Notes on: A Randomized Polynomial-Time Simplex Algorithm for Linear Programming, by J. Kelner and D. Spielman

Dimitrios Kanoulas

December 9, 2008

1 Introduction

The authors present a randomized simplex algorithm for solving LP Problems, which has polynomial (into number of bits of the input representation) running time, with high probability.
The algorithm they propose is the first randomized algorithm of Simplex method, that solves in polynomial time a LP problem.

Previous Results

- Dantizg $\longrightarrow$ Simplex method for LPs, which takes exponential time in worst case.
- {Ellipsod Method, Various interior-point method} $\longrightarrow$ Polynomial time methods, but they are geometric algorithms, which are completelly different from Simplex method [Different from trying to find the solution, by walking along the vertices and edges defined by constraints]

Definitions and Geometric Structures

- We define as an LP the constrained optimization problem of the form:

$$\begin{array}{rc} \text{maximize} & \text{cx} \\ \text{subjet to:} & Ax \leq b \\ & \\ \text{Where} & x \in \mathbb{R}^d \\ & c \in \mathbb{R}^d \\ & b \in \mathbb{R}^n \\ & A \in \mathbb{R}^{nxd} \end{array}$$

  Note: c is the objective function, A a matrix and b, c the column vectors.
  Simplex method finds a solution to such a problem.

- The set of feasible points of our LP is a polyhedron P := $\{x|Ax \leq b\}$
  If P is non-empty then we have a convex polyhedron, where vertices are defined by d constraints, there where they are tight: $a_i x = b_i$ and a_i are the rows of matrix A.

Recall that:

- A polyhedron is a set $S \subset \mathbb{R}^n$: $S = \{x \in \mathbb{R}^n | Ax \leq b\}$
- A polytope is a bounded polyhedron.

2 Algorithm Overview

The Randomized polynomial-Time Simplex Algorithm that finds a solution to LP is as follows:

1. Reduce the starting LP to a problem such that it is needed only to certify the boundedness.
   Our new problem is a problem of determining whether the set of linear constraints defines an unbounded polyhedron or not.
   Note: Boundedness does not depend on the RHS b-vector of the constraints.

2. Perturb randomly the RHS b-vector of the constraints and run the Shadow-Vertex Simplex method on perturbed polytope to generate certificate of boundedness for the new problem.
   To certify boundedness means to find a polynomial-step path from a starting vertex to a pair of vertices that minimizing and maximizing an objetive function

3. If the algorithm fails, alter the distribution on perturbations and run the Shadow-Vertex-Simplex Method again.

Finally: The number of iterations of this loop is polynomial on the bit-length of the input, with high probability (w.h.p.)

Lets analyze now one-by-one the steps of the Algorithm.

3 Reduction: LP to Certificate Boundedness

Intuition

To use the Shadow-vertex Simplex method efficiently we need to perturb randomly the b-vector of the constraints. To do that without affecting the solutions of our starting LP we need to do that reduction.

Reduction

Starting LP:

$$\begin{array}{rc} \text{maximize} & \text{cx} \\ \text{subjet to:} & Ax \leq b \\ & x \in \mathbb{R}^d \end{array} \quad (1)$$

From the primal LP we can easily find its dual LP.
Dual LP:

$$\begin{array}{rc} \text{minimize} & \text{by} \\ \text{subjet to:} & A^T y = c \\ & y \geq 0 \end{array} \quad (4)$$

Note: Both of them are feasible and bounded. Thus from the strong duality theorem they have the same optimal solutions.

If we want the soltuions of both primal and dual LPs, they are provided by the constraints:

$$\begin{array}{c} Ax \leq b \\ x \in \mathbb{R}^d \\ A^T y = c \\ y \geq 0 \\ cx = by \end{array} \quad (5)$$

Then we reduce that to a new form which has the same feasible solutions:

$$\begin{array}{c} A_1^T z = 0 \\ z > 0 \end{array} \quad (6)$$

where A_1 consists A, b and c.

If (6) is non-degenerate [If $z \in \mathbb{R}^n$ is a basic solution then no more than n constraints are active at z] then the solution to (6) is equivalent to a certificate of the boundedness of the system:

$$A_1 w \leq b_1 \quad (7)$$

where b_1 choice doesn't matter, because it doesn't affect the boundedness.[3]

Note: It's been proved that we can produce such a non-degenerate system w.h.p.
Thus with random selection of b_1 in (7), we can solve (1).
So our new problem is defined in equation (7).

Algorithm

Instead of certify the boundedness of problem (7), we can certify boundedness of (1) by finding the vertices minimizing and maximizing the objective function cx.
Then provided that the system is non-degenerate, which it is w.h.p. under our choice of RHS, this can be converted into a solution to (7).

4 Shadow-Vertex-Simplex Method

Notation

P → convex polyhedron
S → 2-dimensional subspace
Shadow of P onto S → projection of P onto S.
Shadow of P onto S is a polygon. Each vertex/edge of this polygon is the image of some ertex/edge of P.

Idea

Find such a 2-dimensions plane S, where the set of vertices of P that projects onto the boundary of the shadow polygon, are exactly the vertices of P that optimizes the objective function in S.

Possible Problem

It is possible the number of edges of the shadow to be exponentially large. Thus we have to guarantee that this will not happen with high probability.

Shadow-Vertex-Simplex Method

1. Input: a starting vertex u_0 of P.
2. Choose some random objective function f in S which is been optimized at u_0.
3. Set S=span(c,f). S is the 2-dimentional subspace which has c and f as bases.
   [In this way we choose a plane in which our idea above becomes true]
4. Find the shadow of P onto S.
5. If no degeneracies occur, then each vertex y on P that projects onto the boundary of the shadow has a unique neighbor on P that projects onto the next vertex of the shadow in clockwise order.
6. Looking all the vertices on P that map to the boundary of the shadow we can move from vertex that optimize f to the pair of vertices that optimize c and -c respectively, which means that we have a certificate to the boundedness problem we mentioned in previous section.

7. We allow the method to run for constant number of steps.
   If the method finds such a pair of vertices it returns this pair.
   Else it returns fail and it runs again with different petrurbation of b-vector.

In this way we start from u_0 that optimizes f and we can go to a pair of vertices u_1, u_2 that optimize c and -c.
The time complexity is bounded by the number of edges of the shadow polygon.
So it is needed a guarantee (by perturbing the b-vector) that the number of edges will be polynomially large with high probability.

5 Shadow Size in k-Near-Isotropic Case

We will prove that by perturbing randomly the b-vector of constraints of our LP, we can guarantee that the number of edges are not exponentially large w.h.p.

Notation

1. A polytope P is in k-near-isotropic position if: B(0,1) $\subseteq$ P $\subseteq$ B(0,k), where B(0,r) is the ball of radius r, centered at the origin.

2. We define the polytope P:= $\{x|\forall i, a_i^T x \leq 1\}$

3. We define the perturbed-polytope Q:= $\{x|\forall i, a_i^T x \leq 1+r_i\}$, where r_i is independent exponentially distributed random variable with expectation λ
   [Pr[$r_i \geq$ t] = $e^{-t/l}$, t $\geq$ 0]

Intuition

There are two cases:

1. Polytope is in k-near-isotropic position

2. Polytope is in non k-near-isotropic position

I will give the idea for the first case and mention the very general idea for the second one.
The intuition is that if the polytope is in k-near-isotropic position AND the distance of the facets (edges/vertices) from the origin are randomly perturbed [Perturbation of b-vector of the constraints], then the expected number of edges of the projection Q onto a random 2-dimentions plane S is expected to be polynomial in bit-length of the input w.h.p.
That means that the shortest path from a starting vertex to a pair of vertices that maximizes and minimizes the objective function cx, is expected to have a number of steps polynomial in bit-length of the input w.h.p.

Proof for k-near-isotropic position

To prove this we will prove that:

1. The total length of all edges on the boundary of projection is expected to be upper bounded

2. Our perturbation will cause the expected length of each edge to be lower bounded.

By these two we can conclude straitforward that we have a bound on the expected number of edges that appears on the boundary of the projection.

Assume the 2-dimensions base V=span(u,w), where u, w are uniformly random unit vectors.

Proposition 5.1 *The perimeter of the shadow of P onto V is at most: $2\pi k$ (because of the bound that B(0,k) gives)*

Proposition 5.2 *The perimeter of the shadow of Q onto V is at most:* $2\pi k(1+r)$ *(because* $Q \subseteq (1+r)P$*, where* $r = \max_i r_i$*)*

Proposition 5.3 *The expected value of r is:* $E[r] \leq \lambda\ ln(ne)$*, where* λ *is the expectation of r and n the number of constraints.*

Lemma 5.4 *Upper Bound: The expected perimeter of the shadow of Q onto V is at most:* $2\pi k(1 + \lambda(ne))$

Lemma 5.5 *Lower Bound: The expected edge length appears on the shadow is at least:* $\frac{\lambda}{6\sqrt{d}n}$

Theorem 5.6 *The expected number of edges are:* $E[\#edges] \leq \frac{12\pi k(1+\lambda(ne))\sqrt{d}n}{\lambda}$*, which is the bound of the number of the edges.*
(Comes from the two Lemmas)

By the Theorem we can see that the expected number of edges of the shadow of the perturbed polytope Q is bouned by an expression that is a polynomial on number of bits of input representation.

Proof Intuition for Non-k-near-isotropic position

Note: For each polytope there exists a change of coordinates such that puts the polytope into a k-near-isotropic position in a way that does not change the structure of the polytope.
If the polytope is not in a k-near-isotropic position, then it's been proved in the paper that it is in an ellipse. If the Shadow-vertex Simplex method fails in a limited amount of time, we can transform the coordinates such that the ellipse becomes the unit ball, without changing the solutions of our problem. Since the ellipse has at least twice the volume of the unit ball, each time a bad event happens, the volume of the polytope is reduced at least by half. In this way the polytope comes closer to a k-near-isotropic position in each loop.
The Theorem below comes from this procedure and puts a bound to the number of edges of the shadow of the polytope.

Theorem 5.7 *Let* $\|a_i\| \leq 1$*,* r_i*s to be exponentially distributed random variables with expectation* λ *and* $Q := \{x | \forall i, a_i^T x \leq 1 + r_i\}$*.*
If $p \leq \frac{1}{\sqrt{d}}$ *and w are uniformly random unit vectors and* $t > 1$*, then:* $E[ShadowSize_{span(v,w)}(Q \cap B(0,t))] \leq \frac{42\pi t(1+\lambda \log n)\sqrt{d}n}{\lambda p}$

6 Summary

To sum up, the outline of the proposed algorithm is,

1. Reduce the starting LP to a problem of certifying boundedness
2. Choose the correct 2-dimensions plane S such that our solution is on the boundary of the shadow of the polytope.s
3. Perturb randomly b vector, to guarantee that the number of edges of the boundary of the shadow is polynomially large.
4. Move from a starting vertex of the boundary of the shadow to this that optimizes c or -c objective function in polynomial time.
5. If the algorithm fails, alter the distribution on perturbations and run the algorithm again.

References

[1] J. Kelner and D. Spielman *A Randomized Polynomial-Time Simplex Algorithm for Linear Programming* STOC 2006

[2] D. Bertsimas and J. Tsitsiklis *Introduction to Linear Optimization*

[3] N. Megiddo and R. Chandrasekaran *On the epsilon-perturbation method for avoiding degeneracy* Operations Research Letters, 1989.

[4] K. Mehlhorn *The Running Time of the Simplex Method* Notes 2001

[5] P. Parrilo and S. Lall *Linear Inequalities and Elimination* Notes 2003