Totally Unimodular Matrices

$$\sum_{j=1}^{n} a_{ij}x_j \geq b_j \quad \text{for } 1 \leq i \leq m$$

m constraints, n variables

Vertex solution: unique solution of n linearly independent tight inequalities

$$\sum_{j=1}^{n} a_{ij}x_j = b_j \quad \text{for } 1 \leq i \leq n$$

Can be rewritten as:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

That is:

$$Ax = b$$

Totally Unimodular Matrices

Assuming all entries of A and b are integral

When does $Ax = b$ has an integral solution x?

By Cramer's rule

$$Ax = b \iff x = A^{-1}b \iff \forall i : x_i = \frac{det(A^i)}{det(A)}$$

where A^i is the matrix with each column is equal to the corresponding column in A except the i-th column is equal to b.

x would be integral if det(A) is equal to +1 or -1.

Totally Unimodular Matrices

A matrix $A \in Z^{m \times n}$ is **totally unimodular** if the determinant of each square submatrix of A is 0, -1, or +1.

Theorem 1: If A is totally unimodular,
then every vertex solution of $Ax \geq b$ is integral.

Proof (follows from previous slides):

1. a vertex solution is defined by a set of n linearly independent tight inequalities.
2. Let A' denote the (square) submatrix of A which corresponds to those inequalities.
3. Then A'x = b', where b' consists of the corresponding entries in b.
4. Since A is totally unimodular, det(A) = 1 or -1.
5. By Cramer's rule, x is integral.

Example of Totally Unimodular Matrices

A totally unimodular matrix must have every entry equals to +1,0,-1.

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{Guassian elimination} \quad \longrightarrow \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 - b_2 + b_3 \\ b_2 - b_3 \\ b_3 \end{pmatrix}$$

And so we see that x must be an integral solution.

Example of Totally Unimodular Matrices

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

is not a totally unimodular matrix, as its determinant is equal to 2.

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \rightarrow \quad \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}$$

x is not necessarily an integral solution.

Totally Unimodular Matrices

$$\min \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_j$$

y_i

$$x_j \geq 0$$

$$\max \sum_{i=1}^{m} b_i y_i$$

$$\sum_{i=1}^{m} y_i a_{ij} \leq c_j$$

$$y_i \geq 0$$

Primal

$$\min c \cdot x$$

$$Ax \geq b$$

$$x \geq 0$$

$$\max b \cdot y$$

$$A^T y \leq c \quad \longleftarrow$$

$$y \geq 0$$

Dual

Transpose of A

Theorem 2: If A is totally unimodular,

then both the primal and dual programs are integer programs.

Proof: if A is totally unimodular, then so is it's transpose.

Application 1: Bipartite Graphs

Let A be the incidence matrix of a bipartite graph.

Each row i represents a vertex v(i),

and each column j represents an edge e(j).

A(ij) = 1 if and only if edge e(j) is incident to v(i).

edges

v_1 v_2 v_3

e_2 e_3

e_1

e_5

e_4

e_6

v_4 v_5 v_6
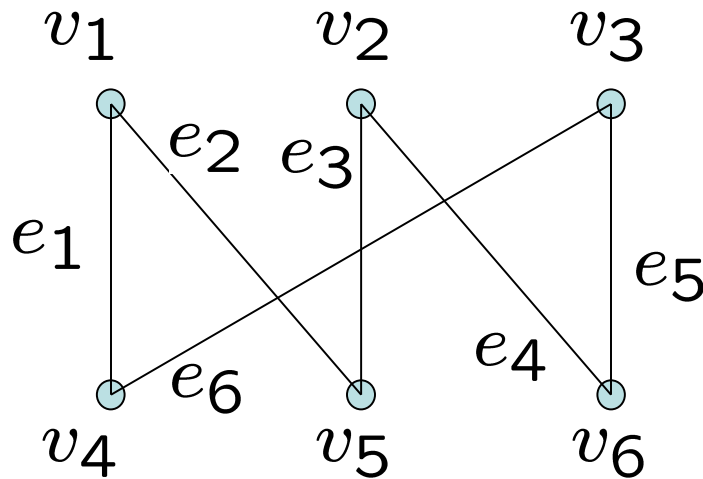
vertices

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Application 1: Bipartite Graphs

We'll prove that the incidence matrix A of a bipartite graph is totally unimodular.

Consider an arbitrary square submatrix A' of A.
Our goal is to show that A' has determinant -1,0, or +1.

Case 1: A' has a column with only 0.   Then det(A')=0.

Case 2: A' has a column with only one 1.

$$A' = \begin{pmatrix} 1 & a^T \\ 0 & A'' \end{pmatrix}$$

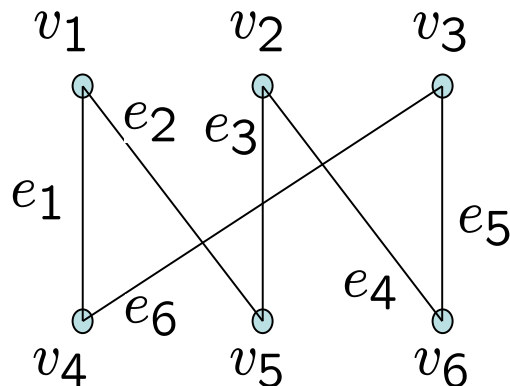By induction, A'' has determinant -1,0, or +1.
And so does A'.

Application 1: Bipartite Graphs

Case 3: Each column of A' has exactly two 1.

We can write $A' = \begin{pmatrix} A^{up} \\ A^{down} \end{pmatrix}$ +1 -1

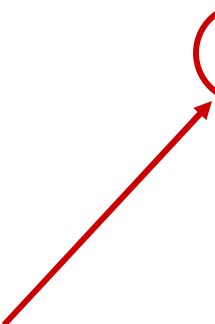Since the graph is bipartite, each column has one 1 in A^{up} and one 1 in A^{down}

So, by multiplying +1 on the rows in A^{up} and -1 on the columns in A^{down},

we get that the rows are linearly dependent, and thus det(A')=0, and we're done.

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{ +1} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{ -1}$$

Application 1: Bipartite Graphs

Maximum bipartite matching

$$\max \sum_{e \in E(G)} x_e$$

$$\sum_{e \in \delta(v)} x_e \leq 1$$

$$x_e \geq 0$$

$$\max \quad x \cdot \mathbf{1}$$

$$Ax \leq \mathbf{1}$$

$$x_e \geq \mathbf{0}$$

Incidence matrix of a bipartite graph,

hence totally unimodular,

and so yet another proof that this LP is integral.

Application 1: Bipartite Graphs

$$\max \sum_{e \in E(G)} x_e$$

$$\sum_{e \in \delta(v)} x_e \leq 1$$

$$x_e \geq 0$$

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$
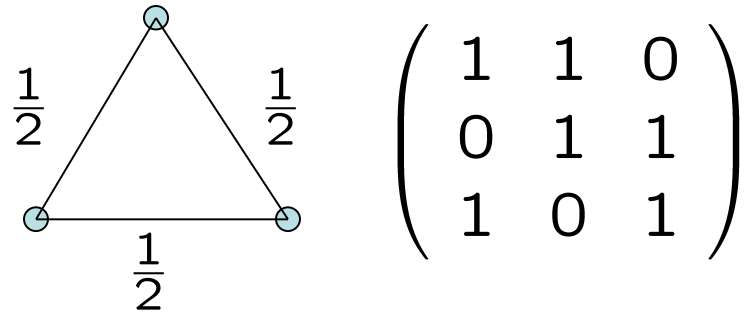
The linear program for general matching does not come from a totally unimodular matrix, and this is why Edmonds' result is regarded as a major breakthrough.

Application 1: Bipartite Graphs

$$\max \sum_{e \in E(G)} x_e \qquad \min \sum_{v \in V(G)} y_v$$

$$y_v \rightarrow \sum_{e \in \delta(v)} x_e \leq 1 \qquad \sum_{\{u,v\}=e} y_u + y_v \geq 1$$

$$x_e \geq 0 \qquad y_v \geq 0$$

Theorem 2: If A is totally unimodular,

then both the primal and dual programs are integer programs.

Maximum matching <= maximum fractional matching <= minimum fractional vertex cover <= minimum vertex cover

Theorem 2 show that the first and the last inequalities are equalites.

The LP-duality theorem shows that the second inequality is an equality.

And so we have maximum matching = minimum vertex cover.

Application 2: Directed Graphs

Let A be the incidence matrix of a directed graph.

Each row i represents a vertex v(i),
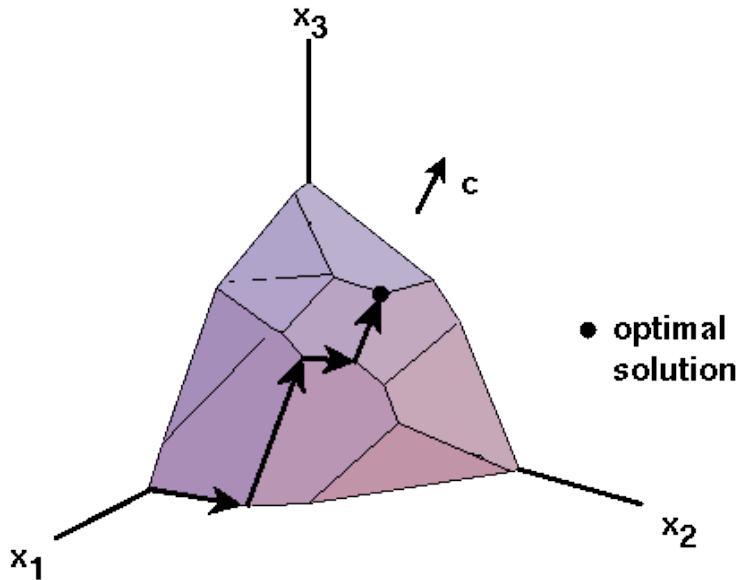
and each column j represents an edge e(j).

- A(ij) = +1 if vertex v(i) is the tail of edge e(j).

- A(ij) = -1 if vertex v(i) is the head of edge e(j).

- A(ij) = 0 otherwise.

The incidence matrix A of a directed graph is totally unimodular.

Consequences:

(2) The max-flow problem (even min-cost flow) is polynomial time solvable.

(3) Max-flow-min-cut theorem follows from the LP-duality theorem.

Simplex Method

Simplex Algorithm:
- Start from an arbitrary vertex.
- Move to one of its neighbours which improves the cost. **Iterate**.

For combinatorial problems, we know that vertex solutions correspond to combinatorial objects like matchings, stable matchings, flows, etc.

So, the simplex algorithm actually defines a combinatorial algorithm for these problems.

Simplex Method

For example, if you consider the bipartite matching polytope and run the simplex algorithm, you get the augmenting path algorithm.

The key is to show that two adjacent vertices are differed by an augmenting path.

Recall that a vertex solution is the unique solution of n linearly independent inequalities. So, moving along an edge in the polytope means to replace one tight inequality by another one. There is one degree of freedom and this corresponds to moving along an edge.

Summary

How to model a combinatorial problem as a linear program.

See the geometric interpretation of linear programming.

How to prove a linear program gives integer optimal solutions?

Prove that every vertex solution is integral.
- ✓ By convex combination method.
- ✓ By linear independency of tight inequalities.
- ✓ By totally unimodular matrices.
- ✓ By shifting technique.

Polynomial Time Solvable Problems

Stable matchings

Bipartite matchings

Weighted Bipartite matchings

General matchings

Maximum flows

Minimum spanning trees

Shortest paths

Graph orientation

Minimum Cost Flows

Matroid intersection

Submodular Flows

Packing directed trees

Connectivity augmentation

Linear programming