Math 233B: Polyhedral techniques in combinatorial optimization
Instructor: Jan Vondrák
Lecture date: 1/10/2017 **Original scribe: Inbal Talgam-Cohen**

In this lecture we introduce the subject of the course, and review linear programming and duality.

1 Introduction

In combinatorial optimization problems, the optimization is over a combinatorial (discrete) set, e.g. all spanning trees of a graph. A problem typically looks like this:

$$\max\{w(F) : F \in \mathcal{F}\}$$

where w is some weight function. Often, combinatorial optimization problems can be formulated as linear programming problems, where the optimization is over a polyhedron:

$$\max\{w^T x : x \in P(\mathcal{F})\}$$

where $P(\mathcal{F})$ is the *polyhedral characterization* of the problem. The LP fomulation is typically easy to solve and is more robust to variations in the problem definition (e.g., optimize over all bounded-degree spanning trees instead of all spanning trees). P-time solvability usually goes hand-in-hand with a "nice polyhedral description", although this is certainly not always the case.

In Part 1 of the course we will cover classical, P-time solvable problems such as matching (bipartite and non-bipartite), spanning trees and matroids (including matroid intersection). We will *not* discuss flows and cuts. In Part 2 of the course we will discuss some more recent topics: the methods of iterated rounding and dependent randomized rounding, with some applications. We also plan to cover some recent lower bounds on the "extension complexity" of certain polytopes — informally, the impossibility of making an LP more compact by introducing additional variables. More on this later.

2 Review of Linear Programming

A linear program is a problem of the form:

$$\max\{c^T x : x \in \mathbb{R}^n, Ax \leq b\}$$

where $c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$. While the mathematical machinery applies to data in $\mathbb{R}$, we will actually assume rational data for computational purposes. Geometrically, let $P = \{x : Ax \leq b\}$ be a *polyhedron* (or a *polytope* if P is bounded), then the LP problem is to find a point $x \in P$ which is as far as possible in the direction c.

Definition 1 (Vertex) *$v \in P$ is a* vertex *of P if $\{v\} = P \cap H$ for some hyperplane $H = \{x : w^T x = \lambda\}$ such that $w^T x \leq \lambda$ for all $x \in P$.*

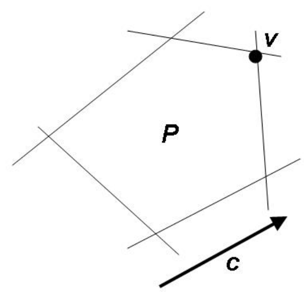

Figure 1: A polytope P with vertex v; we optimize in direction $\vec{c}$.

Definition 2 (Convex Hull) *We define the* convex hull *of a set of points A to be*

$$\text{conv}(A) := \{x : x = \sum_{a \in A} \alpha_a a \textit{ for some } \alpha_a \geq 0, \sum_{a \in A} \alpha_a = 1\}$$

Fact 1 *Every polytope is a convex hull of its vertices.*

Fact 2 *For every polytope P and vector c, $\max\{c^T x : x \in P\}$ is attained at a vertex.*

Intuitively, from a geometric viewpoint it's not hard to see that going in direction $\vec{c}$ until stuck always leads to a vertex.

2.1 LP Duality

With every *primal* LP $\max\{c^T x : Ax \leq b\}$ we can associate a *dual* LP $\min\{b^T y : y \geq 0, A^T y = c\}$. Note that the objective of the primal appears in the RHS of the dual and vice versa, and also that the number of variables in the primal is the number of constraints in the dual and vice versa. There are simple rules for obtaining the dual constraints from the primal variables:

$x_i \in \mathbb{R}$ $\rightsquigarrow$ equality constraint
$x_i \in \mathbb{R}_+$ $\rightsquigarrow$ inequality constraint ($\geq$ for min LP and $\leq$ for max LP)
$x_i \in \mathbb{R}_-$ $\rightsquigarrow$ reverse inequality constraint ($\geq$ for max LP and $\leq$ for min LP)

Theorem 3 (Strong LP Duality) *If either the primal or dual LP have a finite optimum, then*

$$\max\{c^T x : Ax \leq b\} = \min\{b^T y : y \geq 0, A^T y = c\}.$$

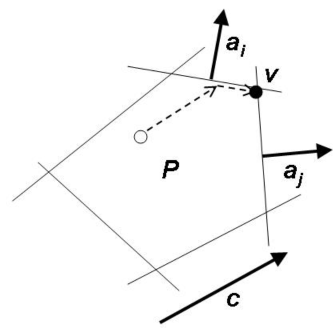

Figure 2: Cancellation of forces at optimum.

In Section 2.1.1 below we give an intuitive explanation as to why strong duality holds. We now describe a consequence of strong duality called *complementary slackness*. We begin with the following theorem, which follows directly from the feasibility constraints of the primal and dual LPs.

Theorem 4 (Weak LP Duality) *For every pair of primal and dual feasible solutions x, y:*

$$c^T x = (A^T y)^T x = y^T Ax \leq y^T b = b^T y.$$

Note that the rightmost inequality uses the dual condition $y \geq 0$.

By Theorem 4, $c^T x \leq b^T y$. When does it hold that $c^T x = b^T y$? By strong duality, this holds if x, y are optimal solutions, and by weak duality this is the only possible case. Furthermore, if $c^T x = b^T y$ then $y^T Ax = y^T b$. We can rewrite this as $\sum_{j \in [m]} y_j((Ax)_j - b_j) = 0$, and again using that $y \geq 0$ we conclude that $((Ax)_j - b_j) = 0$ for every j such that $y_j > 0$. The following theorem summarizes the above discussion.

Theorem 5 (Complementary Slackness) *For every pair of primal and dual feasible solutions x, y, the pair of solutions is optimal iff for every j such that $y_j > 0$, $(Ax)_j = b_j$.*

2.1.1 Physical "proof" of strong duality and complementary slackness

Push a particle along $\vec{c}$ until it gets stuck at point x by constraints $Ax \leq b$. So x is a primal optimal solution (we're assuming here that the optimum is finite). Let a_j be the j'th row of matrix A. If the particle gets stuck by constraint $a_j^T x \leq b_j$, then this constraint is pushing the particle in direction $-\vec{a_j}$ with a certain force that we denote by $y_j(-\vec{a_j})$ (note that $y_j \geq 0$, because the constraint $a_j^T x \leq b_j$ can only push the particle in the direction of decreasing $a_j^T x$). By cancellation of force:

$$\vec{c} + \sum_j y_j(-\vec{a_j}) = \vec{0}.$$

Equivalently, in matrix notation we write $A^T y = c$. Thus, y is a dual feasible solution. Since only tight constraints can push the particle, $y_j > 0$ implies $(Ax)_j = b_j$, i.e. complementary slackness holds. This means that $c^T x = y^T Ax = y^T b$, proving strong duality.

2.2 Polarity

We will now discuss a different kind of geometric duality.

Definition 6 (Polar Polyhedron) *Let $P = \{x : Ax \leq b\}$, then the* polar polyhedron *of P is:*

$$P^* = \{y : \forall x \in P, x^T y \leq 1\}$$

The polar polyhedron has the same dimension as the original. Constraints (facets) of the polar correspond to vertices of the original and vice versa.

Fact 3 *Let $P = \{x : Ax \leq b\}$ and let $\vec{a_i}$ be the i'th row of A. If P has 0 in its interior (i.e., if $b > 0$), then $P^* = \mathrm{conv}(\{\frac{1}{b_i}\vec{a_i} : i \in [m]\})$ and $(P^*)^* = P$.*

Example 1 *The polar polyhedron of a hypercube $\{x : \forall 1 \leq i \leq n, |x_i| \leq 1\}$ is a* cross-polytope *$\{\pm\vec{e_i} : 1 \leq i \leq n\}$. In particular, the polar polyhedron of a 3-dimensional cube is an octahedron.*

2.2.1 Duality of Optimization vs. Separation

Theorem 7 *Linear optimization over P is equivalent to the* separation problem *over P^*, i.e. given a point y, either answer that $y \in P^*$ or return a separating hyperplane $\vec{w}$ such that $\forall z \in P^*, w^T z < w^T y$.*

Proof: Consider the following decision version of LP, equally powerful: Given P and a point c, either answer that $\max\{c^T x : x \in P\} \leq 1$ or return a point $x \in P$ such that $c^T x > 1$. The optimization version of LP reduces to the decision version of LP via binary search. We now show that the decision version is equivalent to the separation problem over P^*.

Given P and a point c, we have $\max\{c^T x : x \in P\} \leq 1 \iff \forall x \in P, c^T x \leq 1 \iff c \in P^*$. If $\max\{c^T x : x \in P\} > 1$, then $x \in P$ achieving this maximum serves as a certificate that $c \notin P^*$, because $y^T x \leq 1$ for all $y \in P^*$, whereas $c^T x > 1$. Conversely, any hyperplane separating c from P^* corresponds to a point $x \in P$ such that $c^T x > 1$. □

2.2.2 P-Time Solvability of Linear Programming

The Ellipsoid method of Khachiyan achieves the following:

1. Solves a LP problem in time polynomial in the size of the LP (the number of bits in an explicit representation);

2. Even if the LP is exponentially large, if the separation problem can be solved in polynomial time then the Ellipsoid method can solve the optimization problem in polynomial time using separation as a subroutine.

By polarity, we conclude that separation/optimization over P and P^* are all poly-time equivalent.

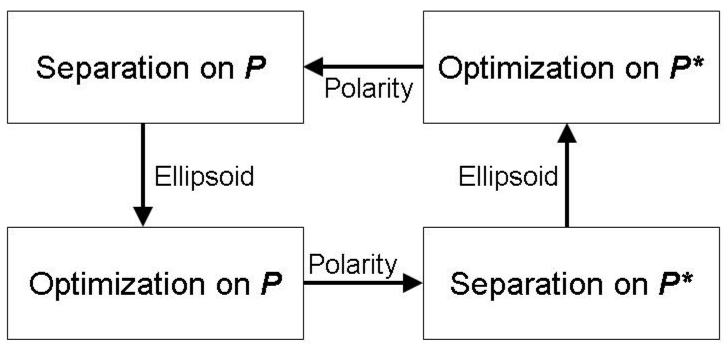

Figure 3: Summary of relations.