1 Introduction

1.1 Motivation

Many problems in optimization deal with delivering units of some sort from a source to a destination through some network, be it coffee from warehouses to retail locations through transport routes, traffic from one city to another through highways, or signals from a computer to a server through wires. Such problems are Network Flow problems, which we will look at in this lecture.

1.2 Networks versus Graphs

It is important to make a distinction between networks and graphs. A network, like a graph, is a set of objects and pairs of these objects, which we call **nodes** and **arcs** (rather than vertices and edges), the distinction being that the pairs are ordered; that is, in a graph, the edge uv is the same as the edge vu, but in a network, uv and vu represent distinct arcs.

For this reason, we must also make a distinction between objects dealing with edges and similar objects dealing with arcs (such as paths, cycles, trees, etc.). For the purposes of the the maximum flow problem, we need only worry about paths. Here, we define a **path** as a sequence $v_0v_1, v_1v_2, ..., v_{k-1}v_k$ where each $v_{i-1}v_i$ is an arc in our network for $i = 1, 2, ..., k$, whereas an **unordered path** is a sequence $v_0v_1, v_1v_2, ..., v_{k-1}v_k$ where, for each i, one of $v_{i-1}v_i$ and v_iv_{i-1} is an arc in our network.

2 Formulating the Problem

2.1 Flow Requirements

For a problem to be classified as a maximum flow problem, we need some data in addition to a set of nodes and a set of arcs. We must have some node s with no incoming arcs which we will call a **source node**, and a node t with no outgoing arcs which we will call the **sink node**. The problem must relate to sending as many particles from s to t as possible through the network. Furthermore, we will require that for any node u in our network, there is a (directed) path from s to u.

Each arc a in our network must have some sort of upper bound on the number of units of good that can be sent through it. We call this the **capacity** of arc a, and denote it $c(a)$.

2.2 Definition of Flow

Definition 2.1. Let $G = (N, A)$ be a network where every $uv \in A$ has capacity $c(uv) > 0$. For every $(u, v) \in N \times N$, where $uv \notin A$, let $c(uv) = 0$. Let $s \in N$ be our source node, and let $t \in N$ be our sink node.

A **flow** is a function $f : N \times N \to \mathbb{R}$ satisfying

- $f(uv) = -f(vu)$ for all $(u, v) \in N \times N$ (skew symmetry)
- $\sum_{v \in N} f(uv) = 0$ for all $u \in N, u \neq s, t$ (flow conservation)

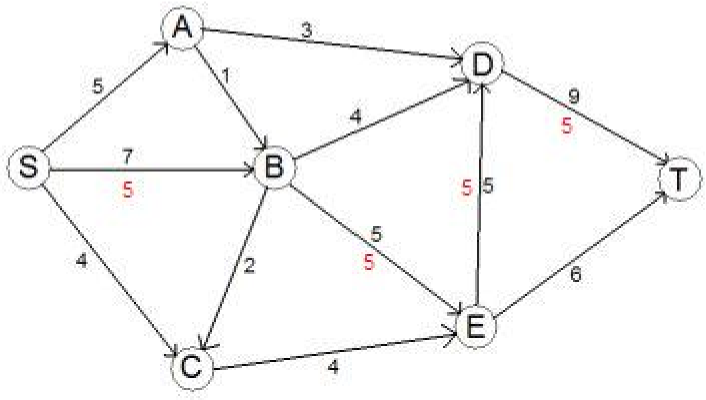
Additionally, we say that a flow is **feasible** if it satisfies $0 \leq f(uv) \leq c(uv)$ for all arcs $uv \in A$ (we call this the *feasibility constraint*). We say that the **value** of the flow is $\sum_{v \in N} f(sv) = \sum_{v \in N} f(vt)$ (note that this equality holds as a result of flow conservation and skew symmetry).

Note that if, for vertices u and v, we have $uv \in A, vu \notin A$, we need only track the value $f(uv)$ since we can calculate the value $f(vu)$ from $f(uv)$ by the skew symmetry property. If we have $uv \notin A, vu \notin A$, then we needn't track either value if we are concerned only with feasible flows, since for any feasible flow, we must have $f(uv) = f(vu) = 0$.

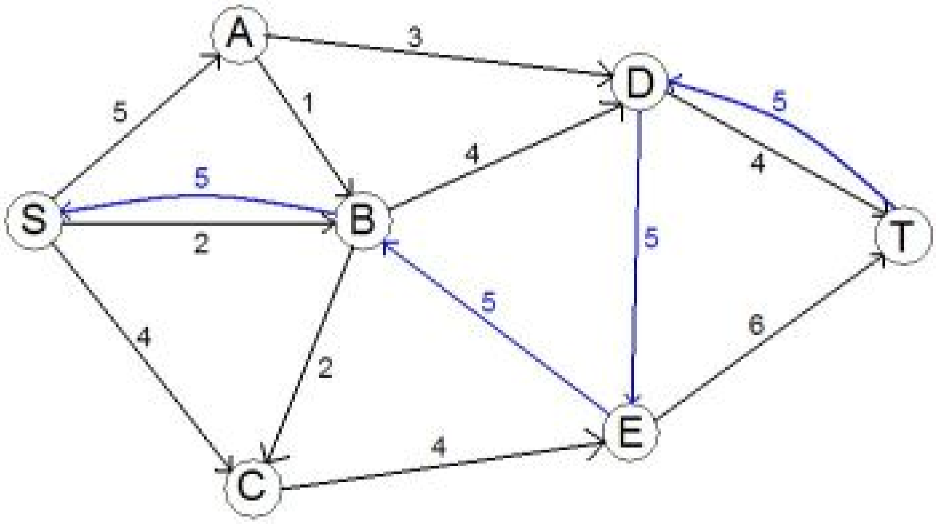
2.3 The Residual Network

Definition 2.2. Let the **residual capacity** of $(u, v) \in N \times N$ be given by $c_f(uv) = c(uv) - f(uv)$. This defines the **residual network** $G_f = (N, A_f)$, where $uv \in A_f$ if $c_f(uv) > 0$.

Example 2.3. Suppose we have the following network, where the capacities of the arcs are written in black and are marked next to the arcs, together with a feasible flow of value 5, where the flow on an arc (if nonzero) is marked in red next to the arc (recall that by signifying $f(SB) = 5$, this implies $f(BS) = -5$):



Then the residual graph is as follows, where arcs appearing in the residual graph but not the original graph are marked in blue:



2.4 Augmenting Paths

Definition 2.4. An **augmenting path** is a (directed) path $P = v_0v_1, ..., v_{k-1}v_k$ in the residual graph where $v_0 = s, v_k = t$. We say that the **residual capacity** of P is $c_P = \min_i\{c_f(v_{i-1}v_i)\}$.

Note that the residual capacity of a path P is the largest value of flow we can push along the path P while maintaining a feasible flow.

Example 2.5. In the residual graph above, $P =$ SA,AB,BD,DT is an augmenting path with residual capacity $\min\{5, 1, 4, 4\} = 1$.

3 Ford-Fulkerson Algorithm

Given a network $G = (N, A)$ with capacities c, a source node s, and a sink node t, the following algorithm with return a flow from s to t of maximum value.

1. Initiate $f(uv) = 0 \;\; \forall uv \in A$.
2. Construct the residual graph G_f.
3. If no augmenting path exists in G_f, stop; f is maximal.
4. Find an augmenting path P in G_f and let its residual capacity be c_P.
5. For each arc $uv \in P$,
   (a) $f(uv) \leftarrow f(uv) + c_P$
   (b) $f(vu) \leftarrow f(vu) - c_P$
6. Go to (2).

4 Max-Flow Min-Cut Theorem

We must now justify why the Ford-Fulkerson algorithm returns a maximal flow.

Definition 4.1. We say that a **cut** of our network is a division of our nodes into (S, T) where $s \in S, t \in T$, $S \cup T = N$, and $S \cap T = \emptyset$. We say that the **capacity** of the cut is $\sum_{u \in S, v \in T, uv \in A} c(uv)$.

Note that this definition of a cut differs slightly from the definition of a cut we've seen previously; here, we require that $s \in S$ and $t \in T$.

Theorem 4.2 (Max-Flow Min-Cut Theorem)**.** *The maximal flow value in a network is equal to the minimal cut capacity of all cuts in the network.*

Proof. Since, for any feasible flow, we have $f(a) \leq c(a) \;\; \forall \, a \in A$, then, for any cut, the amount of flow that can cross the cut is bounded above by the capacity of the cut. Thus it suffices to show that equality holds for some flow and some cut in the network.

Take the flow returned by the Ford-Fulkerson algorithm such that there is no augmenting path in the residual graph. Let S be the set of all vertices u for which there is a (directed) path from s to u in the residual graph. Let T be $N \backslash S$. Note that $t \in T$ since otherwise, there is a path from s to t, and such a path would be an augmenting path which we have assumed does not exist.

Similarly, no arc $uv \in A$ where $u \in S, v \in T$ is in the residual graph, else the path from s to u together with the arc uv would be a path from s to v, meaning $v \in S$. This means that for all such arcs $uv \in A$ where $u \in S, v \in T$, $f(uv) = c(uv)$ (since $c_f(uv) = c(uv) - f(uv) = 0$).

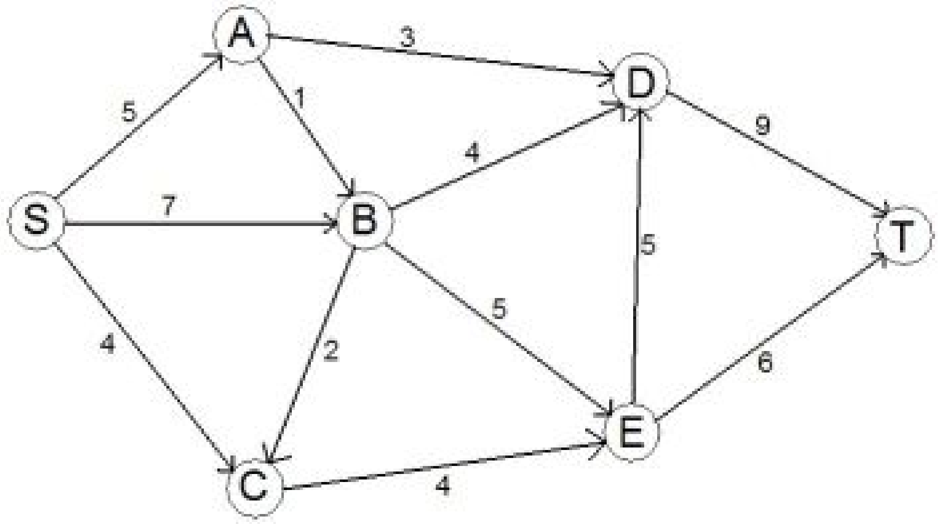
For any arc $uv \in A$ where $u \in T, v \in S$, the flow on uv must be zero, else the residual capacity of the arc vu would be positive, and thus the path from s to v together with the arc vu is a path from s to u in the residual graph, meaning $u \in S$. So, for any arc $uv \in A$ where $u \in T, v \in S, f(uv) = 0$.

We note that the value of a flow f is given by $\sum_{u \in S, v \in T, uv \in A} f(uv) - \sum_{u \in T, v \in S, vu \in A} f(uv)$ (that is, the amount of flow going from s to t is the amount of flow going from S into T, minus the amount that comes back from T into S). By our arguments above, this is equal to $\sum_{u \in S, v \in T, uv \in A} c(uv) - \sum_{u \in T, v \in S, vu \in A} 0$, which is precisely the capacity of the cut (S, T).

Hence, we have found a flow and a cut for which the value of the flow is equal to the capacity of the cut, finishing our proof. □

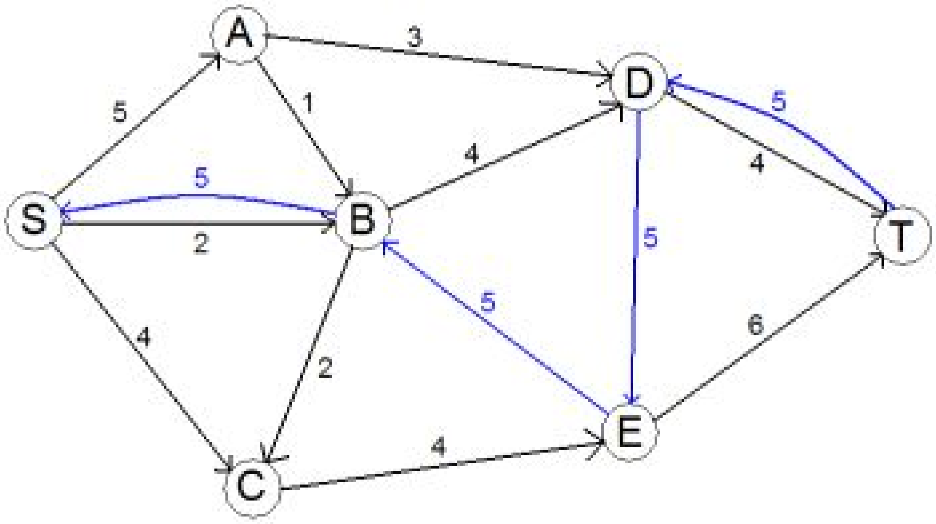
4.1 Example Problem

Consider the following network, with the capacity of each arc is marked beside the arc:

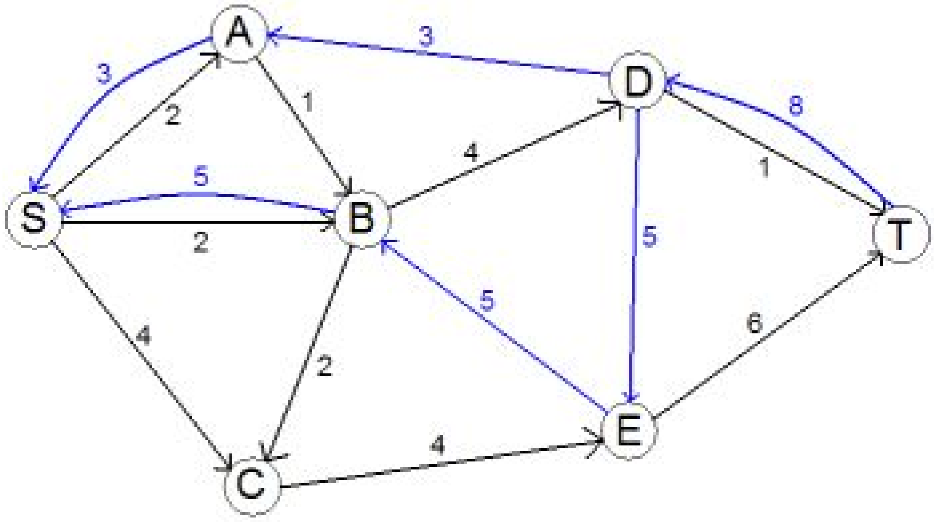


We first initialize the flow f by setting $f(a) = 0$ for all arcs a. Since the flow is zero, the residual graph is the same as the original graph. We choose the path SB, BE, ED, DT with capacity 5 on which to send flow, to obtain the flow $f(SB) = f(BE) = f(ED) = f(DT) = 5$ (we note only the flow values that have changed from the previous flow).

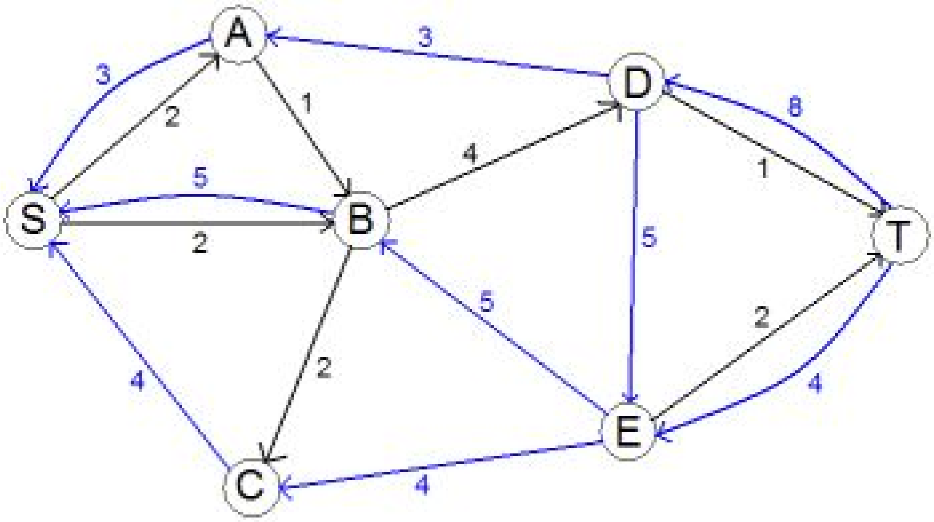
Doing so gives us a new residual graph. We again mark the arcs that are in only the residual graph in blue, and mark the arcs with the residual capacities:



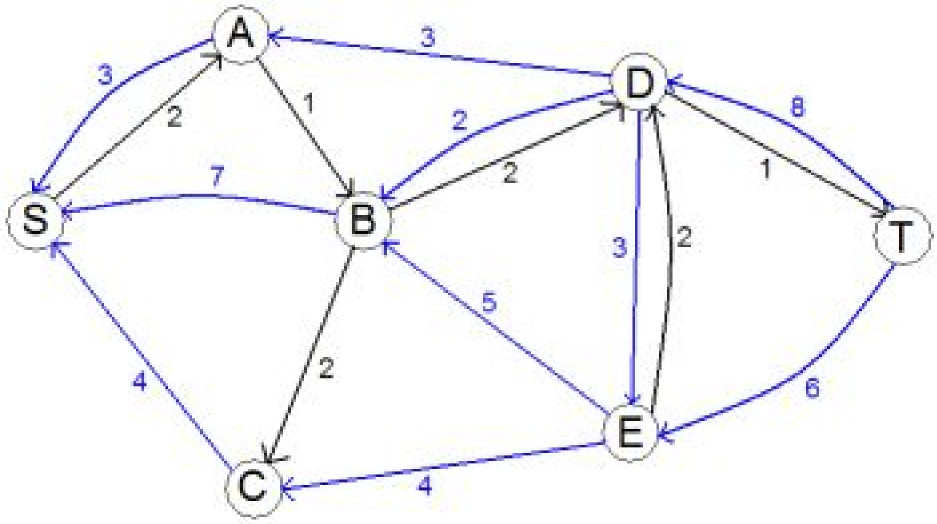
We choose the augmenting path SA, AD, DT with capacity 3 from the residual graph and change the flow values of the associated arcs: $f(SA) = f(AD) = 3, f(DT) = 8$. our new residual graph is as follows:



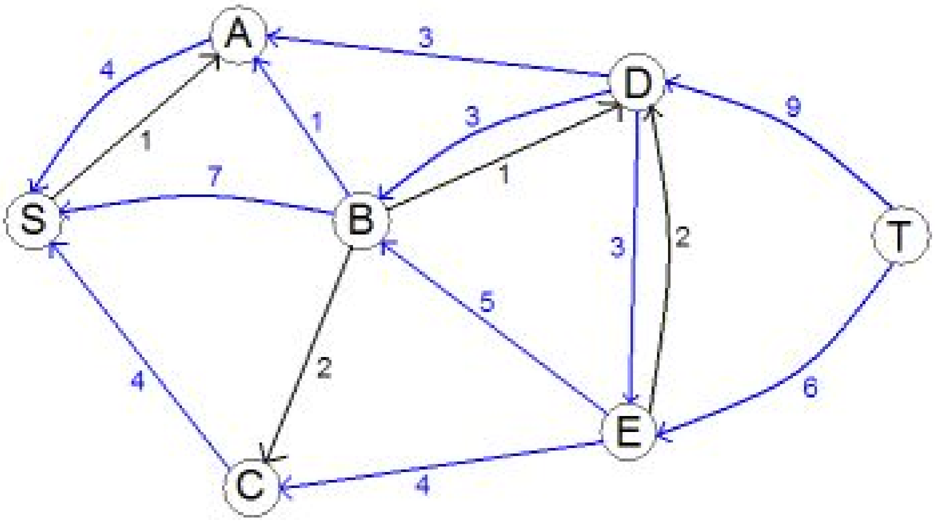
We choose the augmenting path SC, CE, ET with capacity 4. Then $f(SC) = f(CE) = f(ET) = 4$. Our new residual graph is as follows:



Now we choose the augmenting path SB, BD, DE, ET with capacity 2. Then $f(SB) = 7, f(BD) = 2, f(ED) = 3, f(ET) = 6$. Notice that in this case, we have actually *decreased* the flow on the arc ED, but in doing so we have increased the overall *value* of the flow. We get the following residual graph:



We choose the augmenting path SA, AB, BD, DT with capacity 1. Now $f(SA) = 4, f(AB) = 1, f(BD) = 3, f(DT) = 9$. The residual graph is as follows:



We see the the residual graph has no s,t-path, and conclude that our current flow is maximal. This flow has value 15. To find a cut whose capacity is equal to 15, we can take the cut $(\{S, A\}, \{B, C, D, E, T\})$ which has capacity $3 + 1 + 7 + 4 = 15$.

5 Further Reading

[1] David G. Luenberger, *Linear and Nonlinear Programming*, 2nd Edition, 1984
[2] F. Hillier, G. Lieberman, *Introduction to Operations Research*, McGraw Hill, 2001