Typical and Extremal Linear Programs*

*Günter M. Ziegler***

Abstract. Viewed geometrically, the simplex algorithm on a (primally and dually non-degenerate) linear program traces a monotone edge path from the starting vertex to the (unique) optimum. *Which* path it takes depends on the pivot rule.
In this paper we survey geometric and combinatorial aspects of the situation: How do "real" linear programs and their polyhedra look like? How long can simplex paths be in the worst case? Do short paths always exist? Can we expect randomized pivot rules (such as `Random_Edge`) or deterministic rules (such as Zadeh's rule) to find short paths?

MSC 2000. 90C05, 52B11
Key words. Geometry of linear programs, polytopes and polyhedra, degeneracy, 2-dimensional shadows, long paths, deformed products, Hirsch conjecture, short paths, deterministic and randomized pivot rules.

1 Introduction

What can geometry contribute to the study and understanding of linear programs and of the simplex algorithm? This little survey attempts to sketch a variety of answers to this question: We want to show that geometry and geometric insights can contribute both to the understanding of "real" linear programs and to the construction of "extremal" examples on which (certain variants of) the simplex algorithm would or should "behave badly." For this, we will not be so naive to assume that the nice, symmetrical and "interesting" examples of polytopes that lie at the core of modern polytope theory such as the permuto-associahedron displayed in Figure 1 have much to do in their combinatorics and geometry with the polytopes and polyhedra that arise as the feasible regions of linear programs "in practice." Indeed, we have to admit that of course we do not know how a "real" linear program

*Partially supported by DFG
**Inst. Mathematics, MA 6-2, TU Berlin, 10623 Berlin, Germany; ziegler@math.tu-berlin.de

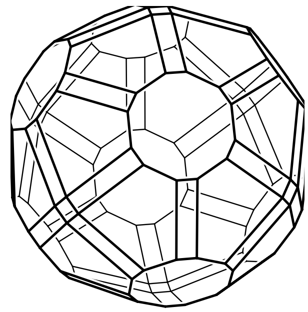


Figure 1. *A nice polytope [30].*

in, say, $d = 1000$ variables and with $n = 5000$ constraints "looks like." (This is, if you ask for a geometric impression rather than just looking at the input file.)

But still this may be a reasonable question: Take for an example the smallest entry of the netlib collection, the `afiro` linear program, as displayed in Figure 2. What can we say about the geometry of such a program? Well, we can say a lot. Before we start the discussion, let us just fix a bit of notation, and — more importantly — match a few terms that have become standard in linear programming and in polytope theory, respectively. So, for the following d will be used for the dimension of a problem; in geometry this will be the dimension of a polytope, in linear programming this will be the number of (free) variables. Similarly, n will denote the number of facets of a polytope, corresponding to the number of (essential) inequalities for a linear program. Standard arguments (linear algebra, perturbation) allow one — at least for the analysis of extremal examples — to assume that the linear program under consideration is *primally nondegenerate* (that is, the pohyhedron is *simple*), that the *objective function* is given by the last variable x_d (and thus we are trying to maximize the *height* of a point in the polyhedron), that the program is *dually nondegenerate* (that is, the polyhedron has *no horizontal edges*), and finally that the feasible region is *bounded* (and hence we are looking at a *polytope* rather than a *polyhedron*).

2 Real Linear Programs

What do "real" linear programs "look like," and how can we analyze and picture them? In this section, we will sketch two approaches to this question. The first one is based on the fact that computational polytope theory (cf. [16]) has made enormous progress since the sixties, so that now we can compute and analyze the full polyhedron at least for small linear programs, and try to understand it. The second approach is based on the shadow boundary algorithm, which can be (ab)used to compute 2-dimensional pictures (projections) of linear programs.

The shadow boundary algorithm for linear programming starts at a feasible vertex of the polyhedron, for which it is easy to construct a linear function $c^t x$ that is optimal at the given vertex, and then traces the sequence of optimal vertices

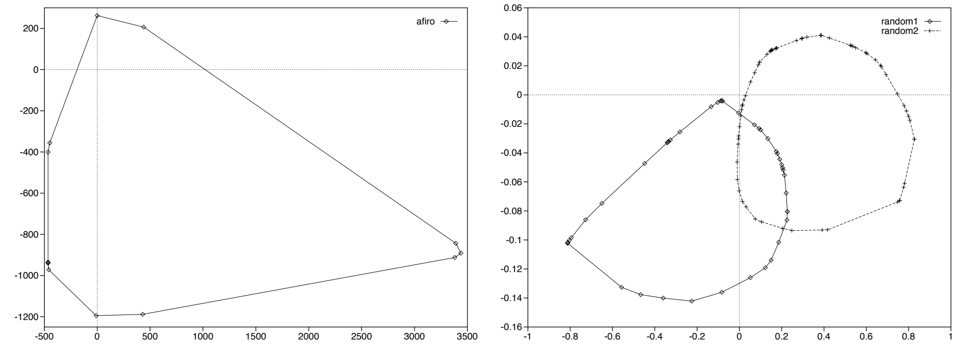
```
/* objective function: */
min: + 10 X39 + -0.48 X36 + -0.6 X23 + -0.32 X14 + -0.4 X02 ;

/* constraints */
X51 : + 1 X38 + 1 X16 <= 300;
X50 : + 1 X26 + 1 X04 <= 310;
X49 : + -1 X37 + 0.326 X09 + 0.313 X08 + 0.313 X07 + 0.301 X06 <= 0;
X48 : + -1 X24 + 0.301 X01 <= 0;
X47 : + 0.107 X31 + 0.108 X30 + 0.108 X29 + 0.109 X28 + -1 X15 <= 0;
X46 : + 0.109 X22 + -1 X03 <= 0;
X45 : + 2.279 X35 + 2.249 X34 + 2.219 X33 + 2.191 X32 + -1 X25 + 2.429 X13\\
                         + 2.408 X12 + 2.386 X11 + 2.364 X10 <= 0;
X43 : + -1 X35 + 1 X31 <= 0;
X42 : + -1 X34 + 1 X30 <= 0;
X41 : + -1 X33 + 1 X29 <= 0;
X40 : + -1 X32 + 1 X28 <= 500;
R23 : + 1 X39 + 1 X37 + -1 X36 + 1 X31 + 1 X30 + 1 X29 + 1 X28 = 44;
R22 : + 1 X38 + -0.37 X31 + -0.39 X30 + -0.43 X29 + -0.43 X28 = 0;
X44 : + 1.4 X36 + -1 X23 <= 0;
X27 : + 1 X22 <= 500;
R20 : + 1 X26 + -0.43 X22 = 0;
R19 : + 1 X25 + 1 X24 + 1 X23 + -1 X22 = 0;
X20 : + -1 X13 + 1 X09 <= 0;
X19 : + -1 X12 + 1 X08 <= 0;
X18 : + -1 X11 + 1 X07 <= 0;
X17 : + -1 X10 + 1 X06 <= 80;
R13 : + 1 X16 + -0.86 X09 + -0.96 X08 + -1.06 X07 + -1.06 X06 = 0;
R12 : + 1 X15 + 1 X14 + -1 X09 + -1 X08 + -1 X07 + -1 X06 = 0;
X21 : + 1.4 X14 + -1 X02 <= 0;
X05 : + 1 X01 <= 80;
R10 : + 1 X04 + -1.06 X01 = 0;
R09 : + 1 X03 + 1 X02 + -1 X01 = 0;%
```

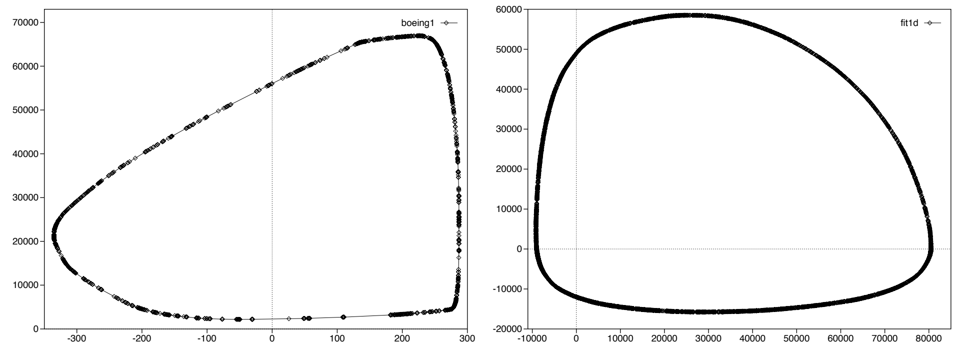
Figure 2. *The* `afiro` *linear program.*

that appear while the linear objective function is linearly interpolated between the starting objective function $c^t x$ and the final objective function $d^t x$. Geometrically, this procedure computes a part of the boundary of a 2-dimensional projection of the feasible region of the linear program. If we continue the procedure, by next interpolating between $d^t x$ and $-c^t x$, then between $-c^t x$ and $-d^t x$, and finally between $-d^t x$ and $c^t x$, we compute the full 2-dimensional projection. Thus we obtain *pictures*. This idea was developed, implemented and tested in the Diplomarbeit of S. Fischer [4], who produced interesting and somewhat surprising pictures of the polyhedra of linear programs in the netlib library. The picture gallery of Figure 3 of rather typical examples is supposed to illustrate that some linear programs have rather sharp angles (in their 2-dimensional projections) while others appear to be rather round (with many vertices very close to each other in a 2-dimensional projection). Here the two directions of projections are typically taken to be the given objective function and the first or last variable.

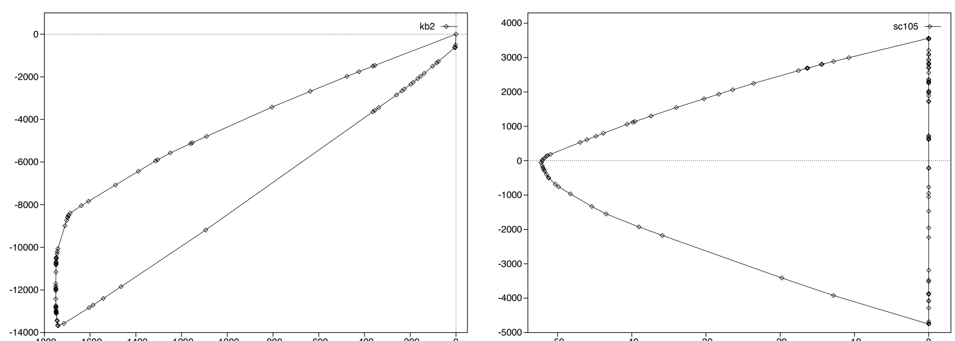
On the other hand, for “small” linear programs such as `afiro` a complete analysis of the linear program is possible. The `Polymake`-system of Gawrilow and Joswig [8, 9] is a tool that for a linear program/polyhedron given by a list of



The afiro problem … and two random projections of the afiro LP



The boeing1 problem and the fit1d problem have rather “round” projections



while kb2 and … sc105 display distinctive “corners.”

Figure 3. *Six shadows of linear programs (from [4])*

inequalities would (attempt to) run a convex hull algorithm to determine the list of vertices, then produce all kinds of combinatorial information “asked for,” such as the number of vertices, the graph, the graph diameter, etc. This was carried out in the Diplomarbeit of D. Weber [34, pp. 20/21].

For example, the afiro-problem has 32 variables, but it includes 8 equations, so the dimension of the polytope is $d = 24$. It has 19 explicit inequality constraints, plus the 32 nonnegativities, but many constraints are redundant. Indeed, the polytope has only $n = 29$ facets. It has 1654 vertices, of which only 78 are degenerate. Thus, the minimal degree is 24, the maximal degree is 39. (These degeneracy data are, of course, only observed if we interpret the coefficients of the `afiro` problem as rational numbers.) The average vertex degree is 24.71, not much more than the minimal degree. The linear program has 20433 edges, of which 11718 are horizontal: This is more than half of them, perhaps the first big surprise! With respect to the given objective function, the linear program has a unique maximal vertex, which has the maximal degree 39; on the other side, there are 4 minimal vertices for the given objective function, all of which are simple (of degree 24). The minimal vertices describe a face of dimension 2 (a quadrilateral). The (graph-theoretic) distance between the minimal vertices to the maximal one is just 2, while the diameter of the polyhedron is 5. (Thus the problem satisfies the Hirsch bound, as discussed below, with equality!) What else do you want to know about this linear program? Chances are that your question can easily be answered by the `Polymake` system.

3 Long Paths

A lot of effort has been put into the goal to understand the "worst case" of the simplex algorithm. In particular, we are trying to resolve the central question:

The Complexity of Linear Programming. *Is there a strongly polynomial (simplex) algorithm for linear programming?*

A natural approach to resolve this question is to construct and understand "bad examples" of linear programs for (selected) variants of the simplex algorithm. Thus the development of bad examples and the understanding of pivot rules should, ideally, be closely connected. We would expect that bad examples show that certain pivot rules are not good; on the other hand, they should tell us how pivot rules have to be designed in order to escape the "bad examples." This program has been worked out only partially up to now. It has produced the "deformed product" examples of bad linear programs, which managed to fool all the *classical deterministic* pivot rules for linear programming into exponential behaviour.

The first and classical example of bad linear programs is given by the Klee-Minty cubes [21]. These are deformed d-dimensional cubes for which there is a monotone path (that is, a path on which the objective function increases strictly) through all the 2^d vertices. The classical Dantzig pivot rule as well as various lexicographic rules can be made to be exponential on these examples.

In his linear programming book [29, p. 76], Manfred Padberg talks about what he calls "worstcasitis": Following Klee and Minty's initial breakthrough there was a whole flood of papers that produced bad/exponential examples for all kinds of pivot rules in linear programming. Although some of these constructions are quite ingenious, one gets the feeling that they all more or less work along the same lines, and produce the same type of bad examples, namely "deformed products." And indeed, a precise and systematic concept of "deformed products" was formalized by Amenta and Ziegler [1], for which the following is essentially true:

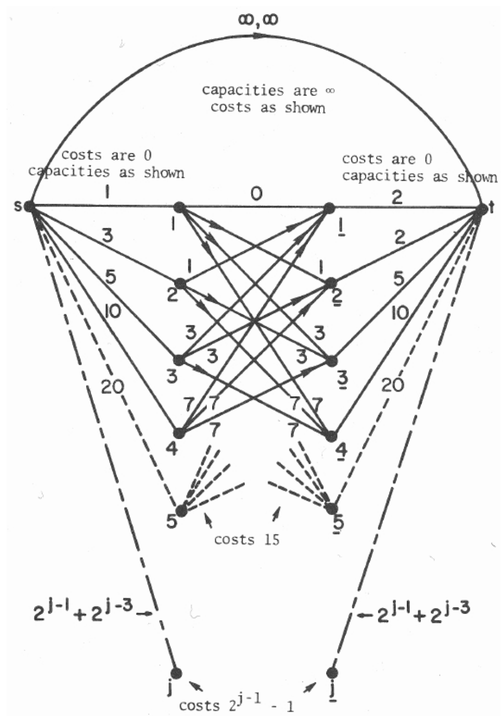


Figure 4. *The bad network problem from* [35]

"Theorem": *The Klee-Minty cubes and all other published bad examples of linear programs (that is, exponential examples for the simplex algorithm with various privot rules) can be constructed and analyzed as iterated deformed products.*

We refer to Amenta and Ziegler [1] for details of this construction, including many examples and pictures. Here we want to present just one additional example not shown in [1]. Namely, Zadeh [35] presented a sequence of min-cost flow problems N_k for which the network simplex algorithm produces an exponential number of steps. It may seem surprising that even these examples are iterated deformed products:

Theorem [35] [27]. *The polytope P_k of Zadeh's network N_k satisfies:*

- *The network N_k has $2k+2$ nodes and k^2+k+2 edges.*
- *The corresponding polytope P_k has dimension k^2-k+1 and k^2+k-1 facets.*
- *The polytope P_k is an iterated deformed product of $2k-2$ simplices:*

$$P_k \cong \Delta_k \times \Delta_{k-1} \times (\Delta_{k-2})^2 \times \ldots \times (\Delta_1)^2.$$

  Thus P_k has $(k+1)!(k-1)!$ vertices.
- *On this iterated deformed product, the network simplex algorithm (with the common "Path," "M-Path," "Primal Dual," and "Cycle" pivot rules) will trace a monotone path through $2^k+2^{k-2}-1$ vertices.*

4 Longest Paths

How long can monotone paths be on a linear program of dimension d with n constraints? This is an extremal problem that was perhaps first asked by Klee in 1965 [19], and for which the complete answer is not really known (although the general impression about this may be different).

Indeed, let us assume here and in the following that we are dealing with d-dimensional polytopes with n facets that are simple, with an objective function that is nondegenerate. Then we can consider the following three quantities:

- $M(d,n)$ is defined as the maximal number of vertices on a monotone path on a simple d-polytope with n facets. This is the quantity we are after: it represents the worst case for the simplex algorithm with the most stupid choice of pivots, in the worst possible example.
- $M_{ubt}(d,n)$ is defined as the maximal number of vertices for a d-dimensional polytope with n facets. Clearly this represents an upper bound for $M(d,n)$, and a claim by Motzkin [28] led to the "upper bound conjecture" that the maximum is given by the dual of a cyclic polytope $C_d(n)$. The upper bound theorem was proved by McMullen in 1970 [26]; so we now know that

$$M_{ubt}(d,n) \quad = \quad f_{d-1}(C_d(n)) \quad = \quad \binom{n-\lceil\frac{d}{2}\rceil}{\lfloor\frac{d}{2}\rfloor} + \binom{n-1-\lceil\frac{d-1}{2}\rceil}{\lfloor\frac{d-1}{2}\rfloor}.$$

- $M_{sh}(d,n)$ is the maximal number of any 2-dimensional projection of a simple d-polytope with n facets. This quantity is of interest since it represents the worst case for the simplex algorithm with the shadow vertex rule on a d-dimensional problem with n constraints. This is also the rule for which Borgwardt [2] has shown that the simplex algorithm is polynomial (essentially linear) on average for a reasonable model of "random linear program." It is also the Gass-Saaty rule for parametric linear programming.

In summary, we now have a chain of inequalitites

$$M_{sh}(d,n) \quad \le \quad M(d,n) \quad \le \quad M_{ubt}(d,n),$$

where we know the exact value of the right hand side, we have exponential lower bounds for the left hand side, but the quantity in the middle is the one that "we are after." But how tight are these bounds? Do we always have equality?

To illustrate the gap in our knowledge, let us first discuss the "diagonal" case of $n = 2d$. In this case the maximal number of vertices is roughly given by $M_{ubt} \approx \left(\frac{27}{4}\right)^{d/2} \approx 2.6^d$, while the lower bound is at least $2^d \le M_{sh}(d,2d)$, as is shown by the deformed cubes of Goldfarb [10] [1, Section 4.3]. Of course there is a huge gap between 2^d and 2.6^d, and we do not know which of the two bounds is closer to the truth.

For a second example, let us just consider the case $d = 4$. In this case we have $M_{ubt}(4,n) = \frac{1}{2}n(n-3)$, and this value is achieved by the duals of cyclic 4-polytopes. On the other hand, the 2-dimensional projections of duals of cyclic polytopes have at most $3n$ vertices, and these do not represent the best lower bound for $M_{sh}(d,n)$ [1, Section 5.4].

It is clear (see [1]) that $M_{sh}(d,n) = M(d,n) = M_{ubt}(d,n)$ holds for $d \leq 3$ as well as for $n \leq d+2$. A current study of Kaibel, Pfeifle, and Ziegler [15] is resolving at least the first few nontrivial precise values. In particular, for $d = 4$ and $n = 7, 8$ we have

$$12 \quad = \quad M_{sh}(4,7) \quad < \quad M(4,7) \quad = \quad M_{ubt}(4,7) \quad = \quad 14$$

and

$$16 \quad \leq \quad M_{sh}(4,8) \quad \leq \quad M(4,8) \quad = \quad M_{ubt}(4,8) \quad = \quad 20.$$

In particular, the lower bound of $M(4,8) \geq 17$, first achieved by C. Schultz [31] on the dual of a cyclic polytope, is better than the lower bound of 16 that gets from the Klee-Minty cubes.

On the other hand, we know from an enumeration of "Hamiltonian abstract objective functions with the Holt-Klee property" that the value of 20 is *not* achieved for duals of cyclic polytopes $C_4(8)$: These are customarily taken as the canonical examples of polytopes that yield equality in the upper bound theorem, but they are not the only ones. For $d = 4$ and $n = 8$ there are exactly two other types of neighborly 4-polytopes with 8 vertices, called N_8 and N_8^* by Grünbaum [11, p. 125], and on both of these equality $M(4,8) = 20$ is achieved.

In summary, the available data don't contradict a conjecture that

$$M(d,n) \quad = \quad M_{ubt}(d,n)$$

holds for all $n > d > 1$. The author believes that this conjecture is indeed quite plausible. In any case, it is interesting that neither deformed products nor dual-to-cyclic polytopes give worst-possible results.

5 Short Paths

Let us now reverse the question: We are not any more asking for bad examples for a given pivot rule, but we rather ask for a pivot rule that is good on all examples. Thus we are trying to answer a tandem of two questions:

- Is there always a short path "to the top"?
- … and can one find one?

Our geometric model/interpretation is again that we are studying a simple d-dimensional simple polytope for which the last coordinate x_d is a linear objective function that is not constant on any edge. A *short path* is any path whose length (number of vertices) is polynomial in the number n of facets (and in the dimension $d < n$).

A natural approach, trying to provide a positive answer to the questions above, is to construct and analyze (new) pivot rules that have a chance to be (at least) polynomial. Thus we are looking for short monotone paths from any given vertex of the polytope to the (unique) top vertex. This is doomed to fail if there are no such short paths at all, perhaps not even nonmonotone short paths. A key question, first apparently posed by W. Hirsch in 1957, is the following:

Conjecture 1 (The Hirsch Conjecture [3, pp. 160, 168]). *Does every d-dimensional polytope with n facets have graph-theoretic diameter at most $n-d$?*

This is a famous/notorious question, and there have been many diverse attempts to provide an answer. We refer to the extensive survey by Klee and Kleinschmidt [20] for information, as well as to [36, Lecture 3] for more recent updates. In particular, the following two conjectures are known to be equivalent to the Hirsch Conjecture:

Conjecture 2 (The d-Step Conjecture). *Is it true that for all $d > 1$ and for all d-polytopes with $2d$ facets, and for any two vertices u and v that do not lie in a common facet, there is a path from u to v of length d?*

Conjecture 3 (The Nonrevisiting Path Conjecture). *For any two vertices on a simple polytope, is there always a path between them that does not leave and then revisit any one of the facets?*

The Hirsch Conjecture is old, classical, interesting, important, and still unsolved. More concretely, the status of the conjecture may be summarized as follows:

- The Hirsch conjecture is true for $d \leq 3$, but not proved for any $d > 3$ [22].
- The d-Step Conjecture is known to be true for $d \leq 5$ [22].
- The Hirsch Conjecture is tight for all $n > d \geq 8$: for any parameters in this range, there is a d-polytope with n facets that has graph-theoretic diameter exactly $n-d$ [12] [5].
- No polynomial upper bound is known for the diameter of a simple d-polytope with n facets; the best upper bound of $n^{\log d+1}$ is due to Kalai and Kleitman [18].

In an attempt to provoke the construction of interesting (counter)examples for the Hirsch Conjecture, the following "rather daring" conjecture was published in 1995:

Conjecture 4 (The "Strong Monotone" Hirsch Conjecture [36]). *For any simple d-polytope with n faces and for any generic linear objective function, is there always a* monotone *path from the (unique) minimal vertex to the (unique) maximal vertex of length at most $n-d$?*

This conjecture escapes the counterexamples to the "Monotone Hirsch Conjecture" by Todd [33], for which the starting vertex is not the minimal one. It also escaped an attack by Holt and Klee [13]. Thus this conjecture is still open.

How about pivot rules that have a chance to find polynomial paths on linear programs? In the following, we want to discuss three such rules.

Pivot Rule I ("Random_Edge"). *Given any vertex that is not the top vertex, choose one of the outgoing improving edges, uniformly with equal probability.*

Warning: This pivot rule sounds simple, but it seems to be awful to analyze in any nontrivial example. Its status may be summarized as follows:

- On the Klee-Minty cubes this pivot rule has essentially quadratic running times:
$$\frac{d^2}{8\log d} \quad\leq\quad E_d \quad\leq\quad \binom{d+1}{2};$$
see Gärtner, Henk, and Ziegler [7].

- The running time of this pivot rule is at most quadratic on all iterated deformed product examples.

- Thus this rule might as well be quadratic in expected running time on every example, but no sub-exponential upper bound on the expected running time has been proved.

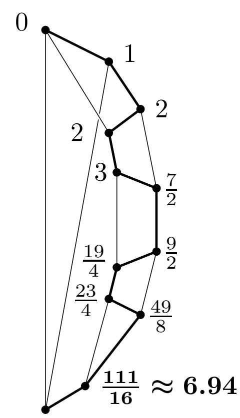


Figure 5. *A bad* 3*-dimensional example for* Random_Edge (from [32])

As a challenge (and a nice example to illustrate the complexity and behaviour of the Random_Edge rule), we ask for the maximal expected running time on a 3-dimensional simple polytope with n facets and $2n-4$ vertices. For this Figure 5 indicates a classe of examples for which the expected running time from the "worst" vertex is roughly $\frac{4}{3}n$. The expected running time for every starting vertex can be computed recursively for the Random_Edge rule: It is 0 for the top vertex, and for every other vertex it is 1 plus the average of the expected running times when starting at its upper neighbors. The resulting values are also indicated for the example in Figure 5. We refer to Kaibel et al. [32] for a detailed discussion of the Random_Edge rule on 3-dimensional polytopes — whose worst-case behaviour poses surprisingly tricky problems (for example, the factor $\frac{4}{3}$ is not worst-possible).

Pivot Rule II ("RANDOM_FACET"). *At any given starting vertex that is not the top vertex, move up if there is a unique edge on which the objective function is increasing. If there is more than one such edge, choose a random facet that contains the given vertex, and solve the linear program restricted to this facet recursively (that is, by calling* RANDOM_FACET*).*

This rule may seem a bit more contrived than RANDOM_EDGE, but it turns out to be sometimes much more accessible to analysis. It was introduced by Kalai [17], and simultaneously (in a dual simplex algorithm setting) by Matoušek, Sharir, and Welzl [25]. Its status may be summarized as follows:

- The running time of RANDOM_FACET is at most quadratic (in d) on the Klee-Minty cubes. In fact, one can come up with an exact formula that yields the exact expected running time for every single starting vertex on the Klee-Minty cubes [7].
- There is a sub-exponential (but not polynomial) upper bound for the expected running time of the RANDOM_FACET simplex algorithm of

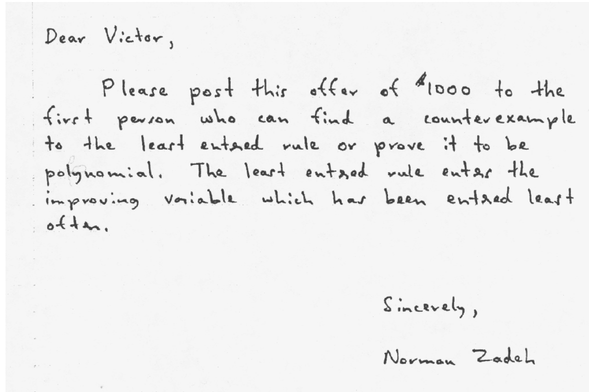
$$O\left(d^2 n + c^{\sqrt{d \log d}}\right),$$

  due to Kalai [17], Matoušek, Sharir, and Welzl [25].
- RANDOM_FACET is *slow* on the Matoušek-cubes [24]: These are edge orientations of the d-dimensional cubes that are not in general geometrically realizable. Combinatorially they may be described recursively as follows: In the bottom facet of the d-cube take any Matoušek-orientation; all the vertical edges are directed upward; on the top facet, we copy the directions from the bottom facet, except that all the edges of any given parallel class may be reversed (simultaneously).
- But amazingly, there is again a *quadratic* upper bound for the running time of RANDOM_FACET on any Matoušek-cube that is geometrically realizable, as was shown by Gärtner [6].

Finally, we want to discuss a deterministic rule which still (as far as we know) has a chance to be polynomial in the worst case:

Pivot Rule III ("LEAST_ENTERED"). *At any given vertex that is not the top vertex, among the increasing edges that leave the vertex choose any edge that leaves a facet that has been left least often on the previous moves.*

Thus this rule is deterministic, but its choices depend heavily on previous choices. Also it cannot be purely implemented just on the graph: It is essential that we know for each edge the (unique!) facet of the polytope that it leaves. The formulation given here depends heavily on the fact that we deal with a simple polytope: Given any edge incident to a vertex v of a simple polytope, there is always a *unique* facet that contains v but not the edge.



Dear Victor,

Please post this offer of $1000 to the first person who can find a counterexample to the least entered rule or prove it to be polynomial. The least entered rule enters the improving variable which has been entered least often.

Sincerely,

Norman Zadeh

Figure 6. *Zadeh's offer*

The Least_Entered rule was proposed by Norman Zadeh around 1980, and he offered $1000 to anyone who can prove or disprove that this rule is polynomial in the worst case; see the text of Figure 6 in Zadeh's handwriting (from a letter to Victor Klee, reproduced with his kind permission). Just to encourage the readers to try their luck on this problem, we want to mention that according to a recent magazine report [23], Norman Zadeh is now a successful businessman for whom it should be no problem to pay for the prize once you have solved the problem. Good luck!

Bibliography

[1] N. Amenta and G. M. Ziegler, *Deformed products and maximal shadows*, Advances in Discrete and Computational Geometry (B. Chazelle, J. E. Goodman, and R. Pollack, eds.), Contemporary Mathematics, vol. 223, Amer. Math. Soc., Providence RI, 1998, pp. 57–90.

[2] K. H. Borgwardt, *The simplex method. a probabilistic analysis*, Algorithms and Combinatorics, vol. 1, Springer-Verlag, Berlin Heidelberg, 1987.

[3] G. B. Dantzig, *Linear programming and extensions*, Princeton University Press, Princeton, 1963, Reprint 1998.

[4] S. Fischer, *Zweidimensionale Projektionen von linearen Programmen*, Diplomarbeit, TU Berlin, 1998, 69 pages.

[5] K. Fritzsche and F. B. Holt, *More polytopes meeting the conjectured Hirsch bound*, Discrete Math. **205** (1999), 77–84.

[6] B. Gärtner, *Combinatorial linear programming: Geometry can help*, Proc. 2nd Workshop "Randomization and Approximation Techniques in Computer Science (RANDOM)" (Heidelberg), Lecture Notes in Computer Science, vol. 1518, Springer-Verlag, 1998, pp. 82–96.

[7] B. Gärtner, M. Henk, and G. M. Ziegler, *Randomized simplex algorithms on Klee-Minty cubes*, Combinatorica **18** (1998), 349–372.

[8] E. Gawrilow and M. Joswig, *Polymake: A software package for analyzing convex polytopes*, `http://www.math.tu-berlin.de/diskregeom/polymake/`.

[9] ______, *`Polymake`: A framework for analyzing convex polytopes*, Polytopes — Combinatorics and Computation (G. Kalai and G.M. Ziegler, eds.), DMV Seminar, vol. 29, Birkhäuser-Verlag, Basel, 2000, pp. 43–73.

[10] D. Goldfarb, *On the complexity of the simplex algorithm*, Advances in optimization and numerical analysis (Dordrecht) (S. Gomez, ed.), Kluwer, 1994, Proc. 6th Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico, January 1992, pp. 25–38.

[11] B. Grünbaum, *Convex Polytopes*, Interscience, London, 1967; second edition (V. Kaibel, V. Klee, G.M. Ziegler, eds.), Graduate Texts in Math. 221, Springer-Verlag New York 2003.

[12] F. B. Holt and V. Klee, *Counterexamples to the strong d-step conjecture for $d \geq 5$*, Discrete Comput. Geometr **19** (1998), 33–46.

[13] ______, *Many polytopes meeting the conjectured Hirsch bound*, Discrete Comput. Geometry **20** (1998), 1–17.

[14] V. Kaibel, R. Mechtel, M. Sharir, and G. M. Ziegler, *The simplex algorithm in dimension three*, Preprint, 24 pages, September 2003; `math.CO/0309351`.

[15] V. Kaibel, J. Pfeifle, and G. M. Ziegler, *On the monotone upper bound problem*, in preparation, 2002.

[16] G. Kalai and G. M. Ziegler (eds.), *Polytopes — combinatorics and computation*, DMV Seminars, vol. 29, Birkhäuser-Verlag, Basel, 2000.

[17] G. Kalai, *A subexponential randomized simplex algorithm*, Proc. 24th ACM Symposium on the Theory of Computing (STOC), ACM Press, 1992, pp. 475–482.

[18] G. Kalai and D. J. Kleitman, *A quasi-polynomial bound for the diameter of graphs of polyhedra*, Bulletin Amer. Math. Soc. **26** (1992), 315–316.

[19] V. Klee, *Heights of convex polytopes*, J. Math. Analysis Appl. **11** (1965), 176–190.

[20] V. Klee and P. Kleinschmidt, *The d-step conjecture and its relatives*, Math. Operations Research **12** (1987), 718–755.

[21] V. Klee and G. J. Minty, *How good is the simplex algorithm?*, Inequalitites, III (O. Shisha, ed.), Academic Press, New York, 1972, pp. 159–175.

[22] V. Klee and David W. Walkup, *The d-step conjecture for polyhedra of dimension* $d < 6$, Acta Math. **117** (1967), 53–78.

[23] S. Lubove, *See no evil. Much of the raunchy porn on the internet wouldn't exist were it not for the help of a handful of legitimate companies operating quietly in the background*, Forbes (September 17, 2001), 68–70.

[24] J. Matoušek, *Lower bounds for a subexponential optimization algorithm*, Random Structures and Algorithms **5** (1994), 591–607.

[25] J. Matoušek, M. Sharir, and E. Welzl, *A subexponential bound for linear programming*, Proc. Eighth Annual ACM Symp. Computational Geometry (Berlin 1992), ACM Press, 1992, pp. 1–8.

[26] P. McMullen, *The numbers of faces of simplicial polytopes*, Israel J. Math. **9** (1971), 559–570.

[27] T. Morstein, *Die Probleme von Zadeh zum Netzwerk-Simplex sind deformierte Produkte von Polytopen*, Diplomarbeit, TU Berlin, 1999, 23 pages.

[28] T. S. Motzkin, *Comonotone curves and polyhedra*, Bulletin Amer. Math. Soc. **63** (1957), 35, Abstract.

[29] M. Padberg, *Linear Programming*, second ed., Algorithms and Combinatorics, vol. 12, Springer-Verlag, Heidelberg, 1999.

[30] V. Reiner and G. M. Ziegler, *Coxeter-associahedra*, Mathematika **41** (1994), 364–393.

[31] C. Schultz, *Schwierige lineare Programme für den Simplex-Algorithmus*, Diplomarbeit, TU Berlin, 2001, 81 pages.

[32] M. Sharir and G. M. Ziegler, *The random-edge simplex algorithm on three-dimensional polytopes*, in preparation, 2002.

[33] M. J. Todd, *The monotonic bounded Hirsch conjecture is false for dimension at least* 4, Math. Operations Research **5** (1980), 599–601.

[34] D. Weber, *Kombinatorische Analyse einiger linearer Programme*, Diplomarbeit, TU Berlin, 1999, 29 pages.

[35] N. Zadeh, *A bad network problem for the simplex method and other minimum cost flow algorithms*, Math. Programming **5** (1973), 255–266.

[36] G. M. Ziegler, *Lectures on Polytopes*, revised ed., Graduate Texts in Mathematics, vol. 152, Springer-Verlag, New York, 1995; Revised edition, 1998.

Note added in proof. There is substantial recent progress on the "monotone upper bound problem" discussed in Section 14.4: The inequality $M_{ubt}(d,n) \leq M(d,n)$ is tight (holds with equality) for the cases

$n \leq d+2$**:** see B. Gärtner, J. Solymosi, F. Tschirschnitz, P. Valtr, and E. Welzl, One line and n points, in: Proc. 33rd Ann. ACM Symp. Theory of Computing (STOC), ACM Press 2001, pp. 306-315, and

$d \leq 4$**:** see J. Pfeifle, Extremal Constructions for Polytopes and Spheres, Dissertation, TU Berlin 2003.

However, the inequality is not tight in general:

$$M(6,9) < M_{ubt}(6,9) = 30$$

was established by J. Pfeifle and G. M. Ziegler, On the Monotone Upper Bound Problem, Preprint, 15 pages, TU Berlin 2003; math.CO/0308186; Experimental Math., to appear.