15-853 Algorithms in the Real World **Lecturer: Prof. R. Ravi**
Topic: Logistics Lecture 1 **Scribe: Krithika Venkataramani**

Logistics are useful in variety of applications like

- Supply chain management,
- Transportation planning
- Facility location
- Inventory planning

Solutions to these problems involve integer programming. However, integer programming is an NP hard problem and so here, some heuristic algorithms which yield approximate solutions are discussed.

We discuss two versions of network design which often arise in practice – Buy-at-Bulk (*BB*) network design and Deep-Discount (*DD*) network design. An example of each is given below [1].

Example 1: Telecom network provisioning

There is a single base station, multiple towers, each of which has a different demand. Cables have to be installed to route demands. Cables are graded by capacity

Ex: T1 – 25 calls at $25/mile
or T4 – 400 calls at $100/mile

Higher capacity cables have lower cost/unit capacity/unit length).

We need to minimize the cost of installing and routing cables from the towers to the base station.

This is an example of the Buy-at-Bulk (*BB*) network design.

Example 2: Transport Network Design

There is a single production center, multiple markets, each with a different demand. Contractors offer various contract types depending on volume.

Ex. $10K to enter low-volume contract with rate of $2 per car per mile
or $15K to enter high-volume contract with rate of $1 per car per mile.

For higher volumes, there is a lower rate/unit flow/unit length).

Each market wishes to route a certain amount of demand from the production center. We need to minimize the cost of transportation of demands from the production center to all markets.

This is an example for the Deep-Discount (*DD*) network design.

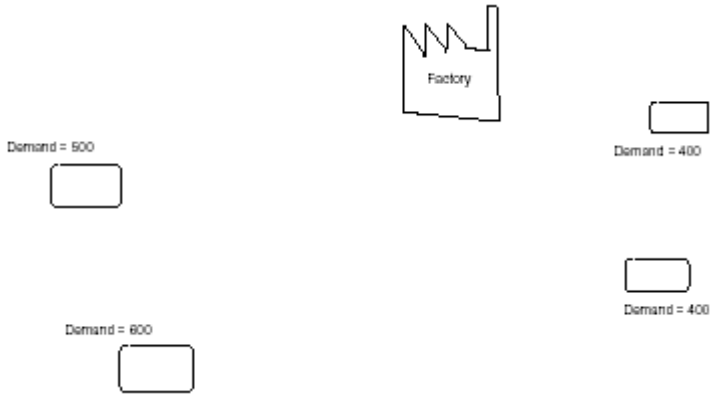

Figure 1. An example of a network design problem. We need find a network that minimizes cost of transportation of demands from the factory to the various markets.

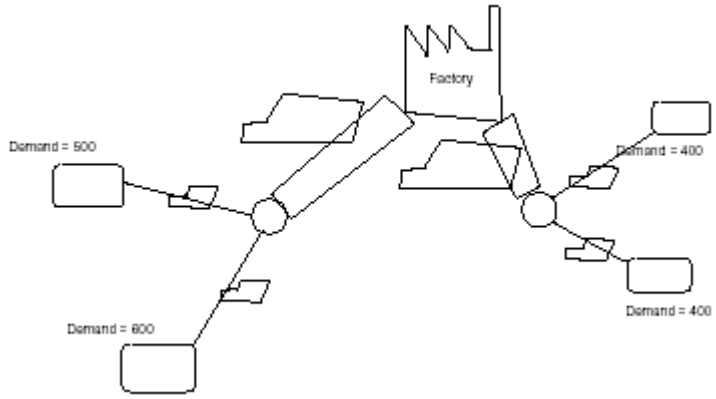

Figure 2. A solution to the network design problem with two aggregate points.

Problem Definition

We are given an undirected graph $G = (V,E)$ with a single sink $t \in V$,
multiple sources $\{v_j\}_1^S = S \subset V$ with demands $\{dem_j\}_1^S : S \to \mathrm{R}$ to be routed to the sink,
edge lengths $l_e : E \to \mathrm{R}^+$. [2]

Buy-at-Bulk (*BB*) Network Design

We have a set of k cable types available to us to purchase and install. Each cable i has the characteristics
u_i – capacity
c_i – cost per unit length

The cost per unit capacity per unit length of a high capacity cable is considerably less than that of a low capacity cable, reflecting an economy of scale, i.e. the payoff for buying at bulk may be very high.

Let us order the capacities as $u_0 < u_1 < \ldots < u_{k-1}$. Then $\frac{c_0}{u_0} > \frac{c_1}{u_1} > \ldots > \frac{c_{k-1}}{u_{k-1}}$

We wish to buy cables such that we have enough capacity to support (simultaneous) flow requirements. There is no flow cost; the only cost incurred is the purchase price of the cables. We also need to find the flow routing for demands to the sink.

The problem is to minimize the cost of installing cables, i.e.

$$\min \sum_{\text{edges } e} \sum_{\substack{\text{cables K}(e) \\ \text{installed in e}}} c_{\mathrm{K}(e)} l_e$$

This is an NP hard problem even when $k = 1$.

Deep-Discount (*DD*) Network design

We have a set of k cable types/ k contract types available to us to purchase. The cable/contract i has the characteristics
p_i - price of entering contract i / installing cable i per unit length (fixed cost)
r_i – rate per unit flow per unit length (variable cost)

The cost per unit flow along an edge is proportional to the edge length. There is an economy of scale here too; we can reduce the cost per unit flow of routing by paying a certain fixed cost (again proportional to the length of the edge). Let us order the rates as $r_0 > r_1 \ldots > r_{k-1}$. The rate $r_0 = 1$ and the price $p_0 = 0$ correspond to not using any discount. (we may scale our cost functions so that this is true in general). Without loss of generality the prices are ordered as $p_0 < p_1 \ldots < p_{k-1}$.

If we install cable i at edge e of length l_e and route flow f_e through it, the contribution to the cost is $l_e(p_i + f_e r_i)$. We may therefore view the installation of cable i at an edge as paying a fixed cost $p_i l_e$ in order to obtain a discounted rate r_i of routing along this edge.

We need to enter into contracts/install cables and find the flow routing for demands to the sink.

The problem is to minimize the routing cost, i.e. $\min \sum_{\text{edges } e} l_e(p_{K(e)} + f_e r_{K(e)})$

Definition 1:

The performance guarantee of an algorithm A for a minimization problem P, is the worst ratio over all input instances I, of the cost (objective function value) of the solution returned by A to that of the optimum (minimum).

$$\rho_P(A) = \max_{\text{instances } I \text{ of } P} \frac{c(\text{soln}(I))}{c(\text{opt}(I))}$$

Smaller approximation factors are better.

Approximate equivalence of *BB* and *DD*

Given a Buy-at-Bulk problem instance *BB = (G,c,u)* on a graph *G* with *k* cables types having

costs $c_i, 1 \le i \le k$ and

capacities $u_i, 1 \le i \le k$,

we can transform it into a Deep-Discount problem instance *DD* by a mapping Φ

$$BB \xrightarrow{\Phi} DD$$

by setting edge costs

$p_i = c_i, \forall i$ (price of installation per unit length),

$r_i = \frac{c_i}{u_i}, \forall i$ (rate of routing per unit flow per unit length)

and call this $\Phi(BB)$.

Conversely, given a *DD* instance *DD = (G,p,r)* with *k* contracts, we can transform it into a *BB* instance using a mapping Ψ

$$DD \xrightarrow{\Psi} BB$$

by setting costs and capacities

$c_i = p_i, \forall i$ (cost per unit length),

$u_i = \frac{p_i}{r_i}, \forall i$ (capacity)

and call it $\Psi(DD)$.

It is easy to see that the two transformations are inverses of each other. For a problem instance X, we abuse notation to let X also denote the cost of a feasible solution to it. Let X^* denote the cost of an optimal solution to X.

Lemma 1: $\Phi^*(BB) \leq 2BB^*$

$\Psi^*(DD) \leq 2DD^*$

Proof: Exercise 1.

These transformations are useful in cases where it is easier to find a solution in one domain, say, for the Deep-Discount problem. Then, we can transform the Buy-at-Bulk *BB* problem into the *DD* problem and find a solution for the *DD* instance. If we need the solution in the *BB* domain, we can transform the *DD* solution back into the *BB* domain. The obtained solution in the *BB* domain will be at most worse than the optimal *BB* solution by a factor of 4.

We will restrict our discussion to the Deep-Discount problem as solutions for the *BB* problem can be obtained by transformations and will not be worse than a factor of 4 from the optimal solution.

Example 1: One edge case

We have one source with demand 500 and one edge routing the demand from this source to the sink.

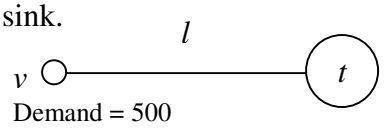

Figure 3: Single edge example

Cable Type	Price (p_i)	Rate (r_i)	Total Price/unit length
T1	$25	$1	$525
T4	$100	$0.5	$350
T10	$400	$0.1	$450

Table 1: Rates and prices for different types of cables

The lower envelope of the cost function, as seen in figure 4 is a concave function. This is due to linearity of the rates of contracts.

Solution: From figure 4, for the single edge case, when demand is 500, the total cost is minimum when the second cable type is used.

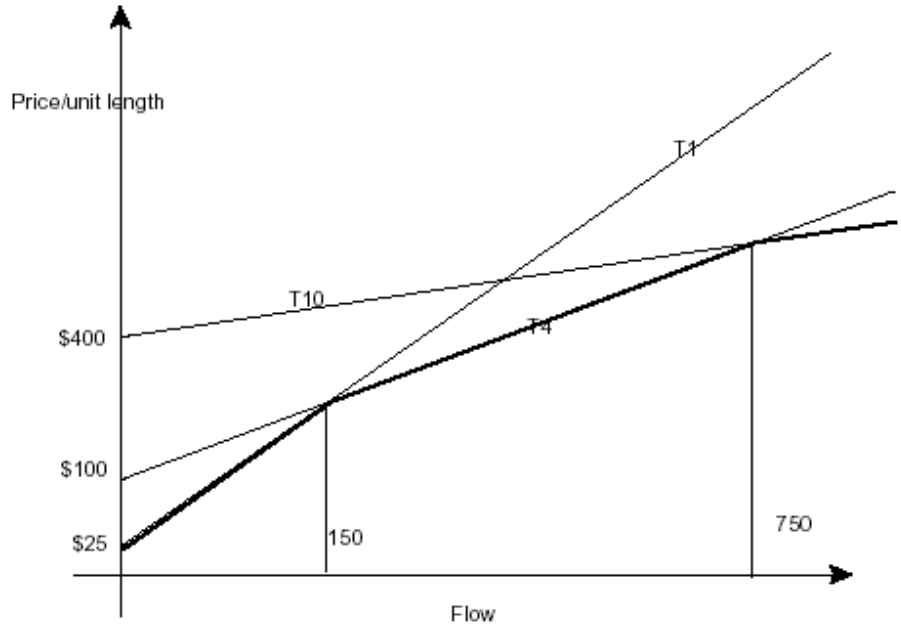

Figure 4: Price/unit length Vs. Flow for three types of cables.

Definition 2:

A function f is said to be concave iff $f(\frac{a+b}{2}) \geq \frac{f(a)+f(b)}{2}$.

Example 2: Two source case

We have two sources V_1 and V_2 with demands 400 and 100 respectively and we have to route these through one aggregate point P_1 to the sink t.

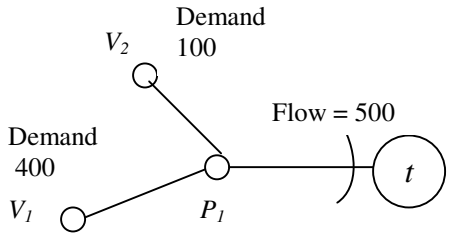

Figure 5: A two source example

The same types of cables are present as in the previous example.

Solution: The second cable type is best for the edge from V_1 to P_1. The first cable type is best for the edge from V_2 to P_1. The flow from P_1 to t is 500 and from the figure 2, it is best to use the second cable type in this edge.

Moral: There is only one cable type in any edge of an optimal solution.
The flow never decreases or splits on a source-sink path.

Special Case 1: All prices are zero

We are given an undirected graph $G = (V,E)$ with a single sink $t \in V$,
multiple sources $\{v_j\}_1^S = S \subset V$ with demands $\{dem_j\}_1^S : S \to \mathrm{R}$ to be routed to the sink,
edge lengths $l_e : E \to \mathrm{R}^+$.

We have a set of k cable types/ k contract types available to us to purchase. The cable/contract i has the characteristics
p_i - price of entering contract i / installing cable i per unit length
r_i – rate per unit flow per unit length
All prices p_i's are zero.

When all the prices p_i's are zero, we should choose the contract with the minimum rate r_{min} in each edge. We should take the shortest path to the sink t.

Define a distance function for each edge e: $dist_e = l_e . \min_{i \in e} r_i$

Solution: The solution to this special case is the shortest path tree rooted at t under the distance function *dist.*

Special Case 2: All rates are zero

We are given an undirected graph $G = (V,E)$ with a single sink $t \in V$,
multiple sources $\{v_j\}_1^S = S \subset V$ with demands $\{dem_j\}_1^S : S \to \mathrm{R}$ to be routed to the sink,
edge lengths $l_e : E \to \mathrm{R}^+$.

We have a set of k cable types/ k contract types available to us to purchase. The cable/contract i has the characteristics
p_i - price of entering contract i / installing cable i per unit length
r_i – rate per unit flow per unit length
All rates r_i's are zero.

When all rates r_i's are zero we should choose the contract with the minimum price p_{min} in each edge.

Suppose we have a further restriction that all nodes in the graph have demand = 1.

Define a cost function: $c_e = \min_{i \in e} p_i$

Solution: The solution to this special case is the minimum spanning tree under the cost function c_e.
In general, the solution is a Steiner tree for the terminals $R = S \cup \{t\}$

Steiner Tree: It is a minimum weight tree connecting a designated set of vertices, called terminals in a weighted graph. The tree may include non-terminals, which are called Steiner points. The sum of distances from a Steiner point to all other points connecting it is minimum.

Input: Graph *G(V,E)*
Edge costs $c : E \to R^+$
Terminals: $k \leq V$
Output: We need to find a connected subgraph spanning R.
Objective: This has to be done while minimizing the length of the output subgraph.

Finding a Steiner tree is an NP hard problem. However, there are several approximations to it. Adam and Myers [3] have found an approximation constant of 60 to the Steiner tree problem.

Special Case 3: All rates are 1 and all prices are small but non-zero.

We are given an undirected graph *G = (V,E)* with a single sink $t \in V$,
multiple sources $\{v_j\}_1^S = S \subset V$ with demands $\{dem_j\}_1^S : S \to R$ to be routed to the sink,
edge lengths $l_e : E \to R^+$.

We have a set of k cable types/ k contract types available to us to purchase. The cable/contract *i* has the characteristics
p_i - price of entering contract *i* / installing cable *i* per unit length
r_i – rate per unit flow per unit length.
All $p_i = \frac{1}{LD}$ and the rate $r_i = 1,\ 1 \leq i \leq k$
where $L = \sum_{e \in Edges} l_e$ and $D = \sum_{j=1}^{S} dem_j$
with the restriction that all demands are 1.

The solution is a special kind of shortest tree.

Exercise 4: What is the solution?

Structure of an optimum solution to the *DD* problem

The following assumptions are true for the *DD* problem

1. In an optimal solution, there is at most one contract per edge.
2. The support graph of the optimal solution is acyclic (in undirected graphs), i.e. it is a tree
3. There is path monotonicity. Since cable types are monotone in flow quantity, all paths from sources to sink are monotonic.

Exercise 2: The first two statements do not hold for the *BB* case.

Lemma 2: Optimal Solution to the *DD* problem is a tree.

Proof: Let $K^*(e)$ be the optimal contract on the edge *e.*

Define a new distance function $dist_e = l_e.r_{K^*(e)}$

Once the fixed cost for the contract is paid, the routing cost is minimized if we route along a shortest path according to the distance function $dist_e$. Therefore, there is an optimum solution which routes along the shortest path according to such a distance function. As a result, the support graph of the flow is a tree. In particular, it is a shortest path tree under *dist* .

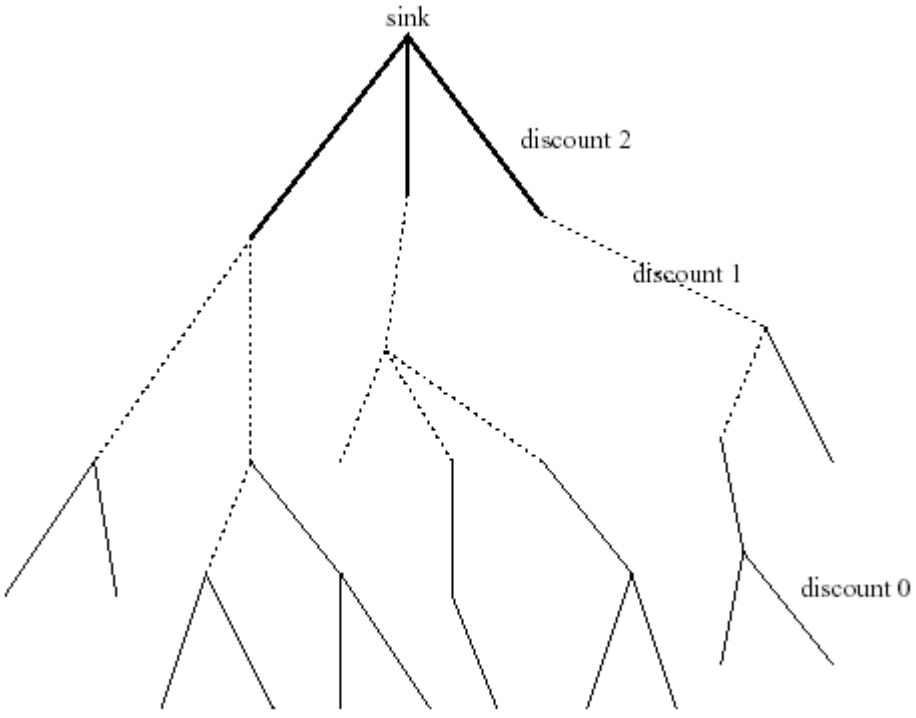

Figure 6: Structure of an optimal solution. The discount types correspond to different cables/contracts, and are ordered. Discount 0 corresponds to the least price and highest rate cable.

We can get a lower bound to the *BB* and *DD* problems from linear programming solution relaxations. These are useful in gaining insight into these problems. Lower bounds can be obtained in the following ways.

1. Heuristic algorithms can be obtained by rounding into integer programming
2. Search trees could be pruned and so we can reduce it to a simpler problem.
3. A quick a dirty solution can be obtained by looking from a high level. Path monotonicity could be used to come up with reasonable solutions.
4. Graphs can be restricted. For example, by saying that a cable cannot be used unless at least half the capacity is used. This would only result in a extra constant factor times the optimal solution.

References:

1. Talk: "Approximation algorithms for buy-at-bulk network design". http://www.gsia.cmu.edu/andrew/ravi/public/caltech.PDF
2. "On the integrality gap of a natural formulation of the single-sink buy-at-bulk network design problem", N. Garg, R. Khandekar, G. Konjevod, R. Ravi, F. S. Salman, A. Sinha, *Proceedings of the Conference on Integer Programming and Combinatorial Optimization (IPCO* 2001).
   http://www.andrew.cmu.edu/~asinha/papers/dd.ps
3. Adam and Myers