(x(1), y(1))}{\rho_{k1} + \rho_{k2}},$$

it cannot be an extreme point.

□

Define an arc (i, j) to be *shared* if $x_{ij}^k + x_{ji}^k > 0$ for both commodities $k = 1, 2$. Define arc (i, j) to be a *shared forward arc* if $x_{ij}^1 > 0$ and $x_{ij}^2 > 0$, or $x_{ji}^1 > 0$ and $x_{ji}^2 > 0$. Arc (i, j) is a *shared reverse arc* if $x_{ij}^1 > 0$ and $x_{ji}^2 > 0$, or $x_{ji}^1 > 0$ and $x_{ij}^2 > 0$. Path P is shared if both $x_{ij}^1 + x_{ji}^1 > 0$ and $x_{ij}^2 + x_{ji}^2 > 0$ for all arcs (i, j) on the path.

Lemma 2 *There exists an optimal solution for UTC with at most one shared path such that all shared arcs are shared forward arcs or all shared arcs are shared reverse arcs.*

Proof
Consider any extreme point solution (x, y) with exactly one path P_k for

each commodity $k = 1.2$. Suppose arc $(i.j)$ is a shared forward arc, i.e., $x_{ij}^1 = x_{ij}^2 = 1$, and arc (u,v) is a shared reverse arc; i.e., $x_{uv}^1 = x_{vu}^2 = 1$. Without loss of generality assume that flow is directed from node j to node u for commodity 1 and node j to node v for commodity 2. Let $w_1(j,v)$ and $w_2(j,v)$ denote the cost of the arcs on path P_1 and P_2 between nodes j and v. If $w_1(j,v) > w_2(j,v)$, we can reduce the cost by re-routing commodity 1 flow between nodes j and v to path P_2. Hence. $w_1(j,v) \leq w_2(j,v)$. Since we are considering an optimal solution, $w_1(j,v) = w_1(j,u) + w_{uv}$ and $w_2(j,v) = w_2(j,u) - w_{uv}$. It follows that $w_1(j,u) + 2w_{uv} \leq w_2(j,u)$. Therefore. we can re-route commodity 2 flow between nodes j and u to path P_1 without increasing cost. But then, arc (u,v) is no longer a reverse shared arc. By repeating this procedure. we can eliminate all reverse arcs if there is a shared forward arc.

□

This result implicitly uses the fact that there are no flow costs, i.e., that the coefficient of x_{ij}^k in the objective fuunction is zero. Thus, the polyhedron defined by the constraints of UTC may have extreme points with more than one shared path or with both forward and reverse shared arcs. However, given the cost structure. it is sufficient to consider optimal solutions with the following property.

Corollary 1 *There exists an optimal solution in which each commodity has one path, and thus,* $x_{ij}^k = 0$ *or* 1 *for all arcs. Further, in this optimal solution, either all shared arcs are shared forward arcs or all shared arcs are shared reverse arcs.*

Remark 1 *In the case of two commodity flows, there exist optimal flows that are multiples of* 0.5. *However, in the case of two commodity uncapacitated network design, flows are integral.*

This result allows us to classify the optimal solution as either a *forward* or a *reverse* solution. Thus. in a forward solution. flows of both commodities on any shared path or arc are in the same direction. In a reverse solution. flows on any shared path or arc are in opposite directions.

We now derive some optimality conditions based on shortest distances. Let $a(i.j)$ be the shortest distance from i to j using w_{ij} as arc costs.

Lemma 3 *The objective function value ν corresponding to some feasible solution to UTC is optimal if and only if*
(i) $\nu \leq a(O_1, D_1) + a(O_2, D_2)$
(ii) $\nu \leq a(O_1, i) + a(O_2, i) + a(i, j) + a(j, D_1) + a(j, D_2)$ *and* $\nu \leq a(O_1, j) + a(O_2, j) + a(i, j) + a(i, D_1) + a(i, D_2)$ *for any two nodes i and j,*
(iii) $\nu \leq a(O_1, i) + a(O_2, j) + a(i, j) + a(j, D_1) + a(i, D_2)$ *and* $\nu \leq a(O_1, j) + a(O_2, i) + a(i, j) + a(i, D_1) + a(j, D_2)$ *for any two nodes i and j.*

Proof

The righthand sides of conditions (i), (ii) and (iii) are costs of feasible solutions. If any one of the conditions are not satisfied. then there is a lower cost solution. Therefore, if ν is optimal. the conditions are satisfied. The righthand sides of (ii) are the costs of feasible *forward solutions* and that of (iii) the cost of feasible *reverse solutions.* If ν is not optimal, then there is a solution with lower cost. By Lemma 2. there is an optimal solution with at most one shared path which has either all forward shared arcs or all reverse shared arcs. Hence. one of the conditions must be violated.

□

We describe the so called *two-path* algorithm to solve the problem. For the forward problem we define $s_k = O_k$ and $t_k = D_k$, while for the reverse problem, we define $s_1 = O_1$, $t_1 = D_1$, $s_2 = D_2$ and $t_2 = O_2$.

The algorithm adds a super source node s. a super sink node t, arc (s, t) of cost $a(s_1, t_1) + a(s_2, t_2)$, arcs (s, j) of cost $a(s_1, j) + a(s_2, j)$, and arcs (j, t) of cost $a(j, t_1) + a(j, t_2)$. It then uses any standard algorithm to find the shortest path between s and t. Notice that there are two passes for the algorithm. one for the forward problem. and the other for the reverse problem. Choose the shorter of the two shortest paths. If arc (s, j) belongs to the shortest path, replace it by the shortest paths from s_1 to j and from s_2 to j. Similarly. if arc (j, t) belongs to the shortest path. replace it by the shortest paths from j to t_1 and from j to t_2. We show later that this gives the optimal solution.

We use Dijkstra's algorithm here because we can find the distance label $\pi_j(f)$ ($\pi_j(b)$) for each node $j \in N$, which as we show later, represents the minimum cost of sending one unit of flow from nodes s_1 and s_2 to node j in the forward (reverse) problem. These labels are useful in proving that the reformulation in Section 3 is the convex hull of feasible integer solutions to *UTC*.

Algorithm two-path.
Solve the shortest path problem between nodes O_k and D_k for $k = 1, 2$. Assume that the shortest path trees rooted at O_1, O_2 and D_2 are known.

Let $A(i)$ be the set of arcs adjacent to node i.
begin
for $\delta = f, b$ **do**
begin
if $\delta = f$ **then** $s_k = O_k, t_k = D_k$ for $k = 1, 2$
if $\delta = b$ **then** $s_1 = O_1, t_1 = D_1,\ s_2 = D_2, t_2 = O_2$
add additional nodes s and t, and arc (s, t) of cost $w_{st} = a(s_1, t_1) + a(s_2, t_2)$
add arcs (s, j) of cost $w_{sj} = a(s_1, j) + a(s_2, j)$
add arcs (j, t) of cost $w_{jt} = a(j, t_1) + a(j, t_2)$
Initialise
$S \longleftarrow \Phi$
$\pi_j(\delta) = \infty$, $pred(j, \delta) \longleftarrow s$ for $j \in N$
$\pi_s(\delta) = 0$
while $|S| < n$ **do**
begin
let $i \in \overline{S}$ be a node for which $\pi_i(\delta) = \min\ \{\pi_j(\delta) : j \in \overline{S}\}$
$S \longleftarrow S \cup \{i\}$, $\overline{S} \longleftarrow \overline{S} - \{i\}$
for $j \in A(i)$ **if** $\pi_j(\delta) > \pi_i(\delta) + w_{ij}$ **then**
begin
$\pi_j(\delta) = \pi_i(\delta) + w_{ij}$
$pred(j, \delta) \longleftarrow i$
end
end$\{while\}$
end
$OPT = \min\ \{\pi_t(f),\ \pi_t(b)\}$
end$\{two - path\}$

Algorithm two path takes $O(n^2)$ iterations if the distances $a(s_k, j)$ and $a(j, t_k)$ are known. However, these distances can be obtained by finding the simple shortest path trees rooted at nodes s_k and t_k in at most $O(n^2)$ time. The complexity of the two path algorithm is therefore $O(n^2)$.

Theorem 1 *Algorithm two path solves the two commodity problem.*

Proof

Let $H(\delta)$ for $\delta = f, b$ denote the graphs obtained from G by adding nodes s and t, and arcs (s,j), (j,t) and (s,t). It follows from Dijkstra's algorithm that $\pi_j(\delta)$ is the shortest distance from node s to node j in $H(\delta)$. For any node $j \neq s, t$, let i be the first node not equal to s on some shortest path to j in $H(\delta)$. Then, the shortest distance is $a(s_1, i) + a(s_2, i) + a(i, j)$. But this is the cost of reaching node j from nodes s_1 and s_2 through node i. Consider the cost of reaching node j from s_1 and s_2 through any other node u (u might equal s_1, s_2 or j), where u is the first node not equal to s. This cost equals $a(s_1, u) + a(s_2, u) + a(u, j)$. But this is the cost of a path from s to j in $H(\delta)$. Therefore, $a(s_1, i) + a(s_2, i) + a(i, j) \leq a(s_1, u) + a(s_2, u) + a(u, j)$, and hence, $\pi_j(\delta)$ is the minimum cost of reaching node j from nodes s_1 and s_2.

Now consider the label $\pi_t(\delta)$ and any shortest path from s to t. Suppose the shortest path is not the arc (s,t). Let j be the last node not equal to t on this shortest path, and let i be the first node not equal to s. The cost of this path is $\pi_j(\delta) + a(j, t_1) + a(j, t_2) = a(s_1, i) + a(s_2, i) + a(i, j) + a(j, t_1) + a(j, t_2)$. If the shortest path is the arc (s,t), then the cost is $a(s_1, t_1) + a(s_2, t_2)$. In either case, the cost represents the cost of a feasible solution to $TFOC$.

Consider any feasible solution to UTC with exactly one shared path. Let u and v be the first and last nodes on the shared path. The cost of this solution is $a(s_1, u) + a(s_2, u) + a(u, v) + a(v, t_1) + a(v, t_2)$. But this represents the cost of a path from s to t in $H(\delta)$. A feasible solution to $TCOF$ without any shared path costs $a(s_1, t_1) + a(s_2, t_2)$, which is the cost of the arc (s,t). Hence, $\pi_t(\delta)$ is the minimum cost of reaching t_1 and t_2 from s_1 and s_2 in $H(\delta)$, and hence $OPT = \min\ \{\pi_t(f),\ \pi_t(b)\}$ is the cost of any optimal solution to UTC.

3 The Convex Hull

Several combinatorial problems can be solved in polynomial time. However, the convex hull of feasible integer solutions, if known, often has an exponential number of constraints. For instance, in the case of the spanning tree problem, for any $S \subset N$ if $A(S)$ denotes the set of arcs with both end nodes in S, then the inequalities $\sum_{e \in A(S)} y_e \leq |S| - 1$ completely describe the convex hull of integer solutions. However, there are $O(2^n)$ such inequalities. An extended reformulation in a polynomial number of variables and constraints is also known for this problem where we use a multi commodity formulation (see Magnanti and Wolsey (1995)). This extended formulation has $O(mn)$ variables and constraints whereas the original integer formulation has $O(m)$ variables. Similarly, the the convex hull of the single item uncapacitated lot sizing problem based on a natural formulation of the problem has $O(n)$ variables and an exponential number of constraints. and an extended formulation has $O(n^2)$ variables and constraints (see Pochet and Wolsey (1994)).

However. for the two commodity uncapacitated network design problem. we obtain the convex hull with $O(m)$ variables and $O(n)$ constraints. The natural formulation UTC has $O(m)$ variables and $O(m)$ constraints. We motivate the discussion by first showing that the linear programming relaxation of UTC gives rise to fractional optimal solutions.

Example 1

Consider a 4 node graph with $O_1 = 1$, $O_2 = 2$, $D_1 = 3$ and $D_2 = 4$. Arc costs are $w_{12} = w_{34} = 50$ and $w_{13} = w_{24} = 100$. An optimal solution is $x^1_{13} = y_{13} = 1$ and $x^2_{24} = y_{24} = 1$ with cost 200. However, the linear programming relaxation has the optimal solution $x^1_{13} = 0.5$, $x^1_{12} = x^1_{24} = x^1_{43} = 0.5$, $x^2_{24} = 0.5$. $x^2_{21} = x^2_{13} = x^2_{34} = 0.5$. and $y_{12} = y_{13} = y_{24} = y_{34} = 0.5$ with cost 150.

□

We therefore need a tighter reformulation if we want to obtain a complete description of the convex hull of integer solutions. Consider the following formulation and variable definitions. based on the characterisation of optimal solutions in Section 2. For arcs without shared flows. let $e^k_{ij}(f)$ and $e^k_{ij}(b)$ denote the flow on arc (i, j) for the forward and reverse solution repectively. Similarly, let $h_{ij}(f)$ and $h_{ij}(b)$ denote the flow on a shared arc where $h_{ij}(\delta) \in \{0.1\}$ for $\delta = f, b$. We define shared path P to be *maximal* if all shared arcs belong to it. We now reformulate the problem using the previous results. Assume $w_{ij} = w_{ji}$ for all $(i. j) \in A$.

Reformulation $R2$

$$\text{Min } \nu = \sum_{(i,j)\in A} \sum_{\delta=f,b} w_{ij}(e^1_{ij}(\delta) + e^2_{ij}(\delta) + h_{ij}(\delta))$$

subject to:

$$\sum_i (e^k_{ij}(\delta) + h_{ij}(\delta) - e^k_{ji}(\delta) - h_{ji}(\delta)) = 0, \ j \neq O_k, D_k; \quad k = 1 \text{ or } \delta = f \qquad (1)$$

$$\sum_i (e^2_{ij}(b) - h_{ij}(b) - e^2_{ji}(b) + h_{ji}(b)) = 0, \ j \neq O_2, D_2 \qquad (2)$$

$$\sum_i \sum_{\delta=f,b} (e^1_{ij}(\delta) - e^1_{ji}(\delta) + h_{ij}(\delta) - h_{ji}(\delta)) = \begin{cases} -1 & j = O_1 \\ 1 & j = D_1 \end{cases} \qquad (3)$$

$$\sum_i [\sum_{\delta=f,b} (e^2_{ij}(\delta) - e^2_{ji}(\delta)) + h_{ij}(f) - h_{ij}(b) - h_{ji}(f) + h_{ji}(b)] = \begin{cases} -1 & j = O_2 \\ 1 & j = D_2 \end{cases} \qquad (4)$$

$$e^k_{ij}(\delta), \ h_{ij}(\delta) \in \{0,1\}.$$

In this reformulation, constraint (1) ensures that flow conservation for commodity k is maintained for any node $j \neq O_k$ or D_k, $k = 1$ or $\delta = f$: either flow is on a shared arc or on a non-shared arc. Constraint (2) similarly ensures flow conservation for any node $j \neq O_k$ or D_k, $k = 2$ and $\delta = b$. However. the direction of the shared arcs are reversed for $k = 2$ to ensure reverse flows. $LR2$ is the linear programming relaxation obtained by relaxing the 0-1 constraints. Notice that the fractional solution in Example 1 is not feasible for $LR2$. and we obtain an optimal solution.

Any solution (x, y) to the two commodity problem UTC with at most one shared path can be transformed to obtain a feasible solution to the reformulation as follows. Assume that since $w_{ij} \geq 0$. $y_{ij} = x^k_{ij} + x^k_{ji}$ for all arcs. If all arcs on any shared path are all forward shared arcs, set $e^k_{ij}(f) = x^k_{ij}$ for all arcs that are not shared, and set $h_{ij}(f) = 1$ for all arcs on the shared path. We thus obtain a feasible solution. If the shared path has only reverse shared arcs. set $e^k_{ij}(b) = x^k_{ij}$ for all arcs that are not shared, and set $h_{ij}(b) = x^1_{ij} = 1$ for all arcs on the shared path to obtain a feasible solution for the reformulation. However. as Figure 1 for Example 2 shows. even this formulation gives fractional solutions.

INSERT FIGURE 1 HERE

Example 2

Arc costs are shown beside the arcs in Figure 1. Node 1 is the origin for both commodities, node 6 is the destination for commodity 1, and node 7 the destination for commodity 2. An optimal solution with cost 40 is

$$e_{12}^1 = e_{26}^1 = e_{14}^2 = e_{47}^2 = 1.$$

However, the following fractional solution

$$h_{12} = h_{14} = h_{35} = e_{26}^1 = e_{43}^1 = e_{56}^1 = e_{47}^2 = e_{23}^2 = e_{57}^2 = 0.5$$

costs only 37.5.

□

In this example, fractional flows share an arc, then split and then again combine to share arcs. To avoid this, we introduce some additional sets of variables as follows. Suppose there is a shared maximal path from node i^* to node j^* with commodity 1 flowing from i^* to j^*. We say that the shared path *starts* in node i^* and *ends* in node j^*. If $k = 1$ or $\delta = f$, then flow of commodity k on arc (i, j) is said to occur *before* the shared path if the flow has not yet entered node i^* and is said to occur *after* the shared path if it has left node j^*. If $k = 2$ and $\delta = b$, then flow of commodity 2 on arc (i, j) is said to occur *before* the shared path if the flow has not yet entered node j^* and is said to occur *after* the shared path if it has left node i^*. We define the following 0-1 variables.

For each node $j \in N$ let:

$u_j(\delta) = 1$ if the shared path starts in node j

$v_j(\delta) = 1$ if the shared path ends in node j

For each arc $(i, j) \in A$ let :

$e_{ij}^k = 1$ only if commodity k flow occurs before the shared path

$g_{ij}^k = 1$ only if commodity k flow occurs after the shared path

$h_{ij}(\delta) = 1$ if it is a shared arc.

The uncapacitated two commodity UTC problem can now be reformulated as follows.

Reformulation $UTC(R)$.

$$\text{Min } \nu = \sum_{(i,j)\in A} w_{ij}(e_{ij}^1 + e_{ij}^2 + g_{ij}^1 + g_{ij}^2 + \sum_{\delta=f,b} h_{ij}(\delta))$$

subject to:

$$\sum_i (e_{ij}^1 - e_{ji}^1) - u_j(f) - u_j(b) = \begin{cases} -1 & j = O_1 \\ 0 & otherwise \end{cases} \quad (5)$$

$$\sum_i (e_{ij}^2 - e_{ji}^2) - u_j(f) - v_j(b) = \begin{cases} -1 & j = O_2 \\ 0 & otherwise \end{cases} \quad (6)$$

$$\sum_i (h_{ij}(\delta) - h_{ji}(\delta)) + u_j(\delta) - v_j(\delta) = 0 \quad (7)$$

$$\sum_i (g_{ij}^1 - g_{ji}^1) + v_j(f) + v_j(b) = \begin{cases} 1 & j = D_1 \\ 0 & otherwise \end{cases} \quad (8)$$

$$\sum_i (g_{ij}^2 - g_{ji}^2) + v_j(f) + u_j(b) = \begin{cases} 1 & j = D_2 \\ 0 & otherwise \end{cases} \quad (9)$$

$$e_{ij}^k, g_{ij}^k, h_{ij}(\delta) \geq 0 \text{ for } k = 1,2;\ \delta = f,b.$$

Any solution (x,y) to the two commodity problem UTC with at most one shared path can be transformed to obtain a feasible solution to the reformulation as follows. Assume that since $w_{ij} \geq 0$. $y_{ij} = x_{ij}^k + x_{ji}^k$ for all arcs. If there is no shared path, then set $e_{ij}^k = x_{ij}^k$ for all arcs and set $u_{D_k}(f) = v_{D_k}(f) = 1$ to obtain a feasible solution to the reformulation. Otherwise, set $e_{ij}^k = x_{ij}^k$ ($g_{ij}^k = x_{ij}^k$) for all arcs (i,j) before (after) the shared path. If there is a shared forward path from node i^* to node j^*, then set $h_{ij}(f) = 1$ for all arcs on the shared path. Set $u_{i^*}(f) = v_{j^*}(f) = 1$. If there is a shared reverse path from node i^* to node j^*, then set $h_{ij}(b) = 1$ for all arcs on the shared path. Set $u_{i^*}(b) = v_{j^*}(b) = 1$. We thus obtain a feasible solution.

The reformulation has $12m + 4n$ variables: e_{ij}^k and e_{ji}^k, g_{ij}^k and g_{ji}^k for $k = 1,2$, $h_{ij}(\delta)$ and $h_{ji}(\delta)$ for $\delta = f,b$, and $u_j(\delta)$ and $v_j(\delta)$ for $\delta = f,b$. The original integer programming formulation has $5m$ variables, four flow and one fixed charge variable for each arc. Therefore, both formulations have $O(m)$ variables. The reformulation has $O(n)$ constraints whereas the original formulation has $O(m+n)$ constraints. We show that reformulation $UTC(R)$ completely describes the convex hull of feasible integer solutions.

Theorem 2 *The reformulation UTC(R) completely describes the convex hull of integer solutions for the uncapacitated two commodity network design problem.*

Proof.

The dual $DUTC(R)$ of $UTC(R)$ is given below.

$$\text{Max } \sum_{k=1}^{2}(\beta^k_{D_k} - \alpha^k_{O_k})$$

subject to:

$$\begin{array}{lrcl}
e^k_{ij}: & \alpha^k_j - \alpha^k_i & \leq & w_{ij} \\
g^k_{ij}: & \beta^k_j - \beta^k_i & \leq & w_{ij} \\
h_{ij}(\delta): & \eta_j(\delta) - \eta_i(\delta) & \leq & w_{ij} \\
u_j(f): & \eta_j(f) - \alpha^1_j - \alpha^2_j & \leq & 0 \\
v_j(f): & \beta^1_j + \beta^2_j - \eta_j(f) & \leq & 0 \\
u_j(b): & \eta_j(b) + \beta^2_j - \alpha^1_j & \leq & 0 \\
v_j(b): & \beta^1_j - \alpha^2_j - \eta_j(b) & \leq & 0
\end{array}$$

Using the $\pi_j(\delta)$ values from algorithm two path, let

$\alpha^k_j = a(O_k, j)$,
$\beta^1_j = a(O_1, D_1) - a(j, D_1)$,
$\beta^2_j = OPT - a(O_1, D_1) - a(j, D_2)$,
$\eta_j(f) = \pi_j(f)$, and
$\eta_j(b) = \pi_j(b) + a(O_1, D_1) - OPT$.

Notice that $\alpha^k_{O_k} = 0$, $\beta^1_{D_1} = a(O_1, D_1)$ and that $\beta^2_{D_2} = OPT - a(O_1, D_1)$. Hence if the dual variables are feasible, they are optimal.

From algorithm two path notice that $\pi_j(f) \leq a(O_1, j) + a(O_2, j)$ and that $\pi_j(b) \leq a(O_1, j) + a(j, D_2)$. Since $\pi_j(f)$ is the minimum cost of sending one unit of flow from each of the nodes O_1 and O_2 to node j, and since $a(j, D_1) + a(j, D_2)$ is an upper bound on the cost of sending one unit from node j to each of nodes D_1 and D_2, it follows that

$$\pi_j(f) + a(j, D_1) + a(j, D_2) \geq OPT.$$

Since $\pi_j(b)$ is the minimum cost of sending one unit of flow from node O_1 to node j, and one unit from j to D_2, and since $a(j, D_1) + a(O_2, j)$ is an upper bound on the cost of sending one unit from node j to node D_1 and from O_2 to j, it follows that

$$\pi_j(b) + a(j, D_1) + a(O_2, j) \geq OPT.$$

It is now easy to verify that these values of the dual variables satisfy dual feasibility.

4 The Capacitated Problem

We now consider the capacitated network design problem. The multi commodity problem can be described as follows. Consider an undirected graph $G = (N, A)$, with node set N, arc set A, and origin destination pairs O_k, D_k, with demand of d_k unit between every pair for $k = 1, \ldots, K$. Capacity can be purchased in batches of C units 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.

Magnanti, Mirchandani and Vachani (1995) have studied the *two facility version* of the problem, where capacity is available in batches of 1 or C units, and describe facets and strong valid inequalities for the problem. Chopra, Gilboa and Sastry (1996) studied the single origin-destination version of the one and two facility problem where they describe an exact algorithm and an extended formulation for the problem. The multi commodity one facility problem $MCOF$ is NP-complete since the uncapacitated version is NP-complete. The problem can be formulated as follows.

Problem $MCOF$

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

subject to:

$$\sum_j (x^k_{ji} - x^k_{ij}) = \begin{cases} -d_k & \text{if } i = O_k \\ d_k & \text{if } i = D_k \\ 0 & \text{otherwise} \end{cases}$$

$$C y_{ij} \geq \sum_{k=1}^{2} (x^k_{ij} + x^k_{ji})$$

$$x, y \geq 0; y \text{ integer}.$$

Let $d_k = \mu_k C + r_k$ where we define r_k, the *residue* as $r_k = C$ if d_k is a multiple of C. Define the following two problems associated with $TCOF$. The first is the *full flow problem FF* of sending $\mu_k C$ units from O_k to D_k, and the other is the *residual flow problem RF* of sending r_k units from O_k to D_k. These problems can be formulated as follows.

Problem FF

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

subject to:

$$\sum_j (x_{ji}^k - x_{ij}^k) = \begin{cases} -\mu_k C & \text{if } i = O_k \\ \mu_k C & \text{if } i = D_k \\ 0 & \text{otherwise} \end{cases}$$

$$C y_{ij} \geq \sum_{k=1}^{2} (x_{ij}^k + x_{ji}^k)$$

$$x, y \geq 0; y \text{ integer.}$$

Problem RF

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

subject to:

$$\sum_j (x_{ji}^k - x_{ij}^k) = \begin{cases} -r_k & \text{if } i = O_k \\ r_k & \text{if } i = D_k \\ 0 & \text{otherwise} \end{cases}$$

$$\min\{C, r_1 + r_2\} y_{ij} \geq \sum_{k=1}^{2} (x_{ij}^k + x_{ji}^k)$$

$$x, y \geq 0; y \text{ integer.}$$

Lemma 4 *The full flow problem can be solved by finding separately the shortest paths from O_k to D_k for $k = 1, 2$ using w_{ij} as arc costs.*

Proof

Let $e_{ij}^k = x_{ij}^k / C$. Then the problem reduces to the flow balance equations requiring μ_k units of flow between each $O_k - D_k$ pair, and the constraints $y_{ij} \geq \sum_{k=1}^{2} (e_{ij}^k + e_{ji}^k)$.

The dual of the problem therefore is

$$\max \sum_k \mu_k(\alpha^k_{D_k} - \alpha^k_{O_k})$$

$$\begin{aligned} &\text{subject to:} \\ \alpha^k_j - \alpha^k_i - \gamma_{ij} &\leq 0 \\ \gamma_{ij} &\leq w_{ij}. \end{aligned}$$

We obtain a dual feasible solution by setting $\gamma_{ij} = w_{ij}$ and $\alpha^k_j = a(O_k, j)$. Consider the primal solution obtained by sending μ_k units of commodity k flow on the shortest path from O_k to D_k for $k = 1, 2$ using w_{ij} as arc costs, and setting $y_{ij} = \sum_k(x^k_{ij} + x^k_{ji})$ on all arcs. It is easy to verify that the primal and dual solutions satisfy complementary slackness conditions.

□

We modify the definition of a shared arc as follows. Arc (i, j) is *shared* if $x^1_{ij} + x^1_{ji} > 0$, $x^2_{ij} + x^2_{ji} > 0$ and

$$\lceil \frac{x^1_{ij} + x^1_{ji}}{C} \rceil + \lceil \frac{x^2_{ij} + x^2_{ji}}{C} \rceil > \lceil \frac{x^1_{ij} + x^1_{ji} + x^2_{ij} + x^2_{ji}}{C} \rceil.$$

Notice that according to this definition, if $x^k_{ij} + x^k_{ji} = \rho C$ for some integer ρ, then arc (i, j) cannot be shared. Thus, an arc is shared only if $(\rho - 1)C < x^k_{ij} + x^k_{ji} < \rho C$ for both commodities for some integer $\rho \geq 1$. We define two paths P_1 and P_2 as independent if any arc $(i, j) \in P_1 \cap P_2$ is not shared. Notice that if two paths are independent, then there is no cost saving even if they have arcs in common. Moreover, if y_{ij} equals 0 or 1 on all arcs $(i, j) \in P_1 \cup P_2$, then $y_{ij} \geq 2$ if arc $(i, j) \in P_1 \cap P_2$. For any cut $S \subset N$ of nodes, let $A(S) = \{(i, j) : i \in S, j \notin S, \text{ or } j \in S, i \notin S\}$ denote the arcs across the cut. We show that there are two cases to consider for $TCOF$, each of which is easy to solve. The first case is when $r_1 + r_2 \leq C$, and the second case is when $r_1 + r_2 > C$. We first establish a preliminary result.

Lemma 5 *There is an optimal solution for TCOF with at most one shared path.*

Proof

We show that we can find an optimal solution with at least μ_k paths between

O_k and D_k, each of which is independent of all other paths. Consider any optimal solution to $TCOF$ and let y_{ij}^* denote the value of variables y_{ij} in this solution. Let $r^* = 1$ if any cut $S \subset N$ separating O_1, O_2 from D_1, D_2 has capacity $\mu_1+\mu_2+1$ (which is possible only if $r_1+r_2 \leq C$), and $r^* = 2$ if all cuts in the optimal solution have a capacity of $\mu_1 + \mu_2 + 2$. Since $w_{ij} \geq 0$, we can restrict attention to optimal solutions with $y_{ij} \leq \mu_1+\mu_2+r^*$ on any arc (i,j). Transform the network to obtain the graph G^* as follows. Split each arc (i,j) into $\mu_1 + \mu_2 + r^*$ parallel arcs with capacity b_{ij}^m for $m = 1, \ldots, \mu_1 + \mu_2 + r^*$, and set $b_{ij}^m = 1$ if $y_{ij}^* \geq m$ and set $b_{ij}^m = 0$ if $y_{ij}^* < m$.

For any commodity k and any cut set $S \subset N$ such that $O_k \in S$, $D_k \notin S$, $\sum_{(ij)\in A(S)} y_{ij}^* \geq \mu_k + 1$ in the optimal solution to $TCOF$. Therefore, $\sum_{(ij)\in A(S)} \sum_m b_{ij}^m \geq \mu_k + 1$. By Mengers's theorem (see Bondy and Murty (1976)), there are $\mu_k + 1$ arc distinct paths between O_k and D_k in G^*.

Similarly, for any cut $S \subset N$ such that $O_1, O_2 \in S$ and $D_1, D_2 \notin S$, $\sum_{(ij)\in A(S)} \sum_m b_{ij}^m \geq \mu_1 + \mu_2 + r^*$. Add a super source node s and a super sink node t and add arcs $(s, O_1), (D_1, t)$, with capacity $\mu_1 + 1$, and arcs $(s, O_2), (D_2, t)$ with capacity $\mu_2 + r^* - 1$.

Solve the maximum flow problem between s and t on this network. Clearly the maximum flow equals $\mu_1+\mu_2+r^*$. If the total flow between O_1 and D_1 is μ_1+1, then the total flow between O_2 and D_2 is μ_2+r^*-1. Hence, the total capacity of all paths between O_1 and D_1 is $\mu_1 + 1$ and between O_2 and D_2 is $\mu_2 + r^* - 1$. If $r^* = 2$, this implies that there are $\mu_k + 1$ arc distinct paths with unit capacity between O_k and D_k in G^*. Hence, in the original graph G, we can send $\mu_k C + r_k$ units of flow from O_k to D_k along paths that are independent. If $r^* = 1$, then we can send $\mu_k C$ units on independent paths in G, and there is a shared path for the residual flows.

Suppose the total flow between O_1 and D_1 in the maximum flow in G^* is μ_1+1-m for some $m > 0$. From the integrality property of maximum flows, assume that m is an integer. Then, there must be m units of flow from O_1 to D_2 and m units of flow from O_2 to D_1. Call these m units the *diverted* flows. We also have $\mu_2 + r^* - m - 1$ units of flow from O_2 to D_2. Notice that $m \leq \min\{\mu_1 + 1,\ \mu_2 + r^* - 1\}$. Clearly, there are m *diverted* paths between O_1 and D_2, and m diverted paths between O_2 and D_1. Label these paths as $P_1(q)$ and $P_2(q)$, $1 \leq q \leq m$, and let $\mathcal{P}_1$ and $\mathcal{P}_2$ denote these sets of paths. Further, given the y_{ij}^* values in the optimal solution to $TCOF$, we can send $(\mu_1 + 1 - m)C$ units of flow from O_1 to D_1 on paths each of which is independent of any other path. Similarly, we can send $(\mu_2 + r^* - m - 1)C$ units of flow from O_2 to D_2 on paths each of which is independent of any other path.

Transform the graph G^* as follows. Delete the two super nodes and add super nodes s_0 and t_0 and arcs $(s_0, O_1), (D_1, t_0)$ with capacity $\mu_1 + 1$, and arcs $(s_0, D2), (O_2, t_0)$ with capacity $\mu_2 + r^* - 1$. Solve the maximum flow problem between s_0 and t_0 in G^o. Clearly, the maximum flow equals $\mu_1 + \mu_2 + r^*$. If the total flow between O_1 and D_1 equals $\mu_1 + 1$, we are done.

Otherwise, we have diverted flows (i.e., flows between O_1 and O_2, and between D_2 and D_1), of magnitude $0 < m_0 \leq \min\{\mu_1 + 1,\ \mu_2 + r^* - 1\}$, and two sets of paths $\mathcal{P}_3$ and $\mathcal{P}_4$ each with m_0 paths $P_3(q)$ and $P_4(q)$, $1 \leq q \leq m_0$ connecting the node pairs $O_1 - O_2$ and $D_2 - D_1$. Let $m^* = \min\{m, m_0\}$. Clearly, we can send $(\mu_1 + 1 - m^*)C$ units of commodity 1 flow between O_1 and D_1, and $(\mu_2 + r^* - 1 - m^*)C$ units of commodity 2 flow between O_2 and D_2 on independent paths, and there are m^* paths that are diverted, i.e. that do not connect O_k to D_k. If $m \leq m_0$, arbitrarily delete $m_0 - m$ paths from each of the sets of paths $\mathcal{P}_3$ and $\mathcal{P}_4$. If $m > m_0$, arbitrarily delete $m - m_0$ paths from each of the sets of paths $\mathcal{P}_1$ and $\mathcal{P}_2$. Thus we have four sets of paths $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$, and $\mathcal{P}_4$ each having m^* paths. Consider the graph obtained from the union of these paths. We can choose one path from each of these 4 sets and arbitrarily establish a one to one correspondence between them.

Consider any four paths P_1, P_2, P_3, P_4 in this graph connecting $O_1 - D_2$, $O_2 - D_1$, $O_1 - O_2$ and $D_1 - D_2$, obtained from this one to one correspondence. Let i_1 be the last node common to paths P_1 and P_3 starting from node O_1, i_2 the last node common to P_2 and P_3 starting from node O_2, i_3 the last node common to P_2 and P_4 starting from node D_1, and i_4 the last node common to P_1 and P_4 starting from node D_2. For any node $i \in P_1$, let $\alpha(i)$ be the sequence number on the path starting from O_1 with $\alpha(O_1) = 1$.

If $\alpha(i_1) \geq \alpha(i_4)$, then we can proceed from O_1 to i_4 along P_1, and then switch to P_4 and proceed from i_4 to D_1. Similarly, we can proceed from O_2 to i_1 along P_3, and from i_1 to D_2 along P_1. Thus we have two arc disjoint paths, one between O_1 and D_1, and the other between O_2 and D_2, which are therefore, independent.

Similar arguments establish that we have arc disjoint paths if:

$\beta(i_2) \geq \beta(i_3)$, where $\beta(i)$ is the sequence number of node i on path P_2 starting from node O_2, or
$\gamma(i_1) \geq \gamma(i_2)$, where $\gamma(i)$ is the sequence number of node i on path P_3 starting from node O_1, or
$\delta(i_3) \geq \delta(i_4)$, where $\delta(i)$ is the sequence number of node i on path P_4 starting from node D_1.

Finally, suppose $\alpha(i_1) < \alpha(i_4)$, $\beta(i_2) < \beta(i_3)$, $\gamma(i_1) < \gamma(i_2)$ and $\delta(i_3) < \delta(i_4)$. This implies we have the configuration in Figure 2. Notice that even if there are arcs $(i,j) \in P_1 \cap P_2$ (or $(i,j) \in P_3 \cap P_4$), these arcs are independent, and hence, have been shown schematically as two separate arcs in the figure, one on P_1 and the other on P_2, (one on P_3 and the other on P_4) each with unit capacity.

INSERT FIGURE 2 HERE

If the cost of the path from i_1 to i_4 on P_1 using w_{ij} as arc costs, is at most equal to the cost of the path from i_2 to i_3 on P_2, we can delete arcs on P_2 between i_2 to i_3 and add an extra unit of capacity between i_1 to i_4 along P_1. This does not increase the cost. But we now obtain a solution with two independent paths, one between O_1 and D_1, and the other between O_2 and D_2. If the cost of the path from i_1 to i_4 is greater than the cost of the path from i_2 to i_3, we can delete arcs on P_1 between i_1 and i_4, and add arcs between i_2 to i_3 along P_2. Thus, we have obtained two independent paths connecting the two origin destination pairs from the 4 paths P_1, P_2, P_3, P_4.

By using the same procedure for each set of four corresponding paths, we can obtain m^* independent paths between O_k and D_k for $k = 1,2$. If $r^* = 2$, we obtain $\mu_k + 1$ independent paths between O_k and D_k. If $r^* = 1$, we obtain μ_k independent paths between O_k and D_k, and one additional path between each of the origin destination pairs which share exactly one path.

□

If $r_1 + r_2 > C$, then any cut separating O_1, O_2 from D_1, D_2 in G^* has a capacity of at least $\mu_1 + \mu_2 + 2$, and hence, the maximum flow in G^* equals at least $\mu_1 + \mu_2 + 2$. Hence, there are $\mu_k + 1$ independent paths for each commodity k. Therefore, with the given values of y^*_{ij} in the optimal solution to $TCOF$, we can send $\mu_k C + r_k = d_k$ units of commodity k from O_k to D_k

for each commodity on independent paths. We have therefore established the following result.

Theorem 3 *If $r_1 + r_2 > C$, then TCOF can be solved by sending d_k units along the shortest path between O_k and D_k for $k = 1, 2$ using w_{ij} as arc costs.*

If $r_1 + r_2 > C$, the convex hull for $TFOC$ is therefore represented by the usual network flow constraints for two shortest path problems, one for each commodity, in which we send $\mu_k + 1$ units from origin O_k to destination D_k.

Theorem 4 *If $r_1 + r_2 \leq C$, we can solve TCOF by solving the full flow and the residual flow problems separately. Moreover, the residual flow problem reduces to the uncapacitated two commodity problem UTC.*

Proof

As shown in Lemma 5, there are μ_k independent paths between each origin destination pair in G^*. Therefore, for the problem $TCOF$, we can send $\mu_k C$ units of flow on the shortest path between O_k and D_k. Hence $TCOF$ can be solved by solving FF and RF separately.

Consider the residual flow problem RF. Since $r_1 + r_2 \leq C$, $y_{ij} = 1$ is a sufficient value for y_{ij} on any shared arc. Hence, the constraint $\min\{C, r_1 + r_2\} y_{ij} \geq \sum_{k=1}^{2} (x_{ij}^k + x_{ji}^k)$ can be replaced by the constraints $r_k y_{ij} \geq x_{ij}^k + x_{ji}^k$ for $k = 1, 2$. The problem can therefore be recast as follows by using the substitution $r_k e_{ij}^k = x_{ij}^k$.

$$
\begin{aligned}
\text{Min} \quad & \sum_{(i,j) \in A} w_{ij} y_{ij} \\
& \text{subject to:} \\
& \sum_j (e_{ji}^k - e_{ij}^k) = \begin{cases} -1 & \text{if } i = O_k \\ 1 & \text{if } i = D_k \\ 0 & \text{otherwise} \end{cases} \\
& y_{ij} \geq e_{ij}^k + e_{ji}^k \\
& e, y \geq 0; y \text{ integer.}
\end{aligned}
$$

This is precisely the two commodity uncapacitated problem.

If $r_1 + r_2 \leq C$, then the convex hull is represented by the constraints for the full flow problem FF and the constraints for the uncapacitated problem's reformulation $UTC(R)$. Problem FF can be solved by any shortest path algorithm, and problem $UTC(R)$ by the two path algorithm in Section 2.

5 Extension to Multi Commodity

For the uncapacitated multi commodity problem the following result holds.

Lemma 6 *There is an optimal solution in which each commodity k flows on exactly one path between O_k and D_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.

□

Using this result, it is perhaps possible to generalize the two path algorithm to the multi commodity case. For each node i and any subset $Q \subset \{1, \ldots, K\}$ of the set of commodities, we can calculate $\pi_i(Q)$ which is the optmimum cost of reaching node i from origin nodes $O_k : k \in Q$. This approach, if it works, would give an exponential time exact algorithm since $\pi_i(Q)$ needs to be calculated for all subsets Q.

Notice that in the two commodity capacitated problem $TCOF$, we show that we can solve the full flow and the residual flow problems separately, and that there is an optimal solution in which the flow on any arc equals 0, r_k, $\mu_k C$ or $\mu_k C + r_k$. This enables us to classify arcs as either full flow or residual flow arcs and solve the problem efficiently. However, the following example for the three commodity problem shows that these results do not hold.

Example 3
Consider a 4 node, 4 arc problem with arc capacity 10. Let node s be the source node for the three commodities whose destination nodes are 1, 2 and 3. Let arc costs be $w_{s1} = w_{s2} = 2$, $w_{13} = w_{23} = 1$, and let $d_1 = d_2 = 4$, and $d_3 = 12$.

An optimal solution with cost 6 is :

$y_{s1} = y_{s2} = y_{13} = y_{23} = 1$, and $x^1_{s1} = x^2_{s2} = 4$, $x^3_{s1} = x^3_{s2} = x^3_{13} = x^3_{23} = 6$.

However, the full flow problem has the optimal solution:

$y_{s1} = y_{13} = 1$, $x^3_{s1} = x^3_{13} = 10$

with cost 3, and the residual flow problem has the solution:

$y_{s1} = y_{13} = y_{23} = 1$, and $x^1_{s1} = x^2_{s2} = x^2_{13} = x^2_{32} = 4$, $x^3_{s1} = x^3_{13} = 2$

with cost 4. The total cost is therefore 7.

□

Therefore a simple generalisation of the solution approach for the two commodity problem is unlikely to help in solving the problem. However, an efficient heuristic which assumes that arcs flows are either full or residual can perhaps be obtained by generalising the two path algorithm for the two commodity case.

6 Conclusions

We characterize optimal solutions of the two commodity network design problem and show that the search for optimal solutions can be confined to solutions with at most one shared path for both the capacitated and uncapacitated problems. Using this characterization, we describe the convex hull of integer solutions to the uncapacitated problem using $O(m)$ variables. We also describe an $O(n^2)$ algorithm to solve the problem. For the capacitated two commodity problem, we show that we can separate the problem into the full flow and the residual flow problems. We further show that the problem can be solved using any standard shortest path algorithm when total residue is large (i.e., $r_1 + r_2 > C$), and that it reduces to the uncapacitated case when total residue is small (i.e., when $r_1 + r_2 \leq C$).

A generalization of the two path algorithm can perhaps be used to obtain an exact algorithm for UMC and efficient heuristics for $MFOC$. It might also be useful to find more general conditions under which the multi commodity capacitated problem is easy.

Acknowledgements

The author is indebted to Yogesh Agarwal for the proof of Lemma 5.

7 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.

Bondy, J.A., and U.S.R.Murty, "Graph Theory with Applications", North Holland, Amsterdam (1976).

Chopra,S., I.Gilboa and T.Sastry (1996) "Algorithms and Extended Formulations for One and Two Facility Network Design", *Integer Programming and Combinatorial Optimization, 5th International IPCO Conference, 1996, Springer.*

Hu, T.C. (1963) "Multi-commodity Network Flows" *Operations Research* 11, 344-360.

Magnanti,T.L., P.Mirchandani and R.Vachani (1995) "Modelling and Solving the Two Facility Capacitated Network Loading Problem", *Operations Research*, 43, 142-157.

Magnanti, T.L. and L.A.Wolsey, "Optimal Trees", *Handbooks in OR & MS*, vol 7, Elsevier Science.

Pochet, Y. and L.A.Wolsey (1994), "Algorithms and Reformulations for Lot Sizing Problems", CORE Discussion paper 9427, Universite Catholique de Louvain.

Sakarovitch, M. (1973) "Two Commodity Network Flows and Linear Programming", *Math.Prog.* 4, 1-20.

Seymour, P.D. (1979) "A Short Proof of the Two-Commodity Flow Theorem", *Journal of Combinatorial Theory* B 26, 370-371.

Seymour, P.D. (1980) "Four-Terminus Flows", *Networks* 10, 79-86.

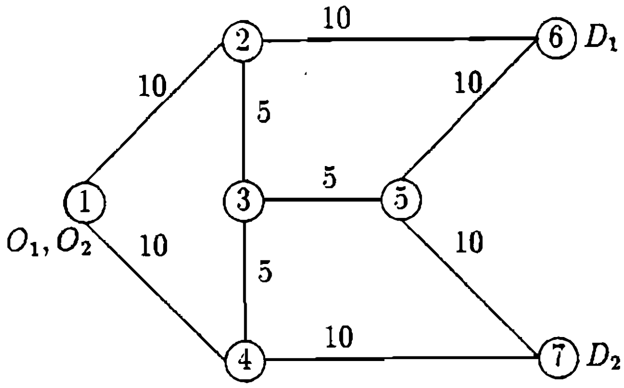


FIGURE 1

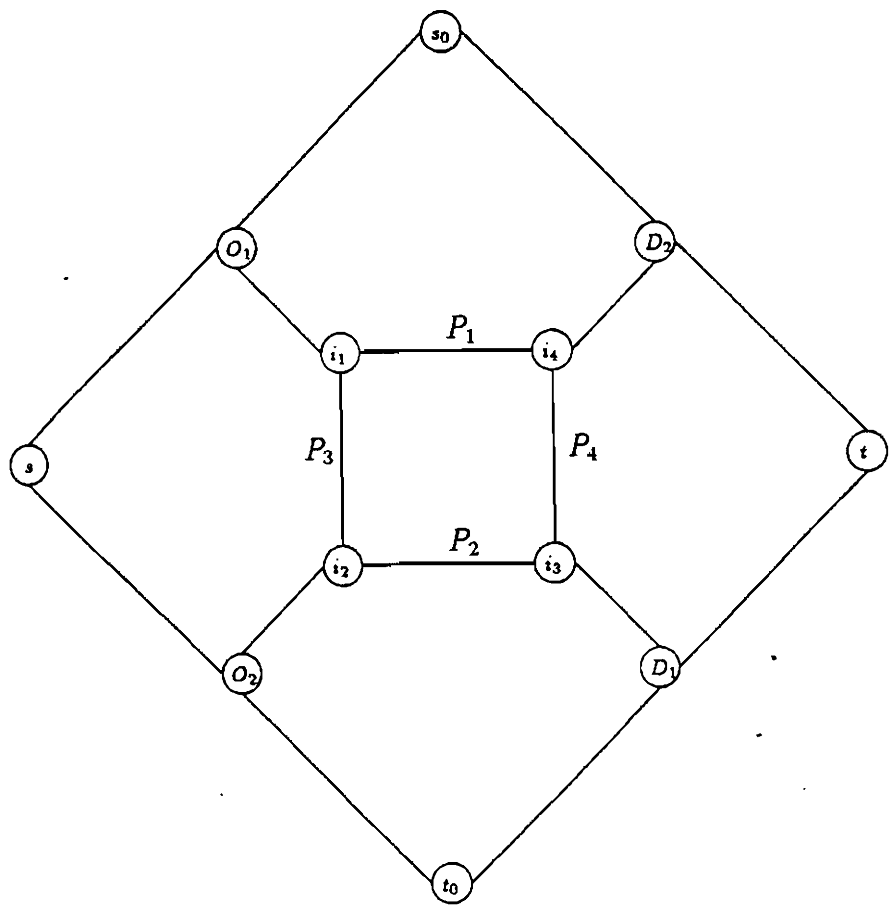


FIGURE 2