An Application to Network Design

Consider the following network design problem. Given N nodes and a demand of transporting data from node i to node j ($i, j = 1, \ldots, N,\ \ i \neq j$) at a speed of b_{ij} Mbit/s.

We can build links between any pair of nodes. The cost for unit capacity (=1 Mbit/s) on a link from node i to j is a_{ij}. Higher capacity costs proportionally more, lower capacity costs proportionally less.

Set $a_{ii} = 0,\ b_{ii} = 0$ for all i so we do not have to take care of the case when $i = j$ in the formulas.

The goal is to design which links will be built and with how much capacity, so that the given demand can be satisfied and the overall cost is minimum.

Let us find an LP formulation of the problem!

Let z_{ij} be the capacity we implement on link (i, j). This is not given, this is what we want to optimize. If the result is $z_{ij} = 0$ for some link, then that link will not be built.

With this notation the cost of link (i, j) is $a_{ij}z_{ij}$, so the objective function to be minimized is

$$Z = \sum_{i,j} a_{ij} z_{ij}$$

To express the constraints, let $x_{ij}^{(kl)}$ be the amount of flow on link (i, j) that carries traffic from node k to l (not known in advance). Then the total traffic on link (i, j) is obtained by summing up these variables for all k, l :

$$\sum_{k,l} x_{ij}^{(kl)}.$$

Thus, the capacity constraint for link (i, j) can be expressed as

$$\sum_{k,l} x_{ij}^{(kl)} \leq z_{ij}.$$

The flow conservation should hold for each piece of flow, that is, for the flow carrying traffic between each source-destination pair k, l. To express this concisely, let us define new constants by

$$d_i^{(kl)} = \begin{cases} b_{kl} & \text{if} \quad i = k \\ -b_{kl} & \text{if} \quad i = l \\ 0 & \quad \text{otherwise} \end{cases}$$

The value of $d_i^{(kl)}$ shows whether node i is source, sink or transshipment node for the $k \to l$ flow.

Thus, the flow conservation constraints (one for each node and for each flow) can be written as

$$\sum_j x_{ij}^{(kl)} - \sum_r x_{ri}^{(kl)} = d_i^{(kl)} \qquad (\forall i, k, l)$$

Collecting all the pieces, the LP formulation is:

$$\min\ Z = \sum_{i,j} a_{ij} z_{ij}$$

subject to

$$\begin{aligned}
\sum_j x_{ij}^{(kl)} - \sum_r x_{rj}^{(kl)} &= d_i^{(kl)} && (\forall i, k, l) \\
\sum_{k,l} x_{ij}^{(kl)} - z_{ij} &\leq 0 && (\forall i, j) \\
x_{ij} &\geq 0 && (\forall i, j) \\
z_{ij} &\geq 0 && (\forall i, j)
\end{aligned}$$

<u>Comments</u>

- This task is a variant of flow problems that are often referred as *Multicommodity Flow* because several flows are handled simultaneously.

- This model contains several major simplifications. They make the problem solvable via linear programming, but at the price of distancing the model from reality. Try to list some of these simplifications!

- Having found the LP formulation, a "brute force" way to solve it would be to apply a general purpose LP algorithm. Given that the number of variables and constraints is large, it would be very slow. Below we show that there is, however, a clever way of circumventing the complexity, using the special features of the problem.

A Fast Solution

Observe that the cheapest way of sending b_{ij} amount of flow from node k to l is to send it all along a path for which the sum of the link costs is minimum. If this path consists of nodes $k = i_1, i_2, \ldots, i_{r-1}, i_r = l$, then the resulting piece of cost is

$$b_{kl}(a_{i_1,i_2} + \ldots + a_{i_{r-1},i_r})$$

Due to the linear nature of the model, these costs simply sum up, independently of each other. This suggests the following simple algorithm:

- Find a minimum cost path between each pair k, l of nodes, with edge weights a_{ij}. This can be done by any standard shortest path algorithm that you met in earlier courses.
- Set the capacity of link (i, j) to the sum of those b_{kl} values for which (i, j) is on the min cost path found for k, l.
- The optimum cost can be expressed explicitely. Let E_{kl} be the set of edges that are on the min cost $k \to l$ path. Then, according to the above, the optimal cost is:

$$Z_{opt} = \sum_{k,l} \left(b_{kl} \sum_{(i,j) \in E_{kl}} a_{ij} \right).$$