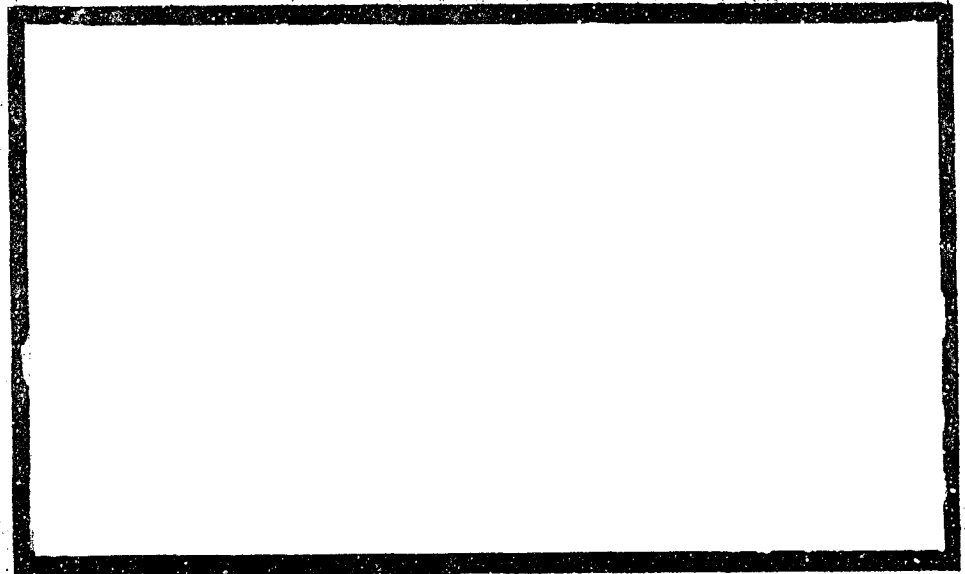


OPERATIONS RESEARCH GROUP

AD693140

AD693140



DDC
REFILED
SEP 15 1969
RECEIVED
B

Columbia University

School of Engineering and Applied Science

NEW YORK, N. Y. 10027

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

22

Operations Research Group
Columbia University
New York, New York

AN EXPERIMENTAL EVALUATION
OF
SOME METHODS OF SOLVING
THE
ASSIGNMENT PROBLEM

by

Michael Florian and Morton Klein

Technical Report No. 41
Dec. 30, 1969

This research was supported by the Army,
Navy, Air Force and N.A.S. under a
contract administered by the Office of
Naval Research

Contract Number N000 14-67-A-0108

Reproduction in whole or in part is per-
mitted for any purpose of the United
States Government.

ABSTRACT

Computational experiments were conducted with three methods for solving the assignment problem: Kuhn's Hungarian method, a primal method due to Balinski and Gomory, and a negative cycle method proposed by Klein. Kuhn's method is seen to be the best of the three.

AN EXPERIMENTAL EVALUATION OF SOME METHODS OF SOLVING
THE ASSIGNMENT PROBLEM*

By

Michael Florian** and Morton Klein
Columbia University

1. INTRODUCTION AND SUMMARY

In [4] a general method was proposed for finding minimal cost flows in networks; for reasons which will be made apparent in the next section we call it the "negative cycle" method. The purpose of this paper is to report on some computational experiments with two variants of this method as applied to the special minimal cost flow problem known as the "assignment problem", with the Balinski-Gomory algorithm for the assignment problem [1], and with the Hungarian method for the assignment problem proposed by Kuhn [5].

The results of our experiments, detailed in section 4 are easily summarized: Kuhn's Hungarian method was the most efficient (i.e., the fastest) of the methods tested. Over the range of problem sizes with which we worked (10 x 10 to 100 x 100 problems) it was approximately two to five times

* This research was supported by the Army, Navy, Air Force and NASA under a contract administered by the Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the United States Government.

** Currently at the University of Montreal.

faster than the next best Balinski-Gomory algorithm. The efficiency of variants of the negative cycle method is directly dependent on the efficiency of the sub-routine which is used to locate such cycles. Two such sub-routines - the Floyd-Murchland method for finding all shortest routes in a network ([2], [6]) and the Ford-Fulkerson algorithm for finding shortest routes from a particular vertex to all others [3] were tested. Although the Ford-Fulkerson algorithm was found to be substantially the better of the two, it is still not good enough to make the negative cycle method competitive with the others.

2. THE ASSIGNMENT PROBLEM

In brief, the assignment problem is to fill n jobs by as many men at least total cost. Let $a_{ij} > 0$ represent the cost of assigning man i ($i = 1, \dots, n$) to job j ($j = n+1, \dots, 2n$), A an arbitrary assignment of n men to n jobs, $v(A)$ the cost of such an assignment (i.e., $v(A) = \sum_A a_{ij}$), and $\mathcal{A}$ the set of all $(n!)$ possible assignments. Then, an optimal assignment A^0 , by definition, satisfies

$$(1) \quad v(A^0) \leq v(A) \quad , \quad \text{for all } A \in \mathcal{A}$$

i.e.

$$\sum_{A^0} a_{ij} \leq \sum_A a_{ij} \quad , \quad \text{for all } A \in \mathcal{A}.$$

Evidently an assignment A^* is optimal if and only if there is no assignment A such that

$$(2) \quad \sum_A a_{ij} - \sum_{A^*} a_{ij} < 0 \quad , \quad \text{for all } A \neq A^*.$$

Now, suppose we associate the following bipartite directed graph $G(A^*)$ with assignment A^* . Vertices of the first part will correspond to men and those of the second part to jobs. A directed edge connects job j to man i

with associated cost $-a_{ij}$, if $a_{ij} \in A^*$; otherwise, an edge connects man i to job j with associated cost a_{ij} . Thus for a small (say $n = 3$) problem with, say, $A^* = \{a_{i,i+3} : i = 1,2,3\}$ the graph has the following appearance.

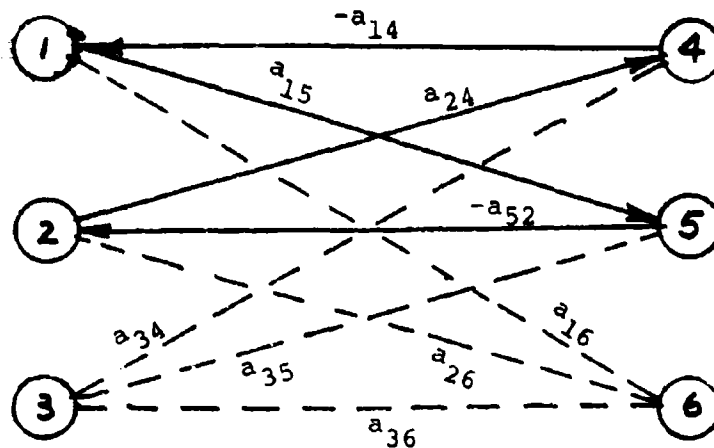


Figure 1: The Graph $G(A^*)$

In such a graph any directed cycle C consists of an even number of distinct directed edges with alternate edges associated with assignment A^* . A cycle is of interest because it can be used as a way of comparing the value of A^* with that of another assignment, say A' , where A' is identical to A^* except for those components associated with the edges of the cycle C . Within C the components of A^* are associated with the negative cost edges and those of A' with the positive cost edges. For example, in Figure 1, suppose $C = \{(1,5), (5,2), (2,4), (4,1)\}$ (shown with solid edges), then the components of A' in C are (1,5) and

(2,4) while those of A^* are (5,2) and (4,1). The difference in value between A' and A^* is $a_{15} - a_{52} + a_{24} - a_{41}$. If this difference is negative, then A' is a better assignment than A^* .

More generally, given a directed cycle C in $G(A^*)$ and an associated assignment $A(C)$, the value of the cycle $v(C)$ is given by

$$\begin{aligned}
 (3) \quad v(A) - v(A^*) &= \sum_{A(C)} a_{ij} - \sum_{A^*} a_{ij} \\
 &= \sum_{CCA} a_{ij} - \sum_{CCA^*} a_{ij} = v(C)
 \end{aligned}$$

From (2) we know that the assignment A^* will be optimal if and only if there is no assignment A such that $v(C) < 0$. In terms of the graph $G(A^*)$ this means that A^* is optimal if and only if $G(A^*)$ contains no cycles whose value is negative.

3. A METHOD FOR SOLVING THE ASSIGNMENT PROBLEM

The preceding remarks suggest the following general method for solving the assignment problem [4].

Preliminary: Choose an arbitrary assignment.

Step 1: Construct the graph associated with the assignment.

Step 2: Test the graph for the existence of a cycle whose value is negative.

(a) If there are none - stop; the given assignment is optimal.

(b) If a negative valued cycle exists, go to Step 3.

Step 3: Trace the negative cycle (i.e. locate it) and exchange the components of the current assignment with those of the other components of the cycle. This yields a new improved assignment. Return to Step 1.

Testing the graph for the existence of a negative cycle can be done in a variety of ways. However, all of those known to us to be "reasonably" efficient, are methods for solving shortest route problems in graphs containing some negative costs (distances). The two which we have explored are:

- 1) The Floyd-Murchland method for finding all shortest routes in a network [2], [6] and

- 2) The Ford-Fulkerson algorithm [3] for finding a shortest route from a particular vertex to all others in the network.

Both of the above algorithms "work", in the sense of finding the shortest paths, if the network does not contain any negative cycles, and both break down if such cycles exist. This condition, when it occurs, leads to Step 3 of the proposed procedure: tracing the cycle so that a new improved assignment can be determined. Here also we have explored two methods:

- 1) the Ford-Fulkerson method used above, also provides a simultaneous cycle tracing routine, and
- 2) the use of a "routing" matrix which, when constructed simultaneously with the application of the Floyd-Murchland method, serves to trace the vertices involved in the negative cycle.

The details of the two methods are given in Appendices A and B.

4. THE COMPUTATIONAL EXPERIMENT

One hundred distinct assignment problems (10 each of size 10×10 , 20×20 , ..., 100×100) were generated using a uniform distribution on the integers from 0 to 50. Each of the 10×10 to 50×50 problems were used to test all four computational methods using an IBM 360/75 computer. These first runs demonstrated that the negative cycle method, with either sub-routine, was not competitive with the other two, hence the remainder of the experiment involved only the Balinski-Gomory and Kuhn algorithms. The results are given in Table 1, below.

PROBLEM SIZE	NEGATIVE CYCLE METHOD		BALINSKI-GOMORY METHOD	KUHN'S HUNGARIAN METHOD
	WITH FLOYD-MURCHLAND	WITH FORD-FULKERSON		
10 x 10	.63 (.28, 1.65)	.10 (.05, .17)	.04 (.02, .05)	.02 (.02, .03)
20 x 20	7.7 (3.4, 12.5)	.70 (.35, 1.02)	.18 (.12, .25)	.10 (.07, .13)
30 x 30	40 (29, 50)	2.2 (1.9, 2.5)	.55 (.43, .77)	.26 (.15, .50)
40 x 40	121 (67, 156)	4.4 (2.2, 6.1)	1.1 (.9, 1.4)	.38 (.17, .73)
50 x 50	209 (75, 444)	8.1 (4.9, 10.4)	1.9 (1.5, 2.3)	.51 (.35, .70)
60 x 60			3.1 (2.7, 3.9)	.92 (.62, 1.36)
70 x 70			4.6 (4.0, 5.1)	1.5 (.9, 2.5)
80 x 80			6.4 (5.7, 7.3)	1.4 (1.1, 1.8)
90 x 90			9.1 (8.1, 10.0)	1.8 (1.4, 3.2)
100 x 100			11.4 (10.1, 12.6)	2.0 (1.5, 2.6)

TABLE 1: AVERAGE SOLUTION TIMES (SECS.) - 10 TRIALS EACH;
(MINIMUM AND MAXIMUM SOLUTION TIMES IN PARENTHESIS).

The relative merits of the two methods of detecting and tracing negative cycles may be evaluated more precisely by looking at the time per iteration. The reason for this is that each iteration (except the last in each trial) includes exactly one successful search for such a cycle; thus the average time per iteration is approximately equal to the time needed to locate and trace a negative cycle. The results of such an evaluation are shown in Table 2. As can be seen the Ford-Fulkerson shortest route algorithm is still the better method.

PROBLEM SIZE	NEGATIVE CYCLE METHOD	
	WITH FLOYD-MURCHLAND	WITH FORD-FULKERSON
10 x 10	.63/3 = .21	.10/2 = .05
20 x 20	7.7/9 = .86	.70/5 = .14
30 x 30	42/16 = 2.6	2.2/8 = .28
40 x 40	121/20 = 6.0	4.4/10 = .44
50 x 50	209/15 = 14	8.1/12 = .67

TABLE 2: AVERAGE SOLUTION TIME (SECS.) PER AVERAGE
NUMBER OF ITERATIONS TO SOLUTION

5. CONCLUDING REMARKS

A) Our evaluation of the performance of the algorithms tested assumes that the computer programs employed for each method are equally "efficient" in implementing the algorithms. If this is not the case, we believe that the large difference between the performances of the various algorithms suggest that the results would hold for improved programs.

B) Although the negative cycle method showed up as the least efficient of those tested, it still is potentially improvable. It awaits the development of an efficient means of detecting (and tracing) the existence of such cycles in a directed graph. Its current, and perhaps only, virtue is its ease of presentation in the classroom - independent of linear programming theory.

C) The assignment problem experiments suggest similar relative performances for the generalizations of each of these methods for the transportation problem. If this is true, the Ford-Fulkerson algorithm (a generalization of Kuhn's method) for the transportation problem is also likely to be the best of the group.

A P P E N D I X A

NEGATIVE CYCLE ALGORITHM FOR THE ASSIGNMENT PROBLEM WITH FORD & FULKERSON SHORTEST ROUTE METHOD

1. Start computations with a feasible solution.

1a. Determine an initial solution using Dantzig's rule.†

Assign $x_{pq} = 1$ for $a_{pq} = \min_{i,j} [a_{ij}]$

Cross out row p and column q

$x_{p'q'} = 1$ for $a_{p'q'} = \min_{i \neq p, j \neq q} [a_{ij}]$ and so on

until a complete assignment is found.

1b. Define the matrix $A'(x)$

$$a'_{ij} = \begin{cases} -a_{ij} & \text{if } x_{ij} = 1 \\ a_{ij} & \text{if } x_{ij} = 0 \end{cases}$$

2. Use the following heuristic rule†† to determine an entry point for step 3.

For $i = 1, 2, \dots, n$ we have an assignment j_i such that

† One of the presumed advantages of a "primal" method is its' ability to use "good" starting solutions. Since we do not know of any studies indicating whether this rule is better than others, its selection was arbitrary.

†† The purpose of this rule is to try to get as "close" to a negative cycle as possible. Since if the initial point is far from such a cycle, it will take longer to find it.

$x_{ij} = 1$ and calculate

$$\Delta_i = \left\{ \min_{j \neq j_i} \{a_{ij}\} + \min_{k \neq i} \{c_{kj_i}\} \right\}, \Delta_i^* = \min_i \Delta_i$$

$$\text{and } \{j_i^* = r \mid \Delta_i^* = \min_i \Delta_i\}$$

Start at column r and apply the labeling routine in an attempt to find the negative cycle.

Let (M_i, d_i) denote the label of column j in $A'(x)$. M_i indicates the existence of a chain from the initial point to M_i and an edge toward the column (vertex) j such that the total cost (distance) is d_i .

Similarly a row label is (J_j, d_j) .

3. If J_r is the initial point - Label column r $(-, 0)$

3a. Mark each row with label $(J_r, |a'_{ir}|)$

3b. Mark each column $j \neq r$, $(M_i, d_i + \bar{a}_{ij})$,
where $\bar{a}_{ij} = \{a'_{ji} \mid a'_{ji} \leq 0 \text{ and } x_{ij} = 1\}$

3c. Label each row by

$$\{J_k, \min_j \{d_j + |a'_{ij}|\}\}$$

where J_k represents the column at which the minimum is attained.

3d. Label each column $(M_i, d_i + \bar{a}_{ij})$

Continue steps b and c until the initial point's label becomes negative. If a set of row labels is duplicated - stop - optimal solution reached.

4. The negative cycle C , is identified by tracing back from the initial point according to the succession of adjacent

row and column labels until the initial point is encountered for the second time or some other vertex is encountered twice, indicating a negative cycle that does not involve the initial point.

5. The negative cost cycle is represented by a row vector $C(X)$. Modify the flow X as follows

$$x'_{ij} = \begin{cases} x_{ij} , & \text{if } (M_i, J_j) \notin C \\ 0 , & \text{if } (M_i, J_j) \in C \text{ and } x_{ij} = 1 \\ 1 , & \text{if } (M_i, J_j) \in C \text{ and } x_{ij} = 0 \end{cases}$$

return to step 1b.

A P P E N D I X B

NEGATIVE CYCLE ALGORITHM FOR THE ASSIGNMENT PROBLEM BASED ON THE FLOYD-MURCHLAND "ALL SHORTEST ROUTES" METHOD

1. Start computations with feasible solution

1a. Use Dantzig's rule

Assign $x_{pq} = 1$ for $a_{pq} = \min_{i,j} [a_{ij}]$.

Cross out row p and column q

$x_{p'q'} = 1$ for $a_{p'q'} = \min_{\substack{i \neq p \\ j \neq q}} [a_{ij}]$

and so on until a complete assignment is found.

1b. Define D^0 a $2n \times 2n$ matrix with elements

$d_{ij}^0 = \infty$, $i, j = 1, \dots, n$ and $i, j = n+1, \dots, 2n$;
 $= a_{ij}$, $i = 1, \dots, n$ and $j = n+1, \dots, 2n$;
 $= -a_{ij}$, $x_{ji} > 0$, $i = n+1, \dots, 2n$, $j = 1, \dots, n$;
 $= \infty$, $x_{ji} = 0$, $i = n+1, \dots, 2n$, $j = 1, \dots, n$.

Note: In the computations, the matrix $X = \{x_{ij}\}$ is not required.

2. Test for the existence of a negative cycle by applying the Floyd-Murchland algorithm to the D^0 matrix.

2a. For $k = 1, \dots, n$ compute

$d_{ij}^{(k)} = \min [d_{ik} + d_{kj}, d_{ij}^{(k-1)}]$ for $i, j \neq k$,

and

2b. Record "ROUTE" on routing matrix $U = \{u_{ij}\}$

$$u_{ij}^{(k)} = \begin{cases} k & , \quad \text{if } d_{ik} + d_{kj} < d_{ij}^{(k-1)} \\ u_{ij}^{(k-1)} & , \quad \text{otherwise} \end{cases}$$

and

$$u_{ij}^{(0)} = [0]_{2n \times 2n}.$$

If a substitution is made in $[d_{ij}]$ such that the shortest route from $i \rightarrow j$ is through vertex k , the index k serves to trace the path of the shortest distance from $i \rightarrow j$.

2c. Test for all k the value of d_{ii}

If $d_{ii} \geq 0$ for all i and $k < n$, go to 2a.

If $d_{ii} \geq 0$ for all i and $k = n$, stop, optimal solution reached.

If $d_{ii} < 0$ for some i record $r = \{i | d_{ii} < 0\}$ and go to step 3.

3. Cycle Tracing

Let C be a row vector that contains the cycle elements.

The maximum size of this vector is $(2n + 1)$.

3a. Initiate search with $c_1 = r$; $c_2 = u_{rr}$; $c_3 = r$

Let k denote the number of elements in C , ($k = 3$ at this stage).

3b. For $i = 1, \dots, 2n$

Let $i_1 = c_i$; $i_2 = c_{i+1}$.

If $u_{i_1 i_2} = 0$ return to start of 3b.

If $u_{i_1 i_2} \neq 0$ then modify c_i as follows

$$c'_{i+1} = u_{i_1 i_2} ,$$

$$c'_i = c_i ,$$

$$c'_{j+2} = c_{j+1} \quad \text{for } j = 1, \dots, k$$

and

$$k' = k + 1;$$

then go back to start of 3b.

When this procedure terminates, $C = (c_i)$ contains the cycle elements and k equals the number of entries in the cycle.

The number of vertices involved in the cycle is $k - 1$.

4. Modify solution

4a. For $i = 1, \dots, k$

Let $i_1 = i$, $i_2 = i + 1$

4b. If $c_{i_1} > c_{i_2}$ go to 4c, otherwise go to 4d.

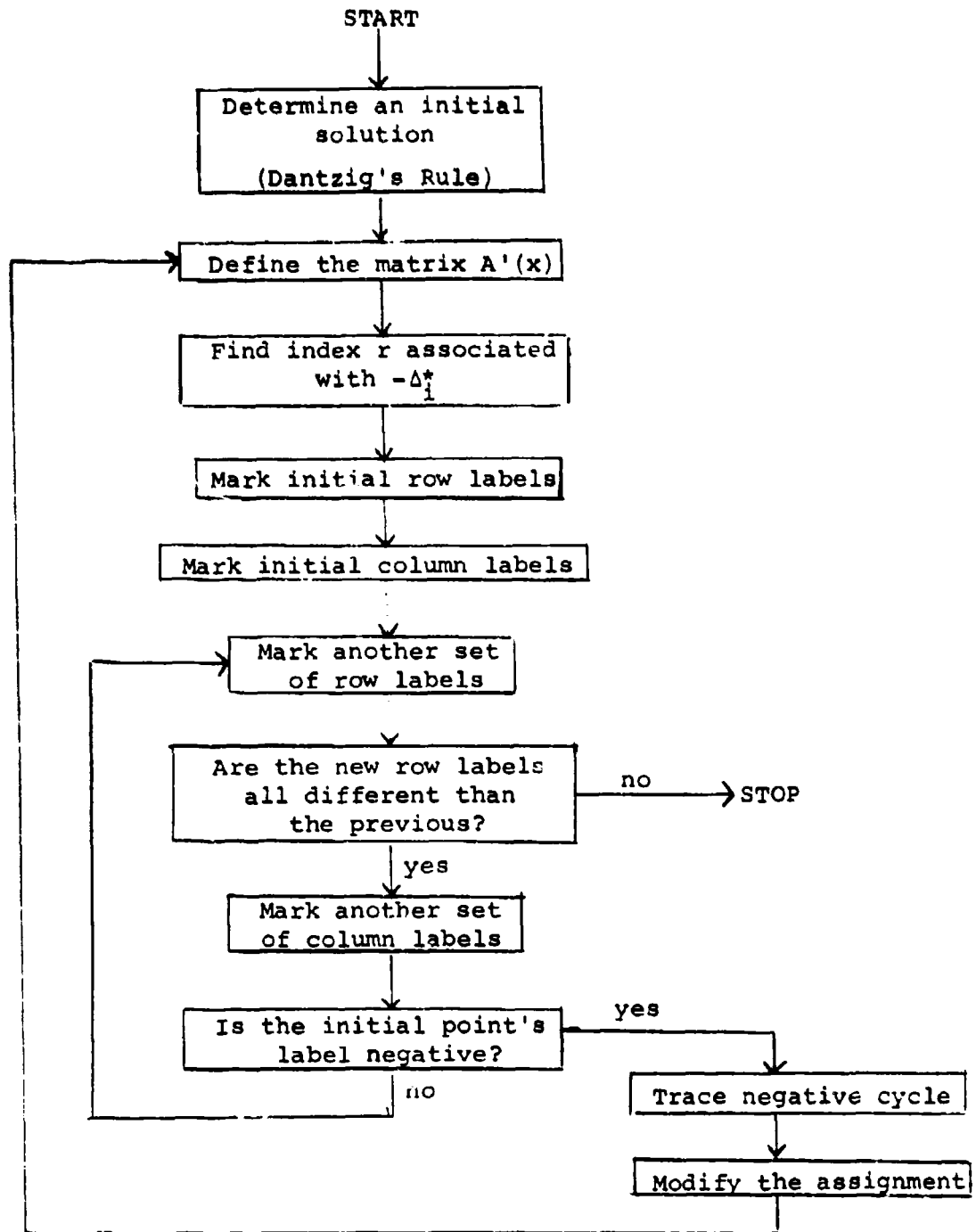
4c. If $d_{i_1 i_2} < 0$ put $d_{i_1 i_2} = \infty$,

otherwise $d_{i_1 i_2} = -d_{i_1 i_2}$ and return to 4a.

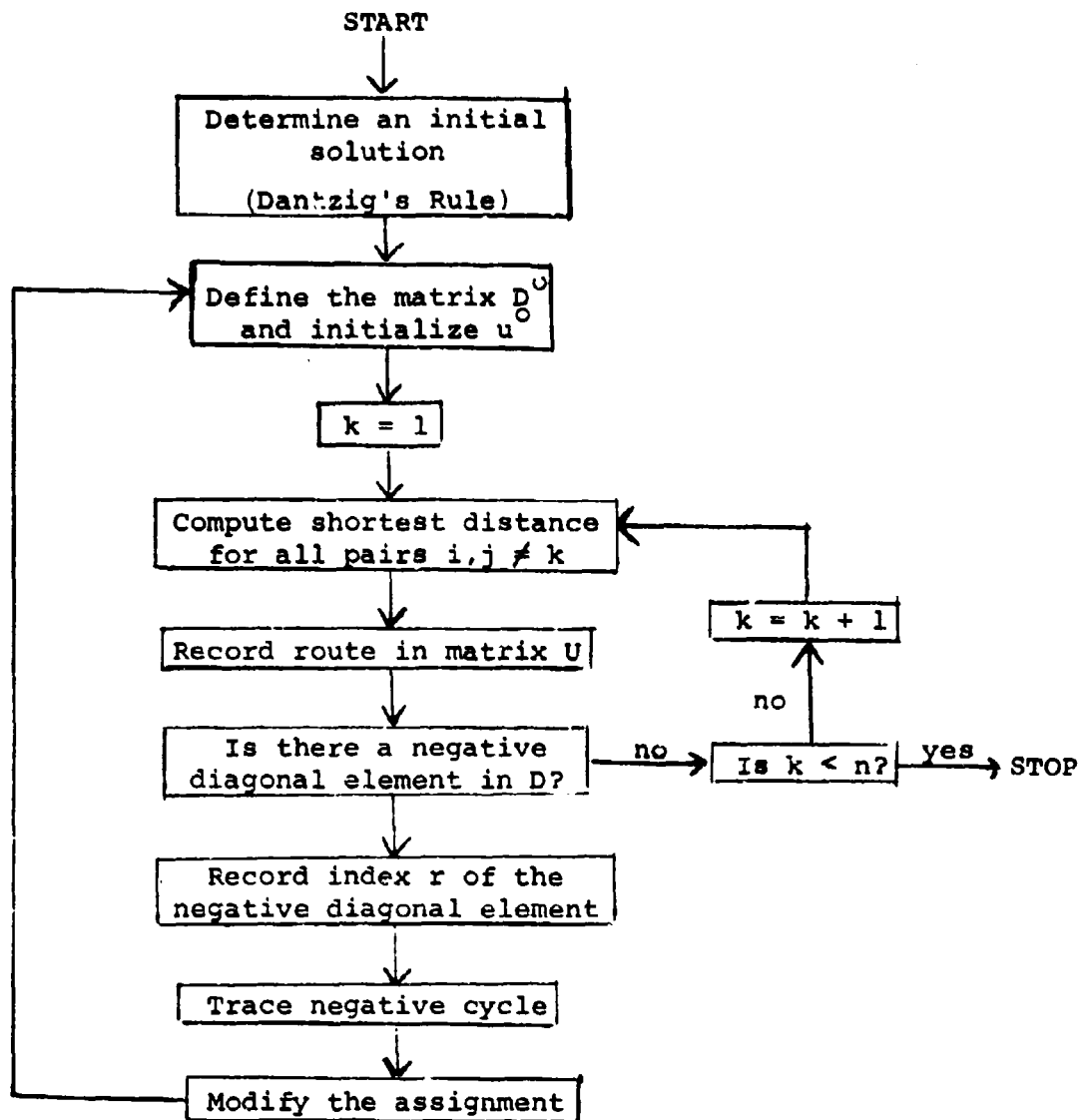
4d. If $d_{i_2 i_1} < 0$ put $d_{i_2 i_1} = \infty$,

otherwise $d_{i_2 i_1} = -d_{i_1 i_2}$ and return to 4a.

When this procedure terminates for $i = 1, \dots, k$ return to step 2.



FLOW CHART FOR NEGATIVE CYCLE ALGORITHM
WITH FORD & FULKERSON SHORTEST ROUTE MEHTOD



FLOW CHART FOR NEGATIVE CYCLE ALGORITHM
 BASED ON THE FLOYD-MURCHLAND "ALL SHORTEST ROUTES" METHOD

REFERENCES

1. Balinski, M.L. and Gomory, R.E., "A Primal Method For the Assignment and Transportation Problems", Management Science, Vol. 10, No. 3 (1964) pp 578-593.
2. Floyd, R.W., "Algorithm 97: Shortest Path", Communications of the ACM, Vol. 5, No. 6 (1962) pp 345.
3. Ford, L.R., Jr. and Fulkerson, D.R., Flows in Networks, Princeton University Press, Princeton, N.J., 1952.
4. Klein, M., "A Primal Method for Minimal Cost Flows with Applications to the Assignment and Transportation Problems", Management Science, Vol. 14, No. 3 (1967) pp 205-220.
5. Kuhn, H.W., "The Hungarian Method for the Assignment Problem", Naval Research Logistics Quarterly, Vol. 12, No. 2 (1957) pp 32-38.
6. Murchland, J.D., "A New Method for Finding All Elementary Paths in a Complete Directed Graph", London School of Economics Report LSE - TNT - 92, October 1965.

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
Columbia University		Unclassified	
		2b. GROUP	
3. REPORT TITLE			
As experimental Evaluation of Some Methods of Solving the Assignment Problem			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
Technical Report			
5. AUTHOR(S) (First name, middle initial, last name)			
Michael Florian and Morton Klein			
6. REPORT DATE		7b. TOTAL NO. OF PAGES	7c. NO. OF REFS
Dec. 30 1968		21	6
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S)	
N000 14-67-A-0108		Operations Research Group	
9. PROJECT NO.		Columbia University, School of	
		Engineering and Applied Science	
		9d. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
10. DISTRIBUTION STATEMENT			
Releasable without limitations on dissemination			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
		Logistics and Mathematical	
		Statistics Branch, Office of Naval	
		Research, Washington, D.C.	
13. ABSTRACT			
Computational experiments were conducted with three methods for solving the assignment problem: Kuhn's Hungarian method, a primal method due to Balinski and Gomory, and a negative cycle method proposed by Klein. Kuhn's method is seen to be the best of the three.			

DD FORM 1473

Security Classification

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Assignment problem Linear programming Network Flows						

Security Classification