

## LECTURE 11:

### On average running time of the simplex method

---

Theory of linear and integer programming

Mar. 2013

#### Crossreference

Simplex method

Polyhedral theory

### 11.1 Fast recap: the simplex method

What is simplex method?

phase I : Find a bfs.

phase II : Start with a bfs and work towards a optimal solution.

Geometrical interpretation of phase II: Start with a initial vertex of the polyhedron and move in each iteration to a adjacent vertex until eventually either an optimal vertex is reached or the algorithm recognizes at some vertex that the problem is unbounded.

Since Phase I works in a similar manner to Phase II, and Phase II admits a better geometrical explanation, we concentrate on Phase II.

### 11.2 Shadow-vertex pivot rule

The simplex method relies heavily on the pivot rule. A different pivot rule leads to a 'variant' of the simplex method. There are many different rules such as... In particular, the shadow-vertex pivot rule introduced by Gass and Satty works extremely well in probabilistic analysis and allows an intuitive geometric description. We not explain it in detail.

Let  $x_0$  denote the given initial vertex of polyhedron  $P$ . Since  $x_0$  is a vertex of the polyhedron, there exists an objective function  $\bar{c}^\top x$  which is maximized by  $x_0$  subject to the constraint  $x \in P$ . If  $x_0$  is not the optimal solution of the linear program, then  $c$  and  $\bar{c}$  are linearly independent (note that the converse this statement is not true) and thus span a plane. Project the polyhedron  $P$  onto this plane  $\text{span}(c, \bar{c})$ .

The Shadow, that is the projection of  $P$  on to his plane has a few useful properties:

1. The vertex  $x_0$  is projected onto a vertex of the projection.
2. The optimal solution  $x^*$  is projected onto a vertex of the projection.
3. Each vertex of the projection is the image of the a vertex of the polyhedron.
4. Each edge of the projection is the image of an edge between two adjacent vertices of the polyhedron.

Observe that the simplex algorithm in dimension two is very easy: it just follows the edges of the polyhedron. Apply the simplex method to the projection and the walk along the edges of the projection corresponds to a walk along the edges of the original polyhedron. Furthermore, once the optimal solution

on the polyhedron is found, we can compute the pre-image on the polyhedron, which is an optimal solution of the linear program.

(The number of steps performed by the simplex algorithm with shadow vertex pivot rule is upper bounded by the number of vertices of the two-dimensional projection of the polyhedron. Hence, bounding the expected number of vertices on the polygon is the crucial step for bounding the expected running time of the simplex algorithm with shadow vertex pivot rule. Spielman and Teng consider first the case that the polyhedron  $P$  is projected onto a fixed plane specified by two fixed vectors  $c$  and  $\bar{c}$ . They show the following result on the expected shadow size, that is, the number of vertices of the projection.)

Shadow-vertex pivot rule mathematical description(cf. Theory of linear and integer programming): Given a vertex  $x_k$  of the polyhedron  $P$  maximizing  $(\lambda c + \bar{c})x$  over  $P$  for some  $\lambda$

Geometrically, shadow-vertex pivot rule means the following: Project  $P$  on to the plane spanned by  $c$  and  $\bar{c}$ . We obtain a polyhedron  $Q$  in two dimensions. The vertices of  $Q$  are exactly the projection of those vertices  $x$  of  $P$  which maximize  $(\lambda c + \mu \bar{c})x$  for some  $\lambda$  and  $\mu$ . The step from  $x_k$  to  $x_{k+1}$  in the step (48) means in  $Q$  that we go from the projection of  $x_k$  to the adjacent vertex in the direction of  $c$ . This explains the name 'shadow vertex pivot rule'.

graphs explanation.

### 11.3 Efficiency of algorithms: Gaps between theory and practice

In theoretical computer science, an algorithm is typically judged by its worst-case performance. An algorithm with good worst-case performance is very desirable because it performs well on all possible inputs. On the other hand, a bad worst-case performance does not necessarily imply that the algorithm performs also badly in practice. The most prominent example is probably the simplex algorithm for solving linear programs. For most deterministic pivot rules that have been suggested, examples are known showing that in the worst case the simplex algorithm can take an exponential number of steps, but the simplex algorithm is still one of the most competitive algorithms for solving linear programs in practice. It is fast and reliable even for large-scale instances and for the pivot rules that have been shown to require an exponential number of iterations in the worst case. Examples on which the simplex algorithm needs many iterations are quite artificial and occur only very rarely in practice. This behavior is by no means an exceptional property of the simplex algorithm.

This might motivate to study the average-case performance rather than the worst case performance. But average-case analyses are often problematic because it is not clear how to choose a reasonable probability distribution on the set of inputs. Many average-case analyses assume a uniform distribution. However, for most problems, instances chosen uniformly at random do not reflect typical instances. For example, linear programs that are obtained by choosing each coefficient in the constraint matrix uniformly at random have very special properties with high probability and are very different from linear programs that occur in practical applications.

Engineers:

Theorists:

Reason for discrepancy:

How to make theory more consistent with practice? More compatible with practice? Find a more realistic performance measure.

How to explain the huge gap between the provably bad worst case bounds and the efficiency of the natural in practice? A natural approach is to apply probabilistic analysis: assume that the input data are distributed according to some prespecified distribution(a realistic distribution will be the best, however we don't know what the distribution of real world looks like.) and show on the average the number of pivots is small.

## 11.4 A probabilistic model

Sign-invariant model(SIM)

It's extremely simple and only relies on a finite number of reflections and symmetries. In analysis it's more sufficient to consider a somehow relaxed version of sign-in-variance, the so-called flipping model.  $\max\{\pm c^T x | a_1^T \leq \beta_1; \dots; a_m^T \leq \beta_m\}$ . Here  $\leq$  indicates that one of the relations  $\leq$  or  $\geq$  is valid in the formulation of the instance. And we independently determine the  $m$  directions of the relations, each one with probability  $1/2$  for  $\geq$  and with probability  $1/2$  for  $\leq$ . Then we generate exactly  $2^{m+1}$  problem instances out of the data set  $A, b, c, \bar{c}$ . The idea of averaging is to solve all  $2^{m+1}$  instances, to sum up the required pivot steps and to divide by the number of instances.

Here we consider/ we restrict ourselves to a simplified model which captures all the difficulty /complexity in polyhedral aspects. provides insight into the average running study.

## 11.5 average running time

**Theorem 11.1** *The class of LP-program given as  $\max\{c^T x | Ax \leq b\}$ , can be solved with the simplex method, using the shadow vertex pivot rule, in at most  $\frac{1}{2}n$  pivot iterations on the average, under any probability measure satisfying (47).*

*Proof.* Let the LP-program  $\max\{c^T x | Ax \leq b\}$  be given, together with a vertex (extreme point/basic feasible solution)  $x_0$  of  $\{x | Ax \leq b\}$  and a vector  $\bar{c}$  such that  $x_0$  maximizes  $\bar{c}^T x$  over  $Ax \leq b$ . Let  $m$  and  $n$  be the numbers of rows and columns of  $A$ .

The idea of the proof is to calculate the average pivot steps over some collections of LP.

Let  $z_1, \dots, z_t$  be those vectors in  $\mathbb{R}^n$  which can be determined by some choice of  $n$  linearly independent equations from  $Ax = b$ . So  $t = \binom{m}{n}$  with probability 1. Consider the class  $\mathcal{L}$  of LP-problems  $\max\{\bar{c}^T x | \hat{A}x \leq \hat{b}\}$  which arise from  $\max\{c^T x | Ax \leq b\}$  by reversing some (or none) of the inequality signs and/or replacing  $c$  by  $-c$ , and for which

$$\max\{\bar{c}^T x | \hat{A}x \leq \hat{b}\} \quad (11.1)$$

is finite. We claim  $|\mathcal{L}| = 2\binom{m}{n}$ . Indeed, by (47) (the probabilistic measure condition), with probability 1, each  $\max\{\bar{c}^T x | \hat{A}x \leq \hat{b}\}$  is uniquely<sup>1</sup> attained<sup>2</sup> by a vertex of  $\hat{A}x \leq \hat{b}$ , and hence by one of the vectors  $z_i$ . Moreover, for each  $z_i$  there is exactly *one* choice of  $\hat{A}x \leq \hat{b}$  for which  $\max\{\bar{c}^T x | \hat{A}x \leq \hat{b}\}$  is attained by  $z_i$ . [Since, with probability 1, there are exactly  $n$  inequalities in  $\hat{A}x \leq \hat{b}$ , say  $\hat{A}'x \leq \hat{b}'$ , which are satisfied by  $z_i$  with equality. Hence the inequality sign is strict in the other  $m - n$  inequalities, and therefore we can not reverse the sign. Moreover,  $\bar{c}$  must be in the cone generated by the rows of  $\hat{A}'$ , and hence also here the signs<sup>5</sup> are uniquely determined.] So to  $z_i$  there correspond the programs  $\max\{cx | \hat{A}x \leq \hat{b}\}$  and  $\max\{-cx | \hat{A}x \leq \hat{b}\}$ <sup>6</sup>. Therefore,  $|\mathcal{L}| = 2\binom{m}{n}$ .

Next consider the class of edges and rays formed by the systems  $\hat{A}x \leq \hat{b}$ . Each set of  $n - 1$  equations from  $Ax = b$  gives a line(1-dimensional affine subspace) in  $\mathbb{R}^n$ , containing  $m - n + 1$  of  $z_i$ . The  $z_i$  subdivide this line into  $m - n$  edges and two rays.

Let  $l$  be such a line, and let  $z_i$  and  $z_j$  be two neighboring points on  $l$ (i.e. the line segment  $\overline{z_i z_j}$  contains no other points from  $z_1, \dots, z_t$ ). Then the edge  $\overline{z_i z_j}$  is traversed in at most (in fact, exactly) one of the programs in  $\mathcal{L}$ . For suppose  $\overline{z_i z_j}$  is traversed in more than one of these programs  $\max\{\bar{c}^T x | \hat{A}x \leq \hat{b}\}$ . Then starting

<sup>1</sup>By distribution assumption, any  $n$  or less vectors of  $c$  and row vectors in  $A$  are linearly independent. Even the optimal is attained in a line, it will lead to dependence in these vectors. Not to mention a face.

<sup>2</sup>Finiteness assumption in (11.1)

<sup>3</sup>This observation comes from the definition of class  $\mathcal{L}$ , which requires that for all instances in  $\mathcal{L}$ ,  $\max\{\bar{c}^T x | \hat{A}x \leq \hat{b}\}$  is finite.

<sup>4</sup>By complementary slackness in  $\max\{c^T x | Ax \leq b\} = \min\{b^T y | y^T A = c, y \geq 0\}$ ,  $\bar{c}$  is the non-negative combination of all tight(active) constraints).

<sup>5</sup>Since  $\bar{c}$  is fixed, and is a non-negative combination of all directions of tight constraints, so all this tight constraints can not be flipped (reversed) otherwise  $\bar{c}$  will no longer be in the cones generated by all the directions of these constraints

<sup>6</sup>The only 2 relations that are revertible/flip-able in  $\mathcal{L}$

with an arbitrary feasible program in  $\mathcal{L}$ , we can always obtain a new feasible program in  $\mathcal{L}$  by reverting or flipping operation. However, it turns out that for any feasible program from  $\mathcal{L}$ , neither the constraints nor the objective function is revertible.

By (48), there is a  $\lambda$  such that

$$\overline{z_i z_j} = \{x | \hat{A}x \leq \hat{b}\} \cap \{x | (\bar{c} + \lambda \hat{c})x = (\bar{c} + \lambda \hat{c})z_i\} \quad (11.2)$$

and such that  $(\bar{c} + \lambda \hat{c})x \leq (\bar{c} + \lambda \hat{c})z_i$  is valid for  $\{x | \hat{A}x \leq \hat{b}\}$ . By (47) we know that with probability 1 there are exactly  $n - 1$  of the inequalities  $\hat{A}x \leq \hat{b}$ , which are tight on  $\overline{z_i z_j}$ , say  $\hat{A}'x \leq \hat{b}'$ . Hence the inequality signs in the other inequalities cannot be reversed. Moreover, there is only one vector  $u$  and only one scalar  $\lambda$  with  $u^\top \hat{A}' = \bar{c} + \lambda \hat{c}$  (with probability 1). (This result is based on 2 assumption of the model in (47). First, with probability 0,  $n$  or less vectors from  $c$ ,  $\bar{c}$  and rows vectors in  $A$  are linearly dependent. Second, note that  $A$  is a  $m$  by  $n$  matrix, implying that  $x \in \mathbb{R}^n$ . So there are at most  $n$  linearly independent vectors. Since  $n - 1$  vectors in  $\hat{A}'$  and together with  $c$  and  $\bar{c}$  we have  $n + 1$  vectors in all, they must be linearly dependent with probability 1.) Since  $(\bar{c} + \lambda \hat{c})x = (\bar{c} + \lambda \hat{c})z_i$  is a supporting half-space, we know that  $u > 0$ . (First of all, it's easy to get that no element in  $u$  is zero, otherwise remaining vectors are linearly dependent contradicting with the assumption (47). Second,  $\bar{c} + \lambda \hat{c}$  must be in the cone generated by the rows of  $\hat{A}'$ , otherwise,  $(\bar{c} + \lambda \hat{c})x = (\bar{c} + \lambda \hat{c})z_i$  cannot be a supporting half space.) The uniqueness of  $u$  implies the uniqueness of the inequality signs in  $\hat{A}' \leq \hat{b}'$ . (Since fixed  $\bar{c} + \lambda \hat{c}$  is a positive linear combination of rows in  $\hat{A}'$ , if some inequality in  $\hat{A}' \leq \hat{b}'$  is revert-able, we will get another  $u$  and  $\lambda$  satisfying equality  $u^\top \hat{A}' = \bar{c} + \lambda \hat{c}$ , finally yielding that some  $n$  vectors are linearly dependent.) Finally, let without loss of generality  $\bar{c}z_i < \bar{c}z_j$ . If  $\hat{c}z_i > \hat{c}z_j$ , then (49) implies  $\hat{c} = c$ . If  $\hat{c}z_i < \hat{c}z_j$ , (49) implies  $\hat{c} = -c$ . So there is also only one choice for  $\hat{c}$ . Concluding, the edge  $\overline{z_i z_j}$  is traversed in only one LP in  $\mathcal{L}$ .

Similarly, each ray  $z_i + y\mathbb{R}^+$  is traversed only if  $\bar{c}y > 0$  (by (51)), and then in only one LP in  $\mathcal{L}$ .

Now on each line determined by  $n - 1$  equations from  $Ax = b$  there are  $m - n$  edges and one ray  $z_i + y\mathbb{R}^+$  with  $\bar{c}y > 0$ . So the total number of such edges and such rays is  $(m - n + 1)\binom{m}{n-1} = \binom{m}{n}n$ . Therefore, the average number of pivot steps is at most:

$$\frac{\binom{m}{n}n}{|\mathcal{L}|} = \frac{1}{2}n. \quad (11.3)$$

□

## References

- [CW87] D. COPPERSMITH and S. WINOGRAD, "Matrix multiplication via arithmetic progressions," *Proceedings of the 19th ACM Symposium on Theory of Computing*, 1987, pp. 1–6.