Fall 2016, MATH-566

Simplex method is not polynomial time - Klee-Minty Cube

Source: Chapter 5 of Linear and Nonlinear Programming, Luenberger and Ye

Simplex method outline: Convert problem to $A\mathbf{x} = \mathbf{b}$. Find a basic feasible solution. Perform pivot steps until no variable can increase.

Geometry of simplex method - start at vertex of the polytope of feasible solutions and find a path on edges of the polytope to the optimal vertex.

Consider greedy way - always go in the direction that maximizes the slope in the objective function.

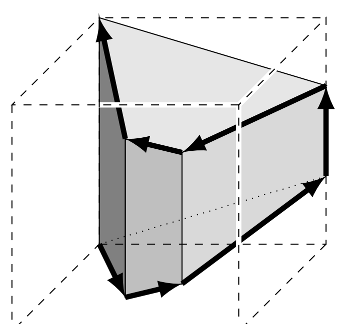
$$
(P)\begin{cases}
\text{maximize} & 100x_1 + 10x_2 + x_3 & & \\
\text{subject to} & x_1 \leq 1 & (A) \\
& 20x_1 + x_2 \leq 100 & (B) \\
& 200x_1 + 20x_2 + x_3 \leq 10000 & (C) \\
& x_1 \geq 0 & (D) \\
& x_2 \geq 0 & (E) \\
& x_3 \geq 0 & (F)
\end{cases}
$$

Notice that a vertex is given by intersection of three of the halfspaces. That is, pick three of the equations to be satisfied with equality and it gives a vertex.

Steps in simplex method:

step	x_1	x_2	x_3	value of objective	equalities
0	0	0	0	0	$(D),(E),(F)$
1	1	0	0	100	$(A),(E),(F)$
2	1	80	0	900	$(A),(B),(F)$
3	0	100	0	1000	$(D),(B),(F)$
4	0	100	8000	9000	$(D),(B),(C)$
5	1	80	8200	9100	$(A),(B),(C)$
6	1	0	9800	9900	$(A),(E),(C)$
7	0	0	10000	10000	$(D),(E),(C)$

Corresponds to a travel in cube



How many vertices will be in n dimensional cube?

1: *If $n = 50$ and computer examines one million points in one second, how long will it take to finish the computation?*

Solution: $n = 50$ gives about 10^{15} vertices. It gives 33 years.

Klee-Minty cubes are known for different rules too. But the simplex algorithm works great in practice.

The Ellipsoid Method

Problem: Let $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$. Find a point in P. (given a polytope, find one point in it)

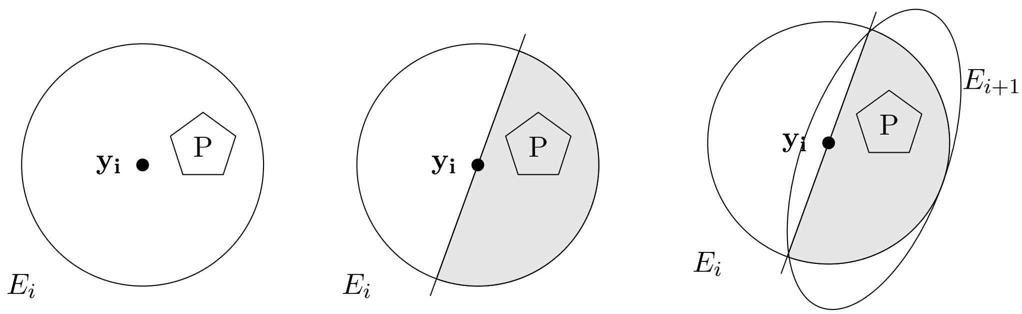
Extra assumptions:

- $\exists R \in \mathbb{R}, P \subseteq B(\mathbf{0}, R)$
- $\exists r \in \mathbb{R}, \exists \mathbf{c} \in \mathbb{R}^n, B(\mathbf{c}, r) \subseteq P$

In other words, P is in a big ball with radius R and contains a small ball of radius r.
The R and r are part of the running time.

Algorithm:

1. $E_1 := B(0, R)$, $i := 1$
2. if center $\mathbf{y}_i$ of E_i in P, point found
3. if $\mathbf{y}_i \notin P$, there is a separating hyperplane cutting out half of E_i
4. Pick E_{i+1} to be the smallest ellipsoid containing the half of E_i that contains P
5. $i := i + 1$ and goto 2.



Claim: If $E_i \in \mathbb{R}^n$ and E_{i+1} is the smallest ellipsoid contain $\frac{1}{2}$ of E_i, then

$$\frac{volume(E_{i+1})}{volume(E_i)} < e^{\frac{-1}{2(n+1)}} < 1.$$

2: *Compute an upper bound on*

$$\frac{volume(E_{i+2(n+1)})}{volume(E_i)}$$

Solution:

$$\frac{volume(E_{i+2(n+1)})}{volume(E_i)} < \left(e^{\frac{-1}{2(n+1)}}\right)^{2(n+1)} = e^{-1}.$$

3: *How many iterations of the algorithm are needed? (Use that $B(\mathbf{c}, r) \subset P$.)*

Solution: If volume of $volume(E_i) < volume(B(\mathbf{c}, r))$, we would get a contradiction since $B(\mathbf{c}, r) \subset P \subset E_i$. We need $O(n)$ steps to reduce volume by half and we do it $\log\left(\frac{volume(B,R)}{volume(B,r)}\right)$ times.

$$O\left(n \cdot \log\left(\frac{R^n}{r^n}\right)\right) = O\left(n^2 \log\left(\frac{R}{r}\right)\right)$$

One iteration takes $O(n^2)$ operations and volume of balls is at most exponential (in size of input numbers). Not a practical algorithm in speed.