Two Lectures on the Ellipsoid Method for Solving Linear Programs*

1 Lecture 1: A polynomial-time algorithm for LP

Consider the general linear programming problem:

$$\begin{array}{rl} \delta^* = \max & cx \\ S.T. & Ax \le b,\ x \in \mathbb{R}^n, \end{array} \tag{1}$$

where $A = (a_{ij})_{i,j} \in \mathbb{Z}^{m \times n}$ is an $m \times n$ integer matrix, and $b = (b_i)_i \in \mathbb{Z}^m$ and $c = (c_j)_j \in \mathbb{Z}^n$ are integer vectors. We denote by a_i^T the i^{th} row of A.

In the *unit-cost* model of computation, the cost of multiplying two integers x and y is $\text{cost}(x \cdot y) = 1$. However, in the *bit-model*, it is $\text{cost}(x \cdot y) = \ell(x) + \ell(y)$, where

$$\ell(x) = \log(1 + |x|) + 1 = \text{number of bits used to represent } x \text{ in binary.}$$

For instance, consider the following code for computing $x = a^{2^k}$: 1. $x \leftarrow a$; 2. for $i = 1, \ldots, k$, set $x \leftarrow x^2$. Then in the unit-cost model, the cost of this code is k, while in the bit-model, it is $\log x \approx 2^k \ell(a)$.

Let $\ell = \max\{\ell(a_{ij}), \ell(b_i), \ell(c_j)\}$, and let

$$L = \sum_{i,j} \ell(a_{ij}) + \sum_i \ell(b_i) + \sum_j \ell(c_j)$$

be the total number of bits needed to represent the input. In the bit-model of computation, an algorithm is said to run in polynomial-time, if the number of operations ($\{+, -, *, /\}$) is at most $\text{poly}(n, m, \ell)$ and the bit length of all numbers involved in the computation is at most $\text{poly}(n, m, \ell)$. As we shall see, linear programming is polynomial-time solvable in the bit-model.

*These notes were taken by Khaled Elbassioni (Max-Planck-Institut für Informatik, Saarbrücken, Germany; (elbassio@mpi-sb.mpg.de)) while attending a graduate course on Linear Programming taught by Leonid Khachiyan in Fall 1996 at Rutgers University.

The Ellipsoid method

Let $\varepsilon > 0$ be a given constant. We call $\tilde{x}$ an ε-approximate solution of (1) if $c\tilde{x} \geq \delta^* - \varepsilon$ and $a_i^T \tilde{x} \leq b_i + \varepsilon$, for $i = 1, \ldots, m$.

Assumption 1 *We know $R \in \mathbb{R}_+$, such that (1) has an optimal solution x^* in the Euclidean ball $B_R = \{x \ / \ \|x\| \leq R\}$.*

Let $h = \max\{|a_{ij}|, |b_i|, |c_j|\}$, i.e., $h = 2^\ell - 1$. Then under Assumption 1, the Ellipsoid method computes an ε-approximate solution of (1) in $O((n+m)n^3 \log\left(\frac{Rhn}{\varepsilon}\right))$ arithmetic operations over $O(\log\left(\frac{Rhn}{\varepsilon}\right))$-bit numbers.

Fact 1 *Let V be the unit ball $V = \{x \ / \ \|x\| \leq 1\}$, and let $V^- = V \cap \{x \ / \ x_n \geq 0\}$. Then there's an ellipsoid E' such that*

- $V^- \subseteq E'$
- $\frac{\operatorname{vol} E'}{\operatorname{vol} V} \leq e^{\frac{-1}{2(n+1)}} \approx 1 - \frac{1}{2n}$.

Proof. Let the center of E' be $(0, 0, \ldots, 0, \frac{1}{n+1}) = \frac{1}{n+1}\mathbf{e}_n$, and $\alpha = 1 - \frac{1}{n+1} \approx 1 - \frac{1}{n}$ (see Figure 1). The equation of E' is

$$\frac{\left(x_n - \frac{1}{n+1}\right)^2}{\alpha^2} + \frac{x_1^2}{\beta^2} + \frac{x_2^2}{\beta^2} + \cdots + \frac{x_{n-1}^2}{\beta^2} \leq 1.$$

At point y in Figure 1: $x_n = 0$ and $x_1^2 + x_2^2 + \cdots + x_{n-1}^2 = 1$, and we get $\frac{(1/n+1)^2}{(n/n+1)^2} + \frac{1}{\beta^2} = 1$, which implies

$$\beta = \sqrt{1 + \frac{1}{n^2 - 1}} \approx 1 + \frac{1}{2n^2}.$$

In the plane, an ellipse E with principal axes' lengths α and β, has an area of $\pi\alpha\beta$, and hence the ratio of area of E to that of the unit circle V is $\alpha\beta$. In 3-dimensions the volume of an ellipsoid with principal axes'lengths α, β, and γ is $\frac{4}{3}\pi\alpha\beta\gamma$, while the volume of the unit ball is $\frac{4}{3}\pi$, giving a ratio of $\alpha\beta\gamma$. (This follows specifically from the fact that the ellipse E is obtained by transforming the circle using the linear map $x' = Ax$, where $A = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}$, and thus $\operatorname{vol}(E) = |\det(A)| \operatorname{vol}(V)$.) Generalizing, we get that $\operatorname{vol}(E')/\operatorname{vol}(V) = \alpha\beta^{n-1}$. Using the inequality $1 + x \leq e^x$, valid for all $x \in \mathbb{R}$ (see Figure 2), we get $\alpha \leq e^{-\frac{1}{n+1}}$ and $\beta \leq e^{\frac{1}{2(n^2-1)}}$, and thus

$$\frac{\operatorname{vol} E'}{\operatorname{vol} V} = \alpha\beta^{n-1} \leq e^{-\frac{1}{n+1}} e^{\frac{n-1}{2(n^2-1)}} = e^{-\frac{1}{2(n+1)}}.$$

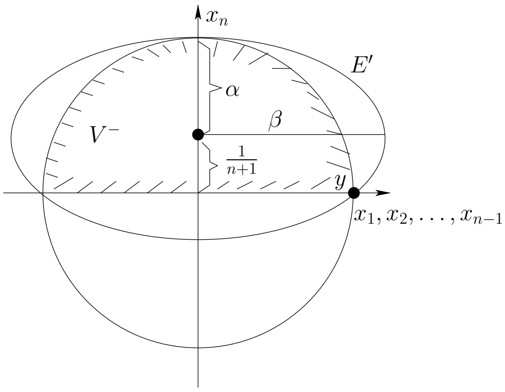

Fig. 1

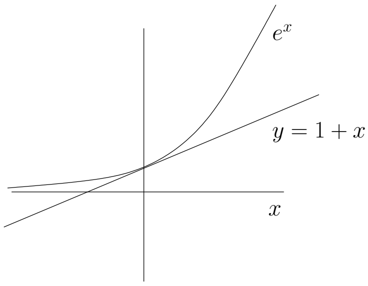

Fig. 2

Fact 2 *Let E be an ellipsoid in $\mathbb{R}^n$ centered at η. Given a non-zero vector $a \in \mathbb{R}^n$, consider the hyperplane $\pi = \{x \ / \ a^T(x-\eta) = 0\}$, and let E^+ and E^- be the two halves of E obtained by cutting E with π. Then there's an ellipsoid E' such that*

- $E^- \subseteq E'$
- $\frac{\operatorname{vol} E'}{\operatorname{vol} E} \leq e^{\frac{-1}{2(n+1)}} \approx 1 - \frac{1}{2n}$.

Proof. This follows by from Fact 1 by applying a linear transformation (rotation) that maps E to the unit ball V. Such a transformation does not change the ratio of volumes or inclusion. Note that an ellipsoid E can be represented as $E = \{x \ / \ x = \eta + Qz, \ \|z\| \leq 1\}$, where $\eta \in \mathbb{R}^n$ is the center, and Q is an $n \times n$ matrix. Given η, Q and a vector $a \in \mathbb{R}^n$, we can get the center η' and matrix Q', corresponding to the ellipsoid $E' \supseteq E \cap \{x \ / \ a^T(x-\eta) \leq 0\}$ in Fact 1 as follows. With respect to the z-coordinates, we can write the inequality determined by π as $a^T(x-\eta) = a^T Qz = (Q^T a)^T z \leq 0$. Thus the vector $\mathbf{e}_n$ if Fact 1 corresponds (by an orthonormal transformation) to the vector $-Q^T a / \|Q^T a\|$ in our case. Hence, the center of E' in the z-coordinates is $z' = -\frac{1}{n+1} \frac{Q^T a}{\|Q^T a\|}$. Let $\mu = \frac{Q^T a}{\|Q^T a\|}$ (which can be computed with n^2 operations), then returning to the x-coordinates we get

$$\eta' = \eta - \frac{1}{n+1} Q\mu \tag{2}$$

$$Q' = \sqrt{\frac{n^2}{n^2-1}} \left[Q + \left(\sqrt{\frac{n-1}{n+1}} - 1 \right) Q\mu\mu^T \right]. \tag{3}$$

Note that the computation in (2), (3) can be done with n^2 operations.

Recall that $\delta^* = \max\{cx \ / \ a_i^T x \leq b_i, \text{ for } i = 1, \ldots, m\}$. Let $X_\varepsilon^* = \{x \ / \ cx \geq \delta^* - \varepsilon, \ a_i^T x \leq b_i + \varepsilon, \text{ for } i = 1, \ldots, m, \ \|x\| \leq R\}$. Assumption 1 implies that

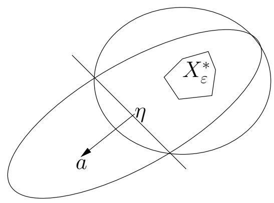

Fig. 3

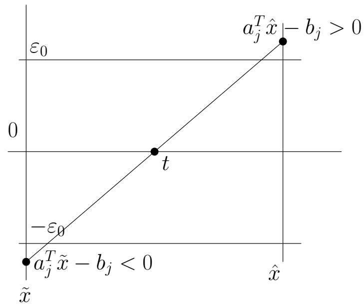

Fig. 4

$X_\varepsilon^* \neq \emptyset$ because the exact solution $x \in B_R$ belongs to X_ε^* (we assume such a solution exists). It also implies that

Fact 3 $\frac{\mathrm{vol}\, X_\varepsilon^*}{\mathrm{vol}\, B_R} \geq \left(\frac{\varepsilon}{h\sqrt{n}R}\right)^n$.

Proof. Let x^* be an exact optimal solution, i.e., $\delta^* = cx^*$. Let $y \in \mathbb{R}^n$ be such that $\|y - x^*\| \leq r$ (we will select r later). Then writing $y = x^* + \eta$, we have $\|\eta\| \leq r$ and hence $cy = cx^* + c\eta \geq \delta^* - \|c\|\|\eta\|$ (by the Cauchy-Schwartz's Inequality). Then , if we set $r = \frac{\varepsilon}{h\sqrt{n}}$, we get $\|c\|\|\eta\| \leq \|c\|r \leq h\sqrt{n}r \leq \varepsilon$. Similarly, $a_i^T x^* \leq b_i$ implies $a_i^T(x_i^* + \eta) \leq b_i + \epsilon$. Thus the ball of radius r, centered at x^*, is contained in the set X_ε^* and hence $\frac{\mathrm{vol}\, X_\varepsilon^*}{\mathrm{vol}\, B_R} \geq \frac{\mathrm{vol}\, B_r}{\mathrm{vol}\, B_R} \geq \left(\frac{r}{R}\right)^n$.

2 Lecture 2

Recall Facts 2 and 3. Now we prove the following:

Fact 4 *Suppose we are given an ellipsoid E centered at η such that $X_\varepsilon^* \subseteq E$ but $\eta \notin X_\varepsilon^*$. Then in $O((n+m)n)$ operations, we can compute a new ellipsoid E' such that (1) $X_\varepsilon^* \subseteq E'$ and (2) $\mathrm{vol}\, E' \leq e^{-\frac{1}{2(n+1)}}\, \mathrm{vol}\, E$.*

Proof. This can be done as follows (see Figure 3). Check if $a_i^T\eta \leq b_i + \varepsilon$, for $i \in M = \{1, \ldots, m\}$. If for $i_* \in M$, $a_{i_*}^T\eta > b_i + \varepsilon$, then $a_{i_*}^T x > a_{i_*}^T\eta$ implies $x \notin X_\varepsilon^*$. Thus, for any $x \in X_\varepsilon^*$, we must have $a_{i_*}^T x \leq a_{i_*}^T\eta$, i.e., $a_{i_*}^T(x - \eta) \leq 0$. Thus in this case, we set $a = a_{i_*}$ as the normal of the cutting hyperplane in Fact 2.

Now suppose that η is an ε-feasible solution. Then we check whether $\|\eta\| \leq R$. If $\|\eta\| > R$, then $\eta^T(x - \eta) \leq 0$ for all $x \in X_\varepsilon^*$. Thus we set $a = \eta$ in this case. Finally, suppose that $a_i^T\eta \leq b_i + \varepsilon$ for all $i \in M$ and $\|\eta\| \leq R$. If $\eta \notin X_\varepsilon^*$, then $c\eta < \delta^* - \varepsilon$. Hence, any x such that $cx < c\eta < \delta^* - \varepsilon$ is not in X_ε^*. Thus, $cx \geq c\eta$, or $-c(x - \eta) \leq 0$, for any ε-approximate solution. So we set a to $-c$ in this case, and apply Fact 2.

The Ellipsoid method starts with $E_0 = B_R$ and generates a sequence of ellipsoids $E_0 = B_R, E_1, E_2, \ldots, E_K, E_{K+1}$. As long as η_k (the center of E_k) is not in X_ε^*, we obtain a new ellipsoid E_{k+1} such that $X_\varepsilon^* \subseteq E_{k+1}$, and $\frac{\operatorname{vol} E_{k+1}}{\operatorname{vol} E_k} \leq e^{-\frac{1}{2(n+1)}}$. In particular,

$$\operatorname{vol} X_\varepsilon^* \leq \operatorname{vol} E_K \leq e^{-\frac{K}{2(n+1)}} \operatorname{vol} B_R.$$

From Fact 3, we get

$$\operatorname{vol} B_R \left(\frac{\varepsilon}{h\sqrt{n}R}\right)^n \leq \operatorname{vol} X_\varepsilon^* \leq \operatorname{vol} E_K \leq e^{-\frac{K}{2(n+1)}} \operatorname{vol} B_R,$$

from which we get an upper bound on K:

$$K \leq N = 2n(n+1) \ln \frac{R\sqrt{n}h}{\varepsilon} = O(n^2 \log \frac{Rnh}{\varepsilon}).$$

Termination Criterion: We run the algorithm for K iterations and always maintain the ε-feasible center with highest objective value: suppose, for some k, $a_i^T \eta_k \leq b_i + \varepsilon$, for all $i \in M$ and $\|\eta_k\| \leq R$, then we store η_k. Suppose also that, for some $l > k$, $a_i^T \eta_l \leq b_i + \varepsilon$, for all $i \in M$ and $\|\eta_l\| \leq R$. If $c\eta_l > c\eta_k$, then we replace η_k by η_l.

Thus the total number of operations is $O((n+m)n^3 \log \frac{Rnh}{\varepsilon}) = O(mn^3 \log \frac{Rnh}{\varepsilon})$ (if $m < n$, we can find in $O(m)$ time a basis and switch to a smaller problem). It can be shown also that the required accuracy is $O(\log \frac{Rnh}{\varepsilon})$.

How to round and ε-approximate solution to an exact one

Recall that we assume the matrices A, b, and c to be integral. Given an integral matrix A, denote by

$$\Delta(A) = \max\{|\det B| \ / \ B \text{ is a square submatrix of } A\}.$$

For e.g. if $A = \begin{bmatrix} 7 & 3 & 1 \\ 3 & 2 & 1 \end{bmatrix}$, then $\Delta(A) = 7$. Recall that for a vector $v \in \mathbb{R}^n$, $\|v\|_\infty = \max_i\{|v_i|\}$.

Lemma 1 *If problem (1), with integer matrices A and b, has an optimal solution, then it has an optimal solution x^* such that $\|x^*\|_\infty \leq n\|b\|_\infty \Delta(A)$.*

Proof. By the fundamental Theorem of Linear Programming, there exists $M' \subseteq M = \{1, \ldots, m\}$ such that $a_i^T x = b_i$, for all $i \in M'$. Then by Cramer's rule $x_j = \frac{\Delta_j}{\Delta}$, where Δ is a non-zero subdeterminant of A, and Δ_j is a subdeterminant of $[A \mid b]$. Since Δ is an integer, $|x_j| \leq |\Delta_j|$. But $\Delta_j = \sum_i b_i \Delta_i'$, where Δ_i' is a subdeterminant of A. Using $b_i \leq \|b\|_\infty$ and $|\Delta_i'| \leq \Delta(A)$, we get that $|x_j| \leq n\|b\|_\infty \Delta(A)$.

Thus we can set $R = hn^{3/2}\Delta(A)$ to guarantee that $\|x^*\| \leq R$ (using the fact that $\|\cdot\| \leq n^{\frac{1}{2}} \|\cdot\|_\infty$).

Lemma 2 *Suppose that δ^* is finite, then $\delta^* = \frac{t}{s}$, where t and s are integers such that $|s| \leq \Delta(A)$.*

Indeed,

$$\delta^* = c_1 x_1^* + c_2 x_2^* + \cdots + c_n x_n^* = \frac{c_1\Delta_1 + \cdots + c_n\Delta_n}{\Delta}.$$

Lemma 3 *Consider the system*

$$Ax \leq b, \ x \in \mathbb{R}^n, \tag{4}$$

and let $\varepsilon_0 = \frac{1}{(n+2)\Delta(A)}$. If there is an ε_0-approximate solution to (4) (i.e. $a_i^T\tilde{x} \leq b_i + \varepsilon_0$, for $i \in M$, is feasible), then (4) has an exact solution.

Proof. Consider

$$\begin{array}{ll} \varepsilon^* = \min & \varepsilon \\ S.T. & a_i^T x \leq b_i + \varepsilon, \ i \in M. \end{array} \tag{5}$$

Suppose that $Ax \leq b$ is infeasible. Then $\epsilon^* > 0$, and hence by Lemma 2, $\epsilon^* = \frac{t}{s}$ for some integers t and s, where $|s| \leq \Delta([A \mid -\mathbf{e}]) \leq (n+1)\Delta(A)$. This gives $\varepsilon^* \geq \frac{1}{|s|} \geq \frac{1}{(n+1)\Delta(A)}$. Thus using $\varepsilon = \varepsilon_0 = \frac{1}{(n+2)\Delta(A)}$ will make (4) feasible.

Now suppose that $a_i^T\tilde{x} \leq b_i + \varepsilon_0$, $i \in M$. We can compute an exact solution using mn^2 operations as follows. Let $I_0 = \{i \in M \ / \ |a_i^T\tilde{x} - b_i| \leq \varepsilon_0\}$. Then $a_j^T\tilde{x} < b_j - \varepsilon_0$ for $j \in M \setminus I_0$. The system $a_i^T x = b_i$, $i \in I_0$, is solvable, by Lemma 3, because $\tilde{x}$ is an ε_0-approximate solution for it (the system can be written as $a_i^T x \leq b_i$, $-a_i^T x \leq -b_i$, $i \in I_0$, and $\Delta\left(\begin{bmatrix} A_{I_0} \\ -A_{I_0} \end{bmatrix}\right) \leq \Delta(A_{I_0}) \leq \Delta(A)$, where A_{I_0} is the submatrix of A with rows indexed by I_0). Let $\hat{x}$ be an exact solution for this system, i.e., $a_i^T\hat{x} = b_i$, for $i \in I_0$. If $\tilde{x}$ does not satisfy the original system, then we proceed as follows. Define $x(t) = (1-t)\tilde{x} + t\hat{x}$, for $0 \leq t \leq 1$. Then for $i \in I_0$, we have

$$\begin{array}{rcl} |a_i^T x(t) - b_i| & = & |(1-t)a_i^T\tilde{x} + t a_i^T\hat{x} - (1-t)b_i - tb_i| \\ & = & |(1-t)[a_i^T\tilde{x} - b_i] + t[a_i^T\hat{x} - b_i]| \\ & \leq & (1-t)|a_i^T\tilde{x} - b_i| + t|a_i^T\hat{x} - b_i| \leq (1-t)\varepsilon_0 \leq \varepsilon_0. \end{array}$$

Thus $x(t)$ satisfies the constraints in I_0 for all $t \in [0,1]$. Now we obtain a new ε_0-approximate solution with a wider set of tight constraints, by choosing t as follows (see Figure 4):

$$t = \min_{j \in M \setminus I_0} \left\{ -\frac{a_j^T\tilde{x} - b_j}{a_j^T\hat{x} - a_j^T\tilde{x}} \ / \ a_j^T\hat{x} - b_j > 0 \right\}.$$

Let j_{min} be such a minimizer in $M \setminus I_0$. Then $a_{j_{min}}^T x(t) - b_{j_{min}} = (1-t)(a_{j_{min}}^T\tilde{x} - b_{j_{min}}) + t(a_{j_{min}}^T\hat{x} - b_{j_{min}}) = 0$. We add j_{min} to I_0, replace $\tilde{x}$ by $x(t)$, and repeat the procedure again. So this way, we do at most n iterations of mn^2 operations each (solving a system of linear equations), but it can be done in $O(mn^2)$ in total.

To consider the objective function: We use, instead of ε_0,

$$\varepsilon_1 = \frac{1}{4n^{5/2}\Delta^3(A)\|c\|}.$$

We do the same procedure as before without caring about the objective function. We claim that the result is optimal. Indeed, we know that $a_i^T\hat{x} = b_i$ and $a_i^T\tilde{x} = b_i + \eta_i$ for $i \in I_0$, where $\|\eta\|_\infty \leq \varepsilon_1$. Thus $a_i^T(\hat{x} - \tilde{x}) = -\eta_i$, for $i \in I_0$, and setting $\hat{x} - \tilde{x} = y$, we get (again from Cramer's rule) that $\|y\|_\infty \leq n\|\eta\|_\infty\Delta(A)$, and thus $\|\hat{x} - \tilde{x}\|_\infty \leq n\varepsilon_1\Delta(A)$. This gives $|c\hat{x} - c\tilde{x}| \leq \|c\|\|\hat{x} - \tilde{x}\| \leq \|c\|n^{3/2}\varepsilon_1\Delta(A)$.

At the end of the rounding algorithm we obtain x^* such that $Ax^* \leq b$ and $|cx^* - c\tilde{x}| \leq \|c\|n^{5/2}\varepsilon_1\Delta(A)$ (each iteration of the procedure contributes at most $\|c\|n^{3/2}\varepsilon_1\Delta(A)$ to this difference). Thus

$$\begin{aligned} cx^* &\geq c\tilde{x} - \|c\|n^{5/2}\varepsilon_1\Delta(A) \geq \delta^* - \|c\|n^{5/2}\varepsilon_1\Delta(A) - \varepsilon_1 \\ &\geq \delta^* - 2\|c\|n^{5/2}\varepsilon_1\Delta(A) = \delta^* - \frac{1}{2\Delta^2(A)}. \end{aligned}$$

But $cx^* = \frac{t'}{s'}$, where $t', s' \in \mathbb{Z}$ and $|s'| \leq \Delta(A)$ (as in Lemma 2), and $\delta^* = \frac{t}{s}$, $|s| \leq \Delta(A)$. Since $cx^* \leq \delta^*$, we have $|cx^* - \delta^*| \leq \frac{1}{2\Delta^2(A)}$. On the other hand, if $cx^* \neq \delta^*$, then

$$|cx^* - \delta^*| = \left|\frac{t'}{s'} - \frac{t}{s}\right| = \left|\frac{t's - ts'}{ss'}\right| \geq \frac{1}{ss'} \geq \frac{1}{\Delta^2(A)} > \frac{1}{2\Delta^2(A)}.$$

Thus we must have $cx^* = \delta^*$.

Summary

Using

$$\varepsilon_1 = \frac{1}{4n^{5/2}\Delta^3(A)\|c\|} \text{ and } R = n^{3/2}h\Delta(A),$$

the Ellipsoid method requires $O(mn^3 \log \frac{Rhn}{\varepsilon_1}) = O(mn^3 \log(hn\Delta(A)))$ operations.

Denote by $L = L(A, b, c)$ the number of bits required to write down the problem, and let $L(A) = \sum_{i,j} \log(1 + |a_{ij}|)$. We note that $\Delta(A) \leq 2^{L(A)}$ since $\Delta(A) \leq \prod_{i,j}(1 + |a_{ij}|)$. Also $n \leq 2^{L(A)}$ and $h \leq 2^{L(A,b,c)}$. Thus the total number of operations required by the Ellipsoid method is $O(mn^3L)$.