On the complexity of Linear Programming

Clovis C. Gonzaga
Dept. of Mathematics – Federal University of Santa Catarina
Florianópolis, SC, Brazil
clovis@mtm.ufsc.br

April, 1995

Abstract

In this paper we show a simple treatment of the complexity of Linear Programming. We describe the short step primal-dual path following algorithm and show that it solves the linear programming problem in $O(\sqrt{n}L)$ iterations, each one with a bound of $O(n^3)$ arithmetical computations.

1 Introduction

The first polynomial time algorithm for linear programming was published by Khachiyan [5] in 1978, and had a complexity bound of $O(n^4L)$ arithmetical computations, where n is the space dimension, and L is the bit length of the input data (to be discussed below). This bound was lowered by Karmarkar [4] in a landmark paper of 1984, which opened a new field of research now known as "interior point methods".

Karmarkar's algorithm solves a linear programming problem in $O(nL)$ iterations, each of which includes the solution of a linear system in $\mathbb{R}^n$. The complexity of each iteration is $O(n^3)$ operations, but this bound was lowered by Karmarkar to $O(n^{2.5})$ operations by using reinversion techniques, thus achieving an overall bound of $O(n^{3.5}L)$ operations. Not only this complexity bound is lower than Khachiyan's, but the resulting algorithm has an excellent performance in practical problems.

In 1986, Renegar [12] described the first algorithm with a complexity bound of $O(\sqrt{n}L)$ iterations, although still with the same overall complexity as Karmarkar's. His approach was based on methods of centers, and followed for the first time the central path (to be described below). In 1987, Gonzaga [1] and Vaidya [14] obtained simultaneously algorithms with the complexity of $O(n^3 L)$ operations, the former by means of the traditional logarithmic barrier function, and the latter by extending Renegar's results.

From then on, many algorithms with $O(\sqrt{n}L)$-iteration complexity were developed. Primal algorithms followed the three lines mentioned above: potential reduction methods using Karmarkar's potential function, methods of centers following Renegar's approach, and primal path following methods based on the barrier function.

Starting in 1987 with a paper by Kojima, Mizuno and Yoshise [7], the scene was gradually dominated by primal-dual algorithms. These methods are elegant, efficient, and are easily extended to a large class of linear complementarity problems and to quadratic programming. Primal-dual algorithms with a complexity of $O(\sqrt{n}L)$ iterations were first obtained by Kojima, Mizuno and Yoshise [8] and independently by Monteiro and Adler [11].

The evolution of the area is described in the surveys by Todd [13], by Gonzaga [2] and by den Hertog [3]. In this paper we present the basic short step primal-dual path following algorithm in a setting very similar to that of [8, 11], and show how the complexity bound is proved.

Notation. Given a vector x, d, ϕ, the corresponding upper case symbol denotes the diagonal matrix X, D, Φ defined by the vector. The symbol e will denote the vector of ones, with dimension given by the context.

We denote componentwise operations on vectors by the usual notations for real numbers. Thus, given two vectors u, v of the same dimension, uv, u/v, etc. denotes the vectors with components $u_i v_i$, u_i/v_i, etc.. This notation is consistent as long as componentwise operations are given precedence over matrix operations. Note that $uv \equiv Uv$ and if A is a matrix, then $Auv \equiv AUv$, but in general $Auv \neq (Au)v$.

2 Complexity issues

In this section we study a feasible linear programming problem with integer data. We will show that instead of computing an exact optimal solution of the problem, it is enough to find a feasible point with cost sufficiently near its optimal value, and then to perform a rounding procedure.

The primal linear programming problem is:

$$
(LP) \qquad \begin{array}{rrcl} \text{minimize} & c^T x & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0, \end{array}
$$

where $c \in I\!R^n$, $b \in I\!R^m$ and $A \in I\!R^{m \times n}$ is a full-rank matrix.

Assume that all data are integer (or equivalently rational).

Define the number l as the total length of the input in binary representation, that is, the summation of the number of bits used by all entries in A, b and c. The significance of this number is the following: no operation containing products and sums of data without repetitions can result in a number greater than 2^l. The determinant of a matrix is dominated by the product of the column norms, and hence $|\det M| \leq 2^l$ for any square submatrix M of $[\, A \mid b \,]$.

Define $L = 2l + n + 1$.

The next lemma shows that 2^{-L} is indeed a very small number.

Lemma 2.1 *Let v be the cost of an optimal solution of (LP). If x is a vertex of the feasible set, then:*
(i) For $i = 1, \ldots, n$, either $x_i = 0$ or $x_i > 2^{-L}$.
(ii) Either $c^T x = v$ or $c^T x > v + 2^{-L}$.
If the feasible set is bounded then for any feasible point x,
(iii) $c^T x \leq 2^L$.

Proof. Let x be a vertex of the feasible set. The system $Ax = b$ can be partitioned as

$$A_B x_B + A_N x_N = b,$$

where x_B is a vector made of the positive components of x and A_B is a matrix with the corresponding columns of A. Then $x_N = 0$ and the system can be written as $A_B x_B = b$.

If x is a vertex then the columns in A_B are linearly independent. Assume that A_B is square (or eliminate redundant equations to make it square). Then the solution of the system is given by Cramer's rule, for $i \in B$:

$$x_i = \frac{\Delta_i}{\Delta},$$

where $\Delta = \det A_B$ and Δ_i is the determinant obtained by substituting b for the column A_i.

Let l_1 and l_c be respectively the total number of bits used respectively by $\{A, b\}$ and by c.

(i) is true because Δ_i is integer and $|\Delta| \leq 2^{l_1} < 2^L$.

(ii) Let x^* be an optimal vertex. By the process above, we have either $x_i^* = 0$ or $x_i^* = \Delta_i^*/\Delta^*$, with Δ_i^* integer and $\Delta^* \leq 2^l$. It follows that $\Delta\Delta^* c^T(x - x^*)$ is integer. Hence, either $c^T(x - x^*) = 0$ or $c^T(x - x^*) \geq 1/\Delta\Delta^* \geq 2^{-2l} > 2^{-L}$.

To prove (iii), consider first the vertex x as above and $i \in B$. Since $|\Delta_i| \leq 2^{l_1}$ and $\Delta \neq 0$ is integer, $x_i \leq 2^{l_1}$. It follows that $e^T x \leq n2^{l_1} \leq 2^L$, and since $c_i x_i \leq 2^{l_1 + l_c} = 2^l$, $c^T x \leq n2^l \leq 2^L$. This establishes (iii) for all vertices of S. When S is bounded, any feasible point can be expressed as a convex combination $\sum_{j \in J} \lambda_j x^j$ of vertices, where $\lambda_j \geq 0$ for $j \in J$ and $\sum_{j \in J} \lambda_j = 1$. It follows that $c^T x = \sum_{j \in J} \lambda_j c^T x^j \leq 2^L \sum_{j \in J} \lambda_j = 2^L$, completing the proof. ∎

This lemma implies that if we find a vertex with cost smaller than 2^{-L} then it must be an optimal solution.

We now state without proof a known result (see [9]): given any feasible point x there exists a process called "purification of x" that finds with no more than $O(n^3)$ computations a vertex $\tilde{x}$ such that $c^T \tilde{x} \leq c^T x$.

The purification algorithm is similar to the simplex method, reducing one variable to zero in each iteration by a pivoting operation, so that no more than $n - 1$ pivots are needed to reach a vertex.

Finally, given x such that $c^T x \leq v + 2^{-L}$, where v is the cost of an optimal solution, the purification must result in an optimal vertex.

3 The primal-dual problem

We shall consider the primal and dual linear programming problems

$$(LP) \qquad \begin{array}{rrcl} \text{minimize} & c^T x & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0, \end{array}$$

and

$$(LD) \qquad \begin{array}{rrcl} \text{maximize} & b^T y & & \\ \text{subject to} & A^T y + s & = & c \\ & s & \geq & 0, \end{array}$$

where $c \in I\!R^n$, $b \in I\!R^m$ and $A \in I\!R^{m\times n}$ is a full-rank matrix.

The following fact will be frequently used: given a primal feasible solution x and a dual feasible solution (y, s), the duality gap between these solutions is given by

$$x^T s = c^T x - b^T y.$$

This is seen by direct substitution: $x^T s = x^T(c - A^T y) = c^T x - (Ax)^T y = c^T x - b^T y$.

Optimality conditions. A feasible primal-dual solution (x, y, s) is optimal if and only if the duality gap is zero. Note that since in this case $x, s \geq 0$, $x^T s = 0$ if and only if $xs = 0$ (using componentwise multiplication). This leads to the following optimality conditions (which coincide with the Karush-Kuhn-Tucker conditions) for the primal and dual problems:

$$\begin{array}{rcl} xs &=& 0 \\ Ax &=& b \\ A^T y + s &=& c \\ x, s &\geq& 0. \end{array} \tag{1}$$

Remark on notation. Since A has full rank by hypothesis, the vector y is univocally determined by x and s in the feasibility equations. Thus, we can refer to a primal-dual solution simply as (x, s), instead of (x, y, s).

We define the feasible set for (1) by

$$F = \{(x, s) \in I\!R^{2n} \mid Ax = b \ , \ A^T y + s = c \ , \ y \in I\!R^m, x, s \geq 0\},$$

and the set of interior points by

$$F^0 = \{(x, s) \in F \mid x, s > 0\}.$$

And now we present our main assumption for this paper. We shall postpone the discussion of this assumption to the end of this section, after presenting one more hypothesis.

Assumption 1. $F^0 \neq \emptyset$.

The central path. Primal-dual algorithms work directly with the optimality conditions. Solving (1) is difficult, and one source of difficulty is the combinatorial nature of the complementarity condition $xs = 0$: one must detect which of each pair of variables (x_i, s_i) will be null at an optimal solution.

The essential feature of interior point path following methods is to work with the following perturbed conditions, where $\mu \in I\!R$, $\mu > 0$ and as usual, $e = [\,1 \cdots 1\,]^T$:

$$\begin{aligned} xs &= \mu e \\ Ax &= b \\ A^T y + s &= c \\ x, s &\geq 0. \end{aligned} \tag{2}$$

This system has a unique solution $(x(\mu), s(\mu))$ for any $\mu > 0$, and defines a smooth curve $\mu > 0 \mapsto (x(\mu), s(\mu))$ called *central path.* The central path runs through the set of interior points, keeping a comfortable distance from the non-optimal faces of F, and ends at an optimal solution (x^*, s^*) known as the central optimum. For proofs of these facts, see for instance [6].

Adding the equations $x_i(\mu)s_i(\mu) = \mu$, we see that $x(\mu)^T s(\mu) = n\mu$. Taking $x^T s$ as an objective to minimize, it is enough to find (approximately) points on the central path with $\mu \to 0$. This is what path following algorithms do, by following a homotopy approach as in the model below:

Algorithm 3.1 *Data:* $\epsilon > 0$, $\mu^0 > 0$.

$k := 0$.

REPEAT

Find *approximately* $(x(\mu^k), s(\mu^k))$.
Choose $\mu^{k+1} < \mu^k$.
$k := k + 1$.

UNTIL $\mu^k < \epsilon$.

The proximity measure.

The word 'approximately' in the algorithm model above will be our concern now. Given $\mu > 0$, we want to find (x, s) such that $xs/\mu = e$. The deviation of this condition for a given feasible pair (x, s) is given by the proximity measure

$$\delta(x, s, \mu) = \|\frac{xs}{\mu} - e\|.$$

Now we show that if $\delta(x,s,\mu)$ is small, then the duality gap $x^T s$ is again well related to μ for such *nearly central points.*

Assume that $\delta(x,s,\mu) \in (0,1)$. Then $xs/\mu = e + \rho$, where $\|\rho\| = \delta(x,s,\mu)$. Pre-multiplying this expression by e^T, we obtain

$$x^T s \leq (n + \delta(x,s,\mu)\sqrt{n})\mu. \tag{3}$$

Path following methods work with nearly central points. This becomes clear if we define the following neighborhood of the central path:
Let $\alpha \in (0,1)$ be given (usually taken as $\alpha = 0.5$). Our neighborhood of the central path will be defined as

$$\mathcal{T}_\alpha = \bigcup_{\mu \in (0,\infty)} \{(x,s) \mid \delta(x,s,\mu) \leq \alpha\}.$$

The algorithms will generate sequences in $\mathcal{T}_\alpha$.

Assumption 2. An initial point (x^0, s^0) and a value $\mu^0 < 2^L$ are available so that $\delta(x^0, s^0, \mu^0) \leq 0.5$.

Remarks on the assumptions. The assumptions 1. and 2. simplify the theory very much, but they are not necessary. The most general setting of the linear programming problem is the following: given the problem (1), either find an optimal solution or certify that no one exists. Note that the problem can be infeasible.

A paper by Ye, Todd and Mizuno [15] shows how to modify the statement of a problem with the introduction of some extra variables, so that the resulting equivalent problem satisfies both our assumptions, and has the following property: an optimal solution always exists (as a consequence of the assumptions), and characteristics of the optimal solutions allow us either to retrieve an optimal solution for the original problem, or to certify that no optimal solution exists.

4 The Newton step

Each iteration of the algorithm above starts with an interior point (x,s) and a value of μ, and tries to approach $(x(\mu), s(\mu))$. Ideally, we want to find

$$x^+ = x + u \quad , \quad s^+ = s + v$$

such that $(x^+, s^+) \in F$ and $x^+ s^+ = \mu e$.

The direction (u, v) must be feasible, i.e, $A(x+u) = Ax$ and for some y^+, $A^T y^+ + s + v = A^T y + s$. This is equivalent to saying that $u \in \mathcal{N}(A)$ and $v \in \mathcal{R}(A^T)$, where $\mathcal{N}, \mathcal{R}$ denote respectively the null space and the range space of a matrix.

The Newton step solves this approximately by linearizing (2). The linearization is straightforward, and leads to the following system, which has a unique solution:

$$\begin{array}{rcl} xv + su & = & -xs + \mu e \\ u & \in & \mathcal{N}(A) \\ v & \in & \mathcal{R}(A^T) \end{array} \tag{4}$$

In what follows we shall prove what seems to be the most important result in interior point methods: we prove that in the neighborhood $\mathcal{T}_\alpha$, Newton's method is extremely efficient. The theorem will be stated with a little more generality than needed in this paper.

This proof will use the following result on vector norms, due to Mizuno [10].

Lemma 4.1 *If $u, v \in I\!\!R^n$ are such that $u^T v \geq 0$, then*

$$\|uv\| \leq \frac{1}{\sqrt{8}} \|u+v\|^2.$$

Theorem 4.2 *Given $(x, s) \in F$ and $\mu > 0$ such that $\delta(x, s, \mu) = \delta < 1$, let (x^+, s^+) be the result of a Newton step (4) from (x, s). Then*

$$\delta(x^+, s^+, \mu) \leq \frac{\delta^2}{\sqrt{8}(1-\delta)} \tag{5}$$

and for any $\lambda \in [0, 1]$,

$$\delta(x + \lambda u, s + \lambda v, \mu) \leq (1-\lambda)\delta(x, s, \mu) + \lambda^2 \delta(x^+, s^+, \mu). \tag{6}$$

Besides this, if $\delta \leq 0.7$, then for any $\lambda \in [0, 1]$ the point $((x + \lambda u), (s + \lambda v))$ is an interior solution.

Proof. We start by scaling the Newton equations to obtain a simpler form. Let $d \in I\!\!R^n$ be a positive vector, and let us do a change of variables (a scaling) in the Newton equations by setting

$$x = d\bar{x}\ ,\ u = d\bar{u}\ ,\ s = d^{-1}\bar{s}\ ,\ v = d^{-1}\bar{v}.$$

Note that $\bar{x}\bar{s} = xs$ and $\bar{u}\bar{v} = uv$. The system (4) becomes

$$\begin{aligned} \bar{x}\bar{v} + \bar{s}\bar{u} &= -\bar{x}\bar{s} + \mu e \\ \bar{u} &\in \mathcal{N}(AD) \\ \bar{v} &\in \mathcal{R}(DA^T). \end{aligned}$$

Now we choose $d = \sqrt{x/s}$. It follows that $\bar{x} = \bar{s} = \sqrt{xs}$, and the first equation in the system (4) can be written as

$$\bar{u} + \bar{v} = -\frac{\mu}{\bar{x}}\left(\frac{\bar{x}\bar{s}}{\mu} - e\right). \tag{7}$$

We know that $\|\bar{x}^2/\mu - e\| = \|xs/\mu - e\| = \delta$ and hence for each component $i = 1, \dots, n$, $|\bar{x}_i^2/\mu - 1| \le \delta$. It follows that $\bar{x}_i^2/\mu \ge 1 - \delta$, and consequently for $i = 1, \dots, n$,

$$\frac{\mu}{\bar{x}_i^2} \le \frac{1}{1-\delta}.$$

Using this result and (7), we conclude that

$$\|\bar{u} + \bar{v}\|^2 \le \frac{\mu}{1-\delta}\|\frac{\bar{x}\bar{s}}{\mu} - e\|^2 = \mu\frac{\delta^2}{1-\delta}. \tag{8}$$

Now we establish a bound on the proximity of points along the direction $(\bar{u}, \bar{v})$.
We have:

$$\begin{aligned} (x + \lambda u)(s + \lambda v) &= xs + \lambda(xv + su) + \lambda^2 uv \\ &= (1-\lambda)xs + \lambda\mu e + \lambda^2 uv, \end{aligned}$$

using (4). It follows that

$$\frac{(x + \lambda u)(s + \lambda v)}{\mu} - e = (1-\lambda)\left(\frac{xs}{\mu} - e\right) + \lambda^2\frac{uv}{\mu}.$$

In particular for $\lambda = 1$,

$$\delta(x^+, s^+, \mu) = \|\frac{x^+s^+}{\mu} - e\| = \|\frac{uv}{\mu}\|,$$

and we get for $\lambda \in [0, 1]$,

$$\delta(x + \lambda u, s + \lambda v, \mu) \le (1-\lambda)\delta(x, s, \mu) + \lambda^2\delta(x^+, s^+, \mu),$$

establishing (6).

Now we use Lemma 4.1. Since $u \in \mathcal{N}(A)$ and $v \in \mathcal{R}(A^T)$, $\bar{u}^T\bar{v} = u^Tv = 0$ and hence

$$\|uv\| = \|\bar{u}\bar{v}\| \leq \frac{\|\bar{u}+\bar{v}\|^2}{\sqrt{8}}.$$

Merging this with (8), we conclude that

$$\delta(x^+, s^+, \mu) = \|\frac{uv}{\mu}\| \leq \frac{\delta^2}{\sqrt{8}(1-\delta)},$$

proving (5).

Now we establish the assertion on feasibility. Since (u, v) is a feasible direction by construction, all that remains to be proved is that the segment $\{(x+\lambda u, s+\lambda v) \mid \lambda \in [0,1]\}$ does not touch the boundary of $I\!R^n_+$. Calculating (5) for $\delta \leq 0.7$, we obtain $\delta(x^+, s^+, \mu) < 1$, and it follows from (6) that for $\lambda \in [0,1]$, $\delta(x + \lambda u, s + \lambda v, \mu) < 1$.

Assuming by contradiction that for some $\bar{\lambda} \in [0,1]$ and some $i = 1, \ldots, n$, either $x_i + \bar{\lambda} u_i = 0$ or $s_i + \bar{\lambda} v_i = 0$, we get

$$\delta(x + \bar{\lambda}u, s + \bar{\lambda}v, \mu) \geq \left| \frac{(x_i + \bar{\lambda}u_i)(s_i + \bar{\lambda}v_i)}{\mu} - 1 \right| = 1.$$

This contradicts the last inequality above, completing the proof. ∎

This means that Newton's method is very efficient for solving (2) in the sense that it reduces the proximity measure quadratically in a large region. We showed a little more: not only the complete Newton step is efficient, but the proximity falls continuously along the Newton direction.

5 The short step algorithm

Now we describe the short step path following algorithm and prove that it solves problem (1) in $O(\sqrt{n}L)$ steps, each one consisting of a Newton step. We specify the procedures in the homotopy model above.

Algorithm 5.1 *Data:* $\epsilon > 0$, $(x^0, s^0) \in \mathcal{T}_\alpha$, $\mu^0 \in (0, 2^L)$ *with* $\delta(x^0, s^0, \mu^0) \leq 0.5$.

$k := 0$.

REPEAT

Set $x := x^k$, $s := s^k$, $\mu := \mu^k$.

Compute the Newton step (u, v) by (4), and set

$$(x^{k+1}, s^{k+1}) = (x^+, s^+) := (x, s) + (u, v).$$

Set $\mu^{k+1} := (1 - \frac{0.2}{\sqrt{n}})\mu$.

$k := k + 1$.

UNTIL $\mu^k < \epsilon$.

Lemma 5.2 *The algorithm 5.1 generates a sequence of feasible iterates, and* $\delta(x^k, s^k, \mu^k) \leq 0.5$ *for* $k = 0, 1, 2, \ldots$.

Proof. The result is true for $k = 0$. Assume that for some $k = 0, 1, \ldots,$ (x^k, s^k) is feasible and $\delta(x^k, s^k, \mu^k) \leq 0.5$. We must prove that in this iteration (x^+, s^+) is feasible and $\delta(x^+, s^+, \mu^{k+1}) \leq 0.5$.
From Theorem 4.2, after the Newton step with $\mu = \mu^k$, (x^+, s^+) is feasible and

$$\delta(x^+, s^+, \mu) \leq \frac{0.25}{0.5\sqrt{8}} < 0.2.$$

We must show that

$$\left\|\frac{x^+ s^+}{(1 - \frac{0.2}{\sqrt{n}})\mu} - e\right\| \leq 0.5.$$

We have

$$\begin{aligned}
\left\|\frac{x^+ s^+}{\mu} - (1 - \frac{0.2}{\sqrt{n}})e\right\| &\leq \left\|\frac{x^+ s^+}{\mu} - e\right\| + \left\|\frac{0.2}{\sqrt{n}}e\right\| \\
&= \delta(x^+, s^+, \mu) + 0.2 \\
&\leq 0.4 \\
&\leq 0.5(1 - \frac{0.2}{\sqrt{n}}),
\end{aligned}$$

because $1 - 0.2/\sqrt{n} \geq 0.8$, completing the proof. ∎

The short step algorithm is obviously not practical, but it gives an immediate proof of polynomiality, as follows.

Theorem 5.3 *The algorithm 5.1 with* $\epsilon = 2^{-2L}$ *stops in* $O(\sqrt{n}L)$ *iterations, with a primal-dual pair* $(\hat{x}, \hat{s})$. *A purification from* $\hat{x}$ *results in an optimal primal vertex.*

Proof. By construction, the algorithm gives

$$\mu^k \leq (1 - \frac{0.2}{\sqrt{n}})^k \mu^0,$$

and hence

$$\log(\frac{\mu^k}{\mu^0}) \leq k \log(1 - \frac{0.2}{\sqrt{n}}) \leq -0.2\frac{k}{\sqrt{n}},$$

since $\log(1-\lambda) \leq -\lambda$ for $\lambda \in (0,1)$. Since $\mu^0 \leq 2^L$ and at all iterations $\mu^k \geq \epsilon = 2^{-2L}$, it follows that

$$k \leq -5\sqrt{n} \log \frac{2^{-2L}}{2^L} \leq 15\,\sqrt{n}L \log 2.$$

This proves that $k = O(\sqrt{n}L)$. Using (3) and the stopping rule of the algorithm, the resulting point $(\hat{x}, \hat{s})$ satisfies

$$\hat{x}^T \hat{s} \leq (n + 0.5\sqrt{n})\mu^k < 2n2^{-2L} < 2^{-L}.$$

The analysis in section 2 ensures that the purification from $\hat{x}$ results in an optimal vertex because $c^T\hat{x} - v \leq \hat{x}^T\hat{s}$, completing the proof. ∎

Acknowledgement. I thank my student Marco A. Menezes for suggestions and corrections that greatly improved this paper.

References

[1] C. C. Gonzaga. An algorithm for solving linear programming problems in $O(n^3L)$ operations. In N. Megiddo, editor, *Progress in Mathematical Programming : Interior Point and Related Methods*, pages 1–28. Springer Verlag, New York, 1989.

[2] C. C. Gonzaga. Path following methods for linear programming. *SIAM Review*, 34(2):167–227, 1992.

[3] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming, Algorithms and Complexity.* PhD thesis, Faculty of Mathematics and Informatics, TU Delft, NL–2628 BL Delft, The Netherlands, September 1992.

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

[5] L. G. Khachiyan. A polynomial algorithm for linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.

[6] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. *A unified approach to interior point algorithms for linear complementarity problems*, volume 538 of *Lecture Notes in Computer Science.* Springer Verlag, Berlin, Germany, 1991.

[7] M. Kojima, S. Mizuno, and A. Yoshise. A primal–dual interior point algorithm for linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming : Interior Point and Related Methods*, pages 29–47. Springer Verlag, New York, 1989.

[8] 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.

[9] K. O. Kortanek and J. Zhu. New purification algorithms for linear programming. *Naval Research Logistics Quarterly*, 35:571–583, 1988.

[10] S. Mizuno. A new polynomial time method for a linear complementarity problem. *Mathematical Programming*, 56:31–43, 1992.

[11] R. D. C. Monteiro and I. Adler. Interior path following primal–dual algorithms : Part I : Linear programming. *Mathematical Programming*, 44:27–41, 1989.

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

[13] M. J. Todd. Potential-reduction methods in mathematical programming. Technical report, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1995.

[14] P. M. Vaidya. An algorithm for linear programming which requires $O((m+n)n^2 + (m+n)^{1.5}nL)$ arithmetic operations. *Mathematical Programming*, 47:175–201, 1990. Condensed version in : *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 29–38, 1987.

[15] Y.Ye, M. J. Todd, and S. Mizuno. An $O(\sqrt{n}l)$-iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research*, 19:53–67, 1994.