1 Single Commodity Multi-path Flows

In this chapter, we consider single commodity multi-path flow problems. Here each unit of flow must be sent along several edge (node) disjoint paths. The first of these is the *maximum multi-path flow* problem and its description is given below:

2 Problem

Given a network $G = [N;\ A]$, a special node, s, called the source or origin and a node, t, called the sink or destination and positive numbers $u_{i,j}$ representing the capacity of arc $(i, j) \in A$, we want to maximize the total shipment from s to t where each unit of shipment must be sent along q disjoint paths (we consider edge disjoint paths here; node disjoint paths can also be done). This problem is called the **maximum multi-path flow problem** in the literature and is the starting point in this area. The early work is this area is recent and due to Kishimoto.

We would like to obtain results similar to that in the regular maximum flow problem such as max-flow-min-cut theorem, a linear programming node-arc formulation, an arc-chain formulation, the correspondence between the two, a strongly polynomial algorithm, and an augmenting method to solve the problem. After this we can consider extensions to multi-terminal flow problems and multi-commodity flow problems.

3 Formulation

3.1 Arc Chain Formulation I:

Let A represent the incidence matrix whose rows correspond to arcs and columns to the union of q arc disjoint $s - t$ paths. The following represents the linear program corresponding to the above mentioned multi-path maximum flow problem:

$$\begin{aligned} Ax &\leq u \\ x &\geq 0 \\ &\max e^t x \end{aligned}$$

This resembles the arc-chain formulation in the regular maximum flow problem with the appropriate change in the definition of the matrix. The node-arc formulation is more complex and this is the main contribution of Kishimoto paper. From now on we will restrict our attention to the case where $q = 2$. The changes needed for arbitrary values will be clear.

3.2 Node Arc Formulation:

Let $f_{i,j}$ be the flow on arc $(i,j) \in A$ and F be the total flow across the network. Then:

$$\sum_j (f_{i,j} - f_{j,i}) = \begin{cases} F & i = s \\ 0 & i \neq s,t \\ -F & i = t \end{cases}$$

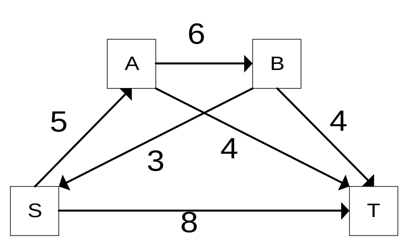
$$0 \leq f_{i,j} \leq \min[u_{i,j}, \frac{F}{2}] \text{ for } (i,j) \in A$$

$$\max F$$

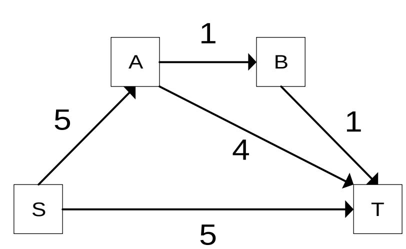
The notion of a *cut* or *a cut set separating* s and t can be defined in several slightly different ways. One of these is as a partition of the node set N into two disjoint sets S and $\bar{S}$ with $s \in S$ and $t \in \bar{S}$.

Lemma 1: Let $[F, f]$ be any feasible flow and let $(S, \bar{S})$ be any cut separating s and t. Then $F \leq \beta_2(S, \bar{S})$ where $\beta_2(S, \bar{S}) = \min[\frac{\sum_{i \in S} \sum_{j \in \bar{S}} u_{i,j}}{2}, \sum_{i \in S} \sum_{j \in \bar{S}} u_{i,j} - \max_{\substack{i \in S \\ j \in \bar{S}}} u_{i,j}]$. $\beta_2(S, \bar{S})$ is called the $2-$value of the cut $(S, \bar{S})$. If equality holds then $[F, f]$ is optimal to the maximum flow problem and $(S, \bar{S})$ is a cut whose $2-$ value is minimum among all cuts separating s and t.

It takes some work to show that the above formulation and the lemma are correct. It is easy to show that the constraints in the linear program and the upper bound in the lemma hold for any feasible solution. It is the converse that takes work. We may restrict our attention to flows that are acyclic without loss. We will show that for any pair $\{f, F\}$, satisfying the conditions of the formulation, there exists a set of edge disjoint pairs of paths that together carry the total value. For this we use bipartite matching. This is best illustrated with an example.

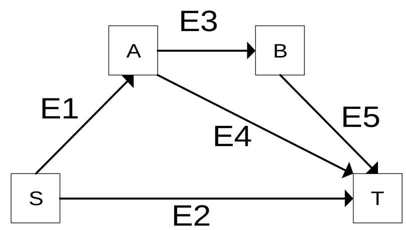


Capacities

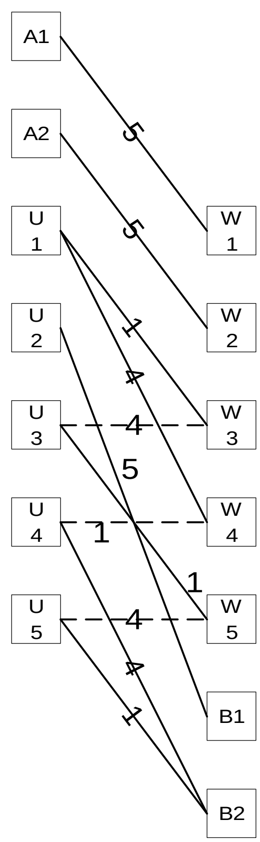


Flows

In order to decompose these arc flows into bipath flows, we number the arcs in the second diagram arbitrarily as shown below:

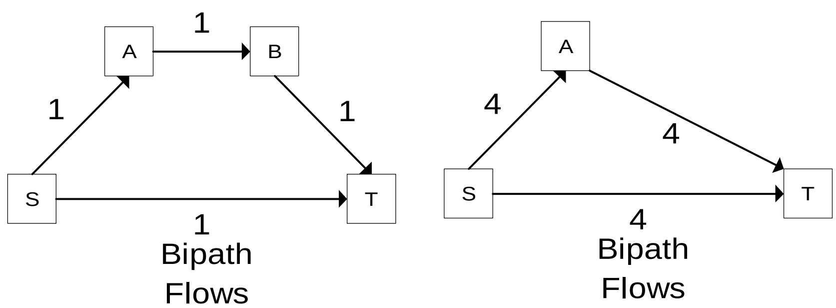


We can see that flow on arc E1 is distributed between the arcs E3 and E4 (In general, there may be many ways to do this at a node. We can use any arbitrary one for our purposes.) . Since the above network is acyclic, this distribution is easily found. We use this now. We construct a graph shown below:



Note that sum at any node is exactly equal to $\frac{F}{q}$.

This graph is now decomposed into matchings. Each of these matchings give us double paths. This is shown below:



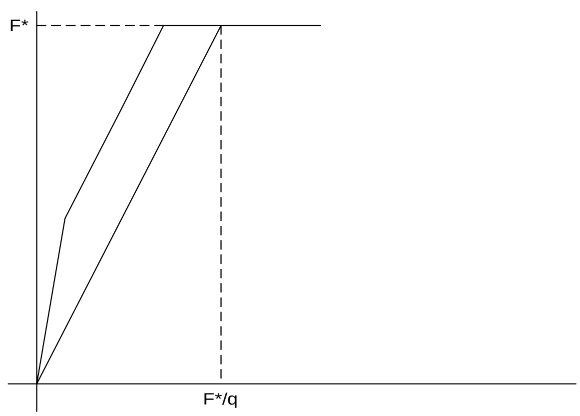
This shows that the linear programming formulation is correct. To solve this linear program, we instead solve the following parametric maximum flow problem first:

$$\sum_j (f_{i,j} - f_{j,i}) = \begin{cases} F & i = s \\ 0 & i \neq s, t \\ -F & i = t \end{cases}$$

$$0 \leq f_{i,j} \leq \min[u_{i,j}, \lambda] \text{ for } (i,j) \in A$$

$$\max F$$

Let this value be denoted by $F^*(\lambda)$. We intersect this curve with the straight line with slope 2 (q in general) to get the value of the above flow. This is shown by the following diagram:



We note that the line with slope 2 cuts the other at a point where the slope of the other is either 1 or 0. These pieces of the other line correspond to cut values and slope of 1 indicates one arc with flow equal to λ and else none. This proves the lemma.

3.3 REFERENCES:

KTK W. Kishimoto, M. Takeuchi, and G. Kishi: Two-route flows in an undirected network, IEICE Trans. J-75-A (1992), 1699-1717 (in Japanese)

KT W. Kishimoto, and M. Takeuchi: On m-route flows in a network, IEICE Trans. J-76-A (1993), 1185-1200 (in Japanese)

KT W. Kishimoto, and M. Takeuchi: On two-route flows in an undirected network, IEICE Technical Report, CAS90-19, DSP-90-23, (1990) (in Japanese)

K W. Kishimoto: A method for obtaining the maximum multi-route flows in a network, Networks 27 (1996), 279-291.

AO C.C. Aggarwal and J.B. Orlin: On Multiroute Maximum Flows in Networks, Networks, 39, (2001), 43-52.