

Towards Randomized Strongly Polynomial Algorithms for Linear Programming

Srinivasan Krishnan*, John. H. Reif*

June 24, 1991

Abstract

The issue of strongly polynomial algorithms for linear programming has been unresolved for some time now. This paper proposes a new approach that is based on randomization. The key idea of this paper is to polynomially bound the length of a random walk when a certain condition (Conjecture 1) holds. We prove this condition for dimensions two and three. A general proof of the lemma (still open) would automatically result in a randomized strongly polynomial algorithm for linear programming.

1 Problem Statement

We consider the following form of the LP problem. Consider a system of linear inequality constraints:

$Ax \leq b$ where A is a $n \times d$ matrix, b is a given n -vector, x is a d -vector (x_1, \dots, x_d) .

we wish to determine

$$x_1^* = \max\{x_1 \mid Ax \leq b\}$$

Given any point p of the polytope, we will denote by $f(p)$ the value of the objective function at p . We will further assume that the polytope is non-degenerate and therefore the optimum point is unique. (See [1] for a discussion on this).

2 Random Walks

Consider a non-degenerate polytope in d dimensions with n constraints. Every vertex of the polytope has n edges incident on it. Now consider those edges which lead to an improvement in the objective function. Let us call them positive edges.

Consider now a random walk along the points of the polytope p_1, p_2, \dots that monotonically improves the objective function, i.e. $f(p_i) < f(p_{i+1})$ until we reach point p_m which is the optimum.

The following key conjecture may now be stated with respect to this walk.

Conjecture 1: Given any point p on the polytope exactly one of the following hold:

- p is the optimum point on the polytope with respect to the objective function. ...[1]
- all points q on the polytope have a higher objective function value than p[2]

* Duke University, Dept. of Computer Science, Durham, NC 27705.

This work was supported by Air Force Contract AFOSR-87-0386, ONR Contract N00014-87-K-0310, DARPA/ISTO Contract N00014-88-K-0458, ARO Contract DAAL03-88-K-0195, and NASA/CESDIS Subcontract 550-63 NAS 5-30428 URSA.

- there exists an positive edge e from p to some point, say q , which satisfies either or both of the properties:

- One of the d constraints comprising q have never been visited before.
- One of the d constraints comprising p will be never visited again

...[3]

Theorem 1: Given that conjecture 1 holds, there exists a randomized strongly polynomial algorithm for linear programming.

Proof: (Constructive) Consider the following algorithm RANDOM:

Step1: Begin at point p_1 (Iteration 1).

Step2: (Iteration i , point p_i) If p_i is not optimal, choose a random positive edge e . Move from p_i to p_{i+1} via e . Otherwise return p_i .

Step 3: Go to step2.

Analysis of RANDOM: Let m be the length of the resulting walk. (We measure the length of the walk by the number of points visited in the path. The complexity of the algorithm is clearly related polynomially to the length of the walk.) Then at most 2 of the m points can satisfy cases 1 and 2 of the conjecture. Thus the remaining $m - 2$ points satisfy case 3.

Our goal is now to show that the expected length of the walk is polynomially related to n and d . The intuition behind the proof is that given any point that satisfies case 3 of the conjecture, we will choose the right edge with probability $1/n$. Hence the expected number of trials before we actually choose the right edge is n . Further we will show that we cannot satisfy case 3 of the conjecture more than a certain number of times, thus bounding the expected length of the walk.

Consider a point p on the walk that satisfies case 3 of the conjecture. Let e be the edge from p to (say) q that satisfies case 3. Then:

Lemma 2: e is chosen by algorithm RANDOM with probability at least $1/d$.

Proof: p has at most d edges incident on it. Therefore at most d positive edges emanate from p . Since algorithm RANDOM chooses at random from among the positive edges with equal probability, the probability of choosing e is at least $1/d$. \square

Lemma 3: The expected length of the walk in algorithm RANDOM between successive selections of the right edge (i.e one that satisfies conjecture 1) is d .

Proof: Successful Selections of edges are independent events that occur with equal probability $1/d$. Hence the expected time between successful selections is d . \square

Lemma 4: The number of successful selections that can occur in the random walk before reaching the optimum is no more than $2n$.

Proof: Every successful selection implies that either some constraint c_1 has been visited for the first time or some constraint c_2 has been visited for the last time. There are a total of n constraints. Hence, they can be visited for the first and last times at most 2 times each and hence successful selections can occur no more than $2n$ times. \square

Lemma 5: The expected length of the random walk is $O(dn)$. Proof: follows directly from lemmas 3 and 4.

The proof of Theorem 1 follows directly from lemma 5 and the observation that no more than d edges are investigated at each point on the walk.

Thus the algorithm takes an expected time $O(d^2n)$.

3 Proofs in 2 and 3 dimensions

Now we give proofs of the conjecture in the 2-dimensional and 3-dimensional cases.

3.1 The 2-Dimensional Case

This is fairly trivial. Consider a point that does not satisfy cases 1 and 2 of the conjecture

In 2 dimensions every edge is a constraint of the polytope and contains only 2 points. As a result while moving along any positive edge we are guaranteed never to see that edge again. (In other words if we move along positive edge e from p to q , then since e contains only points p and q , subsequent points along the walk r with $f(r) > f(q)$ cannot contain constraint e ever again.)

The situation is visualized in figure 1.

3.2 The 3 Dimensional Case

The proof here is somewhat more involved.

Since the polytope is assumed to be non-degenerate each point on the polytope has exactly 3 edges from it.

Let us divide the points that do not satisfy cases 1 and 2 of the conjecture into 2 categories:

Type1: Those points with exactly 1 of the 3 edges incident being a positive edge. (Refer to figure 2).

Type2: Those points with exactly 2 of the 3 incident edges being a positive edge. (Refer to figure 3).

Type1: Refer to figure 2.

Consider Constraint C. We claim that point P is the point with the best objective function value that lies on constraint C; i.e., $\forall p$ lying on constraint C, $f(p) \leq f(P)$ *Proof:* Clearly P is a local optimum on C. By convexity therefore P is also the global optimum on C. Therefore when we move from P to Q we are guaranteed that during the further course of the walk we will never meet constraint C again.

Type2: Refer to Figure 3.

We need to consider the following subcases.

Consider points Q and R leading from P. These points may themselves be either type1 or type 2 vertices. We therefore consider the following 2 subcases.

- At least one of Q or R is a type1 vertex. (Refer to figure 4).

Let us assume without loss of generality that Q is a type 1 vertex. Suppose we choose to move to Q. We claim then that constraint E is being visited for the first time.

Proof: It is clear that Q is a local minimum as far as constraint E is concerned. By convexity then, Q is also a global minimum for E. Hence conjecture holds for this case.

- Both P and Q are type 2 vertices. (Refer to figure 5).

In this case, let us assume without loss of generality that Q has a higher objective function value than R. Suppose we choose to move to Q. Then we claim that constraint F will never be visited again by the walk.

Proof: Q has a higher objective function value than R. But R is the optimum as far as constraint F is concerned. Hence R can be never visited again by the random walk. Hence the conjecture holds in this case too. \square

4 Side Effects in 2 and 3 dimensions: deterministic linear time algorithms

An interesting side-effect of the lemma in 2 and 3 dimensions is to yield deterministic linear time algorithms.

In the 2 dimensional case, there is only one edge at any point that leads to an improvement of the objective function. Hence algorithm RANDOM really has no choice at all, it will terminate in n steps.

In the 3 dimensional case consider the following replacement to step 2 of algorithm RANDOM.

If P is a type 1 vertex, there is no choice, only one edge leads up, choose it.

If P is a type 2 vertex then

if Both Q and R are type 2 vertices move to the vertex that leads to a greater improvement
else move to the type 1 vertex among Q and R .

(choice does not matter if both are type 1 vertices)

It is clear that this algorithm satisfies the conjecture at each step. Hence it is guaranteed to terminate within $2n$ steps. We do a constant amount of work at each step. Hence a linear time algorithm.

5 Open problems

The main open problem that is left unanswered by this paper is a general proof of the conjecture in d dimensions. In fact, it is not clear that the conjecture holds in general, however, counter examples, seem very difficult to construct. One reason why it is difficult to extend the same proof technique (by cases) to higher dimensions is that the number of cases keep increasing and certain cases are not amenable to easy treatment.

The more challenging question of a deterministic strongly polynomial algorithm is also left unanswered by this paper. One possible approach would be to use the the proof technique used in d dimensions to construct a deterministic algorithm, as we have done in 2 and 3 dimensions.

References

- [1] H. Edelsbrunner. *Algorithms in combinatorial geometry*. EATCS Monographs on Theoretical Computer Science. Springer Verlag, 1987.

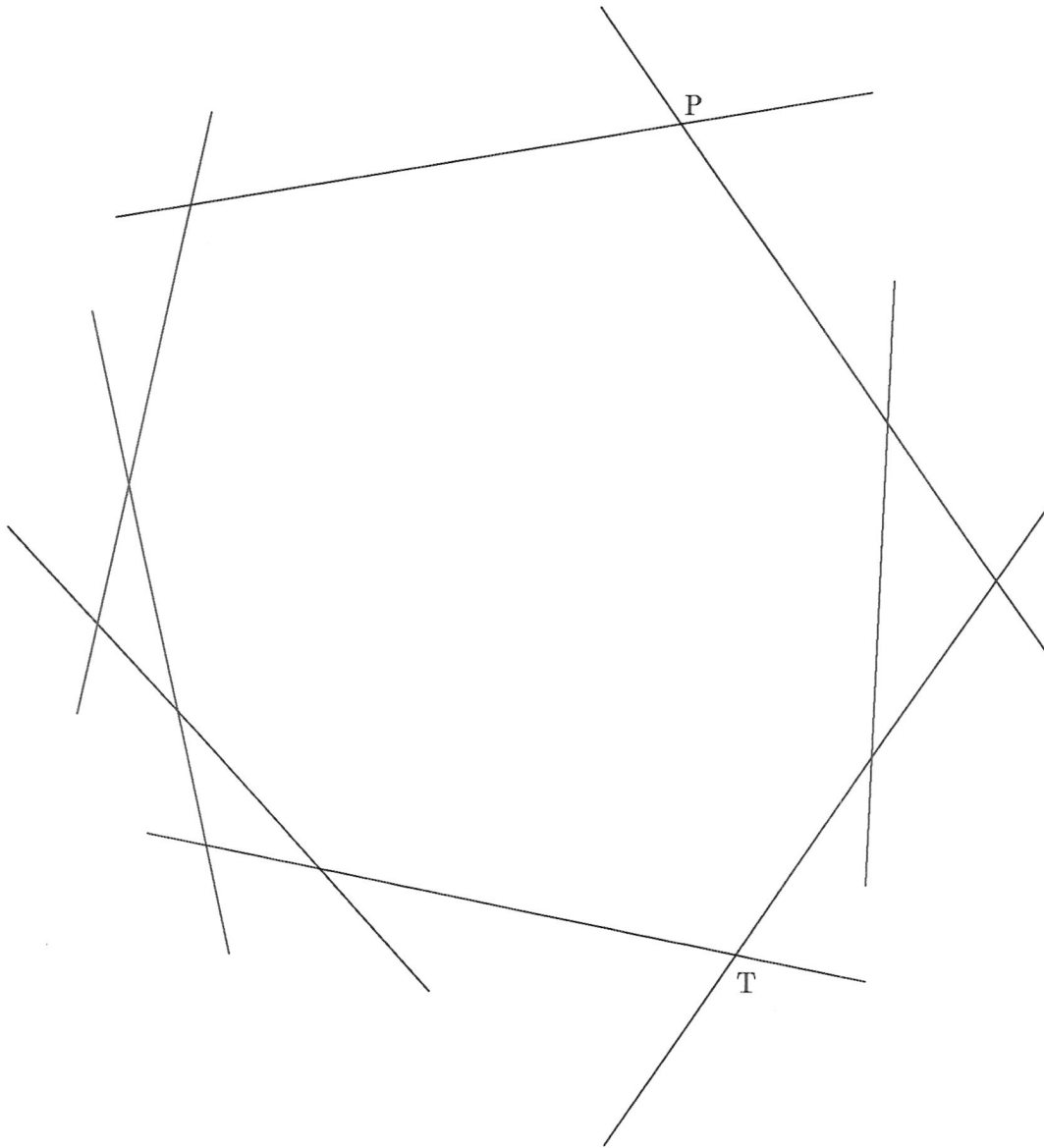


Figure 1: The 2 Dimensional Case

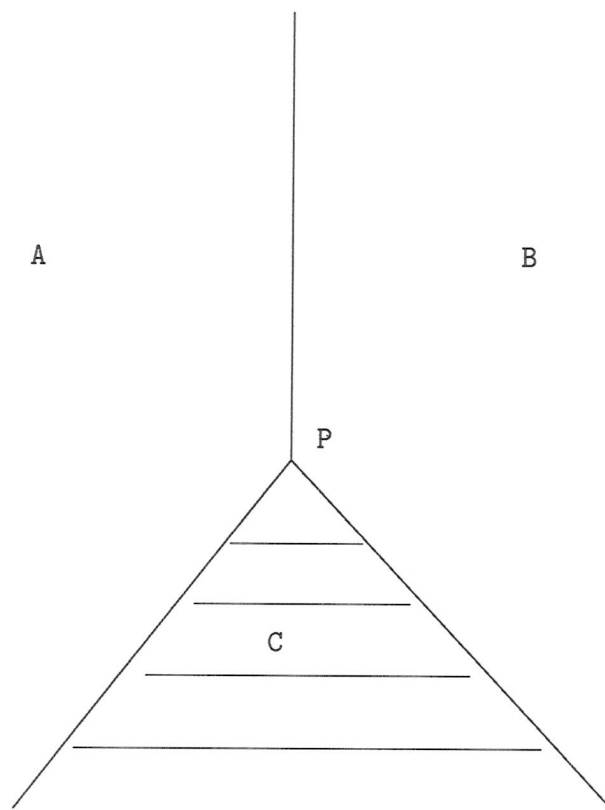


Figure 2: Points of the Type 1

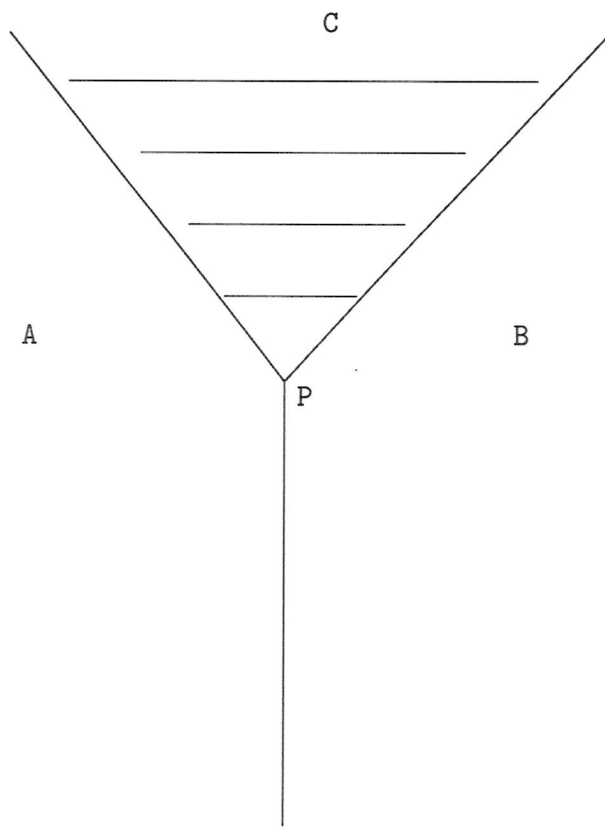


Figure 3: Points of the Type 2

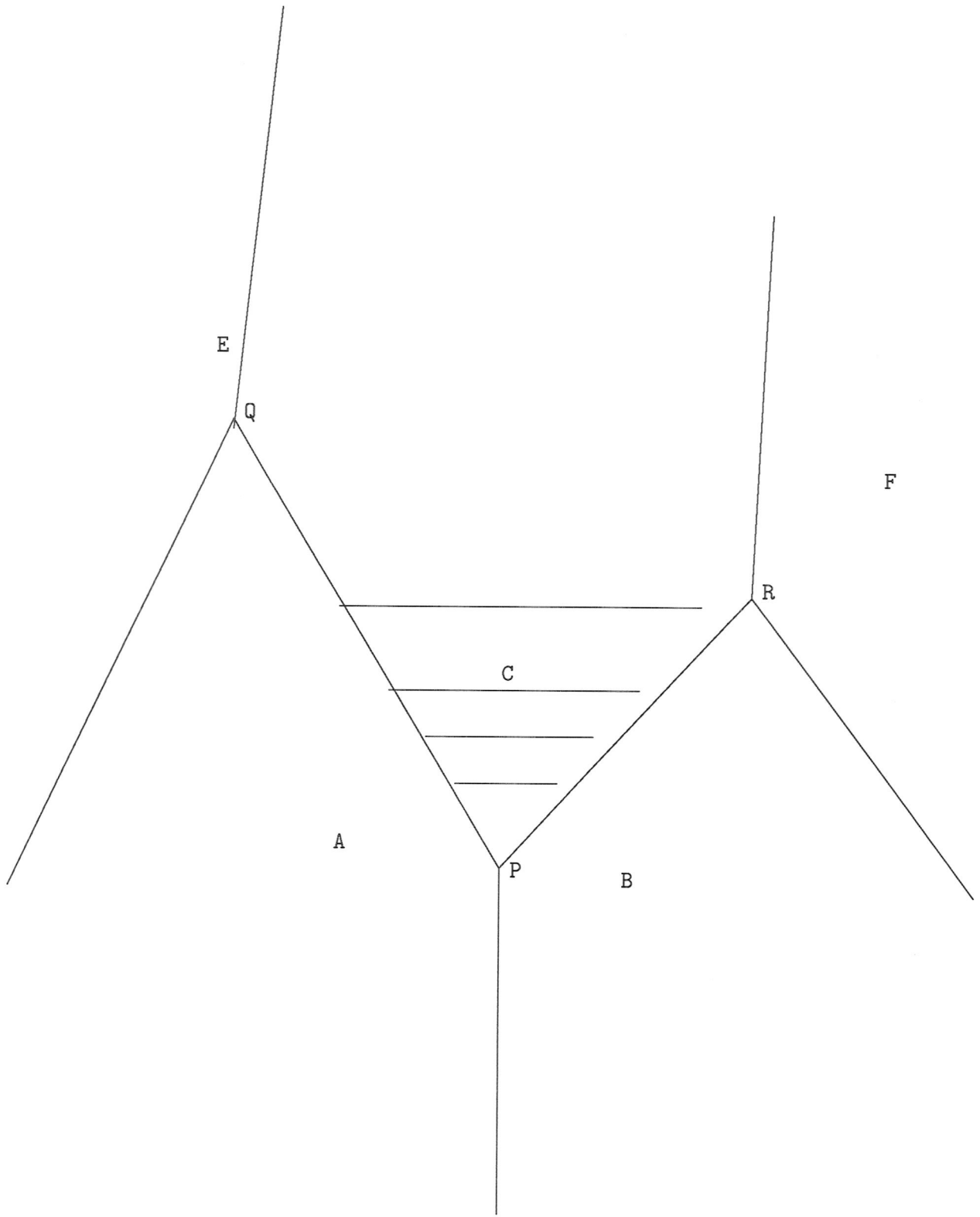


Figure 5: Subcase of Type 2 points