ORIE 6300 Mathematical Programming I	November 4, 2014
Lecture 20	
Lecturer: David P. Williamson	*Scribe: Tiandong Wang*

1 Overview of the Ellipsoid Method

The goal of ellipsoid method is to find a point x in a bounded polyhedron $P = \{x : Cx \leq d\}$ or correctly declare $P = \emptyset$. The main idea here is as follows:

(i) Start with a sphere centered at the origin (i.e. $a_0 = 0$) which is sufficiently large to contain P. Note that a sphere of radius 2^L is good enough, where L denotes the number of bits needed to encode C and d.

(ii) Check if $a_k \in P$. If so, return a_k.

(iii) Otherwise, we have $a_k \notin P$ because $C_j a_k > d_j$ for some j. Then we divide the ellipsoid E_k through its center a_k with a hyperplane parallel to $C_j x = d_j$.

(iv) Compute the new ellipsoid E_{k+1} centered at $a_k + 1$ containing the "good" half of E_k.

This method makes progress by showing the volume of the ellipsoid shrinks by a factor of $e^{\frac{1}{2(n+1)}}$ in each iteration. Starting with a sphere of volume $2^{O(nL)}$, we will show that the algorithm can terminate if the volume is $2^{-\Omega(nL)}$. Note that for every $O(n)$ iterations, the volume decreases by a factor of e, and hence after $O(n^2L)$ iterations we will have the desirable results.

2 Termination of the Ellipsoid Method

2.1 Modifying the LP

Recall that any vertex $\widetilde{x}$ of $Cx \leq d$ has $|\widetilde{x}_j| \leq 2^L$. So we can restate the system as below:

$$\begin{gathered} Cx \leq d \\ x \leq 2^L e \\ -x \leq 2^L e, \end{gathered}$$

where e is the vector of all ones.

Lemma 1 *1. If $Cx \leq d$ is infeasible, so is $Cx \leq d + \frac{2^{-L}}{n+2}e$.*
2. If $Cx \leq d$ is feasible, then $\exists \widehat{x}$ such that

$$B(\widehat{x}, \frac{2^{-2L}}{n+2}) \subseteq \{x : Cx \leq d + \frac{2^{-L}}{n+2}e\},$$

where $B(x, r)$ is a ball centered at x with radius r.

Proof: 1. If $Cx \le d$ is infeasible, then by Farkas' Lemma,

$$c^T y = 0, \quad d^T y = -1, \quad y \ge 0$$

must be feasible. As argued before, there exists a vertex $\widehat{y}$ such that $|\widehat{y}| \le 2^L e$. Also, we know that at most $n+1$ components of $\widehat{y}$ are non-zero. Therefore,

$$\begin{aligned}(d + \frac{2^{-L}}{n+2} e)^T \widehat{y} &= d^T \widehat{y} + \frac{2^{-L}}{n+2} \sum_i \widehat{y}_i \\ &< d^T \widehat{y} + 1 = -1 + 1 = 0.\end{aligned}$$

Hence, $\widehat{y}$ satisfies

$$c^T \widehat{y} = 0, \quad (d + \frac{2^{-L}}{n+2} e)^T \widehat{y} < 0, \quad \widehat{y} \ge 0.$$

Then by Farkas' Lemma, we know that

$$Cx \le d + \frac{2^{-L}}{n+2} e$$

is infeasible.
2. Let $\widehat{x}$ be feasible for $Cx \le d, x \le 2^L e, -x \le 2^L e$. Pick any $x \in B(\widehat{x}, \frac{2^{-2L}}{n+2})$, then

$$\begin{aligned}C_j x &= C_j \widehat{x} + C_j (x - \widehat{x}) \\ &\le d_j + \parallel C_j \parallel \parallel x - \widehat{x} \parallel \\ &\le d_j + 2^L \cdot \frac{2^{-2L}}{n+2} \\ &= d_j + \frac{2^{-L}}{n+2},\end{aligned}$$

where C_j is the jth row of the matrix C. It then follows that

$$B(\widehat{x}, \frac{2^{-2L}}{n+2}) \subseteq \{x : Cx \le d + \frac{2^{-L}}{n+2} e\}.$$

$\square$

The main idea of this lemma is that we are actually running ellipsoid method on the system

$$Cx \le d + \frac{2^{-L}}{n+2} e.$$

If this new system is feasible, so is $Cx \le d$. If, instead, $Cx \le d$ is feasible, then $Cx \le d + \frac{2^{-L}}{n+2} e$ has volume at least $(\frac{2^{-2L}}{n+2})^n$ (i.e. the volume of the ball $B(\widehat{x}, \frac{2^{-2L}}{n+2})$). If the volume of the current ellipsoid is less than this quantity, then $Cx \le d$ is infeasible.

Another thing to note is that we could possibly find some x such that $Cx \le d + \frac{2^{-L}}{n+2} e$, but $Cx > d$. However, we can deal with this issue by recalling the algorithm presented at the end of Lecture 18, where we showed how to find $x \in P$ given the oracle that we need to correctly state if P is "nonempty" or "empty".

In conclusion, we see that if we add bounding constraints and perturb the RHS, we obtain a system for which the ellipsoid method will terminate, and in the case when the obtained solution happens to be infeasible in the original system, we have an algorithm to find a feasible solution instead.

2.2 Using Ellipsoid Method to Solve LPs

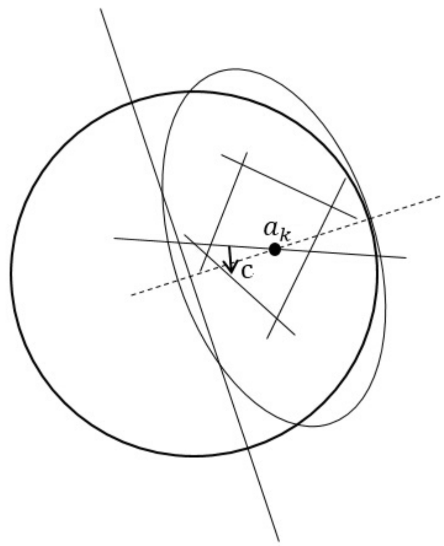

Figure 1: A graphical illustration of an objective function cut.

Previously, we gave a 3-step process for finding optimal solution to $min\{c^T x : Ax \le b, x \ge 0\}$ using the ellipsoid method. Now we can use the following instead:

- Given the current ellipsoid $E(a_k, A_k)$, if $a_k \notin \{x : Ax \le b, x \ge 0\}$ since $A_j a_k > b_k$ for some j, then we can use hyperplane $A_j x \le A_j a_k$ to divide the region into two parts.
- If $a_k \in \{x : Ax \le b, x \ge 0\}$, use hyperplane $c^T x \le c^T a_k$ to separate the region and keep the region that contains an optimal solution. This is called an *objective function cut.* A graphical illustration is given in Figure 1.

This idea can be extended to a polynomial time algorithm for optimizing linear programs.

3 Application of the Ellipsoid Method

Recall the dual of the maximum multicommodity flow problem in Problem Set 8 Q2:

$$\begin{aligned} \min \sum_{a \in A} & u_a z_a \\ \sum_{a \in P} & z_a \ge 1 \qquad \forall P \in \mathcal{P}_i, \quad \forall i \\ & z_a \ge 0, \end{aligned}$$

where $\mathcal{P}_i$ denotes the set of all possible $s_i - t_i$ paths. Using ellipsoid method, we can solve this LP without writing down all constraints (i.e. we do not need to give entire LP as our input). Given

$z = a_k$, we only need to check if z satisfies all constraints or not. If so, we use objective function cuts to find optimal solutions. If not, we just need to return or find some violated constraints. If further, we can do this in polynomial time, then we can solve the LP in polynomial time.

If $\exists i$ and $P \in \mathcal{P}_i$ such that $\sum_{a \in P} z_a < 1$, then this is a violated constraint. Indeed, for $\forall P \in \mathcal{P}_i$, $\sum_{a \in P} z_a \geq 1$ if and only if the length of the shortest $s_i - t_i$ path (using z_a as the length of arc a) is greater than or equal to 1. We know that the minimum-cost spanning tree problem can be solved in polynomial time. Therefore, we can check if z is feasible or not in polynomial time, and return a violated constraint if not. In this sense, we can actually solve this LP in polynomial time.

Generally, given any LP, if there is a polynomial time *separation oracle*, then we can solve the LP in polynomial time. *Separation oracle* here means that given x, check if x is feasible for the LP, and return a violated constraint if not. In fact, we can use ellipsoid method for solving general convex optimization, given a polynomial-time algorithm that computes the separating hyperplane, and given that the logarithm of the ratio of the initial bounding sphere and final volume is polynomial in the input size.