Integer Programming
ISE 418

Lecture 11

Dr. Ted Ralphs

Reading for This Lecture

- Nemhauser and Wolsey Sections II.1.1-II.1.3, II.1.6
- Wolsey Chapter 8
- CCZ Chapter 3, Section 7.5

Describing $\mathrm{conv}(\mathcal{S})$

- We have seen that, in theory, $\mathrm{conv}(\mathcal{S})$ has a finite description.
- If we "simply" construct that description, we could turn our MILP into an LP.
- So why aren't IPs easy to solve?
  - The size of the description is generally HUGE!
  - The number of facets of the TSP polytope for an instance with 120 nodes is more than 10^{100} times the number of atoms in the universe.
  - It is physically impossible to write down a description of this polytope.
  - Not only that, but it is very difficult in general to generate these facets (this problem is not polynomially solvable in general).

For Example

- For a TSP of size 15
  - The number of subtour elimination constraints is $16{,}368$.
  - The number of *comb inequalities* is $1,993,711,339,620$.
  - These are only two of the know classes of facets for the TSP.
- For a TSP of size 120
  - The number of subtour elimination constraints is 0.6×10^{36}!
  - The number of comb inequalities is approximately 2×10^{179}!

Valid Inequalities Revisited

- Recall that the inequality denoted by (π, π_0) is *valid* for a polyhedron $\mathcal{P}$ if $\pi x \leq \pi_0 \; \forall x \in \mathcal{P}$.
- Note that an inequality (π, π_0) is valid if and only if

$$\pi_0 \geq \max_{x \in \mathcal{P}} \pi^\top x$$

- Alternatively, an inequality (π, π_0) is valid if

$$\pi_0 \geq F(b),$$

  where F is a dual function with respect to the optimization problem

$$\max_{x \in \mathcal{P}} \pi^\top x$$

- Thus, there is an inextricable link between valid inequalities and optimization.

Cutting Planes

- The term *cutting plane* usually refers to an inequality valid for $\text{conv}(\mathcal{S})$, but which is violated by the solution to the (current) LP relaxation.
- Cutting plane methods attempt to improve the bound produced by the LP relaxation by iteratively adding cutting planes to the initial LP relaxation.
- Adding such inequalities to the LP relaxation *may* improve the bound (this is not a guarantee).

The Separation Problem

- Formally, the problem of generating a cutting plane can be stated as follows.

  Separation Problem: Given a polyhedron $\mathcal{Q} \subseteq \mathbb{R}^n$ and $x^* \in \mathbb{R}^n$, determine whether $x^* \in \mathcal{Q}$ and if not, determine (π, π_0), an inequality valid for $\mathcal{Q}$ such that $\pi x^* > \pi_0$.

- This problem is stated here independent of any solution algorithm.
- However, it is typically used as a subroutine inside an iterative method for improving the LP relaxation.
- In such a case, x^* is the solution to the LP relaxation (of the current formulation, including previously generated cuts).
- We will see that the difficulty of solving this problem exactly is strongly tied to the difficulty of the optimization problem itself.
- Any algorithm for solving the separation problem can be immediately leveraged to produce an algorithm for solving the optimization problem.
- This algorithm is know as the *cutting plane algorithm*.

Generic Cutting Plane Method

Let $\mathcal{P} = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$ be the initial formulation for

$$\max\{c^\top x \mid x \in \mathcal{S}\}, \tag{MILP}$$

where $\mathcal{S} = \mathcal{P} \cap \mathbb{Z}^r_+ \times \mathbb{R}^{n-p}_+$, as defined previously.

Cutting Plane Method

$\mathcal{P}_0 \leftarrow \mathcal{P}$
$k \leftarrow 0$
while TRUE **do**
  Solve the LP relaxation $\max\{c^\top x \mid x \in \mathcal{P}_k\}$ to obtain a solution x^k
  Solve the problem of separating x^k from $\mathrm{conv}(\mathcal{S})$
  if $x^k \in \mathrm{conv}(\mathcal{S})$ **then**
    STOP
  else
    Determine an inequality (π^k, π^k_0) valid for $\mathrm{conv}(\mathcal{S})$ but for which $\pi^\top x^k > \pi^k_0$.
  end if
  $\mathcal{P}_{k+1} \leftarrow \mathcal{P}_k \cap \{x \in \mathbb{R}^n \mid (\pi^k)^\top x \leq \pi^k_0\}$.
  $k \leftarrow k + 1$
end while

Questions to be Answered

- How do we solve the separation problem in practice?
- Will this algorithm terminate?
- If it does terminate, are we guaranteed to obtain an optimal solution?

The Separation Problem as an Optimization Problem

Separation Problem: Given a polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ and $x^* \in \mathbb{R}^n$, determine whether $x^* \in \mathcal{P}$ and if not, determine (π, π_0), a valid inequality for $\mathcal{P}$ such that $\pi x^* > \pi_0$.

- Closer examination of the separation problem for a polyhedron reveals that it is in fact an optimization problem.
- Consider a polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ and $x^* \in \mathbb{R}^n$.
- The separation problem can be formulated as

$$\max\{\pi x^* - \pi_0 \mid \pi^\top x \leq \pi_0 \ \forall x \in \mathcal{P}, (\pi, \pi_0) \in \mathbb{R}^{n+1}\} \qquad \text{(SEP)}$$

  along with some appropriate normalization.
- When $\mathcal{P}$ is a polytope, we can reformulate this problem as the LP

$$\max\{\pi x^* - \pi_0 \mid \pi^\top x \leq \pi_0 \ \forall x \in \mathcal{E}\},$$

  where $\mathcal{E}$ is the set of extreme points of $\mathcal{P}$.
- When $\mathcal{P}$ is not bounded, the reformulation must account for the extreme rays of $\mathcal{P}$.

Normalization and the 1-Polar

- Assuming w.l.og. that 0 is in the interior of $\mathcal{P}$, the set of all inequalities valid for $\mathcal{P}$ is given by

$$\mathcal{P}^* = \{\pi \in \mathbb{R}^n \mid \pi^\top x \leq 1 \ \forall x \in \mathcal{P}\}$$

  and is called its *1-Polar*.

- Then we can normalize (SEP) by taking $\pi_0 = 1$.

- If $\mathcal{P} \subseteq \mathbb{R}^n$ is a polyhedron containing the origin, then

  1. $\mathcal{P}^*$ is a polyhedron;
  2. $\mathcal{P}^{**} = \mathcal{P}$;
  3. $x \in \mathcal{P}$ if and only if $\pi^\top x \leq 1 \ \forall \pi \in \mathcal{P}^*$;
  4. If $\mathcal{E}$ and $\mathcal{R}$ are the extreme points and extreme rays of $\mathcal{P}$, respectively, then

$$\mathcal{P}^* = \{\pi \in \mathbb{R}^n \mid \pi^\top x \leq 1 \ \forall x \in \mathcal{E}, \pi^\top r \leq 0 \ \forall r \in \mathcal{R}\}.$$

- A converse of the last result also holds.
  - If the polar is described by a finite set of points and rays, then these constitute generators for the polyhedron.
  - However, these sets need not be minimal.

Interpreting the Polar

- The polar is the set of all valid inequalities, but without some normalization, it contains all scalar multiples of each inequality.
- The 1-Polar of a polyhedron is the set of all valid inequalities as long as 0 is in the interior.
- The 1-Polar has a built-in normalization.
- There is a one-to-one correspondence between the facets of the polyhedron and the extreme points of the 1-Polar when
  - the polyhedron is full-dimensional and
  - the origin is in its interior,
- Hence, the separation problem can be seen as an optimization problem over the polar.

Solving the Separation Problem

- The separation problem (SEP) for $\mathcal{P}$ has a large number of inequalities in principle (one for each extreme point).
- Can we solve it efficiently?
  - In principle, it can itself be solved by a cutting plane algorithm!
  - This is a bit circular...this requires solving the separation problem for the set

    $$\{\pi \in \mathbb{R}^{n+1} \mid \pi^\top x \leq 1 \ \forall x \in \mathcal{E}\}$$

    of members of the *1-Polar*).
  - It is easy to see, however that the separation problem for the 1-Polar can be formulated as

    $$\max\{\pi^* x \mid x \in \mathcal{P}\},$$

    which is an optimization problem over $\mathcal{P}$!

The Membership Problem

Membership Problem: Given a polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ and $x^* \in \mathbb{R}^n$, determine whether $x^* \in \mathcal{P}$.

- The membership problem is a decision problem and is closely related to the separation problem.
- In fact, if we take the dual of (SEP), we get

$$\min_{\lambda \in \mathbb{R}^{\mathcal{E}}_{+}} \left\{ 0^\top \lambda \mid E\lambda = x^*, 1^\top \lambda = 1 \right\}, \qquad \text{(MEM)}$$

  where E is a matrix whose columns are the extreme points of $\mathcal{P}$.
- In other words, we try to express x^* as a convex combination of extreme points of $\mathcal{P}$.
- When this LP is infeasible, the certificate is a separating hyperplane.
- We solve this LP by column generation (more details to come).
- In each iteration, a new column is "generated" by optimizing over $\mathcal{P}$.
- We can picture this algorithm in the "primal space" to understand what it's doing.

Example: Separation Algorithm with Optimization Oracle

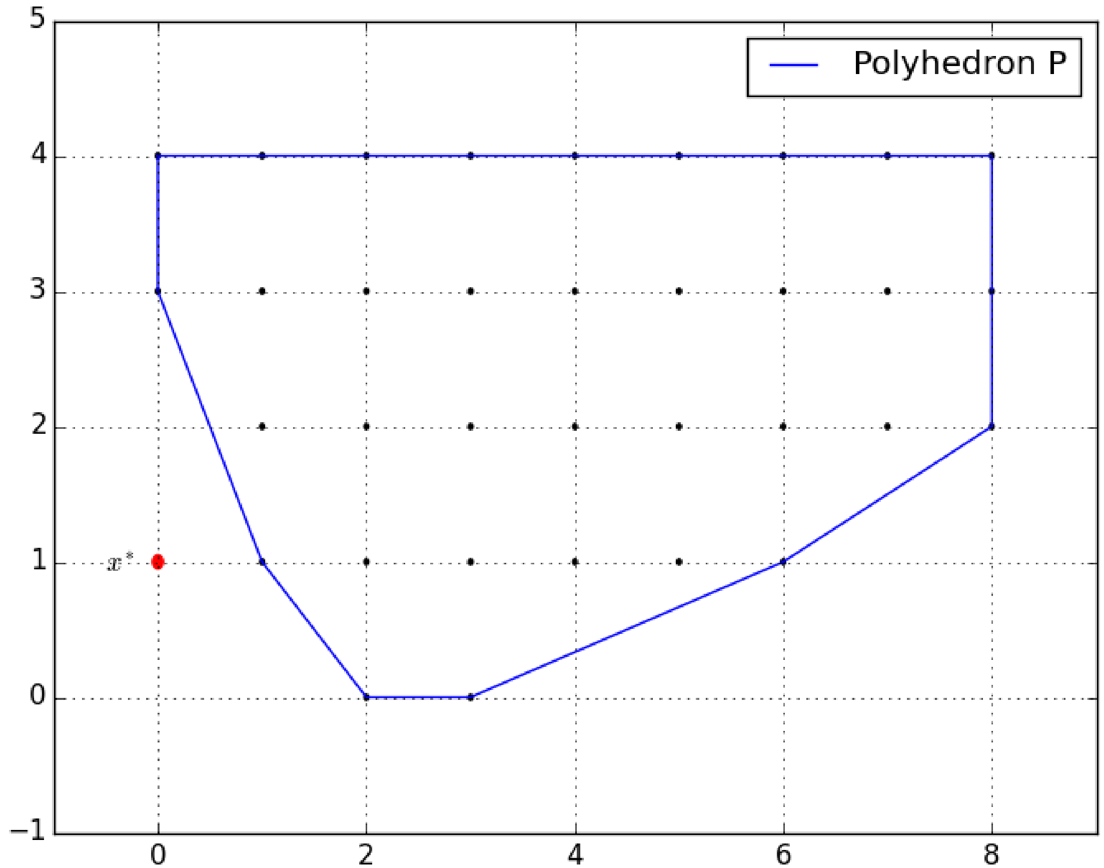

Figure 1: Polyhedron and point to be separated

Example: Separation Algorithm with Optimization Oracle

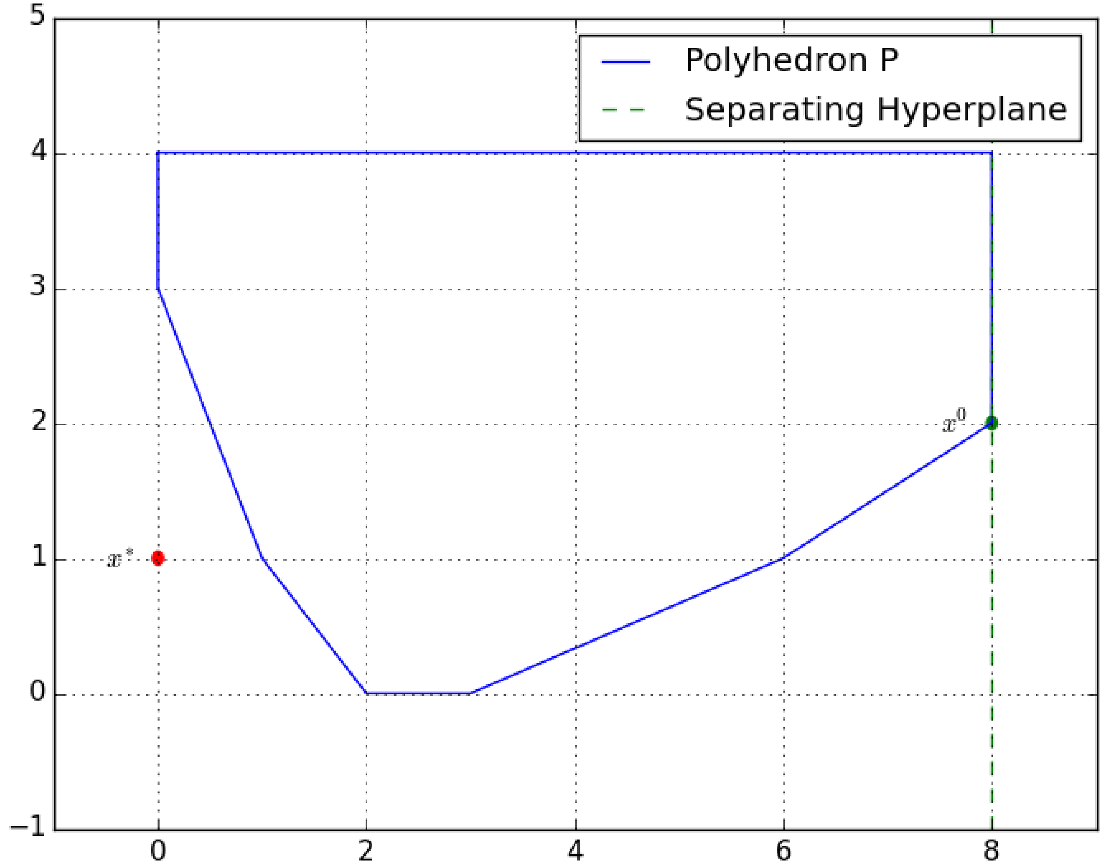

Figure 2: Iteration 1

Example: Separation Algorithm with Optimization Oracle

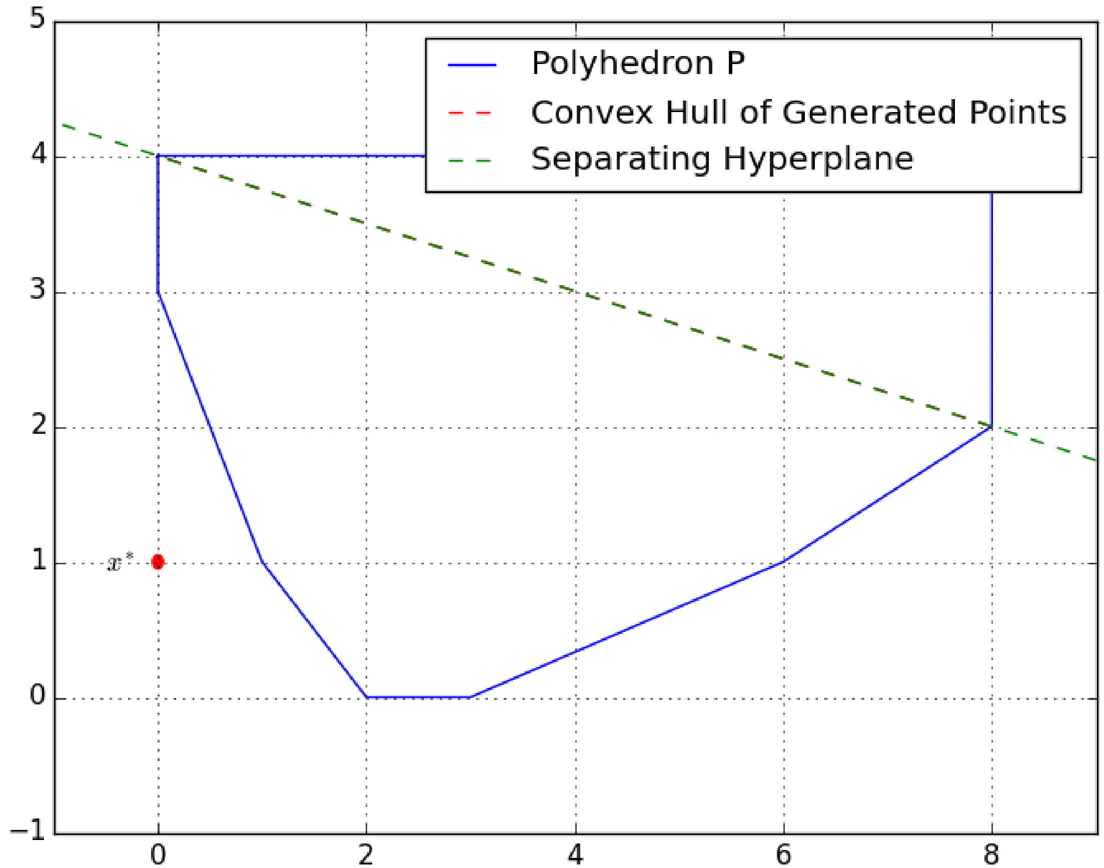

Figure 3: Iteration 2

Example: Separation Algorithm with Optimization Oracle

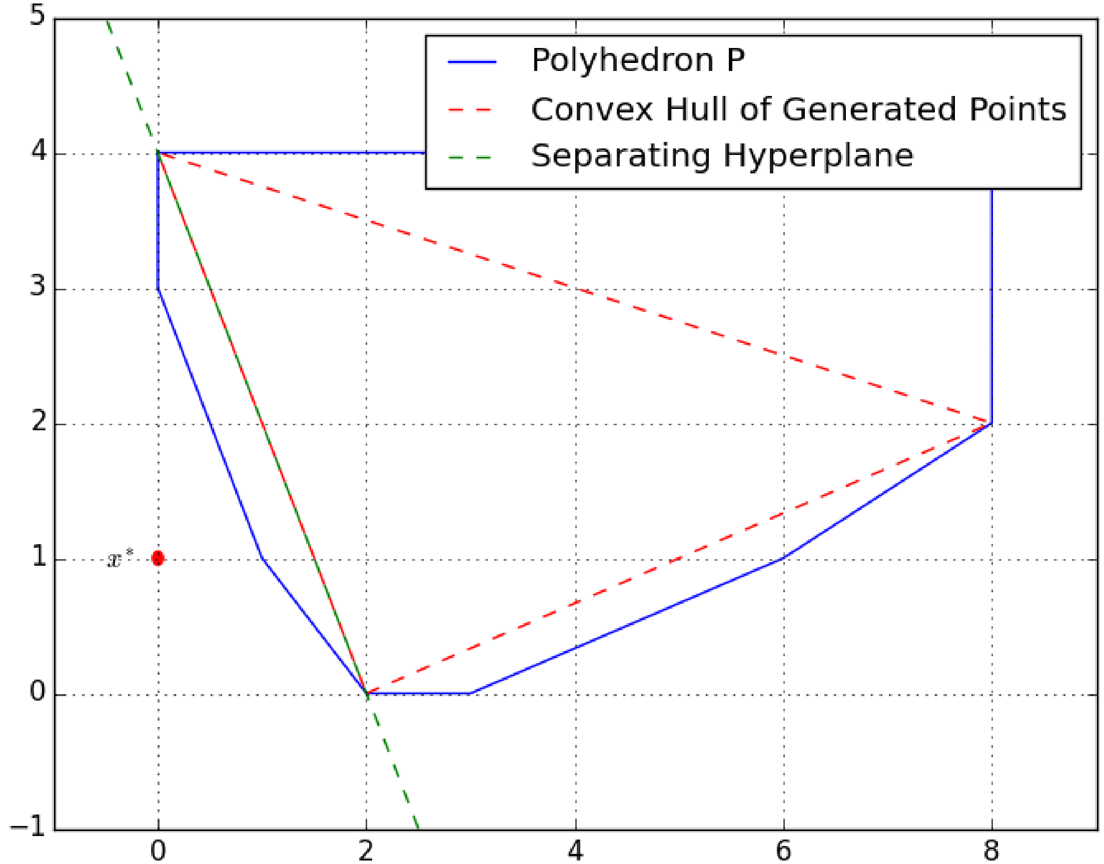

Figure 4: Iteration 3

Example: Separation Algorithm with Optimization Oracle

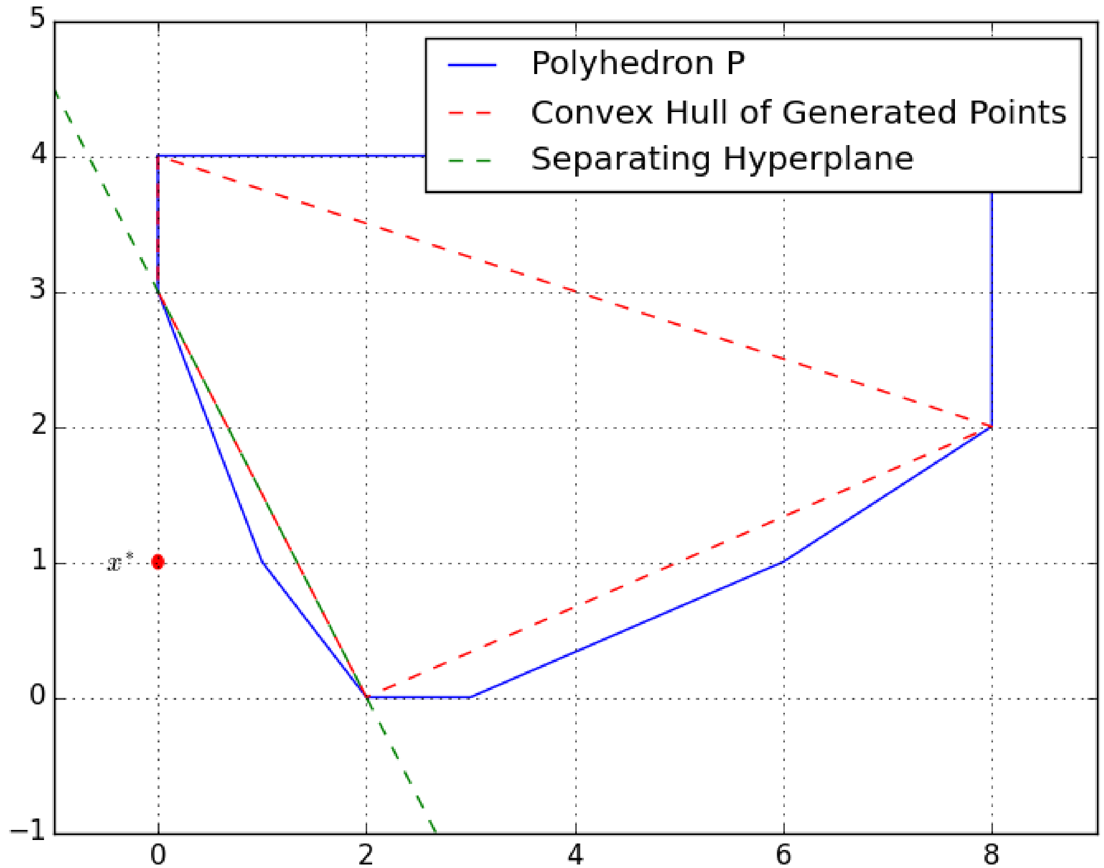

Figure 5: Iteration 4

Example: Separation Algorithm with Optimization Oracle

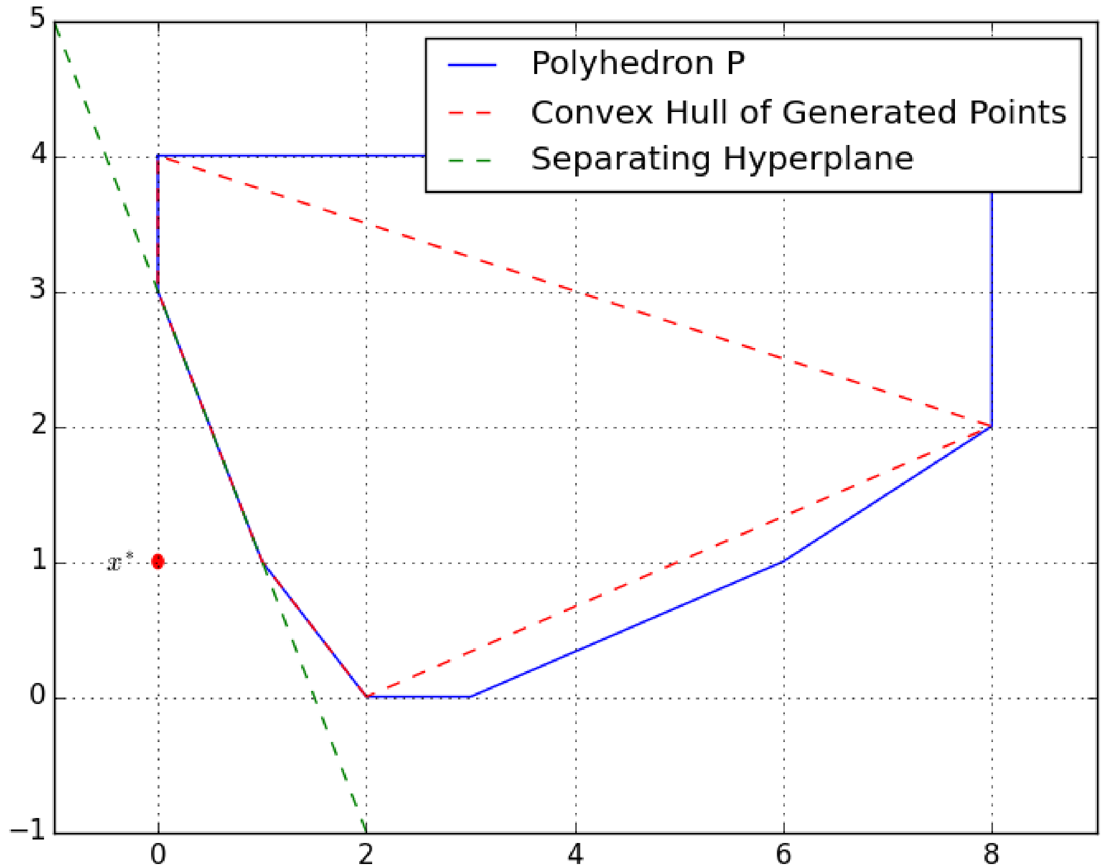

Figure 6: Iteration 5

Formal Equivalence of Separation and Optimization

Separation Problem: Given a polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ and $x^* \in \mathbb{R}^n$, determine whether $x^* \in \mathcal{P}$ and if not, determine (π, π_0), a valid inequality for $\mathcal{P}$ such that $\pi x^* > \pi_0$.

Optimization Problem: Given a polyhedron $\mathcal{P}$, and a cost vector $c \in \mathbb{R}^n$, determine x^* such that $cx^* = \max\{cx : x \in \mathcal{P}\}$.

Theorem 1. *For a family of rational polyhedra $\mathcal{P}(n, T)$ whose input length is polynomial in n and $\log T$, there is a polynomial-time reduction of the linear programming problem over the family to the separation problem over the family. Conversely, there is a polynomial-time reduction of the separation problem to the linear programming problem.*

- The parameter n represents the dimension of the space.
- The parameter T represents the largest numerator or denominator of any coordinate of an extreme point of $\mathcal{P}$ (the *vertex complexity*).
- The *ellipsoid algorithm* provides the reduction of linear programming separation to separation.
- *Polarity* provides the other direction.

Proof: The Ellipsoid Algorithm

- The ellipsoid algorithm is an algorithm for solving linear programs.
- The implementation requires a subroutine for solving the *separation problem* over the feasible region (see next slide).
- We will not go through the details of the ellipsoid algorithm.
- However, its existence is very important to our study of integer programming.
- Each step of the ellipsoid algorithm, *except that of finding a violated inequality*, is polynomial in
  - n, the dimension of the space,
  - $\log T$, where is the largest numerator or denominator of any coordinate of an extreme point of $\mathcal{P}$, and
  - $\log \|c\|$, where $c \in \mathbb{R}^n$ is the given cost vector.
- The entire algorithm is polynomial if and only if the separation problem is polynomial.

Classes of Inequalities

- As we have just shown, producing general facets of $\mathrm{conv}(\mathcal{S})$ is as hard as optimizing over $\mathcal{S}$.
- Thus, the approach often taken is to solve a "relaxation" of the separation problem.
- This "relaxation" is usually obtained in one of several ways.
  - It can be obtained in the usual way by relaxing some constraints to obtain a more tractable problem.
  - The "structure" of the inequalities may be somehow restricted to make the right-hand side easy to compute.
  - We may also use a dual function to compute the right-hand side rather than computing the "optimal" right-hand side.
- We will see examples of the second approach in later lectures.
- In either of the first two cases, the class of inequalities we want to generate typically defines a polyhedron $\mathcal{C}$.
- $\mathcal{C}$ is what we earlier called the *closure*.
- The separation problem for the class is the separation problem over the closure.