

Polytopes, their diameter, and randomized simplex

Presentation by: Dan Stratila
Operations Research Center

Session 4: October 6, 2003

Based primarily on:

Gil Kalai. A subexponential randomized simplex algorithm (extended abstract).
In *STOC*. 1992. [Kal92a].

and on:

Gil Kalai. Linear programming, the simplex algorithm and simple polytopes.
Math. Programming (Ser. B), 1997. [Kal97].

Structure of the talk

1. Introduction to polytopes, linear programming, and the simplex method.
2. A few facts about polytopes.
3. Choosing the next pivot. Main result in this talk.
4. Subexponential randomized simplex algorithms.
5. Duality between two subexponential simplex algorithms.
6. The Hirsch conjecture, and applying randomized simplex to it.
7. Improving diameter results using an oracle for choosing pivots.

Polytopes and polyhedra

A polyhedron $P \subseteq \mathbb{R}^d$ is the intersection of finitely many halfspaces, or in matrix notation $P := \{x \in \mathbb{R}^d : Ax \leq b\}$, where $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$. A polytope is a bounded polyhedron.

Dimension of polyhedron P is $\dim(P) := \dim(\text{aff}(P))$, where $\text{aff}(P)$ is the affine hull of all points in P .

A polyhedron $P \in \mathbb{R}^d$ with $\dim(P) = k$ is often called a k -polyhedron. If $d = k$, P called full-dimensional. (Most of the time we assume full-dimensional d -polyhedra, not concerned much about the surrounding space.)

An inequality $ax \leq \beta$, where $a \in \mathbb{R}^d$ and $\beta \in \mathbb{R}$, is called valid if $ax \leq \beta$ for all $x \in P$.

Vertices, edges, ..., facets

A **face** F of P is the intersection of P with a valid inequality $ax \leq \beta$, i.e. $F := \{x \in P : ax = \beta\}$.

Faces of dimension $d - 1$ are called **facets**, 1 ... edges, and 0 ... vertices. Vertices are points, \Leftrightarrow basic feasible solutions (algebraic), or extreme points (linear cost).

Since $0x \leq 0$ is valid, P is a d -dimensional face of P . $0x \leq 1$ is valid too, so \emptyset is a face of P , and we define its dimension to be -1 .

Some vertices are connected by edges, so we can define a graph $G = (V(G), E(G))$, where $V(G) = \{v : v \in \text{vert}(P)\}$ and $E(G) = \{(v, w) \in V(G)^2 : \exists \text{ edge } E \text{ of } P \text{ s.t. } v \in E, w \in E\}$.

For unbounded polyhedra often a ∞ node is introduced in $V(G)$, and we add graph arcs (v, ∞) whenever $v \in E$ where E is an unbounded edge of P .

Example of a 3-polytope

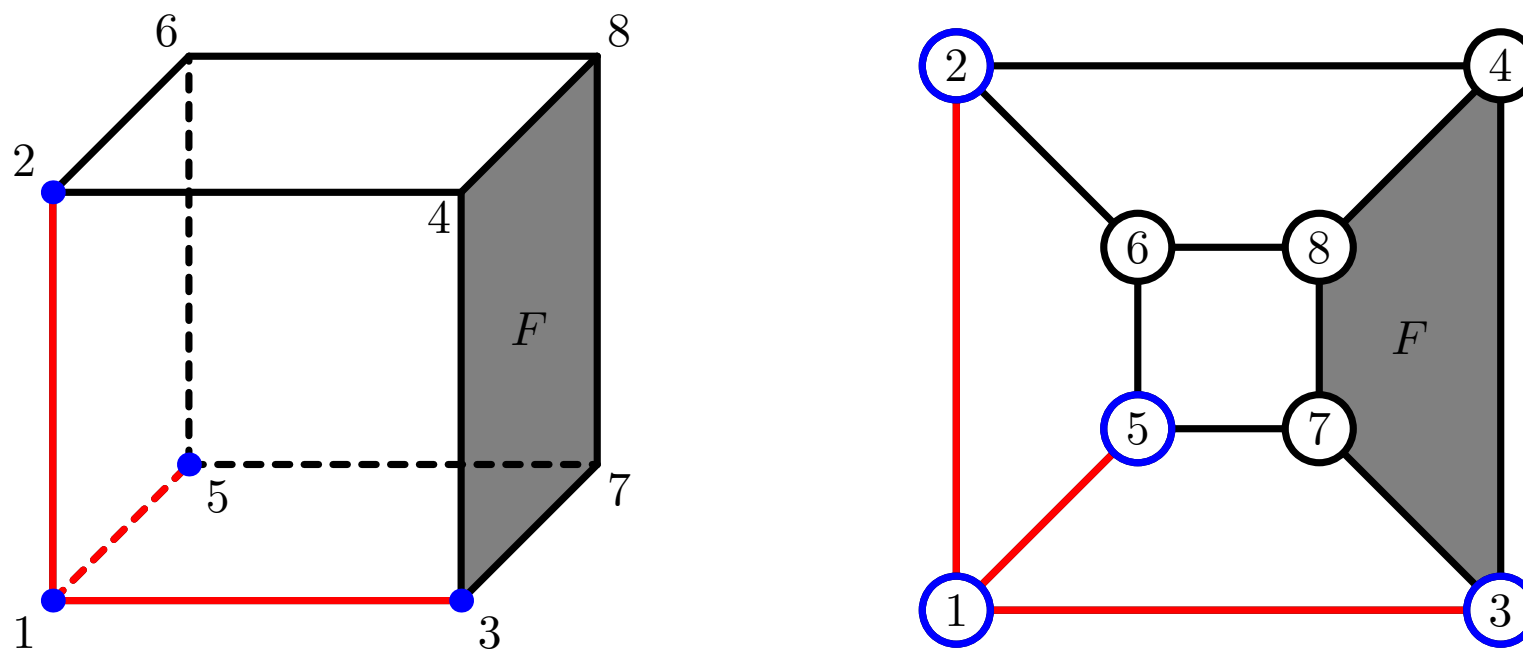


Figure 1: A 3-polytope (left) and its graph (right). Four vertices, three edges, and facet F are shown in corresponding colors.

Linear programming and the simplex method

A linear programming problem $\max\{cx : Ax \leq b\}$ is the problem of maximizing a linear function over a polyhedron.

- If problem bounded (cost of feas. sol. finite), optimum can be achieved at some vertex v .
- If problem unbounded, can find edge E of $P = \{x \in \mathbb{R}^d : Ax \leq b\}$ s.t. cx is unbounded on the edge.
- If problem bounded, vertex v is optimal $\Leftrightarrow cv \geq cw$ for all w adjacent to v (for all $(v, w) \in E(G)$).

Geometrically, the simplex method starts at a vertex (b.f.s.) and moves from one vertex to another along a cost-increasing edge (pivots) until it reaches an optimal vertex (optimal b.f.s.).

Vertices as intersections of facets

Any polytope can be represented by its facets $P = \{x \in \mathbb{R}^d : Ax \leq b\}$, or by its vertices $P = \text{conv}(\{v : v \in \text{vert}(P)\})$.

If vertices are given, then LP is trivial—just select the best one. Most of the time, facets are given. Number of vert. exponential in number of facets makes generating all vertices from the facets impractical.

Represent a vertex v as intersection of d facets. Any vertex is situated at the intersection of at least d facets; any non-empty intersection of d facets yields a vertex.

When situated at a vertex v given by $\bigcap_{i=1}^d F_i$, easy to find all adjacent vertices. Remove each facet F_i , and intersect with all other facets not in $\{F_1, \dots, F_d\}$.

Except when...

Degeneracy and simple polytopes

When a vertex is at the intersection of $> d$ facets, procedure above may leave us at the same vertex. Worse, sometimes **need** such changes before can move away from a vertex in cost-increasing direction.

This is (geometric) degeneracy. In standard form degenerate vertices yield degenerate b.f. solutions. Other “degenerate” b.f. solutions may appear because of redundant constraints.

If all vertices of P belong to at most d facets (\Rightarrow exactly d), P is called simple. Simple polytopes correspond to non-degenerate LPs, and have many properties [Zie95, Kal97].

We restrict ourselves to simple polytopes. Ok for two reasons: 1) any LP can be suitably perturbed to become non-degenerate; 2) perturbation can be made implicit in the algorithms.

A few facts about polytopes

Disclaimer: results not used or related to subexponential simplex pivot rules (main result in this talk).

The f -vector: $f_k(P) := \#$ of k -faces of P .

Degrees: let $\deg_c(v)$ w.r.t. to some objective function c be the $\#$ of neighboring vertices w with $cw < cv$.

The h -vector: $h_{k,c}(P) := \#$ of vertices of degree k w.r.t. objective c in P .

Note: there is always one vertex of degree d , and one of degree 0.

Property: $h_{k,c}(P) = h_k(P)$, independent of c .

$$h_{k,c}(P) = h_k(P), \text{ proof (1/2)}$$

Proof. Count $p := |\{(F, v) : F \text{ is a } k\text{-face of } P, v \text{ is max. on } F\}|$, in two ways.

Pick facets. Because c in general position $\Rightarrow v$ unique for each F , hence $p = f_k(P)$.

On the other hand, pick a vertex v , and assume $\deg_c(v) = r$. Let $T = \{(v, w) : cv > cw\}$, by definition $|T| = r$.

For simple polytopes, each vertex v has d adjacent edges, and any k of them define a k -face F that includes v .

So, # of k -facets that contain v as local maximum is $\binom{|T|}{k} = \binom{r}{k}$.

$$h_{k,c}(P) = h_k(P), \text{ proof (2/2)}$$

Summing over all $v \in \text{vert}(P)$, we obtain $f_k(P) = \sum_{r=k}^d h_{r,c}(P) \binom{r}{k}$.

Equations linearly independent in $h_{r,c}$. This completely determines $h_{r,c}(P)$ in terms of $f_k(P)$. But $f_k(P)$ independent of c , so same true for $h_r(P)$. \square

The Euler Formula and Dehn-Sommerville Relations

We can express $h_k(P) = \sum_{r=k}^d (-1)^{r-k} f_r(P) \binom{r}{k}$.

We know that $h_0(P) = h_d(P) = 1$, hence $f_0(P) - f_1(P) + \cdots + (-1)^d f_d(P) = 1$,
or $f_0(P) - f_1(P) + \cdots + (-1)^{d-1} f_{d-1}(P) = 1 - (-1)^d$.

In 3 dimensions, $V - E + F = 2$.

Back to $h_{k,c}(P)$, note that if $\deg_c(v) = k$ then $\deg_{-c}(v) = d - k$.

Because of independence of c , we obtain the Dehn-Sommerville Relations:
 $h_k(P) = h_{d-k}(P)$.

Cyclic polytopes and the upper bound theorem

A cyclic d -polytope with n vertices is defined by n scalars t_1, \dots, t_n as $\text{conv}(\{(t_i, t_i^2, \dots, t_i^d) : i = \overline{1, d}\})$. Can use other curves too.

All cyclic d -polytopes with n vertices have same structure, denote by $C(d, n)$.

The polar $C^*(d, n) := \{x \in (\mathbb{R}^d)^* : xv \leq 1, \forall v \in C(d, n)\}$ is a simple polytope.

Property: $C(d, n)$ has the maximum number of k -facets for any polytope with n vertices.

The polar $C^*(d, n)$ has the maximum number of k -facets for any polytope with n facets (the face lattice).

Exact expression for f_{k-1} elaborate, but a simple one is $f_{k-1} = \sum_{i=0}^{\min\{d, k\}} \binom{d-i}{k-i} h_i(P)$. For more interesting details, see [Zie95].

Abstract objective functions and the combinatorial structure

An abstract objective function assigns a value to every vertex of a simple polytope P , s.t. every non-empty face F of P has a unique **local** maximum vertex.

AOFs are gen. of linear objective functions. Most results here apply.

The combinatorial structure of a polytope is all the information on facet inclusion, e.g. all vertices, all edges and the vertices they are composed of, all 3-facets and their composition, etc.

Lemma: Given graph $G(P)$ of simple polytope P , connected subgraph $H = (V(H), E(H))$ with k vertices defines a k -face if and only if \exists AOF s.t. all vertices in $V(H)$ come before all vertices in $V(G(P)) \setminus V(H)$.

Property: The combinatorial structure of any simple polytope is determined by its graph.

Main result in this talk—context

In the simplex algorithm we often make choices on which vertex to move to next. Criteria for choosing the next vertex are called pivot rules.

In the early days, “believed” simple rules guarantee a polynomial number of vertices in path. Klee and Minty [KM72] have shown exponential behaviour.

After that, not known even if LP can be solved in polynomial time at all, until [Kha79]. But still,

- Finding a pivot rule (deterministic or randomized) that would yield a polynomial number of vertex changes—open since simplex introduced.

For some $f(n)$, exponential: $f(n) \in \Omega(k^n)$, $k > 1$. Polynomial: $f(n) \in O(n^k)$ for some fixed $k \geq 1$. Subexponential $f(n) \notin O(n^k)$ for any fixed $k \geq 1$ and $f(n) \notin \Omega(k^n)$ for any fixed $k > 1$.

Main result in this talk

Shortly before a different technique in [SW92], shortly afterwards a subexponential analysis for it in [MSW96].

- The first randomized pivot rule that yields **subexponential** expected path length (presented from [Kal92a, Kal97]).

Expectation over internal random choices of algorithm; applicable to all LP instances.

Immediate application to diameter of polytopes and the Hirsch conjecture (more about diameters and the Hirsch conjecture later).

Algorithm 1

Simplest randomization (Dantzig, others): next vertex random with equal prob. among neighboring cost-increasing vertices. Hard to analyze in general; Gärtner, Henk and Ziegler show quadratic **lower** bounds on Klee-Minty cubes.

Reminder: P Given $P = \{x \in \mathbb{R}^d : Ax \leq b\}$, so in LP terms: $d = \#$ of variables, $n = \#$ of constraints. Also given $c \in \mathbb{R}^d$.

A1-1 (parameter r , start vertex v):

1. Find vertices on r facets F_1, F_2, \dots, F_r s.t. $\forall F_i, cv < \max\{cx : x \in F_i\}$.
2. Choose a facet F_k at **random** from F_1, F_2, \dots, F_r with equal probability.
3. Solve $\max\{cx : x \in F_k\}$ recursively. Let the optimum vertex be w .
4. Finish solving the problem from w recursively.

How is this a simplex algorithm?

A1-1: Implementation of step 1

For step 1, easy to find the first d facets. For the rest $r - d$ facets, let $k := d, z := v$ and proceed as follows:

1. Solve an LP from z with only the k facets recursively. Let result be z .
2. If z feasible for original problem, optimum found, A1-1 terminates.
3. Otherwise, first edge E on path that leaves P gives new facet F . Let z be the point in $E \cap F$. If r facets, **stop**; otherwise **go to** step 1.

Up to now we are tracing a path along the vertices of the original problem.

A1-1: Implementation of steps 2 and 3

First, note that when solving $\max\{cx : x \in F_k\}$, tracing a path along vertices of F_k . This is also a path along vertices of P , since we are working with F_k in [its dimension](#).

If $k = r$ or $k = r - 1$, then last vertex $\in F_k$, can continue our path in step 3.

But if $k < r - 1$, then [backtracking](#) from the last vertex found when discovered. Not “honest” simplex.

Easy to fix. Since facet F_k is chosen uniformly among facets F_1, \dots, F_r , this can be done by choosing uniformly among F_{i_1}, \dots, F_{i_r} , where i_1, \dots, i_r order in which facets encountered by step 1.

So, generate random k before step 1, and stop once reached k -th facet (Kalai also offers another workaround).

A1-1: Implementation of step 4

A facet F is **active** w.r.t. a vertex v if $\exists w \in \text{vert}(F)$ s.t. $cw > cv$.

Apply algorithm recursively from w using only those facets which are active. At most $n - 1$ such facets (F_k from step 3 cannot be active).

Complexity analysis

Let $f_1(P, c) := \mathbb{E}[\# \text{ of pivots when solving } \max\{cx : x \in P\} \text{ by A1}]$. Let $f_1(d, n) := \max \{ f_1(\{x \in \mathbb{R}^d : Ax \leq b\}, c) : A \in \mathbb{R}^{n \times d}, c \in \mathbb{R}^d, b \in \mathbb{R}^n \}$.

First part of analysis: probabilistic reasoning to obtain a recurrence relation on $f_1(d, n)$.

Second part: solving the recurrence (using generating functions).

A1-1: Analysis of step 1

It takes $f_1(d, i)$ to solve an LP in d variables with i facets using A1-1. So step 1 takes at most $\sum_{i=1}^r f_1(d, i)$.

In step 2, note that there is at least one vertex in the path for each random number generated.

In step 3, the expected complexity is $f_1(d - 1, n - 1)$.

After step 3, we only need to consider the active facets w.r.t. w . How many? Assume facets F_1, \dots, F_r are ordered according to their top vertex. Then selecting facet $i \Rightarrow$ at most $n - i - 1$ active facets w.r.t. w .

So with probability $\frac{1}{r}$ we will have $n - i - 1$ active facets, for $i = 1, 2, \dots, r$.

Rec.: $f_1(d, n) = 2 \sum_{i=d}^r f_1(d, i) + f_1(d - 1, n - 1) + \frac{1}{r} \sum_{n-r-1}^{n-1} \sum_{i=d}^l f_1(d, n - i)$.

(The real) Algorithm 1

Before analyzing the recurrence, we improve A1-1.

A1 (parameter $c > \frac{1}{2}$, start vertex v):

1. Starting from v , find vertices on r active facets F_1, F_2, \dots, F_r .
2. Choose a facet F_k at random from F_1, F_2, \dots, F_r with equal probability.
3. Solve $\max\{cx : x \in F_k\}$ recursively. Let the optimum vertex be w .
4. Let $l := |\{F : F \text{ active w.r.t. } w\}|$. If $l > (1 - c)n$ then let $v := w$ and go to step 1; otherwise, finish solving recursively from w .

Let $r := \max\{\frac{n}{2}, d\}$. What is probability of not returning to step 1? If $r = n$ easily geometric with ratio = $P(\text{no return}) = P(l < (1 - c)n) = 1 - c$.

In general, analysis more complicated.

A1: the recurrence

$$f_1(d, n) \leq \frac{2}{1-c} \sum_{i=d}^r f_1(d, i) + \frac{1}{1-c} f(d-1, n-1) + \frac{1}{(1-c)n} \sum_{i=\lfloor cn \rfloor}^r f_1(d, n-i)i. \quad (1)$$

Taking $b = \frac{1}{1-c}$, we get a bound of $f_1(d, n) \leq b^d (6n)^{\log_b n} \binom{d + \log_b n}{\log_b n}$.

Taking $c = 1 - \frac{1}{\sqrt{d}}$, we obtain

$$f_1(d, n) \leq n^{16\sqrt{d}}. \quad (2)$$

Algorithm 2

Delete step 4, repeat steps 1–3 until step 1 detects an optimal vertex. Equivalent to setting $c := 1$ in 1.

A2 (start vertex v):

1. Starting from v , find vertices on r active facets F_1, F_2, \dots, F_r . If unable to find r distinct active facets \Rightarrow opt. vertex found.
2. Choose a facet F_k at random from F_1, F_2, \dots, F_r with equal probability.
3. Solve $\max\{cx : x \in F_k\}$ recursively. Let the optimum vertex be w .
4. Delete inactive facets, set $v := w$, and go to step 1.

Recurrence is:

$$f_2(d, n) \leq f_2(d-1, n-1) + \sum_{i=d}^{n/2} g(d, i) + \frac{2}{n} \sum_{i=1}^{n/2} g(d, n-i). \quad (3)$$

A2: the bounds

By solving the recurrences, we get $f_2(d, Kd) \leq 2^{C\sqrt{Kd}}$, and $f_2(d, n) \leq n^C \sqrt{\frac{d}{\log d}}$.

When co-dimension ($m := n - d$) is small the following bound is very useful:
 $2^{C\sqrt{m \log d}}$.

Next: the interesting A3.

Algorithm 3

A3 (start vertex v):

1. From d facets containing v , select a facet F_0 at random, with equal probability.
2. Apply A3 to F_0 recursively, and let w be the optimum.
3. Set $v := w$ and go to step 1.

Simple! This algorithm is the dual of the algorithm discovered by Sharir and Welzl [SW92] (more about this later).

For now, note that in a simple polytope, there can be at most 1 non-active facet adjacent to any vertex v , unless v is optimal.

A3: the recursion

First, if all facets active, with probability $\frac{1}{d}$ the chosen facet yields $n - i$ active facets at step 3, for $i = 1, \dots, d$.

Second, if one facet inactive, with probability $\frac{1}{d-1}$ the chosen facet yields $n - i$ facets at step 3, for $i = 1, \dots, d - 1$.

Second alternative is worse, so we factor it in and obtain recursion $f_3(d, n) \leq f(d - 1, n - 1) + \frac{1}{d-1} \sum_{i=1}^{d-1} f(d, n - i)$.

This yields bound $f_3(d, n) \leq e^{C\sqrt{n \log d}}$. A4, which we do not present now, gives $e^{C\sqrt{d \log n}}$, better, like A2.

Subexponential behaviour

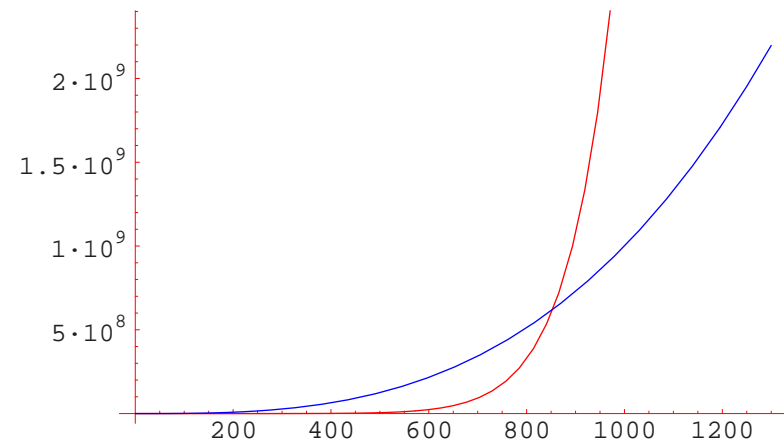
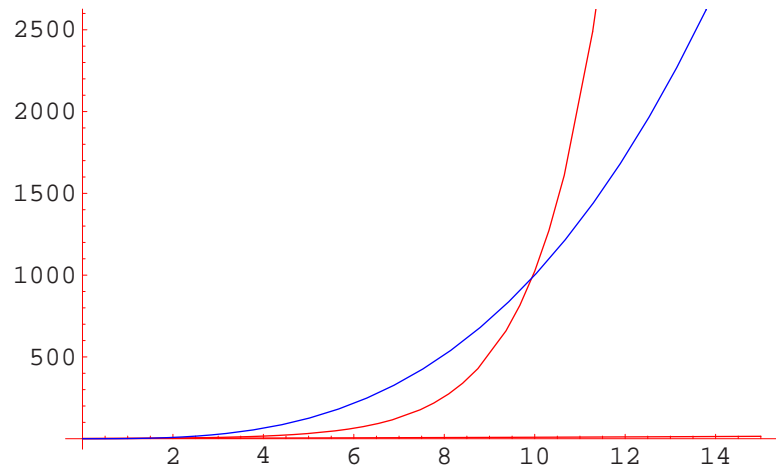


Figure 2: Asymptotic behaviour of the exponential 2^d , the polynomial d^3 and the subexponential $2^{\sqrt{d}}$.

The duality of A4 and the Sharir-Welzl algorithm

Following [Gol95], we show what the Sharir-Welzl algorithm (Algorithm B) [MSW96] does to the polytope of the dual LP. Reminder: algorithm B was called BasisLP in the second part of Session 3.

Unlike before, we'll use lots of traditional LP terminology. Let H be a set of constraints, and B a set of constraints that define a basis.

B (set of constraints H , basis C with $C \subseteq H$):

1. Begin at C .
2. Choose random constraint $h \in H \setminus C$
3. Solve LP recursively with constraints $H \setminus \{h\}$, from C . Let result be B .
4. If B violates h , then form new basis $C' := \text{basis}(B, h)$; otherwise optimum found.
5. Let $C := C'$ and go to step 1.

The dual problems

Primal (we run B on this problem):

$$\max\{cx : Ax \leq b\}. \quad (4)$$

Dual (we see what B does to this problem):

$$\min\{yb : y \geq 0, yA = c\}. \quad (5)$$

We will imagine the dual problem in the $yA = c$ space, so only the inequality constraints $y \geq 0$ define facets; $yA = c$ is simply an affine transformation of the space.

The correspondence, up to step 3

$C :=$ initial feasible basis $\leftrightarrow C :=$ initial feasible vertex. (Not true.)

Choosing random $h \in H \setminus C \leftrightarrow$ choosing random facet $y_h \geq 0$ that contains C :

- We know, by complementary slackness, that un-tight constraints in the primal correspond to 0-level variable components in the dual.
- Only the $y_i \geq 0$ constraints define facets in the dual polytope.
- Active constraint at a vertex C defines a facet that contains C .

Solve recursively the LP with constraints $H \setminus h$ starting from $C \leftrightarrow$ solve recursively LP on facet $y_h = 0$ starting from C .

- If we remove a constraint in the primal, this is the same as requiring $y_h = 0$ in the dual.

The correspondence, step 4

In both cases we obtain a new basis ($\Leftarrow \Rightarrow$ vertex) B .

If B violates $h \Leftarrow \Rightarrow B$ not optimal: infeasible primal solutions correspond to suboptimal dual solutions.

$C' := \text{basis}(B, h)$ is a pivot operation $\Leftarrow \Rightarrow C' :=$ move away along an unique edge from $y_h = 0$. **But**, there is no “move along an edge” in A3!

Slight adjustment to A3, in order to achieve perfect duality: in step 1, pick a random facet among d active facets.

After we found optimum w on facet F_0 , this facet is not active (since w optimum on it). Moreover, this facet is defined by $d - 1$ of the edges at w (in a simple polytope, every $d - 1$ edges at a vertex define a facet, and conversely).

The correspondence, steps 4 and 5

Hence only one remaining edge can be cost-improving \Rightarrow any simplex algorithm will take it. So, our algorithm takes it when it tries to find the d -th facet.

But this implies that the choice of d facets available to A3 is exactly the same as the choice of d facets available after moving along the unique edge.

In steps 5 and 1 this yields the same choice of un-tight constraints in the primal!

So, a variant of the Sharir-Welzl algorithm B when followed on the dual polytope is exactly the same as the (slightly modified) Kalai algorithm A3.

The Hirsch conjecture

The diameter $\delta(G(P))$ of a polytope is the diameter of its graph, i.e. the longest shortest path between any pair (v, w) of vertices. Denote by $\delta(\vec{G}(P))$ the longest shortest path in the cost-function directed graph of P .

Let $\Delta(d, n) := \max\{\delta(G(P)) : P \text{ is a } d\text{-polytope with } n \text{ facets}\}$. Let $H(d, n) := \max\{\delta(\vec{G}(P)) : P \text{ is a } d\text{-polytope with } n \text{ facets, } c \text{ is any cost function}\}$.

Clearly, the simplex algorithm cannot guarantee a better performance on P than $\delta(G(P))$. Moreover, $\Delta(d, n) \leq H(d, n)$.

Conjecture (Hirsch, [Dan63]): $\Delta(P) \leq n - d$.

False for unbounded polyhedra [KW67]. Lower bound of $\Delta(d, n) \geq n - d + \lfloor d/5 \rfloor$.
Still best lower bound!

Status of Hirsch conjecture for polytopes

Still open!

Exponential bound $\Delta(d, n) \leq n2^{d-3}$ [Larman, 1970].

Until recently (w.r.t. 1992) no sub-exponential bound known.

Bounds of $n^{2 \log d + 3}$ and $n^{\log d + 1}$ in [Kal92b, KK92] respectively.

How randomized pivot rules affect the Hirsch conjecture? A randomized simplex algorithm gives only **hope** that a deterministic algorithm with the same complexity may be devised.

But, because $\mathbb{E}[\dots]$ is over choices, at least one of these choices (even if we don't know it), yields a path of length less than $\mathbb{E}[\dots]$.

A friendly oracle

So all our bounds on the number of simplex pivots, immediately apply to $H(d, n)$ (since simplex takes only monotone paths) and hence to $\Delta(d, n)$.

In algorithms A1–A4 we spend at most $O(d^2n)$ for each pivot, and generate at most 1 random number per pivot.

What if we allow much more time per pivot? Result will still apply to the Hirsch conjecture. Do not want algorithms such as “construct the graph, find the shortest path, then parse it”, since analysis is equivalent to Hirsch conjecture.

But, can still make use of a more powerful oracle that makes choices at each pivot step. Works from within Algorithm 4.

Algorithm 4

A4 (start vertex v):

1. From the active facets w.r.t v , select a facet F_0 at random, with equal probability.
2. Find vertices recursively until reached F_0 , or until optimum found.
3. Solve recursively on F_0 . Let result be v .
4. Go to step 1.

As mentioned, bound of $e^{C\sqrt{d\log n}}$.

Now instead of selecting F_0 at random, order all facets in increasing order F_1, \dots, F_n of $\max\{cx : x \in F_i\}$. Select F_0 s.t. $\max\{x : x \in F_0\}$ above the median ($i > \lfloor n/2 \rfloor$ in the ordering).

Let $f_4(d, n)$ be the number of steps using the oracle.

A4: the recursion

At most $f_4(d, n/2)$ pivots for step 2. Pivots from step 4 (counting everything that happens) is again $f_4(d, n/2)$. Clearly, step 3 takes $f_4(d - 1, n - 1)$. Hence:

$$f_4(d, n) \leq 2f_4(d, \lfloor n/2 \rfloor) + f_4(d - 1, n - 1) + 1. \quad (6)$$

To solve this, let $\phi(d, t) := 2^t f_4(d, 2^t)$. Then, from (6), we obtain $\phi(d, t) \leq \phi(d - 1, t) + \phi(d, t - 1)$.

By simple combinatorial reasoning (counting all paths to the bottom), this yields $\phi(d, t) \leq \binom{d+t}{d}$. So $f_4(d, n) \leq n \binom{\log n + d}{\log n}$.

Finally, by combinatorics $\binom{a+b}{a} \leq a^b$ (or b^a), we obtain $f_4(d, n) \leq n^{\log d + 1}$.

How smart the oracle?

Not too smart:

1. Solve $\max\{cx : x \in F_i\}$ for each facet F_i using some polynomial LP algorithm.
2. Rank all values.

Only needs to be done once per instance. Hence bound $n^{\log d+1}$ can be achieved by a polynomial-pivot-time **deterministic** simplex algorithm!

Not combinatorial; overkill.

References

- [Dan63] George B. Dantzig. *Linear programming and extensions*. Princeton University Press, Princeton, N.J., 1963.
- [Gol95] Michael Goldwasser. A survey of linear programming in randomized subexponential time. *ACM SIGACT News*, 26(2):96–104, 1995.
- [Kal92a] Gil Kalai. A subexponential randomized simplex algorithm (extended abstract). In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 475–482. ACM Press, 1992.
- [Kal92b] Gil Kalai. Upper bounds for the diameter and height of graphs of convex polyhedra. *Discrete Comput. Geom.*, 8(4):363–372, 1992.
- [Kal97] Gil Kalai. Linear programming, the simplex algorithm and simple polytopes. *Math. Programming*, 79(1-3, Ser. B):217–233, 1997.
- [Kha79] L. G. Khachiyan. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR*, 244(5):1093–1096, 1979.
- [KK92] Gil Kalai and Daniel J. Kleitman. A quasi-polynomial bound for the diameter of graphs of polyhedra. *Bull. Amer. Math. Soc. (N.S.)*, 26(2):315–316, 1992.
- [KM72] Victor Klee and George J. Minty. How good is the simplex algorithm? In *Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin)*, pages 159–175. Academic Press, New York, 1972.
- [KW67] Victor Klee and David W. Walkup. The d -step conjecture for polyhedra of dimension $d < 6$. *Acta Math.*, 117:53–78, 1967.

- [MSW96] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4-5):498–516, 1996.
- [SW92] Micha Sharir and Emo Welzl. A combinatorial bound for linear programming and related problems. In *STACS 92 (Cachan, 1992)*, volume 577 of *Lecture Notes in Comput. Sci.*, pages 569–579. Springer, Berlin, 1992.
- [Zie95] Günter M. Ziegler. *Lectures on polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995.