15-859E: Advanced Algorithms **CMU, Spring 2015**
Lecture #10: Linear Programs, and the Perfect Matching Polytope February 4, 2015
Lecturer: Anupam Gupta Scribe: Colin White

In this lecture, we will start by reviewing basic concepts and definitions for linear programming. Then we will discuss a linear program for the minimum perfect matching problem in bipartite and non-bipartite graphs.

1 Linear Programming

We start with some basic definitions.

Definition 10.1. A *polyhedron* in $\mathbb{R}^n$ is the intersection of a finite number of half spaces.

A polyhedron is a convex region which satisfies some number of linear constraints. A polyhedron in n dimensions with m constraints is often written compactly as $K = \{Ax \leq b\}$, where A is an m by n matrix of constants, x is an n by 1 vector of variables, and b is an m by 1 vector of constants.

Definition 10.2. A *polytope* $K \in \mathbb{R}^n$ is a polyhedron such that $\exists R > 0$ where $K \subseteq B(\mathbf{0}, R)$.

In other words, a polytope is a bounded polyhedron. Now we can define a linear program in terms of a polyhedron.

Definition 10.3. For some integer n, a polyhedron K, and an n by 1 vector c, a *linear program* in n dimensions is

$$\text{minimize} \sum_{i=1}^{n} c_i x_i \text{ subject to } \bar{x} \in K$$

We can also have linear programs that maximize some objective function. Just flip the sign of all components of c. Also note that K need not be bounded to have a solution. For example, the following linear program has a solution even though the polyhedron is unbounded:

$$\min\{x_1 + x_2 \mid x_1 + x_2 \geq 1\}. \tag{10.1}$$

Now we will present three different definitions about types of points that may appear in a polytope.

Definition 10.4. Given a polytope K, a point $x \in K$ is an *extreme point of* K if there do not exist $x_1, x_2 \in K$, $x_1 \neq x_2$, and $\lambda \in [0, 1]$, such that $x = \lambda x_1 + (1 - \lambda) x_2$.

In other words, an extreme point of K cannot be written as the average of two other points in K. See Figure 10.1 for an example.

Now we move to another definition about points in K.

Definition 10.5. A point $x \in K$ is a *vertex of* K if there exists an n by 1 vector $c \in \mathbb{R}^n$ such that $c^\intercal x < c^\intercal y$ for all $y \neq x$, $y \in K$.

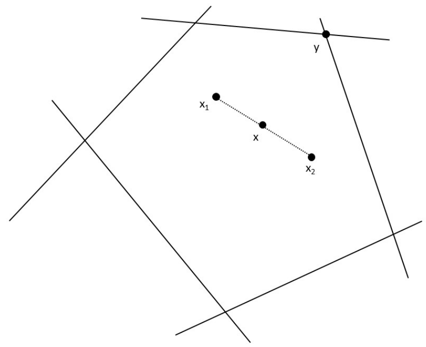

Figure 10.1: y is an extreme point, but x is not.

So, a vertex is the unique optimizer for some objective function. Note that there may be a linear program that does not have any vertices, as in Equation 10.1. Any assignment to x_1 and x_2 such that $x_1 + x_2 = 1$ minimizes $x_1 + x_2$, but none of these are strictly better than other points on that line. And no other objective function has a minimum in $x_1 + x_2 \geq 1$.

Now we consider one last definition about points in K.

Definition 10.6. Given a polytope $K \in \mathbb{R}^n$, a point $x \in K$ is a *basic feasible solution (bfs) to K* if there exist n linearly independent constraints in K which x satisfies at equality.

For example, let $K = \{a_i^\intercal x \leq b_i\}$ such that all constraints are linearly independent. Then x^* is a basic feasible solution if there exist n values of i such that $a_i^\intercal x^* = b_i$, and for the other values of i, $a_i^\intercal x^* \leq b_i$ (x^* must satisfy all constraints because it is in K).

As you may have guessed by now, the last three definitions are all related. In fact, the following fact shows they are all equivalent.

Fact 10.7. *Given a polyhedron K, and a point $x \in K$. Then the following are equivalent:*

1. *x is a basic feasible solution,*
2. *x is an extreme point, and*
3. *x is a vertex.*

The proof is straightforward, and we will not present it here. Now we will show the main fact for this section.

Fact 10.8. *For a polytope K and an LP*=$\min\{c^\intercal x \mid x \in K\}$*, there exists an optimal solution $x^* \in K$ such that x^* is an extreme point/vertex/bfs.*

This fact suggests an algorithm for LPs when K is a polytope: just find all of the extreme points/vertices/bfs's, and pick the one that gives the minimum solution. There are only $\binom{m}{n}$ vertices to check in K, where m is the total number of constraints and n is the dimension (because we pick n constraints out of m to make tight).

Note that Fact 10.8 can be proven with weaker conditions than K being a polytope, but in this lecture, we will stick to polytopes.

Also note that when the objective function is perpendicular to a constraint, then there could be infinitely many solutions, but Fact 10.8 just states that there exists one optimal solution that is an extreme point/vertex/bfs.

We finish off this section with one more definition, which will help us construct an LP for bipartite matching in the next section.

Definition 10.9. Given $x_1, x_2, \ldots, x_N \in \mathbb{R}^n$, the *convex hull* of $x_1, \ldots, x_n$ is

$$\text{CH}(x_1, \ldots, x_n) = \left\{ x \,\middle|\, \exists \lambda_1, \ldots, \lambda_N \text{ s.t. } \sum_{i=1}^{N} \lambda_i = 1,\ \lambda_i \geq 0,\ \text{and } x = \sum \lambda_i x_i \right\} \tag{10.2}$$

In words, the convex hull of points $x_1, \ldots, x_n$ is the intersection of all convex sets containing $x_1, \ldots, x_n$. From that description, it is easy to see that every convex hull is also a polytope. We also know the following fact:

Fact 10.10. *Given a polytope K, then $K = CH(x \mid x$ is an extreme point of $K)$.*

2 Bipartite Matchings

Now we go back to the problem of finding a min cost perfect matching (which we have considered in previous lectures). For now, we will stick to bipartite graphs $G = (L, R, E)$.

Let us denote the matchings in G as bit-vectors in $\{0, 1\}^{|E|}$ For example, see Figure 10.2.

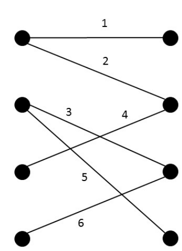

Figure 10.2: This graph has one perfect matching using edges 1, 4, 5, and 6, so we can represent it as $[1, 0, 0, 1, 1, 1]$.

This allows us to define an $|E|$ dimensional polytope which contains all perfect matchings. Let us try the obvious choice for such a polytope, and see if it gives an efficient LP:

$$C_{PM} = CH(x \in \{0, 1\}^{|E|} \mid x \text{ represents a perfect matching in } G).$$

The LP to find the min cost perfect matching of a bipartite graph with edge weights defined in vector c is

$$\min\{c^\mathsf{T} x \mid x \in C_{PM}\}.$$

Then the solution will be at a vertex of C_{PM}, which by construction represents a perfect matching.

This is great news, because we do not have to deal with a fractional solution to the LP. But, there is one problem. It is a huge pain to write down C_{PM}. Can we find a more compact way to write it down? Let's try the following idea.

$$K_{PM} = \left\{ x \in \mathbb{R}^{|E|} \middle| \forall l \in L, \sum_r x_{lr} = 1 \text{ and } \forall r \in R, \sum_l x_{lr} = 1 \text{ and } x \geq 0 \right\}$$

This polytope enforces that the weights of edges leaving every vertex is 1, so it seems plausible that it is a polytope for perfect matching. This would be much easier to use in an LP, so now we would like to show that K_{PM} is the same as C_{PM}.

Theorem 10.11. $K_{PM} = C_{PM}$.

We start with the easy direction, $C_{PM} \subseteq K_{PM}$. Define χ_M as the indicator function for the edges in a matching M.

Fact 10.12. $C_{PM} \subseteq K_{PM}$.

Proof. Clearly, for all perfect matchings M, $\chi_M \in K_{PM}$ since a perfect matching satisfies the constraints that an edge weight of 1 leaves every vertex. It follows that

$$C_{PM} = \text{CH}(\chi_M \mid M \text{ is a perfect matching}) \subseteq K_{PM}$$

□

Now we must show that $K_{PM} \subseteq C_{PM}$. It suffices to show that all extreme points/vertices/bfs's of K_{PM} belong to C_{PM}. We will prove this three ways, using the three definitions from the last section.

Proof. Extreme points:

Suppose x^* is an extreme point of K_{PM}. We must show that $x^* \in C_{PM}$. Let $\text{supp}(x^*)$ denote the edges for which $x^*_e \neq 0$. First we will prove that $\text{supp}(x^*)$ is acyclic.

Suppose that $\text{supp}(x^*)$ contains a cycle $x_1, x_2, \ldots, x_l$. Since the graph is bipartite, l is even. All of these vertices are in the support, so each have nonzero weight. Then there exists an ϵ such that for all x_i, the weight of x_i is $> \epsilon$.

Then we can create a new point $x^*_1 \in K$ by adding ϵ to the weight of each x_i where i is odd, and subtracting ϵ to the weight of each x_i, where i is even. Similarly, we define x^*_2 by adding ϵ from the even i's, and subtracting ϵ form the odd i's. But then $x^* = \frac{1}{2}x^*_1 + \frac{1}{2}x^*_2$, violating our assumption that x^* is an extreme point. See Figure 10.3.

Therefore, there are no cycles in the support of x^*. So there must be a leaf v in the support. Then the single edge leaving v must have weight 1. But this edge goes to another vertex u, and because x^* is in K_{PM}, then this vertex cannot have any other edges without violating its constraint. So u and v are a matched pair. Now take u and v out of the graph. In the remaining graph, there cannot be a cycle for the same reason as before, so we perform the same logic inductively to show that x^* is a perfect matching. Then $x^* \in C_{PM}$. □

Our second proof was covered in the last lecture.

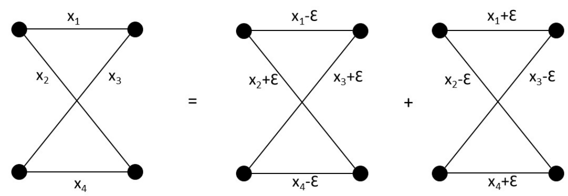

Figure 10.3: There cannot be a cycle in $\text{supp}(x^*)$, because this violates the assumption that x^* is an extreme point.

Proof. Vertices:

Suppose $x^* \in K_{PM}$ is a vertex of K_{PM}. Then it is an optimizer to some objective function. Recall that in the last lecture, we showed that any unique cost function must be a perfect matching. So $x^* \in C_{PM}$. □

Now we give a proof using the definition of bfs. Recall that K_{PM} contains $2n + m$ constraints: one constraint for each of the $2n$ vertices forcing the weight of edges leaving that vertex to sum to 1, and then m constraints for nonzero edge weights.

Proof. Basic feasible solutions:

Let x^* be a basic feasible solution of K_{PM}. Then there exist m linearly independent constraints in K_{PM} which are tight.

Assume that none of these tight constraints were a nonzero edge weight constraints. Then all the tight constraints were forcing the edge weights coming out of a vertex to be 1. However, these $2n$ constraints cannot all be linearly independent.

To see this, sum up all of the constraints for vertices in L. $\sum_l \sum_r x_{lr} = n$. But notice that this is exactly the same as summing up all of the constraints for vertices in R: $\sum_r \sum_l x_{lr} = n$. Therefore, these $2n$ constraints cannot be linearly independent. So at most $2n - 1$ of the linearly independent tight constraints in x^* can belong to the first $2n$ constraints.

Then $\geq m - (2n - 1)$ constraints from $x_{lr} \geq 0$ are tight. So $\geq m - (2n - 1)$ edges have $x_{lr} = 0$, so $|\text{supp}(x^*)| \leq 2n - 1$. It follows that there must be an edge with length 1. If we pull out this edge, we can inductively perform the same argument on the smaller graph, to show that x^* is a perfect matching. This completes the proof. □

3 Non-bipartite Matchings

Now we move to non-bipartite graphs. We define a polytope that is similar to the bipartite polytope. Let $x(\delta(v))$ denote the weight of all edges incident to v.

$$K_{PM} = \{x \in \mathbb{R}^m \mid \forall v \in V,\ x(\delta(v)) = 1, \text{ and } \forall e \in E,\ x_e \geq 0\}. \tag{10.3}$$

This is not a convex combination of all perfect matchings. For example, a triange graph with each edge weight $\frac{1}{2}$ will satisfy the constraints of this polytope.

So we need to add more constraints to K_{PM}. For a set of vertices S, let $x(\delta(S))$ denote the weight of all edges leaving S.

$$\{\forall S \text{such that } |S| \text{ is odd, } x(\delta(S)) \geq 1\}. \tag{10.4}$$

Now we claim the new K_{PM} is the correct polytope.

Theorem 10.13. $K_{PM} = CH($*all perfect matchings*$)$.

We give a sketch of the proof.

Let x^* be a basic feasible solution in K_{PM} (we would like to show that x^* is a perfect matching). So there are m linearly independent tight constraints. If there already exists an edge such that $x_e^* = 0$, then drop e, and argue about $G \setminus \{e\}$, i.e., G without edge e. If $x_e^* = 1$, then drop $e = (u, v)$, and induct on $G \setminus (u, v)$. Then after this process, for all v, there exist at least two edges in $\text{supp}(v)$. If all vertices have support degree 2, then there must be a cycle. This will cause a contradiction, as we saw in proof 1 for bipartite graphs. Therefore, there exists a vertex with degree ≥ 3 in the support. But then the number of edges in the support is greater than the number of vertices. From this, we can show there is at least one $x(\delta(S)) \geq 1$ constraint that is tight. Call it S^*.

Take S^*, and shrink it down to one vertex. Then we induct on S^* itself, and on the remaining graph with the single vertex as S^*.