Interim Report of Project Related To Course

Algorithm
CS – 204

Submitted by :-

Gyanendra Tiwari
M.Sc. [Mathmatics]
Roll No: 01212305

Networks :- Networks are pervasive. They arise in numerous application setting and in many forms. Physical networks are perhaps the most common and the most readily

identifiable classes of networks; and among physical networks, transportation networks are perhaps the most visible in our everyday lives. Often, These networks model homogeneous facilities such as rail-beds or highways. But on other occasions, they corresponds to composite entities that model, for example, complex distribution and logistics decisions. The traditional operation research "transportation problem" is illustrative.

Physical networks are not limited to transportation setting; they also arise in several other disciplines of applied science and engineering, such as mathematics, chemistry and electrical, communications, mechanical and civil engineering. When physical network occur in these different disciplines, their nodes, arcs and flows model many different types of physical entities.

For example, in a typical communication network, nodes will represent telephone exchanges and transmission facilities, arcs will denote copper cables or fiber optic links and flow would signify the transmission of voice messages or of data.

Following figure shows some typical associations for the nodes, arcs and flows in a variety of physical networks.

Applications	**Physical analog of nodes**	**Physical analog of arcs**	**Flow**
Communication systems	Telephone exchange, computers, satellites, transmission facilities	Cables, fiber optic links, microwave relay links	Voice messages, data, video transmissions
Hydraulic systems	Pumping station, reservoirs, lakes	Pipelines	Water, gas, oil, hydraulic fluids
Integrated computer circuits	Gates, registers, processors	Wires	Electrical current
Mechanical systems	Joints	Rods, beams, springs	Heat, energy
Transportation systems	Intersection, rail yards, airports	Highways, airline routes, rail-beds	Passengers, freight vehicles, operators

Figure : 1 Ingredients of some common physical networks.

Here we will discuss some very important models of **Networks Flow** :

Minimum Cost Flow Problem :- The minimum cost flow model is the most fundamental of all network flow problems.The problem is easy to state: We wish to determine a least cost shipment of a commodity through a network in order to satisfy demands at certain nodes from available supplies at other nodes.In minimum cost flow problem a digraph G is given and capacities defined as $u : E(G) \rightarrow \mathbb{R}_+$, numbers $b : V(G) \rightarrow \mathbb{R}$ with $\Sigma_{v \in V(G)} b(v) = 0$, and weights $c : E(G) \rightarrow \mathbb{R}$. We have to find a b-flow f whose cost $c(f) := \Sigma_{e \in E(G)} f(e).c(e)$ is minimum (or decide that none exits).

Sometimes one also allows infinite capacities. In this case an instance can be unbounded.

Shortest Path Problem :- The shortest path problem is perhaps the simplest of all network flow problems. For this problem we wish to find a path of minimum cost (or length) from a specified *source node s* to another specified *sink node t,* assuming that each arc $(i,j) \in A$ has an associated cost (or length) c_{ij} . Some of the simplest applications of the shortest path problem are to determine a path between two specified nodes of a network that has minimum length, or a path that takes least time to traverse, or a path that has maximum reliability.

This basic model has applications in many different problem domains, such as equipment replacement, project scheduling, cash flow management, message routing in communications systems and traffic flow through congested cities. If we set $b(s) = 1$, $b(t) = -1$ and $b(i) = 0$ for all other nodes in minimum cost flow problem, the solution to the problem will send 1 unit of flow from node *s* to node *t* along the shortest path. The shortest path problem also models situations in which we wish to send flow from a single-source node to a single-sink node in an uncapacitated network. That is , if we wish to send *v* units of flow from node *s* to node *t* and the capacity of each arc of the network is at least *v*, we would send the flow along the shortest path from node *s* to node *t*. If we want to determine shortest paths from the source node *s* to every other node in the network, then in the minimum cost flow problem we set $b(s) = (n - 1)$ and $b(i) = -1$ for al other nodes. [We can set each arc capacity u_{ij} to any number larger than (n – 1).] The minimum cost flow solution would then send *v* unit flow from node *s* to every other node *i* along a shortest path.

Maximum Flow Problem :- The maximum flow problem is in a sense a complementary model to the shortest path problem. The shortest path problem models situations in which flow incurs cost but is not restricted by any capacities; in contrast, in the maximum flow problem flow incurs no costs but restricted by flow bounds. The maximum flow problem seeks a feasible solution that sends the maximum amount of flow from a specified source node *s* to another specified sink node *t*. If we interpret u_{ij} as the maximum flow rate of arc (i,j), the maximum flow problem identifies the maximum steady-state flow that the network can send from node *s* to node *t* per unit time.

Examples of the maximum flow problem include determining the maximum steady-state flow of (1) petroleum products in a pipeline network, (2) cars in a

road network, (3) messages in a telecommunication network and (4) electricity in a electrical network.

We can formulate this problem as a minimum cost flow problem in the following manner. We set $b(i) = 0$ for all $i \in \mathbb{N}$, $c_{ij} = 0$ for $(i,j) \in A$, and introduce an additional arc (t,s) with cost $c_{ts} = -1$ and flow bound $u_{ts} = \infty$.Then the minimum cost flow solution maximizes the flow on arc (t,s); but since any flow on arc (t,s) must travel from node s to node t through the arcs in A [since each $b(i) = 0$], the solution to the minimum cost flow problem will maximize the flow from node s to node t in the original network.

Assignment Problem :- The data of the assignment problem consist of two equally sized sets $\mathbb{N}_1$ and $\mathbb{N}_2$ (i.e., $|\mathbb{N}_1| = |\mathbb{N}_2|$), a collection of pairs A is a subset of $\mathbb{N}_1 \times \mathbb{N}_2$ representing possible assignments, and a cost c_{ij} associated with each element $(i, j) \in A$. In the assignment problem we wish to pair, at minimum possible cost , each object in $\mathbb{N}_1$ with exactly one object in $\mathbb{N}_2$.

Examples of the assignment problem include assigning people to projects, jobs to machines, tenants to apartments, swimmers to events in a swimming meet, and medical school graduates to available internships. The assignment problem is a minimum cost flow problem in a network $G = (\mathbb{N}_1 \cup \mathbb{N}_2, A)$ with $b(i) = 1$ for all $i \in \mathbb{N}_1$, $b(i) = -1$ for all $i \in \mathbb{N}_2$, and $u_{ij} = 1$ for all $(i, j) \in A$.

Transportation Problem :- The transportation problem is a special case of the minimum cost flow problem with the property that the node set $\mathbb{N}$ is partitioned into two subsets $\mathbb{N}_1$ and $\mathbb{N}_2$ (of possibly unequal cardinality) so that (1) each node in $\mathbb{N}_1$ is a supply node, (2) each node of $\mathbb{N}_2$ is demand node, and (3) for each arc (i, j) in A, $i \mathbb{N}_1$ and $j \in \mathbb{N}_2$.

The classical example of this problem is the distribution of goods from warehouses to customers. In this context the nodes in $\mathbb{N}_1$ represent the warehouses, the nodes in $\mathbb{N}_2$ represent customers (or, more typically, customer zones), and an arc (i, j) in A represents a distribution channel from warehouses i to customer j.

Circulation Problem :- The circulation problem is a minimum cost flow problem with only transshipment nodes; that is, $b(i) = 0$ for all $i \in \mathbb{N}$. In this instance we wish to find a feasible flow that honors the lower and upper bounds l_{ij} and u_{ij} imposed on the arc flows x_{ij}. Since we never introduce any exogenous flow into the network or extract any flow from it, all the flow circulates around the network. We wish to find the circulation that has the minimum cost.

The design of a routing schedule of a commercial airline provides one example of a circulation problem. In this setting, any airplane circulates the among the airports of various cities; the lower bound l_{ij} imposed on an arc (i, j) is 1 if the airline needs to provide service between cities i and j, and so must dispatch an airplane on his arc (actually, the nodes will represent a combination of both a physical location and a time of day so that an arc connects, for example, New York City at 8 A.M. with Boston at 9 A.M.).

Convex Cost Flow Problems :- In the minimum cost flow problem, we assume that the cost of the flow on any arc varies linearly with the amount of flow. Convex cost

flow problems have a most general cost structure; The cost is a convex function of the amount of flow.

Flow costs vary in convex manner in numerous problem settings, including (1) power losses in an electrical network due to resistance, (2) congestion costs in a city transportation network and (3) expansion costs of a communication network.

Generalized Flow problem :- In the minimum cost flow problem, arcs conserve flows (i.e., the flow entering an arc equals the flow leaving the arc). In generalized flow problems, arcs might "consume" or "generate" flow. If x_{ij} units of flow enter an arc *(i,j)*, then $\mu_{ij}\, x_{ij}$ units arrive at node *j*; μ_{ij} is a positive *multiplier* associated with arc. If $0 < \mu_{ij} < 1$, the arc is lossy, and if $1 < \mu_{ij} < \infty$, the arc is gainy. Generalized network flow problems arise in several application contexts.

Examples, (1) power transmission through electric lines, with power lost with distance traveled, (2) flow of water through pipelines or canals that lose water due to seepage or evaporation, (3) transportation of a perishable commodity and (4) cash management scenarios in which arcs represent investment opportunities and multipliers represent appreciation of an investment's value .

Multi-commodity Flow Problem :- The minimum cost flow problem models the flow of a single commodity over a network. Multi-commodity flow problems arise when several commodities use the same underlying network. The commodities may either be differentiated by their physical characteristics or simply by their origin-destination pairs. Different commodities have different origins and destinations, and commodities have separate mess balance constraints at each node. However, the sharing of the common arc capacities binds the different commodities together. In fact, the essential issue addressed by the multi-commodity flow problem is the allocation of the capacity of each arc to the individual commodities in a way that minimizes overall flow costs.

Multi-commodity flow problems arise in many practical situations, including (1) the transportation of passengers from different origins to different destination within a city; (2) the routing of non-homogeneous tankers (non-homogeneous in terms of speed, carrying capability and operating costs); (3) the worldwide shipment of different varieties of grains (such as corn, wheat, rice and soybeans) from countries that produce grains to those that consume it; and (4) the transmission of messages in a communication network between different origin-destination pairs.

Some Other Models :-

(i) Minimum Spanning Tree Problem
(ii) Matching Problem

The **MINIMUM COST FLOW PROBLEM** is quite general and has a couple of interesting special cases. The uncapacitated case ($u = \infty$) is sometimes called the transshipment problem. An even more restricted problem, also known as the transportation problem, has been formulated quite early by Hitchcock [1941] and others.

HITCHCOCK PROBLEM

Instance: A digraph G with V(G) = A *U* B and E(G) is a subset of A x B. Supplies $b(v) \geq 0$ for $v \in A$ and demands $-b(v) \geq 0$ for $v \in B$ with $\Sigma_{v \in V(G)}\, b(v) = 0$. Weights $c : E(G) \rightarrow \mathbb{R}$.

Task: Find a *b*-flow *f* in (G,∞) of minimum cost (or decide that none exists).

In the **HITCHCOCK PROBLEM** it causes no loss of generality to assume that *c* is nonnegative: Adding a constant α to each weight increases the cost of each *b*-flow by the same amount, namely by $\alpha\Sigma_{v \in A}\, b(v)$. Often only the special case where *c* is nonnegative and E(G) = A x B is considered.

Obviously, any instance of the **HITCHCOCK PROBLEM** can be written as an instance of the **MINIMUM COST FLOW PROBLEM** on a bipartite graph with infinite capacities. It is less obvious that any instance of the **MINIMUM COST FLOW PROBLEM** can be transformed to an equivalent (but larger) instance of the **HITCHCOCK PROBLEM**.

If we have any type of above mentioned problem then we can easily convert that problem in to the **MINIMUM COST FLOW PROBLEM** and we can solve the **MINIMUM COST FLOW PROBLEM** by using the following four algorithms:

---

(1). MINIMUM MEAN CYCLE-CANCELLING ALGORITHM

Input: A digraph G, capacities $u : E(G) \rightarrow \mathbb{R}_+$, numbers $b : V(G) \rightarrow \mathbb{R}$ with $\Sigma_{v \in V(G)}\, b(v) = 0$, and weights $c : E(G) \rightarrow \mathbb{R}$.

Output: A minimum cost *b*-flow *f*.

ALGORITHM

1. Find a *b*-flow *f*.
1. Find a circuit *C* in G_f whose mean weight is minimum.
   If *C* has nonnegative total weight (or G_f is acyclic) **then stop.**
2. Compute $\gamma := \min_{e \in E(C)} u_f(e)$. Augment *f* along *C* by γ.
   Go to (2).

NOTE :- As described above, (1) can be implemented with any algorithm for the Maximum Flow Problem . For Maximum Flow Problem we can use Ford-Fulkerson Algorithm, Edmonds-Karp Algorithm, Dinic's Algorithm, Goldberg-Tarjan Algorithm and Push-Relabel Algorithm.

---

(2). SUCCESSIVE SHORTEST PATH ALGORITHM

Input: A digraph G, capacities $u : E(G) \to \mathbb{R}_+$, numbers $b : V(G) \to \mathbb{R}$ with $\Sigma_{v \in V(G)} b(v) = 0$, and conservative weights $c : E(G) \to \mathbb{R}$.

Output: A minimum cost b-flow f.

ALGORITHM

1. Set $b' := b$ and $f(e) := 0$ for all $e \in E(G)$.
2. **If** $b' = 0$ **then stop,**
   else choose a vertex s with $b'(s) > 0$.
   Choose a vertex t with $b'(t) < 0$ such that t is reachable from s in G_f.
   If there is no such t **then stop**. (There exits no b-flow.)
3. Find an s-t path in G_f of minimum weight.
4. Compute $\gamma := \min \{ \min_{e \in E(P)} u_f(e) , b'(s) , - b'(t) \}$.
   Set $b'(s) := b'(s) - \gamma$ and $b'(t) := b'(t) + \gamma$. Augment f along P by γ.
   Go to (2).

---

(3). CAPACITY SCALING ALGORITHM

Input: A digraph G with infinite capacities $u(e) = \infty$ ($e \in E(G)$), numbers $b : V(G) \to \mathbb{Z}$ with $\Sigma_{v \in V(G)} b(v) = 0$, and conservative weights $c : E(G) \to \mathbb{R}$.

Output: A minimum cost b-flow f.

ALGORITHM

1. Set $b' := b$ and $f(e) := 0$ for all $e \in E(G)$.
   Set $\gamma = 2^{\lfloor \log B \rfloor}$, where $B = \max \{1,(1/2) \Sigma_{v \in V(G)} | b(v) | \}$.
2. **If** $b' = 0$ **then stop,**
   else choose a vertex s with $b'(s) \geq \gamma$
   Choose a vertex t with $b'(t) \leq - \gamma$ such that t is reachable from s in G_f.
   If there is no such s or t **then go to** (5).
3. Find an s-t path P in G_f of minimum weight.

4. Set $b'(s) := b'(s) - \gamma$ and $b'(t) := b'(t) + \gamma$. Augment f along P by γ. **Go to** (2).
5. **If** $\gamma = 1$ **then stop.** (There exits no b-flow.)
Else set $\gamma := (\gamma/2)$ and **go to** (2).

---

(4). ORLIN'S ALGORITHM

Input: A digraph G with infinite capacities $u(e) = \infty$ ($e \in E(G)$), numbers $b : V(G) \rightarrow \mathbb{R}$ with $\Sigma_{v \in V(G)} b(v) = 0$, and conservative weights $c : E(G) \rightarrow \mathbb{R}$.

Output: A minimum cost b-flow f.

ALGORITHM

1. Set $b' := b$ and $f(e) := 0$ for all $e \in E(G)$.
Set $r(v) := v$ for all $v \in V(G)$. Set $F := \emptyset$.
Set $\gamma := \max_{v \in V(G)} | b'(v) |$.
2. **If** $b' = 0$ **then stop.**
3. Choose a vertex s with $b'(s) > ((n-1)/n)\gamma$.
If there is no such s **then go to** (4).
Choose a vertex t with $b'(t) < (-1/n)\gamma$ such that t is reachable from s in G_f .
If there is no such t **then stop**.(There exits no b-flow.)
Go to (5).
4. Choose a vertex t with $b'(t) < (- (n-1)/n)\gamma$.
If there is no such t **then go to** (6).
Choose a vertex s with $b'(s) > (1/n)\gamma$ such that t is reachable from s in G_f .
If there is no such s **then stop**.(There exits no b-flow.)
5. Find an s-t path P in G_f of minimum weight.
Set $b'(s) := b'(s) - \gamma$ and $b'(t) := b'(t) + \gamma$. Augment f along P by γ.
Go to (2).
6. **If** $f(e) = 0$ for all $e \in E(G) \setminus F$ **then**
set $\gamma := \min \{ \gamma/2, \max_{v \in V(G)} | b(v)| \}$.
else set $\gamma := \gamma/2$.
7. **For** all $e = (x,y) \in E(G) \setminus F$ with $f(e) > 8n\gamma$ **do:**
Set $F := F \cup \{e, \bar{e}\}$.
Let $x' := r(x)$ and $y' := r(y)$. Let Q be the x'- y' path in F.
If $b'(x') > 0$ **then** augment f along Q by $b'(x')$,
else augment f along the reverse of Q by $- b'(x')$.

Set $b'(y') := b'(y') + b'(x')$ and $b'(x') := 0$.
For all $e = (v, w) \in E(G) \setminus F$ with $r(v) = x'$ and $r(w) = y'$ **do:**
Augment f along $\bar{e}$ plus the v-w path in F by $f(e)$. Delete e.
Set $r(z) := y'$ for all vertices z reachable from y' in F.

8. **Go to** (2).

Some Useful Applications

- Reallocation of Housing
- Assortment of Structural Steel Beams
- Tournament Problem
- Leveling Mountainous Terrain
- Rewiring of Typewriters
- Pairing Stereo Speakers
- Measuring Homogeneity of Bimetallic Objects
- Electrical Networks
- Determining an Optimal Energy Policy
- Racial Balancing of Schools