Mathematical Programming 79 (1997) 217-233

Linear programming, the simplex algorithm and simple polytopes

Gil Kalai [1]

Institute of Mathematics, Hebrew University of Jerusalem, Jerusalem, Israel

Received 27 April 1997; accepted 7 May 1997

Abstract

In the first part of the paper we survey some far-reaching applications of the basic facts of linear programming to the combinatorial theory of simple polytopes. In the second part we discuss some recent developments concerning the simplex algorithm. We describe subexponential randomized pivot rules and upper bounds on the diameter of graphs of polytopes. © 1997 The Mathematical Programming Society, Inc. Published by Elsevier Science B.V.

Keywords: Simplex algorithm; Randomized pivot rule complexity; Combinatorial theory of simple polytopes

1. Introduction

A convex polyhedron is the intersection P of a finite number of closed halfspaces in R^d. P is a d-dimensional polyhedron (briefly, a d-polyhedron) if the points in P affinely span R^d. A convex d-dimensional polytope (briefly, a d-polytope) is a *bounded* convex d-polyhedron. Alternatively, a convex d-polytope is the convex hull of a finite set of points which affinely span R^d.

A (nontrivial) face F of a d-polyhedron P is the intersection of P with a supporting hyperplane. F itself is a polyhedron of some lower dimension. If the dimension of F is k we call F a k-face of P. The empty set and P itself are regarded as trivial faces. 0-faces of P are called *vertices*, 1-faces are called *edges* and $(d-1)$-faces are called *facets*. For material on convex polytopes and for many references see Ziegler's recent book [32].

The set of vertices and (bounded) edges of P can be regarded as an abstract graph called the graph of P and denoted by $G(P)$.

[1] Email: kalai@math.huji.ac.il.

PII S0025-5610(97)00061-0

We will denote by $f_k(P)$ the number of k-faces of P. The vector $(f_0(P), f_1(P), \ldots, f_d(P))$ is called the f-vector of P. Euler's famous formula $V - E + F = 2$ gives a connection between the numbers V, E, F of vertices, edges and 2-faces of every 3-polytope.

A convex d-polytope (or polyhedron) is called *simple* if every vertex of P belongs to precisely d edges. Simple polyhedra correspond to *non-degenerate* linear programming problems. When you cut a simple polytope P near a vertex v with a hyperplane H which intersect the interior of P, the intersection $P \cap H$ is a $(d-1)$-dimensional simplex S. The vertices of S are the intersections of edges of P which contain v with H and the $(k-1)$-dimensional faces of S are the intersection of k-faces of P with H. The following basic property of simple polytopes follows:

- Let P be a simple d-polytope and let v be a vertex of P. Every set of k edges adjacent to v determines a k-dimensional face of P which contains the vertex v.

In particular there are precisely $\binom{d}{k}$ k-faces in P containing v and altogether 2^d faces (of all dimensions) which contain v.

Linear programming and the simplex algorithm

Linear programming is the problem of maximizing a linear objective function ϕ subject to a finite set of linear inequalities. The relevance of convex polyhedra to linear programming is clear. The set P of feasible solutions for a linear programming problem is a polyhedron.

There are two fundamental facts concerning linear programming the reader should keep in mind:

- If ϕ is bounded from above on P then the maximum of ϕ on P is attained at a face of P, in particular there is a vertex v for which the maximum is attained. If ϕ is not bounded from above on P then there is an edge of P on which ϕ is not bounded from above.
- A sufficient condition for v to be a vertex of P on which ϕ is maximal is that v is a *local maximum*, namely $\phi(v)$ is bigger or equal than $\phi(w)$ for every vertex w which is a neighbor of v.

The *simplex algorithm* is a method to solve a linear programming problem by repeatedly moving from one vertex v to an adjacent vertex w of the feasible polyhedron so that in each step the value of the objective function is increased. The specific way to choose w given v is called the *pivot rule*.

The d-dimensional simplex and the d-dimensional cube

The d-dimensional simplex S_d is the convex hull of $d+1$ affinely independent points in R^d. The faces of S_d are themselves simplices. In fact, the convex hull of every subset of vertices of a simplex is a face and therefore $f_k(S_d) = \binom{d+1}{k+1}$. The graph of S_d is the complete graph on $d+1$ vertices.

The d-dimensional cube C_d is the set of all points $(x_1, x_2, \dots, x_d)$ in R^d such that for every i, $0 \leqslant x_i \leqslant 1$. The vertices of C_d are all the $(0,1)$ vectors of length d and two vertices are adjacent (in the graph of C_d) if they agree in all but one coordinates, $f_k(C_d) = 2^{d-k}\binom{d}{k}$.

2. Applications of the fundamental properties of linear programming to the combinatorial theory of simple polytopes

Let P be a simple d-polytope, and let ϕ be linear objective function which attains different values on different vertices of P. Call such a linear objective function *generic*. (Actually it will be enough to assume only that ϕ is not constant on any edge of P.)

The fundamental fact concerning linear programming is that the maximum of ϕ on P is attained at a vertex v and that a sufficient condition for v to be the vertex of P on which ϕ is maximal is that v is a *local maximum*, namely $\phi(v)$ is strictly bigger than $\phi(w)$ for every vertex w which is a neighbor of v.

Every face F of P is itself a polytope and ϕ attains different values on distinct vertices of F. Among the vertices of F there is a vertex on which ϕ is maximal and again this vertex is the only vertex in F which is a local maximum of ϕ in the face F.

These considerations have far-reaching applications on the understanding of the combinatorial structure of simple polytopes. We refer the reader to Ziegler's book [32] for historical notes and for references to the original papers. Our presentation is also quite close to that in [26]. We hope that the theory of h-numbers described below will reflect back on linear programming but this is left to be seen.

Degrees and h-numbers

Let P be a simple d-polytope and let ϕ be a generic linear objective function. For a vertex v of P define the *degree* of v denoted by $\deg(v)$ to be the number of its neighboring vertices with smaller value of the objective function. Clearly, $0 \leqslant \deg(v) \leqslant d$.

Define now $h_k(P)$ to be the number of vertices of P of degree k. This number as we defined it depends on the objective function ϕ but we will soon see that it is actually independent from ϕ. We can see one sign for this already; no matter what ϕ is there will always be precisely one vertex of degree d (on which ϕ attains the maximum) and one vertex of degree 0 (on which ϕ attains the minimum). This follows at once from the fact that local maximum = global maximum.

To continue we will count pairs of the form (F, v), where F is a k-face of P and v is a vertex of F which is a local maximum (hence a global maximum) of ϕ in F.

On the one hand the number of such pairs is precisely $f_k(P)$ - the number of k-faces of P. This is because every k-face has a unique local maximum.

On the other hand, let us compute how many pairs contain a given vertex v of P. This depends only on the degree of v. Assume that $\deg(v) = r$ and consider the set of edges of P

$$T = \{[v, w] : \phi(v) > \phi(w)\}.$$

Thus $|T| = r$. As we mentioned above every set S of k edges containing v determines a k-face $F(S)$ containing v. In this face the set of edges containing v is precisely S. In order for v to be a local maximum in this face it is necessary and sufficient that for every edge $[v, w]$ in S, $\phi(v) > \phi(w)$. This occurs if and only if $S \subset T$. Therefore, the number of k-faces containing v for which v is a local maximum is precisely the number of subsets of T of size k, namely $\binom{r}{k}$.

Summing over all vertices v of P and recalling that $h_k(P)$ denotes the number of vertices of degree k we obtain

$$(*) \qquad \sum_{r=0}^{d} h_r(P)\binom{r}{k} = f_k(P), \quad k = 0, 1, \ldots, d.$$

Note that this formula describes the f-vector of P – $(f_0(P), f_1(P), \ldots, f_d(P))$ as an upper triangular matrix (with ones on the diagonal) times the h-vector of P – $(h_0(P), h_1(P), \ldots, h_d(P))$. Therefore, the h-numbers are in fact linear combinations of the face numbers, and in particular they do not depend on the linear objective function ϕ.

Put

$$F_P(x) = \sum_{k=0}^{d} f_k(P)x^k, \qquad H_P(x) = \sum_{k=0}^{d} h_k(P)x^k.$$

Relation $(*)$ gives

$$\begin{aligned} H_P(x+1) &= \sum_{r=0}^{d} h_r(P)(x+1)^r \\ &= \sum_{k=0}^{d}\Big(\sum_{r=0}^{d} h_r(P)\binom{r}{k}\Big)x^k = \sum_{k=0}^{d} f_k(P)x^k = F_P(x). \end{aligned}$$

Therefore, $H_P(x) = F_P(x-1)$ and

$$h_k(P) = \sum_{r=0}^{d}(-1)^{r-k} f_r(P)\binom{r}{k}.$$

In particular,

$$\begin{aligned} &h_0(P) = f_0(P) - f_1(P) + f_2(P) - \cdots + (-1)^d f_d(P), \\ &h_1(P) = f_1(P) - 2f_2(P) + 3f_3(P) - \cdots + (-1)^{d-1} d f_d(P), \\ &h_2(P) = f_2(P) - 3f_3(P) + 6f_4(P) - \cdots + (-1)^{d-2}\binom{d}{2} f_d(P), \\ &h_d(P) = f_d(P)\ (=1), \qquad h_{d-1}(P) = f_{d-1}(P) - d, \\ &h_{d-2}(P) = f_{d-2}(P) - (d-1)f_{d-1}(P) + \binom{d}{2}. \end{aligned}$$

For the simplex S_d, $h_k = 1$ for every k. The graph of S_d is the complete graph on $d+1$ vertices and for every generic objective function there will by precisely one vertex of degree k for $1 \leqslant k \leqslant d$. For the cube C_d , $h_k = \binom{d}{k}$. To see this consider the objective function ϕ which is the sum of the coordinates. (This is not a generic objective function but it is not constant on edges of the polytope and this is sufficient for our purposes.) The vertices of degree k are precisely those having $\phi(v) = k$ and there are $\binom{d}{k}$ such vertices.

Euler formula and the Dehn–Sommerville relations

For a generic linear objective function there is a unique maximal vertex and a unique minimal vertex. Therefore, $h_0(P) = h_d(P)$ and by the formulas above we obtain $f_0(P) - f_1(P) + f_2(P) - \cdots + (-1)^d f_d(P) = 1$, which is Euler Formula usually written:

$$f_0(P) - f_1(P) + f_2(P) - \cdots + (-1)^{d-1} f_{d-1}(P) = 1 - (-1)^d.$$

More generally, if ϕ is a generic linear objective function then so is $-\phi$. However, if v is a vertex of a simple polytope P and v has degree k w.r.t. ϕ then v has degree $d-k$ w.r.t. $-\phi$.

This gives the Dehn–Sommerville relations

$$h_k(P) = h_{d-k}(P).$$

The Dehn–Sommerville relations are the only linear equalities among face numbers of simple d-polytopes.

The cyclic polytopes

The *cyclic* d-polytope with n vertices $C(d,n)$ is the convex hull of n distinct point on the *moment curve* $x(t) = (t, t^2, \ldots, t^d) \subset R^d$. This is a remarkable class of polytopes and the reader should consult [10,26,32] for their properties. $C^*(d,n)$ will denote a polar polytope to $C(d,n)$. (For the definition of polarity see [10,26,32].) $C^*(d,n)$ is a simple d-polytope with n facets.

The upper bound theorem

Motzkin conjectured that the maximal number of vertices (and more generally of k-dimensional faces) for d-polytopes with n facets is attained by $C^*(d,n)$, the polar-to-cyclic d-polytopes with n facets. This conjecture was proved by McMullen [23]. It is easy to reduce this conjecture to simple polytopes, and to calculate the h-numbers of $C^*(d,n)$, see [32,26]. This gives

$$h_k(C^*(d,n)) = h_{d-k}(C^*(d,n)) = \binom{n-d+k-1}{k},$$

for $1 \leqslant k \leqslant [d/2]$.

Since the face numbers are linear combination of h numbers with *nonnegative* coefficients the upper bound theorem follows from the following relations (and the Dehn–Somerville relations):

$$h_{d-k}(P) \leqslant \binom{n-d+k-1}{k}, \quad 1 \leqslant k \leqslant [d/2].$$

Proof. Consider a generic linear objective function ϕ which gives higher values to vertices in a facet F than to all other vertices. (To construct such an objective function start with an objective function whose maximum is attained precisely on the facet F and then make a slight perturbation to make it generic.) Every vertex v of degree $k-1$ in F has precisely one neighbor not in F and therefore the degree of v in P is k. This gives

$$(*) \qquad h_{k-1}(F) \leqslant h_k(P).$$

Next,

$$(**) \qquad \sum h_k(F) = (k+1)h_{k+1}(P) + (d-k)h_k(P),$$

where the sum is over all facets F of P.

To prove $(**)$ consider a vertex v of degree k in P. The vertex v is adjacent to d edges and every subset of $d-1$ out of them determine a facet. The degree of v is $k-1$ in every facet determined by $d-1$ edges adjacent to v where one of the k edges pointing down (w.r.t. ϕ) is deleted and there are k such facets. The degree of v is k in the remaining $d-k$ facets.

$(*)$ and $(**)$ give the upper bound relations

$$h_{d-k}(P) \leqslant \binom{n-d+k-1}{k}$$

by induction on k. For $k=1$ we have equality $h_{d-1} = n-d$. For $k \geqslant 1$ we obtain

$$(d-k+1)h_{d-k+1}(P) + kh_{d-k}(P) = \sum h_{d-k}(F) \leqslant nh_{d-k+1}(P)$$

Therefore $kh_{d-k}(P) \leqslant (n-d+k-1)h_{d-k+1}(P)$, i.e. $h_{d-k}(P) \leqslant [(n-d+k-1)/k]h_{d-k+1}(P)$. Further, assuming the upper bound relation for $k-1$ we obtain for k

$$h_{d-k}(P) \leqslant \frac{n-d+k-1}{k}\binom{n-d+k}{k-1} = \binom{n-d+k-1}{k}. \qquad \square$$

Abstract objective functions and telling the polytope from its graph

Consider an ordering $\prec$ of the vertices of a simple d-polytope P. For a nonempty face F we say that a vertex v of F is a local maximum in F if v is larger w.r.t. the ordering

$\prec$ than all its neighboring vertices in F. An *abstract objective function (AOF)* of a simple d-polytope is an ordering which satisfies the basic property of linear objective functions:

- Every nonempty face F of P has a unique local maximum vertex.

If P is a simple d-polytope and $\prec$ is a linear ordering of the vertices we define, as before, the degree of a vertex v w.r.t. the ordering as the number of adjacent vertices to v that are smaller than v w.r.t $\prec$. Thus, the degree of a vertex is a nonnegative number between 0 and d. Let $h_k^{\prec}$ be the number of vertices of degree k. Finally, put $F(P)$ to be the total number of nonempty faces of P.

Claim 1.

$$\sum_{r=0}^{d} 2^k h_k^{\prec} \geqslant F(P),$$

and equality holds if and only if the ordering $\prec$ is an AOF.

Proof. Count pairs (F, v) where F is a non-empty face of P (of any dimension) and v is a vertex which is local maximum in F w.r.t. the ordering $\prec$. On the one hand every vertex v of degree k contributes precisely 2^k pairs (F, v) corresponding to all subsets of edges from v leading to smaller vertices w.r.t. $\prec$. Therefore the number of pairs is precisely $\sum_{r=0}^{d} 2^k h_k^{\prec}$. On the other hand, the number of such pairs is at least $F(P)$ (every face has at least one local maximum) and it is equal to $F(P)$ iff every face has exactly one local maximum i.e if the ordering is an AOF. $\square$

Claim 2. A connected k-regular subgraph H of $G(P)$ is the graph of a k-face, if and only if there is an AOF in which all vertices in H are smaller than all vertices not in H.

Proof. If H is the graph of a k-face F of P then consider a linear objective function ψ which attains its minimum precisely at the points in F. (By definition for every non-trivial face such a linear objective function exists.) Now perturb ψ a little to get a generic linear objective function ϕ in which all vertices of H have smaller values than all other vertices.

On the other hand if there is an AOF $\prec$ in which all vertices in H are smaller than all vertices not in H, consider the vertex v of H which is the largest w.r.t. $\prec$. There is a k-face F of P determined by the k edges in H adjacent to v and v is a local maximum in this face. Since the ordering is an AOF, v must be larger than all vertices of F hence the vertices of F are contained in H and the graph of F is a subgraph of H. But the only k-regular subgraph of a connected k-regular graph is the graph itself and therefore H is the graph of F. $\square$

Claims 1 and 2 provide a proof to a theorem of Blind and Mani [3].

Theorem 2.1. *The combinatorial structure of a simple polytope is determined by its graph.*

Indeed, claim 1 allows us to determine just from the graph all the orderings which are AOF's. Using this, claim 2 allows to determine which sets of vertices form the vertices of some k-dimensional face. Let us mention that the proof gives a very poor algorithm (exponential in the number of vertices) and it is an open problem to find better algorithms.

Further facts without such simple geometric proofs

One of the most important developments in the theory of convex polytopes is the complete description of h-vectors of simple d-polytopes, conjectured by McMullen and proved by Stanley and Billera and Lee. See [2,30,24].

A crucial part of this characterization is the following: For every simple d-polytope

$$h_1(P) \leqslant h_2(P) \leqslant \cdots \leqslant h_{\lfloor d/2 \rfloor}(P).$$

In words: the number of vertices of degree k is smaller or equal than the number of vertices of degree $k+1$, when $k \leqslant [d/2]$. It is a challenging problem to find a direct *geometrical* proof for this inequality. (The existing proofs have algebraic ingredients and are very difficult.)

The effect of a single random pivot step on the degree

One possible measure for the progress of a certain pivot rule of the simplex algorithm would be via the degree of the vertices. Unfortunately, it seems difficult to predict how the degrees of vertices will behave in a path of vertices given by some pivot rule.

Starting with a random vertex of a simple polytope it is possible to say what will be the effect on the degree of a *single* random pivot step. By a random pivot step we mean the following: Starting with a vertex v we choose at random one of the d neighboring vertices w. If $\phi(w) > \phi(v)$ we move to w and otherwise we stay at v.

The average degree $E_0(P)$ of vertices in a simple d-polytope (which is the expected degree of a random vertex) is by the Dehn–Sommerville relations $d/2$. The average degree $E_1(P)$ of a vertex of P obtained by a single random pivot step (as described above) starting from a random vertex v is: $1 + 2f_2(P)/f_1(P)$. For example, for the d-cube $E_1(P) = \frac{1}{2}d + \frac{1}{2}$. (Similar formulas exist if we choose at random an r-face containing v and move from v to its highest vertex.)

To prove the formula for $E_1(P)$ note that the probability that after one random pivot step we reach a (specific) vertex w of degree k is $[1/f_0(P)]2k/d$. Indeed, if we start at w (this occurs with probability $1/f_0(P)$) then with probability k/d we stay at w. If we start with one of the k "lower" neighbors of w (altogether this occur with probability $k/f_0(P)$) then we reach w after one step with probability $1/d$. It follows that

$$E_1(P) = \frac{1}{f_0(P)} \sum_{k=0}^{d} (2k^2/d) h_k(P),$$

which equals $1 + 2f_2(P)/f_1(P)$ by the formulas above. Note that $E_1(P)$ does not depend on the objective function. This is no longer true if we are interested in $E_2(P)$ the average degree after two random pivot steps. The following problem (of independent interest) naturally arises.

Problem. Let P be a simple d-polytope and ϕ be a generic linear objective function. Let $h_{i,j}$ be the numbers of pairs of adjacent vertices v, w such that $\phi(v) < \phi(w)$ and $\deg(v) = i$, $\deg(w) = j$. What can be said about the collection of numbers $(h_{i,j} : 1 \leqslant i, j \leqslant d)$.

This array of numbers depends on the objective function and not only on the polytope. It will be interesting to describe the possible $h_{i,j}$ numbers even for the special case when the polytope is combinatorially isomorphic to the d-dimensional cube. (The question is interesting also for abstract objective functions.)

Arrangements

We would like to close this section with the following remark: Consider an arrangement of n hyperplanes in general position in R^d, and a generic linear objective function ϕ. This arrangement divides R^d into simple d-polyhedra. The average value of $h_k(P)$ over all these polyhedra is $\binom{d}{k}$. To see this just note that every vertex v in the arrangement belongs to 2^d d-polyhedra and has degree k in $\binom{d}{k}$ of these polyhedra. Similarly, the average h-vector over r-dimensional faces of the arrangement is the h-vector of the r-dimensional cube.

3. The Hirsch conjecture and subexponential randomized pivot rules for the simplex algorithm

In this section we describe recent developments concerning the simplex algorithm. We describe subexponential randomized pivot rules and recent upper bounds for the diameter of graphs of polytopes. The algorithms we consider should be regarded in the general context of LP algorithms discovered by Megiddo [25], Clarkson [5], Seidel [28], Dyer, Dyer and Frieze [7] and many others. But we will not attempt giving this general picture here. For the use of randomized algorithms in computational geometry the reader is referred to Mulmuly's book [26]. Another word of warning is that the language we use is quite different than the usual LP terminology, and we leave it to the interested reader to make the translation.

The complexity of linear programming

Given a linear program $\max\langle b, x\rangle$ subject to $Ax \leqslant c$ with n inequalities in d variables, we denote by L the total input size of the problem when the coefficients are described in binary. We denote by $C_A(d, n, L)$ the number of arithmetic operations needed – *in the worst case* – by an algorithm A to solve a linear programming problem with d variables,

n inequalities and input size L. The (worst-case) complexity of linear programming is (roughly) the function $C(d, n, L)$ which describes for every value of d, n, L the smallest possible value of $C_A(d, n, L)$ over all possible algorithms.

Khachiyan's breakthrough result [12] was that the complexity of the ellipsoid method E is a polynomial function of d, n and L, namely $C_E(d, n, L) \leqslant p(d, n)L$. Other algorithms which improve on Khachiyan's original bound (and also had immense practical impact on the subject) were found by Karmarkar and many others.

By considering solutions to all subsets of d from the n inequalities we can easily see that $C(d, n, L) \leqslant f(d, n)$, i.e., linear programming can be solved by a number of arithmetic operations which is a function of d and n and independent of the input size L. It is an outstanding open problem to find a *strongly polynomial algorithm* for linear programming; that is to find an algorithm which requires a polynomial number in d and n of arithmetic operations which is independent from L. Denote $C(d, n) = \max_L C(d, n, L)$.

Klee and Minty [18] and subsequently others have shown that several common pivot rules for the simplex algorithm are exponential in the worst case.

Explaining the excellent performance of the simplex algorithm in practice (especially in view of the exponential worst-case behavior of various pivot rules) is a major challenge. The results on the average case behavior of the simplex algorithm provide one such explanation. (See Borgwardt's book [4] for a description of his work and for references to other works, or [29].) The fact that the complexity of linear programming is polynomial (by Khachiyan's result) even if not via the simplex algorithm provides another partial explanation.

Of course, finding a pivot rule which requires a polynomial number of steps in the worst case or even proving that there are always a polynomial number of pivot steps leading to the optimal vertex (without prescribing an algorithm to find these steps) are very desirable.

Using randomness for pivot rules

We will consider now randomized algorithms. Namely, algorithms which depend on internal random choices. Given such a randomized algorithm A we denote by $C_A^R(d, n)$ the *expected* number of arithmetic operation needed – *in the worst case* – by A on a LP-problem with d variables and n inequalities. $C^R(d, n)$ will be the minimal value of $C_A^R(d, n)$ over all possible algorithms A. Clearly $C^R(d, n) \leqslant C(d, n)$. (Note: we are interested in a *worst case analysis* of the average running time where the randomization is internal to the algorithm. This is in contrast with average case analysis where the LP problem itself is random.)

Perhaps the simplest random pivot rule is to choose at each step at random with equal probabilities a neighboring vertex with a higher value of the objective function. Unfortunately, it seems very difficult to analyze this rule for general problems. Recently, Gärtner, Henk and Ziegler [9] managed to analyze the behavior of random pivoting on the "Klee–Minty cube".

The Hirsch conjecture

Let $\Delta(d,n)$ denote the maximal diameter of the graphs of d-polyhedra P with n facets and let $\Delta_b(d,n)$ denote the maximal diameter of the graphs of d-polytopes with n vertices.

Given a d-polyhedron P, a linear objective function ϕ which is bounded from above on P and a vertex v of P, denote by $m(v)$ the minimal length of a *monotone* path in $G(P)$ from v to a vertex of P on which ϕ attains its maximum. Let $H(d,n)$ be the maximum of $m(v)$ over all d-polyhedra P with n facets, all linear functionals ϕ on R^d and all vertices v of P. (A monotone path is a path in $G(P)$ on which ϕ is increasing.)

Let $M(d,n)$ be the maximal number of vertices in a monotone path in $G(P)$ over all d-polyhedra P with n facets and all linear functionals ϕ on R^d.

Clearly

$$\Delta(d,n) \leqslant H(d,n) \leqslant M(d,n).$$

$H(d,n)$ can be regarded as the number of pivot steps needed by the simplex algorithm when the pivots are chosen by an oracle in the best possible way. $M(d,n)$ can be regarded as the number of pivot steps needed when the pivots are chosen by an adversary in the worst possible way.

In 1957 Hirsch conjectured [6] that $\Delta(d,n) \leqslant n-d$. Klee and Walkup showed that the Hirsch conjecture is false for unbounded polyhedra. The Hirsch conjecture for polytopes is still open. The special case asserting that $\Delta_b(d,2d)=d$ is called the d-step conjecture and it was shown by Klee and Walkup to imply the general case.

Theorem 3.1 (Klee and Walkup [19], 1967).

$$\Delta(d,n) \geqslant n-d+\min\{[d/4],[(n-d)/4]\}.$$

Theorem 3.2 (Holt and Klee [11], 1997). *For all $d \geqslant 14$ and $n > d$*

$$\Delta_b(d,n) \geqslant n-d.$$

Theorem 3.3 (Larman [20], 1970).

$$\Delta(d,n) \leqslant n2^{d-3}.$$

Theorem 3.4 (Kalai and Kleitman [17], 1992).

$$\Delta(d,n) \leqslant n\binom{\log n + d}{\log n} \leqslant n^{\log d+1}.$$

Klee and Minty [18] considered a certain geometric realization of the d-cube (called now the "Klee–Minty cube") to show that

Theorem 3.5 (Klee and Minty [18], 1972). $M(d,2d) \geqslant 2^d$.

Subexponential randomized pivot rules

We will assume (and there is no loss of generality assuming this) that the LP problem is non-degenerate (i.e. the feasible polyhedron is simple) and that a vertex v of the feasible polyhedron is given. With a slight change of terminology all the algorithms and results we describe apply to the degenerate case.

Several years ago the author [16] and independently Matoušek, Sharir and Welzl [22] found a randomized subexponential pivot rule for LP, thus proving that

$C^R(d,n) \leqslant \exp(K\sqrt{d \log n})$.

(Slightly sharper bounds are described below.) In my paper various variants of the algorithm were presented and we will see here two variants. The first and simplest variant is one of my originals and is equivalent (in a dual-setting) to the Sharir-Welzl algorithm [27] on which [22] is based. The second variant presented here is a joint work with Martin Dyer and Nimrod Megiddo. It is a better and simplified version of other variants from [16]. All these algorithms apply to abstract objective functions and even more general settings. See also Gärtner's paper [8].

Consider an LP problem of optimizing a linear objective function ϕ over a d-polyhedron P and a vertex v of P. Our aim is to reach top(P) which is a vertex of P on which the objective function is maximal or an edge of P on which the objective function is unbounded from above.

Algorithm I.

- Given a vertex $v \in P$ choose a facet F containing v at random.
- Apply the algorithm on F until reaching $w = top(F)$.
- repeat the algorithm from w.

Remark. The algorithm terminates if v = top(P). If v = top(F') for some facet F' containing v (in which case v has only one improving edge) we choose F at random from the other $d-1$ facets containing F. (Unless v = top(P) there is at most one such facet F'.)

Algorithm II. Choose at random an ordering of the facets $F_{\pi(1)}, F_{\pi(2)}, \dots, F_{\pi(n)}$.

- Phase I: Apply the algorithm until you reach a vertex in $F_{\pi(1)}$ (or reach $top(P)$).
- Phase II: Apply the algorithm recursively inside $F_{\pi(1)}$ until reaching w = top$(F_{\pi(1)})$
- Phase III: Delete the facet $F_{\pi(1)}$ from the ordering and continue to run the algorithm from w.

Phase I and phase III are performed w.r.t. the initial random ordering of the n inequalities but in phase II you have to find again a new random ordering of the facets.

Analysis of the rules

We say that a facet F of P is *active* w.r.t. the vertex v if $\phi(v) < \max\{\phi(x) : x \in F\}$. We will study the number of pivot steps as a function of the number of variables d and the number of active facets n. The number of pivot steps will not depend on the total number of facets N. However, we do not assume that we know while running the algorithms which facets are active and the number of arithmetic operations per pivot step depends therefore (polynomially) also on N. Note that in Algorithm II only the ordering of the active facets matters.

For a linear programming problem U with d variables and N inequalities and a feasible vertex v for U such that there are n active facets w.r.t. v, we denote by $f(U, v)$ the expected number of pivot steps needed by algorithm I on the problem U starting with the vertex v. $f(d, n)$ denotes the maximal value of $f(U, v)$ over all problems U and vertices v. The function $f(d, n)$ is not decreasing with n. Similarly, $g(d, n)$ will be the average number of pivot steps in the worst case problem for Algorithm II.

Analysis of Algorithm I. We start with a situation where there are n active facets. Let $F_1, F_2, \ldots, F_d$ be the facets containing v, ordered such that $\phi(\mathrm{top}(F_1)) \leqslant \phi((\mathrm{top}(F_2))$ $\leqslant \cdots \leqslant \phi(top(F_d))$ Note that (unless $v = \mathrm{top}(P)$) at most one (namely only F_1) of these facets can be non-active. The average number of steps needed to reach $\mathrm{top}(F)$ from v is at most $f(d-1, n-1)$.

If F_1 is active then with probability $1/d$ the chosen random facet F equals F_i for $i = 1, 2, \ldots, d$ and then after reaching $w = \mathrm{top}(F)$ there are at most $n - i$ active facets remaining and the average number of steps needed to reach $\mathrm{top}(P)$ from w is at most $f(d, n-i+1)$. Averaging over i we get that the average number of steps needed to reach $\mathrm{top}(P)$ from w is at most $1/d \sum_{i=1}^{d} f(d, n-i)$.

If F_1 is not active then $F = F_i$ with probability $1/(d-1)$ for $i = 2, 3, \ldots, d$, and by the same token the average number of steps needed to reach $\mathrm{top}(P)$ from w is at most $1/(d-1) \sum_{i=1}^{d-1} f(d, n-i)$. This is (slightly) higher than the previous expression by the monotonicity of $f(d, n)$ as a function of n. In sum,

$$f(d,n) \leqslant f(d-1, n-1) + \frac{1}{d-1} \sum_{i=1}^{d-1} f(d, n-i).$$

This gives $f(d, n) \leqslant \exp(K\sqrt{n \log d})$, see [22].

Analysis of Algorithm II. For phase II we need at most $g(d-1, n-1)$ steps on the average. For phase III we can repeat the argument of the previous algorithm. With probability $1/n$ there are (at most) $n - i$ active facets left after reaching $\mathrm{top}(F_{\pi(1)})$, for $i = 1, 2, \ldots, n$. So the average number of pivot step for this phase is at most $(1/n) \sum_{i=1}^{n} g(d-1, i)$. We claim now that the average number of pivot steps for phase I is also at most $(1/n) \sum_{i=1}^{n} g(d-1, i)$.

To see this note:

- As long as we run the algorithm from v meeting only vertices in r active facets we can regard ourself running the algorithm from v in the LP-problem obtained by deleting the inequalities corresponding to the other active facets. This LP problem has only r active facets. Since the average number of pivot steps needed for this problem is at most $g(d,r)$ we conclude that after an average number of $g(d,r)$ pivot steps we either reach top(P) or reach vertices in more than r active facets.
- The pivot steps taken running the algorithm while meeting vertices on r active facets do not depend on the ordering of the remaining active facets. Therefore the identity of the active facet to be the next we meet (which is a probability distribution on the remaining active facets) does not depend on the ordering of the remaining $n-r$ active facets.

It follows that with probability $1/n$ the facets $F_{\pi(1)}$ will be the ith active facet to be met, $i = 1,2,\ldots,n$.

So we get,

$$g(d,n) \leqslant g(d-1,n-1) + \frac{2}{n}\sum_{i=1}^{n} g(d,n-i).$$

This relation implies the following:

(1) $g(d,n) \leqslant \exp(K\sqrt{d\log n})$

(2) If d and n are comparable we get a better estimate: $g(d,Bd) \leqslant \exp[K(B)\sqrt{d}]$. [$K(B)$ is a constant depending on B.]

(3) The following estimates are useful when $t = n-d$ is small w.r.t. n,

$$g(d,d+t) \leqslant K\left(\frac{1+\epsilon}{\epsilon}\right)^{t} d^{\epsilon}, \quad g(d,d+t) \leqslant K(\log d)^{t-1}.$$

These bounds apply to $f(d,d+t)$ as well.

(4) The following estimates are useful when d is small w.r.t. n:

$$g(d,n) \leqslant K\left(\frac{2+\epsilon}{\epsilon}\right)^{d} n^{1+\epsilon},$$

for every $\epsilon > 0$ and $g(d,n) \leqslant K(\log n)^{d-1}n$.

It is possible to use generating function techniques to get a precise asymptotic for $f(d,n)$ and $g(d,n)$. It follows from the recursion that $n!g(d,n)$ is bounded above by $b(d,n)$ – the number of permutations of $\{1,2,\ldots,n\}$ such that each cycle in the permutation (considered as a product of disjoint cycles) is decorated by a nonnegative integer and by a plus or minus sign such that the sum of the integers is d. For $b(d,n)$ there is the closed formula

$$b(d,n) = \sum 2^k\, c(n,k)\binom{d+k-1}{k-1},$$

where $c(n,k)$ is the number of permutations of $\{1,2,\ldots,n\}$ with k cycles. ($c(n,k)$ is the absolute value of the Stirling number of the first kind.) However, for the asymptotic facts described above (without getting the precise constants), the simplest proofs are by direct estimations.

Remark. Matoušek [21] found remarkable classes of abstract objective functions on the d-dimensional cube for which the expected number of pivot steps for Algorithm I described above is indeed $\exp(C\sqrt{d})$. Further understanding of similar examples may shed light on some of the problems described in this section.

LP duality

LP duality allows us to move from a problem with d variables and n inequalities to the dual problem with $n-d$ variables and n inequalities. Note that the running time of the algorithms as well as the bounds on the diameter are *not* invariant under LP duality. The upper bounds on $\Delta(d,n)$ as well as on the running time for the algorithms described here agree with the common wisdom that when n is large w.r.t. d it is better to move to the dual problem. However, note that the average number of pivot steps of Algorithm II is rather small (close to linear) even when d is fixed and n tends to infinity.

It is an interesting problem to study the relations between the combinatorics (e.g. the face numbers, h-numbers etc.) of the feasible polyhedra for an LP problem and for its dual.

Non-deterministic analysis of the rule and application to the Hirsch problem

Now let us consider again Algorithm II but this time let us assume that the random choices are made by a friendly oracle, and that we can instruct the oracle to make as good as possible choices. Studying non-deterministic performance of randomized algorithms is important for understanding the algorithm, but in this case this is of particular importance since it is immediately related to the Hirsch problem discussed above.

First we order the active facets $F_1,\dots,F_n$ so that $\phi(\text{top}(F_1)) \leqslant \phi((\text{top}(F_2)) \leqslant \dots \leqslant \phi(\text{top}(F_n))$. The instructions for the oracle are as follows: the only condition on the first active facet $F = F_{\pi(1)}$ is that top(F) is above the median. So when you run the algorithm you declare the first facet F you reach with top(F) above the median as $F_{\pi(1)}$. Of course, this instruction applies recursively for the first and last stages as well as when you run the algorithm inside $F_{\pi(1)}$.

Let $h(d,n)$ is the number of pivot steps made with the help of our friendly oracle instructed above. in phase I we need at most $h(d,n/2)$ steps. Indeed, as long as we met vertices only on m active facets we can consider ourselves as running the algorithm in the polyhedra where the inequalities correspond to the other active facets are deleted and the number of pivot steps is at most $h(d,m)$. So in $1+h(d,n/2)$ pivot steps we must reach (either top(P) or) vertices in more than $n/2$ active facets and hence we must reach a vertex in a facet F with top(F) above the median. When we reach top(F)

then the number of remaining active facets is smaller than $d/2$. Therefore also in step III we need at most $h(d,n/2)$ steps. Thus,

$$h(d,n) \leqslant 2h(d,n/2) + h(d-1,n-1) + 1.$$

This recursion gives

$$h(d,n) \leqslant n\binom{\log n + d}{\log n} \leqslant n^{\log d+1}.$$

How clever should the oracle be? Not so much! The oracle should be able to run LP problems and this is polynomial via Khachiyan. By a well-known result of Tardos [31] we do not even need to consider the objective function in the input size. So we get

Theorem 3.6. *Let P be a d-polytope with n facets described by a system of inequalities with input size L. Let v be a vertex of P and ϕ be a linear objective function. Then there is an algorithm which finds in $T \leqslant h(d,n)$ steps a monotone path of length T from v to* $\mathrm{top}(P)$ *and each step is performed by a polynomial number (in d, n and L) of arithmetic operations.*

Conclusion

The situation concerning the Hirsch conjecture and the (worst-case) complexity of the simplex algorithm is rather frustrating. We are short of polynomial bounds for the diameter, and despite the simplicity of the proofs for the known bounds we cannot push them any further. For $n = 2d$ we cannot find a randomized pivot rule which will require $\exp(d^{1/2-\epsilon})$ pivot steps for some $\epsilon > 0$, even if the feasible polytope is combinatorially equivalent to a d-dimensional cube. And we cannot find a deterministic pivot rule (without randomization) which is not exponential. We leave these tasks for you the reader.

References

[1] N. Amenta and G. Ziegler, Deformed products and maximal shadows, Preprint 502/1996, TU Berlin, to appear.
[2] L.J. Billera and C.W. Lee, A proof of the sufficiency of McMullen's conditions for f-vectors of simplicial convex polytopes, *J. Combin. Theory, Series A* 31 (1981) 237–255.
[3] R. Blind and P. Mani, On puzzles and polytope isomorphism, *Aequationes Math.* 34 (1987) 287–297.
[4] K. H. Borgwardt, *The Simplex Method, A Probabilistic Analysis, Algorithms and Combinatorics*, Vol 1 (Springer, Berlin, 1987).
[5] K.L. Clarkson, A Las Vegas Algorithm for linear programming when the dimension is small, *J. ACM* 42 (2) (1995) 488–499.
[6] G.B. Dantzig, *Linear Programming and Extensions* (Princeton University Press, Princeton, NJ, 1963).
[7] M.E. Dyer and A. Frieze, Random walks, totally unimodular matrices and a randomized dual simplex algorithm, *Mathematical Programming* 64 (1994) 1–16.
[8] B. Gärtner, A subexponential algorithm for abstract optimization problems, *SIAM J. Comput.* 24 (1995) 1018–1035
[9] B. Gärtner, M. Henk and G. Ziegler, Randomized simplex algorithms on Klee–Minty cubes, to appear.

[10] B. Grünbaum, *Convex Polytopes* (Interscience, London, 1967).
[11] F. Holt and V. Klee, Many polytopes meeting the conjectured Hirsch bound, (1997), to appear.
[12] L.G. Khachiyan, A polynomial algorithm in linear programming, *Soviet Math. Doklady* 20 (1979) 191-194.
[13] G. Kalai, A simple way to tell a simple polytope from its graph, J. Combin. Theory, Series A 49 (1988) 381-383.
[14] G. Kalai, The diameter of graphs of convex polytopes and f-vector theory, in: *Applied Geometry and Discrete Mathematics, The Klee Festschrift*, DIMACS Series in Discrete Mathematics and Computer Science, Vol. 4 (1991) 387-411.
[15] G. Kalai, Upper bounds for the diameter of graphs of convex polytopes, *Discrete Computational Geometry* 8 (1992) 363-372.
[16] G. Kalai, A subexponential randomized simplex algorithm, in: *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing* (ACM Press, Victoria, 1992) 475-482.
[17] G. Kalai and D. J. Kleitman, A quasi-polynomial bound for diameter of graphs of polyhedra, *Bull. Amer. Math. Soc.* 26 (1992) 315-316.
[18] V. Klee and G.J. Minty, How good is the simplex algorithm, in: O. Shisha, ed., *Inequalities III* (Academic Press, New York, 1972) 159-175.
[19] V. Klee and D. Walkup, The d-step conjecture for polyhedra of dimension $d < 6$, *Acta Math.* 133, pp. 53-78.
[20] D.G. Larman, Paths on polytopes, *Proc. London Math. Soc.* 20 (1970) 161-178.
[21] J. Matoušek, Lower bounds for a subexponential optimization algorithm, *Random Structures and Algorithms* 5 (1994) 591-607.
[22] J. Matoušek, M. Sharir and E. Welzl, A subexponential bound for linear programming, in: *Proceedings of the 8th Annual Symposium on Computational Geometry* (1992) 1-8.
[23] P. McMullen, The maximal number of faces of a convex polytope, *Mathematika* 17 (1970) 179-184.
[24] P. McMullen, On simple polytopes, *Inven. Math.* 113 (1993) 419-444.
[25] N. Megiddo, Linear programming in linear time when the dimension is fixed, *J. ACM* 31 (1984) 114-127.
[26] K. Mulmuley, *Computational Geometry. An Introduction Through Randomized Algorithms* (Prentice-Hall, Englewoods Cliffs, NJ, 1994).
[27] M. Sharir and E. Welzl, A Combinatorial bound for linear programming and related problems, *Proceedings of the 9th Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, Vol. 577 (Springer, Berlin, 1992) 569-579.
[28] R. Seidel, Small-dimensional linear programming and convex hulls made easy, *Discrete Computational Geometry* 6 (1991) 423-434.
[29] A. Schrijver, *Theory of Linear and Integer Programming* (Wiley-Interscience, New York, 1986).
[30] R. Stanley, The number of faces of simplicial convex polytopes, it Adv. Math. 35 (1980) 236-238.
[31] E. Tardos, A strongly polynomial algorithm to solve combinatorial linear programs, *Operations Research* 34 (1986) 250-256.
[32] G.M. Ziegler, *Lectures on Polytopes*, Graduate Texts in Mathematics, Vol. 152 (Springer, New York, 1995).