

# 15. A randomized Algorithm for Linear Programming

Lecture on Monday 16<sup>th</sup> November, 2009 by Bernd Gärtner <gaertner@inf.ethz.ch>

Let us recall the setup from last lecture: we have a linear program of the form

$$\begin{aligned} \text{(LP)} \quad & \text{maximize} \quad c^T x \\ & \text{subject to} \quad Ax \leq b, \end{aligned} \tag{15.1}$$

where  $c, x \in \mathbb{R}^d$  (there are  $d$  variables),  $b \in \mathbb{R}^n$  (there are  $n$  constraints), and  $A \in \mathbb{R}^{n \times d}$ . The scenario that we are interested in here is that  $d$  is a (small) constant, while  $n$  tends to infinity.

The goal of this and the next lecture is to present a randomized algorithm (due to Seidel[2]) for solving a linear program whose expected runtime is  $O(n)$ . The constant behind the big-Oh will depend exponentially on  $d$ , meaning that this algorithm is practical only for small values of  $d$ .

To prepare the ground, let us first get rid of the unboundedness issue. We add to our linear program a set of  $2d$  constraints

$$-M \leq x_i \leq M, \quad i = 1, 2, \dots, d, \tag{15.2}$$

where  $M$  is a symbolic constant assumed to be larger than any real number that it is compared with. Formally, this can be done by computing with rational functions in  $M$  (quotients of polynomials of degree  $d$  in the “variable “ $M$ ), rather than real numbers. The original problem is bounded if and only if the solution of the new (and bounded) problem does not depend on  $M$ . This is called the *big- $M$  method*.

Now let  $H$ ,  $|H| = n$ , denote the set of original constraints. For  $h \in H$ , we write the corresponding constraint as  $a_h x \leq b_h$ .

**Definition 15.3** For  $Q, R \subseteq H$ ,  $Q \cap R = \emptyset$ , let  $x^*(Q, R)$  denote the lexicographically largest optimal solution of the linear program

$$\begin{aligned} LP(Q, R) \quad & \text{maximize} \quad c^T x \\ & \text{subject to} \quad a_h x \leq b_h, \quad h \in Q \\ & \quad \quad \quad a_h x = b_h, \quad h \in R \\ & \quad \quad \quad -M \leq x_i \leq M, \quad i = 1, 2, \dots, d. \end{aligned}$$

If this linear program has no feasible solution, we set  $x^*(Q, R) = \infty$ .

What does this mean? We delete some of the original constraints (the ones not in  $Q \cup R$ , and we require some of the constraints to hold with equality (the ones in  $R$ ). Since every linear equation  $a_h x = b_h$  can be simulated by two linear inequalities  $a_h x \leq b_h$  and  $a_h x \geq b_h$ , this again assumes the form of a linear program. By the big- $M$  method, this linear program is bounded, but it may be infeasible. If it is feasible, there may be several optimal solutions, but choosing the lexicographically largest one leads to a unique solution  $x^*(Q, R)$ .

Our algorithm will compute  $x^*(H, \emptyset)$ , the lexicographically largest optimal solution of (15.1) subject to the additional bounding-box constraint (15.2). We also assume that  $x^*(H, \emptyset) \neq \infty$ , meaning that (15.1) is feasible. At the expense of solving an auxiliary problem with one extra variable, this may be assumed w.l.o.g. (Exercise).

## 15.1 Helly's Theorem

A crucial ingredient of the algorithm's analysis is that the optimal solution  $x^*(H, \emptyset)$  is already determined by a constant number (at most  $d$ ) of constraints. More generally, the following holds.

**Lemma 15.4** *Let  $Q, R \subseteq H, Q \cap R = \emptyset$ , such that the constraints in  $R$  are independent. This means that the set  $\{x \in \mathbb{R}^d : a_h x = b_h, h \in R\}$  has dimension  $d - |R|$ .*

*If  $x^*(Q, R) \neq \infty$ , then there exists  $S \subseteq Q, |S| \leq d - |R|$  such that*

$$x^*(S, R) = x^*(Q, R).$$

The proof uses *Helly's Theorem*, a classic result in convexity theory.

**Theorem 15.5 (Helly's Theorem[1])** *Let  $C_1, \dots, C_n$  be  $n \geq d + 1$  convex subsets of  $\mathbb{R}^d$ . If any  $d + 1$  of the sets have a nonempty common intersection, then the common intersection of all  $n$  sets is nonempty.*

Even in  $\mathbb{R}^1$ , this is not entirely obvious. There it says that for every set of intervals with pairwise nonempty overlap there is one point contained in all the intervals. We will not prove Helly's Theorem here but just use it to prove Lemma 15.4.

**Proof.** (**Lemma 15.4**) The statement is trivial for  $|Q| \leq d - |R|$ , so assume  $|Q| > d - |R|$ . Let

$$L(R) := \{x \in \mathbb{R}^d : a_h x = b_h, h \in R\}$$

and

$$B := \{x \in \mathbb{R}^d : -M \leq x_i \leq M, i = 1, \dots, d\}.$$

For a vector  $x = (x_1, \dots, x_d)$ , we define  $x_0 = c^T x$ , and we write  $x > x'$  if  $(x_0, x_1, \dots, x_d)$  is lexicographically larger than  $(x'_0, x'_1, \dots, x'_d)$ .

Let  $x^* = x^*(Q, R)$  and consider the  $|Q| + 1$  sets

$$C_h = \{x \in \mathbb{R}^d : a_h x \leq b_h\} \cap B \cap L(R), \quad h \in Q$$

and

$$C_0 = \{x \in \mathbb{R}^d : x > x^*\} \cap L(R).$$

The first observation (that may require a little thinking in case of  $C_0$ ) is that all these sets are convex. The second observation is that their common intersection is empty. Indeed, any point in the common intersection would be a feasible solution  $\tilde{x}$  of  $LP(Q, R)$  with  $\tilde{x} > x^* = x^*(Q, R)$ , a contradiction to  $x^*(Q, R)$  being the lexicographically largest optimal solution of  $LP(Q, R)$ . The third observation is that since  $L(R)$  has dimension  $d - |R|$ , we can after an affine transformation assume that all our  $|Q| + 1$  convex sets are actually convex subsets of  $\mathbb{R}^{d-|R|}$ .

Then, applying Helly's Theorem yields a subset of  $d - |R| + 1$  constraints with an empty common intersection. Since all the  $C_h$  do have  $x^*(Q, R)$  in common, this set of constraints must contain  $C_0$ . This means, there is  $S \subseteq Q, |S| = d - |R|$  such that

$$x \in C_h, h \in S \Rightarrow x \notin C_0.$$

In particular,  $x^*(S, R) \in C_h$  for all  $h \in S$ , and so it follows that  $x^*(S, R) \leq x^* = x^*(Q, R)$ . But since  $S \subseteq Q$ , we also have  $x^*(S, R) \geq x^*(Q, R)$ , and  $x^*(S, R) = x^*(Q, R)$  follows.  $\square$

## 15.2 Convexity, once more

We need a second property of linear programs on top of Lemma 15.4; it also a consequence of convexity of the constraints, but a much simpler one.

**Lemma 15.6** *Let  $Q, R \subseteq H, Q \cap R \neq \emptyset$  and  $x^*(Q, R) \neq \infty$ . Let  $h \in Q$ . If*

$$\alpha_h x^*(Q \setminus \{h\}, R) > b_h,$$

*then*

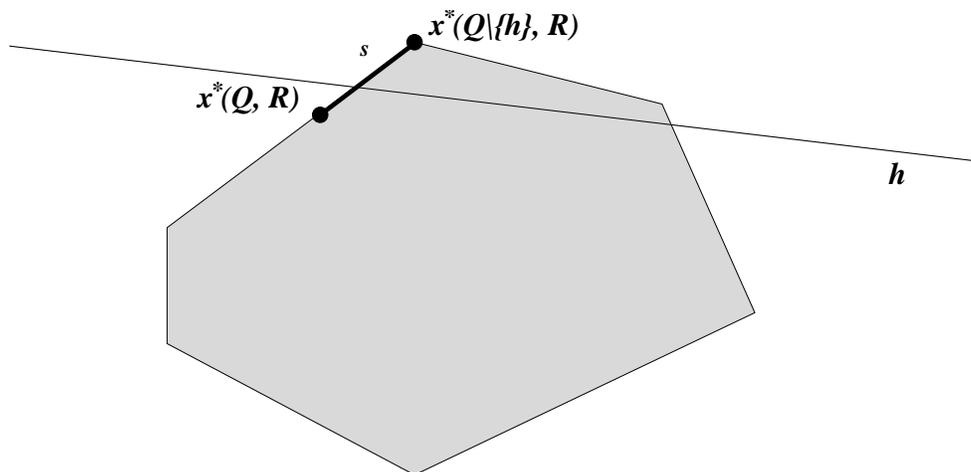
$$x^*(Q, R) = x^*(Q \setminus \{h\}, R \cup \{h\}).$$

Before we prove this, let us get an intuition. The vector  $x^*(Q \setminus \{h\}, R)$  is the optimal solution of  $LP(Q \setminus \{h\}, R)$ . And the inequality  $\alpha_h x^*(Q \setminus \{h\}, R) > b_h$  means that the constraint  $h$  is violated by this solution. The implication of the lemma is that at the optimal solution of  $LP(Q, R)$ , the constraint  $h$  must be satisfied with equality in which case this optimal solution is at the same time the optimal solution of the more restricted problem  $LP(Q \setminus \{h\}, R \cup \{h\})$ .

**Proof.** Let us suppose for a contradiction that

$$\alpha_h x^*(Q, R) < b_h$$

and consider the line segment  $s$  that connects  $x^*(Q, R)$  with  $x^*(Q \setminus \{h\}, R)$ . By the previous strict inequality, we can make a small step along this line segment without violating the constraint  $h$  (Figure 15.1). And since both  $x^*(Q, R)$  as well as  $x^*(Q \setminus \{h\}, R)$  satisfy all other constraints in  $(Q \setminus \{h\}, R)$ , convexity of the constraints implies that this small step takes us to a feasible solution of  $LP(Q, R)$  again. But this solution is lexicographically larger than  $x^*(Q, R)$ , since we move towards the lexicographically larger vector  $x^*(Q \setminus \{h\}, R)$ ; this is a contradiction.  $\square$

Figure 15.1: *Proof of Lemma 15.6*

### 15.3 The Algorithm

The algorithm reduces the computation of  $x^*(H, \emptyset)$  to the computation of  $x^*(Q, R)$  for various sets  $Q, R$ , where  $R$  is an *independent* set of constraints. Suppose you want to compute  $x^*(Q, R)$  (assuming that  $x^*(Q, R) \neq \infty$ ). If  $Q = \emptyset$ , this is “easy”, since we have a constant-size problem, defined by  $R$  with  $|R| \leq d$  and  $2d$  bounding-box constraints  $-M \leq x_i \leq M$ .

Otherwise, we choose  $h \in Q$  and recursively compute  $x^*(Q \setminus \{h\}, R) \neq \infty$ . We then check whether constraint  $h$  is violated by this solution. If not, we are done, since then  $x^*(Q \setminus \{h\}, R) = x^*(Q, R)$  (why?). But if  $h$  is violated, we can use Lemma 15.6 to conclude that  $x^*(Q, R) = x^*(Q \setminus \{h\}, R \cup \{h\})$ , and we recursively compute the latter solution. Here is the complete pseudocode.

```

 $\mathcal{LP}(Q, R)$ :
  IF  $Q = \emptyset$  THEN
    RETURN  $x^*(\emptyset, R)$ 
  ELSE
    choose  $h \in Q$  uniformly at random
     $x^* := \mathcal{LP}(Q \setminus \{h\}, R)$ 
    IF  $a_h x^* \leq b_h$  THEN
      RETURN  $x^*$ 
    ELSE
      RETURN  $\mathcal{LP}(Q \setminus \{h\}, R \cup \{h\})$ 
  END
END

```

To solve the original problem, we call this algorithm with  $\mathcal{LP}(H, \emptyset)$ . It is clear that the algorithm terminates since the first argument  $Q$  becomes smaller in every recursive call. It is also true (Exercise) that every set  $R$  that comes up during this algorithm is indeed an independent set of constraints and in particular has at most  $d$  elements. The correctness of the algorithm then follows from Lemma 15.6.

In the next lecture, we will analyze the (expected) runtime of this algorithm, using Lemma 15.4.

## Questions

44. *What is Helly's Theorem?* Give a precise statement and outline the application of the theorem for linear programming (Lemma 15.4).
45. *Outline an algorithm for solving linear programs!* Sketch the main steps of the algorithm and the correctness proof! Also explain how one may achieve the preconditions of feasibility and boundedness.

## References

- [1] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*, Springer-Verlag, Heidelberg, West Germany, 1987.
- [2] R. Seidel, Small-dimensional linear programming and convex hulls made easy, *Discrete Comput. Geom.* **6** (1991), 423–434.