Chinese Business Review, Apr.-June 2023, Vol. 22, No. 2, 86-94
doi: 10.17265/1537-1506/2023.02.005

Practical Application of Out-of-Kilter Algorithm

Irma Ibrišimović
University of Tuzla, Tuzla, Bosnia and Herzegovina
Zoran Jasak, Aldijana Omerović, Elvir Čajić
FINra, Tuzla, Bosnia and Herzegovina

The algorithm under this name, together with the variants, is a method that solves the problems of optimal flow and costs. Examples of such problems are planning and procurement, scheduling by contractors, distribution and supply systems, transport on the road or rail network, electricity transmission, computer and telecommunications networks, pipe transmission systems (water, oil, …), and the like. The main goal of any business organization is to increase profits and satisfy its customers. Because business is an integral part of our environment, their goals will be limited by certain environmental factors and economic conditions. The out-of-kilter algorithm is used to solve a complex allocation problem involving interactive and conflicting personal choices subject to interactive resource constraints. The paper presents an example of successful use of this algorithm and proposes an extension to the areas of corporate and social planning. Customer demand, warehousing, and factory capacity were used as input for the model. First, we propose a linear programming approach to determine the optimal distribution pattern to reduce overall distribution costs. The proposed model of linear programming is solved by the standard simplex algorithm and the Excel-solver program. It is noticed that the proposed model of linear programming is suitable for finding the optimal distribution pattern and total minimum costs.

Keywords: out-of-kilter algorithm, linear programming, minimization, minimum cost flow, simplex algorithm

Everywhere we look in real life we will notice different networks. Computer networks, highways, telecommunications networks, water delivery systems, and many others are familiar to all of us. In each of these problem settings, we often want to send some product from one point to another, usually as efficiently as possible, that is, by the shortest path or some minimum cost flow pattern. Network optimization has always been a major problem in operations research, computer science, applied mathematics, and many other areas of engineering, science, and management. Many applications in these fields not only occur "naturally" on some transparent physical network, but also in apparently non-networked situations. The minimum cost flow problem is the most fundamental of all network flow problems. Minimum cost flow problems occur in almost all industries, including communications, agriculture, manufacturing, transportation, healthcare, retail, education, energy, and medicine.

Irma Ibrišimović, Master of Applied Mathematics, professor, Faculty of Science, University of Tuzla, Tuzla, Bosnia and Herzegovina.
Zoran Jasak, PhD, professor, College of Finance and Accounting, FINra, Tuzla, Bosnia and Herzegovina.
Aldijana Omerović, Master of Information Technologies, professor, College of Finance and Accounting, FINra, Tuzla, Bosnia and Herzegovina.
Elvir Čajić, Master of Mathematics and Physics, professor, College of Finance and Accounting, FINra, Tuzla, Bosnia and Herzegovina.
Correspondence concerning this article should be addressed to Irma Ibrišimović, Amalije Lebeničnik LA1, 75000 Tuzla, Bosnia and Herzegovina.

The problem is easy to state: We want to determine the lowest cost of delivering goods through the network in order to satisfy the demands at certain nodes from the available stocks at other nodes. This model has a number of well-known and lesser-known applications: product distribution from plant to warehouse or from warehouse to customers; routing of vehicles through the street network; etc. Many researchers have investigated the minimum cost flow problem. Their solution methods can be divided into graphic techniques and linear programming methods. Ford and Fulkerson (1972) developed a classic and still frequently used method for solving this problem, the primal-dual method whose algorithm is based on the theory of linear programming.

Jewel, Busaker, and Grown independently developed a sequential shortest path algorithm. These researchers showed how to solve the minimum cost flow as a series of shortest path problems with arbitrary branch lengths (Jewel, 1958). If tip potentials are used in the calculations, Edmonds and Karp (1972) noted that it is possible to implement these algorithms so that shortest path problems have negative branch lengths. Minty and Fulkerson developed an algorithm known as the out-of-kilter algorithm. Aashtiani and Magnanti described an efficient implementation of this algorithm. The cycle cancellation algorithm is attributed to Klein. Bertsekas and Tseng developed the relaxation algorithm and conducted extensive computational research on it. Grigoriadis, Kennington, and Wang described an efficient implementation of the relaxation algorithm and network simplex (Wang, Tang, & Zhao, 2013).

A large number of real-world applications can be modeled using multi-objective minimum cost network flows. Damian and Garrett (1991) in their paper entitled "Minimum Cost Flow Problem and Simple Network Solution Method" in the Irish distribution network have Dublin and Belfast as supply nodes, while Cork, Galway, Limerick, and Waterford were demand nodes. The spanning tree technique was used to find the optimal solution. Shigeno, Iwata, and McCormick (2000) discussed various algorithms. They explained how efficient Edmond-Karp and Push-relabel algorithms are suitable for maximum throughput problems.

The paper provides an example of successful use of the out-of-kilter algorithm and solves the problem of minimum cost flow when supply, demand, and distribution cost per unit quantity are known. The case study is carried out on the example of the XY company. The problem was set as a linear programming problem and was solved using the Simplex algorithm and Excel-solver.

Out-of-Kilter Algorithm

This algorithm was designed by Fulkerson and is an efficient means of solving the "minimum cost flow" problem in the network. In a directed network where:

(i) there is no exogenous flow, i.e., the total flow along the branches leading to each node is equal to the total flow along the branches leading from the node;

(ii) there is an upper limit u_{ij} and a lower limit l_{ij} on the flow along each branch ij in the network;

(iii) and there is a cost c_{ij} associated with each unit of flow along branch ij;

the out-of-kilter algorithm finds a circulation in the network in which the flow is conserved at each node, the flow along each branch lies between the specified upper and lower bounds, and the total cost of flowing through the network is minimized. By building appropriate networks, the algorithm can be used to solve problems of shortest route, maximum throughput, transportation, allocation, and transshipment. Here, a more complex assignment problem is considered according to the ordered choices of interacting individuals, with two interaction constraints. It should be noted that if the lower and upper limits are set on all branches as integers, then the minimum cost flow will also be an integer on each branch. This is, of course, of particular importance in the scheduling problem where we want to allocate time to only one job. Since the only non-zero labels are the rankings on the student/project ranking branches, the algorithm, seeking a minimum-cost solution, will allocate as many high-ranking (small-sized) branches as possible.

To explain how to use the out of kilter algorithm, we will use the following example of a problem:

At one of the universities, every student is required to work on a “project”. Since they will be spending several months on this project and since it is desirable that they show considerable “interest” in it, they are asked to “rank” the available projects in descending order. For this reason, one goal of the problem is to assign, i.e., give everyone as high a rank as possible to the extent that interactions with other people’s rankings allow it. There is an upper and possibly a lower limit to the number of students who can work on each project; that’s why we have “design capacity” limits. It is also desirable that the supervisory workload of the staff be shared on some managerial basis. They represent a second set of constraints, and since some projects have shared or multiple supervisor involvement, they interact with the first set of constraints.

Expressing the Problem in Terms of a Network Project

The network representation of the flow of the problem is shown in Figure 1, where three numbers next to each branch indicate the lower and upper limits of the allowed flow and the cost that should be associated with the flow of one unit in that branch. Since in this case there are 31 students requiring projects, a flow of 31 must be achieved between Node 1 and Node 104. The purpose of the branch (104, 1) is to ensure that exactly 31 units flow from Node 1 to Node 104. The branches (1, 2) to (1, 32) ensure that each student gets exactly one project. Flow along these branches need not be associated with costs.

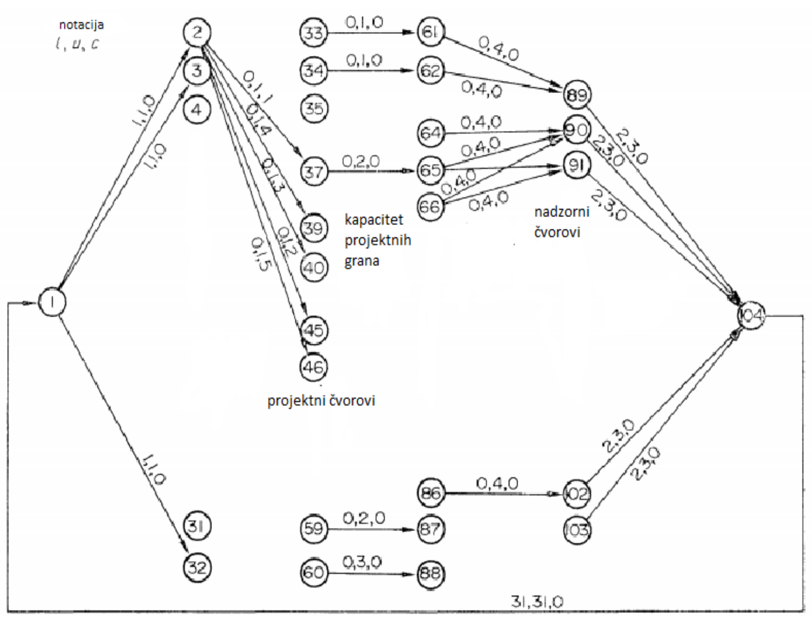

Figure 1. Network diagram of the problem flow.

The diagram (Figure 1) is simplified for clarity, but the ranking branches of Student 2’s project are drawn as an illustration. For his first five rankings, he chose, in descending order, Projects 37, 45, 40, 39, and 46. The price associated with the corresponding branch is the ranking number. A second set of project Nodes, 61-88, are drawn and arrows join the project node to form “project capacity” branches. The upper limit on each of these arcs is the maximum number of students that the leader specifies for that project. Nodes 61-66 connect projects to monitoring nodes 89-90. An illustration of shared responsibility is shown for Projects 65 and 66 between Supervisory Nodes 90 and 91. In fact, there was one case where six managers had to share responsibility for a large project. The upper bounds are chosen so as not to limit the flow at this stage. The final "collection" node is 104. If the equal load criterion were chosen, the fact that there were 31 students and 15 supervisors would mean that no staff member should supervise fewer than two nor more than three students. Branches are marked with lower and upper bounds according to this criterion. This, then, the branch (104, 1) completes the network.

Change of Branch Numbers

Since only the first five rankings of each student are considered, it may happen that the interactions of their rankings with the applied constraints combine in such a way that they do not allow a feasible solution. Initially, the branch (104, 1) could be labeled 0, 31, -99,999, and the branches (1, 2) to (1, 32) could be labeled 0, 1, 0. Since the algorithm searches for the solution of minimal costs, this would make projects available to more students. If all 31 are not set, a sixth rank of each student could be added, or the limit could be relaxed slightly. The lower and upper bounds of the constraints can be chosen to allow a feasible solution, but "weighted" through their cost label with some negative number, to ensure that they are applied if possible. In this case, the burden of "equality" on managers was not desirable as it includes some managers with heavy administrative duties. Furthermore, the supervisors are involved in different number of projects and have different other duties. To solve this problem, each leader branch is duplicated; the first is marked as 1, 2, -40, the second 0, 0, 0 or 0, 1, 0, or 0, 2, 0 depending on the circumstances of the given leader. Actual values would have to be determined by the people concerned. In particular, management should give some thought to the relative size of the weights applied to the various goal branches and constraints. A different ranking method may also be requested. Since the difference in desirability between first and second, and second and third rank students is not constant, he could be given 100 points to allocate (applied as negative costs) among as many projects as he wanted. They would warn him that if he puts them all on one project branch, he will link all other projects to the branch at zero cost. So if he did not get his first and only choice, he would probably get the project that everyone else "left behind".

Findings

1. Expressing the problem in this format can help management see the consequences of the weighting that can be given to different objectives and constraints in conflict; and thus could help improve the quantification of those weightings.

2. An "unbiased" result is obtained and once the numbers are fixed, the lower level of management can take over the processing of the problem.

3. The high efficiency of the algorithm in computing time would allow an iterative approach if infeasible solutions are found that could return the loop to the above point (1) where the management can be asked to reconsider its goals or open the imposed constraints.

4. We have solved a complex assignment problem involving personnel selection, interacting with possibly contradictory choices of other people, subject to two mutual constraints.

Model Formation

A Real Network Flow Model

Real networks can be modeled as a directed graph $G\ (V, E)$, where V is the set of vertices ($|V| = n$) and E is the set of directed branches. Each branch $(i, j) \in X$ is associated with a set of non-negative values, which are called network branch attributes. Examples of network branch attributes are port cost, distance, time, brand, etc. One of the important parts in network operation is routing. Routing can be thought of as sending goods from one network vertex to another. The routing task consists of finding a path $P(s, t)$ suitable for a given application that has end-to-end constraints. In the next chapter, we will present the minimum cost flow problem.

The Minimum Cost Flow Problem

Reference: Consider the network $G = (V, E)$ with $|V| = n$ and let $b \in R_n$. Here, b_i denotes the amount of flow entering or leaving the network at Node $i \in V$. If $b_i > 0$, we say that the source pushing b_i is a unit flow. Furthermore, let c_{ij} denote the cost associated with one unit of flow on branches $(i, j) \in E$, and l_{ij} and u_{ij} respectively denote the lower and upper limits of the flow over this branch. The minimum cost flow problem is to find the flow x_{ij} that conserves flow on each branch, satisfies the upper and lower bounds, and minimizes the total cost. The single-commodity linear network flow problem, linear minimum cost, is defined as:

$$min\{c_{ij} * x_{ij}\}$$

under the conditions:

$$\sum_{(i,j)\in E} x_{ij} + \sum_{(j,i)\in E} x_{ji} = b_i$$

for each $i \in V$—we call flow conservation constraints and capacity constraints:

$$l_{ij} \leq x_{ij} \leq u_{ij}$$

We have assumed that all the data are integers and we want to find the optimal integer value solution. Without loss of generality, we can further assume that all branch capacities are finite, all branch costs are negative, and that the problem has a feasible solution. It also implies that:

$$\sum_{i\in V} b_i = 0$$

A Case Study

For the case study, we will take a company "XY" that has two factories. In addition, it has four warehouses with storage areas. The company sells its products to six customers, K1, K2, K3, K4, K5, and K6. Customers can ship either from the warehouse or from the factory (see Figure 2).

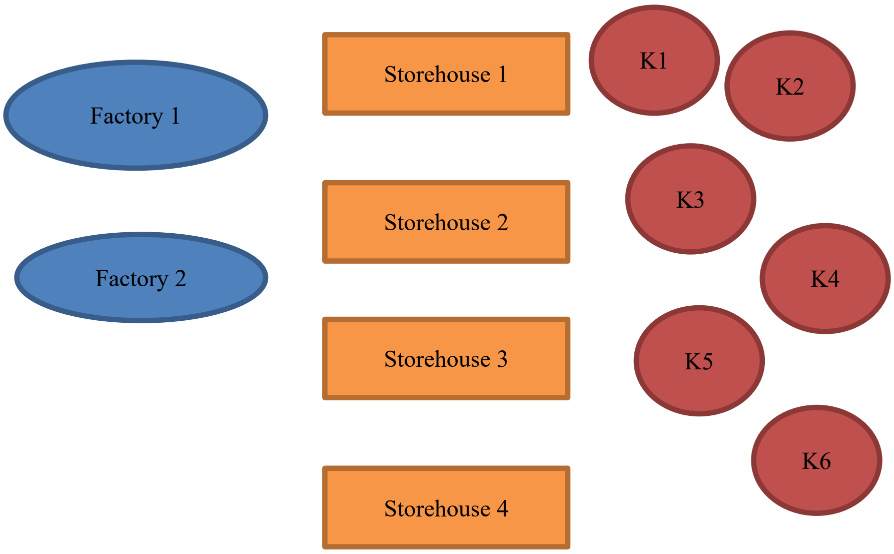

Figure 2. Company "XY".

Distribution costs are known; are given in Table 1 (A dash indicates the impossibility of certain suppliers for certain warehouses or customers).

Table 1

Distribution Costs

Delivery according to	Supplier					
	F1	F2	S1	S2	S3	S4
S1	0.5	-				
S2	0.5	0.3				
S3	1.0	0.5				
S4	0.2	0.2				
K1	1.0	2.0	-	1.0	-	-
K2	-	-	1.5	0.5	1.5	-
K3	1.5	-	0.5	0.5	2.0	0.2
K4	2.0	-	1.5	1.0	-	1.50
K5	-	-	-	0.5	0.5	0.5
K6	1.0	-	1.0	-	1.5	1.5

Each factory has a lower (annual) capacity that cannot be exceeded:

Factory 1	150,000
Factory 2	200,000

Each warehouse has a maximum (annual) flow that cannot be exceeded:

Warehouse 1	70,000
Warehouse 2	50,000
Warehouse 3	100,000
Warehouse 4	40,000

Each customer has the following monthly requirements that must be fulfilled:

K1	50,000
K2	10,000
K3	40,000
K4	35,000
K5	60,000
K6	20,000

Application

Factories, warehouses, and customers will be numbered below:

Factory: 1, 2

Warehouse: 1, 2, 3, 4

Customers: 1, 2, 3, 4, 5, 6

Decision Variables

A description of the decision variables used to create the model is given below:

x_{ij} = quantity sent from factory i to warehouse j, i = 1, 2, j = 1, 2, 3, 4;

y_{ij} = quantity sent from factory i to customer k, i = 1, 2, k = 1, 2, 3, 4, 5, 6;

z_{ij} = quantity sent from warehouse j to customers k, j = 1, 2, 3, 4; k = 1, 2, 3, 4, 5, 6.

There are 44 such variables.

The Objective Function

The decision marker must determine the lowest price for the delivery of goods through the network. Therefore, the objective function is formulated as follows:

$$\sum_{i=1}^{2}\sum_{j=1}^{4} c_{ij}\, x_{ij} + \sum_{i=1}^{2}\sum_{j=1}^{4} d_{ij}\, y_{ij} + \sum_{i=1}^{2}\sum_{j=1}^{4} e_{ij}\, z_{ij}$$

where the coefficients c_{ij}, d_{ik}, ie_{jk} are tabulated.

Limitations

1. Factory capacities:

$$\sum_{j=1}^{2} x_{\mathrm{ij}} + \sum_{j=1}^{6} y_{\mathrm{ik}} \leq \text{capacity} \;\; \text{where} \;\; \mathrm{i} = 1{,}2$$

2. Quantity in warehouses:

$$\sum_{k=1}^{2} x_{\mathrm{ij}} \leq \text{capacity} \;\; \text{where} \;\; \mathrm{j} = 1{,}2{,}3{,}4$$

3. Quantity out of stock:

$$\sum_{k=1}^{6} z_{\mathrm{jk}} = \sum_{i=1}^{2} x_{\mathrm{ij}} \;\; \text{where} \;\; \mathrm{j} = 1{,}2{,}3{,}4$$

4. Customer requirements:

$$\sum_{i=1}^{2} y_{\mathrm{ik}} + \sum_{j=1}^{4} z_{\mathrm{jk}} = \text{request} \;\; \text{where} \;\; \mathrm{k} = 1{,}2{,}3{,}4{,}5{,}6$$

The solutions of the minimum cost flow model using the standard simplex algorithm and Excel are presented in Table 2.

Table 2

Display of Value Changes in the Algorithm

From		According to		Flow	Price
F1	Factory 1	S1	Warehouse 1	0	0.5
F1	Factory 1	S2	Warehouse 2	0	0.5
F1	Factory 1	S3	Warehouse 3	0	1
F1	Factory 1	S4	Warehouse 4	40,000	0.2
F1	Factory 1	1	Customer 1	50,000	1
F1	Factory 1	2	Customer 2	0	1,000
F1	Factory 1	3	Customer 3	0	1.5
F1	Factory 1	4	Customer 4	0	2
F1	Factory 1	5	Customer 5	0	1,000

Table 2 to be continued

F1	Factory 1	6	Customer 6	200,000	1
F2	Factory 2	S1	Warehouse 1	0	1,000
F2	Factory 2	S2	Warehouse 2	50,000	0.3
F2	Factory 2	S3	Warehouse 3	55,000	0.5
F2	Factory 2	S4	Warehouse 4	0	0.2
F2	Factory 2	1	Customer 1	0	2
F2	Factory 2	2	Customer 2	0	1,000
F2	Factory 2	3	Customer 3	0	1,000
F2	Factory 2	4	Customer 4	0	1,000
F2	Factory 2	5	Customer 5	0	1,000
F2	Factory 2	6	Customer 6	0	1,000
S1	Warehouse 1	1	Customer 1	0	1,000
S1	Warehouse 1	2	Customer 2	0	1.5
S1	Warehouse 1	3	Customer 3	0	0.5
S1	Warehouse 1	4	Customer 4	0	1.5
S1	Warehouse 1	5	Customer 5	0	1,000
S1	Warehouse 1	6	Customer 6	0	1
S2	Warehouse 2	1	Customer 1	0	1
S2	Warehouse 2	2	Customer 2	10,000	0.5
S2	Warehouse 2	3	Customer 3	0	0.5
S2	Warehouse 2	4	Customer 4	35,000	1
S2	Warehouse 2	5	Customer 5	5,000	0.5
S2	Warehouse 2	6	Customer 6	0	1,000
S3	Warehouse 3	1	Customer 1	0	1,000
S3	Warehouse 3	2	Customer 2	0	1.5
S3	Warehouse 3	3	Customer 3	0	2
S3	Warehouse 3	4	Customer 4	0	1,000
S3	Warehouse 3	5	Customer 5	55,000	0.5
S3	Warehouse 3	6	Customer 6	0	1.5
S4	Warehouse 3	1	Customer 1	0	1,000
S4	Warehouse 4	2	Customer 2	0	1,000
S4	Warehouse 4	3	Customer 3	40,000	0.2
S4	Warehouse 4	4	Customer 4	0	1.5
S4	Warehouse 4	5	Customer 5	0	0.5
S4	Warehouse 4	6	Customer 6	0	1.5
Total cost			198.5		

Note. * A value of 1,000 indicates the unavailability of certain suppliers for certain warehouses or customers.

The data in Table 2 show that the model found this distribution pattern to cost 198.5 per month. Storage capacities are exhausted in Warehouse 2 and Warehouse 3. Storage capacities can be changed within certain limits. For unused Warehouses 1 and 4, changing the capacity with these constraints has no effect on the optimal solution. Finally, we conclude that the proposed model is suitable for minimizing the distribution costs of the research area.

References

Damian, J. K., & Garrett, M. O. (1991). The minimum cost flow problem and the network simplex solution method (Masters' dissertation, National University of Ireland, University College Dublin, 1991).

Edmonds, J., & Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *JACM, 19*(2), 248-264.

Ford, L. R., & Fulkerson, D. R. (1972). *Flows in networks*. Princeton: Princeton University Press.

Fulkerson, D. R. (1961). An out-of-kilter method for minimal cost flow. *Journal of the Society for Industrial and Applied Mathematics, 9,* 18-27.

Jewell, W. S. (1958). Optimal flow through networks. *Technical Report 8*. Operations Research Center, Massachusetts Institute of Technology.

Klein, M. (1967). A primal method for minimal cost flows with applications to the assignment and transportation problems. *Manage Sci, 14*(3), 205-220.

Shigeno, M., Iwata, S., & McCormick, S. T. (2000). Relaxed most negative cycle and most positive cut canceling algorithms for minimum cost flow. *Mathematics of Operations Research, 25*(1), 76-104.

Wang, G. L., Tang, W. S., & Zhao, R. Q. (2013). An uncertain price discrimination model in labor market. *Soft Computing, 17*(4), 579-585.