BALKANJM 01 (2013) 117-130

Contents lists available at BALKANJM

BALKAN JOURNAL OF MATHEMATICS

journal homepage: www.balkanjm.com

An Application of Network Simplex Method for Minimum Cost Flow Problems

Ergun EROGLU[*a]

[a]*Istanbul University Business Administration Faculty*

ARTICLE INFO

Article history:
Received 26 July 2013
Accepted 5 September 2013
Available online 31 October 2013

Keywords:
Network Optimization
Minimum Cost Network Flow
Network Simplex
Network Dual Feasible Solution

ABSTRACT

Networks are more convenient for modeling because of their simple mathematical structure that can be easily represented with a graph. This simplicity takes an advantage with regard to algorithmic efficiency. In this paper, an implementation of network simplex algorithm is described for solving the minimum cost network flow problem which is one of the most fundamental and significant problems in the optimal design on a generalized network with the additional constraint. Network flow problem can be defined by a given set of nodes and arcs with known cost parameters for each arc and fixed external flow for each node. The optimization problem is to send flow from a set of supply nodes, through the arcs of a network, to a set of demand nodes, at minimum total cost subject to the arc capacity constraints. The simplex algorithm applied to the network flow programming problem. Network simplex method describes basic solutions for the network flow programming problem and provides procedures for computing the primal and dual solutions associated with a given basis to find the optimal solution.

1. Introduction

Operations research (OR) is the application of scientific methods, techniques and tools to problems involving the operations of a system so as to provide those in control of the operations with optimum solutions to the problem.” [1]. If one mentions about Operation research, the word “optimization” comes to remind.

Optimization is to find the best value of the variables that make optimal the objective function satisfying a set of constraints. It originated in the 1940s, when George Dantzig used

[*]Correspondingauthor: *E-mail*: eroglu@istanbul.edu.tr (E. Eroglu).

mathematical techniques for generating "programs" for military application [2]. Optimization technology allows researchers to search for optimal solutions to complex business, economics, computer science and engineering problems. Optimization problems are often classified as linear or nonlinear depending on whether the relationships in the problem are linear or nonlinear with respect to the variables [3].

Network optimization models have been most exciting developments in OR in recent years. They are widely used in optimization problems which have countless practical applications in various fields including commodity transportation, telecommunication systems, network design, resource planning, scheduling, railroad and highway traffic planning, electrical power distribution, project planning, facilities location, resource management, and financial planning and much more. The fundamental question in network optimization is how to efficiently transport some entity (commodity, product, electrical power, vehicles, water etc.) from one point to another in a network [4].

Network optimization is a special type of linear programming model. Some special types of network optimization models include: transportation problems, assignment problems, shortest path problems, minimum spanning tree problems, maximum flow problems, Chinese postman problem, knapsack problem and minimum cost flow problems [5].

Transportation problem was first studied by a Russian mathematician, L.V. Kantorovich, in a paper entitled Mathematical Methods of Organizing and Planning Production (1939) [6]. One of the most fundamental network flow problems is the Minimum Cost Flow Problem (MCF).

MCF Problem is to send flow from a set of supply nodes, through the arcs of a network, to a set of demand nodes, at minimum total cost, and without violating the lower and upper bounds on flows through the arcs. MCF often plays an important role in modeling operations management problems such as production and inventory planning, supply chain management, multi echelon inventory planning, capacity expansion, etc.

MCF Problem and the Network Simplex Method (NSM) were initially developed quite independently. The MCF has its origins in the formulation of the classic transportation type problem, while NSM, as its name suggests is deduced from the Simplex Method for solving Linear Programming Problems [7], [8].

Many network optimization models actually are special types of linear programming problems. Therefore we begin to study of network flow problems with a review of linear programming (LP) problems. Let the number of decision variables x_j's be N, and the number of constraints be M. Minimization of a LP Problem in primal and dual mode takes the fallowing generic form [2], [3]:

Primal Mode *Dual Mode*

$$\max\ Z = \sum_{j=1}^{N} c_j x_j \qquad \min\ G = \sum_{i=1}^{M} b_i y_i$$

$$\sum_{j=1}^{N} a_{ij} x_j \leq b_i \qquad \sum_{i=1}^{M} a_{ji} y_i \geq c_j$$

$$x_j \geq 0 \qquad y_i \geq 0$$

For each constraint in the primal we introduce a variable for the dual problem. For each variable in the primal, we introduce a constraint in the dual. Depending on whether the primal constraint is an equality or inequality constraint, the corresponding dual variable is either unrestricted in sign (URS) or restricted in some way, respectively. In addition, depending on whether a variable in the primal is unrestricted in sign or sign constrained, we have an equality or inequality constraint, respectively in the dual. There has been an example of an LP problem which is written in two different modes like primal and dual.

Primal	*Dual*
Equality Constraint	Free Variable
Inequality Constraint	Nonnegative Variable
Free Variable	Equality Constraint
Nonnegative Variable	Inequality Constraint

Primal Problem	*Dual Problem*
$\max\ Z = c_1 x_1 + c_2 x_2$	$\min\ G = y_1 b_1 + y_2 b_2 + y_3 b_3$
$s.t: a_{11}x_1 + a_{12}x_2 \geq b_1 / y_1$	$s.t: a_{11}y_1 + a_{21}y_2 + a_{31}y_3 \geq c_1$
$a_{21}x_1 + a_{22}x_2 \geq b_2 / y_2$	$a_{12}y_1 + a_{22}y_2 + a_{32}y_3 \geq c_2$
$a_{31}x_1 + a_{32}x_2 \leq b_3 / y_3$	$y_1 \geq 0$, y_2 is URS, $y_3 \leq 0$
$x_1, x_2 \geq 0$	

Primal: *2 Variables, 3 Inequalities* Dual: *3 Variables, 2 Inequalities*

In linear programming, the strong duality theorem states that, when solutions to both the primal and dual are equal to each other, then that is the optimal solution of the linear programming problem [3].

$$\sum_{j=1}^{N} c_j x_j = \sum_{i=1}^{M} b_i y_i$$

Linear programming has a wide range of applications and extensions and can be applied to a wide variety of real world problems, including network flow problems. One type of network flow problem is minimum cost flow problem. MCF problems define a special class of linear programs. The solution algorithm described in this paper is based on the primal simplex algorithm for linear programming. To determine optimality conditions it is necessary to provide both the primal and dual linear programming models for the network flow problem.

2. Minimum Cost Flow Problem

2.1 Definition and the Notation

The Minimum Cost Flow Problem is to minimize the total cost of flows along all arcs of a network, subject to conservation of flow at each node, and upper and lower bounds on the flow along each arc.

A network is a collection of points, called *vertices (nodes)*, and a collection of lines, called *edges (arcs)*, connecting these points. Network topology is only one part of the graph. A network can be visualized by drawing the *nodes* as circles and the *arcs* as lines between them.

For a directed network, the lines are arrows pointing in the appropriate directions. For a given network $G = (\mathcal{N}, \mathcal{A})$, which is defined by a set of nodes ($\mathcal{N}$), and a set of arcs ($\mathcal{A}$) connecting the nodes.

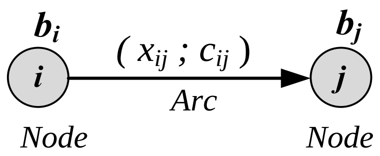


Figure 1. Representation of Node and Arc

There are three types of nodes in a minimum cost flow problem: supply node, demand node, and transshipment node. A supply node is defined as a node where the flow out of the node exceeds the flow into the node. Similarly, a demand node is where the flow into the node exceeds the flow out of the node. A transshipment node is where the flow into the node equals the flow out of the node. For example, a distribution network would include the sources of the goods being distributed (supply nodes), the customers (demand nodes) and intermediate storage facilities (transshipment nodes) [7].

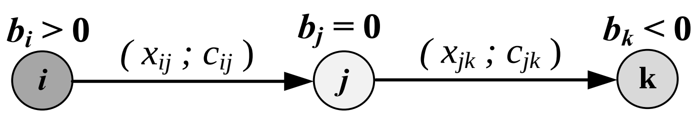


Figure 2. Types of Nodes

We write $(i, j) \in \mathcal{A}$ to say that there is an arc between nodes $i \in \mathcal{N}$ and $j \in \mathcal{A}$. In a directed network, the arc (i, j) is regarded as extending from node i to node j Typically, a directed network model involves a flow or transportation of something along the arcs, in the specified directions. In an undirected network, the arc (i, j) just represents a connection between nodes i and j An undirected network model may allow flows in either direction along an arc, or may not involve explicit flows at all.

2.2 Brief Definitions about Network Graphs

To understand a network flow model, some key terms must be defined.

- In an undirected graph arcs are unordered pairs of nodes $\{i, j\} \in \mathcal{A}$, for $i, j \in \mathcal{N}$. In a *directed graph* arcs are ordered pairs of nodes $(i, j) \in \mathcal{A}$, for $i, j \in \mathcal{N}$. We call a directed graph a *network*.

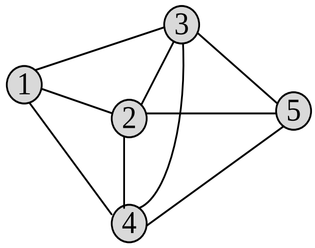


Figure 3. An undirected graph

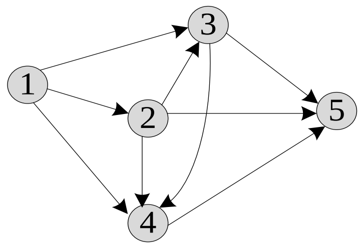
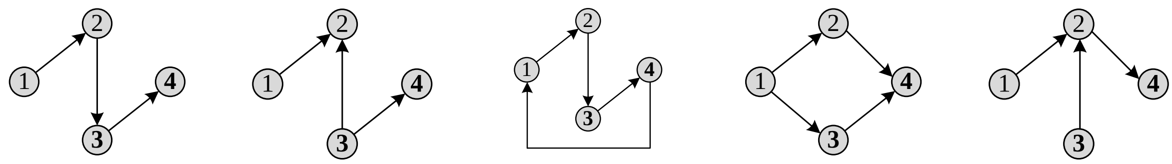


Figure 4. A directed graph

- A *walk* is a list of nodes $i_1, \ldots i_K$ such that, for each $k = 1, \ldots, K$, $\{i_k, i_{k+1}\} \in A$ for an undirected graph and (i_k, i_{k+1}) or $(i_{k+1}, i_k) \in A$ for a directed graph.
- A *path* is a walk where nodes $i_1, \ldots, i_K$ are distinct. A graph is *connected* if there is a path between any two nodes.
- A *cycle* is a walk where, $i_1 = i_K$ and also, $i_k \neq i_{k+1}$ for $k = 1, \ldots, K-1$ and $K > 2$. A graph is a cyclic if it does not contain any cycles.
- A *tree* is a connected a cyclic graph.
- A *subnetwork* is a graph (N', A') such that $N' \subset N$ and $A' \subset A$.
- A *spanning tree* is a subgraph with $N' = N$ that is also a tree [10],[11].

Some of the definitions are portrayed in graph in the following figures.



a) *A Path* **b)** *A Chain* **c)** *A Circuit* **d)** *A cycle* **e)** *A tree*

Figure 5. Representations of Network Definitions

2.3. Formulation of the Minimum Cost Flow Problem

The LP model of the minimum cost flow problem is shown below:

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij}$$

$$\textit{Subject to:} \sum_{j:(i,j)\in A} x_{ij} - \sum_{k:(k,i)\in A} x_{ki} = b(i) \text{ for all } i \in \text{N}$$

$0 \le x_{ij} \le u_{ij}$ for all $i \in A$

That the total net supply must equal zero can be seen by summing the flow balance equations over all $i \in \mathcal{N}$ resulting in:

$$(\text{Flow out of a node}) - (\text{Flow into a node}) = \text{Net supply at a node}$$

$$\sum_{i=1}^{N} b_i = 0$$

The problem is described in matrix notation as [10], [11]; $\min\{cx,\ \text{Ax=b and } 0 \le \text{l} \le \text{x} \le \text{u}\}$ [16]. where A is a node-arc incidence matrix having a row for each node and a column for each arc.

Note that there is one balance equation for each node in the network. The flow variables x_{ij} have only 0, +1, −1 coefficients in these equations[12].

3. Illustrative Numerical Example

We will consider a numerical example for determining the optimal flows of a network as shown below. In this example, there is a network which has 8 nodes and 10 arcs. Node 1 and node 2 are supply nodes, node 3 and node 4 are transshipment nodes, and node 5, node 6, node 7 and node 8 are demand nodes. The unit costs of arcs and the supplies and demands of the nodes are given below.

$$c_{ij}^T = \left[c_{13} c_{14} c_{23} c_{24} c_{35} c_{36} c_{37} c_{45} c_{46} c_{48}\right]$$

$$c_{ij}^T = \left[4\ 2\ 3\ 3\ 3\ 2\ 5\ 4\ 3\ 5\right]$$

$$b_i^T = \left[b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8\right]$$

$$b_i^T = \left[50\ \ -15\ 0\ \text{-20}\ 0\ 25\ \text{-20}\ 10\ \text{-30}\right]$$

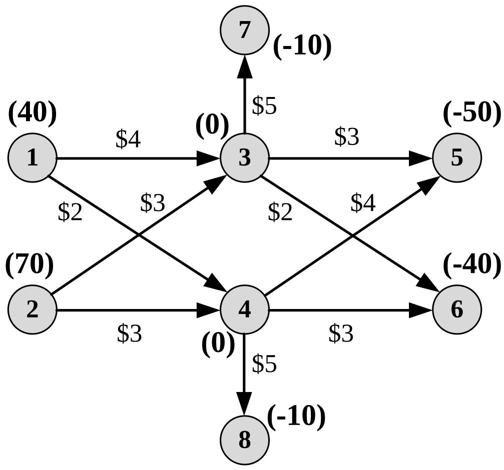


Figure 6. Illustrative Network Example

The objective is to find the optimal flows along the arcs of network in order to minimize the total cost subject to conservation of flow at each node. The objective function and the constraints will be composed as by using decision variables (number of flows; $x_{ij} = \left[x_{13}\ x_{14}\ x_{23}\ x_{24}\ x_{35}\ x_{36}\ x_{37}\ x_{45}\ x_{46}\ x_{48} \right]$ as;

The Objective Function,

$$Z_{\min} = 4\mathrm{x}_{13} + 2\mathrm{x}_{14} + 3\mathrm{x}_{23} + 3\mathrm{x}_{24} + 3\mathrm{x}_{35} + 2\mathrm{x}_{36} + 5\mathrm{x}_{37} + 4\mathrm{x}_{45} + 3\mathrm{x}_{46} + 5\mathrm{x}_{48}$$

Subject to:

		Dual Variables
Node 1:	$x_{13} + x_{14} = 40$	π_1
Node 2:	$x_{23} + x_4 = 70$	π_2
Node 3:	$-x_{13} - x_{23} - x_{35} + x_{36} + x_{37} = 0$	π_3
Node 4:	$-x_{14} - x_{24} - x_{45} + x_{46} + x_{48} = 0$	π_4
Node 5:	$-x_{35} - x_{45} = -50$	π_5
Node 6:	$-x_{36} - x_{46} = -40$	π_6
Node 7:	$-x_{37} = -10$	π_7
Node 8:	$-x_{48} = -10$	π_8

As stated earlier, this MCF problem is generally an integer program, since the decision variables x_{ij} are restricted to be zero or an integer (upper bound).. $0 \le x_{ij} \le u$.

A is the node-arc incidence matrix of a directed graph in our example.

Table 1. Representation of Node-Arc Incidence Matrix

Node / Arc	*1-3*	*1-4*	*2-3*	*2-4*	*3-5*	*3-6*	*3-7*	*4-5*	*4-6*	*4-8*
Node 1	1	1	0	0	0	0	0	0	0	0
Node 2	0	0	1	1	0	0	0	0	0	0
Node 3	-1	0	-1	0	1	1	1	0	0	0
Node 4	0	-1	0	-1	0	0	0	1	1	1
Node 5	0	0	0	0	-1	0	0	-1	0	0
Node 6	0	0	0	0	0	-1	0	0	-1	0
Node 7	0	0	0	0	0	0	-1	0	0	0
Node 8	0	0	0	0	0	0	0	0	0	-1

The matrix A, node-arc incidence matrix of a connected digraph, has one row for each node of the network and one column for each arc. Each column of A contains exactly two nonzero coefficients: a "$+1$" and a " -1". Thus, the columns of A are given by;

$$a_{ij} = e_i - e_j$$

where e_i and e_j are unit vectors in R^m, with ones in the i. and j. positions. Clearly, the A matrix does not have full rank, since the sum of its rows is the zero vectors [13], [14], [17].

Solving a MCF problem with the simplex algorithm, one has therefore that all the basic feasible solutions explored by the algorithm are spanning trees of the flow network. As it occurs for any LP, also in min-cost flow problems one has feasible, infeasible and degenerate bases. A basis is feasible if $x_B = B^{-1}b$. In this case, it can be easily verified solving the system $Bx_B = b$, starting from a leaf of the spanning tree, and verifying that $x_B \geq 0$. [2], [4].

3.1. Feasible Solution and Optimality

To find the optimal solution; first, assume that we have a network with n nodes, which is a spanning tree. In order to show that the variables corresponding to the arcs in the tree constitute a basis, it is sufficient to show that the $(n-1)$ tree variables are uniquely determined [17]. In the simplex method, this corresponds to setting the nonbasic variables to specific values and uniquely determining the basic variables [14].

Therefore, the original network with n nodes must have had $(n-1)$ arcs. Next, we show that if an n node connected subnetwork has $(n-1)$ arcs and no loops, it is a spanning tree.

In a general minimum-cost flow model, a spanning tree for the network corresponds to a basis for the Simplex method. There must be at least two ends in the spanning tree since it contains no loops.

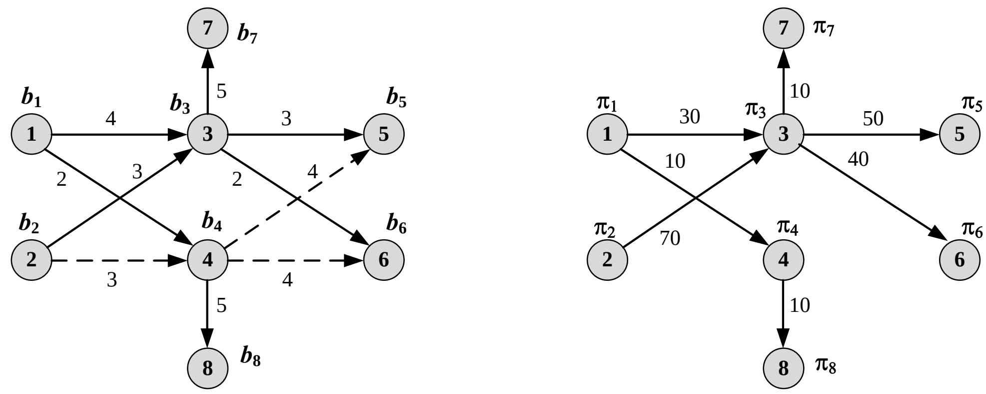


Figure 7. First Step Solution of Example

In this step *Total* Cost=4.30+2.10+3.70+5.10+3.50+2.40+5.10=680

We do not know that, this feasible solution is optimal or not. We show how to efficiently check the optimality of a basic feasible solution, using complementarity slackness conditions to determine the optimality of the problem. Therefore we have to control by finding node potentials and then compute reduced costs for all nontree arcs, using these node potentials.

The dual of the min-cost flow problem is the following:

$$\max\ \mathrm{b}^T \pi \leftrightarrow \pi^T A \le c_B$$

Actually, π_i can be viewed as the linear programming dual variable (dual solution) corresponding to the flow conservation constraint of node i [3]. With respect to a given potential function π, the reduced cost of an *arc* $(i;j)$ is defined as $c_{ij}^{\pi} = \pi_i - \pi_j - c_{ij}$ [8]

For the arcs of basis in our example, the node potentials are:

Arc 1-3	$\pi_1 - \pi_3 = c_{13}$	$\pi_1 - \pi_3 = 4$	$\pi_1 = 7$, $\pi_3 = 3$
Arc 1-4	$\pi_1 - \pi_4 = c_{14}$	$\pi_1 - \pi_4 = 2$	$\pi_4 = 5$, $\pi_1 = 7$
Arc 2-3	$\pi_2 - \pi_3 = c_{23}$	$\pi_2 - \pi_3 = 3$	$\pi_3 = 3$, $\pi_2 = 6$
Arc 3-5	$\pi_3 - \pi_5 = c_{35}$	$\pi_3 - \pi_5 = 3$	$\pi_3 = 3$, $\pi_5 = 0$
Arc 3-6	$\pi_3 - \pi_6 = c_{36}$	$\pi_3 - \pi_6 = 2$	$\pi_3 = 3$, $\pi_6 = 1$
Arc 3-7	$\pi_3 - \pi_7 = c_{37}$	$\pi_3 - \pi_7 = 5$	$\pi_3 = 3$, $\pi_7 = -2$
Arc 4-8	$\pi_4 - \pi_8 = c_{48}$	$\pi_4 - \pi_8 = 5$	$\pi_8 = 0$, $\pi_4 = 5$

$$\pi_1 = 7,\ \pi_2 = 6,\ \pi_3 = 3,\ \pi_4 = 5,\ \pi_5 = 0,\ \pi_6 = 1,\ \pi_7 = -2,\ \pi_8 = 0$$

After finding the node potentials of the arcs of basis, we have to determine the reduced costs for the nonbasic arcs. A feasible flow x is an optimum flow if and only if it admits no negative cost augmenting cycle. If all of the reduced costs for the nonbasic arcs are computed as positive number, the feasible solution is optimal. We now compute reduced costs for all nontree arcs, using node potentials founded before:

Reduced costs for the nonbasic arcs using equation below.

$$(c_{ij}^{\pi} = \pi_i - \pi_j - c_{ij} \ \text{ or } \ c_{ij}^{\pi} = c_{ij} - \pi_i + \pi_j) \qquad [3], [4], [7], [10], [11]$$

Arc 2-4	$\pi_2 - \pi_4 - c_{24} = 6 - 5 - 3 = -2$
Arc 4-5	$\pi_4 - \pi_5 - c_{45} = 5 - 0 - 4 = 1$
Arc 4-6	$\pi_4 - \pi_6 - c_{46} = 5 - 1 - 4 = 0$

If examined the reduced costs, you will see that the reduced cost of *Arc* 4-5 is positive therefore we have to choose the *Arc* 4-5 as entering arc. Now we have to decide to determine the leaving arc. To be able to understand better the topic of leaving arc, we have to constitute a cycle by adding *Arc* 4-5. In this cycle, if the flow on *Arc* 4-5 is θ then the arcs of basis will be:

$$Arc\ 1\text{-}3 = 30-\theta,\ \ Arc\ 3\text{-}5 = 50-\theta \text{ and } Arc\ 1\text{-}4 = 10+\theta,$$

The minimum value of the θ should be 30 for nonnegativity constraints for the flows. So, the flows will be like;

$$Arc\ 1\text{-}3 = 0, Arc\ 3\text{-}5 = 20 \text{ and } Arc\ 1\text{-}4 = 40$$

The flow on *Arc* 1-3 is 0, therefore Arc 1-3 is a leaving arc.

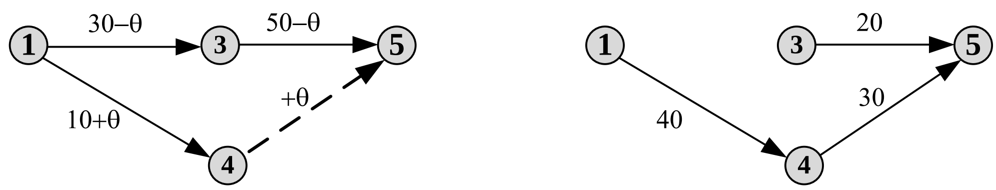


Figure 8. Simple Illustration of Leaving and Entering arc

In our example the entering arc must be *Arc* 4-5. Leaving arc is *Arc* 1-3.

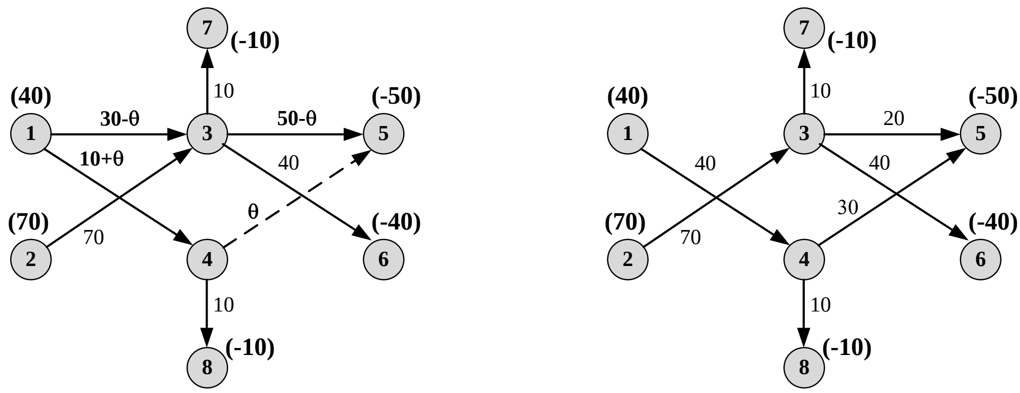


Figure 9. Last Step Solution of Example (Optimal)

In this step, *Total* Cost=2.40+3.70+3.20+2.40+5.10+4.30+5.10=650

We now have to find node potentials again for Basic Nodes:

Arc 1-4	$\pi_1-\pi_4=c_{14}$	$\pi_1-\pi_4=2$	$\pi_4=5,\ \pi_1=7$
Arc 2-3	$\pi_2-\pi_3=c_{23}$	$\pi_2-\pi_3=3$	$\pi_3=4,\ \pi_2=7$
Arc 3-5	$\pi_3-\pi_5=c_{35}$	$\pi_3-\pi_5=3$	$\pi_3=4,\ \pi_5=1$
Arc 3-6	$\pi_3-\pi_6=c_{36}$	$\pi_3-\pi_6=2$	$\pi_6=2,\ \pi_3=4$
Arc 3-7	$\pi_3-\pi_7=c_{37}$	$\pi_3-\pi_7=5$	$\pi_3=4,\ \pi_7=-1$

Arc 4-6	$\pi_4 - \pi_6 = c_{46}$	$\pi_4 - \pi_6 = 3$	$\pi_4 = 5$, $\pi_6 = 2$
Arc 4-8	$\pi_4 - \pi_8 = c_{48}$	$\pi_4 - \pi_8 = 5$	$\pi_8 = 0$, $\pi_4 = 5$

$$\pi_1 = 7,\ \pi_2 = 7,\ \pi_3 = 4,\ \pi_4 = 5,\ \pi_5 = 1,\ \pi_6 = 2,\ \pi_7 = -1,\ \pi_8 = 0$$

Reduced Costs for Nonbasic Nodes [3], [4], [7], [11]:

Arc 1-3	$\pi_1 - \pi_3 - c_{13} = 7 - 4 - 4 = -1$
Arc 2-4	$\pi_2 - \pi_4 - c_{24} = 7 - 5 - 3 = -1$
Arc 4-6	$\pi_4 - \pi_6 - c_{46} = 5 - 2 - 4 = -1$

If reexamined the reduced costs, you will see that all of them are zero or negative. So, the last flow of directed graph is optimal. If one of the reduced costs of an arc is zero, it shows that there is an alternative optimal solution for the example.

4. Case Study for Network Simplex Solution

A company distributes its product which has manufactured in three different cities. These cities are called supply nodes. In addition to three cities, there are 9 cities which use this company's product. These cities are called demand nodes. Also 2 cities have no demands, they are only transshipment nodes. So the problem has 14 nodes and 27 arcs between these nodes. The unit transportation costs for these arcs are given below. The objective of the company is to determine the optimal flows in order to minimize the total cost of its transportation.

For this minimum cost flow problem, the data and the directed graph are given below:

$$b_i^T = [b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{11} b_{12} b_{13} b_{14}]$$

$$b_i^T = [30\ \ -20\ \ -30\ \ 70\ \ -15\ \ -10\ \ 0\ \ \text{-}20\ \ 0\ \ \text{-}10\ \ \text{-}20\ \ 25\ \ 55\ \ \text{-}5]$$

Arcs	Unit Costs ($)	*Arcs*	Unit Costs ($)	*Arcs*	Unit Costs ($)
1-2	2	5-6	6	8-11	11
1-3	3	5-7	7	9-10	10
1-4	4	5-8	8	10-14	14
2-4	4	6-7	7	11-10	10
2-6	6	6-12	12	11-14	14
3-9	9	7-8	8	12-7	7
4-3	3	7-11	11	13-11	11
4-5	5	8-9	9	13-12	12
4-9	9	8-10	10	14-13	13

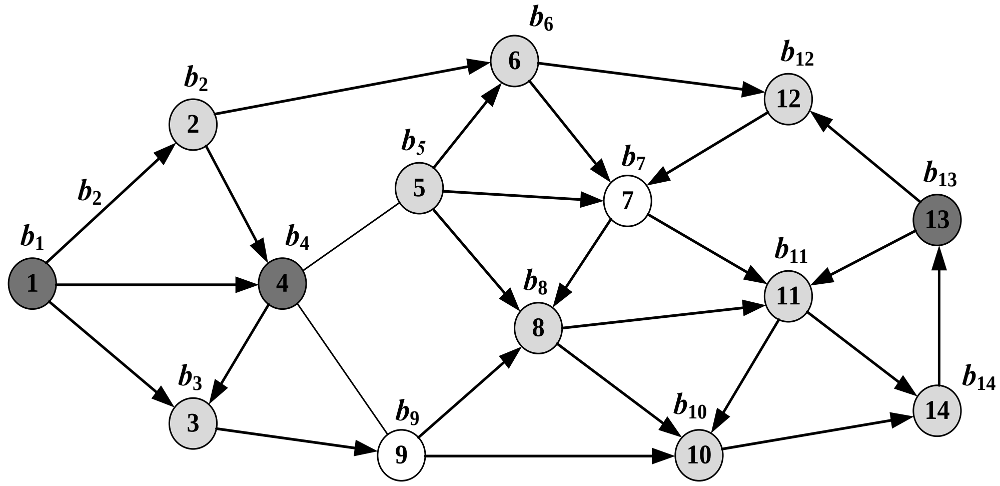


Figure 10. Network graph of company's product flows

To find the optimal solution, network simplex method is applied to a feasible dual solution. As the tool, Microsoft Excel 2010 Solver is used for solving this model. The results are given in table below.

Arcs	*Flow*	*Arcs*	*Flow*	*Arcs*	*Flow*
1-2	20	5-6	10	8-11	0
1-3	10	5-7	0	9-10	10
1-4	0	5-8	15	10-14	0
2-4	0	6-7	0	11-10	0
2-6	0	6-12	0	11-14	5
3-9	0	7-8	5	12-7	5
4-3	20	7-11	0	13-11	25
4-5	40	8-9	0	13-12	30
4-9	10	8-10	0	14-13	0

Total Minimum Transportation (Transshipment) Cost is $675 for the company. The optimal arc flows are shown in figure below. [Bold arcs are basis (tree) and discrete arcs are nonbasic (nontree)].

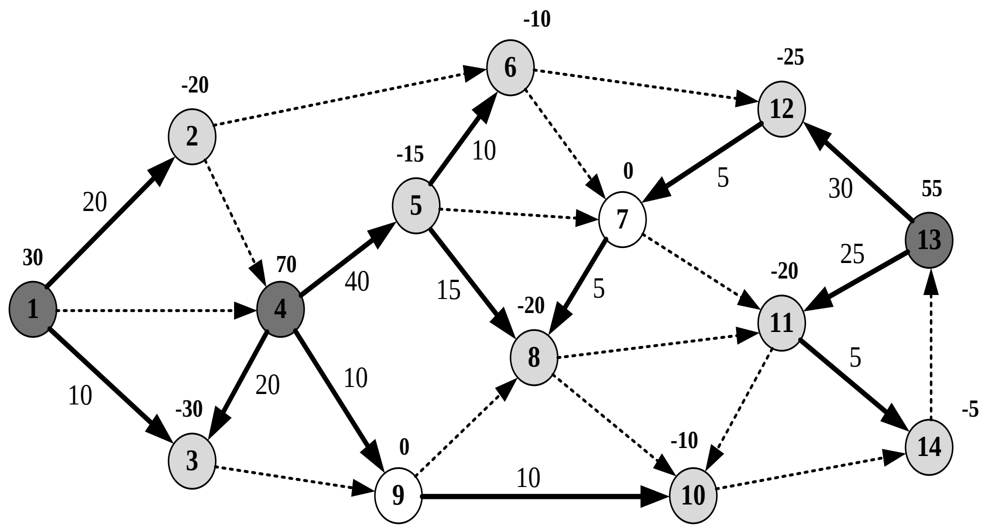


Figure 11. Optimal Solution, Bold Arcs (*Basic*), Others (*Nonbasic*)

5. Conclusion

Networks are very important subclass of linear programs that are intuitive, easy to solve and useful for modeling business problems. Networks provide a useful way to think about problems even if there are additional constraints or variables that preclude use of networks for modeling the whole problem. In this paper, an implementation of network simplex algorithm is described for solving the minimum cost network flow problem. The MCF problem plays a fundamental role in network flow theory and has a wide range of applications. For finding the optimal flows, NSM dual feasible solution has been used for the network problem showing all calculations on an illustrative example. Finally, a bigger network problem is solved using the same method and the results are shown in the end. The results show that if the size of network is moderate, NSM is useful for solving network flows and finding optimal cost.

References

[1] Churchman, CW., Ackoff, R.L and Arnoff, E.L., "Introduction to Operations Research" Hardcover, January 1, 1959.

[2] Dantzig, George B., "Linear Programming and Extensions", Springer Inc, 1963.

[3] Bradley, S.P., Hax, A.C. and T.L. Magnanti, "Applied Mathematical Programming", Addison-Wesley, 1977.

[4] Ahuja, R. K., Magnanti, T. L. and J. B. Orlin, "Network Flows: Theory, Algorithms and Applications" New Jersey, Prentice Hall, Englewood Cliffs, NJ, 1993.

[5] Bertsimas, D. and J.N. Tsitsiklis, "Introduction to Linear Optimization", Athena Scientific, 1997.

[6] Kantorovich, L.V. "Mathematical Methods in the Organization and Planning of Production", translated in Management Science (1960), Volume: 6, p: 336-422, 1939.

[7] Ahuja, R.K., Magnanti, T.L., and J.B. Orlin, "Network Flows" in "Handbooks in Operations Research and Management Science", Volume: 1 Edited by G.L. Nemhauser (Elsevier Science Publications B.V., North Holland), 1989.

[8] Ahuja, R.K. and J. Orlin, "Improved Primal Simplex Algorithms for Shortest Path, Assignment and Minimum Cost Flow Problems", Massachusetts Institute of Technology, Operations Research Center, Working Paper, pp. 189–188, 1988.

[9] Kirali, Z. and P. Kovacs, "Efficient implementations of minimum cost flow algorithms", Acta Univ. Sapientiae, Informatica, 4, p: 67-118, 2012.

[10] Vanderbei, Robert J., "Linear Programming: Foundations and Extensions, Kluwer Academic Publishers, Second Edition, 2001.

[11] Goldberg, A.V., "An Efficient Implementation of a Scaling Minimum-Cost Flow Algorithm", Journal of Algorithms, 22(1), p: 1–29, 1997.

[12] Morrison, David R. Sauppe, Jason J. and Sheldon H. Jacobson, "A Network Simplex Algorithm for the Equal Flow Problem on a Generalized Network", INFORMS Journal on Computing, December, 2011.

13] Goldberg, A. V., Grigoriadis, M. D. and R. E. Tarjan, "Use of dynamic trees in a network simplex algorithm for the maximum flow problem", Mathematical Programming, 50, p: 277-290, 1991.

[14] Bazaraa, Mokhtar S., Jarvis, John J. and Hanif D. Sherali, "Linear Programming and Network Flows", Third Edition, John Wiley & Sons Inc, Hardcover, 2010.

[15] Geranis, G., Paparrizos K. and A. Sifaleras, "On a dual network exterior point simplex type algorithm and its computational behavior", RAIRO-Operations Research 46, p: 211-234, 2012.

[16] Hillier, Frederick S., Lieberman and J. Gerald, "Introduction to Operations Research", McGraw-Hill, New York, 2010.

[17] Amberg, A., Domscheke, W. and S. Braunschweig, (1996). "Capacitated Minimum Spanning Trees: Algorithms using intelligent search", Combinatorial optimization: Theory and Practice, 1, p: 9-39, 1996.