1 Outline

In this lecture, we study

- bipartite matching,
- uncapacitated lot-sizing problem

2 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 14.1 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 14.1, the set of

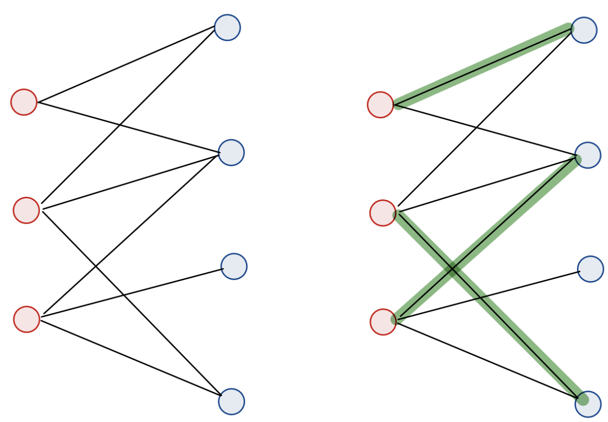

Figure 14.1: Bipartite graph and a matching

green edges gives rise to a matching.

Suppose that each edge $e \in E$ has a weight w_e. Given a set of edges F, the weight of F is defined as the sum of weights of the edges in F, given by,

$$\sum_{e \in F} w_e.$$

The **matching problem** is to find a matching that has the maximum weight.

2.1 Reduction to maximum st-flow for the unweighted case

We first consider the unweighted case, i.e., $w_e = 1$ for $e \in E$. The approach for the unweighted case 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.

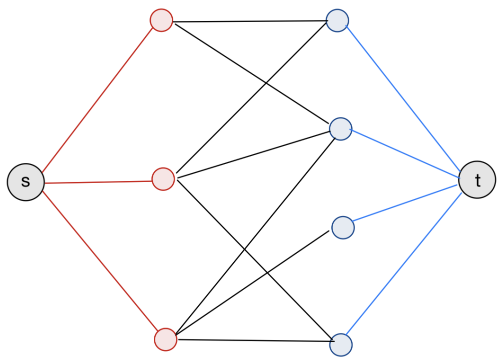

Figure 14.2: Reducing a bipartite graph to a flow newtwork

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.

2.2 Incidence-matrix-based formulation

Recall that the matching problem can be formulated as the following integer program.

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} x_e \leq 1, \quad \forall v \in V, \\ & x_e \in \{0,1\}, \quad \forall e \in E \end{aligned}$$

where for $v \in V$, $\delta(v) = \{e \in E : \text{ one end of } e \text{ is } v\}$. We may represent the integer program in matrix form. Let A be the **vertex-edge incidence matrix** of G. Here, A is defined as

$$a_{v,e} = \begin{cases} 1, & \text{if } v \text{ is adjacent to } e, \\ 0, & \text{otherwise.} \end{cases}$$

For example, consider the following small bipartite graph. The vertex-edge incidence matrix of this

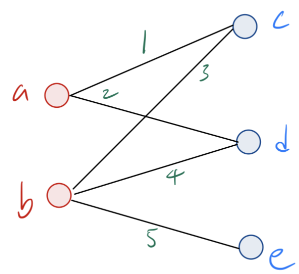

Figure 14.3: Bipartite graph on 5 vertices

graph is given by Table 1. Note that this matrix is totally unimodular! The partition $\{a, b\}$ and

	1	2	3	4	5
a	1	1			
b			1	1	1
c	1		1		
d		1		1	
e					1

Table 1: The vertex-edge incidence matrix of the bipartite graph on 5 vertices

$\{c, d, e\}$ naturally give an equitable bicoloring. In general,

Theorem 14.1. *The vertex-edge incidence matrix of a bipartite graph is totally unimodular.*

Furthermore, the integer programming formulation can be written as

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & Ax \leq 1, \\ & x_e \in \{0,1\}, \quad \forall e \in E \end{aligned}$$

where A is the incidence matrix of the bipartite graph G. As A is totally unimodular, solving its LP relaxation computes a maximum weight bipartite matching.

3 Uncapacitated lot sizing problem

We have n periods and a single product. The following lists problem parameters, costs and demand.

- d_t: demand in period t
- p_t: unit production cost in period t
- i_t: unit inventory cost in period t
- f_t: fixed set-up cost in period t

The following is the list of decision variables we use.

- x_t: amount produced in period t
- s_t: inventory at the end of period t ($s_0 = 0$)
- y_t: variable to indicate production in period t, i.e.

$$y_t = \begin{cases} 1, & \text{if production occurs in period } t \\ 0, & \text{otherwise} \end{cases}$$

Then, the following formulation describes the problem.

$$\begin{aligned} \min \quad & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} i_t s_t + \sum_{t=1}^{n} f_t y_t \\ \text{s.t.} \quad & s_{t-1} + x_t = d_t + s_t, \quad t \in [n] \\ & x_t \leq M_t y_t, \quad t \in [n] \\ & s, x \geq 0 \\ & y_t \in \{0,1\}, \quad t \in [n] \end{aligned}$$

We may eliminate the production variables via

$$x_t = d_t + s_t - s_{t-1}.$$

Let

$$h_t = i_t + p_t - p_{t-1}.$$

Then we obtain

$$\begin{aligned}
\min \quad & \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} f_t y_t \\
\text{s.t.} \quad & d_t + s_t - s_{t-1} \le M_t y_t, \quad t \in [n] \\
& s \ge 0 \\
& y_t \in \{0,1\}, \quad t \in [n]
\end{aligned} \tag{14.1}$$

Next we give another formulation. For $j \ge t$, let

$$\delta_{tj} = \begin{cases} 1, & \text{if no production in periods } t \text{ to } j \\ 0, & \text{otherwise} \end{cases}$$

Consider the second formulation

$$\begin{aligned}
\min \quad & \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} f_t y_t \\
\text{s.t.} \quad & s_{t-1} \ge \sum_{j=t}^{n} d_j \delta_{tj}, \quad t \in [n] \\
& \delta_{tj} \ge 1 - \sum_{i=t}^{j} y_i, \quad t \in [n] \\
& \delta_{tj} \ge 0, \quad t \in [n] \\
& y_t \in \{0,1\}, \quad t \in [n]
\end{aligned} \tag{14.2}$$

Assumption 1 (Wagner-Whitin cost assumption)**.** : $h_t = i_t + p_t - p_{t-1} \ge 0$ for all $t \in [n]$.

As $h_t \ge 0$ for all $t \in [n]$, we may use

$$s_{t-1} \ge \sum_{j=t}^{n} d_j \delta_{tj}$$

to eliminate s_t.

$$\begin{aligned}
\min \quad & \sum_{t=1}^{n} h_t \sum_{j=t+1}^{n} d_j \delta_{(t+1)j} + \sum_{t=1}^{n} f_t y_t \\
\text{s.t.} \quad & \delta_{tj} \ge 1 - \sum_{i=t}^{j} y_i, \quad t \in [n] \\
& \delta_{tj} \ge 0, \quad t \in [n] \\
& y_t \in \{0,1\}, \quad t \in [n]
\end{aligned} \tag{14.3}$$

Theorem 14.2. *The constraint matrix of formulation* (14.3) *is totally unimodular.*

Proof. Matrix with consecutive 1's property in each row. Hence, the corresponding constraint matrix is totally unimodular by Ghouila-Houri's theorem (alternate Blue and Red columns). □