# Optimization Methods Term Paper: Ellipsoid Methods

Akshat Goyal, 2018101075

## I. Introduction

Many methods have been developed to solve Linear Programming Problems over time. For a long time, there was no polynomial time algorithm to solve linear programming problems. Methods like Simplex was widely used which has an exponential time complexity but gives an average-case polynomial time complexity.

In 1979, the Russian mathematician Leonid G. Khachiyan published his famous paper with the title "A Polynomial Algorithm in Linear Programming" [1]. He was able to show that linear programs can be solved efficiently, more precisely that linear programs belong to the class of polynomial solvable problems. He showed how ellipsoid method originally devised for nonlinear nondifferentiable optimization, can be modified in order to check feasibility of a system of linear inequalities in polynomial time.

In this paper, we will explain Ellipsoid Method, how ellipsoid method can be used to check the feasibility of a convex set and then reduction from feasibility to optimization of LP.
.

## II. Ellipsoid Method

The Ellipsoid Method is a method that solves linear programs in a polynomial time. The Ellipsoid Method is designed to solve decision problems rather than optimization problems. Therefore, we will first consider the Feasibility problem which either finds a feasible point in a convex set or outputs that the set is empty. We will then reduce this feasibility problem to optimization problem in polynomial time. But first we will define some key words that will be used in this paper.

### A. Definitions

*1) Ellipsoid:* An ellipsoid is defined as:

$$E(a, A) = \{x \mid (x-a)^T A^{-1}(x-a) \leq 1\}$$

where $A$ is symmetric that is $A = A^T$ and positive definite ($x^T Ax > 0 \ \forall \ x \neq 0$).

*2) Ball:* A Ball is defined as:

$$B(a, R) = E(a, R^2 I)$$

where $a$ is the centre and $R$ is the radius.

### B. Setup

We are given convex set $P$ and we need to find a feasible point in the convex set if exists. For this we need to know the upper and lower bound on the size of convex set. We assume that we are given $R$ such that P lies in $Ball(0, R)$. We are also given that if $P$ is non empty then $Ball(a, r)$ is contained in $P$ for some $a$. We can find $R$ and $r$ for given $P$ in polynomial time. Second we assume that we have a separation (cutting-plane) oracle.

Given a convex set $P$ with

1) $P \subseteq B(0, R) = \{x \mid \|x\| \leq R\}$ and promised that if $P \neq \emptyset$ then $P \geq B(a, r)$ for some $a$.
2) Separation Oracle that answer queries $x$ by
   a) assert $x \in P$ OR
   b) assert $x \notin P$ and giving $c$ such that $c^T y \leq c^T x \ \forall \ y \in P$.

### C. Feasibility Problem

We are given a convex set $P$ and we need to output any $x \in P$ or show that $P = \emptyset$. This is the definition of Feasibility problem. We will first solve Feasibility problem using Ellipsoid Method and then reduce it to optimization problem in the later part.

### D. Idea

We will maintain an ellipsoid $E_k \geq P$ and reduce its volume in each step till we get a point in $P$ or will declare that $P$ is an empty set.

### E. Algorithm

Initialise $E_0 = B(0, R)$.
**for** $k$ = 0 to $M$ **do**
  Let $E_k = (a_k, A_k)$ be current ellipsoid.
  Query separation oracle
  **if** $a_k \in P$ **then**
    Return $a_k$
  **else**
    Given $c_k$ Find

$$E_{k+1} \supseteq E_k \cap \{x \mid c^T x \leq c^T a_k\}$$

  **end if**
**end for**
Return $P = \emptyset$.

Now the main Q is that

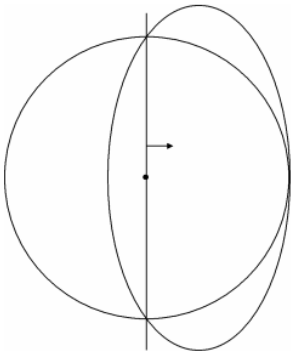

Fig. 1. Unit sphere split by hyperplane $x_1 > 0$

1) What is the value of $M$.
2) How to find $E_{k+1}$ satisfying above constraints.

We will answer these questions in the following sections.

## III. Finding Ellipsoid

Our main question is that is there a good choice for $E_{k+1}$ and how do we find it?

### A. Special Case 1

Let's consider a special case 1:

$E = E(0,1)$ and $c_k$ = -$e_1$ (where $e_i$ is a basis vector with e[i] = 1 only and rest 0).

Let's set

$$E^{'} = \{x \mid \left(\frac{n+1}{n}\right)^2 \left(x_1 - \frac{1}{n+1}\right)^2 + \left(\frac{n^2-1}{n^2}\right) \sum_{i=2}^{n} x_i^2 \leq 1\}$$

where $n$ is the dimension of $P$.

1) Claim 1: If $E = E(0,1)$ is the current ellipsoid and $c_k$ = -$e_1$, then $E^{'}$ is the next ellipsoid such that

$$E^{'} \supseteq E \cap \{x \mid c_k^T x \leq c_k^T y\}$$

. We will prove this in the section Claim 1.
2) Claim 2: $E^{'}$ is smaller in volume than $E$ and hence volume of ellipsoid is decreasing after every iteration.

### B. Claim 1

As $c_k$ = -$e_1$, therefore $x_1 \geq 0 \ \forall \ x \in P$. To prove that $E^{'}$ is the valid ellipsoid, we need to prove that If $x \in E$ and $x_1 \geq 0$, then $x \in E^{'}$.

**Proof:**

Let's simplify first term of the Ellipsoid $E^{'}$

$$\left(\frac{n+1}{n}\right)^2 \left(x_1 - \frac{1}{n+1}\right)^2$$

$$= \left(\frac{n^2-1}{n^2} + \frac{2n+2}{n^2}\right) x_1^2 - 2\left(\frac{n+1}{n^2}\right) x_1 + \frac{1}{n^2}$$

If we add $\left(\frac{n^2-1}{n^2}\right) \sum_{i=2}^{n} x_i^2$ we get

$$\left(\frac{n^2-1}{n^2}\right) \sum_{i=1}^{n} x_i^2 + \frac{(2n+2)(x_1^2 - x_1)}{n^2} + \frac{1}{n^2}$$

1) $\sum_{i=2}^{n} x_i^2 \leq 1$ as $x \in E$.
2) Second term is less than or equal to 0 as $x_1 \geq 0$ and $x_1 \leq 1$. So $x_1^2 - x_1 \leq 0$.
3) Sum of first and third term is less than or equal to 1.

Hence

$$\left(\frac{n^2-1}{n^2}\right) \sum_{i=1}^{n} x_i^2 + \frac{(2n+2)(x_1^2 - x_1)}{n^2} + \frac{1}{n^2} \leq 1$$

for $x \in E$ and $x_1 \geq 0$.

### C. Claim 2

$$\frac{Vol(E^{'})}{Vol(E)} \leq e^{-\frac{1}{2(n+1)}} \leq 1$$

**Proof:**

The volume of an ellipsoid is proportional to products of its side lengths. Using this we get

$$\frac{Vol(E^{'})}{Vol(E)} = \frac{\left(\frac{n}{n+1}\right)\left(\left(\frac{n^2}{n^2-1}\right)^{1/2}\right)^{n-1}}{1}$$

$$= \frac{\left(1 - \frac{1}{n+1}\right)\left(\left(1 + \frac{1}{n^2-1}\right)^{1/2}\right)^{n-1}}{1}$$

Using $e^{-x} \geq 1$ - x $\forall$ x $\geq 0$,

$$\leq e^{-\frac{1}{n+1}} e^{\frac{n-1}{2(n^2-1)}}$$

$$= e^{-\frac{1}{2(n+1)}}$$

After any $O(n)$ iterations, the volume of the ellipsoid will be reduced by a factor of $\approx 2$ 2.

### D. General Case for Unit Sphere

We know the solution for $E = E(0,1)$ and $c_k$ = -$e_1$. Now we will generalize it for any $c_k$ 3. We will reduce this problem to our special case.

First, suppose that $(E_0 = E(0,I))$, the unit sphere centered at origin, but now we have arbitrary constraint $c$. Assume $\|c\| = 1$. (i.e., $c^T c = 1$ ). In order to handle this, the main idea is to reduce to previous case. Consider applying a rotation $y = T(x)$, so that $-e_1 = T(c)$. Then rotate $E_1$ back using $T^{-1}$ 4.

Since $T$ is a rotation, $y = T(x) = Ux$ for some orthonormal matrix $U$ (i.e. $U^T = U^{-1}$ ). We want $Uc = -e_1$, so $c =$

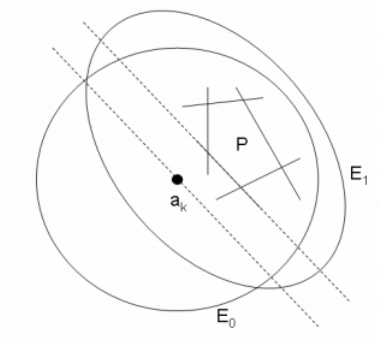


Fig. 2. $E_1$ dividing $E_0$, contains $P$

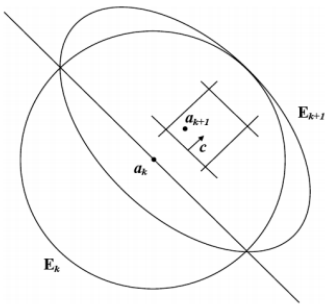


Fig. 3. General Case for Unit Sphere

$-U^{-1}e_1 = -U^T e_1$. In the transformed space, the desired ellipsoid is $\{x \in R^n \mid (Ux-a)^T A^{-1}(Ux-a) \leq 1\}$. Since $U^T U = I$, this is the same as $\{x \mid (Ux-a)^T UU^T A^{-1} UU^T (Ux-a) \leq 1\}$.

Now we observe that

$$\begin{aligned}(Ux-a)^T U &= \left((Ux)^T - a^T\right) U \\ &= \left(x^T U^T - a^T\right) U \\ &= x^T - a^T U \\ &= \left(x - U^T a\right)^T,\end{aligned}$$

and

$$U^T(Ux-a) = x - U^T a$$

where we define

$$U^T a = U^T\left(\frac{1}{n+1}e_1\right) = -\frac{1}{n+1}e =: \hat{a}.$$

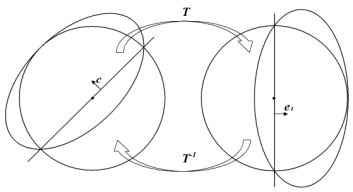


Fig. 4. Rotation

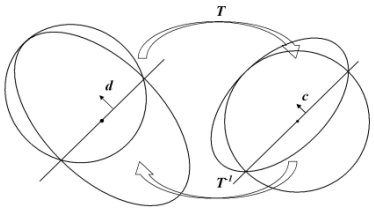


Fig. 5. Case of General Ellipsoid

If we set $\hat{A}^{-1} = U^T A^{-1} U$, then we get

$$\begin{aligned}\hat{A} &= \left(U^T A^{-1} U\right)^{-1} \\ &= U^{-1} A \left(U^{-1}\right)^T \\ &= \frac{n^2}{n^2-1} U^T \left(I - \frac{2}{n+1} e_1 e_1^T\right) U \\ &= \frac{n^2}{n^2-1}\left(I - \frac{2}{n+1}\left(U^T e_1\right)\left(e_1^T U\right)\right) \\ &= \frac{n^2}{n^2-1}\left(I - \frac{2}{n+1}(-c)\left(-c^T\right)\right) \\ &= \frac{n^2}{n^2-1}\left(I - \frac{2}{n+1} cc^T\right).\end{aligned}$$

Therefore in this case,

$$E' = \left\{x \in R^n : (x-\hat{a})^T \hat{A}^{-1}(x-\hat{a}) \leq 1\right\}.$$

Since we only performed a rotation, the volume did not change. So volume $(E') \leq e^{-\frac{1}{2(n+1)}}$ volume $(E_0)$.

### *E. General Case*

Now what if $E$ is not the unit sphere but a general ellipsoid? The idea is to transform $E$ into unit sphere centered at origin via transform $T(x) = y$, apply the result of the previous case, then transform it back via $T^{-1}$ 5.

1) Fact 1: A is positive definite iff A = $B^T$B for some invertible B.
2) Fact 2: Invertible linear transformations preserve volume ratios.

Let $E = E_k = E(a_k, A_k)$. Since $A_k$ is positive definite, $A_k = B^T B$ for some $B$. Then $A_k^{-1} =$

$B^{-1}\left(B^{-1}\right)^T$ , and

$$E(a_k, A_k) = \left\{x : (x-a_k)^T B^{-1}\left(B^{-1}\right)^T (x-a_k) \leq 1\right\}$$

If we set $y = T(x) = \left(B^{-1}\right)^T (x-a_k)$, we will get

$$y^T y \leq 1$$

So $T$ transforms $E_k$ into $E(0,I)$.$T^{-1}(y) = x = B^T y + a_k$. The hyperplane in the original space $d^T x \leq d^T a_k$ becomes $d^T\left(B^T y + a_k\right) \leq d^T a_k$, thus $d^T B^T y \leq 0$ after the transform $T$. We want $c^T y \leq 0$ for $\|c\| = 1$, therefore set

$$c^T = \frac{d^T B^T}{\|d^T B^T\|},$$

hence

$$c = \frac{Bd}{\sqrt{d^T A d}}.$$

In the transformed space, we have

$$E' = \left\{ y : \left( y + \frac{1}{n+1} c \right)^T F^{-1} \left( y + \frac{1}{n+1} c \right) \leq 1 \right\}$$

where

$$F = \hat{A} = \frac{n^2}{n^2 - 1} \left( I - \frac{2}{n+1} cc^T \right)$$

Now substitute $y = \left(B^{-1}\right)^T (x - a_k)$ to get back to the original space. We have

$$M_1 = \left( (B^{-1})^T (x - a_k) + \frac{1}{n+1} c \right)^T$$

$$E_{k+1} = \{ x : M_1 F^{-1} M_1 \leq 1 \}$$

$$E_{k+1} = \{ x : \left( (x - a_k)^T B^{-1} + \frac{1}{n+1} c^T \right) F^{-1} M_1 \leq 1 \}$$

If we set $a_{k+1} = a_k - \frac{1}{n+1} B^T c$, then

$$E_{k+1} = \left\{ x : (x - a_{k+1})^T B^{-1} F^{-1} \left(B^{-1}\right)^T (x - a_{k+1}) \leq 1 \right\}$$

If we set $\hat{F}^{-1} = B^{-1} F^{-1} \left(B^{-1}\right)^T$, then

$$\begin{aligned} \hat{F} = B^T F B &= \frac{n^2}{n^2 - 1} B^T \left( I - \frac{2}{n-1} cc^T \right) B \\ &= \frac{n^2}{n^2 - 1} \left( A_k - \frac{2}{n+1} \left(B^T c\right) \left(B^T c\right)^T \right) \\ &= \frac{n^2}{n^2 - 1} \left( A_k - \frac{2}{n+1} bb^T \right) \end{aligned}$$

where we set $b = B^T c$. Then $a_{k+1} = a_k - \frac{b}{n+1}$, and $A_{k+1} = \hat{F} = \frac{n^2}{n^2-1} \left( A_k - \frac{2}{n+1} bb^T \right)$. Since the ratios of volumes are preserved under linear transformation,

$$\frac{\text{volume}\,(E_{k+1})}{\text{volume}\,(E_k)} = \frac{\text{volume}\,(E')}{\text{volume}\,(E_0)} \leq e^{-\frac{1}{2(n+1)}}$$

## IV. Putting It All Together

We now know how to find ellipsoid at each iteration. But there is still one question left which is what is the value of $M$. We take $M = O(n^2 ln(R/r))$ then

$$\frac{Vol(E_m)}{Vol(E_0)} \leq e^{\frac{-1}{2(n+1)} cn^2 ln(R/r)} \leq \left( \frac{r}{R} \right)^{2n}$$

Therefore, if all the $M$ iterations are covered, it means that $P = \emptyset$ as if P $\notin \emptyset$ then P $\supseteq$ B(a, r) which implies

$$\frac{Vol(E_m)}{Vol(E_0)} \geq \left( \frac{r}{R} \right)^n$$

This proves that Ellipsoid method solves feasibility problem in polynomial time.

## V. Polynomially running time: Avoiding the assumptions

In order to prove that the Ellipsoid Method can solve the system of linear inequalities in polynomial time, one has to generalize the basic ellipsoid algorithm to need not the assumptions that (1) polyhedron $P$ is bounded, (2) $P$ is either empty or full-dimensional and (3) that exact arithmetic is necessary.

In 1980, KHACHIYAN published a paper, [2], discussing all the details and proofs about the Ellipsoid Method which were neglected in his paper from 1979. In Lemma 1 , he showed that

$$P \cap S\left(0, 2^L\right) \neq \emptyset \tag{1}$$

in the case that $P \neq \emptyset$. With this, one can use for $R$ of value $2^L$. However, we cannot just adopt the algorithm above to the new situation. Instead of a lower bound for $\text{vol}(P)$, we would need a lower bound for $\text{vol}\left(P \cap S\left(0, 2^L\right)\right)$. To achieve this, we follow a trick introduced by KHACHIYAN and consider the perturbed system of linear inequalities

$$2^L a_i^\top x \leq 2^L \beta_i + 1 \quad i = 1, \ldots, m \tag{2}$$

Let us abbreviate the corresponding solution set with $P'$, which is in general a polyhedron. KHACHIYAN was able to proof a one-to-one correspondence of the original system and the perturbed one ( 2). This means that $P$ is empty if and only if $P'$ is empty, it is possible to construct a feasible point for original out of ( 2) in polynomial time and the formulation of the perturbed formulation is polynomial in $L$. Furthermore, the new inequality system has the additional property that if $P' \neq \emptyset$ it is full-dimensional. Hence, it is possible to find a (non-empty) sphere included in $P'$. It can be shown, that

$$S\left(\bar{x}, 2^{-2L}\right) \subseteq P' \cap S\left(0, 2^L\right) \tag{3}$$

where $\bar{x}$ is any feasible point of the original system and hence $\bar{x} \in P$. With this argument at hand, it is possible to derive an upper bound for the number of iterations for the Ellipsoid Method by solving the perturbed system ( 2). It can be shown that a feasible point can be found in at most $6n(n+1)L$ iterations, [3].

With the perturbation of the original system and property ( 1), we do no longer require that $P$ is bounded. As a byproduct, polyhedron $P$ has not to be of full-dimension in the case that it is non empty; as system ( 2) is of full-dimension independent of whether $P$ is or not, assuming that $P \neq \emptyset$. As a consequence, the basic ellipsoid algorithm can be generalized to apply for any polyhedron $P$ and the two major assumptions are no longer necessary.

During all of the reasoning, we assumed to have exact arithmetic, meaning that no rounding errors during the computation are allowed. This implies that all data have to be stored in a mathematically correct way. As we use the Turing-machine concept for the running time analysis, we require that all computations have to be done in finite precision. Let us now have a closer look for the reason why this is crucial for ellipsoid algorithm.

The presentation of the ellipsoid with the matrix $B_k$ yields to the convenient update formulas for the new ellipsoid, parameterized by $B_{k+1}$ and $x^{k+1}$. However, to obtain the new center $x^{k+1}$ one has to divide by factor $\sqrt{a_j^\top B_k a_j}$. If we work with finite precision, rounding errors are the consequence, and it is likely that matrix $B_k$ is no longer positive definite. This may cause that $a_j^\top B_k a_j$ becomes zero or negative, implying that the ellipsoid algorithm fails.

Hence, to implement the ellipsoid method, one has to use some modifications to make it numerically stable. One basic idea is to use factorization

$$B_k = L_k D_k L_k^\top$$

for the positive definite matrix $B_k$, with $L$ being a lower triangular matrix with unit diagonal and diagonal matrix $D_k$ with positive diagonal entries. Obtaining such a factorization is quite expensive as it is of order $n^3$. But there are update formulae applying for the case of the ellipsoid algorithm which have only quadratic complexity. Already in 1975, for such a type of factorization, numerically stable algorithms have been developed, insuring that $D_k$ remains positive definite, see **??**.

With a method at hand which can handle the ellipsoid algorithm in finite precision numerically stable, the proof of its polynomial running time is complete.

## VI. From Feasibility to Optimization

We have solved feasibility problem using Ellipsoid Method in polynomial time. Now we will show that feasibility can be reduced to optimization in polynomial time.

### A. Joint Feasibility

1) Check feasibility for Primal. If primal is infeasible, we are done.
2) Check feasibility for Dual. If dual is infeasible, we are done.
3) Otherwise setup joint feasibility LP. e.g. $Ax \leq B$, $x \geq 0$. $A^T y \geq c$, $y \geq 0$, $c^T x = y^T b$.

### B. Binary Search

We can do binary search on the objective value and put the objective in the constraints. Binary search takes O(log) time. Therefore complexity of optimization still remains polynomial.

## VII. Separation Oracle

We assumed during Ellipsoid Method that we have separation oracle. We went from separation to feasibility to optimization. But there can be a reverse path also.

Two examples of Separation Oracle are:

1) The minimum-cost arborescence problem: given a weighted directed graph and a vertex r in it, find a subgraph of minimum cost that contains a directed path from r to any other vertex. The problem can be presented as an LP with a constraint for each subset of vertices, which is an exponential number of constraints. However, a separation oracle can be implemented using n-1 applications of the minimum cut procedure.
2) The maximum independent set problem. It can be approximated by an LP with a constraint for every odd-length cycle. While there are exponentially-many such cycles, a separation oracle that works in polynomial time can be implemented by just finding an odd cycle of minimum length.

## VIII. Applications

### A. Linear Programming

As we have seen in the last section, the Ellipsoid Method solves the problem of finding a feasible point of a system. This problem is closely related to the problem of solving the linear program.

$$\begin{array}{ll} \max_x & c^\top x \\ \text{s.t.} & A^\top x \leq b, \\ & x \geq 0. \end{array} \tag{4}$$

We assume that all data are integral. In the following we briefly discuss two methods of how the optimization problem can be solved in polynomial time via the Ellipsoid Method. This will show that LP is in the class of polynomially solvable problems. From duality theory, it is well known that solving the linear optimization problem is equivalent to finding a feasible point of the following system of linear inequalities.

$$\begin{aligned} A^\top x &\leq b \\ -x &\leq 0 \\ -Ay &\leq -c, \\ -y &\leq 0 \\ -c^\top x + b^\top y &\leq 0 \end{aligned} \tag{5}$$

The third and fourth inequality come from the dual problem of Primal, insuring primal and dual feasibility of $x$ and $y$, respectively. The last inequality results from the Strong Duality Theorem, implying that this inequality always has zero slack. The equivalence of the two problems means in this case that vector $\bar{x}$ of each solution pair $(\bar{x}, \bar{y})$ of ( 5) is an optimal solution of problem ( 4) and $\bar{y}$ is an optimum for the dual problem of ( 4). In addition, to each solution of the optimization problem exists a vector such that this pair is feasible for problem ( 5).

From the equivalence of the two problems ( 4) and ( 5) we immediately conclude that the linear programming problem can be solved in polynomial time; as the input data of ( 5) are polynomially bounded in the length of the input data of ( 4). This argument was used by GÁCS and LOVÁSZ in their accomplishment to KHACHIYAN's work, see [4]. The advantage of this method is that the primal and dual optimization problem are solved simultaneously. However, note that with this method, one has no idea whether the optimization problem is infeasible or unbounded in the case when the Ellipsoid Method proves that problem ( 5) is infeasible. Another disadvantage is that the dimension of the problem increases from $n$ to $n + m$.

Next we discuss the so called bisection method which is also known as binary search or sliding objective hyperplane method. Starting with an upper and lower bound of an optimal

solution, the basic idea is to make the difference between the bounds smaller until they are zero or small enough. Solving system $A^\top x \leq b, x \geq 0$ with the Ellipsoid Method gives us either a vector $\bar{x}$ providing the lower bound $l := c^\top \bar{x}$ for problem ( 4) or in the case that the polyhedron is empty, we know that the optimization problem is infeasible. An upper bound can be obtained, for instance, by finding a feasible vector to the dual problem $Ay \geq c, y \geq 0$. If the Ellipsoid Method proves that the polytope of the dual problem is empty, we can use the duality theory (as we already know that problem ( 4) is not infeasible) to conclude that the optimization problem ( 4) is unbounded. In the other case we obtain vector $\bar{y}$ yielding to the upper bound $u := b^\top \bar{y}$ of problem ( 4), according to the Weak Duality Theorem. Once bounds are obtained, one can iteratively use the Ellipsoid Method to solve the modified problem $A^\top x \leq b, x \geq 0$ with the additional constraint

$-c^\top x \leq -\frac{u+l}{2}$

which is a constraint on the objective function value of the optimization problem. If the new problem is infeasible, one can update the upper bound to $\frac{u+l}{2}$, and in the case that the ellipsoid algorithm computes a vector $\bar{x}$, the lower bound can be increased to $c^\top \bar{x}$ which is greater or equal to $\frac{u+l}{2}$. In doing so, one at least bisects the gap in each step. However, this method does not immediately provide a dual solution. Note that only one inequality is added during the process, keeping the problem size small.

### B. Separation and Optimization

An interesting property of the ellipsoid algorithm is that it does not require an explicit list of all inequalities. In fact, it is enough to have a routine which solves the so called Separation Problem for a convex body $P$ :

Given $z \in R^n$, either conclude that $z \in P$ or give $a$ vector $\pi \in R^n$ such that inequality $\pi^\top x < \pi^\top z$ holds for all $x \in P$.

In the latter case we say that vector $\pi$ separates $z$ from $P$. In the following, we restrict the discussion to the case when $P$ is a polytope meeting the assumptions of ellipsoid algorithm. From the basic ellipsoid algorithm follows immediately that if one can solve the separation problem for polytope $P$ polynomially in $L$ and $n$, then the corresponding optimization problem

$$\begin{array}{ll} \max_x & c^\top x \\ \text{s.t.} & x \in P \end{array} \tag{6}$$

can also be solved in polynomial time in $L$ and $n; L$ is the encoding length of the input data. The converse statement was proven by GROETSCHEL et al., yielding to the following equivalence of separation and optimization:

The separation problem and the optimization problem over the same family of polytopes are polynomially equivalent.

Consider now an example to see how powerful the concept of separation is. Given a graph $G = (V, E)$ with node set $V$ and edges $e \in E$. A stable set $S$ of graph $G$ is defined as a subset of $V$ with the property that any two nodes of $S$ are not adjacent; which means that no edge between them exists in $E$. To look for a maximum one is the maximum stable set problem. This is a well known optimization problem and proven to be $\mathcal{NP}$ - hard, see [5]. It can be modeled, for instance, as the integer program

$$\begin{array}{ll} \max_x & c^\top x \\ \text{s.t.} & x_i + x_j \leq 1 \quad \forall (i,j) \in E \\ & x \in \{0,1\} \end{array} \tag{7}$$

with incidence vector $x$, meaning that $x_i = 1$ iff node $i$ is in a maximum stable set, otherwise it is zero. Constraints ( 7) are called edge inequalities. Relaxing the binary constraints for $x$ gives

$$0 \leq x \leq 1 \tag{8}$$

yielding to a linear program. However, this relaxation is very weak; consider therefore a complete graph. To improve it, one can consider the odd-cycle inequalities

$$\sum_{i \in C} x_i \leq \frac{|C| - 1}{2} \tag{9}$$

for each odd cycle $C$ in $G$. Recognize that there are in general exponentially many such inequalities in the size of graph $G$. Obviously, every stable set satisfies them and hence they are valid inequalities for the stable set polytope. The polytope satisfying the trivial-, edge- and odd-cycle inequalities

$$P := \left\{ x \in R^{|V|} \mid x \; satisfies (\; 7), (\; 8) and (\; 9) \right\} \tag{10}$$

is called the cycle-constraint stable set polytope. Notice that this polytope is contained strictly in the stable set polytope. It can be shown that the separation problem for polytope ( 10) can be solved in polynomial time. One idea is based on a construction of an auxiliary graph $H$ with a double number of nodes. Solving a sequence of $n$ shortest path problems on $H$ solves then the separation problem with a total running time of order $|V| \cdot |E| \cdot \log(|V|)$. With the equivalence of optimization and separation, the stable set problem over the cycle-constraint stable set polytope can be solved in polynomial time. This is quite a remarkable conclusion as the number of odd-cycle inequalities may be exponential. However, note that it does not imply that the solution will be integral and hence, we cannot conclude that the stable set problem can be solved in polynomial time. But we can conclude that the stable set problem for t-perfect graphs [1] can be solved in polynomial time.

## IX. Summary

We studied Ellipsoid Method, solved feasibility problem, went to optimization and proved its polynomial time complexity. The Ellipsoid Method seems to be a promising algorithm to solve problems practically. However, even though many modifications to the basic ellipsoid algorithm have been made, the worst case running time still remains a function in $n, m$ and especially $L$. This raises two main questions. First, is it possible to modify the ellipsoid algorithm to have a running time which is independent of the magnitude of the data, but instead depends only on $n$ and $m-$ or at least any other

algorithm with this property solving LPs? (This concept is known as strongly polynomial running time.) The answer to this question is still not known and it remains an open problem. In 1984 , KARMARKAR introduced another polynomial running time algorithm for LP which was the start of the Interior Point Methods, [6]. But also his ideas could not be used to solve this question. The second question, coming into mind, is how the algorithm performs in practical problems. Unfortunately, it turns out that the ellipsoid algorithm tends to have a running time close to its worst-case bound and is inefficient compared to other methods. The Simplex Method developed by GEORGE B. DANTZIG in 1947 , was proven by KLEE and MINTY to have exponential running time in the worst case, [KM72]. In contrast, its practical performance is much better and it normally requires only a linear number of iterations in the number of constraints.

Until now, the Ellipsoid Method has not played a role for solving linear programming problems in practice. However, the property that the inequalities themselves have not to be explicitly known, distinguishes the Ellipsoid Method from others, for instance from the Simplex Method and the Interior Point Methods. This makes it a theoretically powerful tool, for instance attacking various combinatorial optimization problems which was impressively shown by GRÖTSCHEL, LOVÁSZ and SCHRIJVER.

## REFERENCES


[1] L. G. Khachiyan, *A polynomial algorithm in linear programming*, 1979.

[2] ——, *Polynomial algorithms in linear programming*, 1980.

[3] D. G. Robert G. Bland and M. J. Todd, *The ellipsoid method: A survey*, 1981.

[4] P. G´acs and L. Lov´asz, *Khachiyan's algorithm for linear programming*, 1981.

[5] M. R. Garey and D. S. Johnson, *Computers and intractability, a guide to the theory of np-completeness.* 1979.

[6] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, 1984.