IE 331 OR: Optimization **KAIST, Spring 2023**
Lecture #14: Maximum *st*-flow and bipartite matching April 13, 2023
Lecturer: Dabeen Lee

1 Outline

In this lecture, we cover

- the maximum *st*-flow problem,
- bipartite matching.

2 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.

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

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.

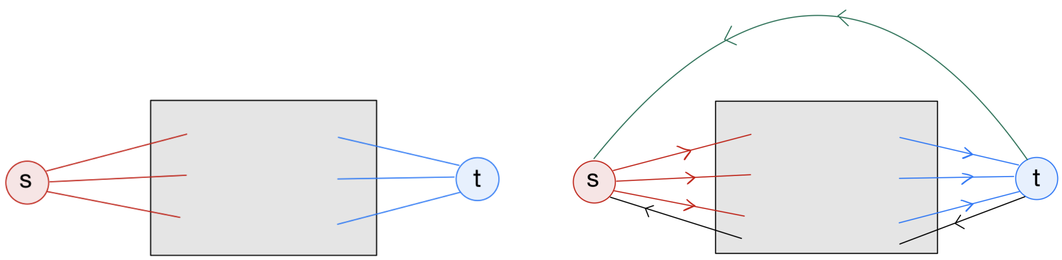

Figure 14.1: Sending flow from s to t

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

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

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

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

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.

3 Bipartite matching

A **bipartite graph** is a graph $G = (V, E)$ where

- the vertex set V is partitioned into two sets V_1 and V_2,
- each edge $e \in E$ crosses the partition, i.e. e has one end in V_1 and the other end in V_2.

For example, Figure 14.2 shows a bipartite graph on 7 vertices where one set contains 3 and the other has 4. A **matching** is a set of edges without common vertices. In Figure 14.2, the set of

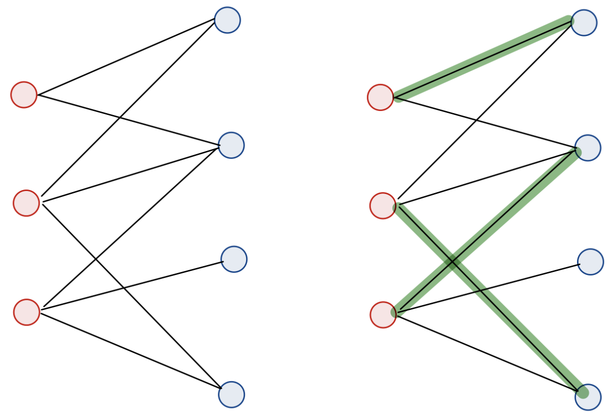

Figure 14.2: Bipartite graph and a matching

green edges gives rise to a matching. The **matching problem** is to find a matching that has the maximum number of edges.

The first approach is to reduce bipartite matching to maximum *st*-flow. Given a bipartite graph $G = (V, E)$ with V partitioned into V_1 and V_2, we run the following transformation procedure.

- Add a source node s and a sink node t.
- Add arcs from s to all vertices in V_1: $\{(s, u) : \ u \in V_1\}$.
- Add arcs to t from all vertices in V_2: $\{(v, t) : \ v \in V_2\}$.
- Direct every edge (u, v) where $u \in V_1$ and $v \in V_2$ so that (u, v) becomes an arc from u to v.
- Set the flow upper bound c_{uv} of every arc (u, v) to 1.

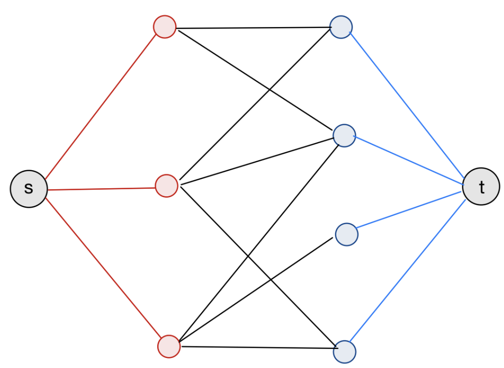

Figure 14.3: Reducing a bipartite graph to a flow newtwork

Then the following linear program computes a maximum *st*-flow over the above network.

$$\begin{aligned} \max \quad & \sum_{u\in V_1} x_{su} \\ \text{s.t.} \quad & \sum_{v\in V_2:(u,v)\in E} x_{uv} - x_{su} = 0, \quad u \in V_1 \\ & x_{vt} - \sum_{u\in V_1:(u,v)\in E} x_{uv} = 0, \quad v \in V_2 \\ & 0 \le x_{su}, x_{vt}, x_{uv} \le 1, \quad (u,v) \in E \end{aligned}$$

In particular, there is an optimal solution x^* that has integer entries only. As each component of x^* is between 0 and 1, we may select

$$M = \{(u,v) \in E : \ x^*_{uv} = 1\}.$$

Note that

$$\sum_{v\in V_2:(u,v)\in E} x^*_{uv} = x^*_{su} \le 1.$$

Therefore, u is connected to at most one edge in M. Similarly,

$$\sum_{u\in V_1:(u,v)\in E} x^*_{uv} = x^*_{vt} \le 1.$$

Therefore, v is connected to at most one edge in M. This implies that M is a matching. In fact, $|M|$ is the size of the matching, and moreover,

$$|M| = \sum_{u\in V_1} x^*_{uv}.$$

This implies that we have just solved bipartite matching by maximum *st*-flow.