CMPUT 675: Topics in Combinatorics and Optimization **Fall 2016**

Lecture 16 (Oct 14): TUM Matrices and Network Matrices

Lecturer: Zachary Friggstad *Scribe: Arnoosh Golestanian*

In this lecture, our goal is to show that the constraint matrix of some problems such as multiple source-sink maximum flow problem is totally unimodular matrix. A large class of TUM matrices is discussed: network matrices. Finally, we state without proof the Ghouila-Houri theorem that provides an equivalent characterization of TUM matrices.

16.1 Bipartite Matching

A graph $G = (V; E)$ is shown in Figure 16.1. The vertex set and edge set of G is $\{1, 2, 3, 4, 5, 6\}$ and $\{a, b, c, d, e, f, g, h\}$, respectively. Graph G is a bipartite graph since the vertices of G can be coloured by two colours, red (vertices 1, 4, 5) and blue (vertices 2, 3, 6).

A linear programming relaxation for bipartite matching is:

$$\begin{array}{lll} \text{maximize} & \sum_{e \in E} c_e x_e & \\ \text{subject to} & & \\ & x(\delta(v)) \leq 1 & \forall\ v \in V \\ & x_e \geq 0 & \forall\ e \in E \end{array}$$

In the constraint matrix A, entry $A_{v,e}$ will be 1, if edge e is incident to vertex v and 0 otherwise.

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0. \end{pmatrix}$$

We will prove such matrices are TUM in this lecture. For an example, consider the cycle $1, 2, 4, 3$ and the submatrix A' of A whose rows are indexed by the vertices of the cycle and columns indexed by the edges of the cycle. Then adding the rows for $1, 4$ and subtracting the rows for $2, 3$ leaves the $\mathbf{0}$ row, so $\det A' = 0$.

16.2 Network Flow

The following is the linear programming formulation for the maximum flow problem. The constraint matrix of this problem is essentially the same as constraint matrix of bipartite matching except we negate entries for edges uv in row u. That is, for each edge e going from vertex u to v, $A_{u,e} = -1$ and $A_{v,e} = 1$.

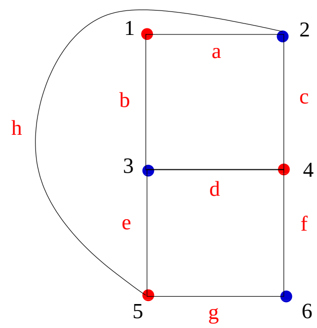

Figure 16.1: Example of Bipartite Matching

$$
\begin{array}{lll}
\text{maximize} & x(\delta^{out}(s)) - x(\delta^{in}(s)) & \\
\text{subject to} & & \\
& x(\delta^{in}(v)) - x(\delta^{out}(v)) = 0 & \forall\ v \in V - \{s,t\} \\
& x_e \leq \mu_e & \forall\ e \in E \\
& x_e \geq 0 & \forall\ e \in E
\end{array}
$$

We will also see the constraint matrix of this LP is totally unimodular. If so, then we have another proof of the fact that if the capacities are integers, then there is a maximum flow that is integral.

16.3 Network Matrices

All of the above examples are in fact very simple cases of a more general class of TUM matrices called network matrices. To define a network matrix, we consider:

- A tree $T = (V; E)$ with $|V| \geq 2$ where each $e \in E$ is in fact directed in some direction (so the undirected version is a tree).
- A nonempty collection of pairs of nodes $\mathcal{P} = \{(s_1, t_1), ..., (s_k, t_k)\}$ in T.

We insist $|V| \geq 2$ and $k \geq 1$ above to ensure the following matrix has a nonzero number of columns and rows.

For each pair $(s_i, t_i) \in \mathcal{P}$ and each edge e lying on the unique $s_i - t_i$ path in T, we say e lies **positively** on this path if it is pointing toward t_i, otherwise we say e lies **negatively** on this path.

We now define a matrix $A \in \{-1, 0, +1\}^{E(T),[k]}$ based on the pair $(T, \mathcal{P})$. Here $[k]$ means set $\{1, 2, ..., k\}$. Specifically, $A_{e,i}$ is $+1$ if e appears positively on the $s_i - t_i$ path, -1 if it appears negatively and 0 otherwise.

We say A this is a **network matrix** and that the combinatorial object $(T, \mathcal{P})$ **presents** this matrix.

Theorem 1 *A network matrix is TUM.*

Proof. We prove this by induction on $|V|$. The base case is when $|V| = 2$. In this case, there is only one edge so only one row. Thus, all square submatrices have size 1 and the definition of A ensures each entry lies in $\{-1, 0, +1\}$.

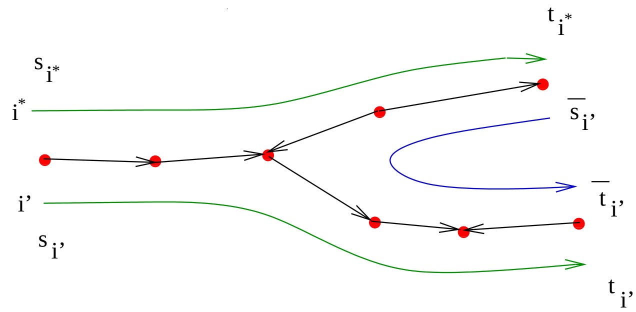

Figure 16.2: Example of Tree T and smaller tree T'

Inductively, suppose any network matrix presented by a pair with at most $|V| - 1$ nodes is a network matrix. Let u be some leaf node and let $e = uv$ be the corresponding edge. We assume e is directed from u to v (i.e. $e \in \delta^{out}(u)$). Otherwise, flip the orientation of e which negates e's row in A. This only negates the determinant of any submatrix including this row.

Let B be a submatrix of A with size $k \times k$. If B does not contain row e, then B is a submatrix of another network matrix presented by:

$$T' = (V - u, E - e), \quad P' = \{(s'_i, t_i) : i = 1, 2, ..., k\}$$

where,

$$s'_i = \begin{cases} v & \text{if } s_i = u \\ s_i & \text{otherwise.} \end{cases}$$

By induction, $\det B \in \{-1, 0, 1\}$.

So, assume B contains row e. Let $\mathcal{Q}$ denote all pairs $(s_i, t_i) \in \mathcal{P}$ having either $s_i = u$ or $t_i = u$. If some pair has $t_i = u$ then swap s_i and t_i in this pair, which corresponds to negating the column for pair i (thus, only negating the determinant of any submatrix including this column).

If $\mathcal{Q} = \emptyset$ then the row of B corresponding to e is $\mathbf{0}$ so $\det B = 0$.

Otherwise, let i^* be any index such that $B_{e,i^*} = 1$. Let submatrix B' be obtained from B by subtracting column i^* in B from all other columns i' with $B_{e,i'} = 1$. Since this is just subtracting one column from some other columns, then $\det B' = \det B$.

Note row e of B' has only 0s apart from $B'_{e,i^*} = 1$. Also note that if $i' \neq i^*$ was such that $B_{e,i'} = 1$ then the column of $B'_{e,i'}$ corresponds to the pair $(t_{i^*}, t_{i'})$. See Figure 16.2, where the new pair is denoted $(\overline{s_{i'}}, \overline{t_{i'}})$.

Now consider the network matrix presented by the tree $T' = (V - u, E - e)$ with pairs

$$\mathcal{P}' = (\mathcal{P} - \mathcal{Q}) \cup \{(t_{i^*}, t_{i'}) : (s_{i'}, t_{i'}) \in \mathcal{Q} - (s_{i^*}, t_{i^*})\}.$$

That is, replace each pair in $\mathcal{Q}$ with the new path (from $\overline{s_{i'}}$ to $\overline{t_{i'}}$) depicted in Figure 16.2.

Let $B'^{[e,i^*]}$ be the submatrix of B' obtained by deleting row e and column i^*. By construction, $B'^{[e,i^*]}$ is a square submatrix of the network matrix presented by $(T', \mathcal{P}')$. So its determinant is either $-1, 0$ or $+1$. Thus,

$$\det B = \det B' = \pm \det B'^{[e,i^*]}$$

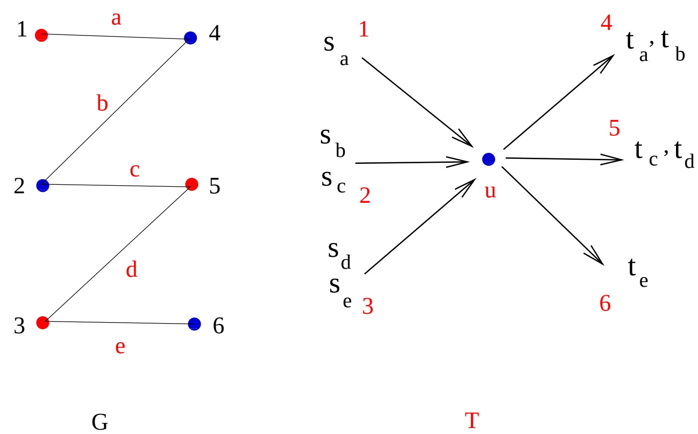

Figure 16.3: Converting graph G to tree T

(where we recall row e of B' has its only nonzero entry appearing in column i^*).

■

16.4 Use of network matrix

Now we can show that the constraint matrix of the bipartite matching relaxation $(x(\delta(v)) \leq 1, \forall v \in V)$ is TUM.

To show this we construct tree T from graph G as shown in Figure 16.3. As you can see, $V(T) = V(G) \cup \{u\}$ and we add new pair (s_e, t_e) in T for each edge e of G.

It can be verified that the network matrix of T is equivalent to the constraint matrix of bipartite matching of G.

In a similar way, we can present the constraint matrix of the network flow problem as a network matrix. Consider the bipartite graph with a copy of V on each side and an edge uv for each directed edge uv in the flow network. Use the same tree T for this bipartite graph as above, except reverse the direction of all edges on the right side. This shows the part of the constraint matrix for the flow-conservation constraints is TUM. To add the capacity constraints, just add a new edge to T for each edge $e = uv$ in the flow network that points from a new node v_e (specific to e) into the copy of u on the left and move the start of the pair (s_e, t_e) to start at v_e.

16.5 Gouila-Houri Theorem

We just state this theorem, without proof. The proof is in the KV textbook and is definitely accessible at the level of this course, if you are curious.

Theorem 2 *Let $A \in \mathbb{R}^{m \times n}$. Then A is totally unimodular if and only if for every subset of row indices $R \subseteq \{1, \ldots, m\}$, we can partition R into two sets $R_1 \dot\cup R_2$ such that*

$$\sum_{i \in R_1} A_i - \sum_{i' \in R_2} A_{i'} \in \{-1, 0, +1\}^n.$$

For example, consider the vertex-edge incidence matrix of a bipartite graph $G = (V_L \cup V_R; E)$. Recall this is $A \in \mathbb{R}^{V_L \cup V_R; E}$ where $A_{v,e} = 1$ if v is an endpoint of e, and 0 otherwise. For any $R \subseteq V_L \cup V_R$, if we set $R_1 = R \cap V_L$ and $R_2 = R \cap V_R$ then we find $\sum_{v \in R_1} A_v - \sum_{v \in R_2} A_v \in \{-1, 0, +1\}$ because every column of A has one 1 from rows in V_L and one 1 from rows in V_R.