Linear Programming
(For the Encyclopedia of Microcomputers)

Nimrod Megiddo*

June 1991

Linear programming is one of the most successful disciplines within the field of operations research. In its *standard form*, the linear programming problem calls for finding nonnegative $x_1, \ldots, x_n$ so as to maximize a linear function $\sum_{j=1}^{n} c_j x_j$ subject to a system of linear equations:

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m \ . \end{aligned}$$

This problem can be stated in vector notation as

$$\begin{array}{rl} \text{Maximize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

where $\boldsymbol{A} \in R^{m \times n}$ is assumed to have linearly independent rows, and $\boldsymbol{b} \in R^m$ and $\boldsymbol{c}, \boldsymbol{x} \in R^n$. In fact, any problem of maximizing or minimizing a linear function subject to linear equations and inequalities can be easily reduced to the standard form.

The *dual* problem of the linear programming problem in standard form is

$$\begin{array}{rl} \text{Minimize} & \boldsymbol{b}^T\boldsymbol{y} \\ \text{subject to} & \boldsymbol{A}^T\boldsymbol{y} \geq \boldsymbol{c} \ . \end{array}$$

The former problem is then referred to as the *primal*. The duality theorem asserts that (i) for any $\boldsymbol{x}$ that satisfies the constraints of the primal and for any $\boldsymbol{y}$ that satisfies the conditions of the dual, $\boldsymbol{c}^T\boldsymbol{x} \leq \boldsymbol{b}^T\boldsymbol{y}$, and (ii) if there exist such $\boldsymbol{x}$ and $\boldsymbol{y}$, then the maximum of the primal equals the minimum of the dual. The duality theorem plays a central role in the theory of linear programming.

*IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120-6099, and School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel.

The linear programming model has been applied in a large number of areas including military applications, transportation and distribution, scheduling, production and inventory management, telecommunication, agriculture and more. Many problems simply lend themselves to a linear programming solution but in many cases some ingenuity is required for the modelling. Linear programming also has interesting theoretical applications in combinatorial optimization and complexity theory.

The classical tool for solving the linear programming problem in practice is the class of simplex algorithms proposed and developed by George Dantzig [13]. The method is based on generating a sequence of bases. A *basis* is a nonsingular submatrix of $\boldsymbol{A}$ of order $m \times m$. The fundamental characteristic of the method is that at some point a basis is reached which provides a solution to the problem. A suitable basis can certify either that the problem has no solution at all or that it is unbounded; otherwise, a basis will be reached which defines optimal solutions for both the primal and the dual problems. Due to the problems of degeneracy, special care is needed to guarantee that the method will not cycle. Bland [7] proved that the "least index" rules guarantees that.

Recently, methods of nonlinear programming methods have also become practical tools for certain classes of linear programming problems.

The computational complexity of linear programming had puzzled researchers even before the field of computational complexity started to develop. The question of finding bounds on the diameter and height of polytopes (see [21]) is closely related to the complexity of the simplex method.

Added text: Kalai and Kleitman [27] recently proved that the diameter of d-dimensional polyhedron with n facets is bounded by $n^{\log_2 d+1}$. Kalai [23] proved that the height of such a polytope is bounded by $n\binom{d+\log_2 n}{d}$.

Practitioners have noticed long ago that the simplex algorithms performed surprisingly well. In particular, the number of iterations seemed linear in the number of rows m and sublinear in the number of columns n in the standard form. Klee and Minty [26], however, found examples where the number of iterations performed by certain variants of the method was exponential.

The area of computational complexity was developed mainly during the 1970's and 1980's. Within this field, the question of the complexity of linear programming was given a new meaning. Complexity theorists were interested in the relation between the running time measured in bit operations and the length L of the representation of the problem with integer data in binary encoding. Khachiyan [25] was the first to show that the linear programming problem was in the class P, that is, it could be solved in time polynomial in the length of the binary encoding of the input. Khachiyan's result was based on the ellipsoid algorithm which had been first proposed by Shor [42] for general convex programming. In the ellipsoid method the problem is reduce to finding a solution to a system of strict inequalities $\boldsymbol{Ax} > \boldsymbol{b}$ whose set of solutions is bounded. It generates a sequence of ellipsoids each of which is guaranteed to contain all the solutions. If a center of any ellipsoid in this sequence is a solution, then it is discovered. Otherwise the process stops when the volume of the current ellipsoid is too small to contain all the solutions if there exist any, so the conclusion in this case is that there are no solutions. For survey on the ellipsoid method see Bland, Goldfarb and Todd [8]. Grötschel, Lovász and Schrijver [20] developed a beautiful theory based on the ellipsoid algorithm and derived complexity results with regard to many problems of combinatorial optimization.

Theorists were quite disappointed when it became clear that the ellipsoid algorithm was not useful for solving linear programming problems in practice. The contrast between the ellipsoid and the simplex methods gave an excellent example that theory could not always be relied upon for predicting applicability. Variants of the simplex method, which were proven to be exponential in the worst case, were very efficient in practice, while the polynomial ellipsoid method was very inefficient. With this observation, researchers became interested in analyzing the behavior of the simplex method from a more practical point of view and at the same time were also searching for other methods of solving the problem.

Breakthroughs in the analysis of the simplex method were made by Borgwardt (see [9] for the original references) and Smale [43]. The former was the first to show that under a certain probabilistic model of distribution of inputs, a certain variant of the method runs in polynomial time. Subsequently, under a different model, a bound of $O(m^2)$ was proven by Todd [46], Adler and Megiddo [2], and Adler, Karp and Shamir [1].

A new polynomial-time algorithm for linear programming was proposed by Karmarkar [24] in 1984. His algorithm works on the problem in the form:

$$\begin{aligned} \text{Minimize} \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \\ & \boldsymbol{e}^T \boldsymbol{x} = 1 \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

where $\boldsymbol{e} = (1, \ldots, 1)^T \in R^n$, assuming (without loss of generality) that the minimum equals 0 and $\boldsymbol{A}\boldsymbol{e} = \boldsymbol{0}$. Karmarkar's algorithm generates a sequence of points in the interior of the domain defined by the constraints which converges to an optimal point. The algorithm is based on repeated centering by a projective scaling transformation. Given the current iterate $\boldsymbol{x}^k$, the transformation maps any vector $\boldsymbol{x} \in R^n$ such that $\boldsymbol{e}^T\boldsymbol{x} = 1$ to a vector $\boldsymbol{x}' \in R^n$, where $x'_j = (x_j/x^k_j)/(1 + \sum_i x_i/x^k_i)$ $(j = 1, \ldots, n)$. Karmarkar's projective scaling algorithm provided an improved upper bound on the complexity of linear programming under the bit operations model. It runs in $O(nL)$ iterations and was reported to be very practical, but most of the computational experience was done with a simplified version of it, called *primal affine scaling*. Here the scaling transformation is simply $x'_j = x_j/x^k_j$. The primal affine scaling algorithm is not believed to run in polynomial time in the worst case. It was proposed by many people independently (e.g., Barnes [6], Vanderbei, Meketon and Freedman [48]) Later, however, it was discovered that the same primal affine scaling algorithm was proposed by Dikin [14] in the 1960's. Gill, Murray, Saunders, Tomlin and Wright [17] established the connection between Karmarkar's algorithm and the classical logarithmic barrier function method of nonlinear programming. Dual versions and primal-dual versions of the affine scaling method have also been studied.

After the publication of Karmarkar's algorithm, researchers developed many algorithms inspired by different features of that algorithm and its analysis. Renegar [38] proposed a path following algorithm which runs in $O(\sqrt{n}L)$ iterations. Renegar solves the problem in the form of the dual. The path he follows consists of the minimizers of the functions

$$F_\mu(\boldsymbol{y}) = \boldsymbol{b}^T\boldsymbol{y} - \mu \sum_j \log(\boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{c})_j \ ,$$

where $\mu > 0$. This logarithmic barrier path is now referred to as the *path of centers* and can be defined analogously in the space of the primal problem. It was studied by Fiacco and McCormick [16] and more recently by Sonnevend [44].

Megiddo [37] proposed the primal-dual framework for following the central path, and specific primal-dual path following algorithms were subsequently proposed by Kojima, Mizuno and Yoshise [29] and by Adler and Monteiro [3].

Karmarkar used a "potential function" in his analysis and it was later realized that algorithms could be developed by operating directly on this potential function. Gonzaga [18] demonstrated that the problem could be solved in $O(nL)$ iterations simply by doing an affine scaling transformation and then searching the direction of the projected gradient of the function

$$\psi(\boldsymbol{x}) = (n + \sqrt{n}) \log \boldsymbol{c}^T\boldsymbol{x} - \sum_j \log x_j \ .$$

Added text: In [19] he also gave an algorithm which runs in a total number of $O(n^3L)$ arithmetic operations.

Potential reduction in primal-dual space also yields an $O(\sqrt{n}L)$ iteration algorithm as shown by Ye [49]. Vaidya [47] used fast matrix multiplication to further improve the complexity of the interior point methods for linear programming. For a unified theory of path following and potential reductin, see Kojima, Megiddo, Noma and Yoshise [28].

It is interesting, at least from a theoretical viewpoint, to settle the computational complexity of linear programming under different models of computation. The polynomial-time result holds only under the so-called logarithmic-cost model [4]. It is still an open question whether or not a system of linear inequalities can be solved in a number of arithmetic operations which is polynomially bounded by the dimensions of the system, independently of the magnitudes of the coefficients. Megiddo [31] gave an algorithm for solving systems of m inequalities and n variables, with at most two variables per inequality (whereas the general case can be reduced to at most three variables per inequality), which runs in a number arithmetic operations bounded by a polynomial in m and n. Such an algorithm is said to run in strongly polynomial time. Improved algorithms for this problem were recently proposed by Cohen and Megiddo [12],

Added text: and by Hochbaum and Naor [22].

The major result to date in the area of strongly polynomial algorithms is due to Tardos [45]. She proposed a general linear programming algorithm whose number of operations is independent of the magnitudes of coefficients in the objective function and the right-hand side vectors, but depends on the coefficients in the matrix $\boldsymbol{A}$. This implies that many combinatorial optimization problems, including the minimum-cost flow problem, can be solved in strongly polynomial time.

An interesting area of theoretical research in linear programming is the complexity of solving n linear inequalities in a fixed number d of variables. Megiddo [32] proved that this problem can be solved in linear time for any fixed d. The coefficient of proportionality was doubly exponential, 2^{2^d}, but this was later improved by Clarkson [10] and Dyer [15] to 3^{d^2}. Even more surprisingly, Clarkson [11] proposed randomized algorithms which solve this problem in expected linear time where the constant of proportionality is polynomial in d, but there is an additive constant which is exponential in d.

Added text: Further improvements were very recently proposed by Kalai [23] and Sharir and Welzl [41].

Seidel [40] gave a very simple randomized algorithm whose expected running time is $O(d!n)$. Alon and Megiddo [5] developed a parallel randomized algorithm for this problem, based on Clarkson's ideas, which runs with a linear number of processors almost surely in constant time.

The book by Schrijver [39] is a good source for the general theory of linear programming. For additional surveys see [34; 35]. In recent years articles on linear programming appear in *Mathematical Programming* and *Mathematics of Operations Research*. Collections of recent articles also appeared in [33; 36; 30].

References

[1] 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," *J. Complexity* **3** (1987) 372–387.

[2] I. Adler and N. Megiddo, "A simplex algorithm whose average number of steps is bounded between two quadratic functions of the smaller dimension," *J. ACM* **32** (1985) 871–895.

[3] I. Adler and R. D. C. Monteiro, "Interior path following primal-dual algorithms. Part I: Linear Programming," *Math. Programming* **44** (1989) 27–42.

[4] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms,* Addison-Wesley, Reading, Mass., 1974.

[5] N. Alon and N. Megiddo, "Parallel linear programming in fixed dimension almost surely in constant time," in: Proceedings of the 31st IEEE Symposium on Foundations of Computer Science (1990), IEEE Computer Society Press, Los Angeles, 1990, pp. 574–582.

[6] E. R. Barnes, "A variation of Karmarkar's algorithm for solving linear programming problems," *Math. Programming* **36** (1986) 174–182.

[7] R. G. Bland, "New finite pivoting rules for the simplex method," *Math. of O.R.* **2** (1977) 103–107. (1986) 174–182.

[8] R. G. Bland, R. G. Goldfarb and M. J. Todd, "The ellipsoid method: a survey," *Operations Research* **29** (1981) 1039–1091. (1986) 174–182.

[9] K. H. Borgwardt, *The Simplex Method – a probabilistic analysis*, Springer-Verlag, Berlin, Heildelberg, New York, 1987.

[10] K. L. Clarkson, "Linear programming in $O(n3^{d^2})$ time," *Information Processing Letters* **22** (1986 21–24.

[11] K. L. Clarkson, "Las Vegas algorithms for linear and integer programming when the dimension is small," in: Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science (1988), pp. 452–456.

[12] E. Cohen and N. Megiddo, "Improved algorithms for linear inequalities with two variables per inequality," in: Proceedings of the 23st Annual ACM Symposium on Theory of Computing (1991), ACM, New York, 1991, pp. 145–154.

[13] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, N.J., 1963.

[14] I. I. Dikin, "Iterative solution of problems of linear and quadratic programming," *Soviet Math. Dokl.* **8** (1967) 674–675.

[15] M. E. Dyer, "On a multidimensional search technique and its application to the euclidean one-center problem", *SIAM J. Comput.* **15** (1986) 725-738.

[16] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968.

[17] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin and M. H. Wright, "On projected Newton barrier methods for linear programming," Math. Programming **36** (1986) 183–209.

[18] C. C. Gonzaga, "Polynomial affine algorithms for linear programming," Math. Programming **49** (1990) 7–21.

[19] C. C. Gonzaga, "An algorithm for solving linear programming problems in $O(n^3L)$

operations," in [36], pp. 1–28.
[20] M. Grötschel, L. Lovász and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, Heildelberg, New York, 1988.
[21] B. Grünbaum, *Convex Polytopes*, Interscience-Wiley, London, 1967.
[22] D. S. Hochbaum and J. Naor, "Simple and fast algorithms for linear and integer programs with two variables per inequality," Technical Report, Dept. of IE/OR, U. C. Berkeley, Berkeley, California, 1991.
[23] G. Kalai, "A subexponentioal randomized simplex algorithm," unpublished manuscript, Institute of Mathematics, Hebrew University of Jerusalem, Jerusalem, Israel, October 1991.
[24] N. Karmarkar, "A new polynomial algorithm for linear programming," *Combinatorica* **4** (1984) 373–395.
[25] L. G. Khachiyan, "A polynomial algorithm in linear programming," *Soviet Math. Dokl.* **20** (1979) 191–194.
[26] V. Klee and G. J. minty, "How good is the simplex algorithm?", in: *Inequalities III* O. Shisha, Ed., Academic Press, New York, 1972, pp. 159–172.
[27] G. Kalai and D. J. Kleitman, "A quasi-polynomial bound for the diameter of graphs of polyhedra," Research Announcement, submitted to *Bull. AMS.*
[28] M. Kojima, N. Megiddo, T. Noma and A. Yoshise, *A unified approach to interior point algorithms for linear complementarity problems,* Lecture Notes in Computer Science, Springer-Verlag, New York, 1991.
[29] M. Kojima, S. Mizuno and A. Yoshise, "A polynomial-time algorithm for a class of linear complementarity problems, *Math. Programming* **44** (1989) 1–26.
[30] J. C. Lagarias and M. J. Todd, Eds., "Mathematical Developments Arising from Linear Programming," *Contemporary Mathematics* **114** (1990).
[31] N. Megiddo, "Towards a genuinely polynomial algorithm for linear programming," *SIAM J. Comput.* **12** (1983) 347–353.
[32] N. Megiddo, "Linear programming in linear time when the dimension is fixed," *J. ACM* **31** (1984) 114–127.
[33] N. Megiddo, Ed., "New Approaches to Linear Programming," *Algorithmica* **1** (1986) a sepcial issue.
[34] N. Megiddo, "On the complexity of linear programming," in: *Advances in economic theory Fifth world congress*, T. Bewley, ed., Cambridge University Press, Cambridge, 1987, pp. 225–268.
[35] N. Megiddo, "Linear programming (1986)," *Annual Review of Computer Science* **2** (1987), 119–145.
[36] N. Megiddo, Ed., *Progress in Mathematical Programming: Interior-Point and Related Methods*, Springer-Verlag, New York, 1988.
[37] N. Megiddo, "Pathways to the optimal set in linear programming," in [36], pp. 131–158.
[38] J. Renegar, "A polynomial-time algorithm based on Newton's method for linear

programming," *Mathematical Programming* **40** (1988) 59–94.

[39] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley-Interscience, New York, 1986.

[40] R. Seidel, " Linear programming and convex hulls made easy," in: Proceedings of the 6th Annual ACM Symposium on Computational Geometry, ACM Press, New York, 1990.

[41] M. Sharir and E. Welzl, "A combinatorial bound for linear programming and related problems," manuscript, 1991.

[42] N. Z. Shor, "Utilization of the operation of space dilation in the minimization of convex function," *Cybernetics* **6** (1970) 7–15.

[43] S. Smale, "On the average number of steps in the simplex method of linear programming," *Mathematical Programming* **27** (1983) 241–262.

[44] G. Sonnevend, "An 'analytic center' for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming," in: Lecture Notes in Control and Information Sciences 84," Springer, New York, 1985, pp. 866–876.

[45] É. Tardos, "A strongly polynomial algorithm to solve combinatorial linear programs," *Operations Research* **34** (1986) 250–256.

[46] M. J. Todd, "Polynomial expected behavior of a pivoting algorithm for linear complementarity and linear programming problems," *Mathematical Programming* **35** (1986) 173–192.

[47] P. Vaidya, "Speeding-up linear programming using fast matrix multiplication," in: Proceedings of the 30st IEEE Symposium on Foundations of Computer Science (1989), IEEE Computer Society Press, Los Angeles, 1990, pp. 332–337.

[48] R. J. Vanderbei, M. S. Meketon and B. A. Freedman, "A modification of Karmarkar's linear programming algorithm," *Algorithmica* **1** (1986) 395–407.

[49] Y. Ye, "An $O(n^3L)$ potential reduction algorithm for linear programming," *Math. Programming* **50** (1991) 239–258.