FINAL PROJECT

ZHEQING ZHANG

Shadow Simplex Method and its Implementation

Abstract. The author presents a randomized simplex algorithm method for solving LP problem, whose running time depends polynomials on the number of bits used to represent its input. In the first section, I will introduce the shadow simplex method and show we can reduce the LP to a special form in which we only need the boundedness of the feasible region. In the second section, we will first prove the useful intuition that the projection of the perturbed polytope on the shadow plane has polynomially many edges and then discuss the algorithm and its implementations.

1. **Introduction**

Linear programming is one of the most fundamental problem in optimization. There are many methods such as simplex method, interior point method, ellipsoid method developed to deal with the LP. Despite that simplex method, developed by Dantzig in 1940s, widely studied for decades and applied in various fields, is one of the oldest methods, it is still very popular now. It is widely accepted that simplex method will run in polynomial time in worst cases. In contrast, algorithms with polynomial run time in worst cases based on ellipsoid method and interior point methods have been developed. However, the two methods in mention differ a lot from the simplex one, since they take advantages of geometric of the feasible region, while simplex method only consider the paths along the edges. And in practical use, simplex method usually beats ellipsoid method even if the latter one has short run time theoretically. So we can ask whether we can modify the simplex method such that it can have polynomial run time like other methods.

In this report, I mainly consider the LP of the form,

$$\begin{aligned} \max \quad & c \cdot x \\ \text{sub} \quad & Ax \leq b, \quad x \in \mathbb{R}^d \end{aligned} \tag{1}$$

where $c \in \mathbb{R}^d$ and $b \in R^n$, and A is an $n \times d$ matrix. And we denote $\{a_1, a_2..., a_n\}$ as the rows of A. Intuitively, the optimal x can only happen at corners of feasible region $P = x|Ax \leq b$, provided that the max is finite. The original simplex method uses some pivot rules to guide its walk and terminates when it arrives at the maximizer. In order to show that the simplex method can run within polynomial steps, we must prove the diameter of the polytope has a polynomial upper bound, which still remains open. Some simplex methods try to consider some polytopes with better diameter. However, they have exponentially many vertices, and no one can guarantee that we can find a path from start point to maximizer with polynomial distance.

2. **Shadow Simplex Method**

Let P be a convex polytope, and let S be a two-dimensional subspace. The projection of P on S is called the shadow of P onto S. The shadow is a polygon and every vertex(edge) of shadow is projected from one vertex(edge) on P (if the projection is non-degenerate). We will use some facts in our implementations.

1. For any vertex in shadow(P), there exists an objective function c in S, such that the pre-image of the vertex maximize c in the polytope P.

2. The edge defined by $a_i x = b_i$, $i \in I$ is projected to be an edge in shadow(P) if and only if the shadow plane S cuts the cone $\{a_i\};\ i \in I$.

The shadow-vertex method runs like follows,

1. Pick a initial point v_0.

2. Find a objective f, which is maximized by v_0 in the polytope P. Then construct the shadow plane $S = span\{c, f\}$.

3. Consider the shadow on S. The shadow is a polygon with shadow$\{v_0\}$ being the start vertex. Start from shadow$\{v_0\}$ and walk clockwise until we find the maximizer x_1 of c in S. By the above facts, the pre image of x_1 is the maximizer of the original problem.

Possibly, the number of edges projected on S will be exponentially large. In ordinary simplex method, we can't get rid of the possibility. However, here I find a randomized simplex algorithm which perturbs the right hand side of the constraint so that the above abnormality will not happen in high probability.

3. Polynomial-Time Simplex Algorithm

First I will state the main thm used in the algorithm[1], and some lemma useful to prove the thm.

1. We consider the perturbed polytope of $P = \{x|Ax \leq 1\}$, $Q = \{x|a_i \cdot x \leq 1 + r_i\}$, where r_i has the property that $Prob(r_i > t) = e^{-\frac{t}{\lambda}}$. Let v and w be uniformly random unit vectors, and let V be their span. Then, the expectation over v, w, and the r_i of the number of facets of the projection of Q onto V is at most
$$\frac{12\pi k(1 + \lambda ln(\pi e))\sqrt{d}n}{\lambda}. \tag{2}$$
2. This is a generalization of the first thm, it gives the upper of the number of edges on the projection of Q on shadow with v near fixed vector u.

We can intuitively know that since we have polynomially many shadow vertices, the worst case is walking through all of them. If the optimal solution is finite, we only have to take polynomially many steps.

4. **Implementation on 3D case**

I am not willing to use the randomized polynomial time method to solve the following problem, since it is too simple. And also in the implementation of shadow vertex method, I used too many 'if' which will consume a lot of time in matlab. If I determine to compare the efficiency, I am afraid the ordinary simplex tabular method will win.

Problem:

$$\begin{aligned}
\min / \max & c_1x_1 + c_2x_2 + c_3x_3 \\
\text{sub to} & x_1 \leq 200 \\
& x_2 \leq 300 \\
& x_1 + x_2 + x_3 \leq 400 \\
& x_2 + 3x_3 \leq 600 \\
& x_1 \geq 0 \\
& x_2 \geq 0 \\
& x_3 \geq 0 \\
& x_1 - x_2 + x_3 \leq 250 \\
& x_1 - x_2 - x_3 \leq 150 \\
& -x_1 + x_2 - 1.1x_3 \leq 250
\end{aligned} \tag{3}$$

I will first try to solve the problem by tabular method and then try to solve it by shadow simplex method. Let $\vec{c} = [4, -5, 1]$.

1. Tabular method . (The matlab code is attached and tabularM.m and singl.m)
The start point we choose is the origin.
From the following graph we can see the optimal solution of the max and min problem.

Optimal solutions

	x value	cx	x_0	iteration
maxcx	$x_{min} = (0, 30, 45.4545)$	850	(0,0,0)	3
mincx	$x_{max} = (200, 0, 50)$	-1454	(0,0,0)	3

Table 1. The optimal solution.

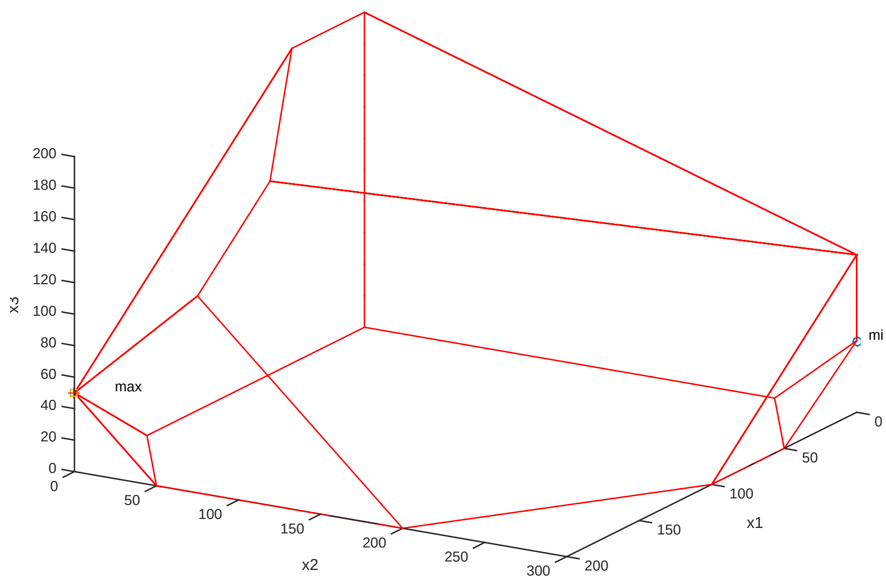

FIGURE 1. vertices of polygon on $span(c, f)$

Here the following are evolution tables for obtaining the maximizer of x.

First tabular

x_1	x_2	x_3	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	b
0	1	1	1	0	0	0	0	0	0	0	-1	0	50
0	1	0	0	1	0	0	0	0	0	0	0	0	300
0	2	2	0	0	1	0	0	0	0	0	-1	0	250
0	1	3	0	0	0	1	0	0	0	0	0	0	600
0	-1	-1	0	0	0	0	1	0	0	0	1	0	150
0	-1	0	0	0	0	0	0	1	0	0	0	0	0
0	0	-1	0	0	0	0	0	0	1	0	0	0	0
0	0	2	0	0	0	0	0	0	0	1	-1	0	100
1	-1	-1	0	0	0	0	0	0	0	0	1	0	150
0	0	-2.1	0	0	0	0	0	0	0	0	1	1	400
0	1	-5	0	0	0	0	0	0	0	0	4	0	600

TABLE 2

second tabular

x_1	x_2	x_3	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	b
0	1	1.	1	0	0	0	0	0	0	0	-1.0000	0	50.0000
0	1	0	0	1.	0	0	0	0	0	0	0	0	300.0000
0	0	0	- 2	0	1.	0	0	0	0	0	1.0000	0	150.0000
0	-2	0	-3	0	0	1	0	0	0	0	3.0000	0	450.0000
0	0	0	1	0	0	0	1	0	0	0	0	0	200.0000
0	-1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	-1.0000	0	50.0000
0	-2	0	-2	0	0	0	0	0	0	1	1.0000	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	200.0000
0	2.1	0	2.1	0	0	0	0	0	0	0	-1.1	1.0000	505.0000
0	6	0	5.	0	0	0	0	0	0	0	-1	0	850.0000

TABLE 3

Third tabular

x_1	x_2	x_3	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	b
0	-1	1	-1	0	0	0	0	0	0	1	0	0	50.0000
0	1	0	0	1	0	0	0	0	0	0	0	0	300.0000
0	2	0	0	0	1	0	0	0	0	-1	0	0	150.0000
0	4	0	3	0	0	1	0	0	0	-3	0	0	450.0000
0	0	0	1	0	0	0	1	0	0	0	0	0	200.0000
0	-1	0	0	0	0	0	0	1	0	0	0	0	0
0	-1	0	-1	0	0	0	0	0	1	1	0	0	50.0000
0	-2	0	-2	0	0	0	0	0	0	1	1	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	200.0000
0	-0.1	0	-0.1	0	0	0	0	0	0	1.1	0	1.0000	505.0000
0	4	0	3	0	0	0	0	0	0	1	0	0	850.0000

TABLE 4

2. Shadow Simplex Method

As we have discussed, I choose the pair ($x_0 = (0, 0, 200), f = (-1, -1, 1)$). x_0 is our start point and fx is optimized by x_0. Pick the shadow plane as $span(f, c)$. The projection of the the polytope on $span(f, c)$ is visualized as follows. I label the vertices of the polygon.

By the previous facts, $shadow(x_0)$, and both $shadow(x_{max})$ and $shadow(x_{min})$ of cx

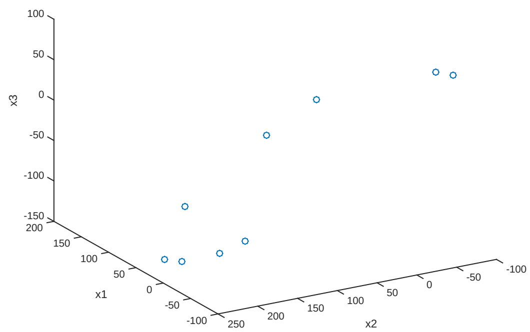

FIGURE 2. vertices of polygon on $span(c, f)$

are vertices of the polygon. Hence we reduce the 3-D problem to 2-D problem. We can use the tabular method again or walk clock wise until we find the local min and max.

Optimal solutions in shadow plane

	shadow x value
maxcx	$x_{min} = (-62.5931, 221.7586, -95.3800)$
mincx	$x_{max} = (159.0164, -51.2295, -42.2131)$

TABLE 5. The optimal solution.

We can easily recover the pre image of the above optimal points, which are exactly what we have in the tabular method. The blue line represents the max point and green line represents the min point.

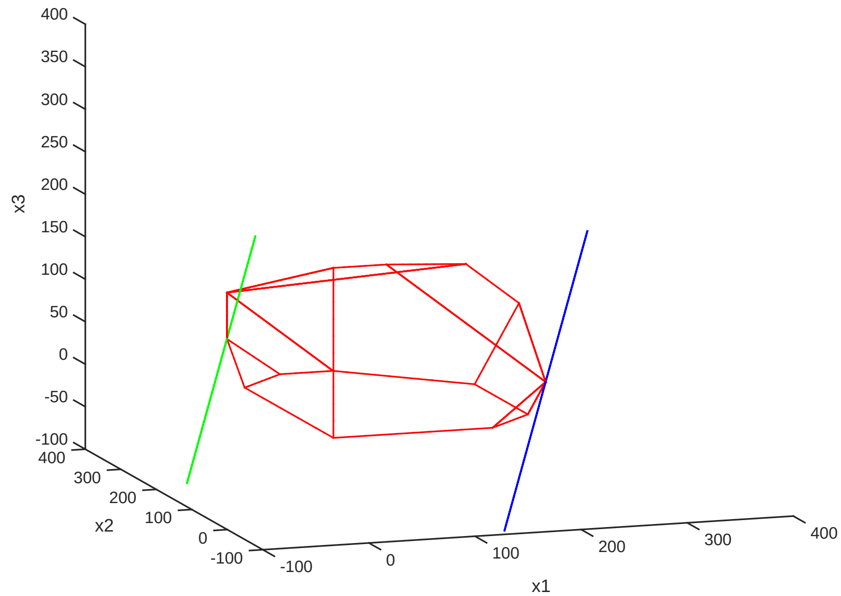

FIGURE 3. vertices of polygon on $span(c, f)$

5. **reference**

1. A Randomized Polynomial-Time Simplex Algorithm for Linear Programming. By Jonathan A. Kelner; Daniel A. Spielman.

2. https://www.cs.berkeley.edu/ vazirani/algorithms/chap7.pdf

3. Theory of Linear and Integer Programming, By Alexander Schrijver, chaper 11.