6 Integer Programs

6.1 Introduction • 6.2 Total Unimodularity • 6.3 Integrality Theorems • 6.4 Matching Polytopes

6.1 Introduction

We will give a simplified and not-too-rigorous introduction to polytopes. Most of the statements about polytopes are also true for polyhedra, but often there are some subtleties or weird exceptions. Since most of the optimization problems that we'll see correspond to polytopes, we'll restrict ourselves to those.

Definitions:
• A *polyhedron* is a set $P \subset \mathbb{R}^n$ of the form $P = \{x : Ax \leq b\}$, for an $m \times n$ matrix A and $b \in \mathbb{R}^m$.
• A *polytope* is a bounded polyhedron. It can be shown that this is equivalent to being the convex hull of a finite set of points.
• A *vertex* of a polytope P is a point $z \in P$ for which there is a hyperplane H such that $P \cap H = z$. Here a hyperplane is a set $\{x : \sum a_i x_i = c\} \subset \mathbb{R}^n$, where not all a_i are 0.
One can prove that a polytope is the convex hull of its vertices.

Lemma 6.1. *For any vertex z of a polytope, there are a nonsingular $n \times n$ submatrix A_z of A and a subvector b_z of b such that z is the unique point satisfying $A_z z = b_z$.*

We won't prove this here. To explain it a bit, observe that rows of A correspond (typically) to hyperplanes that make up the boundary of of the polytope, and a vertex is an intersection of n (or more) such hyperplanes; these correspond to n independent rows of A, which together form A_z. But there are some exceptions that one would have to deal with.
• A polytope is *integral* if all its vertices are integral.
• The *integer set* P_I of a polytope $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is

$$P_I = \{x \in \mathbb{Z}^n : Ax \leq b\},$$

i.e. the points contained in P that have integer coordinates. Clearly we have $P_I \subseteq P$. A polytope P is integral if and only if the convex hull of P_I equals P.

Integer programs: Consider an integer program and its relaxation; their constraints define a polyhedron P; let's assume it is a polytope. This polytope equals the set of solutions of the relaxation, and its integer set P_I is the set of solutions to the integer program. Among the optimal solutions of the linear program there must be vertices.
If the polytope is integral, then the vertices among its optimal solutions are integral, hence they are also optimal solutions of the integer program. So not every optimal solution of the integer program is integral, but if there is an optimal solution there must be an integral one.

Integrality theorems: Below we will prove the integrality theorems that we used for our algorithms in the first few sections, which roughly stated that optimal solutions of a particular integer program were the same as optimal solutions of the relaxation, so we could use the relaxation and what we know about linear programs to solve the combinatorial optimization problem. Now we can say this more precisely: the polytope defined by the relaxation was integral.

The fact that we've been able to find exact polynomial algorithms for most of the problems we've seen so far has a lot to do with their polytopes being integral. After all, we know that there are polynomial algorithms to solve linear programs, so in a roundabout way this gives a polynomial algorithm for any integer program whose polytope is integral. Unfortunately, for many more complicated optimization problems this is not the case, and we will soon see examples of that.

6.2 Total Unimodularity

For some (but not all) optimization problems, we can tell directly from the matrix of constraints in its integer program whether its polytope is integral. In the next section we will use this to prove our integrality theorems.

A matrix is called *totally unimodular* if each square submatrix has determinant $0, 1$ or -1. (a square matrix A is *unimodular* if $|\det(A)| = 1$, but we won't use that here).

Theorem 6.2. *Let $P = \{x : Ax \leq b\}$ be a polytope. If A is totally unimodular and b is integral, then P is integral.*
The same is true for the polytopes $\{x : Ax \leq b, x \geq 0\}$, $\{x : Ax = b, x \geq 0\}$, and any variation of these with $Ax \geq b$ instead of $Ax \leq b$.

Proof. We'll only prove the first case here. The other cases can be shown to follow from the first. For instance, $\{x : Ax \leq b, x \geq 0\}$ is of the first form if one writes it as

$$\begin{bmatrix} A \\ -I \end{bmatrix} x \leq \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Then one just has to show that this larger matrix is also totally unimodular.
So let z be a vertex of $\{x : Ax \leq b\}$, and let A be of size $m \times n$. By the Lemma in the introduction, there is a nonsingular submatrix A_z of A and a subvector b_z of b such that z is the unique point satisfying $A_z z = b_z$. Hence $z = A_z^{-1} b_z$.
As A is totally unimodular, the same is true for A_z, and since it is square and nonsingular we have $\det(A_z) = \pm 1$. By Cramer's rule, each entry of A_z^{-1} is the determinant of a submatrix of A_z divided by $\det(A_z)$, so A_z^{-1} is an integer matrix. Since b_z is assumed integral, it follows that the vertex z is integral. $\square$

Given an undirected graph $G = (V, E)$, we define its *incidence matrix* to be the $|V| \times |E|$ matrix A with rows corresponding to vertices, and columns corresponding to edges, and entries

$$A_{ve} = \begin{cases} 1 & \text{if } v \in e, \\ 0 & \text{if } v \notin e. \end{cases}$$

Theorem 6.3. *If A is the incidence matrix of a graph G, then A is totally unimodular if and only if G is bipartite.*

Proof. Suppose G is not bipartite. This implies that it has an odd cycle, and one can check that the determinant of the incidence matrix of an odd cycle is 2. This is a submatrix of A, so A is not totally unimodular.

Suppose G is bipartite. Let B be a square submatrix of A. We show by induction on the size of B that its determinant equals $0, 1$ or -1. The base case, when B has size 1×1, is obvious. We distinguish three cases (which cover all possibilities, since an incidence matrix has at most two 1's in a column).

- B has a column with only 0's: Then $\det(B) = 0$.
- B has a column with exactly one 1: Then we can write, after permuting rows and columns:

$$B = \begin{pmatrix} 1 & \cdots \\ \overline{0} & B' \end{pmatrix},$$

where $\overline{0}$ denotes a column of 0's, and B' is a smaller matrix. Then $\det(B) = \det(B')$, and by induction $\det(B') \in \{0, \pm 1\}$.

- Every column of B has exactly two 1's: Since G is bipartite, we can write (after permuting rows)

$$B = \begin{pmatrix} B' \\ B'' \end{pmatrix},$$

in such a way that each column of B' contains exactly one 1 and each column of B' contains exactly one 1. If the partite sets of G are U and V, this is done by letting B' correspond to the vertices of U and B'' to the vertices of V. Then adding up the rows in B' gives the vector with all 1's, and so does adding up the rows in B''. But that means the rows are linearly dependent, so $\det(B) = 0$. $\square$

Given a directed graph $G = (V, E)$, we define its *incidence matrix* to be the $|V| \times |E|$ matrix A with rows corresponding to vertices, and columns corresponding to edges, and entries

$$A_{ve} = \begin{cases} 1 & \text{if } v = t(e), \\ -1 & \text{if } v = h(e), \\ 0 & \text{else.} \end{cases}$$

(Recall that if $e = uv$, then $u = t(e)$ is the tail and $v = h(e)$ is the head.)

Theorem 6.4. *If A is the incidence matrix of a directed graph, then A is totally unimodular.*

Proof. Almost identical to that for bipartite graphs, with some minuses thrown in. $\square$

6.3 Integrality Theorems

Bipartite matchings

First we'll look at the linear program that we used for bipartite weighted matchings. Note that it is the relaxation of the integer program which exactly described matchings.

LP for bipartite max weight matchings

$$\text{maximize} \sum_{e \in E(G)} w_e x_e \quad \text{with } x \geq 0,$$
$$\sum_{e \ni v} x_e \leq 1 \quad \text{for } v \in V.$$

The polyhedron for this linear program is really a polytope, because $0 \leq x_e \leq 1$ for all e.

Theorem 6.5. *If G is bipartite, then the polytope defined by the linear program above is integral.*

Proof. The constraints are exactly $Ax \leq \overline{1}$, where A is the incidence matrix of G. Then A is totally unimodular because G is bipartite, and the vector $\overline{1}$ is obviously integral, so the polytope is integral. □

Shortest Paths

Next we'll look at the relaxation of the integer program for shortest paths. Note that the integer program did not exactly describe paths, because non-optimal solutions could involve cycles, but its minima were exactly the shortest paths.

LP for shortest ab-path

$$\text{minimize} \sum w_e x_e \quad \text{with } 0 \leq x \leq 1,$$
$$\sum_{e \in \delta^{\text{in}}(v)} x_e - \sum_{e \in \delta^{\text{out}}(v)} x_e = 0 \quad \text{for } v \in V \backslash \{a, b\},$$
$$\sum_{e \in \delta^{\text{in}}(b)} x_e = 1, \qquad - \sum_{e \in \delta^{\text{out}}(a)} x_e = -1.$$

We're cheating a little bit, because in the lecture on shortest path, we did not include the constraint $x \leq 1$. This made it easier to dualize, but without it the associated polyhedron is not a polytope (on those extraneous cycles the x_e could be arbitrarily large). It's not hard to prove that this doesn't make a difference, as minimal solutions will satisfy $x \leq 1$ anyway.

Theorem 6.6. *The polytope defined by the linear program for shortest paths is integral.*

Proof. The matrix is almost, but not quite, the incidence matrix of the directed graph (actually, with all entries multiplied by -1, but that's no problem). Indeed, most edges occur twice in the constraints, once with a $+$ in the constraint for its head, and once with a $-$ in the constraint for its tail. These correspond to the $+1$ and -1 in the column of this edge.

The only exceptions are edges that have b as a tail or a as a head, which occur only once, and have only one $+1$ or -1 in their column. If we remove the corresponding columns from the matrix, we have a submatrix of the incidence matrix of the graph. So this submatrix is also totally unimodular. Adding the removed columns back in one by one, the total unimodularity is preserved, because these columns have one 1 and the rest 0's.

So the matrix in the LP for shortest paths is totally unimodular, and the vector b is clearly integral, therefore the corresponding polytope is integral. □

Flows
Note that in the lecture on flows, we didn't actually need integrality, because flows were defined as real functions on the edges, so a maximum flow was by definition a solution of a linear program, not of an integer program.
It is still true that when the capacities are integral, then the vertices of the corresponding polytope are integral, and there is a maximum flow that is integral. But this doesn't directly follow from total unimodularity, because the capacities are in the constraint matrix, and they need not be in $\{0, \pm 1\}$.
One can get around this, but it's actually easier to just prove integrality using the augmenting path algorithm. So we won't do it here.

Trees
We only briefly mentioned the integer program for trees, and we didn't need it at all. To prove integrality for it, total unimodularity would not be not much help. But trees are a special case of matroids, which we will see soon, and probably we will then see, and prove in a different way, the integrality theorem for matroid polytopes.

6.4 Matching Polytopes

Matching polytopes: Given a graph G, define its *matching polytope* $\mathcal{M}(G)$ to be the convex hull of the incidence vectors of matchings in G (vectors $(x_e)_{e\in E}$ such that $x_e = 1$ for matching-edges and $x_e = 0$ for non-matching-edges).
Similarly, define its *perfect matching polytope* $\mathcal{PM}(G)$ to be the convex hull of the incidence vectors of perfect matchings in G.
So unlike the previous polytopes, these are defined as convex hulls of finite sets of points, and the goal is to find a set of inequalities that exactly describe them. For bipartite graphs, we have already found those, but for non-bipartite graphs we haven't.

Bipartite graphs: As we just saw, when G is bipartite, then the polytope

$$P = \{x : Ax \le 1, x \ge 0\}$$

is integral, where A is the incidence matrix of G. This means that P equals the convex hull of P_I, which equals $\mathcal{M}(G)$, so $P = \mathcal{M}(G)$.
Similarly, we could show that if G is bipartite then $\mathcal{PM}(G) = \{x : Ax = 1, x \ge 0\}$.

Non-bipartite graphs: For non-bipartite graphs, $P \ne \mathcal{M}(G)$ and $P' \ne \mathcal{PM}(G)$. Indeed, if a graph is not bipartite, then it contains an odd cycle. On an odd cycle there cannot be a perfect matching, but there can be a "fractional" perfect matching: If $x_e = 1/2$ for each e, then $\sum_{e\ni v} x_e = 1$ for each v. Also, the value $\sum x_e$ of this solution will be $|C|/2$, which is larger than the value of the maximum cardinality matching, which is $(|C| - 1)/2$.
So although the obvious integer program still describes all matchings for nonbipartite graphs, its relaxation is inadequate. However, there is a *different* integer program for matchings, which has more constraints, whose relaxation does do the job. We will show this for perfect matchings, and we'll state the relaxation right away.

LP for general max weight perfect matchings

$$\text{maximize} \sum_{e\in E(G)} w_e x_e \quad \text{with } x \geq 0,$$
$$\sum_{e\in\delta(v)} x_e = 1 \quad \text{for } v \in V,$$
$$\sum_{e\in\delta(S)} x_e \geq 1 \quad \text{for } S \subset V(G),\ |S| \text{ odd.}$$

Note that any perfect matching does indeed satisfy the added constraints: For any odd set S of vertices, at least one vertex from S must be matched to a vertex outside S, which means there is a matching edge in $\delta(S)$.
Also note that these constraints are broken by the fractional example given above: If $x_e = 1/2$ for all e in an odd cycle C, then $\sum_{e\in\delta(C)} x_e = 0$.
Geometrically, here's what's happening: The first polytope may have non-integral vertices which have entries $1/2$ on odd cycles (we can prove that those are the only kind), and the extra constraints exactly cut those off, without cutting off any integral vertices, and without creating any new fractional vertices. That's what the following theorem states.
Note that although this theorem shows that one can use linear programming to find perfect matchings, this actually does not directly give a polynomial algorithm, because the number of constraints in the program is exponential.
We will only sketch proof of this theorem, because it would be quite long. But this outline does show that the proof pulls together the observation that we've made so far, and also that there is a vague connection with shrinking blossoms.

Theorem 6.7. *For any graph G, the polytope defined by the linear program above is integral, and equals the perfect matching polytope $\mathcal{PM}(G)$.*

Proof. (sketch) Let $\widetilde{x}$ be a vertex. Then by Lemma 6.1, there is a subset of the constraints for which $\widetilde{x}$ is the unique solution with equality in each of these constraints. More precisely, there is $\widetilde{E} \subset E$, and a set $\mathcal{W}$ of odd $S \subset V$ with $|S| \geq 3$, such that $\widetilde{x}$ is the unique solution to

$$x_e = 0\ \forall e \in \widetilde{E}, \quad \sum_{e\in\delta(S)} x_e = 1\ \forall S \in \mathcal{W}, \quad \sum_{e\in\delta(v)} = 1\ \forall v \in V.$$

First suppose that $\mathcal{W} \neq \emptyset$, and pick any $S \in \mathcal{W}$. Then we can shrink S to a vertex to get a graph G_1, and we can shrink $V\backslash S$ to a vertex to get a graph G_2. One can check (but we won't) that $\widetilde{x}$ gives two vertices $\widetilde{x}_1$ and $\widetilde{x}_2$ of the corresponding polytopes for G_1 and G_2. Since these are smaller, by induction $\widetilde{x}_1$ and $\widetilde{x}_2$ are perfect matchings (actually, convex combinations), and one can check that these can be combined into a maximum weight perfect matching in G which corresponds to $\widetilde{x}$. In particular, $\widetilde{x}$ is integral.
We still have to consider the case $\mathcal{W} = \emptyset$, which also functions as a base case for the induction. In this case, $\widetilde{x}$ is also a vertex of the polytope $\{x : Ax \leq 1, x \geq 0\}$. We claim, but will not prove, that such a vertex x is almost integral, in the sense that $x_e \in \{0, 1, \frac{1}{2}\}$, and the edges e for which $x_e = \frac{1}{2}$ form odd cycles. In other words, the counterexample that we saw above, an odd cycle, is basically the only kind of counterexample.
But $\widetilde{x}$ cannot have such an odd cycle C with weights $1/2$, since it would violate the constraint for $S = C$. Hence all $\widetilde{x}_e \in \{0, 1\}$ and $\widetilde{x}$ corresponds to a perfect matching. $\square$