1 Outline

In this lecture, we study

- the shortest path problem,
- maximum st-flow,
- minimum st-cut,
- the max-flow min-cut theorem,
- bipartite matching.

2 Shortest path problem

Given a directed graph $D = (N, A)$ and two distinct nodes $s, t \in N$, a **(directed) st-path** is a sequence of nodes $v_0,\ v_1,\ \ldots,\ v_\ell$ such that

- $v_0 = s$ and $v_\ell = t$,
- $v_0, \ldots, v_\ell$ are distinct nodes,
- $(v_{i-1}, v_i) \in A$ for $i = 1, \ldots, \ell$.

Here we often call s the **origin node** and t the **destination node**. We can define an st-path with arcs. A directed st-path can be defined as a sequence of arcs $a_1,\ \ldots,\ a_\ell$ such that

- $a_i = (v_{i-1}, v_i)$ for $i = 1, \ldots, \ell$ for some nodes $v_0, \ldots, v_\ell$,
- $v_0 = s$ and $v_\ell = t$,
- $v_0, \ldots, v_\ell$ are distinct nodes.

In general, a (directed) path is an st-path where s and t are the first and the last nodes in the path. Let c_{uv} be the length of arc $(u, v) \in A$. Then the length of a path P is

$$\sum_{(u,v)\in P} c_{uv}$$

where $(u, v) \in P$ means that arc (u, v) is on the path P. Now the problem is to find a shortest st-path, that is, a directed st-path of the minimum length.

We will show that the problem of finding a shortest st-path can be posed as an instance of the minimum cost flow problem. Let $x_{uv} \in \{0, 1\}$ denote the variable for arc (u, v) to indicate whether

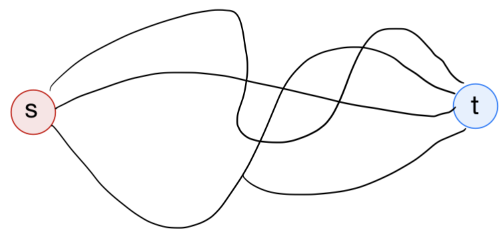

Figure 13.1: st-path as a unit flow

arc $(u,v) \in A$ is chosen to be part of my path. Then we may look at $x \in \{0,1\}^A$ whose components correspond to the arc set A. If x corresponds to the arc set of an st-path, then

$$\sum_{(u,v)\in A} c_{uv}x_{uv}$$

would be the length of the path.

When does a 0,1 vector $x \in \{0,1\}^A$ correspond to an st-path? Observe the following.

- The origin node s has an outgoing arc on the path. No other arc of the path is incident to s. We may model this as
$$\sum_{j\in N:(s,j)\in A} x_{sj} - \sum_{k\in N:(k,s)\in A} x_{ks} = 1.$$
- The destination node t has an incoming arc on the path. No other arc of the path is incident to t. We may model this as
$$\sum_{j\in N:(t,j)\in A} x_{tj} - \sum_{k\in N:(k,t)\in A} x_{kt} = -1.$$
- Let $i \in N \setminus \{s,t\}$. If i is on the path, then i has an incoming arc and an outgoing arc on the path. No other arc is incident to i. If i is not on the path, then no arc of the path is incident to i. This implies that the number of arcs going into i and the number of arcs going out of i are the same. This can modeled as
$$\sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{k\in N:(k,i)\in A} x_{ki} = 0.$$

Therefore, an st-path can be viewed as the source node s sending one unit of flow to the sink node t. More precisely, the origin node s has supply 1, and the destination node t has demand 1. The other nodes have 0 net supply, meaning that they are transhipment nodes. Then the problem can be formulated as

$$\begin{aligned}\min \quad & \sum_{(u,v)\in A} c_{uv}x_{uv} \\ \text{s.t.} \quad & \sum_{j\in N:(s,j)\in A} x_{sj} - \sum_{k\in N:(k,s)\in A} x_{ks} = 1 \\ & \sum_{j\in N:(t,j)\in A} x_{tj} - \sum_{k\in N:(k,t)\in A} x_{kt} = -1 \\ & \sum_{j\in N:(i,j)\in A} x_{ij} - \sum_{k\in N:(k,i)\in A} x_{ki} = 0, \quad \forall i \in N \setminus \{s,t\} \\ & x_{ij} \in \mathbb{Z}_+, \quad \forall (i,j) \in A\end{aligned}$$

The formulation is an instance of the minimum cost flow formulation. Therefore, solving this linear program will return a solution x^* that has integer entries only, which corresponds to a shortest st-path.

A **directed cycle** is a sequence of nodes $v_0,\ v_1,\ \ldots,\ v_\ell$ such that

- $v_0, \ldots, v_{\ell-1}$ are distinct nodes,
- $v_\ell = v_0$.

Remark 13.1. If D contains a directed cycle of negative length, then the linear program is unbounded. If D contains no directed cycle of negative length, then the linear program would have an optimal solution.

3 Maximum st-flow

The minimum cost flow model we learned does not have a designated source or a sink. In this section, we discuss a network flow model with a **sink node** and a **source node**. Let s and t be the source node and the sink node, respectively. The source node s sends flows, and the sink node receives the flows sent by the source. The other nodes are transhipment node, meaning that the othder nodes have 0 net supply. Each arc in the given network has an upper bound on the amount of flowws that it can take, i.e.

$$0 \le x_{ij} \le c_{ij}, \quad (i,j) \in A.$$

Then the problem is to compute the maximum amount of flows that the source node s can send to the sink node t while obeying the flow capacities of arcs.

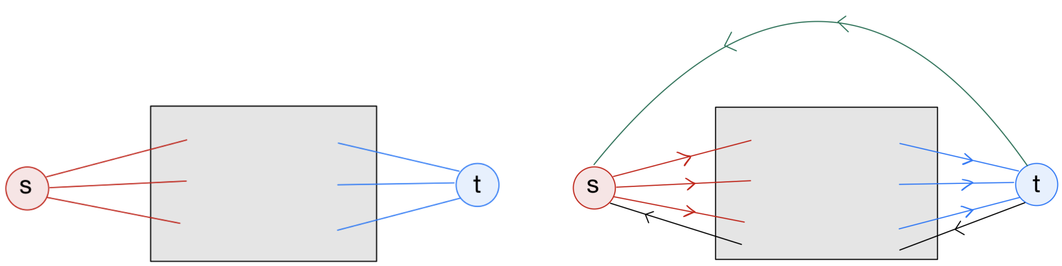

Figure 13.2: Sending flow from s to t

Although this problem seems different from the minimum cost flow problem, we may formulate the problem as a min cost flow model. The common trick is to add a dummy arc from the sink node t to the source node s. This dummy arc (t,s) sends back all the flows coming from s to t. Basically, we impose that

$$x_{ts} = \underbrace{\sum_{k\in N:(k,t)\in A} x_{kt} - \sum_{j\in N:(t,j)\in A} x_{tj}}_{\text{the net amount of flows into } t}.$$

Moreover, $A' = A \cup \{(t,s)\}$ is the arc set of the new network obtained after adding the dummy arc (t,s). Then

$$\begin{aligned} 0 &= x_{ts} + \sum_{j\in N:(t,j)\in A} x_{tj} - \sum_{k\in N:(k,t)\in A} x_{kt} \\ &= \underbrace{\sum_{j\in N:(t,j)\in A'} x_{tj} - \sum_{k\in N:(k,t)\in A'} x_{kt}}_{\text{the net amount of flows into } t \text{ in the new network}} \end{aligned}.$$

Furthermore, the amount of flows that the sink node t receives is equal to the amount of flows that the source node s sends out. Hence, we have

$$\underbrace{\sum_{j\in N:(s,j)\in A} x_{sj} - \sum_{k\in N:(k,s)\in A} x_{ks}}_{\text{the net amount of flows out of } s} = \sum_{k\in N:(k,t)\in A} x_{kt} - \sum_{j\in N:(t,j)\in A} x_{tj} = x_{ts}$$

Then it follows that

$$\begin{aligned} 0 &= \sum_{j\in N:(s,j)\in A} x_{sj} - \sum_{k\in N:(k,s)\in A} x_{ks} - x_{ts} \\ &= \underbrace{\sum_{j\in N:(s,j)\in A'} x_{sj} - \sum_{k\in N:(k,s)\in A'} x_{ks}}_{\text{the net amount of flows out of } s \text{ in the new network}} \end{aligned}.$$

The other nodes in the network are transhipment nodes and are not connected to the dummay arc (t,s), so we have

$$\sum_{j\in N:(i,j)\in A'} x_{ij} - \sum_{k\in N:(k,i)\in A'} x_{ki} = 0, \quad i \in N \setminus \{s,t\}.$$

Then the problem can be formulated as

$$\begin{aligned} \max \quad & x_{ts} \\ \text{s.t.} \quad & \sum_{j\in N:(i,j)\in A\cup\{(t,s)\}} x_{ij} - \sum_{k\in N:(k,i)\in A\cup\{(t,s)\}} x_{ki} = 0, \quad \forall i \in N \\ & 0 \le x_{ij} \le c_{ij}, \quad \forall (i,j) \in A. \end{aligned}$$

Observe that the dummy arc x_{ts} is a free variable, which is equivalent to $-\infty \le x_{ts} \le +\infty$. As this formulation is an instance of the minimum cost flow model, it returns an integer flow as long as the capacities c_{ij} for $(i,j) \in A$ are integers.

4 Minimum st-cut

A **directed** st**-cut** is a set of arcs of the form

$$\delta^+(S) = \{(u,v) \in A : \ u \in S, \ v \notin S\}$$

where $S \subseteq N$ contains s but not t. In words, $\delta^+(S)$ is the set of arcs going out of the node set S. Given arc weights c_{uv} for $(u,v) \in A$, the weight of an st-cut $\delta^+(S)$ is given by

$$\sum_{(u,v)\in\delta^+(S)} c_{uv}.$$

Then the minimum st-cut problem is to find an st-cut whose weight sum is minimized.

We may formulate the minimum st-cut problem as an integer program.

- For nodes s and t, we assign integer variables $y_s, y_t \in \mathbb{Z}_+$ that satisfy

  $$y_s = 0 \quad \text{and} \quad y_t = 1.$$

  Then y_s and y_t satisfy

  $$y_t - y_s = 1.$$

- For $i \in N \setminus \{s,t\}$, we assign an integer variable $y_i \in \mathbb{Z}_+$. We set

  $$y_i = \begin{cases} 0, & \text{if } i \in S, \\ 1, & \text{if } i \notin S. \end{cases}.$$

- For each arc $(i,j) \in A$, we assign variable z_{ij} to indicate whether arc (i,j) is part of the st-cut. We can model this by adding

  $$z_{ij} \geq y_j - y_i.$$

  Here, if $y_j = 1$ and $y_i = 0$, then $i \in S$ and $j \notin S$, which implies that (i,j) is part of $\delta^+(S)$.

Then we deduce

$$\begin{aligned} \min \quad & \sum_{(i,j)\in A} c_{ij} z_{ij} \\ \text{s.t.} \quad & z_{ij} \geq y_j - y_i, \quad (i,j) \in A \\ & y_t - y_s = 1 \\ & z_{ij} \in \mathbb{Z}_+, \quad (i,j) \in A. \end{aligned}$$

Although z_{ij} can take an arbitrary nonnegative integer, if the arc lengths c_{ij} for $(i,j) \in A$ are all nonnegative, then an optimal solution would have $z_{ij} \in \{0,1\}$ for $(i,j) \in A$.

5 The max-flow min-cut theorem

Let us rewrite the LP relaxation of the integer program for solving the minimum st-cut problem.

$$\begin{aligned} \min \quad & \sum_{(i,j)\in A} c_{ij} z_{ij} \\ \text{s.t.} \quad & y_i - y_j + z_{ij} \geq 0, \quad (i,j) \in A \\ & y_t - y_s = 1 \\ & z_{ij} \geq 0, \quad (i,j) \in A. \end{aligned}$$

In fact, the dual of this linear program is precisely

$$\begin{aligned} \max \quad & x_{ts} \\ \text{s.t.} \quad & \sum_{j\in N:(i,j)\in A\cup\{(t,s)\}} x_{ij} - \sum_{k\in N:(k,i)\in A\cup\{(t,s)\}} x_{ki} = 0, \quad \forall i \in N \\ & 0 \leq x_{ij} \leq c_{ij}, \quad \forall (i,j) \in A, \end{aligned}$$

which is the linear program for solving the maximum st-flow problem. Based on the duality relationship, we may observe the following.

1. The constraint matrix for the maximum st-flow formulation is totally unimodular. Then the transpose of it is the constraint matrix for the mimum st-cut formulation. This implies that the LP relaxation of the minimum st-cut formulation has an optimal solution that has integer entries only, which corresponds to a minimum st-cut.

2. The primal optimum is the minimum weight of an st-cut, and the dual optimum is the maximum amount of an st-flow. By strong LP duality, the two values are equal.

Theorem 13.2 (The max-flow min-cut theorem)**.** *Let $D = (N, A)$ be a directed graph with arc weights $c_{ij} \in \mathbb{R}_+$ for $(i,j) \in A$, and let $s, t \in N$ be two distinct nodes. Then the maximum amount of an st-flow is equal to the minimum weight of an st-cut.*