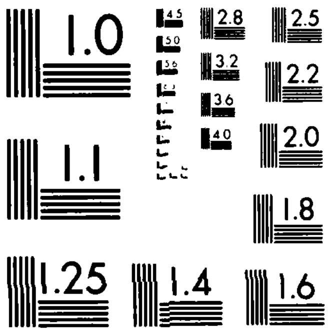
AD-A152 343 A GREEDY ALGORITHM FOR THE TRANSHIPMENT ALONG A SINGLE ROAD PROBLEM(U) TEXAS UNIV AT AUSTIN CENTER FOR CYBERNETIC STUDIES A I ALI OCT 84 CCS-500 1/1

UNCLASSIFIED N00014-82-K-0295 F/G 12/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

(3)

Research Report CCS 500
A GREEDY ALGORITHM FOR THE TRANSHIPMENT
ALONG A SINGLE ROAD PROBLEM

by

A.I. Ali

CENTER FOR CYBERNETIC STUDIES

The University of Texas
Austin, Texas 78712

(3)

Research Report CCS 500
A GREEDY ALGORITHM FOR THE TRANSHIPMENT
ALONG A SINGLE ROAD PROBLEM

by

A.I. Ali

October 1984

This paper was partly supported by ONR Contract N00014-82-K-0295 with the USARI Contract MDSX90#-83-K-0312 Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
College of Business Administration 5.202
The University of Texas at Austin
Austin, Texas 78712-1177
(512) 471-1821

ABSTRACT

This paper presents a specialized algorithm for the transhipment along a single line problem. The problem is a specially structured network flow problem for which the basis structure is such that a greedy algorithm can be employed for solution. The specialized algorithm is on the order of a hundred times faster than the primal simplex method on a graph.

Key Words: Algorithms; Networks.

The problem of transhipment along a single road [2] may be formulated as a minimum cost network flow problem,

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax = r \\ & x \geq 0 \end{aligned}$$

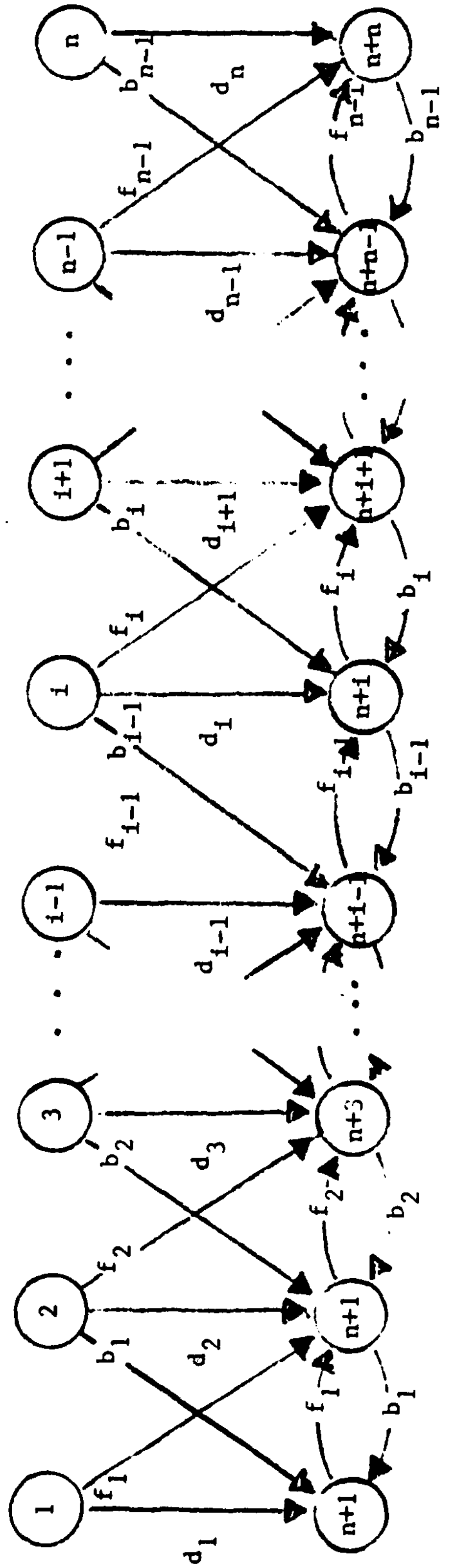
where A is a node-arc incidence matrix for a network on 2n nodes with 5n-4 arcs as given in Figure 1. The vectors x, c, and r are respectively the vectors of decision variables, cost coefficients and requirements. The underlying network consists of supply points, i, (i = 1,2,...,n) each with supply supply s_i and n demand points, j, (j =n+1, n+2, ...,n+n) each with a demand d_j. Arcs of the form (i,j) i = 1,2,...,n, j = n+1,n+2,...,n+n are referred to as transportation arcs and arcs of the form (i,j), i = n+1,n+2,...,n+n, j = n+1,n+2,...,n+n are referred to as transhipment arcs. Arcs (i,n+i+1) i=1,2,...,n-1, and arcs (i,i+1) i = n+1, n+2,...,n+n-1, have cost f_i (these arcs are called forward arcs) ; arcs (i,n+i-1), i = 1,2,...,n-1 and arcs (i,i-1), i = n+1, n+2, ..., n+n-1, have cost b_i (these arcs are called back arcs); arcs (i,n+i), i=1,2,...,n (called direct arcs) have cost d_i. There is an added stipulation that

$$d_i \leq f_i + b_i \tag{1}$$

$$d_i \leq f_{i-1} + b_{i-1} \tag{2}$$

and that total supply be equal to total demand.

The problem is essentially one of determining how the flow



1
2
3
i-1
i
i+1
n-1
n
n+1
n+3
n+n
d1
d2
d3
di-1
di
di+1
dn-1
dn
f1
f2
fi-1
fi
fn-1
b1
b2
bi-1
bi
bn-1

Figure 1.

distributes itself over the arcs of the network. We shall see that this may be determined by asking the following questions:

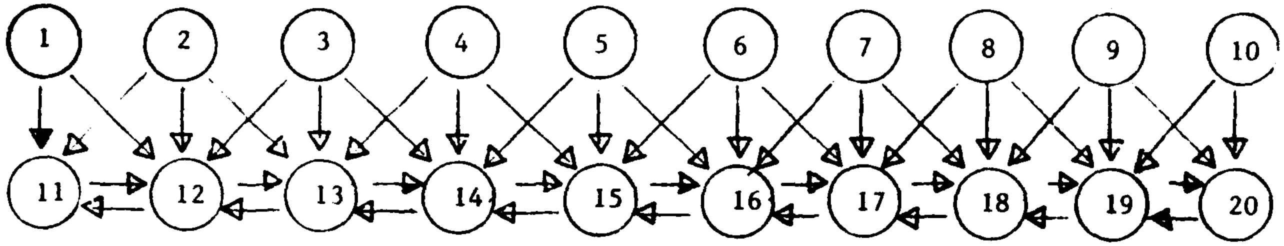
(1) When are transhipments necessary? This essentially asks when does there have to be flow along the transhipment arcs?

(2) When is shipment over the forward and backward arcs preferable to shipment along the direct arcs?

In answering the first question we obtain a decomposition of the flow into flow on arcs necessary for feasibility and flow satisfied by transportation arcs alone. The answer to the second question determines the distribution of the rest of the flow among the forward, back and direct arcs.

In order to answer the first question, we examine the basis structure for the problem: The number of basic arcs for this problem is $2n - 1$, of which at least n must be transportation arcs and at most $n-1$ can be transhipment arcs. Because of the conditions (1) and (2), transhipments arcs will never be admissible for basis entry in primal simplex pivots. As such, the only flow on these arcs possible is flow due to the supply/demand structure. That is, flow necessary for feasibility.

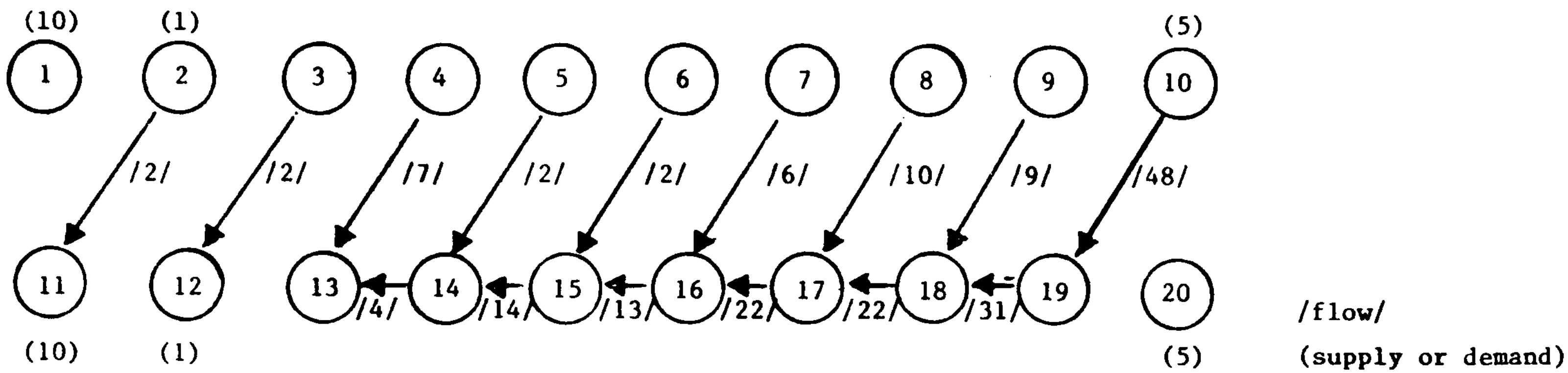
A feasible solution which pulls back flow or pushes forward flow along a transhipment line minimally is trivially obtained. If excess supply exists at some point, then this needs to be pushed forward along the transhipment line. If excess demand exists, then this demand must be met by pulling flow back along the transhipment line. By minimally, is meant the minimal amount of flow required to be along transhipment arcs to ensure feasibility. For example, consider the problem in Figure 2. This problem requires, for feasibility, that the transhipment arcs take on the flow given in Figure 3.



i	s_i	d_i	d_i	f_i	b_i
1	10	12	15	14	9
2	3	3	12	9	13
3	2	11	13	7	6
4	7	12	11	9	8
5	2	5	15	9	6
6	6	19	15	6	15
7	10	5	9	6	13
8	5	18	7	13	12
9	9	17	9	10	13
10	53	5	11	-	-

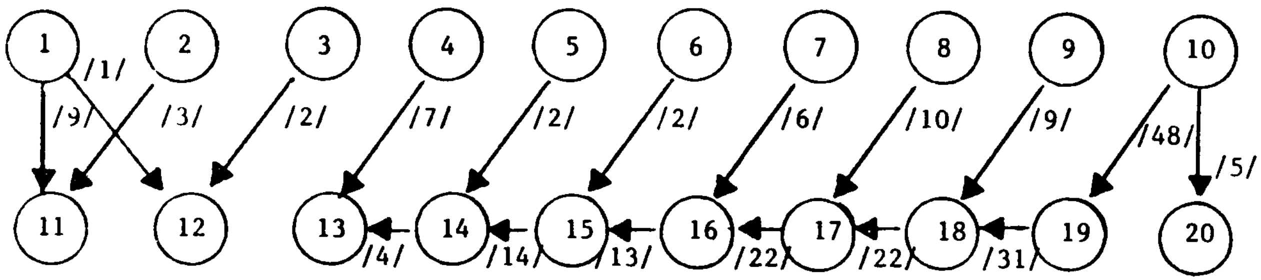
<u>Figure 2.</u>

Flow in arcs to ensure feasibility



(a)

Optimal Flow



(b)

Figure 3.

After the determination of transhipment flow, the supply at node i is equal to the demand at node $n+i$. The question is then whether or not shipment along the forward and backward arcs is cost beneficial when compared with direct shipment. Note that if

$$d_i + d_{i+1} \leq f_i + b_i,$$

then direct shipment is preferable.

It is well known that if a constant is added to the costs on all arcs which emanate from a node, then the solution to the problem remains invariant. Thus, by subtracting d_i from all arcs which emanate from node i, $i = 1,2,3,\ldots,n$, we have a problem in which all direct costs are equal to 0 and the costs on forward and back arcs are given by f_i' and b_i' where

$$f_i' = f_i - d_i \qquad i = 1,2,\ldots,n-1$$

$$b_i' = b_i - d_{i+1} \qquad i = 1,2,\ldots,n-1$$

For two pairs of supply and demand nodes, i, $n+i$ and $i+1,n+i+1$, the question of whether the forward and backward arcs are cost beneficial is answered by examining the quantity, $z_i = f_i' + b_i'$.

If

$f_i' + b_i' < 0$, then arcs $(i,n+i+1)$ and $(i+1,n+i)$ may have flow on them.

It is now apparent that the most cost effective arcs for the entire problem are arcs $(k,n+k+1)$, $(k+1,n+k)$ where k is such that

$$z_k = \min_i \{ z_i : z_i < 0 \} \qquad (3)$$

If all the quantities, z_i are ≥ 0, then, direct shipments are optimal. Thus, by ordering the quantities, z_i in ascending order, we obtain a natural greedy method for the solution of the second part of the problem: In succession, ship as much as is possible, while ensuring feasibility, along forward and back acrs which correspond to the most negative z_i. When there are no more negative z_i left, make remaining shipments along the direct arcs.

The above translates into the following algorithm for the solution of the transhipment along a line problem: The variables back, forward and direct represent the flow on back forward and direct arcs respectively. The back and forward arcs have flow which represent the net flow being shipped from point i+1 to i and from point i to i+1, respectively.

<u>Step_0</u> **Initialize.**

(a) forward(i),back(i),direct(i)=0, flag(i) = off,
    i = 1,2,...,n.

<u>Step_1</u> **Determine flow necessary for feasibility.**

For i = 1 to n-1

(a) **Pull back.** If (supply(i) < demand(n+i))
    back(i) = demand(n+i)-supply(i)
    supply(i+1)=demand(n+i+1)-back(i)
    demand(n+i)=demand(n+i)-back(i)

(b) **Push forward.** If (supply(i) > demand(n+i))
    forward(i)=supply(i)-demand(n+i)
    demand(n+i+1)=demand(n+i+1)-forward(i)
    supply(i)=supply(i)-forward(i)

(c) If(supply(i) = demand(n+i)) skip.

Step 2 Determine rest of flow on forward and back arcs.

(a) Let k be such that

z(k) = min{f(i)-d(i)+b(i)-d(i+1), i=1,...,n-1, flag(i)=off.}

(b) Flag(k)=on.

(c) If(z(k) < 0)

Determine maximum flow which can be put on forward(k),back(k), and make this allocation.

If(z(k) ≥ 0) go to step 3.

Step 3 Determine flow on direct arcs.

For all i such that supply(i) > 0, direct(i)=supply(i).

It is straightforward to see that all steps of the algorithm are linear except for Step 2 which requires a sort and a search to determine the flow. Step 2c involves a simple computation of the minimum of the supplies at point k and point k+1. If there is a series of points, say l,l+1,...,l+t such that z(l), z(l+1),..., z(l+t) all have the same value, then there may be alternate assignments of flow for these points. In such an event, when the flow for z(l-1) or z(l+t+1) is to be assigned, Step 2c involves the comparison of the flows on the forward and back arcs of points l,l+1 and l+2.

Implementation of the Algorithm:

The algorithm has been coded in standard FORTRAN and been tested on randomly generated problems. The computational testing is summarized in Table 1. For purposes of comparison, the problems generated were also solved by a NETFLO [1], a general purpose code for the solution of minimum cost network flow problems. The test problems generated had supplies and demands drawn from a uniform distribution (5,100) and costs distributed uniformly between 1 and 50. The computational times are very stable and show that the algorithmic specialization that results for the transhipment along a line problem is on the order of a hundred times faster than the unspecialized primal simplex algorithm on a network.

Table 1.

Problem Seed	Nodes	Optimal Solution	Solution Time* Greedy Alg.	NETFLO
11111	50	146514	.004	.129
12345	50	128559	.003	.190
34367	50	103761	.003	.122
33889	50	159346	.003	.136
11229	50	741488	.004	.220
15834	100	153026	.010	.213
33890	100	1105341	.010	.685
67811	100	626802	.010	.559
98123	100	618579	.009	.660
58923	100	1094640	.011	.571
58923	150	2285600	.020	1.324
93451	150	1876195	.020	1.699
12345	150	647033	.020	1.305
11111	150	1344982	.019	1.353
78600	150	996977	.019	1.592

*CPU seconds on the Dual Cyber 170/50 at the University of Texas at Austin.

References

1. Helgason, R.V., and Kennington, J. L., Network Programming, John Wiley.

2. Posner, M. and Szwarc, W., "Transhipment Along a Single Road."

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CCS 500	2. GOVT ACCESSION NO. A152 342	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Greedy Algorithm for the Transhipment Along a Single Road Problem		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) I. Ali		8. CONTRACT OR GRANT NUMBER(s) N00014-82-K-0295
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Cybernetic Studies The University of Texas at Austin Austin, Texas 78712		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research (Code 434) Washington, D.C.		12. REPORT DATE October 1984
		13. NUMBER OF PAGES 20
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Algorithms; Networks

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper presents a specialized algorithm for the transhipment along a single line problem. The problem is a specially structured network flow problem for which the basis structure is such that a greedy algorithm can be employed for solution. The specialized algorithm is on the order of a hundred times faster than the primal simplex method on a graph.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

END

FILMED

5-85

DTIC