

AD 663370

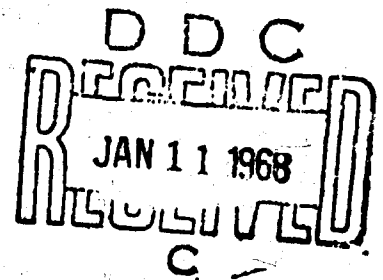
MEMORANDUM

RM-5472-PR

DECEMBER 1967

THE OUT-OF-KILTER ALGORITHM: A PRIMER

E. P. Durbin and D. M. Kroenke



PREPARED FOR:

UNITED STATES AIR FORCE PROJECT RAND

The **RAND** *Corporation*
SANTA MONICA • CALIFORNIA

Best Available Copy

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va 22151

MEMORANDUM

RM-5472-PR

DECEMBER 1967

THE OUT-OF-KILTER ALGORITHM: A PRIMER

E. P. Durbin and D. M. Kroenke

This research is supported by the United States Air Force under Project RAND - Contract No. F44620-67-C-0015 - monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. RAND Memoranda are subject to critical review procedures at the research department and corporate levels. Views and conclusions expressed herein are nevertheless the primary responsibility of the author, and should not be interpreted as representing the official opinion or policy of the United States Air Force or of The RAND Corporation.

DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

PREFACE

A surprising variety of seemingly unrelated problems can be described by the analog of flow in a network. Analysts were presented with both a very general algorithm and a very efficient code for finding solutions to a variety of network flow problems, with the distribution of R. Clasen's computer code for this algorithm (SHARE, RS OKFI), and the publication by D. R. Fulkerson, "An Out-of-Kilter Method for Minimal Cost Flow Problems," J. Soc. Indust. Appl. Math., Vol. 9, 1961, pp. 18-27. In discussing the use of two separate models with various Air Force organizations, one author has pointed out the computational advantages of network flow models over more general mathematical programming models, and has become aware that many operations analysts want to better understand the Out-of-Kilter algorithm; see E. P. Durbin, An Interdiction Model of Highway Transportation, The RAND Corporation, RM-4945-PR, May 1966; and E. P. Durbin and Olivia Wright, A Model for Estimating Military Personnel Rotation Base Requirements, The RAND Corporation, RM-5398-PR, October 1967. This present Memorandum describes the operation and capability of the Out-of-Kilter algorithm under the assumption that the reader is conversant with basic linear programming. It should be useful to analysts and planners interested in a versatile modeling concept and computational tool.

D. M. Kroenke is a cadet at the United States Air Force Academy, and coauthored this Memorandum while on temporary assignment to The RAND Corporation.

SUMMARY

The analog of steady state flow in a network of nodes and arcs may describe a variety of processes. Familiar examples are transportation systems and personnel assignment actions. Arcs generally have cost and capacity parameters, and a recurring problem is that of determining a minimum cost route between two points in a capacitated network. If a process can be modeled as a network, and the criterion for evaluating performance of the process can be related to the variables corresponding to flows in the network, then determining a minimum cost flow is equivalent to determining an optimal set of variables for the process.

Efficient and general methods of solving the minimum cost flow problem are therefore useful and important. Ford and Fulkerson's Out-of-Kilter Algorithm is an extremely efficient and general method for solving such problems. The algorithm operates by defining conditions which must be satisfied by an optimal "circulation" in a network -- roughly, a flow which satisfies capacity restrictions on all arcs and also satisfies stated conservation of flow conditions at all nodes. When such an optimal circulation is determined, all arcs are "in-kilter." At some point in the operation of the algorithm, if such a circulation does not yet exist, some arcs are "out-of-kilter" -- hence the name of the algorithm. The algorithm arbitrarily selects an out-of-kilter arc, and tries to rearrange flows to bring that arc into kilter while not forcing any other arc farther out-of-kilter. If the out-of-kilter arc can be brought into kilter, the algorithm selects another out-of-kilter arc and repeats the process. Since there are only a finite number of arcs, repetition of this procedure eventually results in an optimal solution. If any arc cannot be brought into kilter, the problem cannot be solved.

The Out-of-Kilter algorithm also solves the special network problems of finding maximum flow between two nodes in a costless, capacitated network, and finding the shortest route between two points in a network.

ACKNOWLEDGEMENTS

We are indebted to D. R. Fulkerson for his discussion and assistance, and suggest that those who desire additional information read the complete and elegant book, L. R. Ford, Jr., and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey, 1962.

CONTENTS

PREFACE.....	iii
SUMMARY	v
ACKNOWLEDGEMENTS	vii
Section	
I. NETWORKS.....	1
II. THE OUT-OF-KILTER ALGORITHM.....	4
Introduction.....	4
Operation of the Algorithm.....	8
Non-Breakthrough.....	13
Termination and Optimality.....	17
III. USES AND SPECIAL CASES.....	19

I. NETWORKS

Steady state flows in networks may represent many physical and nonphysical systems. The complex of freeways shown in Fig. 1, linking New York, Chicago, St. Louis, Denver, Los Angeles, Houston, Seattle, and Washington, D. C., is a network. The vehicles moving over that network may be considered homogeneous units of flow. Alternatively, this network might describe a petroleum distribution system. Flows in networks can also represent communications between people in an organization, inventory and production smoothing processes over time and assignment of personnel to jobs.

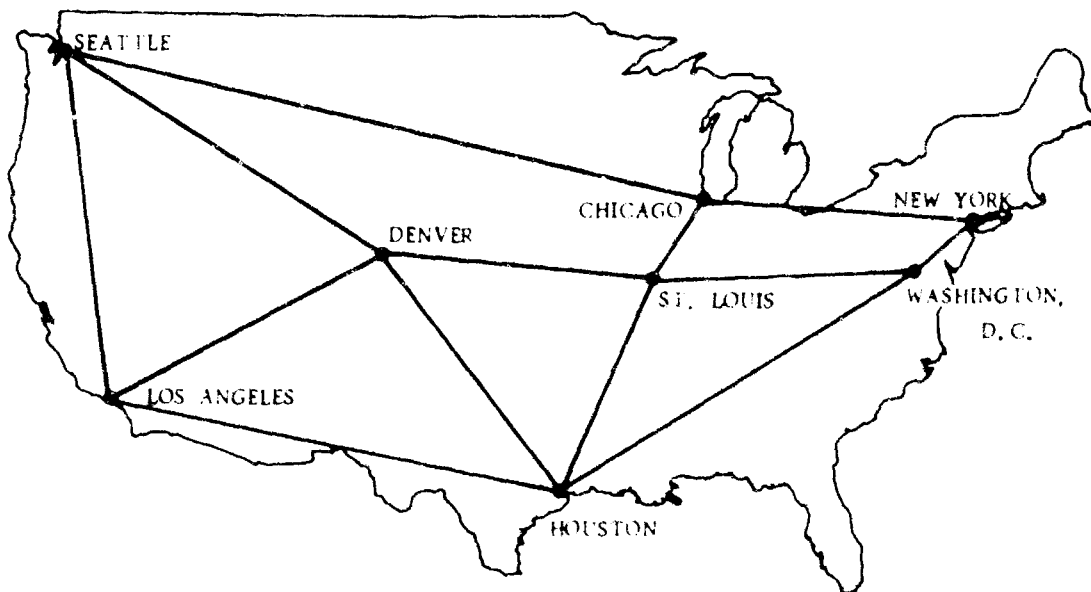


Fig. 1 -- A Network Linking Cities

The common elements of these situations, and hence the abstract definition of a network, are a collection of points called "nodes," (the cities in Fig. 1) and a collection of "arcs" which connect these nodes (the highways or pipelines in Fig. 1). We denote the nodes by single lower case letters; for example, node i , and identify the arcs by naming the nodes they connect, arc (i, j) . Some homogeneous commodity (vehicles, petroleum) can flow over the arcs, and we denote by x_{ij} the amount of commodity flowing on arc (i, j) from node i to node j . If $x_{ij} < 0$ then the commodity flows from j to i .

In most network problems, arcs have cost and capacity characteristics. Generally, some cost is incurred to move a unit from node i to node j , and we denote these unit movement costs by c_{ij} . This may be dollars per unit pumped in the petroleum distribution network. We also frequently find that flow is limited by upper bounds or capacities on the arcs. For example, only a limited number of vehicles per hour can move through the Lincoln Tunnel, and a limited number of barrels per day can move through the pipeline from Houston to St. Louis. We denote these maximum arc capacities u_{ij} . There may also be a requirement for a minimum amount of flow along any arc. We denote this by l_{ij} . Imposing this condition allows us to construct networks with controlled flows that describe particular problems, or that may describe actual minimum demand levels at points in a commodity flow network.

To summarize, a network is characterized by nodes, i ; arcs between the nodes (i, j) ; flow across the arcs, x_{ij} ; unit costs of flow across the arcs, c_{ij} ; upper bounds on flow across the arcs u_{ij} ; and lower bounds on flow across the arcs, l_{ij} . These characteristics can completely characterize steady state flow in a network.

In a problem with no costs, we allow $c_{ij} = 0$. In a problem with no lower bounds but flow only in one direction, we allow $l_{ij} = 0$. If there are no upper bounds, we allow $u_{ij} = +\infty$. We further assume that all costs, flows, and bounds are integers. This is not an overly strong assumption since if all numbers are rational, clearing fractions will yield integers. For computational purposes restriction to rationals suffices. The assumption of integral-valued parameters is used to demonstrate convergence of the Out-of-Kilter algorithm.

A general network problem is finding a minimum cost circulation in a network with arc capacities. In Sec. II we explain Fulkerson's "Out-of-Kilter" algorithm developed to solve this problem. The problem requires that we find flows, x_{ij} , that minimize total cost

$$(1) \quad \sum_{ij} c_{ij} x_{ij}, \quad \text{for all } i \text{ and } j.$$

while at the same time satisfy the constraining conditions

$$(2) \quad l_{ij} \leq x_{ij} \leq u_{ij} \quad \text{for all } i \text{ and } j,$$

and show that in a circulation, what goes into a node must come out of the node. This is represented by

$$(3) \quad \sum_j x_{ji} - \sum_j x_{ij} = 0 \quad \text{for all } i.$$

While there are several ways of solving this problem, and it may be conceptually viewed as a linear programming problem, the Out-of-Kilter algorithm is both the most general of the specialized algorithms and is far more efficient here than a standard linear programming algorithm would be. We discuss some capabilities of the Out-of-Kilter algorithm in Sec. III, and note here only two reasons for emphasizing its utility: (1) all problems solvable by the more specialized network algorithms (maximum flow through a capacitated network, shortest route through a network, etc.) are solvable by the Out-of-Kilter algorithm, and an understanding of it allows construction of the more efficient specialized algorithms; (2) many problems are simply too large to be handled computationally by linear programming algorithms on current computers. Problem formulation as a network permits efficient computational analysis by the Out-of-Kilter algorithm.

II. THE OUT-OF-KILTER ALGORITHM

INTRODUCTION

A set of flows, x_{ij} , that satisfies flow constraints (2) and conservation of flow equations (3) is called feasible. In attempting to find a minimal cost feasible circulation, the Out-of-Kilter algorithm (OKA) operates with both arc costs, c_{ij} , and "node prices," π_i , which arise from duality considerations. We assume the reader has no knowledge of duality theory, and attempt to make the existence of the node prices plausible and their nature clear by means of an economic explanation.* We have previously described the minimal cost circulation problem from the point of view of a producer or distributor. Consider the petroleum distribution network of Fig. 1. Assume some regulatory agency, say the "Federal Petroleum Distribution Commission," has set this distributor a variety of minimum and maximum flow levels on various route segments. The minimum flow levels may reflect consumer demands at points in the network, whereas the maximum flow levels may reflect physical route capacities. The distributor knows the unit transit charges on each route segment and must decide the route structure -- arcs and the amounts moving on each arc -- to meet the induced flow requirements on all arcs at minimum transportation cost.

The minimum flow levels on various route segments have presumably been established at the insistence of consumers. Therefore let π_i denote the price a consumer at node i must pay for a unit of petroleum. The π_i will be related to the amount demanded at some nodes, since as the amount demanded increases, the overall difficulty in supplying it will increase.

The Commission recognizes that the price of petroleum to the consumer in each city of the network should be related to the distributor's overall cost in establishing the entire route structure, and not related to just the transportation costs connecting the source and node i .

* For a further discussion of duality see L. R. Ford and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey, 1962, pp. 26-30 and pp. 113-115. Also see W. J. Baumol, Economic Theory and Operations Analysis, Prentice-Hall, New York, 1965, Chap. 6.

Neither the distributor nor the Commission are initially sure what these node prices, π_i , ought to be, but the Commission -- hopefully concerned with consumer welfare as well as producer profit -- intends to see that consumer costs are no greater than necessary to induce the distributor to establish the route.

The OKA may be pictured as a sequence of systematic decisions the distributor makes with the Distribution Commission watching carefully. At each step the distributor asks, "Which route shall be used to minimize total system costs, taking account not only of transportation costs, but also of the commodity prices at the various cities?" If there is no profitable route at some step, the distributor tells the Commission, "You must now allow increased prices at some cities in order to make it financially possible for me to continue constructing a route." The Commission answers, "Okay, but let's find that route which requires the minimum price increase."

Based on this fanciful exchange, we associate with each node, i , a variable, π_i , that can be considered the price of a unit of the flow commodity at the node. The Distribution Commission defines a net arc cost, $\bar{c}_{ij}$, as

$$(4) \quad \bar{c}_{ij} = c_{ij} + \pi_i - \pi_j.$$

The new cost, $\bar{c}_{ij}$, represents the total cost to the system -- consumer and distributor -- of transporting one unit of flow from node i to node j . This definition compares the cost of retaining a unit at node i with the cost of moving it to node j . In moving a unit of flow from i to j , the commodity price at i , π_i , is foregone, and an actual transportation cost, c_{ij} , is incurred. If the sum of these costs is greater than the commodity price at j , π_j , then it does not pay to ship a unit from i to j . The $\bar{c}_{ij}$ will be positive. On the other hand, if a unit at j costs more than at i plus the transportation cost, $\bar{c}_{ij}$ will be negative -- the system benefits from the move -- and shipment from i to j is profitable. If the value at j , π_j , is balanced exactly by the value at i plus the transportation cost, $(\pi_i + c_{ij})$, then $\bar{c}_{ij} = 0$, and we are indifferent to an additional unit flowing from i to j .

Limitations on permissible flow levels (2) together with possible levels of total system cost (4) yield the following conditions that will be satisfied by an optimal solution to the minimal cost circulation problem (1) - (3).

$$(5) \quad \text{If } \bar{c}_{ij} < 0 \text{ then } x_{ij} = u_{ij}.$$

$$(6) \quad \text{If } \bar{c}_{ij} = 0 \text{ then } l_{ij} \leq x_{ij} \leq u_{ij}.$$

$$(7) \quad \text{If } \bar{c}_{ij} > 0 \text{ then } x_{ij} = l_{ij}.$$

Equation (5) states that when net arc cost is negative (when it is profitable to send the commodity from i to j), flow on the arc ought to be as large as possible. Equation (6) states that when net arc cost is zero, we are indifferent to flow level so long as it meets constraints. Equation (7) states that when net arc cost is positive (a loss is incurred by sending the commodity from i to j), flow on the arc ought to be at the minimum level possible. The OKA is designed to construct a flow meeting these conditions.

Any arc that meets the optimality conditions (5), (6), (7) is defined as "in-kilter." Arcs that do not satisfy these conditions are "out-of-kilter."

Out-of-kilter arcs can be grouped into two categories:

(a) Those that are feasible but not optimal. They have flow which satisfies (2), but prices and flow do not satisfy (5), (6), (7).

(b) Those that are infeasible. Flow is below or above the upper bound. Therefore (2) is not satisfied.

Arcs that are feasible but not optimal must fit one of the following states or conditions.*

* It is not necessary to consider as a special case $\bar{c}_{ij} < 0$ and $x_{ij} < l_{ij}$ or $\bar{c}_{ij} > 0$ and $x_{ij} > u_{ij}$. These cases, though infeasible, are included in conditions I and II. If the algorithm terminates with all arcs in kilter, any originally infeasible arc in state I or II will not only be feasible but optimal.

Condition I: $\bar{c}_{ij} < 0$ and $x_{ij} < u_{ij}$.

Condition II: $\bar{c}_{ij} > 0$ and $x_{ij} > l_{ij}$.

Infeasible arcs fit one of these conditions:

Condition III: $\bar{c}_{ij} > 0$ and $x_{ij} < l_{ij}$.

Condition IV: $\bar{c}_{ij} = 0$ and $x_{ij} < l_{ij}$.

Condition V: $\bar{c}_{ij} = 0$ and $x_{ij} > u_{ij}$.

Condition VI: $\bar{c}_{ij} < 0$ and $x_{ij} > u_{ij}$.

A "kilter-number," as defined below, is associated with each arc.

Arc	
Condition	Kilter Number
I.....	$\bar{c}_{ij} [x_{ij} - u_{ij}]$
II.....	$\bar{c}_{ij} [x_{ij} - l_{ij}]$
III, IV.....	$[l_{ij} - x_{ij}]$
V, VI.....	$[x_{ij} - u_{ij}]$

Notice that in all cases, the kilter number is positive. For the feasible states, I and II, the kilter number is a measure of non-optimality. The kilter number for states III - VI indicates the degree to which an arc is infeasible. In-kilter arcs meet conditions (5), (6), or (7), and therefore have a kilter number of zero.

The OKA operates by arbitrarily selecting an out-of-kilter arc and rearranging flows in an attempt to reduce the kilter number of that arc to zero. During this process the kilter numbers of other arcs do not increase, and may in fact decrease. The algorithm terminates with an optimal solution when kilter numbers of all arcs are zero. Optimality of such a solution is verified by recalling the definition of $\bar{c}_{ij}$, and noting that a kilter number of zero implies Eqs. (5), (6), and (7) hold.

It is slightly more complicated to show that the algorithm terminates. The algorithm selects one out-of-kilter arc at a time and attempts to bring that arc into kilter by rearranging flows without forcing another arc out of kilter. If this process can be completed, the algorithm seeks another out-of-kilter arc. If it cannot be completed, the problem is infeasible. If it takes only a finite number of steps to put an arc into kilter or to determine infeasibility, and there are only a finite number of arcs in the network, the algorithm must terminate. We will later indicate that these conditions are satisfied.

OPERATION OF THE ALGORITHM

Once the minimal cost circulation problem (2) has been formulated, the OKA can be started with any set of node prices, π_i , and any circulation which satisfies the conservation of flow equations (3). This circulation and set of node prices can initially be zero. Arbitrarily select an out-of-kilter arc joining two nodes, say s and t . The fact that the arc (s, t) is out of kilter indicates that it is either profitable or necessary (or possibly both) to ship an additional unit from t to s or from s to t . In either event, flow change is always desired for an out-of-kilter arc. In order to change flow on the arc and yet keep flow in the network balanced (3), another path through the network from t to s must be found along which flow values, x_{ij} , can be changed. In constructing this path, two pieces of information (a "label") must be retained at every node. The first component of the label at a given node j indicates the previous node in the path, and whether present flow moves from i to j , denoted $(i+)$, or from j to i , $(i-)$. The second component of the label is the amount b_j which flow on arc (i, j) is to be changed, $\epsilon(j)$. A complete label at node j is

$$[i\pm, \epsilon(j)].$$

The search proceeds from node t through the network seeking a path back to s . "Labels" are assigned to various nodes so that when a path is found connecting t and s , the labels indicate the direction and magni-

node of necessary flow change along the path. Of course not every node will receive a label in this process. Circulation from Houston to St. Louis can be completed by arcs from St. Louis to Washington and Washington to Houston. Seattle and Los Angeles, for example, would not receive labels.

If a path connecting t and s is not found, the algorithm determines new node prices, π_i , in such a manner that (a) another node will be labeled in the partial (t, s) path, (b) one less arc will be considered for inclusion in the (t, s) path, or (c) if no arcs remain and the path is incomplete, the problem is deemed infeasible.

We next explain some of the labeling rules that are used in attempting to find a path in the network from t to s .^{*} If a path is found we term this "breakthrough." Assume that an arbitrary arc (s, t) is in state I, with $\bar{c}_{st} < 0$ and $x_{st} < u_{st}$. Since $\bar{c}_{st} < 0$ it is profitable to increase the flow, x_{st} . We begin at t by assigning node t the label $[s+, \epsilon(t)]$ where

$$(8) \quad \epsilon(t) = u_{st} - x_{st}.$$

This indicates that flow on this arc should be increased by $\epsilon(t)$ in order to be equal to the upper bound, and hence be optimal. See Fig. 2.

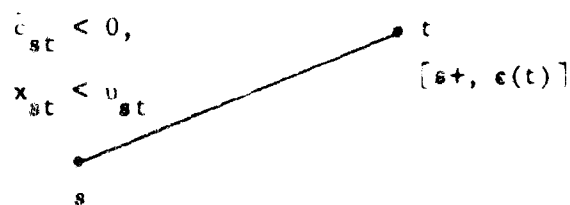


Fig. 2 -- Initial Flow on Out-of-Kilter Arc (s, t)

^{*}We describe cases I and IV. The rules are only slightly different in the other cases, and the method remains basically the same. The important special network flow problem which case IV covers is the "capacitated transportation problem." Section III describes operation of the algorithm in this problem.

The algorithm next seeks an unlabeled node adjacent to t , say j . Assume conditions on the arc (t, j) are either

$$(9a) \quad \bar{c}_{tj} > 0 \text{ and } x_{tj} < l_{tj}$$

or

$$(9b) \quad \bar{c}_{tj} \leq 0 \text{ and } x_{tj} < u_{tj}.$$

In either case we desire to increase flow enough from t to j to correct infeasibility or nonoptimality on the arc. We therefore assign node j the label $[t+, e(j)]$. If condition (9a) holds, we have

$$(10a) \quad e(j) = \min[e(t), l_{tj} - x_{tj}],$$

while for condition (9b) we have

$$(10b) \quad e(j) = \min[e(t), u_{tj} - x_{tj}].$$

The $e(j)$ state the maximum amount by which we can raise flow on the arc (t, j) . See Fig. 3. We would like to increase x_{tj} by $(l_{tj} - x_{tj})$ in case 9a. And we would like to increase x_{tj} by $(u_{tj} - x_{tj})$ in case 9b. But at the same time the previous label at t states that we want to increase flow on (s, t) at most by the amount $e(t)$. Thus we compare $e(t)$ with the new value computed by (10a) or (10b) and select $e(j)$ as the minimum of the two values -- the maximum amount by which we can increase flow along both arcs without violating the feasible bounds or optimality conditions on any arc. This guarantees that flow changes will not increase the kilter number of any arc.

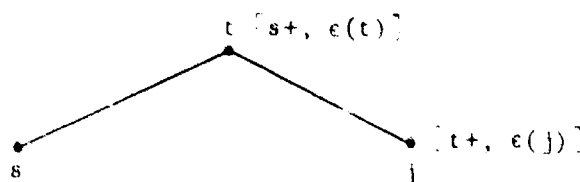


Fig. 3 -- Carrying Forward a Label

Assume that we next encounter an arc connecting nodes s and j . We have then completed a path from s to t to j and on returning to s along the path we wish to increase flow. Thus far the flows encountered have been in the appropriate direction-- s to t and t to j . Suppose, however, that flow on the arc (s, j) is moving from s to j , and that arc (s, j) is in one of the following conditions:

$$(11a) \quad \bar{c}_{sj} \geq 0 \text{ and } x_{sj} > l_{sj}$$

$$(11b) \quad \bar{c}_{sj} < 0 \text{ and } x_{sj} > u_{sj}.$$

We then assign the label $[j-, \epsilon(s)]$ where

$$(12a) \quad \epsilon(s) = \min[\epsilon(j), x_{sj} - l_{sj}] \text{ if (11a) holds}$$

or

$$(12b) \quad \epsilon(s) = \min[\epsilon(j), x_{sj} - u_{sj}] \text{ if (11b) holds.}$$

In either case we want to increase flow on the directed path $s, (s, t); t, (t, j); j, (j, s)$ by the amount $\epsilon(s)$. We therefore decrease flow, x_{sj} , by the amount $\epsilon(s)$. This is equivalent to increasing flow in the desired direction (see Fig. 4) by $\epsilon(s)$.^{*} Recall that $\epsilon(s)$ is the maximum amount by which flow on all arcs can be increased without increasing any kilter number.

We have now broken through -- found a path from t to s through the network, and have determined $\epsilon(s)$, the maximum amount by which flow

^{*} Consider the petroleum distribution network of Fig. 1. Assume present flow is 6 units from Houston to Denver and 4 units from Houston to Washington, D.C. If we desire to increase flow from Denver to Houston by 2 units, it is not reasonable to send 6 units from Houston to Denver and at the same time to send 2 units of the same commodity from Denver to Houston. It would be more profitable to reduce the flow from Houston to Denver to 4 units. This "increases" the flow from Denver to Houston by reducing the amount Houston must ship to Denver, and is an example of increasing forward flow by decreasing reverse flow.

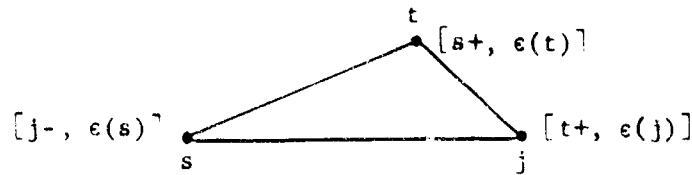


Fig. 4 -- Labeling an Arc with Reverse Flow

along this path can be increased without violating optimality or feasibility conditions. The algorithm now changes flow according to the label at each node. At t we increase flow from s to t by $\epsilon(s)$. At j we increase flow from t to j by $\epsilon(s)$. And at s we decrease flow from s to j by $\epsilon(s)$. Flow from s to t has now increased by $\epsilon(s)$. The amount of flow augmentation, $\epsilon(s)$, was chosen by considering for each arc the amount by which flow could change without violating feasibility or optimality conditions. Therefore, on at least one arc, the one yielding the minimum of the $\epsilon(i)$, flow has actually been set equal to an upper or lower bound. Reviewing the definition of the kilter numbers for conditions I - VI, and the flow augmenting rules, note that flow augmentation makes flow on at least one arc closer to a bound, and therefore at least one kilter number has decreased.*

We summarize operation of the algorithm to this point. After starting with any circulation and node prices, we arbitrarily select an out-of-kilter arc (s, t) and label it t . We next examine all nodes adjacent to t , determine if flow on arcs to these nodes could be appropriately modified to help bring (s, t) into kilter, and label the nodes using rules 9 through 12. For each node so labeled we attempt to repeat this labeling process until a path returning to s is found. Flow along that path is then augmented by the maximum feasible amount, $\epsilon(s)$, which is the minimum of feasible amounts along all arcs on the path.

*The kilter number will not necessarily become zero on any new arc. If the minimum $\epsilon(i)$ occurs on an arc originally satisfying Eq. (6), with original kilter number zero, the kilter number of that arc will remain zero, and kilter numbers on other arcs will decrease, but not necessarily to zero. Notice also that since flow changes by an integral value and since all costs and prices are integers, kilter numbers decrease by integral amounts.

All labels are now erased, and arc (s, t) is examined again. If that arc is in kilter, we seek another out-of-kilter arc. If (s, t) remains out of kilter, the search and label process is repeated. The same path from t to s will not be chosen again because on at least one arc of that path, flow is at an upper or lower bound. If all arcs are in kilter, the optimality and feasibility conditions (5), (6), and (7) are satisfied, and the algorithm has terminated with an optimal solution to (1), (2), and (3).

Labeling rules for out-of-kilter arcs (i, j) in states other than I or IV are similar. When flow on arc (i, j) should be decreased (states II, V, VI), the labeling process starts at i , attempting to move along a path through the network to j . An out-of-kilter arc in state III is treated as one in state I or IV, except that $\epsilon(j) = l_{ij} - x_{ij}$, the amount of flow augmentation needed to remove the infeasibility.

NON-BREAKTHROUGH

Given an out-of-kilter arc (s, t) on which we want to increase flow, our ability to find a path through the network from t to s has depended on the availability of arcs (i, j) in one of two conditions. On an arc (i, j) either we both desire to increase flow x_{ij} and are able to, or we both want to decrease flow x_{ji} and are able to. If, in attempting to find a path for a given out-of-kilter arc (s, t) , we have labeled all nodes we can by rules (9a), (9b), (11a), (11b), and have not returned to node s , we have a "non-breakthrough."

There are two possible difficulties. Either (1) the net arc costs $\bar{c}_{ij}$ provide insufficient incentive for a distributor to move the commodity through the network, or (2) no augmentation is possible because flows x_{ij} are equal to upper bounds and flows x_{ji} are equal to lower bounds. If case 1 occurs, we will systematically change prices π_i in a way to be described, recompute $\bar{c}_{ij}$, and again try to break through. If case 2 occurs we must declare the problem infeasible. This infeasibility can be seen since we reached this point by attempting to bring an out-of-kilter arc (s, t) into kilter. The fact that the arc (s, t) as a positive kilter number indicates that either optimality conditions or feasibility conditions are not satisfied on it. If the

problem is feasible, and only the optimality conditions fail to be satisfied, the repricing of the π_i will eventually bring $\bar{c}_{st}$ to zero, and hence bring the original arc into kilter. The problem is infeasible, however, if flow x_{st} is outside the bounds (l_{st}, u_{st}) , and no path through the network exists to change that flow and bring x_{st} within bounds.

To look more carefully at this non-breakthrough situation, consider the various arcs that can be used for a breakthrough from t back to s . Arcs connecting labeled nodes to labeled nodes are already in the path from t to our present position. They cannot be of further use. Arcs which connect unlabeled nodes to unlabeled nodes are of no use, since we have no means of extending our path to these arcs. Therefore, we need only consider arcs that connect labeled to unlabeled nodes.*

This iteration started with an out-of-kilter arc (s, t) in condition I or IV, and we have been attempting to move flow forward from t through the network to s , and have stopped at a labeled node, i . If arc (i, j) is to be potentially useful, its flow must be less than capacity ($x_{ij} < u_{ij}$ if $x_{ij} > 0$), or if flow is moving in the direction opposite to our path, that flow must be greater than the lower bound ($x_{ji} > l_{ji}$). In the former case we can increase forward flow, and in the latter case we can decrease reverse flow. In either case a useful arc is one that increases net forward flow. For forward flow arcs (i, j) with $x_{ij} < u_{ij}$, and for reverse flow arcs with $x_{ji} > l_{ji}$, conditions (9a), (9b), (11a), (11b) cannot be satisfied. Otherwise, by rules 10 and 12 the nodes j would have received labels. In other words, the current cost structure is incorrect.

Arcs with forward flow below capacity ($x_{ij} < u_{ij}$) have positive net arc cost ($\bar{c}_{ij} > 0$). Otherwise, by rule (9b), both nodes would be labeled. Such arcs must also have $x_{ij} > l_{ij}$, or by (9a) both nodes would be labeled. Arcs with reverse flow above the lower bound

* If there are no such arcs and the lower bound on (s, t) is positive, the problem is infeasible. For s flows into t , and t flows to other network nodes, but no network node ever flows back to s .

$(x_{ji} > l_{ji})$ must have negative net arc cost ($\bar{c}_{ji} < 0$) because of rule (11a), and must have $x_{ji} \leq u_{ji}$ because of rule (11b). Thus, the arc conditions that must exist on arcs connecting labeled nodes, i , to unlabeled nodes, j , in case of a non-breakthrough are

$$(13) \quad \bar{c}_{ij} > 0 \quad \text{and} \quad l_{ij} \leq x_{ij} \leq u_{ij},$$

$$(14) \quad \bar{c}_{ji} < 0, \quad l_{ji} \leq x_{ji} \leq u_{ji}.$$

Equations (13) and (14) represent arcs that may yet be useful in breaking through to the starting out-of-kilter arc (s, t) . The arcs are useful since it is possible that flow can still be modified on them.

Recall that Eq. 4 defines $\bar{c}_{ij}$:

$$\bar{c}_{ij} = c_{ij} + \pi_i - \pi_j.$$

The potentially useful forward arcs have not been used since $\bar{c}_{ij} > 0$. This states that the value given up at node i , π_i , plus the cost of moving an additional unit of flow to j , c_{ij} , is greater than π_j , the value of having the unit at j . It does not pay the distributor to send it. For a reverse arc satisfying (14) it does not pay to reduce back flow.

At this point, however, path construction has halted, and there will be no path through the network from t to s unless it becomes possible to use a previously unused arc satisfying (13) or (14), and to label at least one previously unlabeled node. We will accomplish this by raising the price of the commodity, π_j , at all unlabeled nodes, j . This will make it more profitable to ship toward the unlabeled node on forward arcs (increase x_{ij}) or to reduce flow away from the unlabeled node on reverse arcs (reduce x_{ji}). This repricing action corresponds to a grudging approval by the hypothetical Federal Petroleum Distribution Commission for an increase in commodity prices that consumers will have to pay, with a warning from the Commission: No more than necessary to get flow moving again! We obviously could get flow moving again on at least one unused arc satisfying (13) or (14) by adding a very large amount to all unlabeled node prices, π_j . The Commission will not allow this. We need the minimum sufficient increase in unlabeled node prices.

We can get flow moving again on at least one appropriate forward arc by finding the minimum $\bar{c}_{ij}$ on all the forward arcs, say b_1 , and adding this to π_j at all unlabeled nodes. This would result in a new system cost on each forward arc:

$$\bar{c}'_{ij} = c_{ij} + \pi_i - (\pi_j + b_1).$$

And on the arc that yielded the minimum,

$$\bar{c}'_{ij} = (c_{ij} + \pi_i - \pi_j) - \bar{c}_{ij} = 0.$$

By labeling rule (9b) this last arc can be used, since at system cost of zero we are indifferent to changes in flow. Therefore, node j can be labeled. This action has essentially cost the consumer and the system b_1 per unit of flow arriving at j , which we have had to add to the price, π_j , at the unlabeled nodes.

It might have proven cheaper to reduce back flow on one of the arcs (j, i) . This could have been accomplished by finding b_2 , the magnitude of the smallest negative cost ($\min|\bar{c}_{ji}|$), and adding b_2 to all node prices, π_j , at unlabeled nodes. On each relevant arc with reverse flow this would have resulted in a new system cost given by

$$(15) \quad \bar{c}'_{ji} = c_{ji} + \pi_j - (\pi_i + b_2).$$

On the arc yielding b_2 , the minimum magnitude,

$$(16) \quad \bar{c}'_{ji} = c_{ji} + \pi_j - \pi_i - \bar{c}_{ji} = 0.$$

By labeling rule (11a), this last arc can be used to augment flow and node j can be labeled. The pricing procedure has essentially cost the consumer and the system b_2 per unit. If b_2 is less than b_1 we should use the second procedure. In general, then, the algorithm finds b_1 and b_2 from the arcs defined by (13) and (14), selects b (the smaller of b_1 and b_2), and increases node prices at all unlabeled nodes by an amount b .

Repricing actions have allowed the distributor to add another arc to the (t, s) path at some cost to the purchasers but at the minimum cost to those purchasers. Following the repricing procedure, the algorithm checks the original out-of-kilter arc (s, t) . If it is still out of kilter, the search for a path back to node t is repeated. We summarize the algorithm to this point. Arbitrarily select an out-of-kilter arc (s, t) , and by means of the labeling procedure attempt to find a path through the network from t to s . If such a path is found, appropriately modify the flow along each arc of the path by the amount determined in the labeling process. If a path to s is not immediately found, consider all arcs that are potentially useful in extending the path from t . Modify the node prices on these arcs so that one of these is used to extend the path and continue the labeling procedure. If there are no arcs connecting labeled to unlabeled nodes that satisfy the conditions (13) or (14), then no path is possible and the problem is infeasible. Once this particular arc is brought into kilter, the algorithm continues until all arcs are in kilter (which signifies an optimal solution to the original problem) or until it is impossible to construct a feasible circulation.

TERMINATION AND OPTIMALITY

The OKA concentrates on one out-of-kilter arc at a time. Each time a breakthrough to the out-of-kilter arc occurs, the labeling rules insure improvement in kilter numbers on every arc in the path, including the out-of-kilter arc. We have mentioned the assumption that all values are integral. Kilter numbers therefore improve by integral amounts at each breakthrough. Thus, only a finite number of breakthroughs are possible before the arc is in kilter. There are only a finite number of arcs. If the algorithm is successful in bringing every arc into kilter, the solution is optimal by (5), (6), and (7).

The remaining situations are the non-breakthroughs. When a non-breakthrough occurs and potentially useful arcs exist, price changes are made and at least one of the potentially useful arcs is used to extend the path. Since there are only a finite number of arcs, we must

eventually either break through or discover there are no potentially useful arcs left. In the former case, we decrease the kilter number of the original out-of-kilter arc; in the latter, we declare the problem infeasible.

III. USES AND SPECIAL CASES

While networks can be used to model a variety of actual problems, ingenuity is often called for in formulating the network to describe the problem. If the network can be properly formulated, however, it is far more efficient to solve a minimal cost circulation problem than the equivalent linear programming problem. Furthermore, we frequently need to examine the behavior of a solution as the parameters vary. As Fulkerson points out,^{*} the Out-of-Kilter algorithm is designed to start with any circulation and any set of node prices. Therefore, a previously derived optimal solution can be used to begin a new problem with resultant savings in computation time.

We have described the OKA for cases I and IV. The important special network flow problem case IV covers is the "capacitated transportation problem," or the shipment of a fixed level of flow, v (possibly maximum feasible flow), through a network from node s to node t at minimum cost. In this special case we can start with all flows equal to zero, and all node prices equal to zero. On a "return" arc (t, s) we set the lower bound, l , equal to the upper bound, u , equal to the desired flow value, v . Then the arc (t, s) is out of kilter since the initial flow, $x_{ts} = 0$, is less than the required flow, $l_{ts} = v$. The cost, c_{ts} , may arbitrarily be set to $-\infty$, causing arc (t, s) to be initially out of kilter in state I, or c_{ts} may be set to 0, causing arc (t, s) to be initially out of kilter in state IV.

We mention two other important special cases of the general minimal cost circulation problem. The first is determining maximum flow in a capacitated network, and the second is finding the shortest route through a network in which costs on arcs are times or distances. While there are specialized algorithms for each case, the Out-of-Kilter algorithm handles either, and in the process indicates how to construct a more specialized algorithm.

In order to use the OKA to find the maximum level of flow in a network from node s to node t , we take the following actions:

^{*}Ford and Fulkerson, p. 162.

- (1) add an arc connecting t to s .
- (2) set $\ell_{ij} = 0$ for all i and j .
- (3) set $c_{ij} = 0$ for all i and j , except c_{ts} .
- (4) set $c_{ts} = -M$ where M is very large.
- (5) set $x_{ij} = 0$ and $\pi_i = 0$ for all i and j .
- (6) set $u_{ts} = M$ where M is very large.

On all arcs except (t, s) , costs are zero, node prices are zero, and flow is feasible and zero. Arc (t, s) has a negatively large cost and flow below its upper bound, and is therefore out of kilter, while all other arcs are in kilter. By the labeling rules, all arcs other than (t, s) will stay in kilter.

We first label node s , $[t+, u_{ts} - x_{ts}]$, and search for nodes which can be labeled from s . There are initially several, since $\bar{c}_{sj} = 0$ and $x_{sj} \leq u_{sj}$ on all arcs (s, j) . We continue the labeling process until there is a breakthrough.* We then augment flow on the breakthrough path by the OKA rules.

Since we assigned the capacity on the return arc (t, s) a very large value, the arc will still be out of kilter.** Therefore, repeat the labeling process and further breakthroughs will continue to augment flow from s to t . Finally, there will be no path from s to t on which every arc has flow less than capacity. At this point the algorithm will consider all arcs connecting labeled to unlabeled nodes,*** select the minimum cost, $\bar{c}_{ij}$, on these arcs, add this value to all node prices, π_j , on the unlabeled nodes, and attempt to label them. Generally, a label would be assigned to at least the forward node of the arc on which the minimum $\bar{c}_{ij}$ had been found, for the new $\bar{c}_{ij}$ on that arc would be zero. In this case, however, flow equals capacity on all arcs connecting labeled and unlabeled nodes. Consequently, even with the "repricing," no new labels will be assigned. The algorithm will once again consider all arcs connecting labeled to unlabeled nodes, but there will be at

* If a solution is possible with positive flow ($u_{ij} > 0$ on some path), there will be a breakthrough, since initially flow on all arcs is zero.

** $\bar{c}_{ts} < 0$, since c_{ts} is negatively very large, and $x_{ts} < u_{ts}$.

*** These arcs will now have $x_{ij} = u_{ij}$.

least one less arc, since by our repricing, at least one $\bar{c}_{ij}$ now equals zero. The OKA will thus continue to look at arcs connecting labeled to unlabeled nodes which have $\bar{c}_{ij} > 0$, it will increase π_j , and continue to remove arcs from further consideration. Eventually the only arc left for consideration with a labeled and unlabeled node will be (t, s) .^{*} At this point, π_t is increased by a sufficiently large number so that $\bar{c}_{st} = 0$.^{**} Arc (s, t) is now in kilter. We therefore have maximum flow through the network.

Notice that once flow augmentation stops, we reach a maximum flow. The remainder of the process simply computes the proper prices. We could have terminated the algorithm as soon as there was no longer a flow augmenting path from s to t .

The problem of finding the shortest route through an uncapacitated network is identical to that of finding the minimum cost path from s to t through a network on which each arc has $u_{ij} = 1$, and on which a return arc (t, s) has $u_{ts} = 0$, $u_{ts} = 1$, and $c_{ts} = -M$, where M is very large. The OKA will send one unit of flow from s to t at minimum cost. Assume $c_{ij} > 0$ for each arc. Start the problem with $x_{ij} = 0$ on all arcs, and $\pi_j = 0$ at all nodes. Since $\bar{c}_{ij} > 0$ on all arcs except arc (t, s) , the only out-of-kilter arc is (t, s) . By the labeling rules, node s is labeled $[t+, \epsilon(s)]$, where $\epsilon(s) = u_{ts} - x_{ts} = 1$. The algorithm then attempts to label nodes from s , but since $\bar{c}_{ij} > 0$, it cannot. It therefore revises all π_j at all unlabeled nodes by adding to them the minimum $\bar{c}_{ij}$. Cost on at least one arc (s, j) then becomes $\bar{c}_{sj} = 0$, and on this arc flow is less than capacity. The forward node, j , of this arc receives the label $[s+, \epsilon(j)]$, where $\epsilon(j) = \min[\epsilon(s), u_{sj} - x_{sj}]$. Since u_{sj} is 1, $\epsilon(j) = \epsilon(s)$, which was 1. This indicates that the second label at all labeled nodes will be one, and the problem is only one of breaking through on some path. The algorithm continues forward from all labeled nodes, finding all $\bar{c}_{ij} > 0$, and being forced to increase π_j by the appropriate minimum $\bar{c}_{ij}$. This then allows labeling one more arc forward. Eventually, node t is reached, and flow along the breakthrough path is

^{*} It will be the last one because $c_{ts} = -M$, ($c_{st} = M$), where M is very large. Hence, $\bar{c}_{st} = \pi_s + M - \pi_t > 0$.

^{**} This is the feasible but nonoptimal condition described on page 14.

increased by one unit. All arcs along the path are in kilter since $\bar{c}_{ij} = 0$ for each arc on the path, and $x_{ij} = u_{ij}$. To find the shortest route (or minimum cost path) we need only trace back through the network from t to s , using the first component of the labels. The length or cost of this path is given by π_t .

DOCUMENT CONTROL DATA

1 ORIGINATING ACTIVITY THE RAND CORPORATION		2a REPORT SECURITY CLASSIFICATION UNCLAS//SI//NF	
		2b GROUP	
3 REPORT TITLE THE OUT-OF-KILTER ALGORITHM: A SURVEY			
4 AUTHOR(S) (Last name, first name, initial) Fulkerson, D. R. and D. M. Johnson			
5 REPORT DATE January 1967	6a TOTAL NO. OF PAGES 31	6b NO. OF REFS ---	
7 CONTRACT or GRANT NO. DA-19-07-00-0045	8. ORIGINATOR'S REPORT NO. R-640		
9. AVAILABILITY/LIMITATION NOTICES UNCLAS		9b SPONSORING AGENCY United States Air Force Rand	
10 ABSTRACT A description of the operation and capability of Fulkerson's Out-of-Kilter algorithm (OKA), an extremely efficient and general method for solving minimum cost flow problems. The algorithm operates by defining conditions that must be satisfied by an optimal "circulation" in a network: 1) flow that satisfies capacity restrictions on all arcs and also satisfies stated conservation of flow conditions at all nodes. When such an optimal circulation is determined, all arcs are "in-kilter." At some point in the operation of the algorithm, if such a circulation does not yet exist, then some arcs are "out-of-kilter." The OKA arbitrarily selects an out-of-kilter arc and tries to rearrange flows to bring that arc into kilter without forcing any other arc further out-of-kilter. If the out-of-kilter arc can be brought into kilter, the algorithm selects another out-of-kilter arc and repeats the process. Since there are only a finite number of arcs, repetition of this process eventually leads to an optimal solution. If any arc cannot be brought into kilter, the problem cannot be solved.		11 KEY WORDS Linear programming Models Mathematics Network theory Logistics Transportation	