TRANSPORTATION PROBLEMS WITH LOSSY ARCS

COLLEGE OF NATURAL SCIENCE
DEPARTMENT OF MATHEMATICS

Graduate Project Report Submitted to the Department of Mathematics,Addis Ababa University in Partial Fulfilment of the Requirements for the Degree of Master of Science in Mathematics

Stream:Optimization

Prepared by: Nahusenay Adefris

Advisor: Berhanu Guta(Ph.D)

16, October, 2015
Addis Ababa, Ethiopia

Contents

Addis Ababa University
Department of Mathematics

The undersigned hereby certify that they have read and recommend to the school of graduate studies for acceptance of a project entitled **Transportation Problem With Lossy Arcs** by Nahusenay Adefris in partial fulfillment of the requirements for the degree of master of Science.

Dated: 16,October, 2015

Advisor: _____
Dr.Berhanu Guta

Examining committee: 1. Dr. Semu Mitku _____

2. Dr. Tilahun Abebaw _____

16, October, 2015
ADDIS ABABA ,ETHIOPIA

Acknowledgment

First,I would like to express my deepest gratitude to **God** for giving me patience; I am also grateful to my advisor Dr. Berhanu Guta for his helpful discussion, comments and providing the necessary materials in the preparation of this paper.

Next, I would like to extend my thank to my families and all who encouraged me to complete my project and their heart full help while writing the paper.

Finally, I would like thanks to Department of mathematics, Addis Ababa University for giving the necessary materials throughout the preparation of this paper.

Abstract

The Ordinary Transportation Problem allows to model the situations where the amount of goods leaving the supply points is equal to the amount delivered to the destinations.
In this project, the model of Transportation Problem with lossy arcs, the situations where the amount of goods leaving the supply points is not equal to the amount delivered to the destinations (this is the case, when fragile or perishable goods are transported or the complaints may occur) is presented. Each problem of this type can be transformed to the form of a Generalized Minimum Cost Flow Problem with linear objective function.
The presented solution method uses the ideas applied in the Method of Solving a Generalized Minimum Cost Flow problem.

Keywords:
Transportation Problem with Lossy Arcs, Generalized Transportation Problem, Generalized Minimum Cost Flow Problem and Generalized Network Simplex Algorithm.

Notation

c_{ij}	Cost From Node i to Node j
f_{ij}	Flow From Node i to Node j
u_{ij}	Capacity From Node i to Node j
$NA(i)$	Set of Nodes After i
$NB(i)$	Set of Nodes Before i
$MCFP$	Minimum Cost Flow Problem
TP	Transportation Problem
GTP	Generalized Transportation Problem
$TPWLA$	Transportation Problem With Lossy Arcs
$GMCFP$	Generalized Minimum Cost Flow Problem
δ_{ij}	Flow Reduction Ratio

Chapter 1

Introduction and Some Terminologies for TPWLA

1.1 Introduction

The Transportation Problem With lossy Arcs(TPWLA), arises in many real-life applications and it is the subclass of the generalized transportation Problem (GTP). It has the form of ordinary transportation problem, with additional assumption that the quantities of goods change during the transportation process. It is a version of transportation problem involving the possibility that the amount of good transported from supply points to destination points changes during the transportation process(in particular, if the amount of goods decreases, then the change is represented by the reduction ratio).

More general version, the Generalized Minimum Cost Flow Problem,of TPWLA was described in [1]. This is a generalization of the well-known Minimum Cost Flow Problem, where we assume in addition, that the quantity of good changes during the flow process. One may find various examples of applications of this problem, in particular in the analysis of the financial, mineral or energy networks, aircraft assignment, managing the warehousing goods and funds flows or in the land management, etc. Another interesting application may be found in [4]. Here the authors apply the generalized flows to the problems of transporting perishable products, such as blood, medical nuclear materials, food, pharmaceutical and fast fashion apparels. Various algorithms for generalized flow problems and more detailed considerations found in [1].
In the above publications ([1]*and*[4]) concerning the lossy versions of the problems it was assumed that the demand at every destination point is dis-

cretely distributed.
In this project we use the generalized simplex algorithm to solve the transportation problem with lossy arcs.
Hence, in the first chapter we will see some terminologies and examples, in the second chapter we will see about transportation problem with lossy arcs and finally we will try to discuss about the solution procedure.

1.2 Problem Description

In the ordinary transportation problem we assume that the amount of goods that we send from supply node to the demand node is equal to the amount that actually reach at the destination point. That means,for a balanced problem, we have $\sum a_i$=$\sum b_i$ and for an unbalanced problems$\sum a_i \neq \sum b_i$ we can satisfy the demand requirement by implementing a dummy destination or a dummy source. But there is a case where $\sum a_i > \sum b_i$ and the arcs are lossy due to many factors affecting transportation problems,such as evaporation,theft,taxes and so on. In such a case the demand requirements may or may not be met.

Here we assume that the amount of good from m supplies to n destination points changes during the transportation process. Thus the amount f_{ij}, leaving supply point i, is modified by a multiplier δ_{ij} (in most cases belonging to the interval $(0, 1)$, that means reduction of the amount of good), which is the main focus of this project.

Finally the amount of good delivered to destination j from supply i equals to $\delta_{ij} f_{ij}$. We assume that the unit transportation costs c_{ij} are constant, demand b_j at every destination point must be satisfied, and supply a_i of every supply point must not be exceeded. The model can be thus written as follows.

$$Min\{Z(f) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} f_{ij}\}$$

Subject to:

$$\sum_{i=1}^{m} \delta_{ji} f_{ji} = b_j, \quad where \;\; j = 1, 2, ..., n.$$

$$\sum_{i=1}^{n} f_{ij} \leq a_i, \quad where \;\; i = 1, 2, ..., m.$$

$$f_{ij} \geq 0, \forall (i, j) \in A$$

The values of the multipliers δ_{ij} depend, in general, on the delivery time (that means, as the total delivery time increases the total amount of flow that can be lost also increases and vis-a-vis) and distance between particular source and destination points.

In this project we try to give the solution method to address the following question:
How can we meet the demand requirement by handling losses through transportation process with a minimum transportation cost.

1.3 Some Terminologies and Examples

A ***network*** is a graph $G = (N, A)$ consisting f a set of points called *nodes* denoted by N and a set of lines called the *arcs* denoted by A whose elements are ordered pairs of distinct nodes.

A**Network flow** is a directed graph where each edge has a capacity and each edge receives a flow. Here we associate with each arc $(i, j) \in A$, a cost c_{ij}, and a capacity $u_{ij} \geq 0$. Frequently, we distinguish two special nodes in a graph, the source s and sink t.
An arc (i, j) has two end points i and j. The arc (i, j) is incident to nodes i and j.

A ***flow*** $f : A \to \mathbb{R}^+$ such that the following two conditions are satisfied:
capacity constraint

$$0 \leq f_{ij} \leq u_{ij} \tag{1.1}$$

and
flow conservation

$$\sum_{\{j:(i,j)\in A\}} f_{ij} = \sum_{\{j:(j,i)\in A\}} f_{ji}, \quad \forall i \backslash \{s, t\} \tag{1.2}$$

Capacity Function:
Capacity u of an edge is denoted by u_{ij} which represents the maximum amount of flow(possibly infinity) that can pass through an edge. The capacity u_{ij} limits the amount of flow we are permitted to send into arc (i, j).

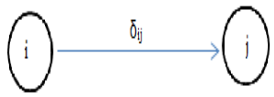
A **Path** in a network is a walk without any repetition of nodes. We can partition the arcs of a path into two groups: forward arcs and backward arcs.An arc (i, j) in the path is a forward arc if the path visits node i prior

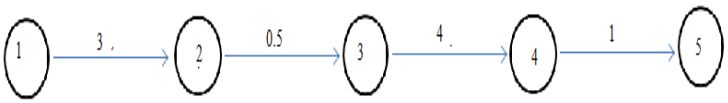
to visiting node j, and is a backward arc otherwise.
A **directed path** in a graph $G = (N, A)$ is a sequence of distinct nodes and arcs satisfying the property that $(i_k, i_{k+1}) \in A$ for each $k = 1, 2, ..., r-1$.

Flow Along Paths
Let P be a path from node i to node j (not necessarily directed) in the network. Let $\overline{P}$ and $\underline{P}$ denote the set of forward arcs and backward arcs in P.
Definition:
The path multiplier of a path P is defined as follows:

$$\delta(P) = \frac{\prod_{(i,j)\in\overline{P}} \delta_{ij}}{\prod_{(i,j)\in\underline{P}} \delta_{ij}} \tag{1.3}$$

Example





If we send 2 units of flow from node 1 to node 2, node 2 received 6 units of flow, node 3 received 3 units of flow, node 4 received 12 units of flow and node 5 received 12 units of flow.
This implies, 2 units were first sent, and eventually 12 units were received. Notice that the path multiplier is 6.

$$\delta(P) = \frac{\prod_{(i,j)\in\overline{P}} \delta_{ij}}{\prod_{(i,j)\in\underline{P}} \delta_{ij}} = \frac{3 \times 0.5 \times 4 \times 1}{1} = 6$$

Property 1.1
If we send 1 unit of flow from node i to node j along a path P, then $\delta(P)$ units become available at node j.

Assumption 1.1:
If $\overline{P}$ or $\underline{P}$ does not exist, we assume that the path multiplier is given by:

$$\delta(P) = \prod_{(i,j)\in P} \delta_{ij}$$

That means, all the arcs are either a forward arcs or a backward arcs. In addition, if there is no any path from node i to node j, we have nothing to do.
A chain in a graph $G = (N, A)$ is a sequence of arcs connecting two nodes without any repetition of nodes. All the arcs are a forward arcs.

A **Cycle** in a network is a chain with an additional arc connecting the two end nodes. By convention, when traversing a cycle, arc (i, j) is a forward arc if node i is visited prior to node j, and is a backward arc if node j is visited prior to node i.

Flow Along Cycles
Let W be a path from node i to itself (not necessarily directed with some orientation)in the network. Let $\overline{W}$ and $\underline{W}$ denote the set of forward arcs and backward arcs in W.
Definition:
The cycle multiplier of a cycle W is defined as follows:

$$\delta(W) = \frac{\prod_{(i,j)\in\overline{W}} \delta_{ij}}{\prod_{(i,j)\in\underline{W}} \delta_{ij}} \tag{1.4}$$

Assumption 1.2:
If $\overline{W}$ or $\underline{W}$ does not exist, we assume that the cycle multiplier is given by:

$$\delta(W) = \prod_{(i,j)\in W} \delta_{ij}$$

That means, W is either a forward arc or a backward arc.
Definition:
1) If $\delta(W) > 1$, then an excess is created in node s,where s is a source node or a demand node. We call this a gainy cycle.
2) If $\delta(W) < 1$, then a deficit is created in node s,where s is a source node or a demand node. We call this a lossy cycle.
3) If $\delta(W) = 1$, then mass balance is conserved in all nodes. We call this a breakeven cycle.

A **Tree** in a network $G = (N, A)$ is a connected graph that contains no cycle.

A graph that contains no cycle is a **forest**. Alternatively, a forest is a collection of trees.

A **Cost**(c_{ij})is the cost that must be paid per unit of flow that goes through an edge.

Bipartite Network:
A network $G = (N, A)$ is said to be a **bipartite network** if its vertex set N can be partitioned in to two non empty subsets N_1 and N_2 such that each edge in A has one end point in N_1 and the other in N_2 (i.e, there are no lines in A joining a pair of points both of which are either in N_1 or in N_2). The partition (N_1, N_2) is then called a bipartition of G and G itself denoted by (N_1, N_2, A). Let $n = |N|$, $n_1 = |N_1|$, $n_2 = |N_2|$, $m = |A|$, and assume without loss of generality that $n_1 \leq n_2$. We call a bipartite network is unbalanced if $n_1 \ll n_2$ and balanced if $n_1 = n_2$.

Definition:
The **arc gain**, δ_{ij}, multiplies the flow at the beginning of the arc to obtain the flow at the end of the arc. When a flow f_{ij} is assigned to an arc, this flow leaves the origin node of the arc. The flow entering the terminal node of the arc is $\delta_{ij}\ f_{ij}$. The arc lower bound, upper bound, and cost all refer to the flow at the beginning of the arc.
If $0 < \delta_{ij} < 1$, then the model losses a flow such as due to evaporation,theft or spoilage.
If $\delta_{ij} > 1$,then the model gain a flow such as due to interest revenue in currency conversion.
In all of these networks, we wish to send some commodity, which we generally call flow, from one node to another node that can be formulated as generalized network or pure network.

1.3.1 Generalized Network Flows

A common scenario of a network flow problem arising in industrial logistics concerns the distribution of a single homogenous product from plants (or origins) to consumer markets (or destinations). The total number of units produced at each plant and the total number of units required at each market are assumed to be known. The product need not be sent directly from source to destination, but may be routed through intermediary points reflecting warehouses or distribution centers.
Further, there may be capacity restrictions that limit some of the shipping links. The objective is to minimize the variable cost of producing and shipping the products to meet the consumer demand.
The sources, destinations, and intermediate points are collectively called

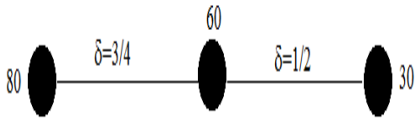
nodes of the network, and the transportation links connecting nodes are termed arcs.
Although a production(or distribution) problem has been given as the motivating scenario, there are many other applications of the general model.

Table 1.1 indicates a few of the many possible alternatives.

Networks	Nodes	Arcs	Flows
Communication	Telephone exchanges	Cables	Voice messages, video
Hydraulic	pumping stations	Pipelines	Hydraulic fluid, water, gas
Financial	Currencies, stocks	Transactions	Money
Transportation	Air ports,	Airline routes	Freight, passengers

In pure network flow problems, we assumed that if f_{ij} units of a commodity enter an arc (i, j) at its tail node i and travel across the arc, then exactly the same f_{ij} units will reach its head node j. This assumption may not hold in some flow models.

Example 1
Consider the following graph



If we start from 80 units of flow, we obtain 60 units after following the first arc and 30 units after the second arc.

Example 2
In a water distribution network, if some quantity of water is shipped across an open canal linking two nodes, some is lost due to evaporation and seepage during transit, and the amount reaching the destination will only be a fraction of the amount that left the origin. The same phenomenon takes place in the transmission of electric power through high voltage transmission lines, because of transmission losses.

Example 3
If we transmit money from one period to the next by holding it in a bank

account (cash flow), because of the interest earned, the amount reaching the destination will be more than the amount that left the origin.
In all these examples there exists a positive multiplier δ_{ij} associated with arc (i, j) such that if a packet of f_{ij} units of the commodity enter the arc (i, j) at node i, and travels through the arc, then by the time it reaches node j, the packet contains $\delta_{ij}\ f_{ij}$ units of the commodity.

- If $0 < \delta_{ij} < 1$, arc (i, j) is said to be lossy; and
- If $1 < \delta_{ij} < \infty$, it is said to be gainy.

In pure networks, $\delta_{ij} = 1$ for all arcs (i, j), flow problems on them have been called pure network flow problems.
If $\delta_{ij} \neq 1$ for at least one arc, the network is called a generalized network, or a network with multipliers, or a network with gains or losses, and flow problems on it are called **generalized network flow problems.**

We assume that the flow variable f_{ij} associated with an arc (i, j) in a generalized network always refers to the amount of material entering this arc at its tail node i for transit to node j. We also assume that all the data on this arc (lower bound, capacity, cost coefficient, multiplier) applies to this variable. Let $G = (N, A, l = (l_{ij}), u = (u_{ij}), \delta = (\delta_{ij}), c = (c_{ij}), S, T)$ be a connected directed generalized network with $|N| = n \geq 2$, $|A| = m$, δ as the vector of multipliers associated with the arcs in A, and the usual meaning for the other symbols.

Remark

- u_{ij} is an upper bound on the flow that we send from i, not the flow that arrives at node j.
- c_{ij} is the cost per unit flow we send from i, not the per unit cost that arrives at node j.

Example
Assume that the flow from i to j is $f_{ij} = 10\ units$,$\delta_{ij} = 0.8$ and $c_{ij} = 4$ dollars
Then node j received 8 units of flow and pay 40 dollars not 32 dollars.

Remark
Let v_s,v_t denote the amounts of material leaving the source node s, and

arriving at the sink node t respectively. Then the flow vector $f = (f_{ij})$is feasible in G if it satisfies

$$\sum_{j \in NA(i)} f_{ij} - \sum_{j \in NB(i)} \delta_{ji} f_{ji} = \begin{cases} v_s, \ i = s; \\ 0, \ i \neq s, t; \\ -v_t, \ i = t. \end{cases} \tag{1.5}$$

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

Because of the multipliers, v_t may not be equal to v_s in (1.5). In the coefficient matrix of the equality constraints in (1.5), each column has exactly two nonzero entries, one of them a 1 and the other a negative of the positive multiplier associated with the arc in G corresponding to this column.
We may be interested in maximizing v_t given v_s subject to (1.5), this problem is known as the minimum loss problem.
A feasible flow vector which maximizes v_t is known as a maximum feasible flow vector. Among all maximum feasible flow vectors, the one which is associated with the smallest value of v_s is known as an optimum maximum feasible flow vector. In general, the constraints in a generalized network flow problem may be inequalities.

For example, the net shipment out of a source node could be $\leq$ the amount available at it. Or, the net shipment reaching a sink node could be $\leq$ the requirement there. Each inequality constraint leads to a slack variable in the system when all the constraints are written as equations, and the column associated with that slack variable contains only one nonzero entry, a +1 or 1. If this nonzero coefficient is in row i, that column corresponds to the self loop (i, i) at node i.
It is said to be a surplus self loop if the nonzero coefficient in its column is 1, slack self loop if that entry is+1. The multiplier associated with a self loop is always taken to be +1, whether it is a surplus or a slack self loop. Hence, for all self loops (i, i) in the network, $\delta_{ii} = +1$.

Thus the generalized network $G = (N, A)$ on which our problem defined is a directed network which may have self loops, and multiple arcs joining the same pair of nodes with the same or different orientations. To handle the self loops, we modify the definitions of the before and after sets for any node i in G to be the following:

- $A(i)$ = Set of head nodes on arcs incident out of i; and 1 if there is a surplus self loop at i.

- $B(i)$ = Set of tail nodes on arcs incident out of i; and 1 if there is a slack self loop at i.

1.3.2 Minimum Cost Flow Problem

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 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 , which is defined by a set of nodes , and a set of arcs connecting the nodes.

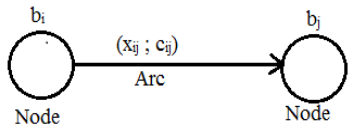


Figure 1.Representation of Node and Arc

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 (or supply nodes), the customers (demand nodes) and intermediate storage facilities (transshipment nodes).
The minimum cost flow problem is a network model that holds a central position among network optimization models, because it encompasses such a broad class of applications.

- Like the maximum flow problem, it considers flow through a network with limited arc capacities.

- Like the shortest-path problem, it considers a cost (or distance) for flow through an arc.

- Like the transportation problem or assignment problem, it can consider multiple sources (or supply nodes) and multiple destinations (or demand nodes) for the flow, again with associated costs.

In fact, all four of these problems are special cases of the pure minimum cost flow problem.
The single commodity minimum cost flow problem is one of the most fundamental models in network flow theory. It is to find a feasible flow of minimum total cost from a set of supply node to a set of demand node in a network with capacity constraints and arc costs. This model can be used directly in various real world applications, which arises in transportation, logistics, telecommunication, network design, resource planning, scheduling and many other industries.
Moreover, it often arises as a sub problem in more complex optimization models, such as multi-commodity flow problems. It can be specialized in to two other problems:

- If the capacity constraint is removed, the problem is reduced to the shortest path problem.

- If the costs are all get equal to zero the problem is reduced to the maximum flow problem.

There are two possible ways of formulating the minimum cost flow problem corresponding to the network models. These are generalized minimum cost flow problem and pure minimum cost flow problem.

1.3.3 Generalized Minimum Cost Flow Problem

A generalized minimum cost flow problem is a generalized network model which is defined by a given set of arcs and a given set of nodes, where each arc has a known capacity, unit cost and each node has an external flow.

Mathematical formulation
Let $G = (N, A)$ be a directed graph consisting $n = |N|$ nodes and $m = |A|$arcs. Each arc $(i, j) \in A$ has an associated cost c_{ij} per unit flow on that arc. We assume that the flow cost is nonnegative that varies linearly with the amount of flow. Each arc $(i, j) \in A$ also has a capacity u_{ij} denoting the maximum amount that can flow on the arc, a lower bound l_{ij} the minimum amount that must flow on the arc, and the gain parameter δ_{ij} represents

gains or losses.
The decision variables are: f_{ij} = the amount of flow through arc $i \to j$

Objective Function:

$$Minimize\ Z(f) = \sum_{(i,j)\in A} c_{ij} f_{ij} \tag{1.6}$$

Subject to:
1. Conservation of flow:
The total outflow minus inflow of the node must be equal to the mass balance (supply or demand values b_i) for every node $i \in N$. Thus using NB as the set of arcs terminating at node i and NA as the set of arcs originating from node i, the flow balance constraints become:

$$\Rightarrow \sum_{(j\in NA(i))} f_{ij} - \sum_{(j\in NB(i))} \delta_{ji} f_{ji} = b_i, \quad where\ \delta_{ji}\ is\ a\ gain. \tag{1.7}$$

The value of b_i depends on the nature of node i, where

- If $b_i < 0$, node i is a sink (or demand) node.
- If $b_i > 0$, node i is a source (or supply) node.
- If $b_i = 0$, node i is a transshipment node.

2. Lower bound and Upper bounds on flow:
This constraint is the flow bound constraint which imposed physical capacity or restrictions on the flow range. That is:

$$l_{ij} \le f_{ij} \le u_{ij} \tag{1.8}$$

In general, the mathematical formulation of a generalized minimum cost network flow problem has the following form:

$$MinimizeZ(f) = \sum_{(i,j)\in A} c_{ij} f_{ij}$$

$$subjectto: \sum_{j\in NA(i)} f_{ij} - \sum_{j\in NB(i)} \delta_{ji} f_{ji} = b_i, \forall i \in N \tag{1.9}$$

$$0 \le f_{ij} \le u_{ij}, \forall (i,j) \in A$$

1.3.4 Pure Minimum Cost Flow Problem

A pure minimum cost flow problem is a pure network model which is defined by a given set of arcs and a given set of nodes, where each arc has a known capacity, unit cost and each node has a fixed external flow.

Mathematical Programming Formulation

The mathematical formulation of pure minimum cost flow problem is the same as to the formulation of generalized minimum cost flow problem except we associate the gain parameter $\delta_{ji} = 1$ and with each node $i \in N$ an integer $b(i)$ to represent its supply or demand. This will guarantee to have integer solutions.

Thus in general, the mathematical formulation of a pure minimum cost network flow problem has the following form:

$$\begin{aligned} &MinimizeZ(f) = \sum_{(i,j)\in A} c_{ij} f_{ij} \\ &subjectto: \sum_{j\in NA(i)} f_{ij} - \sum_{j\in NB(i)} f_{ji} = b_i,\ \forall i \in N \\ &0 \le f_{ij} \le u_{ij},\ \forall (i,j) \in A \end{aligned} \tag{1.10}$$

Definition:

Let $G = (N, A)$ be a directed network with n nodes and m arcs. Let E be the $n \times m$ matrix that has a row associated with each node $i \in N$ in G and a column associated with each arc $e_i \in A$ in G. E is a node-arc incidence matrix if the elements a_{ij} in matrix E has the value as follows:

$$a_{ij} = \begin{cases} -1,\ if\ arc\ e_i\ leaves\ node\ i; \\ +1,\ if\ arc\ e_i\ enters\ node\ i; \\ 0,\ otherwise. \end{cases} \tag{1.11}$$

1.3.5 Augmented Tree and Augmented Forest

Augmented Tree

Let $G^\alpha = (N^\alpha, T^\alpha)$ be a subgraph of $G = (N, A)$ so that $N^\alpha \subseteq N$ and $T^\alpha \subseteq A$. We refer to G^α as an augmented tree if T^α is a spanning tree of the node set N^α together with an extra edge $e = (a, b)$ which is called the extra arc. An augmented tree has a designated node, called its root. Each augmented tree is hanging from its root.

An augmented tree contains exactly one cycle which is formed by adding the

extra arc (a, b) to the tree $T^{\alpha} - (a, b)$; we refer to this cycle as the extra cycle.

Augmented Forest
An augmented forest $G^f = (N, F)$ with $F \subseteq A$ is a collection of node disjoint augmented trees that span all the nodes of the graph.
An augmented forest is a good augmented forest if each of its components is a good augmented tree. We refer to an augmented tree as a good augmented tree if its extra cycle is lossy or gainy (i.e. not a breakeven cycle)

Augmented forest structure
Suppose that the sets F, L and U define a partition of the arc set A and that F is a good augmented forest, L is a non augmented forest arc at their lower bounds, and U is a non augmented forest arc at their upper bounds. So the triple (F, L, U) is an augmented forest structure.

Feasibility Condition
An augmented forest structure (F, L, U) is feasible if we set $f_{ij} = 0$ for all $(i, j) \in L$ and $f_{ij} = u_{ij}$ for all $(i, j) \in U$, and the flow f_{ij} on the augmented forest arcs satisfy the generalized flow definition (or mass balance constraints and flow bounds). That is, a unique flow (f_{ij})on the arcs of the augmented forest will satisfy the equation:

$$\sum_{j \in NA(i)} f_{ij} - \sum_{j \in NB(i)} \delta_{ji} f_{ji} = b_i, \quad for\ all\ i \in N$$

That is, if the flow f_{ij} satisfies the lower and upper bound constraints imposed on all the arcs of the augmented forest, then we say that the structure (F, L, U) is feasible; otherwise it is infeasible.
We say that a feasible augmented forest structure is non degenerate if $0 < f_{ij} < u_{ij}\ \forall (i, j) \in F$; it is degenerate otherwise.

Optimality Condition
We also say that a feasible augmented forest structure (F, L, U) is an optimal augmented forest structure if its associated flow f_{ij} is an optimal solution of the generalized flow problem.

Node Potentials
We associate with each vertex i a node potential function $\pi(i) : N \to \mathbb{R}$. The potential of a node can be interpreted as the dual variable corresponding to the flow conservation constraints in the linear programming formulation of the problem.

Reduced Costs
The *reduced cost* of arc (i, j) with respect to a set of node potentials is then defined as

$$c_{ij}^{\pi} := c_{ij} - \pi(i) + \delta_{ij}\pi(j) \tag{1.12}$$

Theorem 1.1. $(Sufficient Condition : Generalized Flow Optimality Condition)$ A flow f^* is an optimal solution of the generalized minimum cost flow problem if it is feasible and for some vector π of node potentials, the pair (f^*, π) satisfies the following optimality conditions.

- If $0 < f_{ij}^* < u_{ij}$, then $c_{ij}^{\pi} = 0$.
- If $f_{ij}^* = 0$, then $c_{ij}^{\pi} \geq 0$.
- If $f_{ij}^* = u_{ij}$, then $c_{ij}^{\pi} \leq 0$.

Proof:
We first claim that minimizing

$$\sum_{(i,j)\in A} c_{ij} f_{ij}$$

is equivalent to minimizing

$$\sum_{(i,j)\in A} c_{ij}^{\pi} f_{ij}$$

. Let π be a vector that together with the flow f^* satisfies the conditions given above and let f be any arbitrary flow. Consider the following summation:

$$\sum_{(i,j)\in A} c_{ij}^{\pi}(f_{ij} - f_{ij}^*)$$

We claim that each term in the above summation is non negative. We establish this claim by considering three cases.

Case 1:
$0 < f_{ij}^* < u_{ij}$. In this case $c_{ij}^{\pi} = 0$, so the term $c_{ij}^{\pi}(f_{ij} - f_{ij}^*) = 0$

Case 2:
$f_{ij}^* = 0$.In this case $f_{ij} \geq f_{ij}^* = 0$,and $c_{ij}^{\pi} \geq 0$, so the term $c_{ij}^{\pi}(f_{ij} - f_{ij}^*) \geq 0$.

Case 3:
$f_{ij}^* = u_{ij}$.In this case,$f_{ij} \leq f_{ij}^* = u_{ij}$,and $c_{ij}^{\pi} \leq 0$, so the term $c_{ij}^{\pi}(f_{ij} - f_{ij}^*) \geq 0$.

We have shown that $c^\pi(f - f^*) = c^\pi f - c^\pi f^* \geq 0$,or$c^\pi f^* \leq c^\pi f$, which concludes that some feasible flow f^* is optimal.

Property 1.2:(Augmented Forest Structure Optimality Conditions) A feasible augmented forest structure (F, L, U) with the associated flow f^* is an optimal augmented forest structure if for some function $\pi : N \to \mathbb{R}$ of node potentials, the pair (π, f^*) satisfies the following optimality conditions:

- $c_{ij}^\pi = 0$, for all $(i, j) \in F$ (because $0 < f_{ij}^* < u_{ij}$)
- $c_{ij}^\pi \geq 0$, for all $(i, j) \in L$ (because $0 = f_{ij}^*$)
- $c_{ij}^\pi \leq 0$, for all $(i, j) \in U$ (because $f_{ij}^* = u_{ij}$)

Duality for a Generalized Network Flow Problem

The relationship between $\delta_{ij} f_{ij}$ and $\delta_{ij}\pi(j)$ can be viewed by the dual of generalized network flow problem with costs:

$$max \quad \sum_i b(i)\pi(i)$$

$$subjectto : \forall_{ij} : c_{ij} - \pi(i) + \delta_{ij}\pi(j) \geq 0 \tag{1.13}$$

Determining Flows For an Augmented Forest Structure

Let (F, L, U) be an augmented forest structure of the generalized minimum cost flow problem. The augmented forest F contains several augmented trees. The following Compute-Potentials algorithm is performed per tree.
Let $T \cup \{(a, b)\}$ be some augmented tree, and let node h be it's root.

begin
$\pi(h) := \theta$;
$j := thread(h)$;
While $j \neq h$ do
begin
$i := pred(j)$;
if $(i, j) \in A$, then $\pi(j) := \frac{(\pi(i) - c_{ij})}{\delta_{ij}}$;
if $(j, i) \in A$, then $\pi(j) := \delta_{ij}\pi(i) + c_{ij}$;
$j := thread(i)$
end;

for each node i, let the potential $\pi(i)$, as a function of θ, be represented by $f(i) + g(i)\theta$ for some constants $f(i)$ and $g(i)$;
Compute $\theta := \frac{((c_{ab} - f(a) + \delta_{ab} f(b)))}{((g(a) - \delta_{ab} g(b))}$; substitute this value of θ in the expression $\pi(i) = f(i) + g(i)\theta$ to compute the potentials for each node.
end;

Here, we would like to satisfy the condition that $c_{ij}^{\pi} = 0,\ \forall (i,j) \in F$, so we use the following equation :
$c_{ij}^{\pi} = c_{ij} - \pi(i) + \delta_{ij}\pi(j) = 0.$
We use the extra arc (a,b) to determine θ, and update the potentials(they are linear functions of θ).
Remark: In the algorithm above,

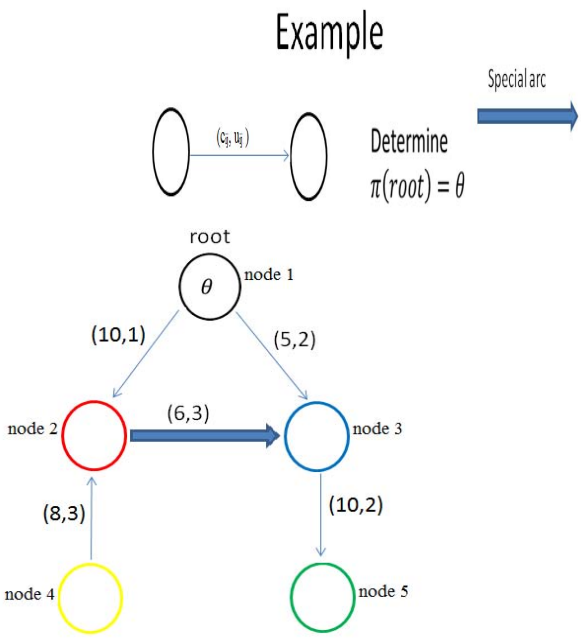
- the predecessor (pred) index refers to the first node in that path (other than node i), since each node i has a unique path connecting it to the root.

- the thread indices define a traversal of a tree, that is a sequence of nodes that walks or threads its way through the nodes of a tree, starting at the root node, and visiting nodes in a "$top-to-bottom$" order, and finally returning to the root node.
  For instance, consider the traversal $1-2-5-6-8-9-7-3-4-1$, so thread(1) =2, thread(2)=5, thread(5)=6, and so on.

In the capacitated flow problem, we start with $f_{ij} = 0$ and $e(i) = b(i)$, then we execute the following statements for every $(i,j) \in U$ do

begin
$f_{ij} := u_{ij}$; :setting the required flow.
$e(i) := e(i) - u_{ij}$; :set deficit.
$e(j) := e(j) + \delta_{ij}u_{ij}$; :set excess.
end;

The purpose of this statement is to set the flow on each arc $(i,j) \in U$ to its upper bound, which creates an additional deficit of u_{ij} units at node i and an additional excess of $\delta_{ij}u_{ij}$ at node j.
The procedure compute flows provides us with a method for determining a flow that satisfies the supply or demand constraints of all the nodes in an augmented tree. Applying this procedure for each augmented tree, we obtain a flow that satisfies the supply or demand constraints of all the nodes. If this flow satisfies the flow bounds on the arc flows, then the augmented forest structure is feasible; otherwise it is infeasible.



Solution

1. Select the special arc be (2,3), going to node 2.
2. $c_{12} - \pi(1) + \delta_{12}\pi(2) = 0$
$\Rightarrow 10 - \theta + 1.\pi(2) = 0$
$\Rightarrow \pi(2) = \theta - 10$
3. $c_{42} - \pi(4) + \delta_{42}\pi(2) = 0$
$\Rightarrow 8 - \pi(4) + 3.(\theta - 10) = 0$
$\Rightarrow \pi(4) = 3\theta - 22$
4. $c_{13} - \pi(1) + \delta_{13}\pi(3) = 0$
$\Rightarrow 5 - \theta + 2.\pi(3) = 0$
$\Rightarrow \pi(3) = \frac{\theta}{2} - 2.5$
5. $c_{35} - \pi(3) + \delta_{35}\pi(5) = 0$
$\Rightarrow 10 - \frac{\theta}{2} + 2.5 + 2.\pi(5) = 0$
$\Rightarrow \pi(5) = \frac{\theta}{4} - 6.25$

By using the special arc (2,3), we get
$c_{23} - \pi(2) + \delta_{23}\pi(3) = 0$
$\Rightarrow 6 - (\theta - 10) + 3(\frac{\theta}{2} - 2.5) = 0$
$\Rightarrow \theta = -17$

This implies that the amount of node potentials at each node is:
$\pi(1) = -17, \pi(2) = -27, \pi(3) = -11, \pi(4) = -73 and \pi(5) = -10.5.$
This values helps us to determine the flow that satisfy the supply or demand

constraints of all nodes.
The definition of reduced cost implies that this node potentials reduces the reduced cost of each unit of flow leaving node, say k, by $\pi(k)$ and increases the reduced cost of each flow unit entering node, say k, by $\pi(k)$.
Time complexity:O(n)
We can compute the time complexity for the algorithm by computing the per- iteration complexity and the total number of iteration needed before the algorithm terminates.
Remark
The flow on the arc of the augmented forest that satisfy the mass balance constraint is unique, so the forest structure is feasible if and only if the flow bounds are satisfied.

Existence of an Optimal Solution
(Properties from Linear Programming)

- Arcs in a set$B$ have a unique solution if and only if B is a good augmented forest.
- A set B of arcs defines a basis of the generalized network problem if and only if B is a good augmented forest.

This means, there exist an optimal solution for a good augmented forest.

Chapter 2

Transportation Problem With Lossy Arcs

In the above chapter we have tried to give the brief discussion about network flow problems such as: generalized network flow problems and pure network flow problems and their solution feasibility, and the optimality conditions. Now in this chapter we try to give a brief explanation and mathematical formulations about two different types of transportation problems, namely the ordinary transportation problem and the transportation problem with lossy arcs.
The ordinary transportation problem is the special case of minimum cost flow problem, where as the transportation problem is the special case of the generalized minimum cost flow problem.
Thus, transportation problem is a subclass of the generalized network flow problem, which means it is the special case of the generalized minimum cost flow problem and the following are their major difference:

- In the transportation problem there is no a transshipment node and all arcs are directed from source nodes to sink nodes, where as in a generalized network flow problem we may have a transshipment nodes, backward arcs(or cycle)and it is possible to have arcs between source or sink nodes..

- In the transportation problem we must have at least two source nodes and two sink nodes, where as in the generalized network flow problem we may have a single source node and a single sink node.

2.1 Ordinary Transportation Problem

Transportation problem is a special class of the linear programming problem. It is concerned with finding an optimal distribution plan for a single commodity. A given supply of the commodity available at a number of sources, there is a specified demand for the commodity at each of a number of destinations, and the transportation cost between each source destination pair is known. In the simplest case the unit transportation cost is constant.
The problem is to find the optimal distribution plan for transporting the products from source to destinations that minimizes the total transportation cost to satisfy the demands at the destination node. Also, transportation problem is defined as a special case of the minimum cost flow problem with the property that the node set N is partitioned into two subsets N_1 and N_2(of possibly unequal cardinality) so that:

- each node in $N_1$is a supply node,
- each node in N_2 is a demand node, and
- for each arc (i, j) in A, $i \in N_1$ and $j \in N_2$.

The classical example of this problem is the distribution of goods from warehouses to customers. In this context the nodes in N_1 represent the warehouses, the nodes in N_2 represent customers (or, more typically, customer zones), and an arc (i, j) in A represents a distribution channel from warehouse i to customer j.
The transportation model can be portrayed in a tabular form by means of a transportation table shown below:

Origin	1	2	. . .	n	Supply(a_i)
1	$\frac{f_{11}}{c_{11}}$	$\frac{f_{12}}{c_{12}}$	. . .	$\frac{f_{1n}}{c_{1n}}$	a_1
2	$\frac{f_{21}}{c_{21}}$	$\frac{f_{22}}{c_{22}}$	. . .	$\frac{f_{2n}}{c_{2n}}$	a_2
.	. . .	. . .	. . .	. . .	. . .
m	$\frac{f_{m1}}{c_{m1}}$	$\frac{f_{m2}}{c_{m2}}$	. . .	$\frac{f_{mn}}{c_{mn}}$	a_m
Demand	b_1	b_2	. . .	b_n	$\sum a_i = \sum b_i$

Table: 2.1:Transportation tableau

This can be seen in the following figure:

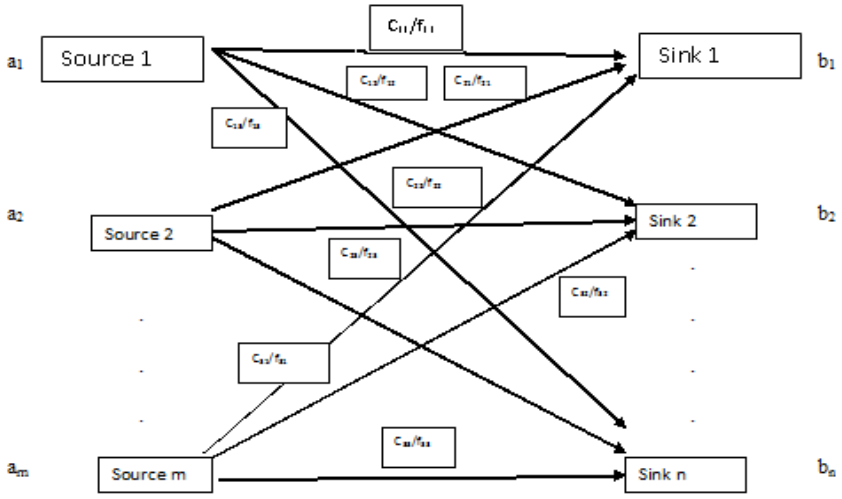


Figure: 2.1. Diagram for Transportation Problem

Here source indicates the place from where transportation will begin, destinations indicates the place where the product has to be arrived and c_{ij}indicates the transportation cost and f_{ij} indicates the arc flows.

2.1.1 Formulation of Transportation Problem

Let us consider the m-supply locations (origins) as S_1,S_2,. . ., S_m and the n-demand locations (destination) as D_1 ,D_2 , ..,D_n respectively. Let $a_i \geq 0$, $i = 1, 2, ..., m$, be the amount available at the i^{th} plant S_i . Let the amount required at the j^{th} shop D_j be $b_j \geq 0$, $j = 1, 2, ..., n$.

$$Minimize \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} f_{ij}$$

$$Subjectto : \sum_{j=1}^{n} f_{ij} = a_i \quad ; i = 1, 2, ..., m \tag{2.1}$$

$$\sum_{i=1}^{m} f_{ij} = b_j \quad ; j = 1, 2, ..., n$$

$$f_{ij} \geq 0, \quad \forall (i, j) \in A.$$

Let the cost of transporting one unit of flow form i^{th} origin to j^{th} destination be c_{ij} , $i = 1, 2, ..., m$, $j = 1, 2, ..., n$. If $f_{ij} \geq 0$ be the amount of flow to be transported from i^{th} origin to j^{th} destination , then the problem is to determine f_{ij} so as to minimize $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} f_{ij}$.

Definition:

In a transportation table, an ordered set of four or more cells is said to form a loop if:

- Any two adjacent cells in the ordered set lie in the same row or in the same column.

- Any three or more adjacent cells in the ordered set do not lie in the same row or in the same column.

Feasible solution:

A feasible solution to a transportation problem is basic if and only if the corresponding cells in the transportation table do not contain a loop.

Degeneracy in Transportation Problem

Transportation with m-origins and n-destinations can have $m+n-1$ positive basic variables, otherwise the basic solution degenerates. So whenever the number of basic cells is less than $m + n - 1$, the transportation problem is degenerate. To resolve the degeneracy, the positive variables are augmented by as many zero-valued variables as is necessary to complete $m+n-1$ basic variable.

Balanced and Unbalanced Transportation Problems

A transportation problem is said to be balanced if the total supply from all sources equals the total demand in the destinations and is called unbalanced otherwise.

Thus for a balanced problem, we have $\sum a_i = \sum b_i$ and for an unbalanced problems$\sum a_i \neq \sum b_i$. The case where $\sum a_i > \sum b_i$ and the arcs are lossy is the main focus of this project.

Basic Feasible Solution of a Transportation Problem

Since a transportation problem is a special case of linear programming problem a basic feasible solution of a transportation problem has the same definition as in for a linear programming problem. However, we observe that in the case of a transportation problem there are only $m+n1$ basic variables out of $m \times n$ unknown, due to reducing in the constraint of the transportation problem.

Consider the first $m+n-1$ constraints of the transportation problem:
$\sum_{i=1}^{m} f_{ij}=b_j$ and $\sum_{j=1}^{n} f_{ij}=a_i$, $i = 1, 2, ..., m$ and $j = 1, 2, ..., n$

Taking summation on both sides, this yields:

$$\sum_{j=1}^{n}\sum_{i=1}^{m} f_{ij} = \sum_{j=1}^{n} b_j \quad and$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \sum_{i=1}^{m} a_i$$

Theorem: 2.1. (Existence of Feasible Solution)

$$\sum a_i = \sum b_i \tag{2.2}$$

is a necessary and sufficient condition for the existence of a feasible solution to the transportation problem.

Proof: i) The condition is necessary

Let there exist a feasible solution to the transportation problem.

$$Minimize \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} f_{ij}$$

$$Subjectto: \sum_{j=1}^{n} f_{ij} = a_i \quad ; i = 1, 2, ..., m$$

$$\sum_{i=1}^{m} f_{ij} = b_j \quad ; j = 1, 2, ..., n$$

$$f_{ij} \geq 0, \quad \forall (i, j) \in A.$$

Then we have

$$\sum_{j=1}^{n}\sum_{i=1}^{m} f_{ij} = \sum_{j=1}^{n} b_j \quad and \quad \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \sum_{i=1}^{m} a_i$$

$$\Rightarrow \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

ii) The condition is sufficient
Let us assume that

$$\textstyle\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = \lambda$$

We assert that there exists a feasible solution given by $f_{ij}=\frac{a_i b_j}{\lambda}$ for all i and j. Clearly$f_{ij} \geq 0$, since $a_i > 0$,$b_j > 0$ for all i and j.
Also,

$$\sum_{j=1}^{n} f_{ij} = \sum_{j=1}^{n} \frac{a_i b_j}{\lambda} = \frac{a_i}{\lambda} \sum_{j=1}^{n} b_j = a_i, i = 1, 2, ..., m; and$$

$$\sum_{j=1}^{m} f_{ij} = \sum_{j=1}^{m} \frac{a_i b_j}{\lambda} = \frac{b_j}{\lambda} \sum_{j=1}^{m} b_j = b_j, j = 1, 2, ..., n.$$

Thus f_{ij} satisfies all the constraints of the transportation problem hence is a feasible solution.

2.2 Transportation problem with lossy arcs

The generalized transportation problem arises in many real life applications and it is a special case of the generalized minimum cost flow problems. Transportation problem with lossy arcs is a subclass of the Generalized Transportation problems which sends flows as possible from the source node to the sink node with a minimum transportation cost and satisfying the demand at the destination by minimizing losses in the arcs.

Let us start with the ordinary generalized transportation problem. A uniform good is transported from m supply points to n destination points. On the way, the amount of a good changes, i.e, the amount delivered to demand point j from supply point i is equal to $\delta_{ij}\ f_{ij}$ where f_{ij} is the amount of the good that leaves supply point i and δ_{ij} is the respective reduction ratio. That is,$0 < \delta_{ij} < 1$, which implies that the arcs are lossy.

In this case, the transportation problem is known as a lossy transportation problem. The unit transportation cost c_{ij} are constants (changes linearly

as the amount of flow), the demand b_j of each demand point j has to be satisfied and the supply a_i of each supply point i cannot be exceeded.
Thus, the model for the ordinary generalized TP has the following form:

$$\begin{aligned} &Minimize Z(f) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} f_{ij} \\ &Subject to : \sum_{i=1}^{m} \delta_{ji} f_{ji} = b_j, j = 1, 2, ..., n; \\ &\sum_{j=1}^{n} f_{ij} \leq a_i, i = 1, 2, ..., m. \\ &0 \leq f_{ij} \leq u_{ij}, \forall_{(i,j)} \in A. \end{aligned} \tag{2.3}$$

where u_{ij} is the flow upper bound on each arc.
Observe that this is very similar to the ordinary transportation problem. The difference appears in the first constraint, where the reduction ratios are included.

2.2.1 Formulation of The TPWLA

In reality when the arcs are losing a fraction of their flow due to evaporation, leakage, theft,taxes and so on, the actual amount of flow delivered at the destination points may not satisfies the demand requirements. In such a cases the transportation problem is formulated as:

$$\begin{aligned} &Minimize Z(f) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} f_{ij} \\ &Subject to : \sum_{i=1}^{m} \delta_{ji} f_{ji} \leq b_j, j = 1, 2, ..., n. \\ &\sum_{j=1}^{n} f_{ij} \leq a_i, i = 1, 2, ..., m. \\ &0 \leq f_{ij} \leq u_{ij}, \forall_{(i,j)} \in A. \end{aligned} \tag{2.4}$$

which indicates that the amount of flow delivered at the sink node does not satisfied the demand requirements. Such model is a formulation for Transportation Problem with lossy arcs when total supply exceeds the total demand but it does not met the demand.
But, if $f_{ij} = 0$ the flow that reaches to the destination point $\sum_{i=1}^{m} \delta_{ji} f_{ji} =$

$0 \neq b_j$ unless $b_j = 0$,then it is an optimal solution and we have nothing to do. Due to this reason the formulation for TPWLA must be:

$$
\begin{aligned}
&MinimizeZ(f) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}f_{ij} \\
&Subjectto : \sum_{i=1}^{m} \delta_{ji}f_{ji} = b_j, j = 1, 2, ..., n. \\
&\sum_{j=1}^{n} f_{ij} \leq a_i, i = 1, 2, ..., m. \\
&0 \leq f_{ij} \leq u_{ij}, \forall_{(i,j)} \in A.
\end{aligned} \tag{2.5}
$$

2.3 Some Transformations

2.3.1. Ordinary TP as a MCFP

The TP is based on a bi-partite graph, where N_1 is the set of source node, and N_2 is the set od demand nodes, such that $N = N_1 \bigcup N_2$, and the set of arcs is defined by $A = (i, j)|i \in N_1 and j \in N_2$.

The objective is to determine the least cost shipping plan from the source to destination. In linear Programming the problem is formulated as a minimum cost flow problem by:

$$Minimize\ Z(f) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}f_{ij}$$

Subject to:

$$
\begin{aligned}
&\sum_{j=1}^{n} f_{ij} = b_j \quad for\ i \in N_1 \\
&-\sum_{i=1}^{m} f_{ji} = -b_i \quad for\ j \in N_2 \\
&0 \leq f_{ij} \leq u_{ij}, \quad \forall_{(i,j)} \in A.
\end{aligned} \tag{2.6}
$$

2.3.2.TPWLA as a GMCFP

The Transportation problem with lossy arcs like the ordinary transportation problem is based on a bi-partite graph, where N_1 is the set of source nodes, and N_2 is the set of sink or destination nodes, such that N = N_1 U N_2 and the set of arcs is defined by $A = (i, j)|i \in N_1 and j \in N_2$, and the arcs are

lossy, that means the flow f_{ij} from the source node is not equal to the flow entering to the sink node, it is reduced by a certain reduction factor δ_{ij} and the amount $\delta_{ij} f_{ij}$ reaches to the sink node.
Thus, in Linear Programming (LP) the formulation of the problem as a generalized minimum cost flow problem is as follows:

$$MinimizeZ(f) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} f_{ij}$$

Subject to:

$$\begin{aligned} -\sum_{i=1}^{m} \delta_{ji} f_{ji} &= -b_i \quad for\ j \in N_2 \\ \sum_{j=1}^{n} f_{ij} &= b_i \quad for\ i \in N_1 \\ 0 \leq f_{ij} &\leq u_{ij}, \quad \forall_{(i,j)} \in A \end{aligned} \tag{2.7}$$

Chapter 3

Solution Procedure for TPWLA

In this chapter we try to give the general solution method to solve a transportation problem with lossy arcs. As we have discussed in the above chapter that a transportation problem with lossy arcs is a special cases of the generalized minimum cost flow problems, and we transformed it to a GMCFP. Since any special cases of linear programming problems can be solved by the solution method for the generalized flow problem, a Transportation problem with lossy arcs also can be solved by using the the simplex algorithms for generalized minimum cost flow problems.

3.1 The Generalized Network Simplex Algorithm(GNSA) for solving TPWLA

The algorithm maintain a (good) feasible augmented forest structure at every iteration (which, in linear programming terminology, is a feasible basis structure) and by performing a pivot operation transforms this solution in to an improved (good) augmented forest structure. The algorithm repeats this process until the augmented forest structure satisfies the optimality conditions.

Algorithm for Generalized network simplex method

Step 1:
Determine an initial feasible augmented forest structure (F, L, U). Let f be the flow and π be the node potentials associated with the initial augmented forest structure.

Step 2:
Check for optimality, if not determine the entering arc (k, l) violating its optimality condition; add arc (k, l) to an augmented forest.
Step 3: Determine the leaving arc (p, q).
Step 4: Update the solutions f (flows) and π(node potentials) and the augmented forest structure.

3.2 Discussion For Various Steps of GNSA

3.2.1. Obtaining an Initial Augmented Forest Structure
In order to obtain an initial augmented forest structure first we introduce an artificial arc (i, i) of sufficiently large cost M and infinite capacity.

- For a supply node i ($i.e$: $b(i) > 0$), because the cycle consisting of the arc (i, i) is a lossy cycle, we can consume the supply of node i, by sending flow along this cycle.
- For a demand node i ($i.e$: $b(i) < 0$), the cycle (i, i) is a gainy cycle and by augmenting flow along the cycle we can generate sufficient flow to satisfy the demand node i.

By setting the flow along arc (i, i) to be f_{ii}=$\frac{e(i)}{(1-\delta_{ii})}$,we can satisfy the supply or demand of the node i. We determine the initial node potentials from the fact that the reduced cost of each arc in F must be zero. Using c^{π}_{ii}=c_{ii}-$\pi(i)$ + $\delta_{ii}\pi(i)$ for each node $i \in N$ yields $\pi(i)$=$\frac{M}{(1-\delta_{ii})}$, where M is the large cost for an artificial arc (i, i).
3.2.2. Optimality Testing and Entering arc
Let π be the corresponding node potential. To determine whether a spanning tree structure is optimal , we check the following optimality conditions:

$$c^{\pi}_{ij} \geq 0, \text{ for every arc } (i, j) \in L$$
$$c^{\pi}_{ij} \leq 0, \text{ for every arc } (i, j) \in U$$

If the spanning tree structure satisfies the conditions ,it is optimal and we terminate the operation. Otherwise, the algorithm selects a violating arcs (non augmented forest arcs) to enter the forest.
Identifying The Entering Arc
Two types of arcs are eligible to enter the augmented forest:

- Any arc $(i, j) \in L$ with $c^{\pi}_{ij} < 0$.

- Any arc $(i, j) \in U$ with $c_{ij}^{\pi} > 0$.

For any eligible arc (i, j), we refer to $|c_{ij}^{\pi}|$ as its violation.
The generalized network simplex algorithm can select any eligible arc as the entering arc to enter the tree and still would terminate finitely(with some provisions for dealing with degeneracy).

Pivot Rules
For selecting the entering arc, the following pivot rules for minimum cost flow problem are also applied for the generalized network simplex algorithms:

- Dantzigs pivot rule, which selects the arc with maximum violation as the entering arc.
- The first eligible arc pivot rule, which selects, in a wraparound fashion, the first arc with positive violation encountered in examining the arc list, and
- The candidate list pivot rule, which maintains a candidate list of arcs with positive violation and selects the arc with maximum violation from the candidate list as the entering arc.

3.2.3. Identifying The Leaving Arc

Suppose that we select arc (k, l) as the entering arc that belongs to the set L (lower bound). Denote y_{ij} as the change in flow units for $(i, j) \in F$ if we use 1 unit of flow on (k, l). If we know the values $y_{ij}, \forall (i, j) \in F$, then it would be easy to know the maximum number of flow we can use on (k, l), so all edges will remain between their lower and upper bound. This promise we have an arc that reached its lower and higher bound-this is the arc we drop. Setting the flow on arc (k, l) to value 1 creates a deficit (or, demand) of 1 unit at node k and excess (or, supply) of δ_{kl} unit at node l.

To determine y_{ij} we follow the following:

- Use the procedure compute flow on the augmented trees containing nodes k and l;
- At first the flow $f_{ij} = 0$ for all edges,and the imbalance vector is:

$$e(i) = \begin{cases} -1, \ for\ i = k; \\ \delta_{kl}, \ for\ i = l; \\ 0, \ for\ i \neq k\ and\ l. \end{cases} \tag{3.1}$$

The arc flow we obtain are the y_{ij} values.

Determine the Maximum Flow on (k, l) within the Bounds

Having determined the y_{ij} values, we can easily compute the maximum additional flow μ on the entering arc (k, l).

- If we add μ units of flow to (k, l), then to keep the flow bounds it requires that:

$$0 \leq f_{ij} + \mu y_{ij} \leq u_{ij}, \forall_{(i,j)} \in F \cup (k, l). \tag{3.2}$$

- If $y_{ij} > 0$ for some arc (i, j), then the flow on the arc increases with μ and will eventually reach the arcs upper bound.

- If $y_{ij} < 0$ for some arc (i, j), then the flow on the arc decreases with μ and will eventually reach the arcs lower bound.

Therefore, if μ_{ij} denotes the maximum possible increase in μ allowed by arc (i, j), then

$$\mu_{ij} = \begin{cases} \frac{(u_{ij} - f_{ij})}{y_{ij}}, \ if \ y_{ij} > 0; \\ \frac{f_{ij}}{(-y_{ij})}, \ if \ y_{ij} < 0; \\ \infty, \ if \ y_{ij} = 0. \end{cases} \tag{3.3}$$

So the largest value of μ for which $f + \mu y$ is feasible is:

$$\mu = \min\{\mu_{ij} : (i, j) \in F \cup (k, l)\} \tag{3.4}$$

Now, augment μ units of flow on the arc (k, l) and change the flow on the augmented forest arcs to $f_{ij} + \mu y_{ij}$.

Define μ, for any arc (i, j) by $\mu = \mu_{ij}$ as blocking arc.Select any blocking arc, say (p, q), as the leaving arc and $y_{pq} \neq 0$. We say that the iteration is non-degenerate if $\mu > 0$, and degenerate if $\mu = 0$. A degenerate iteration occurs only if F is a degenerate augmented forest.

3.2.4. Updating The Augmented Forest

When a leaving arc (p, q) is determined for a given entering arc (k, l) we update the tree structure (the augmented forest structure).

- If the leaving arc is the same as the entering arc, which would happen when $\mu = \mu_{kl} = u_{kl}$, the augmented forest does not change. In this case, the arc (k, l) merely moves from the set L to the set U and vice versa.

- If the leaving arc differs from the entering arc, the augmented forest changes and we need to update the sets F, L and U. Also we need to show that the modified set of arcs in F constitutes a good augmented forest. Since each basis in the generalized network flow problem is a good augmented forest, the generalized network simplex algorithm moves from one good augmented forest to another good augmented forest.

3.2.5. Updating Potentials and Tree Indices

After obtaining the new augmented forest structure, the next step is to update the node potentials. We can recompute the node potentials of the augmented tree(s) using the procedure compute potentials.

The final step in pivot operation is to update various tree indices. We could compute the tree indices from scratch, which would also require $O(n)$ time.

3.2.6. Termination

The generalized network simplex algorithm moves from one feasible augmented forest structure to another until it obtains a structure that satisfies the optimality conditions.

In a pivot operation, the objective function value decreases by the amount μ $|c_{kl}^{\pi}|$ at every iteration.

- If each pivot operation in the algorithm is non-degenerate ($i.e, \mu > 0$), each subsequent augmented forest structure has a smaller cost. Since any network has a finite number of augmented forest structures, and each augmented forest structure has a unique associated cost(which decreases with every iteration),the generalized network simplex algorithm terminates finitely.

- If it is a degenerate pivot, the algorithm might not terminate finitely unless we perform pivots carefully.

Complexity

The worst-case complexity of the generalized network simplex algorithm is the product of the number of iterations and the complexity per-iteration. Although in the worst-case we cannot bound the number of iterations by any polynomial function of n and m. The number of iterations the algorithm performs is generally a low-order polynomial in n and m. And what is the running time per iteration?

The algorithm requires $O(m)$ time to identify the entering arc.

Each of the other operations, such as computing the y_{ij} values, updating the

flows, the potentials, and the tree indices, selecting the leaving arcs,requires $O(n)$ time. The average time per pivot is much less than n. Thus, its computational time grows slower than $O(nm)$, even though its worst-case running time is exponential. In general, the running time for a GNSA is two or three times slower than the simplex algorithm for a minimum cost flow problem.

3.2.7. Dealing With Degenerate Pivots

- We define ε as an n vector whose elements are $(\alpha, \alpha, \alpha, ..., \alpha)$ for a sufficiently small real number .

- If we replace the supply or demand vector b by $b + \varepsilon$ it is possible to prove that the augmented forest structure at every iteration is not degenerate and therefore the algorithm will terminate finitely.

- Moreover, it is possible to show that an optimal augmented forest structure of the perturbed problem is also an optimal augmented forest structure of the original problem. Thus, the perturbation does not affect optimality of the solution.

Summary

The transportation problem with lossy arcs is a special application of the generalized minimum cost flow problem and it can be solved by generalized network simplex algorithm by transforming it to a generalized minimum cost flow problem, since the generalized network simplex algorithm is the fastest available algorithm for solving the generalized minimum cost flow problem.
The generalized minimum cost flow problem is a generalization of the minimum cost flow problem in the sense that arcs do not conserve flow. The generalized minimum cost flow problem is significantly more difficult to solve than the minimum cost flow problem.
The basis of the generalized minimum cost flow problem is a good augmented forest; this fact permits us to perform the steps of the simplex method without maintaining the simplex tableau.
The optimality condition for a good augmented forest are the same as those for the minimum cost flow problem, but with slightly different definition of the reduced cost c_{ij}^{π} of an arc (i, j), in this context, it is $c_{ij}^{\pi} = c_{ij} - \pi(i) + \delta_{ij}\pi(j)$.

Bibliography

1. AHUJA R.K., MAGNANTI T.L., ORLIN J.B., Network Flows. Theory, Algorithms and Applications, Prentice Hall, 1993.

2. BALAS E., IVANESCU P.L., On the Generalized Transportation Problem, Management Science, Vol. 11, No. 1, Series A, Sciences, (Sep., 1964), pp. 188-202.

3. BAZARAA M.S., JARVIS J.J., SHERALI H.D., Linear Programming and Network Flows, Fourth Edition, John Wiley and Sons Inc., 2010.

4. NAGURNEY A., YU M., MASOUMI A. H., NAGURNEY L. S., Networks Against Time. Supply Chain Analytics for Perishable Products, Springer Briefs in Optimization,Springer, 2013.