

**A Cubically Convergent Method
for Locating a Nearby Vertex
in Linear Programming**

*R. A. Tapia
Yin Zhang*

**CRPC-TR89018
December 1989**

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

A Cubically Convergent Method for Locating a Nearby Vertex in Linear Programming¹

R. A. Tapia² and Yin Zhang³

December, 1989

Abstract

Given a point sufficiently close to a nondegenerate basic feasible solution x^* of a linear program, we show how to generate a sequence $\{p^k\}$ that converges to the 0–1 vector $sign(x^*)$ at a Q -cubic rate. This extremely fast convergence enables us to determine, with a high degree of certainty, which variables will be zero and which will be nonzero at optimality and then construct x^* from this information.

Keywords: Linear programming, vertex, Q -cubic convergence.

¹This research was supported in part by NSF Coop. Agr. No. CCR-8809615, AFOSR 89-0363 and DOE DEFG05-86ER 25017. We would like to thank Bob Bixby for helpful discussions.

²Professor, Department of Mathematical Sciences, Rice University, Houston, Texas

³Lecturer, Department of Mathematical Sciences, Rice University, Houston, Texas

1 Introduction

Interior point algorithms for the linear program

$$\begin{aligned} & \text{minimize} && c^T x, \\ & \text{subject to} && Ax = b, \\ & && x \geq 0, \end{aligned} \tag{1}$$

where $c, x \in \mathbf{R}^n$, $b \in \mathbf{R}^m$ and $A \in \mathbf{R}^{m \times n}$ for $m < n$, do not search through the set of vertices of the feasible set to obtain an optimal vertex as does the simplex method. Instead, a primal interior point algorithm generates a sequence of interior (or strictly feasible) points $\{x^k\}$ that converges to an optimal solution. For such an interior point method, the construction of appropriate stopping criteria is often a delicate issue. Additional concerns arise when a highly accurate solution is required or when an optimal vertex is required as is the case in some applications, e.g., integer programming. It is therefore of value to develop an iterative procedure with a high convergence rate that can locate an optimal vertex from a given reasonably good approximate solution.

In this paper, we present and analyze a new method for locating an optimal vertex x^* of a linear program from a given approximate solution. Working with an auxiliary problem, this method generates from the given approximate solution a sequence that converges to the 0–1 vector $\text{sign}(x^*)$. Our usage of the *sign* function takes zero as the sign of zero; otherwise it is the standard usage. Using the information provided by this 0–1 vector, x^* can be recovered from the linear constraints $Ax = b$ by setting to zero those variables that have been identified as zero and then solving for the nonzero variables. The significance of this approach is that the convergence rate of this new method is Q -cubic if the optimal vertex is nondegenerate.

In anticipation of future use, we state the dual linear program of (1)

$$\begin{aligned}
& \text{maximize} && b^T y \\
& \text{subject to} && A^T y + z = c, \\
& && z \geq 0.
\end{aligned} \tag{2}$$

In (1.2) $z \in \mathbf{R}^n$ is the vector of dual slack variables. We will conveniently assume throughout this paper that the matrix A has full rank m .

This paper is organized as follows. In Section 2, we present our method for locating the nondegenerate optimal vertex from an approximate primal solution. In Section 3 we show that this method is Q -cubically convergent and also briefly discuss its implementations as well as its dual variant. In Section 4 we give several numerical examples which demonstrate the convergence behavior of our method. Concluding remarks are given in the final section.

2 The Algorithm

Recently, Tapia and Zhang [1] proposed a technique for identify optimal bases for interior point methods. The technique was motivated by the observation that essentially every interior point method motivated and influenced by the Karmarkar algorithm [2] involves a matrix of the form $DA^T(AD^2A^T)^{-1}AD$, where D is a diagonal matrix that changes at each iteration. In primal algorithms (affine variants of the Karmarkar algorithm, for example, see [3], [4] and [5]), we see $D = \text{diag}(x)$. In dual algorithms (see Adler *et al.* [6] and Monma and Morton [7], for example), we see $D = [\text{diag}(z)]^{-1}$, where z is the vector of dual slack variables. And in primal-dual algorithms (see Kojima *et al.* [8], for example), we see $D^2 = \text{diag}(x)[\text{diag}(z)]^{-1}$. In [1], Tapia and Zhang showed that under suitable assumptions, optimal basis information can be obtained from the diagonal of this matrix $DA^T(AD^2A^T)^{-1}AD$ which they called the indicator vector or simply the indicator. Ye and Todd (1987) [9] perhaps were the first to point out that the diagonal elements of such a matrix contain valuable information.

For a fixed matrix $A \in \mathbf{R}^{m \times n}$ ($m < n$), consider the matrix-valued function $H : \mathbf{R}^n \rightarrow \mathbf{R}^{n \times n}$ defined by

$$H(d) = DA^T(AD^2A^T)^+AD, \quad (3)$$

where $d \in \mathbf{R}^n$, $D = \text{diag}(d)$ and the superscript “+” denotes the generalized inverse. The Tapia-Zhang indicator is defined as the function $q : \mathbf{R}^n \rightarrow \mathbf{R}^n$ obtained as the diagonal of $H(d)$, i.e.,

$$q(d) = \text{diag}(H(d)) \text{ or } q_i(d) = H_{ii}(d), \quad i = 1, 2, 3, \dots, n. \quad (4)$$

This function $q(d)$ is defined for all $d \in \mathbf{R}^n$; however it will not be continuous at points d where the matrix AD^2A^T changes rank in every neighborhood of d . At points d where AD^2A^T has constant rank in some neighborhood of d , $q(d)$ will be infinitely smooth. It is not hard to see that

$$0 \leq q(d) \leq 1$$

for all $d \in \mathbf{R}^n$ because both $H(d)$ and $I - H(d)$ are orthogonal projections (satisfying $P^T = P$ and $P^2 = P$) and, therefore, are positive semi-definite with non-negative diagonals.

We now define a new function $u : \mathbf{R}^n \rightarrow \mathbf{R}^n$ as

$$u(d) = q(Dd). \quad (5)$$

where $D = \text{diag}(d)$. It is evident from the definition of the function q that

$$u(d) = \text{diag}(D^2A^T(AD^4A^T)^+AD^2).$$

Our method for locating an optimal vertex x^* of a linear program from an approximate primal solution x can be described as follows.

Algorithm 1 *Given an approximate primal solution x of the linear program (1) and a small number $\varepsilon \geq 0$, set $p^0 = x$. For $k = 1, 2, \dots$, until the criterion (6) (defined below) is satisfied, compute*

$$p^k = u(p^{k-1}).$$

The stopping criterion is satisfied when

$$\{i : p_i^k \geq 1 - \varepsilon, 1 \leq i \leq n\} \cup \{i : p_i^k \leq \varepsilon, 1 \leq i \leq n\} = \{1, 2, \dots, n\}. \quad (6)$$

In the algorithm, the tolerance ε is allowed to be zero for theoretical purposes. In practice, ε will be chosen as a small positive number.

The procedure of recovering the optimal vertex is as follows. First solve the square linear system $A_1 w = b$ for w , where A_1 is the submatrix of A consisting of the columns of A which correspond to $p_i^k \geq 1 - \varepsilon$ arranged in ascending order of their column indices. Then construct the optimal vertex x^* by extending w to an n -vector by inserting zeros in the positions corresponding to $p_i^k \leq \varepsilon$.

3 The Cubic Convergence

We first state without proof two lemmas concerning the properties of the function $q(d)$ that we will use in this section. These results were proved in [1] and interested readers are referred to that paper.

Lemma 3.1 *For $d \in \mathbf{R}^n$ let $q(d)$ be given by (4) for some A of full row rank. Consider the n -dimensional vector d^* where some components of d^* may be infinite. Assume that the components of d^* can be divided into two sets: S_α and S_β , such that*

1. S_α contains m and S_β contains $n - m$ components of d^* ;
2. all elements in S_α are non-zero and

$$\frac{\max\{|d_i^*| : d_i^* \in S_\beta\}}{\min\{|d_i^*| : d_i^* \in S_\alpha\}} = 0; \quad (7)$$

3. the m by m submatrix of A consisting of columns corresponding to the components in S_α is nonsingular.

Then, as d converges to d^*

$$\lim_{d \rightarrow d^*} q_i(d) = q_i(d^*) = \begin{cases} 1, & \text{if } d_i^* \in S_\alpha, \\ 0, & \text{if } d_i^* \in S_\beta. \end{cases} \quad (8)$$

Since d^* may have infinite components, we define the derivatives of q at d^* by continuity, for example, for $d \in \mathbf{R}^n$,

$$\nabla q_i(d^*) = \lim_{d \rightarrow d^*} \nabla q_i(d), \quad i = 1, 2, \dots, n.$$

Lemma 3.2 *Let $q(d)$ be given as in (4) and d^* satisfy the conditions of Lemma 3.1. Then $q(d)$ is at least twice continuously differentiable at d^* . Moreover, the Jacobian matrix of $q(d)$ vanishes at d^* ; i.e.,*

$$\nabla q_i(d^*) = 0, \quad i = 1, 2, \dots, n. \quad (9)$$

It should not be a surprise that the Jacobian matrix of $q(d)$ is zero at d^* . According to Lemma 3.1 either the maximum ($q_i(d^*) = 1$) or the minimum ($q_i(d^*) = 0$) is reached at d^* for every component of $q(d)$.

An immediate consequence of Lemma 3.2 is that whenever $\nabla^2 q_i(d)$ exists and is continuous at d^* , then

$$\|q(d) - q^*\| \leq O(\|d - d^*\|^2). \quad (10)$$

We now need the following two technical results in order to prove our main convergence result.

Lemma 3.3 *Let $d^* \in \mathbf{R}^n$ be finite and satisfy the three conditions in Lemma 3.1. Then there exist positive constants δ_0 and C_0 such that for all $d \in \mathbf{R}^n$ and $\|d - d^*\| \leq \delta_0$*

$$\|u(d) - p^*\| \leq C_0 \|d - d^*\|^3, \quad (11)$$

where p^* is a zero-one vector of m ones and $n - m$ zeros and $p_i^* = \text{sign}(|d_i^*|)$.

Proof: Since d^* is finite and satisfies the three conditions in Lemma 3.1 with all nonzero elements in S_α and zeros in S_β , so is D^*d^* where $D^* = \text{diag}(d^*)$. Therefore, by Lemma 3.1 $u(d) = q(Dd)$ converges to p^* as d converges to d^* . Also $u(d)$ is infinitely differentiable at D^*d^* .

A direct calculation shows (see [1] for details) that

$$\frac{\partial^2 q_\ell(d^*)}{\partial d_i \partial d_j} = \begin{cases} 2P_{\ell\ell}(d^*), & \text{if } i = j = \ell \text{ and } d_\ell^* = 0, \\ -2(d_\ell^* P_{i\ell}(d^*))^2, & \text{if } i = j \neq \ell, d_\ell^* \neq 0 \text{ and } d_i^* = 0, \\ 0 & \text{otherwise,} \end{cases}$$

where $P_{ij}(d)$ is the (i, j) -th element of the matrix $A^T(AD^2A^T)^{-1}A$. That is, if $d_\ell^* = 0$, the Hessian of $q_\ell(d)$ at d^* has only a single non-zero element in the (ℓ, ℓ) position; and if $d_\ell^* \neq 0$, then the Hessian of $q_\ell(d)$ at d^* has only non-zeros on the diagonal in the positions corresponding to zero d_i^* 's. Therefore, if we expand $q(Dd)$ at D^*d^* using a Taylor expansion and note that its first derivative is zero at D^*d^* (as proved in Lemma 3.2), we have

$$\|u(d) - p^*\| = \|q(Dd) - p^*\| \leq O(\|D_z^k d_z^k\|^2) + O(\|D^k d^k - D^* d^*\|^3),$$

where $d_z^k \in \mathbf{R}^{n-m}$ consists of the components of d^k that converge to zero and $D_z^k = \text{diag}(d_i^k)$. From the fact that $(D^k d^k)_i = (d_i^k)^2$, one can easily verify that

$$\|D_z^k d_z^k\|^2 \leq \|d^k - d^*\|^4 \quad \text{and} \quad \|D^k d^k - D^* d^*\|^3 = O(\|d^k - d^*\|^3).$$

The existence of constants δ_0 and C_0 such that (11) holds follows in a standard fashion from the continuity of the third derivative of $q(Dd)$ at D^*d^* . This completes the proof. \square

Corollary 3.1 *Let $p^* \in \mathbf{R}^n$ be a zero-one vector with m ones and $n - m$ zeros. If the m columns of the matrix A corresponding to the ones in p^* are linearly independent, then there exist positive constants δ and C such that for all $p \in \mathbf{R}^n$ and $\|p - p^*\| \leq \delta$*

$$\|u(p) - p^*\| \leq C\|p - p^*\|^3. \quad (12)$$

Proof: The proof follows directly from Lemma 3.3 because p^* is finite and satisfies the three conditions in Lemma 3.1. \square

Now we are ready to state and prove the Q -cubic convergence result for Algorithm 1.

Theorem 3.1 *Let $\{p^k\}$ be generated by Algorithm 1 with $\varepsilon = 0$. If x is sufficiently close to a nondegenerate optimal vertex x^* of the linear program (1), then $\{p^k\} \rightarrow p^*$ Q -cubically, where $p^* = \text{sign}(x^*)$ is a zero-one vector of m ones and $n - m$ zeros.*

Proof: Choose x such that $\|x - x^*\| \leq \min(\delta_0, \delta)$ and $\rho = \max(C_0, C)\|x - x^*\|^2 < 1$, where δ_0, C_0 and δ, C are positive constants satisfying (11) (with d replaced by x) and (12), respectively. The proof of convergence now follows by induction. Let $e_0 = \|x - x^*\|$ and $e_k = \|p^k - p^*\|$ for $k > 0$. By Lemma 3.3 (with d replaced by x) and our assumptions,

$$e_1 = \|u(p^0) - p^*\| \leq C_0\|x - x^*\|^3 \leq \rho e_0 < e_0.$$

So $e_1 \leq \delta$. By Corollary 3.1 and our assumptions,

$$e_2 = \|u(p^1) - p^*\| \leq C\|p^1 - p^*\|^3 \leq \rho e_1 < e_1.$$

The induction step can now proceed in the same manner. This establishes the Q -linear convergence. The Q -cubic convergence follows directly from (12). \square

Numerical implementation of Algorithm 1 can be carried out basically in two ways. First, if an orthonormal basis $Q(d)$ of D^2A^T is computed by a QR method or an SVD method, then obviously

$$u_i(d) = Q_i(d)^T Q_i(d), \quad i = 1, 2, \dots, n, \quad (13)$$

where $Q_i(d)$ is the i -th row of $Q(d)$. On the other hand, if a Cholesky factor $L(d)$ of AD^4A^T is computed, the triangular system $L(d)Q(d) = AD^2$ needs to be solved to obtain $Q(d)$. Then $u(d)$ can be computed from (13).

In the case that the dual linear program (2) is being solved, a dual version of Algorithm 1 can be devised by letting $p_i^0 = 1/z_i$, $i = 1, 2, \dots, n$, where z is a given approximation to an optimal dual slack vector z^* . It can be easily shown that this dual version has at least Q -quadratic convergence rate for nondegenerate problems because of the estimate

$$\|q(d(z)) - q^*\| \leq O(\|z - z^*\|^2)$$

proved in [1], where $d_i(z) = 1/z_i$, $i = 1, 2, \dots, n$. Also such generated sequence $\{p^k\}$ has the same limit $p^* = \text{sign}(x^*)$.

4 Numerical Examples

We have tested our method using randomly generated problems. All our random numbers are uniformly distributed on the entire real line. For each problem, we generate an m by n matrix A and a vertex x^* . For simplicity, we make the first m elements of x^* nonzero and the rest zero. Correspondingly, we make sure that the first m columns of A are linearly independent. Also we generate a perturbation vector h and let the initial vector x for Algorithm 1 be $x = x^* + \tau h / \|h\|_2$. Notice that the parameter τ is equal to the quantity $\|x - x^*\|_2$. We tested Algorithm 1 on a Sun 3/50 Workstation with a machine epsilon of about 2.22×10^{-16} . Numerical results for four such randomly generated problems with various dimensions are tabulated in Table 1. In this table, the errors $\|p^k - p^*\|_2$, $k = 1, 2, 3, 4$, are given for each problem, where k is the iteration count.

Table 1: Numerical results for 4 random problems

Errors	$n = 10, m = 5$	$n = 20, m = 10$	$n = 40, m = 20$	$n = 80, m = 40$
$\ x - x^*\ $	6.00e-02	7.00e-02	6.00e-02	8.00e-02
$\ p^1 - p^*\ $	3.56e-01	5.48e-02	1.30e-02	2.63e-02
$\ p^2 - p^*\ $	7.91e-03	7.20e-03	1.48e-04	1.45e-04
$\ p^3 - p^*\ $	2.89e-09	6.34e-06	7.78e-12	1.80e-12
$\ p^4 - p^*\ $	9.16e-16	9.93e-16	1.49e-15	3.08e-15

It can be seen from the table that the convergence rate of Algorithm 1 was indeed Q -cubic (up to the point where machine epsilon was reached). We also observed (which is no surprise) cases where x was not close to x^* , and $\{p^k\}$ converged to a

zero-one vector corresponding to a nearby vertex.

5 Concluding Remarks

Working with an auxiliary problem, we have shown that the information needed to construct the nondegenerate optimal vertex of a linear program can be obtained iteratively from a given good approximate solution at a Q -cubic convergence rate. While this result certainly has theoretical value, considerably more research is needed in order to determine its practical value.

References

- [1] R. A. Tapia and Yin Zhang. A fast optimal basis identification technique for interior point linear programming methods. Technical Report TR89-1, Dept. Mathematical Sciences, Rice University, 1989.
- [2] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.
- [3] E. R. Barnes. A variation of Karmarkar’s algorithm for solving linear programming problems. *Math. Prog.*, 36:174–182, 1986.
- [4] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Soviet Math. Dokl.*, 8:674–675, 1967.
- [5] R. J. Vanderbei, M. S. Meketon, and B. A. Freeman. On a modification of Karmarkar’s linear programming algorithm. *Algorithmica*, 1:395–407, 1986.
- [6] I. Adler, N. Karmarkar, M.G.C. Resende, and G. Veiga. An implementation of Karmarkar’s algorithm for linear programming. Technical Report 86-8, Operations Research Center, UC Berkeley, 1986.

- [7] C.L. Monma and A.J. Morton. Computational experience with a dual affine variant of Karmarkar's method for linear programming. *OR Letters*, 6:261–267, 1987.
- [8] M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior point method for linear programming. Technical Report B-188, Department of Information Sciences, Tokyo Institute of Technology, Japan, 1987.
- [9] Y. Ye, and M. J. Todd. Containing and shrinking ellipsoids in the path-following algorithm. Manuscript, 1987, to appear in *Math. Programming*.

