A Study of Indicators for Identifying Zero Variables in Interior-Point Methods

A. El-Bakry
R.A. Tapia
Y. Zhang

CRPC-TR91168
June, 1991

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

A Study of Indicators for Identifying Zero Variables in Interior-Point Methods[1]

A. S. El-Bakry[2], R. A. Tapia[3] and Y. Zhang[4]

Abstract

The ability to identify zero variables early on in an iterative method is of considerable value and can be used to computational advantage. In this work we first give a formal presentation of the notion of indicators for identifying zero variables, and then study various indicators proposed in the literature for use with interior-point methods for linear programming. We present both theory and experimentation that speaks strongly against the use of the variables as indicators; perhaps the most frequently used indicator in the literature. Our study implies that an indicator proposed by Tapia in 1980 is particularly effective in the context of primal-dual interior-point methods.

---

[1] An earlier version of this paper was presented at ICIAM 91, Washington, D.C., July 1991.

[2] Department of Mathematical Sciences and the Center for Research on Parallel Computations, Rice University, Houston, Texas, 77521-1892. This author was supported in part by NSF Coop. Agr. No. CCR-8809615 and AFOSR 89-0363

[3] Department of Mathematical Sciences and the Center for Research on Parallel Computations, Rice University, Houston, Texas, 77521-1892. This author was supported in part by NSF Coop. Agr. No. CCR-8809615, AFOSR 89-0363, DOE DEFG05-86ER25017 and ARO 9DAAL03-90-G-0093

[4] Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD 21228, and visiting member of the Center for Research on Parallel Computations, Rice University, Houston, Texas, 77521-1892.This author was supported in part by NSF Coop. Agr. No. CCR-8809615, NSF DMS-9102761 and DOE DE-FG05-91ER25100

1 Introduction

This paper describes a study of various indicators proposed in the literature for the identification of zero variables in linear programming problems. Our particular focus will be on indicators that can be used in conjunction with primal-dual interior-point methods. We consider the linear programming problem in the standard form

$$\begin{array}{ll} \textit{minimize} & c^T x \\ \textit{subject to} & Ax = b \\ & x \geq 0, \end{array} \tag{1.1}$$

where $c \in \mathbf{R}^n$, $b \in \mathbf{R}^m$, $A \in \mathbf{R}^{m \times n}$ $(m < n)$ and A has full rank m. The first-order optimality conditions for the linear program (1.1) are:

$$F(x, y, \lambda) \equiv \begin{pmatrix} Ax - b \\ A^T\lambda + y - c \\ XYe \end{pmatrix} = 0 \tag{1.2}$$

and

$$(x, y) \geq 0 \tag{1.3}$$

where $X = diag(x)$, $Y = diag(y)$ and e is the n-vector of all ones. A point (x, y, λ) is said to be strictly feasible if it satisfies $Ax = b$, $A^T\lambda + y = c$ and $(x, y) > 0$. A solution pair (x, y) is said to satisfy *strict complementarity* if in addition to complementarity $XYe = 0$, it satisfies $x + y > 0$.

It is now well understood that the primal-dual logarithmic barrier function interior-point methods can be viewed as damped and perturbed Newton's method applied to the nonlinear system of equations (1.2). For more details, see Zhang, Tapia and Dennis [33]. The algorithmic framework for such methods is the following

Algorithm 1.1 (Primal-Dual Interior-Point Method)
Given a strictly feasible point (x^0, y^0, λ^0). *For* $k = 0, 1, \ldots,$ *do*

1. *Choose* $\sigma^k \in (0, 1)$ *and set* $\mu(x, y) = \sigma^k \frac{(x^k)^T y^k}{n}$.

2. *Solve the following system for* $(\Delta x^k, \Delta y^k, \Delta\lambda^k)$:

$$F'(x^k, y^k, \lambda^k)(\Delta x, \Delta y, \Delta\lambda) = -F(x^k, y^k, \lambda^k) + \mu(x^k, y^k)\hat{e} \tag{1.4}$$

3. Choose a step-length $\alpha^k = \min(1, \tau^k \hat{\alpha}^k)$ *for* $\tau^k \in (0,1)$ *and*

$$\hat{\alpha}^k = \frac{-1}{\min((X^k)^{-1}\Delta x^k, (Y^k)^{-1}\Delta y^k)}.$$

4. Form the new iterate

$$(x^{k+1}, y^{k+1}, \lambda^{k+1}) = (x^k, y^k, \lambda^k) + \alpha^k(\Delta x^k, \Delta y^k, \Delta\lambda^k).$$

Note that the choice of step-length α^k guarantees $(x^{k+1}, y^{k+1}) > 0$. In Step 2 $\hat{e} = (0, \ldots, 0, 1, \ldots, 1)^T$, with $n + m$ zero components.

We use the notation

$$\mathcal{B}(x) = \{i : x_i = 0, \quad 1 \leq i \leq n\}$$

to denote the set of indices of zero variables at a feasible point x of problem (1.1). Notice that $\mathcal{B}(x^*)$ may be different for different solutions x^* of the same linear programming problem. However, Theorem 2.1 of Section 2 implies that $\mathcal{B}(x^*)$ is invariant with respect to solutions in the interior of the solution set of (1.1). Hence in this case we may denote the set of indices of zero variables at any solution x^* in the relative interior of the solution set by $\mathcal{B}^*$ and no confusion will arise.

This paper is organized as follows. The structure of the solution set of the linear programming problem is studied in Section 2. In Section 3, we define the indicator function and list some properties that a good indicator should possess. In Section 4 we study one of the earliest indicators proposed, and probably the most frequently used indicator, for identifying $\mathcal{B}(x^*)$, namely the variables as indicators. We demonstrate both theoretically and numerically that this indicator has serious disadvantages. The primal-dual indicator which has been used recently by several researchers is investigated in the context of primal-dual interior-point methods in Section 5. In Section 6 we study the Tapia indicator for the linear programming problem and discuss its behavior in several interior-point methods. Numerical experiments are given in Section 7. These numerical experiments include the study of the usefulness of the variables as indicators as well as the usefulness of the primal-dual indicator in primal-dual interior-point methods. They also include comparisons between the variables as indicators, the primal-dual indicator and the Tapia indicator. Concluding remarks are given in Section 8.

2 Structure of The Solution Set

The structure of the solution set of the linear programming problem will play an important role in explaining the behavior of certain indicators. For this reason we begin this study with an investigation of the structure of the solution set and in particular the distribution of strict complementary solutions within the solution set. We establish our main result for a larger class of problems, namely for monotone complementarity problems, since the proof is essentially the same as that for linear programming.

First, we need some preliminary concepts. Following McLinden [16] and Güler and Ye [9], by the *support* $\sigma(v)$ for $v \in \mathbf{R}^n$ we mean the set of indices of positive components of v, i.e.

$$\sigma(v) = \{i : v_i > 0\}.$$

In particular, the support of a vector with no positive components is the empty set. Consider the *partial order* $\preceq$ on $\mathbf{R}^n$ defined by

$$v \preceq u \quad \textit{if} \quad \sigma(v) \subseteq \sigma(u)$$

Two vectors u and v are said to be *equivalent*, denoted by $u \sim v$, if $u \preceq v$ and $v \preceq u$. An element $v \in U \subseteq \mathbf{R}^n$ is said to be a $\preceq$-*maximal* element of U if

$$u \in U \quad \textit{and} \quad v \preceq u \Longrightarrow u \sim v$$

It is obvious that any subset U of $\mathbf{R}^n$ has at least one $\preceq$-maximal element. Güler and Ye [9] made the following straightforward but key observation concerning the structure of the solution set $\mathcal{T}$.

Observation 2.1 *If $U \subseteq \mathbf{R}^n$ is convex, then all maximal elements of U are equivalent.*

In particular, the above observation implies that if $U^m \subseteq U$ is the set of all maximal elements of U, then the *zero structure* of the elements of U^m is invariant throughout U^m.

A multivalued mapping $T : \mathbf{R}^n \to 2^{\mathbf{R}^n}$ is said to be a *monotone operator* if

$$y \in T(x) \quad \textit{and} \quad \bar{y} \in T(\bar{x}) \Longrightarrow (y - \bar{y})^T(x - \bar{x}) \geq 0.$$

A monotone operator T is said to be *maximally monotone* if the graph $G(T) = \{(x, y) \in \mathbf{R}^n \times \mathbf{R}^n : y \in T(x)\}$ is not properly contained in the graph of any other monotone operator. The *complementarity problem* associated with a maximally monotone operator T is

$$\textit{Find} \quad (x, y) \in G(T) \quad \textit{such that} \quad x^T y = 0 \quad \textit{and} \quad (x, y) \geq 0. \tag{2.1}$$

The set of solutions of problem (2.1) will be denoted by $\mathcal{T}$. Consider the set

$$\mathcal{M} = \{(x^m, y^m) \in \mathcal{T} : (x^m, y^m) \textit{ is maximal}\}.$$

The following proposition is a consequence of Lemma 2.3 in Güler [8]

Proposition 2.1 (Güler) *The solution set $\mathcal{T}$ for the monotone complementarity problem (2.1) is convex.*

The convexity of $\mathcal{T}$ implies, by Observation 2.1, that the zero structure of maximal solutions (or solutions that satisfy strict complementarity if they exist) is invariant. The invariance of the zero structure of solutions that satisfy strict complementarity (which are maximal in complementarity problems) was proved for a special class of linear programming problems by Charnes, Cooper and Thrall [3].

We denote the relative interior of a set U by $ri\ U$. See Rockafellar [22] for a definition of relative interior. Now we state our main result concerning the structure of the solution set $\mathcal{T}$ of problem (2.1).

Theorem 2.1 *Assume that the solution set $\mathcal{T}$ of the monotone complementarity problem (2.1) is nonempty, then*

$$ri\ (\mathcal{T}) = \mathcal{M}.$$

Proof: By Theorem 6.2 of Rockafellar [22], $ri\ (\mathcal{T})$ is nonempty and convex. Let $cl\ (\mathcal{T})$ and $\partial_r\mathcal{T} = (cl\ (\mathcal{T}))\backslash(ri\ (\mathcal{T}))$ denote the closure and the relative boundary of $\mathcal{T}$, respectively. There exists at least one maximal element (x^m, y^m) of $\mathcal{T}$ that lies in $ri\ (\mathcal{T})$. Choose an arbitrary point $(x^b, y^b) \in \partial_r\mathcal{T}$. Consider the convex combination

$$(x^\phi, y^\phi) = \phi(x^b, y^b) + (1-\phi)(x^m, y^m) \quad 0 \leq \phi < 1.$$

It follows directly from Theorem 6.1 of Rockafellar [22] that $(x^\phi, y^\phi) \in ri\ (\mathcal{T})$. We have

$$\sigma(x^\phi, y^\phi) = \sigma(x^b, y^b) \cup \sigma(x^m, y^m) = \sigma(x^m, y^m),$$

which shows that

$$(x^\phi, y^\phi) \in \mathcal{M}, \qquad \phi \in [0, 1).$$

Now consider any point $(x^i, y^i) \in ri\ (\mathcal{T})$. Convexity of $\mathcal{T}$ implies that (x^i, y^i) lies on some line segment connecting (x^m, y^m) and some point $(x^b, y^b) \in \partial_r\mathcal{T}$. But we have seen that all these line segments lie in $\mathcal{M}$. Thus $(x^i, y^i) \in \mathcal{M}$. This proves that

$$ri\ \mathcal{T} \subseteq \mathcal{M}. \tag{2.2}$$

Now we will show that any point in the relative boundary of $\mathcal{T}$ cannot lie in $\mathcal{M}$. Consider $(x^b, y^b) \in \partial_r \mathcal{T}$. Suppose $(x^b, y^b) \in \mathcal{M}$. Let $(x^i, y^i) \in ri\ (\mathcal{T})$. From (2.2), $(x^i, y^i) \in \mathcal{M}$ and hence $(x^i, y^i) \sim (x^b, y^b)$. Without loss of generality, assume that

$$\sigma(x^i) = \{1, \ldots, r\} = \sigma(x^b)$$

and

$$\sigma(y^i) = \{s, \ldots, 2n\} = \sigma(y^b),$$

where $r \leq s$. Now any point on the relative boundary of $\mathcal{T}$ is the intersection of the graph $G(T)$ with at least one of the hyperplanes

$$x_i = 0 \qquad i = 1, \ldots, r$$

or

$$y_i = 0 \qquad i = s, \ldots, 2n.$$

This contradicts the maximality of (x^b, y^b) and completes the proof $\square$.

It is known that the primal-dual formulation of the linear programming problem can be stated as a monotone complementarity problem with graph

$$G_{LP}(T) = \{(x, y) : Ax = b,\ A^T\lambda + y = c \ \textit{for some}\ \lambda \in \mathbf{R}^m.\} \tag{2.3}$$

For more details see Güler and Ye [9]. Consider the set $\mathcal{S}$ consisting of the solutions of the linear programming problem that satisfy strict complementarity. Clearly any element of $\mathcal{S}$ is maximal in $\mathcal{T}$. As an immediate consequence of Theorem 2.1, we can determine the distribution of $\mathcal{S}$ within $\mathcal{T}$ for linear programming problems.

Corollary 2.1 *Assume that there exists at least one point* $(x^o, y^o) \in G_{LP}(T)$ *such that* $(x^o, y^o) > 0$. *Then*

$$ri(\mathcal{T}) = \mathcal{S}.$$

Proof: The well-known Goldman-Tucker theorem states that $\mathcal{S} \neq \emptyset$ for linear programming problems. The proof now follows directly from Theorem 2.1 $\square$.

This result concerning the structure of the solution set of the linear programming problem can also be derived from a study of the structure of the solution set of the linear complementarity problem carried out by Jansen and Tijs [10].

Assume that an interior-point method generates iterates that satisfy

$$\frac{\min(X^k Y^k e)}{(x^k)^T y^k} \geq \beta,$$

where β is positive number. A straightforward implication of Theorem 2.1 and Lemma 2 of Güler and Ye [9] is that $\{(x^k, y^k)\}$ can not have limit points that lie on the relative boundary of the solution set. In particular this means that interior-point methods that satisfy such a bound cannot generate iterates which converge to a vertex solution of the linear programming problem, unless of course the problem has a unique solution.

In Sections 3 and 4, we discuss the effect that the structure of the solution set has on the behavior of certain indicators.

3 The Indicator Function : Definition and History

Following Tapia [23], we use the term *indicator* to denote a function that identifies constraints that are active at a solution of a constrained optimization problem. Although indicators have been used extensively since at least 1984 in the context of linear programming, a unified framework that includes a definition, desired properties, and general guidelines for their use has not been provided. It is a main objective of this work to provide such a framework.

Throughout this paper we will consider iterative procedures of the generic form

$$z^{k+1} = z^k + \alpha^k \Delta z^k.$$

The majority of our discussion centers around primal-dual interior-point methods and in this case our iterates have the form

$$z^k = (x^k, y^k, \lambda^k).$$

In a primal method, e.g., the Karmarkar algorithm, we have $z^k = x^k$. It is natural to define the indicator as a function of z^k and Δz^k and perhaps an auxiliary variable which may represent the step-length α^k or other quantities. However, in the interest of conciseness we will consider the auxiliary variable implicitly in the definition and not formally state its dependence.

Let $(z^k, \Delta z^k)$ be generated by an iterative procedure of the generic form discussed above. By an *indicator* function I we mean any function which assigns to $(z^k, \Delta z^k)$ an n-vector of extended reals $I(z^k, \Delta z^k)$ and satisfies the property that if $z^k \to z^*$, then for $i = 1, \ldots, n$

$$\lim_{k \to \infty} I_i(z^k, \Delta z^k) = \begin{cases} \phi_i, & \text{if } i \in \mathcal{B}(x^*) \\ \theta_i, & \text{if } i \notin \mathcal{B}(x^*) \end{cases} \tag{3.1}$$

for some θ_i and ϕ_i satisfying $\min_i \theta_i > \max_i \phi_i$.

Throughout this paper we use the terms *indicator* and *indicator function* interchangeably. Whenever it is appropriate and no confusion will arise, we write $I(x^k)$, $I(y^k)$, or $I(x^k, y^k)$ instead of $I(z^k, \Delta z^k)$.

It is desirable that an indicator function I possess the following ideal properties:

1. the *sharp separation* property

$$\min_{i \notin \mathcal{B}(x^*)} \theta_i >> \max_{i \in \mathcal{B}(x^*)} \phi_i;$$

2. the *uniform separation* property

$$\theta_i = \theta \; ; \quad i \notin \mathcal{B}(x^*) \quad \textit{and} \quad \phi_i = \phi \; ; \quad i \in \mathcal{B}(x^*)$$

for some constants θ and ϕ. In this case, it is also desirable that θ and ϕ be independent of both the solution and the problem;

3. the indicator is inexpensive to compute;

4. the indicator sequence $\{I(z^k, \Delta z^k)\}$ converges to its limit faster than z^k converges to z^*;

5. the indicator gives reliable information early on in the iterative process;

6. the indicator is scale independent, i.e. it does not change if the variables are scaled by any positive diagonal matrix.

Clearly, an indicator may be effective and not possess all these ideal properties. However, it is our considered opinion that both the sharp and the uniform separation properties are extremely important. Several numerical experiments demonstrating the importance of these two properties are given in Section 7.

If some members of $\mathcal{B}(x^*)$, for some solution x^*, are identified early on in an iterative procedure, then this information can be used to

1. reduce the dimension of the problem by dropping the columns of A corresponding to the zero variables. This reduction may result in significant savings in computational work.

2. help recover an optimal basis for the linear program using techniques along the lines of Megiddo [18].

The task of predicting $\mathcal{B}(x^*)$ has been considered in recent years by many researchers and various indicators have been proposed for this purpose. Gill *et al* [7], Karmarkar and Ramakrishnan [11], McShane, Monma and Shanno [17], Tone [28], Lustig, Martsen, and Shanno [14], Dantzig and Ye [4], and Boggs, Domich, Donaldson and Witzgall [2], among others, proposed the use of variables, either primal or dual, to predict members of $\mathcal{B}(x^*)$. Tapia [23] introduced two indicators in the context of identifying active constraints in nonlinear constrained optimization problems. Kojima [12] proposed an indicator for use in Karmarkar-type algorithms. Ye [30] and Todd [26] introduced two indicators for Karmarkar-type and primal-dual algorithms. Tapia and Zhang [24] proposed an indicator that can be used in primal, dual, or primal-dual interior-point methods. Kovacevic-Vujcic [13] introduced an indicator that is superlinearly faster than the variables in Karmarkar-type methods. The ratio between primal variables and dual slacks was used as an indicator by several researchers including Gay [6], Ye [32], and Lustig [15]. Mehrotra [19] used an indicator based on the relative change in the dual slack variables. Resende and Veiga [21] used the reciprocal of the dual slack variables as indicators. Many of these indicators have been cataloged in Appendix A along with some critical comments.

4 The Variables as Indicators

In both linear and nonlinear programming the use of the variables as indicators is a part of the optimization folklore. In linear programming, Gill *et al.* [7] set primal variables with very small absolute values to zero. Karmarkar and Ramakrishnan [11] suggested using the dual-slack variables as indicators. McShane, Monma and Shanno [17] suggested setting those variables with small absolute value and large dual slack to zero. Boggs, Domich, Donaldson and Witzgall [2] used the primal slacks with large values to remove constraints from the problem using an algorithm based on the method of centers. While this indicator is readily accessible, it has serious disadvantages. It does not satisfy either the sharp separation or the uniform separation property and is scale dependent. Another disadvantage is that in general it does not give information soon enough to save computational work or improve the performance of the algorithm. Some researchers were aware of the deficiencies of this indicator and therefore tried to use it in a conservative manner. Unfortunately, the undesirable aspects of this indicator, namely the lack of both the sharp and uniform separation properties, are inherent in the convergence particulars of interior-point methods and in the structure of the solution set of the problem. In the following, we demonstrate the detrimental effect that

these two factors can have on the behavior of the variables when used as indicators.

1. The effect of convergence particulars of interior-point methods:
   A main difficulty in using an indicator function arises when a threshold sequence $\{\delta^k_{zero}\}$ is to be used in the identification test,

$$I_i(x^k) \leq \delta^k_{zero} \Longrightarrow x^*_i = 0. \tag{4.1}$$

   Since for this indicator $I_i(x^k) = x^k_i$, the sequence $\{\delta^k_{zero}\}$ must satisfy

$$\max_{i \in \mathcal{B}(x^*)} \hat{x}_i < \delta^*_{zero} < \min_{i \notin \mathcal{B}(x^*)} \hat{x}_i$$

   where $\delta^*_{zero} = \lim_{k \to \infty} \delta^k_{zero}$ and $\hat{x}$ is the approximate solution given by the algorithm. In order to identify zero variables early on, the threshold sequence should satisfy

$$\max_{i \in \mathcal{B}(x^*)} x^k_i < \delta^k_{zero} < \min_{i \notin \mathcal{B}(x^*)} x^k_i, \tag{4.2}$$

   for $k > K$, where K is a relatively small positive integer. Since $\max_{i \in \mathcal{B}(x^*)} \hat{x}_i$ and $\min_{i \notin \mathcal{B}(x^*)} \hat{x}_i$ are not known a priori, it is very difficult to construct a sequence $\{\delta^k_{zero}\}$ that satisfies these conditions. In our numerical studies we often found that there was a large gap between components of the final approximate solution generated by the interior-point method and the components of x^*. In fact, we observed numerically the annoying phenomenon that for a final approximate solution generated by an interior-point method, we may have

$$\max_{i \in \mathcal{B}(x^*)} \hat{x}_i > \min_{i \notin \mathcal{B}(x^*)} \hat{x}_i.$$

   This shows that, in practice, a threshold sequence $\{\delta^k_{zero}\}$ that satisfies (4.2) may not exist.

2. The effect of the structure of the solution set:
   An implication of Corollary 2.1 is that if (x^b, y^b) is a point on the relative boundary of $\mathcal{T}$, then there exits at least one component, say x^b_j, such that $x^b_j = 0$, while $x^s_j > 0$ for all $x^s \in ri(\mathcal{T})$. The convexity of $\mathcal{T}$ implies that x^s_j is arbitrarily small for solutions x^s arbitrarily close to the boundary while the corresponding dual slacks are zero. So the use of variables as indicators may be misleading even in the presence of both primal and dual information. Finally, the geometry of the problem may be such that $\mathcal{T}$ is so thin that some positive component x^*_i has a very small value for *all* solutions in the interior of $\mathcal{T}$.

Numerical experiments with the variables as indicator are presented in Section 7. These experiments speak strongly against the use of variables as indicator.

5 The Primal-Dual Indicator

Consider the indicator function

$$I(z^k, \Delta z^k, \alpha^k) = (Y^{k+1})^{-1} X^{k+1} e, \tag{5.1}$$

where $z^{k+1} = z^k + \alpha^k \Delta z^k$, $Y = diag(y)$ and $X = diag(x)$. We will call this indicator the primal-dual indicator since it uses both primal and dual information. This indicator was used recently by several researchers. e.g. Gay [6], Ye [32] and Lustig [15]. If strict complementarity holds, then the primal-dual indicator satisfies both the sharp and the uniform separation properties, namely

$$\lim_{k \to \infty} I(x^k, y^k, \alpha^k) = \begin{cases} 0 & \text{if } i \in \mathcal{B}(x^*) \\ \infty & \text{if } i \notin \mathcal{B}(x^*) \end{cases}$$

The primal-dual indicator does not require nondegeneracy or feasible iterates. Unfortunately, it is scale dependent. Another disadvantage is that the identification test $I_i(z^k, \Delta z^k, \alpha^k) \leq \delta_{zero} \Longrightarrow x_i^* = 0$ is extremely sensitive to the choice of δ_{zero}. Our numerical experiments show that the primal-dual indicator gives reliable information only if the iterates are very close to the solution. In fact in many cases this indicator could not identify all zero variables even at the final approximate solution generated by the interior-point algorithm. These observations motivated us to study in detail the structure of the primal-dual indicator in primal-dual interior-point methods.

The Primal-Dual Indicator in Interior-Point Methods

In the framework of primal-dual interior-point methods, the behavior of the primal-dual indicator can be explained using the following proposition.

Proposition 5.1 *Assume that the sequence of iterates $\{(x^k, y^k, \lambda^k)\}$ has been generated by Algorithm 1. Then for $i = 1, \ldots, n$*

$$\frac{x_i^{k+1}}{y_i^{k+1}} = (2 - \alpha^k) \frac{x_i^k}{y_i^{k+1}} - \frac{x_i^k}{y_i^k} + \alpha^k \frac{\sigma^k}{n} \frac{(x^k)^T y^k}{y_i^k y_i^{k+1}}. \tag{5.2}$$

Proof: The perturbation of the linearized complementary slackness equation gives

$$X(y + \alpha \Delta y) + Y(x + \alpha \Delta x) = (2 - \alpha) XYe + \alpha \mu(x, y) e.$$

Thus

$$X^k Y^{k+1} e + Y^k X^{k+1} e = (2 - \alpha^k) X^k Y^k e + \alpha^k \mu^k e.$$

Multiplying both sides by $(Y^k)^{-1}(Y^{k+1})^{-1}$ completes the proof $\square$.

Examining equation (5.2), we believe that the undesirable behavior of the primal-dual indicator is due to one of the following situations.

- If $x_i^k \to x_i^* > 0$ for some i, then $y_i^* = 0$. In this case

$$\frac{x_i^k}{y_i^{k+1}} - \frac{x_i^k}{y_i^k} \to \infty - \infty,$$

  which is essentially an undefined quantity; and it is not clear that the primal-dual indicator will approach infinity fast enough to be of effective use.

- If $x_i^k \to x_i^* = 0$ for some i, then $y_i^* > 0$. In this case each of the three terms on the right-hand side of (5.12) tends to zero. However, if $y_i^* > 0$, but has a small value (e.g. 10^{-4}) then the denominator of the third term will be much smaller and could cause the primal-dual indicator to have large values. This problem can be partially corrected if we let $\sigma^k \to 0$ fast. Unfortunately, if $y_i^* > 0$ but has a very small value , which may occur as argued in Section 2, then all three terms in (5.2) become small only when the iterates are extremely close to a solution.

6 The Tapia Indicator

In order to identify active and inactive constraints for a nonlinear constrained optimization problem, Tapia [23] suggested using the quotient of successive Lagrange multipliers and the quotient of successive slack variables as indicators in the context of an iterative procedure that enforces linearized complementarity.

For linear programming problems, the Tapia indicators are

$$I_p(x^k) = (X^k)^{-1}X^{k+1}e, \tag{6.1}$$

and

$$I_d(y^k) = e - (Y^k)^{-1}Y^{k+1}e, \tag{6.2}$$

where $x^{k+1} = x^k + \Delta x$, $y^{k+1} = y^k + \Delta y$ and as before $e = (1, \ldots, 1)^T$.

El-Bakry [5] studied the behavior of the Tapia indicators in primal-dual interior-point methods. He also used both indicators to identify and remove zero variables in primal-dual interior-point methods. Some of these results are presented in Section 7. Mehrotra [20] used the Tapia primal indicator I_p in his perturbation method to identify vertex solution

using interior-point methods. The properties and the behavior of these indicators in various interior-point methods are discussed in detail in Section 6.1. For the sake of completeness we present the following proposition which establishes the sharp and uniform separation properties for the Tapia indicators. It is essentially a specialization to problem (1.1) of a general result proved by Tapia [23].

Proposition 6.1 (Tapia) *Assume that the sequence of iterates $\{(x^k, y^k, \lambda^k)\}$, generated by an iterative procedure, converges to a strict complementary solution $\{(x^*, y^*, \lambda^*)\}$ of the first-order necessary conditions for problem (1.1). Assume further that linearized complementarity*

$$X^k \Delta y^k + Y^k \Delta x^k = -X^k Y^k e$$

is satisfied. Then for $i = 1, \ldots, n$

$$\lim_{k-\infty} \frac{x_i^{k+1}}{x_i^k} \longrightarrow \begin{cases} 0 & \text{if } i \in \mathcal{B}(x^*) \\ 1 & \text{if } i \notin \mathcal{B}(x^*) \end{cases} \tag{6.3}$$

and

$$\lim_{k-\infty} (1 - \frac{y_i^{k+1}}{y_i^k}) \longrightarrow \begin{cases} 0 & \text{if } i \in \mathcal{B}(x^*) \\ 1 & \text{if } i \notin \mathcal{B}(x^*), \end{cases} \tag{6.4}$$

where $x_i^{k+1} = x_i^k + \Delta x_i$ and $y_i^{k+1} = y_i^k + \Delta y_i$

It is obvious from the above proposition that $\frac{\Delta x_i^k}{x_i^k} \to 0$ for $i \notin \mathcal{B}^*$, and $\frac{\Delta x_i^k}{x_i^k} \to -1$ for $i \in \mathcal{B}^*$. Hence, the relative change $\frac{\Delta x^k}{x^k}$, which is a restatement of the Tapia indicator, can serve as an indicator. We emphasize that in general the relative change is not a good indicator, a fact well-known from the elementary theory of sequences. In the context of linear programming problems, it is the linearized complementarity and the additional condition of strict complementarity that make the Tapia indicators (or equivalently the relative change) effective. However, in the primal-dual interior-point methods, the iterates satisfy a perturbation of the linearized complementarity equation, namely

$$X \Delta y + Y \Delta x = -XYe + \mu(x, y)e,$$

where $\mu(x, y)$ is the barrier parameter. The question as to whether the Tapia indicators can retain their useful properties in that framework is addressed below.

The Tapia Indicators in Interior-Point Methods

It was observed, in our numerical experiments, that the Tapia indicator is more effective than

the primal-dual indicator in identifying zero variables in most test problems, see Section 7. This led us to investigate the structure of the Tapia indicators in the framework of primal-dual interior-point methods for linear programming. As we shall soon see the fit is surprisingly good.

Proposition 6.2 *Let the sequence of iterates* $\{(x^k, y^k, \lambda^k)\}$ *be generated by Algorithm 1. Assume that*

1. $(x^k)^T y^k \to 0$.
2. $\frac{min(X^k Y^k e)}{(x^k)^T y^k} \geq \frac{\gamma}{n}$ *for all* k *and some* $\gamma \in (0,1)$.
3. $\sigma^k \to 0$ *and* $\tau^k \to 1$.

Then for $i = 1, \ldots, n$

$$\lim_{k-\infty} \frac{x_i^{k+1}}{x_i^k} \longrightarrow \begin{cases} 0 & \text{if } i \in \mathcal{B}(x^*) \\ 1 & \text{if } i \notin \mathcal{B}(x^*) \end{cases}$$

and

$$\lim_{k-\infty} (1 - \frac{y_i^{k+1}}{y_i^k}) \longrightarrow \begin{cases} 0 & \text{if } i \in \mathcal{B}(x^*) \\ 1 & \text{if } i \notin \mathcal{B}(x^*) \end{cases}$$

where $x^{k+1} = x^k + \beta^k \Delta x$ *and* $y^{k+1} = y^k + \beta^k \Delta y$ *for any* $\beta^k \in [\alpha^k, 1]$.

Proof: Consider

$$X\Delta y + Y\Delta x = -XYe + \mu(x,y)e.$$

It is clear that

$$X(y + \beta\Delta y) + Y(x + \beta\Delta x) = (2 - \beta)XYe + \beta\mu(x,y)e.$$

Hence

$$(X^k)^{-1}x^{k+1} + (Y^k)^{-1}y^{k+1} = (2 - \beta^k)e + \beta^k\sigma^k\frac{(x^k)^T y^k}{n}(X^kY^k)^{-1}e. \tag{6.5}$$

Very recently Tapia, Zhang and Ye [25] demonstrated that $(\Delta x_i^k, \Delta y_i^k) \to 0$; hence

$$\frac{\Delta x_i^k}{x_i^k} \to 0 \quad \text{for} \quad i \notin \mathcal{B}^* \quad \text{and} \quad \frac{\Delta y_i^k}{y_i^k} \to 0 \quad \text{for} \quad i \in \mathcal{B}^*. \tag{6.6}$$

It follows from (6.6) and the definition of $\hat{\alpha}^k$ that $\hat{\alpha}^k \to 1$. Hence $\alpha^k \to 1$ and therefore $\beta^k \to 1$. The result follows now from (6.5) and assumptions 2 and 3 $\square$.

The fact that asymptotically we have $\beta^k \to 1$ motivated us to use $\beta^k = 1$ in the calculation of the Tapia indicators. This proved to give superior results in our numerical experiments.

It is extremely satisfying to us that the conditions which guarantee the usefulness of the Tapia indicators, i.e. conditions 1-3 in Proposition 6.2, are exactly the conditions which guarantee fast local convergence, i.e. superlinear convergence of the duality gap sequence to zero (see Zhang, Tapia and Dennis [25]). It is equally satisfying that we obtain this pleasant behavior of the Tapia indicators without the assumption that the iteration sequence converges, as Tapia [23] assumed in Proposition 6.1.

It is clear from the proof of Proposition 6.2 that the assumption that $\sigma^k \to 0$ is crucial in obtaining the 0-1 separation property of the Tapia indicator.

Now we will discuss the use of the Tapia indicator in several interior-point methods. The search directions in many interior-point methods satisfy the following system of equations (see Ye [31])

$$D_x \Delta x + D_y \Delta y = \sigma \frac{x^T y}{n} e - XYe,$$

$$A\Delta x = 0,$$

$$A^T \Delta \lambda + \Delta y = 0.$$

Different interior-point methods correspond to different choices of D_x and D_y. The first equation is of particular interest in our analysis. If $D_x, D_y > 0$ then this equation can be written

$$D_y^{-1} \Delta x + D_x^{-1} \Delta y = \sigma \frac{x^T y}{n} D_y^{-1} D_x^{-1} e - D_y^{-1} D_x^{-1} XYe. \tag{6.7}$$

For the Tapia indicators to retain their effectiveness we should have:

$$D_y^{-1} \Delta x + D_x^{-1} \Delta y \to X^{-1} \Delta x + Y^{-1} \Delta y$$

and

$$\sigma \frac{x^T y}{n} D_y^{-1} D_x^{-1} + D_y^{-1} D_x^{-1} XYe \to -e.$$

Examining the different choices for D_x, D_y and σ for the primal (or dual) affine scaling algorithms, the primal (or dual) potential-reduction algorithms and the primal (or dual) path-following algorithms, it is easy to see that it is extremely unlikely that the Tapia indicators will retain their effectiveness in this context.

In conclusion, we believe that primal-dual algorithms where $\sigma^k \to 0$ are the natural setting for the use of the Tapia indicators. Moreover, in this case the Tapia indicators and the primal-dual interior-point methods are an excellent match.

Finally we list the properties of the Tapia indicators that make them effective in practice as demonstrated in Section 7. These properties are as follows.

1. They are inexpensive to compute.
2. They satisfy both the uniform and the sharp separation properties. The indicator parameters $\phi = 0$ and $\theta = 1$ are independent of the problem.
3. From our numerical experience, the Tapia indicators give reliable information early.
4. They are scale independent when the variables are scaled by any positive diagonal matrix.
5. They do not require feasibility or nondegeneracy.
6. They do not require convergence of the iteration sequence.

7 Numerical Experience

In this section we present several numerical experiments with three indicators: the variables as indicators, the primal-dual indicator, and the Tapia indicator. The purpose of these experiments is to demonstrate the undesirability of variables as indicators, to study the behavior of the primal-dual indicator in primal-dual interior-point methods, and finally to compare between the ability of these indicators to identify zero variables in linear programming. These experiments are performed on a subset of the *netlib* test set using a predictor-corrector primal-dual interior-point code that was developed at Rice University. The code generates a sequence of iterates that approach feasibility and drive the absolute duality gap $c^T x - b^T y$ to zero. For numerical purposes the iterates are generated to drive the *relative gap* $\frac{c^T x - b^T y}{1+|b^T y|}$, rather than the absolute gap, to zero. We will say that a problem is solved to an accuracy of 10^{-d} for some positive integer d if the algorithm is terminated when

$$\max\left(\frac{|c^T x^k - b^T y^k|}{1+|b^T y^k|}, \frac{\|Ax^k - b\|_1}{1+\|x^k\|_1}, \frac{\|A^T\lambda^k + y^k - c\|_1}{1+\|\lambda^k\|_1 + \|y^k\|_1}\right) \leq 10^{-d}.$$

The experiments were done on a Sun 4/490 workstation with 64 Megabytes of memory.

7.1 The Undesirability of Variables as Indicators

The variables were probably the first indicators used to predict $\mathcal{B}^*$. In Section 4 we discussed, in detail, the disadvantages of this indicator. In our numerical experiments, we observed that in many test problems from the *netlib* collection it was extremely difficult, and sometimes even impossible, to distinguish between zero and nonzero variables using only the values of these variables. In the following we will investigate some ideas that have been proposed for using the variables to determine $\mathcal{B}^*$.

- Set $x_i = 0$ if $x_i^k \leq \delta_{zero}$.

  Gill *et al.* [7] chose $\delta_{zero} = 10^{-8}$. In many cases the algorithm terminated with most of the zero variables having values greater than 10^{-5} , e.g. SHARE1B and SCAGR25 (in fact some problems had zero variables of order 10^{-2} when the algorithm terminated, e.g. SCAGR25). So, this choice is very conservative. If we use $\delta_{zero} = 10^{-6}$ some of the nonzero primal variables in PILOT4 and some of the nonzero dual elements in CYCLE have values less than this threshold. So, a good choice of δ_{zero} is extremely difficult to find. We also observed that for a particular problem zero elements may have a wide range of magnitude at the approximate solution generated by the primal-dual interior-point method. For example in GREENBEA the zero variables have magnitudes in the range $(10^{-1}, 10^{-5})$.

- Set $x_i = 0$ if $x_i^k \leq \delta_x$ and $y_i^k \geq \delta_y$.

  From our experience with the *netlib* problems we observed that the final approximate solution generated by a primal-dual interior-point method may not have enough separation between the primal variables and the dual slacks. For example, the pair (x_i, y_i), for some values of i, is of order $(10^{-6}, 10^{-4})$ in LOTFI and BANDM, $(10^{-2}, 10^{-1})$ in SCAGR25 and $(10^{-4}, 10^{-4})$ in both SCAGR25 and FFFFF800 which shows that choosing effective thresholds δ_x and δ_y for a given set of problems is practically impossible. It is also worth mentioning that in SEBA the pair (x_i, y_i), for some values of i is of the order $(10^{-3}, 10^{-1})$ while for a different value of i it is of order $(10^{-4}, 10^{-2})$. This shows that choosing these thresholds is practically impossible even for variables in the same problem. It is interesting to observe that in problem NESM the algorithm terminated with a certain pair equal to $(1.543, 0.000015)$, which gives the impression that the dual slack is zero and the primal variable is nonzero at the solution. Solving the problem to an accuracy of 10^{-15} reduced the value of x_i to 0.21×10^{-9} while the value of dual

slack remained the same, see Table 1. In Table 1, the letter N in the last row means that we could not solve the problem to an accuracy of 10^{-16}. This example shows how misleading the variables can be.

Table 1: NESM: A particular solution pair

	relative gap	x	y	TAPIA For x	PRIMAL DUAL
28	10^{-6}	21.793	1.58D-5	0.878	0.12D+7
29	10^{-6}	17.905	1.57D-5	0.784	0.10D+7
30	10^{-7}	7.946	1.58D-5	0.443	0.50D+6
31	10^{-8}	1.543	1.59D-5	0.26D-2	0.13D+4
32	10^{-8}	0.2648	1.59D-5	0.14D-3	0.14D+2
33	10^{-9}	0.35D-1	1.59D-5	0.57D-5	0.96D-1
34	10^{-11}	0.13D-2	1.59D-5	0.25D-6	0.45D-3
35	10^{-12}	0.44D-4	1.59D-5	0.19D-7	0.16D-5
36	10^{-13}	0.18D-5	1.59D-5	0.26D-8	0.47D-8
37	10^{-15}	0.21D-9	1.59D-5	0.19D-9	0.16D-12
38	10^{-16}	N	N	N	N

Some authors propose choosing δ_{zero} adaptively. Although this idea may slightly improve the results obtained by using the variables as indicators, we believe it will not account for that much improvement. The reason is that, as mentioned in Section 4, at an approximate solution of the primal-dual interior-point method the variables x_i with $i \in \mathcal{B}^*$ have small, but not zero, values. In fact, we observed that the algorithm may terminate with some of these values relatively large. In several cases, some of these values were larger than values of the positive variables, e.g. GREENBEA and NESM. This phenomenon implies that, at least for these problems, the choice for δ_{zero} is practically impossible. One may then suggest that we solve the problem to a greater accuracy so that there is a clear distinction between zero and nonzero variables. Although this idea is conceptually correct, it overlooks three important issues.

1. It is not, generally, known a priori to what accuracy a particular problem should be solved. For example, we solved problem D2Q06C to an accuracy of 10^{-11} and we still

had some pairs (x_i, z_i) of order $(10^{-5}, 10^{-4})$ and even $(10^{-4}, 10^{-4})$. Another example is problem CYCLE which we solved to an accuracy of 10^{-12} and we had pairs of order $(10^{-9}, 10^{-5})$, $(10^{-8}, 10^{-6})$, $(10^{-8}, 10^{-7})$ and $(10^{-7}, 10^{-7})$.

2. Since the linear systems we are solving are necessarily singular for degenerate problems, in some problems these systems may become very ill-conditioned near a solution so that we cannot solve the problem to the desired accuracy. An example of this is problem NESM. We solved this problem to an accuracy of 10^{-15}. We observed that some pairs (x_i, y_i) were of the order $(10^{-7}, 10^{-7})$. Unfortunately, we cannot ask for more accuracy with the given precision.

3. Finally, even if we know the required accuracy and we are able to solve the problem to that accuracy, we miss one of the main objectives of the indicators. This objective is to predict zero variables as early as possible in order to save computational work.

7.2 The Behavior of Several Indicators

This experiment compares the ability of three indicators, the variable as indicator, the Tapia indicator and the primal-dual indicator, to identify zero variables. Naturally, the number of zero variables predicted by each indicator will depend on the indicator's threshold δ_I in the identification test

$$I_i(x^k) \leq \delta_I \Longrightarrow x_i^* = 0.$$

For this experiment we choose $\delta_{variables} = 10^{-6}$, $\delta_{Tapia} = 0.1$ and $\delta_{primal-dual} = 0.1$. The choice of this value of $\delta_{primal-dual}$ is based upon our own experience that if $\delta_{primal-dual} > 1$, the primal dual indicator predicts the wrong set of zero variables more often. In Table 2, the first column shows the problem name. Corresponding to each problem the second column gives the name of the indicator used to predict members of $\mathcal{B}^*$. Columns 3 to 9 gives the quotient $\frac{|\mathcal{B}(x^k)|}{|\mathcal{B}^*|}$, i.e. the percentage of the zero variables correctly identified at the corresponding iteration. These columns correspond to the last 7 iterations before the algorithm terminated. Here M is the total number of iterations required to solve the problem to an accuracy of 10^{-8}. We stress that we count the predicted zero variables only when the set $\mathcal{B}^k \subseteq \mathcal{B}^*$, i.e. $\mathcal{B}^k$ does not have any member i with the corresponding x_i^* positive at the solution.

For the set of problems given in Table 2, the Tapia indicator shows a better ability to predict zero variables. It is also clear that the ability of the variables to determine $\mathcal{B}^*$ early is

minimal. An obvious example is problem GROW7, where the variables, used as indicators, were able to determine only one element of $\mathcal{B}^*$ when the algorithm terminated. All the problems in Table 2 are solved to an accuracy of 10^{-8}. It is worth mentioning here that at

Table 2: Comparison between indicators

PROBLEM	INDICATOR	M−6	M−5	M−4	M−3	M−2	M−1	M
AFIRO	VARIABLES	0	0	0	0	0	0	100
(M=8)	PRIMAL-DUAL	0	0	0	0	0	100	100
	TAPIA	0	0	3	14	93	100	100
ADLITTLE	VARIABLES	0	0	0	0	7	78	100
(M=12)	PRIMAL-DUAL	0	0	0	0	0	0	100
	TAPIA	7	9	52	79	94	100	100
SCSD1	VARIABLES	0	0	0	0	0	100	100
(M=12)	PRIMAL-DUAL	0	0	0	0	0	100	100
	TAPIA	0	0	25	90	100	100	100
SHIP04L	VARIABLES	0	0	11	11	11	98	100
(M=17)	PRIMAL-DUAL	0	0	0	0	0	0	100
	TAPIA	0	0	97	10	100	100	100
SHARE2B	VARIABLES	0	0	0	0	0	73	100
(M=12)	PRIMAL-DUAL	0	0	69	76	81	100	100
	TAPIA	3	30	63	79	97	100	100
GROW7	VARIABLES	0	0	0	0	0	0	2
(M=14)	PRIMAL-DUAL	0	0	0	0	0	98	100
	TAPIA	0	0	0	0	11	96	100

iteration 10 in problem SCSD1 the set $\mathcal{B}^{10}$ that is determined by the primal-dual indicator contains all correct zeros as well as two nonzero variables. This shows that one should be cautious when using the primal-dual indicator. This behavior is observed in other problems as well.

Examples of the undesirable behavior of the primal-dual indicator for some variables in problem SCSD1 is shown in Figures 1 and 2. In Figure 1, the variable is not zero at the solution. The corresponding primal-dual indicator has very small values (less than 1.0)

for 9 iterations (the algorithm terminated at iteration 12) giving the impression that this variable is zero at the solution. In Figure 2, the variable is zero at the solution. The primal-dual indicator has very large values (values larger than 300) until iteration 11. Note that this problem is very well behaved in the sense that the variables start approaching their optimal values reasonably early. Figure 3 shows the behavior of the Tapia indicator for all the variables in the same problem. Although a few of the indicators converge to their terminal values late, most indicators do give the correct information early. Figure 4 gives an example of the Tapia indicator when a *positive* variable has a very small value , i.e., 10^{-4}, where the solid line represents the variable and the dotted line represents the corresponding Tapia indicator. The indicator accurately predicts, starting from iteration 6, that the terminal value of this variable is not zero. Finally, we do not mean to imply that the primal-dual indicator always follows the pattern seen in Figures 1 and 2. In fact, Figure 5 shows an example in which the primal-dual indicator performs very well, again here the solid line represents the variable and the dotted line represents the corresponding primal-dual indicator. However, from our numerical experience, we believe that much care should be taken when the primal-dual indicator is used. Finally, Table 1 gives an example in which both the variables and the primal-dual indicator fail fo give the correct information when the algorithm terminated. The first column of that table gives the iteration count. The second and the third columns give the relative gap and the absolute gap, respectively. The last four columns give the values of the variable, the corresponding dual slack, the Tapia indicator and the primal-dual indicator, respectively, for a given variable in problem NESM. This variable is zero at the solution. We note that when the algorithm stops at iteration 31, the primal-dual indicator has a very large value. So, it fails to give the correct information at this iteration. Note that the Tapia indicator, at the same iteration, correctly indicates that this variable is zero at the solution. We had to take two more iterations in order for the primal-dual indicator to give the correct information.

7.3 A Generic Procedure

In the following we introduce a generic procedure to identify zero variables in linear programming problems. It is our considered opinion that any effective procedure of this kind should have three features

- An effective indicator.
- A good way of handling the information from the indicator.

- A mechanism to detect and recover if an error is made in predicting members of $\mathcal{B}^*$.

It is always beneficial to use more than one indicator (if they are not expensive).

Procedure 7.1
At iteration k,

(i) *Test for errors in $\mathcal{B}(x^k)$. If yes, recover.*

(ii) *Test indicators $I_i, i \in \{1,\ldots,n\} - \mathcal{B}(x^k)$ using the identification criterion*

$$I_i \leq \delta. \tag{7.1}$$

(iii) *Use information from (ii) to update the estimate of $\mathcal{B}(x^k)$.*

For step (ii) we use two indicators, the sum of the Tapia indicators

$$T_i^k = \left|\frac{x_i^{k+1}}{x_i^k}\right| + \left|1 - \frac{y_i^{k+1}}{y_i^k}\right|,$$

and the primal-dual indicator

$$PD_i^k = \frac{x_i^{k+1}}{y_i^{k+1}},$$

where $x_i^{k+1} = x_i^k + \Delta x_i^k$ and $y_i^{k+1} = y_i^k + \Delta y_i^k$. For step (iii), we adopted the strategy proposed by Tapia [23]. This strategy was implemented by Vardi [29] in the context of nonlinear programming problems. The idea is to divide the set $\{1,\ldots,n\}$ at iteration k into three categories, $\mathcal{B}(x^k)$ of indices corresponding to variables that are predicted to be zero, $\mathcal{NB}(x^k)$ of indices corresponding to variables that are predicted to be nonzero, and a third category $\mathcal{U}(x^k) = \{1,\ldots,n\} - (\mathcal{B}(x^k) \cup \mathcal{NB}(x^k))$ consisting of indices corresponding to variables that we feel we do not have enough information to decide to move them to one of the first two categories. There are several ways to specify each of the three categories and the rule to move a variable from one category to another. In our implementation we start with $\mathcal{NB}(x^0) = \mathcal{B}(x^0) = \emptyset$ and $\mathcal{U}(x^0) = \{1,\ldots,n\}$. An index i is moved from $\mathcal{U}(x^k)$ to $\mathcal{B}(x^k)$ if the identification criterion (7.1) is satisfied for the two indicators T_i and PD_i for two consequent iterations. If (7.1) is not satisfied for at least one of the two indicators for more than one iteration then i is moved from $\mathcal{U}(x^k)$ to $\mathcal{NB}(x^k)$. An index $i \in \mathcal{NB}(x^k)$ moves to $\mathcal{U}(x^k)$ if the two indicators satisfy (7.1) at the same iteration. This procedure was tested on a subset of the NETLIB set of LP test problems. The results were satisfying.

Without any recovery technique the procedure predicted the wrong set of zero variables in only 12 problems out of the 83 problems tested. If a simple recovery technique was used the procedure failed only on one problem, NESM, where the procedure predicted only some members of $\mathcal{B}(x^*)$. The performance of our procedure, for some problems from the *netlib*, is given in Table 3. The first column of this table gives the names of the problem solved while columns 2 to 10 give the number of positive variables at the corresponding iteration after the zero variables have been identified and removed. Here M is the total number of iterations required to solve the problem to an accuracy of 10^{-8}. We stress that for the results in Table 3, the procedure was activated when the relative gap was smaller than 10^{-1}. However, some experiments were conducted using the procedure from the first iteration and the results were also promising, although in this case we had to be more conservative in choosing the thresholds δ_{Tapia} and $\delta_{primal-dual}$. The total numbers of iterations required by

Table 3: Identifying zero variables for some test problems

PROBLEM	M−8	M−7	M−6	M−5	M−4	M−3	M−2	M−1	M
FINNIS	985	545	503	482	469	407	394	390	390
PILOT4	1181	899	840	840	756	667	667	611	607
SCRS8	1275	1204	1141	1096	553	379	329	325	325
DEGEN2	595	595	595	555	543	436	70	53	53
GFRD-PNC	1149	1149	1149	1011	872	842	426	407	407
CYCLE	2139	2139	2139	1865	1655	149	1360	1265	1260
MAROS	1906	1906	1906	1832	1806	634	493	480	476
WOOD1P	2395	2288	2181	2151	1767	1418	52	39	39
SHIP12L	5533	5533	5528	5528	5306	5306	3590	726	726
GREENBEA	5283	5283	5051	1790	1639	1543	1429	1384	1375

the ten listed problems are as follows: for FINNIS M=27, PILOT4 M=35, SCRS8 M=26, DEGEN2 M=14, GFRD-PCN M=18, CYCLE M= 26, MAROS M=24, WOOD1P M=14, SHIP12L M= 17 and GREENBEA M=44.

The following remarks are of interest

- If the procedure is used very early in the iterative process, the algorithm may converge to a point on the relative boundary of the solution set, i.e. a solution with more

zero variables than the ones in the relative interior of that set. An example of this phenomenon is problem SCSD6. When the identification procedure was activated when the relative gap was less than 10^{-1}, the final approximate solution had 1168 zero variables. When we used the identification procedure from the first iteration the final approximate solution had 1198 zeros.

- We noticed that identifying and removing zero variables early may actually reduce the total number of iterations required. Some examples are given in Table 4.

Table 4: Saving iterations by removing zero variables

PROBLEM	Total number of iterations without removing zero variables	Total number of iterations if zero variables are removed
CYCLE	26	23
SCSD6	12	10
GREENBEA	45	44
WOOD1P	15	14

8 Concluding Remarks

The main results of this study are Theorem 2.1, Proposition 5.1 and Proposition 6.2. An immediate implication of Theorem 2.1 is that the detrimental behavior that results from using the variables as indicators cannot be avoided. Proposition 5.1 pinpoints the source of the undesirable behavior that the primal-dual indicator exhibits in interior-point methods. Proposition 6.2 states that the Tapia indicator retains its useful properties when used with primal-dual interior-point methods, under the same conditions that guarantee fast local convergence of the duality gap sequence. We emphasize that convergence of the iterate sequence is not required for this result. Our numerical results are interesting and present a solid case against the use of the variables as indicators, and motivate the use of the Tapia indicators over the so-called primal-dual indicator. In conclusion, we strongly believe that the use of indicators can be an extremely useful and powerful tool and deserves further study.

Acknowledgement

The authors acknowledge valuable discussions with Bob Bixby, John Dennis, Irv Lustig and Yinyu Ye. We also acknowledge a very useful discussion with Florian Potra on the material in Section 2.

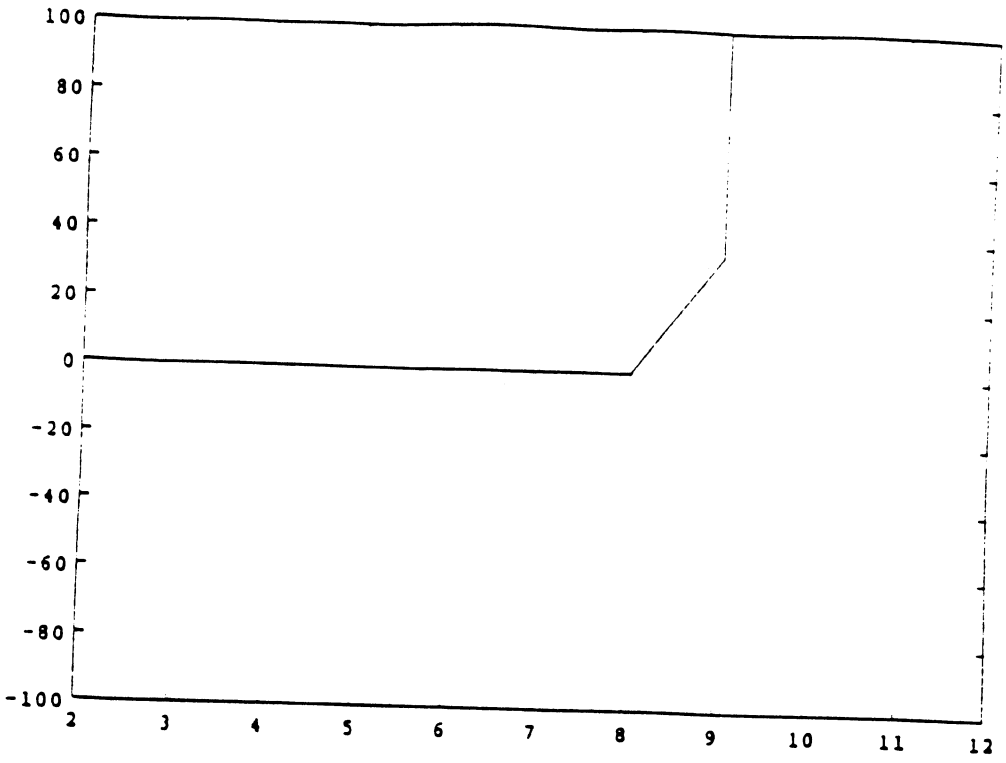

Figure 1: The primal-dual indicator for a nonzero variable.

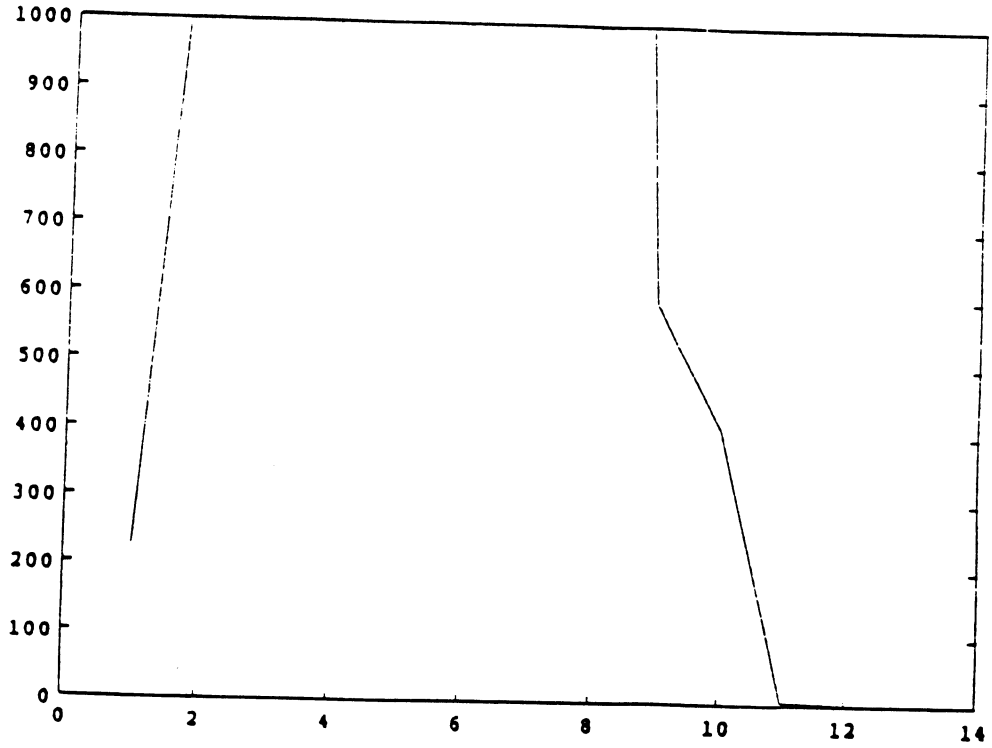

Figure 2: The primal-dual indicator for a zero variable.

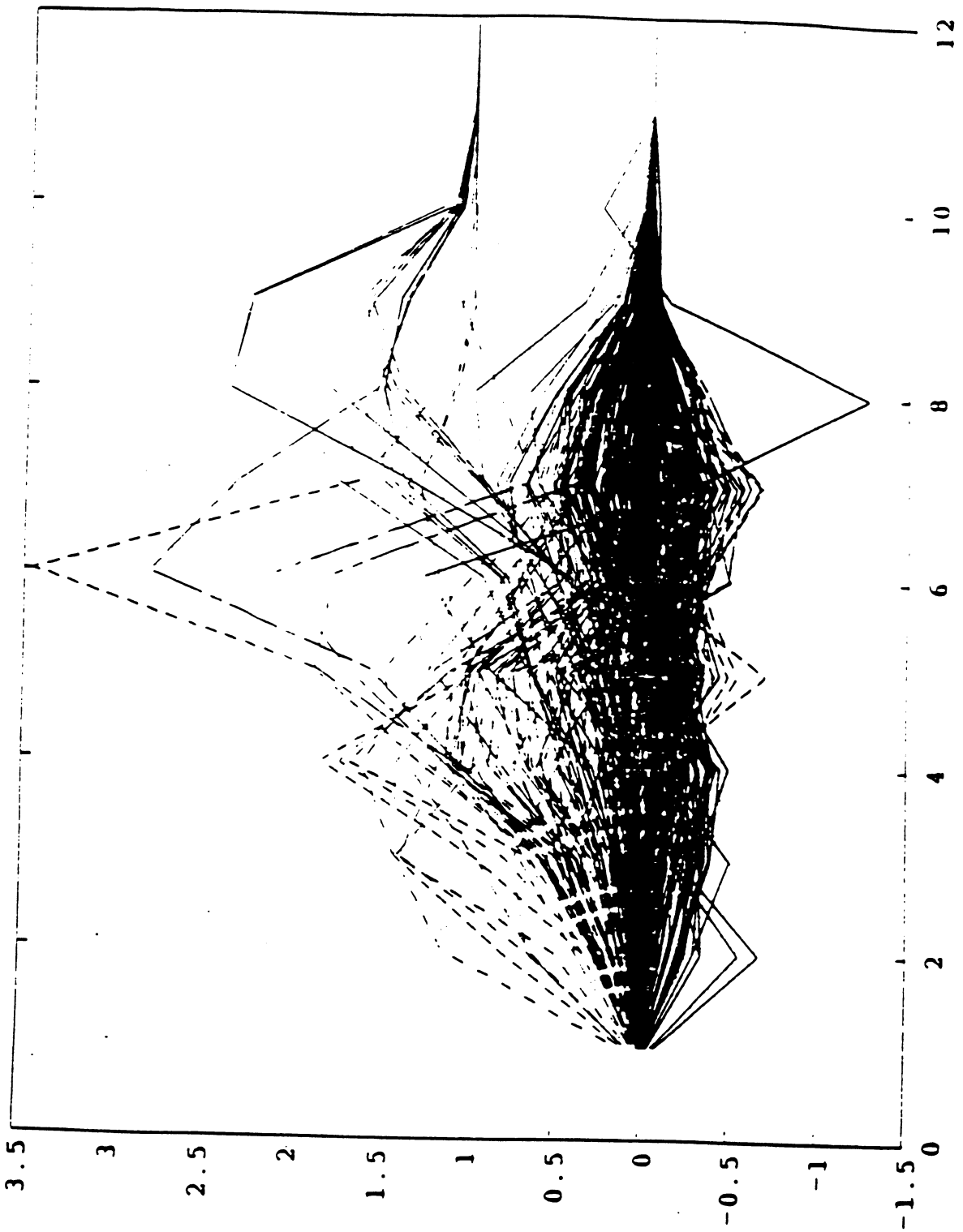

Figure 3: The Tapia indicators for all variables in problem SCSD1.

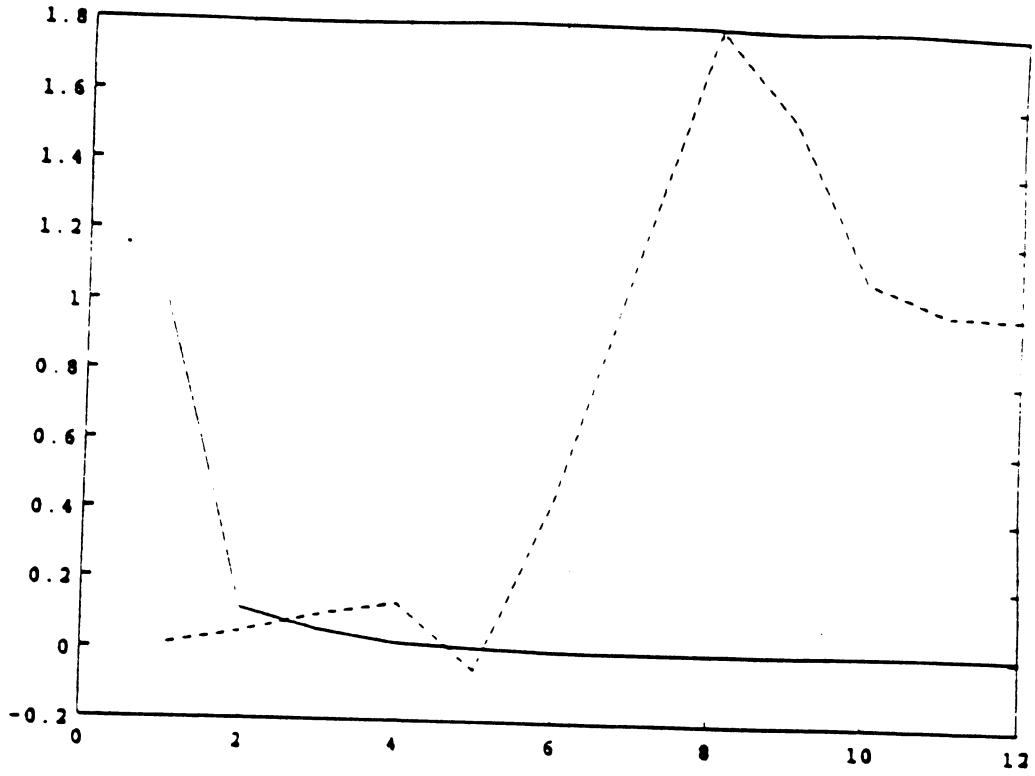

Figure 4: The Tapia indicator for a positive variable with value 10^{-4} at the solution.

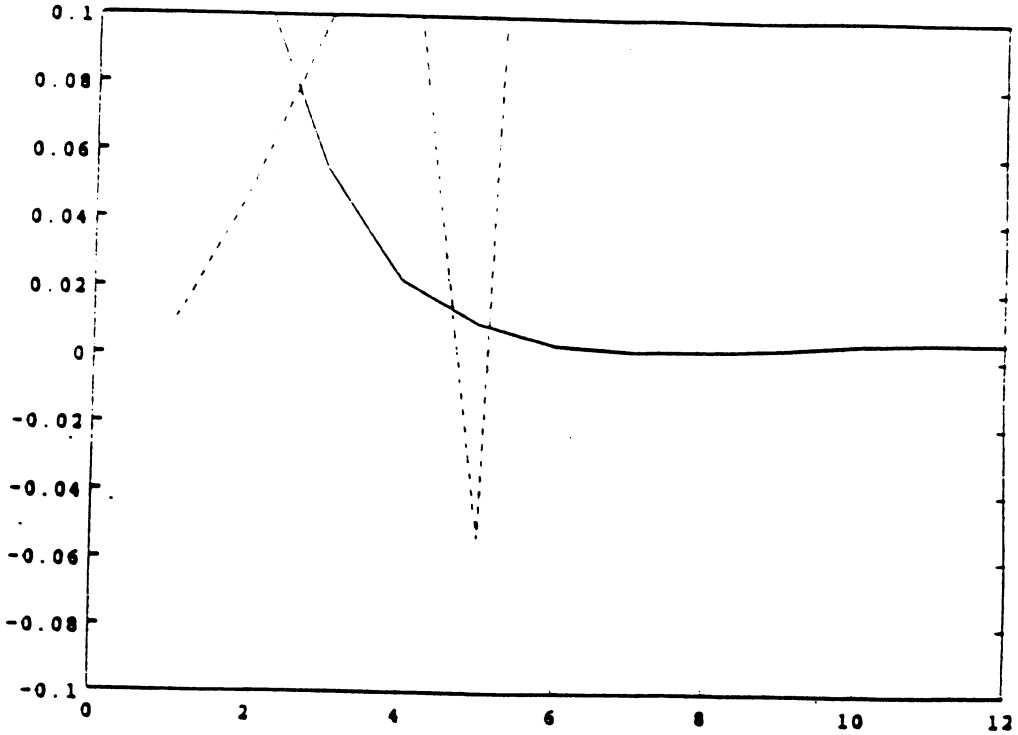

Figure 5: The primal-dual indicator for a positive variable.

References

[1] K. M. Anstreicher. Dual ellipsoids and degeneracy in the projective algorithm for linear programming. In J. C. Lagarias and M. J. Todd, editors, *Mathematical developments arising from linear programming*, pages 141–149. American Mathematical Society, 1990.

[2] P.T. Boggs, P.D. Domich, J.R. Donaldson, and C. Witzgall. Algorithmic enhancements to the method of centers for linear programming problems. *ORSA J. Computing*, 1(3):159–171, 1989.

[3] A. Charnes, W. W. Cooper, and R. M. Thrall. A structure for classifying and characterizing efficiency and inefficiency in data envelopment analysis, 1990. Jesse H. Jones Graduate School of Administration, Rice University. Working Paper 80.

[4] D. B. Dantzig and Y. Ye. A build-up interior method for linear programming: Affine scaling form. Technical Report SOL 90-4, Department of Operations Research, Stanford University, 1990.

[5] A. S. El-Bakry. *On the role of indicators in identifying zero variables in linear programming.* PhD thesis, Department of Mathematical Sciences. Rice University, 1991.

[6] D. M. Gay. Early stopping heuristics that compute an optimal solution for interior-point linear programming algorithms. Presented at the third SIAM conference on optimization, Boston, April 3-5, 1989. AT & T Bell Lab, Murray Hill.

[7] P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin, and Wright M.H. On the projected Newton barrier method for linear programming and equivalence to Karmarkar's projective method. *Math. Prog.*, 36:183–209, 1986.

[8] O. Guler. Existence of interior points and interior paths in nonlinear monotone complementarity problems, 1990. The college of Business Administration, The University of Iowa, Iowa City. Working Paper.

[9] O. Guler and Y. Ye. Convergence behavior of some interior-point algorithms, 1991. Dept. of Management Sciences, The University of Iowa. Working Paper.

[10] M. J. Jansen and S. H. Tijs. Robustness and nondegenerateness for the linear complementarity problem. *Mathematical Programming*, 37:293–308, 1987.

[11] N. Karmarkar and K. Ramakrishnan. Implementation and computational results of the Karmarkar algorithm for linear programming, using an iterative method for computing projections. Technical report, AT&T Bell Laboratories, Murray Hill, NJ, 1988.

[12] M. Kojima. Determining basic variables of optimal solutions in Karmarkar's new LP algorithm. *Algorithmica*, 1:499–515, 1986.

[13] V.V. Kovacevic-Vujcic. Improving the rate of convergence of interior-point methods for linear programming, 1989. Manuscript, Belgrade University, Belgrade, Yugoslavia.

[14] I.J. Lustig, R.E. Marsten, and D.F. Shanno. Computational experience with a primal-dual interior point method for linear programming. Technical Report SOR 89-17, Dept. Civil Eng. and O.R., Princeton University, 1989.

[15] Irvin J. Lustig. An implementation of a strongly polynomial time algorithm for basis recovery. Dept. Civil Eng. and O.R., Princeton University, in preparation.

[16] L. McLinden. An analogue of Moreau's proximation theorem, with application to the nonlinear complementarity problem. *Pacific J. of Math.*, 88:101–161, 1980.

[17] K.A. McShane, C.L. Monma, and D.F. Shanno. An implementation of a primal-dual interior point method for linear programming. *ORSA J. Computing*, 1:70–83, 1989.

[18] N. Megiddo. On finding primal and dual optimal bases. Technical Report RJ 6328 (61997), IBM Almaden Research Center (San Jose, CA), 1988.

[19] S. Mehrotra. Implementations of affine scaling methods: towards faster implementations with complete cholesky factor in use. Technical Report 89-15, Dept. Industral Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, 1989.

[20] S. Mehrotra. On finding a vertex solution using interior point methods. *Linear Algebra and Its Applications*, 152:11111, 1991.

[21] M. G. C. Resende and G. Veiga. A dual affine scaling algorithm for minimum cost network flow, 1990. Manuscript.

[22] R. T. Rockafellar. *Convex analysis.* Princeton University Press, Princeton, NJ, 1970.

[23] R. A. Tapia. On the role of slack variables in quasi-Newton methods for constrained optimization. In L. C. W. Dixon and G. P. Szegö, editors, *Numerical Optimization of dynamic systems*, pages 235–246. North-Holland, 1980.

[24] R. A. Tapia and Yin Zhang. An optimal-basis identification technique for interior-point linear programming methods. *Linear Algebra and Its Applications*, 152:343–363, 1991.

[25] R.A. Tapia, Y. Zhang, and Y. Ye. Superlinear convergence of primal-dual interior-point methods in linear programming revisited. Technical Report TR91-27, Department of Mathematical Sciences, Rice University, Houston, TX. 77251-1892, 1991.

[26] M.J. Todd. Improved bounds and containing ellipsoid in Karmarkar's linear programming algorithm. *Math. of Operations Research*, 13:650–659, 1988.

[27] M.J. Todd and Y. Ye. Containing and shrinking ellipsoids in the path-following algorithm. *Math. Prog.*, 47:1–9, 1990.

[28] K. Tone. An active-set strategy in interior point method for linear programming. Technical Report 90-B-4, Graduate School of Policy Science, Saitama University, Japan, 1991.

[29] A. Vardi. A new minimax algorithm, 1988. Manuscript, Dept. of Mathematics and Computer Science, Drexel University, Philadelphia, PA 19104, to appear in JOTA.

[30] Yinyu Ye. Recovering optimal basis in Karmarkar's polynomial algorithm for linear programming, 1987. Manuscript, Dept. Engineering-Economic Systems, Stanford University.

[31] Yinyu Ye. Interior point algorithms for quadratic programming. Technical report, Dept. of Management Sciences, The University of Iowa, 1989. Working Paper Series No. 89-29.

[32] Yinyu Ye. On the finite convergence of interior-point algorithms for linear programming. Technical report, Dept. of Management Sciences, The University of Iowa, 1991. Working Paper.

[33] Y. Zhang, R. A. Tapia, and J. E. Dennis. On the superlinear and quadratic convergence of primal-dual interior-point linear programming algorithms. Technical Report TR90-6, Dept. Mathematical Sciences, Rice University, Houston, TX. 77251-1892, 1990.

A Existing Indicators

In this appendix we catalogue various indicators that appear in the literature.

A.1 The Kojima Indicator

Kojima [12] proposed a method to be used in Karmarkar-type methods for identifying positive variables. The Kojima indicator is defined as the piecewise linear function:

$$I_i(x^k, \rho) = \min\{c^p_j + \frac{c^T x^k}{n} + (P_{i,j} + \frac{1}{n})\rho : j \neq i\}, \quad i = 1, \ldots, n, \tag{1.1}$$

where P is a specific matrix of the form

$$P = I - (AD)^T M - (1/n)ee^T,$$

where $e = (1, \ldots, 1)^T$, M is a specific $m \times n$ matrix, x^k is a strictly feasible point, $c^p = PDc$ where $D = diag(x^k)$, and ρ is a parameter. Kojima proved that $I_i(x^k, \rho) > 0$ is a sufficient condition for x_i^* to be positive for any solution with nonpositive objective function. On the other hand, it is not clear that the Kojima indicator satisfies the sharp or the uniform separation properties. It also depends on an auxiliary parameter ρ and it is not obvious how it should be chosen. It was noted by Kojima that this indicator requires primal nondegeneracy. The test proposed by Kojima to identify zero variables costs $O(mn)$ arithmetic operations for each variable.

A.2 The Ye-Todd Indicators

Todd [26] proved that all dual optimal solutions are contained in ellipsoids that can be generated as a by-product of the Karmarkar algorithm. Using this information, he proposed an indicator to that can identify a subset of primal variables that are zero at every primal optimal solution. Ye [30] rigorously studied this idea and proposed a closely related indicator in a closed form. The Ye-Todd indicator is

$$I(y^k, \epsilon^k) = y^k - \epsilon^k \, (D^k)^{-1} q^k \tag{1.2}$$

where D^k a positive diagonal matrix with d^k as its diagonal (usually $d^k = x^k$), q^k is the diagonal of the projection matrix $D^k A^T (A(D^k)^2 A^T)^{-1} A D^k$ and $\epsilon^k \geq 0$ is a parameter of the dual-slack ellipsoid

$$\{y : \|D^k(y - y^k)\| \leq \epsilon^k\}$$

that contains all the optimal dual slacks y^*. The parameter ϵ^k is set equal to the duality gap at each iteration. Ye proved that $I(y^k, \epsilon^k) > 0$ implies that $x_i^* = 0$ in all optimal solutions of that problem. The Ye-Todd indicator has interesting theoretical properties. Unfortunately, it is not clear that it satisfies the uniform and sharp separation properties. Also it is expensive to compute and finally, it requires primal nondegeneracy. Anstreicher [1] proposed a modification to the Ye-Todd approach to extend it to problems with primal degeneracy.

Using a similar approach, Ye and Todd [27] described a path-following algorithm for convex quadratic programming problems which uses a sequence of ellipsoids. Each of these ellipsoids contains all of the primal and dual-slack solutions. They propose an indicator, using these ellipsoids, to identify zero variables in the course of an interior-point algorithm for linear programming. Although their indicator has very nice theoretical properties, it is expensive to compute at each iteration. It also requires nondegeneracy if it is desired that all zero variables be identified.

A.3 The Kovacevic-Vujcic Indicator

In an attempt to accelerate the convergence of Karmarkar-type methods, Kovacevic-Vujcic [13] introduced the following indicator

$$I(x^k, \Delta x^k) = x^k + \alpha^k (x^{k+1} - x^k), \tag{1.3}$$

where

$$\alpha^k = \min_{x_i^{k+1} - x_i^k \leq 0} \frac{-x_i^k}{x_i^k - x_i^{k+1}}.$$

Kovacevic-Vujcic proved that this indicator is superlinearly faster than the variables, namely

$$\lim_{k \to \infty} \frac{\|I(x^k, \Delta x^k) - I^*\|}{\|x^k - x^*\|} = 0,$$

where

$$I^* = \lim_{k \to \infty} I(x^k, \Delta x^k).$$

On the other hand, this indicator satisfies neither the sharp nor the uniform separation property. It also requires dual nondegeneracy. Finally we note that while the Kovacevic-Vujcic indicator may be of use in Karmarkar-type methods; it is not clear that it is of use in the context of the recent primal-dual interior-point methods. The reason is that in the more effective implementation of these methods the step is taken to the boundary asymptotically,

see Zhang, Tapia and Dennis [33]. This means that the indicator, asymptotically, will coincide with the iterate.

A.4 The Mehrotra Indicator

Mehrotra [19] consider the quantity

$$I_i(y^k, \Delta y^k, \alpha^k) = \frac{|y_i^{k+1} - y_i^k|}{y_i^k}, \tag{1.4}$$

which measures the relative change in the dual slack y_i^k, as an indicator. Mehrotra used this indicator to drop constraints when implementing a dual affine scaling method.

A.5 The Tapia-Zhang Indicator

In an attempt to uncover an optimal basis of the linear program (1.1), Tapia and Zhang [24] introduced the indicator:

$$I(d^k) = diag[D^k A^T (A(D^k)^2 A^T)^{-1} A D^k] \tag{1.5}$$

where D^k is the diagonal matrix with d^k as its diagonal. The vector d^k can be x^k, y^k or $(Y^k)^{-1}X^k e$. Assuming primal and dual nondegeneracy Tapia and Zhang proved that this indicator satisfies both the sharp and the uniform separation properties, namely

$$\lim_{k\to\infty} I_i(z^k) = \begin{cases} 0 & \text{if } i \in \mathcal{B}(x^*) \\ 1 & \text{if } i \notin \mathcal{B}(x^*) \end{cases} \tag{1.6}$$

They also proved that it is quadratically faster than the variables, i.e.

$$\|I(z^k) - I^*\| \leq O(\|x^k - x^*\|^2)$$

where

$$I^* = \lim_{k\to\infty} I(z^k)$$

This indicator can be used for primal, dual, and primal-dual methods. For some interesting theoretical properties of this indicator see [24]. The disadvantages of this indicator are that it requires primal and dual nondegeneracy and is expensive if evaluated at each iteration.

A.6 The Resende-Veiga Indicator

Resende and Veiga [21] used the reciprocal of the dual slacks as indicators, namely

$$I(y^k) = (Y^k)^{-2}e, \tag{1.7}$$

where $Y = diag(y)$. This indicator satisfies

$$\lim_{k\to\infty} I_i(y^k) = \begin{cases} 1/y_i^{*2} & \text{if } i \in \mathcal{B}(x^*) \\ \infty & \text{if } i \notin \mathcal{B}(x^*) \end{cases}$$

They used the identification criterion

$$I_i(y^k) < 10^{-3}\ \epsilon_d \implies x_i^k = 0$$

where ϵ_d is the geometric mean of the arithmetic and harmonic means of $(1/y_1^2, \ldots, 1/y_n^2)$. The use of this indicator requires strict complementarity. Unfortunately, this indicator satisfies neither the sharp nor the uniform separation property.