Mathematical Programming manuscript No.
(will be inserted by the editor)

Michael J. Todd

The many facets of linear programming [*]

March 15, 2002

Abstract. We examine the history of linear programming from computational, geometric, and complexity points of view, looking at simplex, ellipsoid, interior-point, and other methods.

Key words. linear programming – history – simplex method – ellipsoid method – interior-point methods

1. Introduction

At the last Mathematical Programming Symposium in Lausanne, we celebrated the 50th anniversary of the simplex method. Here, we are at or close to several other anniversaries relating to linear programming: the sixtieth of Kantorovich's 1939 paper on "Mathematical Methods in the Organization and Planning of Production" (and the fortieth of its appearance in the Western literature) [55]; the fiftieth of the historic 0th Mathematical Programming Symposium which took place in Chicago in 1949 on Activity Analysis of Production and Allocation [64]; the forty-fifth of Frisch's suggestion of the logarithmic barrier function for linear programming [37]; the twenty-fifth of the awarding of the 1975 Nobel

School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853–3801, e-mail: miketodd@orie.cornell.edu

Prize in Economics to Kantorovich and to Koopmans (and most disappointingly not to Dantzig – see the article [8] by Balinski for some related history) for their contributions to the theory of optimum allocation of resources; the twentieth anniversaries of Khachiyan's 1979 and 1980 papers [57,58] using the ellipsoid method to prove polynomiality of the linear programming problem; and the fifteenth of Karmarkar's paper introducing the projective method to again establish polynomiality and reinvigorating the study of interior-point methods to such a remarkable extent.

Let me start by giving two quotes from the Nobel prizewinners (two of the notable list of individuals with the initial K who have made significant contributions to linear programming, including also Klee, Khachiyan, Karmarkar, and Kalai, who will figure later in this paper) and one from Dantzig:

Kantorovich writes in the introduction of [55]: "I want to emphasize again that the greater part of the problems of which I shall speak, relating to the organization and planning of production, are connected specifically with the Soviet system of economy and in the majority of cases do not arise in the economy of a capitalist society." (This was undoubtedly added to make the paper, with its disguised decadent ideas of dual prices, more palatable to his communist censors – and we shall see more instances where the science of linear programming came up against the realities of the Cold War. In this regard, see the comments [24] on the translated version by Charnes and Cooper and the following rebuttal by Koopmans. See also the article by B. T. Polyak in this volume.)

A prescient quote from Tjalling Koopmans in the introduction to [64] reads: "It has been found so far that, for any computation method which seems useful in relation to some set of data, another set of data can be constructed for which that method is obviously unsatisfactory." (This compares strikingly with the quote from Bixby et al. [13] at the end of this section.)

In [30], Dantzig writes: "Luckily the particular geometry used in my thesis was the one associated with the columns of the matrix instead of its rows. This column geometry gave me the insight which led me to believe that the *simplex method* would be an efficient solution technique. I earlier had rejected the method when I viewed it in the row geometry because running around the outside edges seemed so unpromising."

Since much has been written about the early history (and pre-history) of linear programming, for example in [29], Chapter 2, [30], and [83], pp. 209–225, this paper will concentrate more on developments since the seventies. I hope to intrigue the reader enough to investigate some of the byways and alleys associated with linear programming as well as the more well-travelled highways. We will look at simplex, ellipsoid, and interior-point methods, and also at least mention some other approaches. Of course, I hope the reader will forgive my personal bias in the topics selected. (Let me mention here Megiddo's article [75], which also surveys some recent developments from a different viewpoint.)

Following the development of the simplex method in 1947 [27], the '50s had been the decade of developing the theoretical underpinnings of linear programming, of extending its applicability in industrial settings and to certain combinatorial problems, and of the first general-purpose codes. The '60s saw the emergence of large-scale linear programming, of exploitation of special structure (again pioneered by Dantzig and Dantzig-Wolfe in [28,31]), and of extensions to quadratic programming and linear complementarity. If the '50s and the '60s were the decades of unbridled enthusiasm, the '70s were the decade of doubt, as the theory of computational complexity was developed and Klee and Minty [60] showed that the simplex method with a common pivot rule was of exponential complexity. We will concentrate on the developments since that time; hope has been restored by new polynomial-time algorithms, by bounds on the expected number of pivot steps, and by amazing computational studies on problems with numbers of variables ranging up to the millions.

Linear programming studies the optimization of a linear function over a feasible set defined by linear inequalities, hence a polyhedron. The problem is in some sense trivial, since it is only necessary to examine a finite number of vertices (and possibly edges), but if one is interested in efficient computation, the topic is wonderfully rich and has been the subject of numerous surprising new insights.

A geometric view can be at times helpful (but also profoundly misleading, as mentioned by Dantzig on several occasions). Let us consider two paradigms: optimization of a linear function over a simplex or over an ellipsoid. In the first case the feasible region is "spiky"; there are few vertices, and any one can be reached from any other in one step: here the simplex method is a natural choice. In the second case, an optimal solution can easily be found by calculus, but there are an infinite number of extreme points. Are the feasible regions of large-scale instances arising in important problem domains more like "quartz crystals" [51], with long edges from one side to the other, so that combinatorial algorithms will be efficient, or more like "disco balls" [50], where simplex-like methods seem doomed to be slow (as in Dantzig's earlier intuition mentioned above) but approximations by ellipsoids as in the ellipsoid method or interior-point methods look promising? The remarkable success of algorithms from both classes suggest that real problems have both features.

After a short section setting our notation, the subsequent sections consider the simplex method, the ellipsoid method, interior-point methods, and other approaches. In each section we start with an overview and historical comments, and then provide some more technical material relating to understanding the methods better or giving relations between them. I have of course omitted much that is highly important (everything concerned with efficient computation and exploitation of special structure, for example!), but I hope to have mentioned some avenues that are new to readers. The paper concludes with some very brief remarks on the future.

Let me close this section with an anecdote: One day in April, I experienced an interesting time warp. I read a paper by Hoffman et al. from 1953 [49], discussing computational experiments comparing three methods for linear programming problems and computing optimal strategies for zero-sum games (the simplex method, fictitious play, and the relaxation method); the authors concluded that the simplex method was most effective, and could even solve large-scale problems with dimensions of the order of 50 by 100, using the SEAC (Standards Eastern Automatic Computer)! (Also see Hoffman's recollections of this [48].) I later attended a seminar by Bob Bixby, also recounting computational experience in solving linear programming problems, and giving results for a particular problem of size 49,944 by 177,628. The contrast was quite striking. Incidentally, Bixby was also comparing three methods (primal and dual simplex methods and a primal-dual interior-point method), and his conclusion was that, for a suite of large problems, the dual simplex method using the steepest-edge pivot rule was the fastest (see [13]). Bixby and his co-authors indicate that the figures given can be viewed as biased against the interior-point (barrier) code, but state: "What can one say in general about the best way to solve large models? Which algorithm is best? If this question had been asked in 1998, our response would have been that barrier was clearly best for large models. If that question were asked now, our response would be that there is no clear, best algorithm. Each of primal, dual, and barrier is superior in a significant number of important instances."

2. Preliminaries

We consider a primal-dual pair of linear programming problems:

$$\begin{aligned} &\text{(P)}\quad \text{minimize } c^T x \quad \text{subject to } Ax = b, x \geq 0,\\ &\text{(D)}\quad \text{maximize } b^T y \quad \text{subject to } A^T y \leq c, \end{aligned}$$

where $c \in R^n$, $A \in R^{m \times n}$ and $b \in R^m$ are given data, and $x \in R^n$ and $y \in R^m$ are the decision vectors. For simplicity of argument, we assume that the matrix

A has full row rank m and that both problems have feasible solutions. Then the feasible region of (P) is the intersection of a $d := n - m$ dimensional affine subspace with the nonnegative orthant in R^n, while the feasible region of (D) is the intersection of n halfspaces in R^m. Both are polyhedra with vertices, by virtue of our assumption on A. Some of our arguments are most easily expressed in one formulation or the other, but it is easy to switch between them: the d-dimensional affine subspace can be parametrized by d variables, and then the feasible region of (P) is the intersection of n halfspaces in R^d; and the dual slack vector $s := c - A^T y$ lies in an m-dimensional affine subspace and is also required to be nonnegative.

Both of these viewpoints consider the geometry of the set of feasible solutions, but another perspective, the column geometry of Dantzig [29], pp. 160 ff., is also very valuable: indeed, without this perspective, the simplex method might have been still-born (see [30]). Here we consider the convex cone C spanned by the vectors $(a_j; c_j)$ in R^{m+1}, where a_j is the jth column of A and c_j the jth component of c. If the objective function is not constant on the feasible region, this cone will have dimension $m+1$. We seek the "lowest" point (in terms of its last component) that lies in the intersection of this cone and the "vertical" line $\{(b; \zeta) : \zeta \in R\}$.

Different solution strategies for linear programming arise from different views of these geometries. Since we are optimizing a linear function over a polyhedron with vertices, an optimal solution (if it exists) will occur at a vertex; it is therefore natural to consider only the vertices and the adjacency between them given by the edges of the polyhedron. This is the viewpoint of the simplex method, and it relies on considering the combinatorial structure of the faces of polyhedra. In terms of the column geometry, we consider the set of simplicial subcones of dimension $(m+1)$ of the cone C described above that intersect the vertical line, with two cones adjacent if they share an m-face. This simplex interpretation is described in [27], in [49], and in more detail in [29].

Other methods instead view the feasible region as a convex set, which can be approximated by simpler convex sets with smooth boundaries over which linear optimization is trivial. Then the focus is on obtaining “good” approximations to the feasible polyhedron by suitable simpler sets.

3. The simplex method

For definiteness we consider the primal simplex method, which moves from vertex to vertex of the primal feasible region until it reaches an optimal solution (or gets an indication of unboundedness). Each vertex corresponds to a basic feasible solution and hence to a choice of m basic variables from the n variables in total.

It is clear that, with hindsight, we can move from the initial vertex to the final vertex in at most m basic exchanges, at each step replacing one of the initial basic variables by one of the final basic variables. However, there is no reason why the intermediate basic solutions should be feasible. Moreover, the simplex method cannot use hindsight, and indeed uses only local information at the current vertex. It is therefore worth emphasizing the

> Remarkable fact: the (primal) simplex method typically requires at most $2m$ to $3m$ pivots to attain optimality.

This serendipitous property was recognized quite early (something similar is mentioned by Hoffman et al. in [49] in 1953), and was stated in Dantzig [29], page 160, as based on empirical experience with thousands of practical problems. Numerical evidence can be found in Wolfe and Cutler [96] and Kuhn and Quandt [66]. This experience was perhaps a factor in discouraging the development of other algorithms from the late '50s and '60s, although some alternatives continued to be advanced. Of course, the dual simplex method of Lemke [69] was of great importance, both for reoptimization following a change in the right-hand sides and to solve problems from scratch (indeed the dual steepest-edge simplex method studied and made practical by Forrest and Goldfarb [36] seems to be the simplex method of choice today), but for our purposes here it can be viewed as the primal simplex method applied to the dual problem. More recent evidence for the claim comes from computational experiments carried out to demonstrate the competitive behavior of the simplex method following the great excitement and provocative computational results of interior-point methods. For example, Bixby [11] gives results for an early version of CPLEX (1.0) on 90 Netlib [39] problems. For 72, the number of total iterations was at most 3 times the row size; for 16, the ratio was between 3 and 7; and for the remaining three, it was 10.7, 39.5 and 469.1. (The last three had "unbalanced" m and n: their $m \times n$ sizes were 1000 x 8806, 24 x 1026, and 25 x 10,500 respectively; here m denotes the number of general linear constraints, while the last two problems had both upper and lower bounds on all the variables, so that perhaps m should be taken as 1050 and 10525, and then the ratios are under 1.2.) On a set of 8 larger problems considered in [12], the ratios for CPLEX 2.2 (primal) were from .46 to 1.99 on three, from 4.52 to 9.02 on four, and 17.58 on the last. Note that these problems (especially the largest) were chosen to present difficulties for linear programming algorithms; their general favorable performance is then even more surprising.

3.1. Diameter

One way to try to explain the success of the simplex method is to study the *diameter* of a polyhedron, the largest number of edges in a shortest path joining two of its vertices. Let $\Delta(d,n)$ denote the largest diameter of a d-dimensional polyhedron with n facets; this represents the best-possible number of iterations for the worst linear programming problem of these dimensions, initiated at the worst vertex. W. M. Hirsch conjectured in 1957 that $\Delta(d,n) \leq n-d$ (so that m steps would be sufficient in the remarkable fact above), see Dantzig [29], pp. 160. It is known that this bound fails for unbounded polyhedra (Klee and Walkup [61]), and it also fails for bounded polyhedra (polytopes) if the path is required to be monotonic with respect to the objective function [87]. The general case for polytopes is still open.

The best bounds available are $\Delta(d,n) \leq 2^{d-3}n$ due to Larman [67], and $\Delta(d,n) \leq n^{1+\log d}$ due to Kalai and Kleitman [54]. The Hirsch conjecture holds for 0–1 polytopes (Naddef [77]) and for dual transportation polyhedra (Balinski

[7]); and the diameters of certain combinatorial polytopes (e.g., the assignment polytope and the asymmetric traveling salesman polytope) are just 2 (Balas and Padberg [6], Padberg and Rao [79]).

Klee and Kleinschmidt [59] give an excellent survey on these matters, and Kleinschmidt [62] provides an update and some algorithmic consequences.

3.2. Exponential and subexponential pivot rules

Klee and Minty [60] were the first to give an example of a class of linear programming problems for which Dantzig's classic most-negative-reduced-cost pivot rule required an exponential number of pivots. Since then exponential examples have been found for several other pivot rules, including the best-neighbor rule which solves linear optimization over the simplex (and the original Klee-Minty examples) in just one pivot, and the steepest-edge rule which is the basis for the fast simplex codes of today.

It is therefore surprising that subexponential (but superpolynomial) pivot rules have been found, by Kalai [52] and Matousek, Sharir, and Welzl [73]. These are all (thus far) randomized, with a best bound on the expected number of pivots of $\exp(K\sqrt{d \log n})$ for some constant K. One version is particularly easy to describe (roughly):

- Given a vertex v, choose a facet F containing v at random.
- Apply the algorithm recursively to find the optimizing vertex w in F.
- Repeat the algorithm from w.

The analysis, and much more fascinating material, is in Kalai [53].

3.3. Probabilistic Analysis

Instead of randomizing the pivot rule, we can ask for the expected behavior of a deterministic rule on a random linear programming problem drawn from some distribution. This was a topic under intense study in the late '70s and early '80s. Major contributions were made by Borgwardt [15,16], Smale [85], Adler [1], Haimovich [47], Adler, Karp, and Shamir [2], Adler and Megiddo [3], and Todd [88].

Borgwardt was the first to obtain a polynomial bound. His final estimate, from [18], gives a bound $0(m^3 n^{1/(m-1)})$ for the expected total number of iterations of a dimension-by-dimension simplex method for the dual problem (D). Here, the data $b, a_1, ..., a_n$, where a_j denotes the jth column of A, are required to come from a rotationally symmetric distribution, and c is the vector of ones. Hence the origin is always feasible, and is used by the algorithm as a starting point. The generated problems are always feasible and have optimal solutions with high probability if $n \gg m$. Some results have also been obtained for a related parametric method where the c_j's are also random and can be negative (Borgwardt [17]).

In contrast, Adler and Megiddo [3], Adler, Karp, and Shamir [2], and Todd [88] deal with the so-called sign-invariant model, and obtain a bound of $0(\min\{d^2, m^2\})$ on the expected total number of iterations to show infeasibility, show unboundedness, or attain optimality for a lexicographic parametric method. One (severe) disadvantage of this probabilistic model is that problems are either infeasible or unbounded with probability approaching 1 as n tends to ∞; however, in the case that $n = 2m$ it is possible to derive a bound of $0(m^{\frac{5}{2}})$ expected total iterations with the expectation conditioned on the problem's having an optimal solution.

For a survey of these and related results, see the 0th chapter of Borgwardt's book [16] or the last section of [18]. There is also a discussion of these issues in [90].

3.4. Big faces, little faces, long edges

To conclude this section we describe some results that try to give some intuition as to why the simplex method works so well.

First, there may be a few "big" faces on which most of the action takes place. Kuhn [65] did some experiments in 1953 in which he "shot" ten random bullets from the centroid of the asymmetric travelling salesman polytope on 5 cities, finding which facet each penetrated. Each shot led to a 21 x 25 LP problem which was (just) within the capability of the SEAC, and each went through a trivial (nonnegativity) facet. In 1991, he repeated the experiment on a larger scale. Of over 150,000 shots, 80% went through these trivial facets, which numbered 20 out of the 390 facets in all.

Of course, polytopes with a relatively small number of vertices can have a large number of facets. The symmetric travelling salesman polytope on 7 cities has 360 vertices and 3,437 facets: for 8 cities, the figures are 2,520 vertices and 194,187 facets (Christof et al. [25]). How many of these are "significant"? Goemans [41] showed that several classes of facets for the graphical travelling salesman problem had a very small effect on the optimal value when their inequalities were added to an LP relaxation. We can infer that perhaps many of these facets are "small."

By polarity, these results have implications for the vertices of polytopes with a relatively small number of facets (i.e., the feasible region of LP problems). There may be many vertices: the d-cube has $2d$ facets and 2^d vertices, and a polytope with $n = 2d$ can have as many as $\Omega(d^{\frac{d}{2}})$ vertices. But many of these vertices may be optimal for a very small set of objective functions, and may be relevant for a very small set of simplex algorithms.

Lastly, let us briefly consider long edges. It has been known for a long time that in dimension $d \geq 4$ there are neighborly polytopes (every pair of vertices is joined by an edge) with arbitrarily many vertices [46, 100]. Clearly such polytopes are not simple (each vertex has degree much larger than the dimension) unless they are simplices, but at least there is some basis representation of the initial vertex such that a single pivot reaches the optimal vertex. As mentioned above, there are classes of polyhedra arising in practice whose diameters are two. To complement these results I ran a very simple experiment to suggest that many polyhedra had "long edges" going from one side of the polyhedron to the other. For $d = 3, 4, ..., 11$, and $n = 2^d$ and 2^{d+1}, I generated 100 polytopes as the convex hull of n d-vectors. For the first type, each component of each vector was independently drawn from the standard normal distribution. For the second type, the resulting vectors were each scaled independently to have length the reciprocal of a number drawn uniformly at random from [0,1]. For the first type of polytope, at least half of the vectors were in fact vertices. I took the two maximally distant vertices and checked whether they were joined by an edge: this was true in between 3% and 49% of the cases, depending on the dimension d. For the second type, only a few of the vectors were vertices (only an average of 46 to 48 for the problems of dimension 10), but from 67% to 100% of the polytopes (and from 99% to 100% for those with $d \geq 7$) had their maximally-distant vertices joined by an edge.

4. The ellipsoid method

The ellipsoid method was originally developed by Yudin and Nemirovski [99] and Shor [84] for convex nonlinear programming, but it become famous when Khachiyan [57,58] used it to obtain a polynomial-time algorithm for linear programming. This was a major theoretical advance, but the popular press misinterpreted it in a rather ludicrous way (with headlines like "Soviet Answer to 'Travelling Salesmen' "; see Lawler's article [68]); later headlines retreated from their earlier excesses by invoking cold-war suspicions (as in the New York Times's "A Russian's Solution in Math Questioned," 21 March 1980, p. A13).

For a problem with n inequalities and integral data with total bit size L, the method generates an approximate solution from which an exact solution can easily be obtained in $0(n^2L)$ iterations, requiring $0(n^4L)$ arithmetic operations on numbers with $0(L)$ digits. (These bounds appear in Khachiyan [58]; his earlier extended abstract [57] gave a bound of $0(n^5L)$ arithmetic operations on numbers with $0(nL)$ digits.) This gives a polynomial bound, and thus Khachiyan was the first to show that linear programming was in the class P of polynomial-time solvable problems. Nevertheless, the performance in practice was typically very close to its worst-case bound, so that despite a number of refinements the method is not competitive for linear programming; for example, see the discussion in Bland et al. [14].

The basic idea of the ellipsoid method is well-known. At each iteration an ellipsoid is given which contains all optimal solutions. By considering the center of the ellipsoid, a hyperplane is constructed so that all optimal solutions lie on one side of the hyperplane (and the center either lies strictly on the other side (a deep cut) or on the hyperplane itself (a central cut)). Then a new ellipsoid is found which contains all points in the old ellipsoid and on the correct side of the hyperplane.

We will give some formulae below (as well as a new interpretation of the process). But first, let us mention that from this geometric view it appears that the method is quite insensitive to the number of inequalities, and seems likely to work well when the feasible region of the linear programming problem is close to a "disco ball" (as opposed to a "quartz crystal") — in this way it seems to complement the simplex method. Also, as long as a suitable hyperplane can be found, there is no need for the problem to be of linear programming type. Hence it can be used for convex programming, and is also highly useful for theoretically analyzing combinatorial optimization problems via the polyhedral combinatorics paradigm. This leads to the famous "separation = optimization" meta-theorem: if you can separate efficiently, you can optimize efficiently (over a suitable collection of convex bodies). We will not pursue this, as it falls outside the scope of this article, but instead refer the reader to the excellent monograph by Grötschel, Lovász, and Schrijver [45].

4.1. Update formulae

Now we briefly discuss the mechanics of an iteration, in order to describe the new interpretation. For simplicity, we assume that we are just trying to find a feasible point for (D), i.e., a point in $Y := \{y : A^T y \leq c\}$. We write the individual constraints as $a_j^T y \leq c_j, j = 1, 2, ..., n$, and assume that $Y \subseteq E_0$, where

$$E_0 := \{y : \|y\|_2 \leq R\}. \tag{1}$$

Any ellipsoid in $\Re^m$ can be written in the form

$$E = E(\bar{y}, B) = \{y : (y - \bar{y})^T B^{-1} (y - \bar{y}) \leq 1\}, \tag{2}$$

where $\bar{y}$ is its center and B is a symmetric positive definite matrix of order m. Indeed, $E_0 = E(y_0, B_0)$, with $y_0 = 0$ and $B_0 = R^2 I$. Given $E_k = E(y_k, B_k)$, $E_{k+1} = E(y_{k+1}, B_{k+1})$ can be constructed as follows:

Find j so that $a_j^T y_k > c_j$ (if none, STOP: $y_k \in Y$). Set

$$y_{k+1} := y_k - \frac{\tau B_k a_j}{(a_j^T B_k a_j)^{\frac{1}{2}}}, \tag{3}$$

$$B_{k+1} = \delta \left(B_k - \sigma \frac{B_k a_j a_j^T B_k}{a_j^T B_k a_j} \right), \tag{4}$$

where $\tau = 1/(m+1)$, $\delta = m^2/(m^2 - 1)$, and $\sigma = 2/(m+1)$. This gives E_{k+1} as the ellipsoid of minimum volume that contains the semi-ellipsoid $\{y \in E_k :$

$a_j^T y \leq a_j^T y_k\}$; if the minimum-volume ellipsoid containing $\{y \in E_k : a_j^T y \leq c_j\}$ is desired, the formulae are similar with different choices for τ, δ, and σ (deep cuts). It can be shown that vol $(E_{k+1})/\mathrm{vol}(E_k) \leq \exp(-1/[2m+2])$, and this systematic volume reduction leads to the complexity bound: see, e.g., Bland et al. [14].

4.2. An alternate representation

There is another way to represent the ellipsoid that leads to a surprising parallel with interior-point methods (see also [89]). Since we assume that $Y \subseteq E_0$, we can find lower bounds on $a_j^T y$ for $y \in Y$ for each j. So suppose $Y = \{y : \ell \leq A^T y \leq c\}$. Now let D be a nonnegative diagonal matrix. Then since $A^T y - \ell \geq 0$ and $A^T y - c \leq 0$ for all $y \in Y$,

$$Y \subseteq \bar{E}(D, \ell) := \{y : (A^T y - \ell)^T D (A^T y - c) \leq 0\}, \tag{5}$$

and the set on the right-hand side is an ellipsoid as long as ADA^T is positive definite. The advantage of this representation is that it gives a short certificate that $E = \bar{E}(D, \ell)$ contains Y. (The disadvantage is that it can only be used with linear programming and not with convex or combinatorial optimization where the constraints are not pre-specified.) From this viewpoint, the ellipsoid method generates a sequence $E_k = \bar{E}(D_k, \ell_k)$ of ellipsoids containing Y. The center of E_k is y_k, the solution of

$$AD_k A^T y = AD_k(\ell_k + c)/2, \tag{6}$$

or equivalently of the weighted least-squares problem

$$D_k^{\frac{1}{2}} A^T y \approx D_k^{\frac{1}{2}} (\ell_k + c)/2. \tag{7}$$

At the kth iteration, the index j of a violated constraint is found, the jth component of the vector ℓ_k is possibly updated, and the jth diagonal entry of the matrix D_k is increased. Since only one entry of D_k is changed, y_k can be updated cheaply (the update is exactly that given in (3)), as can $B_k = (AD_kA^T)^{-1}$. Details can be found in Burrell and Todd [22]. That paper also shows how ℓ_k is updated (each component is guaranteed to be a suitable lower bound by an application of the Farkas lemma).

Besides showing that the quadratic inequality defining each ellipsoid can be viewed as a weighted sum of quadratic constraints that ensure that each $a_j^T y$ lies in an appropriate range, this representation gives a clue to the slow convergence of the ellipsoid method. Suppose the lower-bound vector ℓ remains unchanged. Then the volume of $\bar{E}(D, \ell)$ is a function of just the diagonal entries of D, and the ellipsoid method can be thought of as a coordinate descent method to minimize this nonlinear function. It is known that coordinate descent methods can be very slow, and the ellipsoid method is no exception, although dramatic volume reductions are possible at some iterations. (See also Liao and Todd [71].)

5. Interior-point methods

The idea of moving through the interior of the feasible region goes back at least to Frisch in 1955 [37], who proposed using a logarithmic barrier function.

Barrier and penalty methods for nonlinear programming were studied in depth in the 1960s, leading to the classic text of Fiacco and McCormick [35], but came into disfavor as their numerical drawbacks became more recognized. The modern reincarnation of these methods is due to Karmarkar [56] in 1984, who established the polynomial-time boundedness of the projective method and also obtained some very competitive times (comparing with Fortran implementations of the simplex method) on a widely-used set of test problems. It turns out that the interior-point method implemented by Karmarkar (the affine-scaling method), besides being discovered simultaneously by a number of researchers in the mid 1980s, had in fact been proposed in 1967 and analyzed in 1974 by I. I. Dikin, a student of Kantorovich [33, 34].

Karmarkar's results and claims led to a furor of activity, though the popular press was much more accurate this time (viz. the front-page New York Times article "Breakthrough in Problem Solving" of Monday, November 19, 1984, written by James Gleick of chaos fame). The complexity bound was only slightly better than that of the ellipsoid method ($0(n^{3.5}L)$ arithmetic operations on a problem with n inequalities and integer data of total bit length L, although in practice the behavior is much better). However, the new ideas employed were very intriguing: at each iteration a projective transformation was used to bring the current iterate into the center of the feasible region, and a nonlinear potential function, invariant under such transformations, was used to measure progress. This potential function is close to Frisch's logarithmic barrier, and Gill, Murray, Saunders and Wright [40] showed that Karmarkar's search direction in fact is equivalent to that arising from a barrier function method with a suitable choice of barrier parameter (sometimes negative!) at each iteration.

The idea of making a projective transformation is to bring the current iterate to a point far from the constraints so that a steepest descent step (on the transformed objective function or the potential function) will give good decrease. However, projective transformations are not used much in interior-point methods nowadays. The key concept of making a transformation or changing the metric so the current iterate is in some sense far from the boundary remains highly valuable. I will discuss briefly dual and primal-dual path-following methods, and then make some remarks about potential-reduction algorithms. For more details, I recommend Gonzaga [44] and Wright [97] for path-following methods and Anstreicher [5] and Todd [91] for potential-reduction algorithms.

5.1. Dual path-following

Suppose we have a current strictly feasible solution $\bar{y}$ to the dual problem $\max\{b^T y : A^T y \leq c\}$; strictly feasible means that the dual slack vector $\bar{s} := c - A^T \bar{y}$ is positive in each component (written $\bar{s} > 0$). The largest ellipsoid around $\bar{s}$ that is contained in the nonnegative orthant is $E_s := \{s \in \Re^n : (s - \bar{s})^T \bar{S}^{-2} (s - \bar{s}) \leq 1\}$, where $\bar{S}$ is the diagonal matrix with the components of $\bar{s}$ on its diagonal. Thus the set of feasible dual solutions whose slack vectors lie in E_s is

$$E := \{y \in \Re^m : (y - \bar{y})^T A \bar{S}^{-2} A^T (y - \bar{y}) \leq 1\},$$

and $E \subseteq Y := \{y \in \Re^m : A^T y \leq c\}$. Note that here we have a point $\bar{y}$ that lies in Y, and E is inscribed in Y rather than circumscribing it, but that otherwise the ellipsoid E is remarkably similar to $\bar{E}(D, \ell)$ in (5).

In addition to the fact that $E \subseteq Y$, the matrix $A\bar{S}^{-2}A^T$ appearing therein defines a highly useful local metric at $\bar{y}$. Note that this matrix appears as the Hessian of the logarithmic barrier function

$$f(y) := -\sum_j \ln(c - A^T y)_j \tag{8}$$

evaluated at $\bar{y}$. This is a special case of a very general theory of self-concordant barrier functions developed by Nesterov and Nemirovski: see their monograph [78]. The knowledge of such a barrier function and its derivatives for a convex set is a sufficient condition for devising theoretically efficient algorithms for optimizing a linear function over the set. For any such barrier, the ball of radius 1 at any point defined by the local norm given by the Hessian of the barrier function is always contained in the set. At the analytic center, the point that minimizes the barrier (assuming it exists), a corresponding ball of radius a constant times the so-called complexity value of the barrier (n for the function f above) contains the convex set. Hence the metric defined by the local norm gives a very useful approximation to the local geometry of the set. For our interests, this gives a differential geometry view of a polyhedron as compared to the combinatorial geometry viewpoint of the simplex method.

Let us now return to our strictly feasible point $\bar{y}$ and the metric defined by the matrix $A\bar{S}^{-2}A^T$. With respect to this metric, the steepest ascent direction for the objective function is

$$d_{AFF} := (A\bar{S}^{-2}A^T)^{-1}b \tag{9}$$

(the affine-scaling direction), while the steepest descent direction for the logarithmic barrier function is

$$d_{CEN} := -(A\bar{S}^{-2}A^T)^{-1}A\bar{s}^{-1} \tag{10}$$

(the centering direction), where $\bar{s}^{-1}$ denotes the vector whose components are the reciprocals of those of $\bar{s}$. Dual interior-point methods generally choose as search direction a linear combination of these two directions.

Strictly feasible points $\bar{y}$ where these two directions are diametrically opposed (so that $\mu A\bar{s}^{-1} = b$ for some positive μ) lie on the so-called *dual central path*. Such points maximize the dual penalized function $b^T y - \mu f(y)$ over strictly feasible y. Note that then $\bar{x} := \mu\bar{s}^{-1}$ is a strictly feasible point for (P), and it is not hard to see that $\bar{x}$ minimizes the primal penalized function $c^T x - \mu\sum_j \ln x_j$ over strictly feasible x; $\bar{x}$ is then a point on the *primal central path*. Together $\bar{x}$, $\bar{y}$, and $\bar{s}$ solve

$$\begin{aligned} A^T y + s &= c, \quad s > 0, \\ Ax \qquad &= b, \quad x > 0, \\ x \circ s \qquad &= \mu e, \end{aligned} \tag{11}$$

where $x \circ s$ is the vector of component-wise products of x and s and e is the n-vector of ones.

Note that points satisfying (11) are strictly feasible, with duality gap

$$c^T x - b^T y = x^T s = n\mu.$$

So as μ tends to zero from above, points on the central paths converge to optimal solutions (this requires additional work to show that limits exist). It thus makes sense to approximately follow the central path(s) as μ decreases.

We say that $\bar{y}$ is close to the dual central path if for some $\mu > 0$, the Newton step for the penalized function, $\mu^{-1}d_{AFF} + d_{CEN}$, is below some tolerance δ in the local norm, i.e.,

$$[(b - \mu A\bar{s}^{-1})^T[A\bar{S}^{-2}A^T]^{-1}(b - \mu A\bar{s}^{-1})]^{\frac{1}{2}} \leq \delta\mu.$$

We then try to find such a point for a smaller value of μ, say $\sigma\mu$ for $0 < \sigma < 1$, by taking a Newton step. It turns out that we can choose σ as $1 - O(1/\sqrt{n})$ and guarantee that we stay close to the central path. Iterating this procedure gives an $O(\sqrt{n}\ln(1/\varepsilon))$-iteration algorithm to obtain an ε-optimal solution. Such methods were first developed by Renegar [80].

Note that every iteration requires the solution of a linear system with coefficient matrix $A\bar{S}^{-2}A^T$. This compares nicely with the ellipsoid method (see (6)) except that now every diagonal entry of $\bar{S}^{-2}$ changes, whereas only one entry of D_k changed before. To compensate, far fewer iterations are typically necessary in practical interior-point methods.

5.2. Primal-dual path-following

We have gone into some detail concerning dual path-following methods, in order to highlight their motivation from a special local metric defined on the interior of the dual feasible region with very attractive properties. We shall be much more brief here.

Of course, we can define primal path-following methods by following the arguments of the previous subsection: these require the solution of a linear system with coefficient matrix $A\bar{X}^2A^T$ at each iteration, where $\bar{X}$ is the diagonal matrix containing the components of the current iterate $\bar{x}$.

Primal-dual methods are usually motivated by considering the system of nonlinear equalities (11). Given a strictly feasible triple $(\bar{x}, \bar{y}, \bar{s})$ that is close to the central path (i.e., the Euclidean norm of $\mu^{-1}x \circ s - e$ is small), we move in the direction of the Newton step for (11) with μ replaced by $\sigma\mu$. These methods are admirably discussed and analyzed in Gonzaga [44] and Wright [97].

There is a another way we can view these methods. If we wanted to simultaneously increase the dual penalized function and decrease the primal penalized function, we could solve two linear systems with coefficient matrices $A\bar{S}^{-2}A^T$ and $A\bar{X}^2A^T$. These can be viewed as Newton steps or alternatively as steepest descent steps with respect to the local norms. To avoid the work of solving two linear systems, we can use steepest descent with respect to the local norm at a point intermediate between $\bar{x}$ and $\bar{s}^{-1}$, their geometric mean $\bar{x}^{\frac{1}{2}}\bar{s}^{-\frac{1}{2}}$ (interpreted component-wise). This leads to requiring the solution of just a single linear system to obtain the steepest descent directions for both penalized functions, and the resulting search directions are exactly those found by the primal-dual Newton approach of the previous paragraph.

5.3. Potential-reduction methods

Karmarkar used the primal function

$$\phi_P(x) := n \ln(c^T x - z_*) - \sum_j \ln x_j,$$

where z_* is the known optimal value of (P), to monitor progress of his projective algorithm [56], and showed that this function could be decreased by a constant at each iteration. Later research removed the assumption that the optimal value is known, but it seems better to use instead the primal-dual potential function

$$\Phi_{PD}(x, y) := (n + \rho) \ln(c^T x - b^T y) - \sum_j \ln x_j - \sum_j \ln(c - A^T y)_j,$$

defined for strictly feasible pairs, where ρ is a nonnegative parameter. The latter function was introduced independently by Tanabe [86] and by Todd and Ye [92]; [92] (in a restricted setting) and then Ye [98] and Kojima, Mizuno, and Yoshise [63] proved that, as long as $\rho \geq \sqrt{n}$, it can be reduced by a constant at each iteration.

In both primal and primal-dual guises, this constant reduction suffices to obtain polynomial-time bounds on the number of iterations required to obtain ε-optimal solutions, of $O(n \ln(\frac{1}{\varepsilon}))$ and $O(\rho \ln(\frac{1}{\varepsilon}))$ respectively. Hence the primal-dual method can attain the same bound as path-following methods by choosing $\rho = \theta(\sqrt{n})$. More significantly, these bounds can be achieved without any restrictions on the iterates remaining close to the central path: as long as a sufficient reduction in the potential function is maintained, the iteration bound stays valid.

The symmetric primal-dual search directions used by Kojima et al., while motivated by scaled steepest descent for the potential function, turn out to coincide with those arising from the path-following approach with $\sigma = n/(n+\rho)$. Thus choosing $\rho = \theta(\sqrt{n})$ corresponds to choosing $\sigma = 1 - \theta(1/\sqrt{n})$.

Modern implementation of interior-point methods are usually called path-following variants, although there is rarely an attempt to maintain the iterates in any neighborhood of the central path. The parameter σ is typically chosen adaptively, often very close to zero. Apart from the adaptive choice of σ, these methods could just as easily be viewed as potential-reduction methods which do not check that the potential function is actually reduced.

As a final remark on potential-reduction methods, note that, for the more general area of semidefinite programming, Benson et al. [9,10] have shown that dual potential-reduction methods can exploit the structure of certain classes of problems more effectively than path-following methods.

5.4. Exponential gaps

To conclude this section, I want to point out an interesting parallel between worst-case and typical behaviors of the simplex method and interior-point methods. As I have indicated, the best bound we have on the number of steps of the (deterministic) simplex method grows exponentially with the dimension, while its usual behavior is linear. Moreover, there are examples showing that the gap is real: for many simplex pivot rules, there are examples where the number of pivots required is exponential.

For interior-point methods, the best bounds are polynomial in the dimension, growing with n or $\sqrt{n}$. Due to the cost of each iteration, such growth would be disastrous in practical cases, and thus it is fortunate that the observed growth is much slower — perhaps logarithmic in the dimension (see Lustig et al. [72]). This is again an exponential gap! The question arises as to whether it is “real,” or just an artefact of our inability to prove a better bound. The answer is that the gap is indeed real: for many practical interior-point methods, there are examples showing that the number of iterations grows at least as fast as $\Omega(n^{\frac{1}{3}})$; see Todd and Ye [93].

6. Other methods

Here we collect some miscellaneous methods that have been proposed for linear programming, and provide some references. Despite some encouraging results quoted in some of these papers (often on small random instances), none is regarded as competitive with simplex or interior-point methods (but we should note the excellent results of De Leone and Mangasarian [32] using an SOR (successive over-relaxation) method on very large sparse random problems in 1987).

6.1. Gradient-like methods

These try to follow projected gradient directions from one feasible point to another. Since this seems like such a natural strategy, and apparently so obviously superior to edge-following methods, such methods have been proposed numerous times in slightly varying guises. Indeed, Brown and Koopmans discussed such a method [20] in the same volume that contained Dantzig’s initial publication [27]. Later references include Zoutendijk’s method of feasible directions ([101], Chapter 9), Rosen's projected gradient method [82], the constrained gradient method of J. K. Thurber and Lemke [70], and Chang and Murty's steepest descent gravitational method [23].

6.2. Fictitious play

An iterative method to find optimal strategies and the value of 2-person zero-sum games was devised by Brown [19] and shown to be convergent by Robinson [81]. This method is iterative, and at each stage selects a strategy for each player that is a best response to the empirical mixed strategy exhibited by the previous choices of his or her opponent. Convergence is slow, but the method is very simple to implement. Since linear programming problems can be reduced to the solution of (symmetric) games, this provides a method for general LP problems also.

The idea behind the method can also be based on differential equations, and Brown and von Neumann prove the existence of the value and of optimal strategies based on a system of differential equations in [21]. The proof is semi-constructive, in that no particular technique for solving the resulting initial value problem is mentioned. Apparently the paper is based on results obtained independently by the two authors. This similarity in the approaches is perhaps the reason that Dantzig states in [30] that Hoffman et al. [49] compared a scheme of von Neumann to the simplex method and relaxation methods, whereas their comparison is of Brown's fictitious play method.

A much-cited note of von Neumann, in which he proposed another method for linear programming soon after Dantzig visited him in 1947, eventually appeared in his collected works [95]. Again, a system of differential equations is proposed, but the details of a specific algorithm are not given. The approaches of Brown and of von Neumann are very similar, but while Brown adjusts the weight on just one strategy (and changes the others proportionately), von Neumann adjusts the weights on many of the strategies based on a residual vector.

Finally, a more refined analysis of a related method appears in von Neumann [94]. Again the emphasis is on solving a zero-sum game, but here an explicit bound on the computational complexity (the first in mathematical programming?) is given: to approximate the value of an $m \times n$ matrix game to within a multiple ε of the range of payoffs requires at most $\frac{m+n}{\varepsilon^2}$ iterations, each requiring about $4mn$ flops. If only the $\frac{1}{\varepsilon^2}$ factor could be replaced by $\ln(\frac{1}{\varepsilon})$, this would yield a polynomial algorithm!

6.3. Relaxation methods

The classical relaxation method of Agmon [4] and Motzkin and Schoenberg [76], closely related to iterative methods for linear equations, was of great interest in the 1950s, partly because each iteration requires minimal computation, merely some matrix-vector multiplications. To find a point in $Y := \{y : A^T y \leq c\}$, at every iteration the current point is projected onto the hyperplane defined by a violated constraint (actually, a stepsize $\lambda \in (0, 2]$ is usually included, where $\lambda = 1$ (respectively, 2) corresponds to projection onto (respectively, reflection in) the hyperplane). This method is closely related to the classical Gauss-Seidel and Jacobi iterative methods for "solving" the system $A^T Av = c$ of linear equations.

(Here, at each iteration, one component of v is changed to satisfy the corresponding equation, again usually with a relaxation parameter to speed convergence.) However, it did not perform well in the computational experiments of Hoffman et al. [49] due to its very slow convergence. The appearance of the ellipsoid method (which can be viewed as a variable-metric form of the relaxation method) prompted some further study: see Goffin [42,43].

Much more successful were variants of the successive over-relaxation (SOR) technique applied to either a linear complementarity form of the problem or to an augmented Lagrangian dual problem. Indeed, De Leone and Mangasarian [32] report very encouraging results for this method applied to random sparse linear programming problems of dimensions from $25,000 \times 100,000$ up to $125,000 \times 500,000$. Once again, each iteration requires only some matrix-vector multiplications with the original sparse coefficient matrix. However, choosing the augmented Lagrangian penalty parameter presents some problems.

6.4. Methods based on ideas of computational geometry

Here we collect a number of methods motivated by two concerns: the desire for a strongly polynomial linear programming algorithm (both the ellipsoid and interior-point methods have iteration bounds depending on the bit length of the data, not just its dimension), and the interest in solving low-dimensional problems with a large number of constraints.

Megiddo was the first to obtain a method that is linear in n for fixed dimension d [74]. However, the dependence on d was doubly exponential. Clarkson [26] devised a randomized algorithm needing only $O(d^2 n + d^4 \sqrt{n} \ln n + [O(d)]^{d/2+O(1)} \ln n)$ arithmetic operations. One key idea of Clarkson was to take a sample of the constraints, solve the resulting smaller problem (by another algorithm), and find the set of constraints violated by its solution. These constraints are then either forced into the next sample, or given increased probability, and the process is continued.

Gärtner and Welzl [38] describe an $O(d^2 n + \exp(K\sqrt{d \ln d}))$-operation algorithm that combines two methods of Clarkson with the subexponential pivoting methods of Kalai [52] and Matousek et al. [73]. This paper also gives a nice historical review of this class of methods.

7. The future

What will we see in the next fifty, or even five, years? Linear programming has a history of reinventing itself. At present there is a rough computational parity between simplex and interior-point methods: variants of the simplex method are usually better for small problems, hold a significant edge in post-optimal analysis, and are more effective for some large-scale problems, while interior-point methods hold sway for other large problems. Will our complacency in the status quo be shattered by another computationally effective class of methods? I wouldn't bet on it in the next five years, but over the next ten, I'd take even odds.

On the theoretical side we still have the big questions: does the bounded Hirsch conjecture hold? Is there a polynomial pivot rule for the simplex method?

For interior-point methods, can we give a theoretical explanation for the difference between worst-case bounds and observed practical performance? Can we devise an algorithm whose iteration complexity is better than $O(\sqrt{n}\ln(1/\varepsilon))$ to attain ε-optimality? Can we find a theoretically or practically efficient way to reoptimize?

Let us hope that the next fifty years brings as much excitement as the last!

Acknowledgements. Research supported in part by NSF through grant DMS-9805602 and ONR through grant N00014-96-1-0050

References

1. I. Adler. The expected number of pivots needed to solve parametric linear programs and the efficiency of the self-dual simplex method. Technical report, Department of Industrial Engineering and Operations research, University of California, Berkeley, CA, 1983.
2. I. Adler, R. M. Karp, and R. Shamir. A simplex variant solving an $m \times d$ linear program in $O(\min(m^2, d^2))$ expected number of steps. *Journal of Complexity*, 3:372–387, 1987.
3. I. Adler and N. Megiddo. A simplex algorithm where the average number of steps is bounded between two quadratic functions of the smaller dimension. *Journal of the ACM*, 32:871–895, 1985.
4. S. Agmon. The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6:382–392, 1954.
5. K. M. Anstreicher. Potential reduction algorithms. In T. Terlaky, editor, *Interior Point Methods in Mathematical Programming.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
6. E. Balas and M. Padberg. On the set covering problem. *Operations Research*, 20:1152–1161, 1972.
7. M. L. Balinski. The Hirsch conjecture for dual transportation polyhedra. *Mathematics of Operations Research*, 4:629–633, 1984.

8. M. L. Balinski. Mathematical programming: Journal, society, recollections. In J. K. Lenstra, A. H. G. Rinnooy Kan, and A. Schrijver, editors, *History of Mathematical Programming*, pages 5–18. Elsevier Science Publishers, 1991.

9. S. J. Benson, Y. Ye, and X. Zhang. Mixed linear and semidefinite programming for combinatorial and quadratic optimization. *Optimization Methods and Software*, 11:515–544, 1999.

10. S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

11. R. E. Bixby. Implementing the simplex method: The initial basis. *ORSA Journal on Computing*, 4:267–284, 1992.

12. R. E. Bixby. Progress in linear programming. *ORSA Journal on Computing*, 6(1):15–22, 1994.

13. R. E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling. MIP: Theory and practice – closing the gap. In M. J. D. Powell and S. Scholtes, editors, *System Modelling and Optimization: Methods, Theory, and Applications*, pages 19–49. Kluwer Academic Publishers, 2000.

14. R. Bland, D. Goldfarb, and M. Todd. The ellipsoid method: A survey. *Operations Research*, 29:1039–1091, 1981.

15. K. Borgwardt. The average number of pivot steps required by the simplex method is polynomial. *Zeitschrift für Operations Research*, 26:155–177, 1982.

16. K. H. Borgwardt. *The Simplex Method: A Probabilistic Analysis.* Springer-Verlag, Berlin, 1986.

17. K. H. Borgwardt. Probabilistic analysis of the simplex method. In J. C. Lagarias and M. J. Todd, editors, *Mathematical Developments Arising from Linear Programming.* American Mathematical Society, 1990.

18. K. H. Borgwardt. A sharp upper bound for the expected number of shadow vertices in LP-polyhedra under orthogonal projection on two-dimensional planes. *Mathematics of Operations Research*, 24(3):544–603, 1999. A corrected version of Figure 1 appears in Volume 24(4), pp. 925–984.

19. G. W. Brown. Iterative solution of games by fictitious play. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 374–376. Wiley, New York, 1951.

20. G. W. Brown and T. C. Koopmans. Computational suggestions for maximizing a linear function subject to linear inequalities. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 377–380. Wiley, New York, 1951.

21. G. W. Brown and J. von Neumann. Solutions of games by differential equations. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, pages 73–79. Princeton University Press, Princeton, 1950.

22. B. P. Burrell and M. J. Todd. The ellipsoid method generates dual variables. *Mathematics of Operations Research*, 10:688–700, 1985.

23. S. Y. Chang and K. G. Murty. The steepest descent gravitational method for linear programming. *Discrete Applied Mathematics*, 25:211–239, 1989.

24. A. Charnes and W. W. Cooper. On some works of Kantorovich, Koopmans and others. *Management Science*, 8:246–263, 1962.

25. T. Christof, M. Jünger, and G. Reinelt. A complete description of the traveling salesman polytope on 8 nodes. *Operations Research Letters*, 10:497–500, 1991.

26. K. L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. *Journal of the ACM*, 42:488–499, 1995.

27. G. B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 339–347. Wiley, New York, 1951.

28. G. B. Dantzig. Upper bounds, secondary constraints, and block triangularity in linear programming. *Econometrica*, 23:174–183, 1955.

29. G. B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, NJ, 1963.

30. G. B. Dantzig. Linear programming. In J. K. Lenstra, A. H. G. Rinnooy Kan, and A. Schrijver, editors, *History of Mathematical Programming*, pages 19–31. Elsevier Science Publishers, 1991.

31. G. B. Dantzig and P. Wolfe. The decomposition algorithm for linear programming. *Econometrica*, 29:767–778, 1961.

32. R. De Leone and O. L. Mangasarian. Serial and parallel solution of large scale linear programs by augmented Lagrangian successive overrelaxation. In A. Kurzhanski, K. Neumann, and D. Pallaschke, editors, *Optimization, Parallel Processing and Appli-*

cations, pages 103–124, Berlin, 1988. Springer-Verlag. Lecture Notes in Economics and Mathematical Systems 304.

33. I. I. Dikin. Iterative solution of problems of linear and quadratic programming (in Russian). *Doklady Akademiia Nauk SSSR*, 174:747–748, 1967. English Translation: *Soviet Mathematics Doklady*, 1967, Volume 8, pp. 674–675.

34. I. I. Dikin. On the speed of an iterative process (in Russian). *Upravlaemye Sistemy*, 12:54–60, 1974.

35. A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, 1968. Reprinted as Volume 4 of the SIAM Classics in Applied Mathematics Series, 1990.

36. J. J. Forrest and D. Goldfarb. Steepest-edge simplex algorithms for linear programming. *Mathematical Programming*, 57:341–374, 1992.

37. R. A. K. Frisch. The logarithmic potential method of convex programming. Technical report, University Institute of Economics, Oslo, Norway, 1955.

38. B. Gärtner and E. Welzl. Linear programming — randomization and abstract frameworks. In *Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science 1064*, pages 669–687. Springer-Verlag, Berlin, Germany, 1996.

39. D. M. Gay. Electronic mail distribution of linear programming test problems. *Mathematical Programming Society Committee on Algorithms Newsletter*, 13:10–12, 1985. Problems available from `ftp://netlib2.cs.utk.edu/lp/data/`.

40. P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright. On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Mathematical Programming*, 36:183–209, 1986.

41. M. X. Goemans. Worst-case comparison of valid inequalities for the TSP. *Mathematical Programming*, 69:335–349, 1995.

42. J.-L. Goffin. The relaxation method for solving systems of linear inequalities. *Mathematics of Operations Research*, 5:388–414, 1980.

43. J.-L. Goffin. Variable metric relaxation methods, Part II: The ellipsoid method. *Mathematical Programming*, 30:147–162, 1984.

44. C. C. Gonzaga. Path following methods for linear programming. *SIAM Review*, 34(2):167–224, 1992.

45. M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, Germany, 1988.

46. B. Grunbaum. *Convex Polytopes*. Wiley, New York, 1967.

47. M. Haimovich. The simplex method is very good! On the expected number of pivot steps and related properties of random linear programs. Technical report, Columbia University, 1983.

48. A. J. Hoffman. Linear programming at the National Bureau of Standards. In J. K. Lenstra, A. H. G. Rinnooy Kan, and A. Schrijver, editors, *History of Mathematical Programming*, pages 62–64. Elsevier Science Publishers, 1991.

49. A. J. Hoffman, M. Mannos, D. Sokolowsky, and N. Wiegmann. Computational experience in solving linear programs. *Journal of SIAM*, 1:17–33, 1953.

50. How to make a CD disco ball. See `http://www.ehow.com/eHow/eHow/0,1053,4570,00.html`.

51. Image of a quartz crystal. See `http://www.halcyon.com/nemain/gallery/Quartz_43.html`.

52. G. Kalai. A subexponential randomized simplex algorithm. In *Proceedings of the 24th ACM Symposium on Theory of Computing*, pages 475–482, 1992.

53. G. Kalai. Linear programming, the simplex algorithm and simple polytopes. *Mathematical Programming*, 79:217–233, 1997.

54. G. Kalai and D. J. Kleitman. A quasi-polynomial bound for diameter of graphs of polyhedra. *Bulletin of the American Mathematical Society*, 24:315–316, 1992.

55. L. V. Kantorovich. Mathematical methods in the organization and planning of production (in Russian). *Publication House of the Leningrad State University*, 1939. 68 pp. English Translation: *Management Science*, Volume 6 (1960), pp. 363–422.

56. N. K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

57. L. G. Khachiyan. A polynomial algorithm in linear programming (in Russian). *Doklady Akademiia Nauk SSSR*, 224:1093–1096, 1979. English Translation: *Soviet Mathematics Doklady*, Volume 20, pp. 191–194.

58. L. G. Khachiyan. Polynomial algorithms in linear programming (in Russian). *Zhurnal Vychisitel'noi Matematiki i Matematicheskoi Fiziki*, 20:51–68, 1980. English Translation: *U.S.S.R. Computational Mathematics and Mathematical Physics*, Volume 20, pp. 53–72.

59. V. Klee and P. Kleinschmidt. The d-step conjecture and its relatives. *Mathematics of Operations Research*, 12:718–755, 1987.

60. V. Klee and G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities, III*, pages 159–175. Academic Press, 1972.

61. V. Klee and D. W. Walkup. The d-step conjecture for polyhedra of dimension $d < 6$. *Acta Mathematica*, 133:53–78, 1967.

62. P. Kleinschmidt. The diameter of polytopes and related applications. In T. Bisztriczky, P. McMullen, R. Schneider, and A. I. Weiss, editors, *Polytopes: Abstract, Convex, and Computational*, pages 467–492. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

63. M. Kojima, S. Mizuno, and A. Yoshise. An $O(\sqrt{n}L)$-iteration potential reduction algorithm for linear complementarity problems. *Mathematical Programming*, 50:331–342, 1991.

64. T. C. Koopmans, editor. *Activity Analysis of Production and Allocation*, New York, 1951. Wiley.

65. H. W. Kuhn. On the origin of the Hungarian method. In J. K. Lenstra, A. H. G. Rinnooy Kan, and A. Schrijver, editors, *History of Mathematical Programming*, pages 77–81. Elsevier Science Publishers, 1991.

66. H. W. Kuhn and R. E. Quandt. An experimental study of the simplex method. In N. C. Metropolis et al., editor, *Experimental Arithmetic, High-Speed Computing and Mathematics*, pages 107–124. American Mathematical Society, 1963.

67. D. G. Larman. Paths on polytopes. *Proceedings of the London Mathematical Society*, 20:161–178, 1970.

68. E. Lawler. The great mathematical Sputnik of 1979. *The Mathematical Intelligencer*, 2:190–198, 1980.

69. C. E. Lemke. The dual method of solving the linear programming problem. *Naval Research Logistics Quarterly*, 1:36–47, 1954.

70. C. E. Lemke. The constrained gradient method of linear programming. *Journal of SIAM*, 9:1–17, 1961.

71. A. Liao and M. J. Todd. Solving LP problems via weighted centers. *SIAM Journal on Optimization*, 6:933–960, 1996.

72. I. J. Lustig, R. E. Marsten, and D. F. Shanno. The primal-dual interior point method on the Cray supercomputer. In T. F. Coleman and Y. Li, editors, *Large-Scale Numerical Optimization*, pages 70–80. SIAM, 1990.

73. J. Matousek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.

74. N. Megiddo. Linear programming in linear time when the dimension is fixed. *Journal of the Association for Computing Machinery*, 31(1):114–127, 1984.

75. N. Megiddo. On the complexity of linear programming. In T. Bewley, editor, *Advances in Economic Theory*, pages 225–268. Cambridge University Press, Cambridge, 1987.

76. T. S. Motzkin and I. J. Schoenberg. The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6:393–404, 1954.

77. D. Naddef. The Hirsch conjecture is true for (0,1)-polytopes. *Mathematical Programming*, 45:109–110, 1989.

78. Y. E. Nesterov and A. S. Nemirovski. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms*. SIAM Publications. SIAM, Philadelphia, USA, 1993.

79. M. W. Padberg and M. R. Rao. The traveling salesman problem and a class of polyhedra of diameter two. *Mathematical Programming*, 7:32–45, 1974.

80. J. Renegar. A polynomial-time algorithm based on Newton's method for linear programming. *Mathematical Programming*, 40:59–93, 1988.

81. J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:296–301, 1951.

82. J. B. Rosen. The gradient projection method for nonlinear programming, Part I: Linear constraints. *Journal of SIAM*, 8:181–217, 1961.

83. A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley, Chichester, 1986.

84. N. Z. Shor. Cut-off method with space extension in convex programming problems (in Russian). *Kibernetika*, 13:94–95, 1977. English translation: *Cybernetics* 13(1), 94–96.

85. S. Smale. On the average speed of the simplex method in linear programming. *Mathematical Programming*, 27:241–262, 1983.

86. K. Tanabe. Centered Newton method for mathematical programming. In M. Iri and K. Yajima, editors, *System Modelling and Optimization: Proceedings of the 13th IFIP Conference, Tokyo, Japan, Aug./Sept. 1987*, pages 197–206. Springer-Verlag, Berlin, Germany, 1988. Volume 113 of *Lecture Notes in Control and Information Sciences*.

87. M. J. Todd. The monotonic bounded Hirsch conjecture is false for dimension at least four. *Mathematics of Operations Research*, 5:599–601, 1980.

88. M. J. Todd. Polynomial expected behavior of a pivoting algorithm for linear complementarity and linear programming problems. *Mathematical Programming*, 35:173–192, 1986.

89. M. J. Todd. Polynomial algorithms for linear programming. In H. A. Eiselt and G. Pederzoli, editors, *Advances in Optimization and Control*, pages 49–66. Springer-Verlag, Berlin, 1988.

90. M. J. Todd. Probabilistic models for linear programming. *Mathematics of Operations Research*, 16:671–693, 1991. Erratum: see *Mathematics of Operations Research* 23 (1998), pp. 767–768.

91. M. J. Todd. Potential-reduction methods in mathematical programming. *Mathematical Programming*, 76:3–45, 1997.

92. M. J. Todd and Y. Ye. A centered projective algorithm for linear programming. *Mathematics of Operations Research*, 15(3):508–529, 1990.

93. M. J. Todd and Y. Ye. A lower bound on the number of iterations of long-step and polynomial interior-point methods for linear programming. *Annals of Operations Research*, 62:233–252, 1996.

94. J. von Neumann. A numerical method to determine optimal strategy. *Naval Research Logistics Quarterly*, 1:109–115, 1954.

95. J. von Neumann. Discussion of a maximum problem. In A. H. Taub, editor, *John von Neumann: Collected Works, Volume VI*, pages 89–95. Pergamon Press, Oxford, 1963.

96. P. Wolfe and L. Cutler. Experiments in linear programming. In R. L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 177–200. McGraw-Hill, New York, 1963.

97. S. Wright. *Primal-dual Interior Point Methods*. SIAM, Philadelphia, 1996.

98. Y. Ye. An $O(n^3L)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 50(2):239–258, 1991.

99. D. B. Yudin and A. S. Nemirovski. Informational complexity and efficient methods for the solution of convex extremal problems (in Russian). *Ékonomika i Matematicheskie metody*, 12:357–369, 1976. English translation: *Matekon* 13(2), 3–25.

100. G. Ziegler. *Lectures on Polytopes*. Springer-Verlag, Berlin, 1995.

101. G. Zoutendijk. *Methods of Feasible Directions*. Elsevier, Amsterdam, 1960.