A New Approach to Strongly Polynomial Linear Programming

Mihály Bárász and Santosh Vempala
Brief history of LP

• Max c.x s.t. Ax <= b.
  (several equivalent formulations)

• Simplex method, 1947, Dantzig.
  Many variants.
  Still most popular LP algorithm.
  No polynomial guarantee known.

Assume nondegenerate: each vertex is intersection of n facets.
History: polynomial algorithms

  - Nestorov, Nemirovski, ...
- Perceptron method, Minsky-Papert, 1969
  - Polynomial perceptron, Dunagan-V. 2004

All these methods are geometric. They “scale” space to make the problem easier; complexity depends on #bits in the input.
Strongly polynomial cases

- Linear programming in fixed dimension, Megiddo, 1984.

These are specialized combinatorial algorithms.
Simplex pivot rules

• Assume polyhedron is nondegenerate, i.e., each vertex is the intersection of \( n \) facets.

• Then simplex moves from vertex to vertex, improving objective value.

• The rule for determining the next vertex is a “pivot” rule.
Simplex pivot rules

Deterministic pivot rules

- Dantzig’s largest coefficient rule. Exponential example: Klee and Minty, 1972.
- Steepest increase. Ex: Goldfarb and Sit, 1979.

- All examples are “variations” of Klee–Minty’s, and fit a general construction called deformed products defined by Amenta and Ziegler.
Simplex pivot rule

- Randomized edge rule: Among all pivots that improve the objective pick one at random.
- Complexity is open.
- Best known upper bound (G. Kalai), is subexponential.
- “Is this pivot rule polynomial?”

- This question has dominated the search for strongly polynomial algorithms.
Hirsch conjecture

• Diameter of a polytope with m facets is at most m-n+1
  (Diameter = Diameter of graph induced by vertices and edges of polytope)

• Best known upper bound is super-polynomial (G. Kalai).

• Long-standing open problem to prove the conjecture or even get a polynomial upper bound.
Hirsch vs Simplex

• Complexity of any simplex pivot rule gives an upper bound on diameter of polytope graph.
• Thus proving randomized simplex conjecture would also be a major combinatorial breakthrough.
• Is strongly polynomial LP really a nongeometric, combinatorial question?
But,

- [Matousek-Szabo] Take a polytope combinatorially equivalent to a hypercube; orient the edges arbitrarily, so that each face has a unique sink. There exist orientations for which the random edge pivot rule is exponential! (pivot rule runs by picking a random out-edge at each vertex visited)

- Does this imply strongly polynomial pivot rules are impossible? NO, because not all orientations are geometrically realizable.

- However, it does suggest that the geometry plays an important role.
New Approach

- Algorithm will be affine-invariant
- So complexity will not depend on how the input is scaled.

Two step iteration:
- At current vertex, pick a line to travel along in an affine-invariant manner and move along the line.
- Go to vertex of at least as high objective value in the face reached.
Step 1: Affine-invariant direction

How to pick a direction?
– Compute the set of improving rays (edges that lead to vertices of higher objective value)
– Take a linear combination of the improving rays.

Two candidate rules:
1. Average of all improving rays (centroid rule)
2. Random convex combination of improving rays (random rule).

Lemma: Both rules are affine-invariant.

(i.e. applying an affine transformation before computing the direction gives the same effect as applying it after)
New algorithm, *roughly*

- At current vertex,
  - pick direction
  - Follow to get to new face
  - Go to vertex of at least as high value

- Introduces many geometric shortcuts in the polytope.

- No bearing on Hirsch conjecture!
Step 2: go to vertex

• How to go to a vertex of at least as high value?
  – E.g., Follow gradient, keep adding facets hit as equalities till a vertex is reached.

• Doing this step arbitrarily can lead to an exponential number of iterations.

• So we will go to a vertex in an affine-invariant manner.
Algorithm: AFFINE
INPUT: Polyhedron $P$ given by linear inequalities $a_j \cdot x \leq b_j : j = 1 : m$, objective vector $c$ and vertex $z$.
OUTPUT: A vertex maximizing the objective value, or “unbounded”

While the current vertex $z$ is not optimal, repeat:
1. (Initialize) (a) Let $H$ be the set of indices of active inequalities at $z$
   (b) (Compute edges) For every $t$ in $H$ compute a vector
       \[ v_t : a_h \cdot v_t = 0 \text{ for } h \in H \setminus \{t\} \text{ and } a_t \cdot v_t < 0. \]
   (c) Let $T = \{ t \in H : c \cdot v_t \geq 0 \}$ and $S = H \setminus T$.

2. (Iteration) While $T$ is nonempty, repeat:
   (a) (Compute improving rays) For every $t$ in $T$ compute a vector
       \[ v_t \neq 0 : a_h \cdot v_t = 0 \text{ for } h \in H \setminus \{t\}, c \cdot v_t \geq 0 \]
       and the length of $v_t$ is the largest value for which $z + v_t$ remains feasible.
   (b) (Pick direction) compute a nonnegative combination $v$ of elements of $T$.
   (c) (Move) Let $r$ be maximal for which $z + r \cdot v$ is in $P$, if there is no such maximum, return “unbounded”. Move the current point: $z := z + r \cdot v$.
   (d) (Update inequalities) Let $s$ be the index of an inequality which becomes active. Let $t$ in $T$ be any index such that
       \[ \{ a_h : h \in \{s\} \cup S \cup T \setminus \{t\} \} \text{ is linearly independent.} \]
       Set $S := S \cup \{s\}$; $T := T \setminus \{t\}$ and $H := S \cup T$. 
Algorithm: notes

• In Step 2, one new active inequality is added in each iteration; thus a vertex is reached in at most \( n \) iterations.

• Step 2 can be viewed as a recursive application of the original procedure in lower dimensional faces.

• Each iteration: \( O(mn) \).
Analysis: How many iterations?

• Klee-Minty: n iterations.

• Main Theorem: For any polytope that is a deformed product, Algorithm Affine takes at most n outer iterations. (O(n^2) total iterations).

• Thus, algorithm is efficient on all known simplex counterexamples.
Idea

• Consider $P = V \times W$ (standard product)

• Then vertex $x$ of $P$ can be written as $x = (v,w)$

• Step in $P$ projects to step in $V$ or in $W$. Progress in $P$ goes hand-in-hand with either progress in $V$ or in $W$.

• Thus $f(P) \leq f(V) + f(W)$, where $f$ is the number of vertices visited.
Analysis

- Proof also works for mild perturbations of deformed polytopes, which can change the combinatorial structure considerably (thus algorithm is not specifically designed for these counterexample classes).

- Algorithm makes heavy use of geometric shortcuts through the interior of the polytope.
Next steps

• Polynomial upper bound? (not strongly polynomial, so one could use a geometric scaling type argument showing progress towards optimum).

• Counterexample?

• Upper bound for combinatorial cubes?

• Upper bound for random polytopes? (to get away from the combinatorial structure of known counterexamples)
Thank you!