0.1 Cutting Plane Methods

The goal of this chapter is to present some polynomial-time algorithms for convex optimization. These algorithms use knowledge of the function in a minimal way, essentially by querying the function value and weak bounds on its support/range.

Given a continuously differentiable convex function f, Theorem **??** shows that

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle \text{ for all } y. \tag{1}$$

Let x^* be any minimizer of f. Replacing y with x^*, we have that

$$f(x) \geq f(x^*) \geq f(x) + \langle \nabla f(x), x^* - x \rangle .$$

Therefore, we know that $\langle \nabla f(x), x^* - x \rangle \leq 0$. Namely, x^* lies in a halfspace H with normal vector $-\nabla f(x)$. Roughly speaking, this shows that each gradient computation cuts the set of possible solutions in half. In one dimension, this allows us to do a binary search to minimize convex functions.

It turns out that in , binary search still works. In this chapter, we will cover several ways to do this binary search. All of them follow the same framework, called the *cutting plane method*. In this method, the convex set or function of interest is given by an oracle, typically a separation oracle: for any $x \notin K \subseteq^n$, the oracle finds a vector $v(x) \in^n$ such that

$$v(x)^\top (y - x) \leq 0 \text{ for all } y \in K.$$

Cutting plane methods address the following class of problems.

Problem 0.1 (Finding a point in a convex set)**.** Given $\epsilon > 0, R > 0$, and a convex set $K \subseteq RB^n$ specified by a separation oracle, find a point $y \in K$ or conclude that $K \leq \epsilon^n$. The complexity of an algorithm is measured by the number of calls to the oracle and the number of arithmetic operations.

Remark. To minimize a convex function, we set $v(x) = \nabla f(x)$ and K to be the set of (approximate) minimizers of f. In chapter 0.3, we relate the problem of proving that (K) is small to the problem of finding an approximate minimizer of f.

In this framework, we maintain a convex set $E^{(k)}$ that contains the set K. Each iteration, we choose some $x^{(k)}$ based on $E^{(k)}$ and query the oracle for $v(x^{(k)})$. The guarantee for $v(.)$ implies that K lies in the halfspace

$$H^{(k)} = \{y : g(x^{(k)})^\top (y - x^{(k)}) \leq 0\}$$

and hence $K \subset H^{(k)} \cap E^{(k)}$. The algorithm continues by choosing $E^{(k+1)}$ to be a convex set that contains $H^{(k)} \cap E^{(k)}$.

Algorithm 1: `CuttingPlaneFramework`

Input: Initial set $E^{(0)} \subseteq$ containing K.
for $k = 0, \cdots$ **do**
  Choose a point $x^{(k)} \in E^{(k)}$.
  if $E^{(k)}$ *is "small enough"* **then return** $x^{(k)}$.;
  Find $E^{(k+1)} \supset E^{(k)} \cap H^{(k)}$ where

$$H^{(k)}\{x \in \text{ such that } g(x^{(k)})^\top (x - x^{(k)}) \leq 0\}. \tag{2}$$

end

To analyze the algorithm, the main questions we need to answer are:

1. How do we choose $x^{(k)}$ and $E^{(k+1)}$?
2. How do we measure progress?
3. How quickly does the method converge?
4. How expensive is each step?

Progress on the cutting plane method is shown in the next table.

Year	$E^{(k)}$ and $x^{(k)}$	Iter	Cost/Iter
1965 [?, ?]	Center of gravity	n	n^n
1979 [?, ?, ?]	Center of ellipsoid	n^2	n^2
1988 [?]	Center of John ellipsoid	n	$n^{2.878}$
1989 [?]	Volumetric center	n	$n^{2.378}$
1995 [?]	Analytic center	n	$n^{2.378}$
2004 [?]	Center of gravity	n	n^4
2015 [?]	Hybrid center	n	n^2 (amortized)
2020 [?]	Volumetric center	n	n^2 (amortized)

Table 1: Different Cutting Plane Methods. Omitted polylogarithmic terms. The number of iterations follows from the rate.

■ 0.2 Ellipsoid Method

We start by explaining the Ellipsoid method. We maintain an ellipsoid

$$E^{(k)}\{x \in\colon\ (x - x^{(k)})^\top \left(A^{(k)}\right)^{-1} (x - x^{(k)}) \le 1\}$$

that contains K. After we compute $g(x^{(k)})$ and $H^{(k)}$ via (2), we define $E^{(k+1)}$ to be the smallest volume ellipsoid containing $E^{(k)} \cap H^{(k)}$. The key observation is that the volume of the ellipsoid $E^{(k)}$ decreases by $1 - \Theta(\frac{1}{n})$ every iteration. This volume property holds for any halfspace through the center of the current ellipsoid (not only for the one whose normal is the gradient), a property we will exploit in the next chapter.

Algorithm 2: `Ellipsoid`

Input: Initial ellipsoid $E^{(0)} = \{x \in\colon\ (x - x^{(0)})^\top \left(A^{(0)}\right)^{-1} (x - x^{(0)}) \le 1\}$.
for $k = 0, \cdots$ **do**
  if $E^{(k)}$ *is "small enough"* **then return** $x^{(k)}$.;
  $x^{(k+1)} = x^{(k)} - \frac{1}{n+1} \frac{A^{(k)} v(x^{(k)})}{\sqrt{v(x^{(k)})^\top A^{(k)} v(x^{(k)})}}$.
  $A^{(k+1)} = \frac{n^2}{n^2-1} \left(A^{(k)} - \frac{2}{n+1} \frac{A^{(k)} v(x^{(k)}) v(x^{(k)})^\top A^{(k)}}{v(x^{(k)})^\top A^{(k)} v(x^{(k)})}\right)$.
end

Lemma 0.2. *For the Ellipsoid method (Algorithm 2), we have $E^{(k+1)} < e^{-\frac{1}{2n+2}} E^{(k)}$ and $E^{(k)} \cap H^{(k)} \subset E^{(k+1)}$.*

Remark 0.3. Note that the proof below also shows that $E^{(k+1)} = e^{-\Theta(\frac{1}{n})} E^{(k)}$. Therefore, the ellipsoid method does not run faster for nice functions, making it provably slow in practice.

Proof. Note that the ratio of $E^{(k+1)}/E^{(k)}$ does not change under any affine transformation. Therefore, we can do a transformation so that $A^{(k)} = I$, $x^{(k)} = 0$ and $v(x^{(k)}) = e_1$. This simplifies our calculation. We need to prove two statements: $E^{(k+1)} < e^{-\frac{1}{2n+2}} E^{(k)}$ and that $E^{(k)} \cap H^{(k)} \subset E^{(k+1)}$.

Claim 1: $E^{(k+1)} < e^{-\frac{1}{2(n+1)}} E^{(k)}$.

Note that since $v(x^{(k)}) = e_1$,we have $A^{(k+1)} = \frac{n^2}{n^2-1} \left(I - \frac{2}{n+1} e_1 e_1^\top\right)$. Therefore, we have that

$$\begin{aligned}
\left(\frac{E^{(k+1)}}{E^{(k)}}\right)^2 = \left|\frac{\det A^{(k+1)}}{\det A^{(k)}}\right| &= \left(\frac{n^2}{n^2-1}\right)^n \det\left(I - \frac{2}{n+1} e_1 e_1^\top\right) \\
&= \left(\frac{n^2}{n^2-1}\right)^{n-1} \frac{n^2}{n^2-1} \left(\frac{n-1}{n+1}\right) \\
&= \left(1 + \frac{1}{n^2-1}\right)^{n-1} \left(1 - \frac{1}{n+1}\right)^2 \\
&< \exp\left(\frac{n-1}{n^2-1} - \frac{2}{n+1}\right) = \exp\left(-\frac{1}{n+1}\right)
\end{aligned}$$

where we used $1+x \leq e^x$ for all x.

Claim 2. $E^{(k)} \cap H^{(k)} \subset E^{(k+1)}$.

By the definition of $E^{(k)}$and the assumption $A^{(k)} = I$, we have

$$x^{(k+1)} = -\frac{e_1}{n+1}$$

and, for any $x \in E^{(k)} \cap H^{(k)}$, we have that $\|x\|_2 \leq 1$ and $x_1 \leq 0$. By direct computation, using the fact that

$$\left(I - \frac{2}{n+1} e_1 e_1^\top\right)^{-1} = \left(I + \frac{2}{n-1} e_1 e_1^\top\right)$$

we have

$$\begin{aligned}
&(x + \frac{e_1}{n+1})^\top \left(1 - \frac{1}{n^2}\right) \left(I + \frac{2}{n-1} e_1 e_1^\top\right) (x + \frac{e_1}{n+1}) \\
=& \left(1 - \frac{1}{n^2}\right) \left(\|x\|^2 + \frac{2x_1(1+x_1)}{n-1} + \frac{1}{n^2-1}\right) \\
\leq& \left(1 - \frac{1}{n^2}\right) \left(1 + 0 + \frac{1}{n^2-1}\right) = 1
\end{aligned}$$

where we used that $\|x\|_2 \leq 1$ and $x_1(1+x_1) \leq 0$ (since $-1 \leq x_1 \leq 0$) at the end. This shows that $x \in E^{(k+1)}$. □

Exercise 0.4. Show that the ellipsoid $E^{(k+1)}$ computed above is the minimum volume ellipsoid containing $E^{(k)} \cap H^{(k)}$.

■ 0.3 From Volume to Function Value

Lemma 0.2 shows that the volume of a set containing all minimizers decreases by a constant factor every n steps. In general, knowing that the optimal x^* lies in a small volume set does not provide enough information to find a point with small function value. For example, if we only knew that x^* lies in the plane $\{x : x_1 = 0\}$, we would still need to search for x^* over an $n-1$ dimensional space. However, if the set is constructed by the cutting plane framework (Algorithm 1), then we can guarantee that small volume implies that any point in the set has close to optimal function value.

This in turns implies that we can minimize any convex function with ε additive error in $O(n^2 \log(1/\varepsilon))$ iterations. To make the statement more general, we note that ellipsoid method can be used for non-differentiable functions.

Theorem 0.5. *Let $x^{(k)}$ be the sequence of points produced by the cutting plane framework (Algorithm 1) for a convex function f. Let $\mathcal{V}$ be a mapping from subsets of to non-negative numbers satisfying*

1. *(Linearity) For any set $S \subseteq^n$, any vector y and any scalar $\alpha \geq 0$, we have $\mathcal{V}(\alpha S + y) = \alpha \mathcal{V}(S)$ where $\alpha S + y = \{\alpha x + y : x \in S\}$.*
2. *(Monotonic) For any set $T \subset S$, we have that $\mathcal{V}(T) \leq \mathcal{V}(S)$.*

Then, for any set $\Omega \subseteq E^{(0)}$, we have that

$$\min_{i=1,2,\cdots k} f(x^{(i)}) - \min_{y \in \Omega} f(y) \leq \frac{\mathcal{V}(E^{(k)})}{\mathcal{V}(\Omega)} \cdot \left(\max_{z \in \Omega} f(z) - \min_{x \in \Omega} f(x)\right) .$$

Remark 0.6. We can think $\mathcal{V}(E)$ as some way to measure the size of E. It can be radius, mean-width or any other way to measure "size". For the ellipsoid method, we use $\mathcal{V}(E) = (E)^{\frac{1}{n}}$ for which we have proved volume decrease in Lemma 0.2. We raise the volume to power $1/n$ to satisfy linearity.

Proof. Let x^* be any minimizer of f over Ω. For any $\alpha > \frac{\mathcal{V}(E^{(k)})}{\mathcal{V}(\Omega)}$ and $S = (1-\alpha)x^* + \alpha\Omega$, by the linearity of $\mathcal{V}$, we have that

$$\mathcal{V}(S) = \alpha\mathcal{V}(\Omega) > \mathcal{V}(E^{(k)})$$

Therefore, S is not a subset of $E^{(k)}$ and hence there is a point $y \in S\backslash E^{(k)}$. y is not in $E^{(k)}$. This means it is separated by the gradient at some step $i \leq k$, namely for some $i \leq k$, we have

$$\nabla f(x^{(i)})^\top (y - x^{(i)}) > 0.$$

By the convexity of f, it follows that $f(x^{(i)}) \leq f(y)$. Since $y \in S$, we have $y = (1-\alpha)x^* + \alpha z$ for some $z \in \Omega$. Thus, the convexity of f implies that

$$f(x^{(i)}) \leq f(y) \leq (1-\alpha)f(x^*) + \alpha f(z).$$

Therefore, we have

$$\min_{i=1,2,\cdots k} f(x^{(i)}) - \min_{x\in\Omega} f(x) \leq \alpha \left(\max_{z\in\Omega} f(z) - \min_{x\in\Omega} f(x) \right).$$

Since this holds for any $\alpha > \frac{\mathcal{V}(E^{(k)})}{\mathcal{V}(\Omega)}$, we have the result. $\square$

Combining Lemma 0.2 and Theorem 0.5, we have the following rate of convergence.

Theorem 0.7. *Let f be a convex function on , $E^{(0)}$ be any initial ellipsoid and $\Omega \subset E^{(0)}$ be any convex set. Suppose that for any $x \in E^{(0)}$, we can find, in time $\mathcal{T}$, a nonzero vector $g(x)$ such that*

$$f(y) \geq f(x) \text{ for any } y \text{ such that } g(x)^\top (y - x) \geq 0.$$

Then, we have

$$\min_{i=1,2,\cdots k} f(x^{(i)}) - \min_{y\in\Omega} f(y) \leq \left(\frac{(E^{(0)})}{(\Omega)} \right)^{\frac{1}{n}} \exp\left(-\frac{k}{2n(n+1)} \right) \left(\max_{z\in\Omega} f(z) - \min_{x\in\Omega} f(x) \right).$$

Moreover, each iteration takes $O(n^2 + \mathcal{T})$ time.

Remark 0.8. We usually obtain $g(x)$ from a *separation oracle*.

Proof. Lemma 0.2 shows that the volume of the ellipsoid maintained decreases by a factor of $\exp(-\frac{1}{2n+2})$ in every iteration. Hence, $\frac{1}{n}$ decreases by $\exp(-\frac{1}{2n(n+1)})$ every iteration. The bound follows by applying Theorem 0.5 with $\mathcal{V}(E)(E)^{\frac{1}{n}}$. Next, we note the proof of Theorem 0.5 only used the fact that one side of halfspace defined by the gradient has higher value. Therefore, we can replace the gradient with the vector $g(x)$. $\square$

This theorem can be used to solve many problems in polynomial time. As an illustration, we show how to solve linear programs in polynomial time here.

Theorem 0.9. *Given a linear program $\min_{x\in\Omega} c^\top x$ where $P = \{x \text{ such that } Ax \geq b\}$. Let the diameter of P be $R \max_{x\in\Omega} x_2$ and its volume radius be $r = (\Omega)^{\frac{1}{n}}$. Then, we can find $x \in P$ for which*

$$c^\top x - \min_{y\in P} c^\top y \leq \varepsilon \cdot (\max_{y\in P} c^\top y - \min_{y\in P} c^\top y)$$

in $O(n^2(n^2 + (A))\log(\frac{R}{r\varepsilon}))$ time where (A) is the number of non-zero elements in A.

Remark 0.10. If the dimension n is constant, this algorithm is nearly linear time (linear to the number of constraints)!

Proof. For the linear program $\min_{Ax\geq b} c^\top x$, the function we want to minimize is

$$L(x) = c^\top x + \delta_{Ax\geq b}(x) \quad \text{where} \quad \delta_{Ax\geq b}(x) = \begin{cases} 0 & \text{if } a_i^\top x \geq b_i \text{ for all } i \\ +\infty & \text{otherwise.} \end{cases}. \tag{3}$$

For this function L, we can use the separation oracle $v(x) = c$ if $Ax \geq b$ and $v(x) = -a_i$ if $a_i^\top x < b_i$. If there are many violated constraints, any one of them will do.

We can simply pick $E^{(0)}$ be the unit ball centered at 0 with radius R. We apply Theorem 0.7 to find x such that

$$c^\top x - \min_{y\in P} c^\top y \leq \varepsilon \cdot (\max_{y\in P} c^\top y - \min_{y\in P} c^\top y)$$

in time $O(n^2(n^2 + (A))\log(\frac{R}{r\varepsilon}))$. □

Exercise 0.11. Give a bound on R as a polynomial in $n, \langle A\rangle, \langle b\rangle$, where $\langle A\rangle, \langle b\rangle$ are the numbers of bits needed to represent A, b whose entries are rationals.

To get the solution exactly, i.e., $\varepsilon = 0$, we need to assume the linear program has integral (or rational) coefficients and then the running time will depend on the sizes of the numbers in the matrix A and in the vectors b and c. It is still open how to solve linear programs in time bounded by a polynomial in only the number of variables and constraints (and not the bit sizes of the coefficients). Such a running time is called *strongly polynomial.*

Open Problem. Can we solve linear programs in strongly polynomial time?

■ 0.4 Center of Gravity Method

In the Ellipsoid method, we used an ellipsoid as the current set, and its center as the next query point.

In the center of gravity method, we start with any bounded convex set containing a minimizer, e.g., a large enough cube, and the set maintained is simply the intersection of all halfspaces used so far. The query point will be the center of gravity (or barycenter) of the current set. The measure of progress will once again be the volume (or more precisely, volume radius) of the current set.

It is clear that the volume can only decrease in each iteration. But at what rate? The following classical theorem shows that the volume of the convex body decreases by at least a $1 - \frac{1}{e}$ factor when using the exact center of gravity.

Theorem 0.12. *Let K be a convex body in ^{n} with center of gravity z. Let H be any halfspace containing z. Then,*

$$(K \cap H) \geq \left(\frac{n}{n+1}\right)^n (K).$$

Note that the RHS is at least $1/e$. We prove the theorem later in this chapter. Unfortunately, computing the center of gravity even of a polytope is #P-hard [**?**]. For the purpose of efficient approximations, it is important to establish a stable version of the theorem that does not require an exact center of gravity.

Recall that a nonnegative function is logconcave if its logarithm is concave, i.e., for any $x, y \in^n$ and any $\lambda \in [0, 1]$,

$$f(\lambda x + (1-\lambda)y) \geq f(x)^\lambda f(y)^{1-\lambda}.$$

We refer to Sections **??** for some background on logconcave functions. A distribution p is *isotropic* if the mean of a random variable drawn from the distribution is zero and the covariance is the identity matrix.

The randomized center of gravity method defined as follows: Let $y^1, \ldots, y^N$ be uniform random points from $E^{(k)}$ and define

$$x^{(k)} = \frac{1}{N}\sum_{i=1}^{N} y^i,$$
$$E^{(k+1)} = E^{(k)} \cap H^{(k)}$$

To prove its convergence, we will use a robust version of the following classical theorem of Grunbaum.

Theorem 0.13 (Robust Grunbaum)**.** *Let p be an isotropic logconcave distribution, namely ${}_{x\sim p}x = 0$ and ${}_{x\sim p}x^2 = 1$. For any $\theta \in^n$, $t \in$ we have*

$$\P_{x\sim p}(x^\top\theta \geq t) \geq \frac{1}{e} - t.$$

Proof. By taking the marginal with respect to the direction θ, we can assume the distribution is one-dimensional. Let $P(t) = \P_{x\sim p}(x^\top\theta \leq t)$. Note that $P(t)$ is the convolution of p and $1_{(-\infty,0]}$. Hence, it is logconcave (Lemma **??**). By some limit arguments, we can assume $P(-M) = 0$ and $P(M) = 1$ for some large enough M (to be rigorous, we do the proof below for finite M and a RHS $\epsilon(M)$ instead of zero, then take the limit $M \to \infty$). Since ${}_{x\sim p}x = 0$, we have that

$$\int_{-M}^{M} tP'(t)\, dt = 0$$

Integration by parts gives that $\int_{-M}^{M} P(t)dt = M$. Note that $P(t)$ is increasing logconcave, if $P(0)$ is too small, it would make $\int_{-M}^{M} P(t)dt$ too small. To be precise, since P is logconcave, we have that

$$-\log P(t) \geq -\log P(0) - \frac{P'(0)}{P(0)}t.$$

Or we simply write $P(t) \leq P(0)e^{\alpha t}$ for some α. Hence,

$$M = \int_{-M}^{M} P(t)dt \leq \int_{-\infty}^{\frac{1}{\alpha}} P(0)e^{\alpha t}dt + \int_{1/\alpha}^{M} 1dt = \frac{eP(0)}{\alpha} + M - \frac{1}{\alpha}.$$

This shows that $P(0) \geq \frac{1}{e}$.

Next, Lemma 0.14 shows that $\max_x p(x) \leq 1$. Therefore, the cumulative distribution P is 1-Lipschitz and we have

$$\P_{x\sim p}(x^\top\theta \geq t) \geq \P_{x\sim p}(x^\top\theta \geq 0) - t \geq \frac{1}{e} - t.$$

□

Lemma 0.14. *Let p be a one-dimensional isotropic logconcave density. Then* $\max p(x) \leq 1$.

For a proof of this (and for other properties of logconcave functions), we refer the reader to [**?**].

Exercise 0.15. Give a short proof that $\max_x p(x) = O(1)$ for any one-dimensional isotropic logconcave density.

Using the robust Grunbaum theorem 0.13, we get the following algorithm, which uses uniform random points from the current set. Obtaining such a random sample algorithmically is an interesting problem that we will study in the second part of this book.

Lemma 0.16. *Suppose $y^1, \ldots, y^N$ are i.i.d. uniform random points from a convex body K and $y = \frac{1}{N}\sum_{i=1}^{N} y^i$. Then for any halfspace H not containing y,*

$$((K \cap H)) \leq \left(1 - \frac{1}{e} + \sqrt{\frac{n}{N}}\right)(K).$$

Proof. Without loss of generality, we assume that K is in isotropic position, i.e., ${}_K(y^i) = 0$ and ${}_K(y^i(y^i)^\top) = I$. Then we have $(y) = 0$ and

$$y^2 = \frac{1}{N}y^{i2} = \frac{n}{N}.$$

Therefore,

$$y \leq \sqrt{(y^2)} = \sqrt{\frac{n}{N}}.$$

Thus, we can apply Theorem 0.13 with $t = \sqrt{\frac{n}{N}}$. □

Theorem 0.5 readily gives the following guarantee for convex optimization, again using volume radius as the measure of progress.

Theorem 0.17. *Let f be a convex function on , $E^{(0)}$ be any initial set and $\Omega \subset E^{(0)}$ be any convex set. Suppose that for any $x \in E^{(0)}$, we can find a nonzero vector $g(x)$ such that*

$$f(y) \geq f(x) \text{ for any } y \text{ such that } g(x)^\top (y - x) \geq 0.$$

Then, for the center of gravity method with $N = 10n$, we have

$$\min_{i=1,2,\cdots k} f(x^{(i)}) - \min_{y \in \Omega} f(y) \leq \left(\frac{(E^{(0)})}{(\Omega)} \right)^{\frac{1}{n}} (0.95)^{\frac{k}{n}} \left(\max_{z \in \Omega} f(z) - \min_{x \in \Omega} f(x) \right) .$$

Now, we give a geometric proof Theorem 0.12. Note that one can modify the proof of Theorem 0.13 to get another proof.

Proof. Without loss of generality, assume that the center of gravity of K is the origin and H is the halfspace $\{x : x_1 \leq 0\}$. For each point $t \in$, let $A(t) = K \cap \{x : x_1 = t\}$ be the $(n-1)$-dimensional slice of K with $x_1 = t$. Define $r(t)$ as the radius of the $(n-1)$-dimensional ball with the same $(n-1)$-dimensional volume as $A(t)$.

The goal of the proof is to show that the smallest possible halfspace volume is achieved for a cone by a cut perpendicular to its axis. In the first step, we will symmetrize K as follows: replace each cross-section $A(t)$ by a ball of the same volume, centered at $(t, 0, \ldots, 0)^T$. We claim that the resulting rotationally symmetric body is convex. To see this, note that all we have to show is that the radius function $r(t)$ is concave. For any $s, t \in$, and any $\lambda \in [0, 1]$, we have by convexity of K that

$$\lambda A(s) + (1 - \lambda) A(t) \subseteq A(\lambda s + (1 - \lambda) t)$$

and by the Brunn-Minkowski theorem (Theorem **??**) applied to $A(s), A(t)$, we have

$$_{n-1}(A(\lambda s + (1 - \lambda) t))^{\frac{1}{n-1}} \geq \lambda_{n-1}(A(s))^{\frac{1}{n-1}} + (1 - \lambda)_{n-1}(A(t))^{\frac{1}{n-1}}.$$

From this and the definition of $r(t)$, it follows that

$$r(\lambda s + (1 - \lambda) t) \geq \lambda r(s) + (1 - \lambda) r(t)$$

as desired.

Next consider the subset $K_1 = K \cap \{x : x_1 \leq 0\}$. We replace this subset with a cone C having the same base $A(0)$ and apex at some point along the e_1 axis so that the volume of the cone is the same as (K_1). Using the concavity of the radial function, this transformation can only decrease the center of gravity along e_1. Therefore, proving a lower bound on the transformed body K_1 will give a lower bound for K. So assume we do this and the center of gravity is the origin. Next, extend the cone to the right, so that it remains a rotational cone, and the volume in the positive halfspace along e_1 is the same as $(K \setminus K_1)$. Once again, the center of gravity can only move to the left, and so the volume of K_1 can only decrease by this transformation. At the end we have shown that the lower bound for any convex body follows by proving for a rotational cone with axis along the normal to the halfspace. Now we compute the volume ratio:

$$\frac{(K_1)}{(K)} = \left(\frac{n}{n+1} \right)^n .$$

□

To conclude this section, we note that the number of separation oracle queries made by the center-of-gravity cutting plane method is asymptotically the best possible.

Theorem 0.18. *Any algorithm that solves Problem 0.1 using a separation oracle needs to make $\Omega(n \log(R/\epsilon))$ queries to the oracle.*

Proof. Suppose K is a cube of side length ϵ contained in the cube $[0, R]^n$. Imagine a tiling of the big cube by cubes of side length ϵ. Consider the oracle that always returns an axis parallel halfspace that does not cut any little cube and contains at least half of the volume of the "remaining" region, i.e., the set given by the original cube intersected with all halfspaces given by the oracle so far. This is always possible since for any halfspace either the halfspace or its complement will contain at least half the volume of any set. Thus each query at best halves the remaining volume. To solve the problem, the algorithm needs to cut down to a set of volume ϵ^n starting from a set of volume R^n. Thus it needs at least $n \log_2(R/)$ queries. $\square$

Discussion

In later chapters we will see how to implement each iteration of the center of gravity method in polynomial time. Computing the exact center of gravity is #P-hard even for a polytope [**?**], but we can find an arbitrarily close approximation in randomized polynomial time via sampling. The method generalizes to certain noisy computation models, e.g., when the oracle reports a function value that is within a bounded additive error, i.e., a noisy function oracle.