

Logarithmic Indicators and the
Identification of Subgroups of Variables
in Interior-Point Methods

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

September, 1993

TR93-35

Logarithmic Indicators and the Identification of Subgroups of Variables in Interior-Point Methods ¹

A. S. El-Bakry², R. A. Tapia³ and Y. Zhang⁴

Abstract

The identification of certain groups of variables in optimization problems is an important issue and can be used to computational advantage. In this paper new logarithmic indicators are introduced. It is demonstrated that the logarithmic Tapia indicators have superior ability for identifying several subgroups of variables in the context of primal-dual interior-point methods.

Keywords: Interior-point methods, Logarithmic Tapia indicator function, Identifying subgroups of variables.

Abbreviated Title: The Logarithmic Tapia Indicators.

AMS(MOS) subject classifications: 65K,49M,90C

¹Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt and visiting member of the Center for Research on Parallel Computations, Rice University, Houston, Texas, 77521-1892(elbakry@rice.edu). This author was supported in part by NSF Coop. Agr. No. CCR-8809615 and AFOSR 89-0363 and the REDI Foundation

²Department of Computational and Applied Mathematics and the Center for Research on Parallel Computations, Rice University, Houston, Texas, 77521-1892(rat@rice.edu). 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

³Department 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(y Zhang@math.umbc.edu). 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 concerns the identification of several subgroups of variables in interior-point methods for linear programming. We consider the linear programming problem in the standard form

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

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 dual problem of (1.1) can be stated as

$$\begin{aligned} & \text{minimize} && b^T y \\ & \text{subject to} && A^T \lambda + y = c \\ & && y \geq 0, \end{aligned} \tag{1.2}$$

where $\lambda \in \mathbf{R}^m$, and $y \in \mathbf{R}^m$ are the Lagrange multiplier vectors corresponding to the equality and inequality constraints of problem (1.1) respectively. The multipliers λ are also known as the dual variables and y as the dual slack variables. The first-order optimality (or Karush-Kuhn-Tucker) conditions for problem (1.1) are:

$$F(x, y, \lambda) \equiv \begin{pmatrix} Ax - b \\ A^T \lambda + y - c \\ XYe \end{pmatrix} = 0 \tag{1.3}$$

and

$$(x, y) \geq 0 \tag{1.4}$$

where $X = \text{diag}(x)$, $Y = \text{diag}(y)$ 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 interior-point methods can be viewed as damped and perturbed Newton's method applied to the nonlinear system of equations (1.3). For more details, see Kojima, Mizuno, and Yoshise [14] and Zhang, Tapia, and Dennis [33]. The algorithmic framework for such methods is the following

Algorithm 1.1 (Primal-Dual Interior-Point Method)

Given a starting point (x^0, y^0, λ^0) with $(x^0, y^0) > 0$. For $k = 0, 1, \dots$, 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 $(\Delta x^k, \Delta y^k, \Delta \lambda^k)$:

$$F'(x^k, y^k, \lambda^k)(\Delta x, \Delta y, \Delta \lambda) = -F(x^k, y^k, \lambda^k) + \mu(x^k, y^k)\hat{e} \tag{1.5}$$

3. Choose $\tau^k \in (0, 1)$ and set the steplength $\alpha^k = \min(1, \tau^k \hat{\alpha}^k)$ where

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

4. Form the new iterate

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

We emphasize that the algorithmic parameters σ^k and τ^k are under the control of the algorithm designer with the restriction that $0 < (\sigma^k, \tau^k) < 1$. The parameter σ^k is sometimes called the *centering parameter*. Note that the choice of steplength α^k in Step 3 of Algorithm 1 guarantees $(x^{k+1}, y^{k+1}) > 0$. In Step 2 $\hat{e} = (0, \dots, 0, 1, \dots, 1)^T$, with $n + m$ zero components. We also emphasize that the iteration sequence generated by Algorithm 1 is not necessarily feasible.

We use the notation \mathcal{F} and \mathcal{S} to denote the feasible set and the solution set, respectively, of problem (1.3). In many papers on interior-point methods, the assumption that \mathcal{F} contains strictly feasible points is made. This assumption ensures that \mathcal{S} is nonempty and compact. For reasons that will become clear in Section 4 we will make the assumption that \mathcal{S} is bounded only as needed.

We use the following notation

- $\mathcal{N} = \{1, 2, \dots, n\}$.
- $ri(U)$ and $\partial_r U$ denote the relative interior and the relative boundary of a given set U .
- \mathcal{Z} denotes the set of indices of variables that are zero in every solution of problem (1.1).
- $\bar{\mathcal{Z}} = \mathcal{N} - \mathcal{Z}$ denotes the set of indices of variables that are positive in the relative interior of the solution set of problem (1.1).

It is known that $\mathcal{Z} \cup \bar{\mathcal{Z}} = \mathcal{N}$. It is also known that if we consider the dual problem (1.2), then \mathcal{Z} and $\bar{\mathcal{Z}}$ will be the sets of indices of the dual slacks y_i that are positive and zero, respectively, in the relative interior of the solution set of that problem.

The identification of subgroups of variables in primal-dual interior-point algorithms can be used in several ways to computational advantage. Such applications include, reducing the dimension of the problem, and obtaining a good initial basis when crossing over to a simplex-type method to determine a basic solution. It can also play a role in obtaining highly accurate solutions when combined with the projection method proposed by Ye [31], perturbation methods proposed by Mehrotra [21] or successive projection methods used by El-Bakry [4]. Finally this information can be an important ingredient in building preconditioners for solving linear systems arising in interior-point methods, see Gill, Murray, Ponceleon, and Saunders [8].

Many authors have contributed to the goal of identifying subgroups of variables by introducing indicator functions in interior-point methods. Gill, Murray, Saunders, Tomlin, and Wright [9], Karmarkar and Ramakrishnan [12], McShane, Monma and Shanno [19], Tone [28], Lustig, Marsten, and Shanno [17], Dantzig and Ye [3], and Boggs, Domich, Donaldson and Witzgall [1], among others, proposed the use of variables, either primal or dual, to predict members of \mathcal{Z} . Tapia [24] introduced two indicators in the context of identifying active constraints in nonlinear constrained optimization problems. Kojima [13] proposed an indicator for use in Karmarkar-type algorithms. Ye [30] and Todd [27] introduced two indicators for Karmarkar-type and primal-dual algorithms. Choi and Goldfarb [2] proposed two indicators similar to the Todd-Ye indicators with the advantage that their indicators can be used in algorithms that are not necessarily interior-point methods. Tapia and Zhang [25] proposed an indicator that can be used in primal, dual, or primal-dual interior-point methods. Kovacevic-Vujcic [15] introduced an indicator that is superlinearly faster than the variables in Karmarkar-type methods. The ratio of primal variables and dual slacks was used as an indicator by several researchers including Gay [7], Ye [31], and Lustig [16]. Mehrotra [20] used an indicator based on the relative change in the dual slack variables. Resende and Veiga [23] used the reciprocal of the dual slack variables as indicators. Recently de Vreede [29] proposed ????? For a thorough study of indicator functions in interior-point methods we refer the reader to the recent paper by El-Bakry, Tapia, and Zhang [5]. de Vreede [29] performed numerical comparisons between several indicators. In all these papers, the focus was on the identification of the two basic subgroups of variables mentioned above, namely variables that are zero everywhere in the solution set and variables that are positive in the relative interior of that set. The question as to whether further subgroups of variables can be identified is the theme of this paper.

In El-Bakry [4] and EL-Bakry, Tapia, and Zhang [5], an indicator function $I(x^k, y^k)$ is said to satisfy both the strict and uniform separation properties if

$$\lim_{k \rightarrow \infty} I(x^k, y^k) = \begin{cases} \theta_1 & \text{if } i \in \mathcal{Z} \\ \theta_2 & \text{if } i \notin \mathcal{Z} \end{cases}, \quad (1.6)$$

where $\theta_1 \ll \theta_2$. For brevity we will say that an indicator function satisfies *the $\theta_1 - \theta_2$ separation property* if it satisfies (1.6).

This paper is organized as follows; in Section 2, the Tapia indicators are investigated when the sequence of *centering* parameter $\{\sigma^k\}$ is bounded away from zero. In Section 3 we introduce the logarithmic indicator functions and establish their separation properties. Section 4 is devoted to numerical experiments with a subset of the Netlib [6] set of linear programming test problems. Interpretation of our numerical experimentation is attempted in Section 5. The logarithmic indicators for the dual variables is considered in Section 6. Final conclusions are given in Section 7. Throughout this paper we consider the logarithm to be the natural logarithm.

2 Three Indicators

In this section we list three indicators for identifying zero and nonzero variables in interior-point methods for linear programming. The reasons for choosing these particular indicators are that they are not expensive to compute and that they do not require nondegeneracy or (linear) feasibility assumptions. These indicators are the variables used as indicator, the primal-dual indicator, and the Tapia indicators.

The variables used as indicator is probably the most widely used indicator in constrained optimization. This indicator can be defined for both primal and dual variables

$$V_P(x^k) = x^k \text{ and } V_D(y^k) = y^k.$$

Another common indicator in the area of primal-dual interior-point method is the primal-dual indicator

$$PD(x^k, y^k) = \frac{x^k}{y^k}.$$

Tapia [24] proposed the following two indicators

$$T_P(x^k) = \frac{x^{k+1}}{x^k} \text{ and } T_D(y^k) = \frac{y^{k+1}}{y^k}.$$

The variables as indicator do not satisfy the uniform separation property (1.6). On the other hand both the primal-dual and the Tapia indicators satisfy this property with

- $\theta_1 = 0$ and $\theta_2 = +\infty$ for the primal-dual indicator.
- $\theta_1 = 0$ and $\theta_2 = 1$ for the Tapia indicators.

This 0-1 separation property of the Tapia indicators can be obtained when the sequence of centering parameters $\{\sigma^k\}$ is chosen to converge to zero, see El-Bakry, Tapia, and Zhang [5]. An example that shows the behavior of the primal Tapia indicator for ADLITTLE, one of the Netlib set of test problems, is given in Figure 2. Observe that the trajectories of the Tapia indicators of several variables converging to zero have similar behavior at iteration 9. Others have different behavior at iteration 10. However, it is not clear that this behavior has any special significance, although it may suggest that certain collections of variables, either primal or dual, have similar behavior.

2.1 The Behavior of the Tapia Indicators when $\sigma^k \rightarrow \sigma > 0$

The behavior of the Tapia indicators when $\sigma^k \rightarrow \sigma > 0$ has not been studied so far. One reason for studying such behavior is that in order to converge to the analytic center of the solution set \mathcal{S} the current theory requires that the iteration sequence $\{(x^k, y^k, \lambda^k)\}$ converges and that σ^k be bounded away from zero, see Zhang and Tapia [32]. Another reason is that some effective implementation

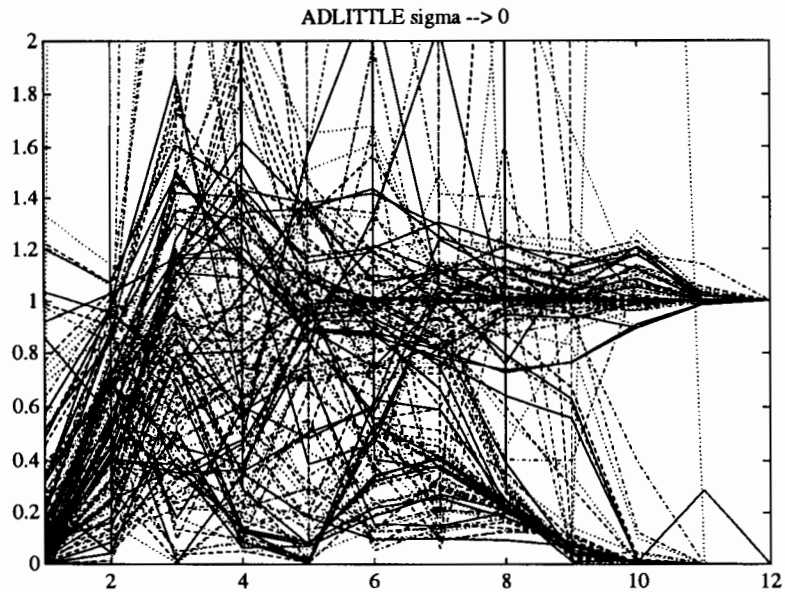


Figure 1: The Tapia primal indicator for all variables in ADLITTLE.

of interior-point methods, such as OB1 [18], locally use a positive, yet very small, constant value for σ^k .

In the case that $\{\sigma^k\}$ is chosen to be bounded away from zero, El-Bakry, Tapia and Zhang [5] pointed out that, with the choice $\alpha^k = \min(1, \tau^k \bar{\alpha}^k)$, the 0-1 separation property of the Tapia indicator cannot be retained even if the iteration sequence converges.

In an attempt to study the behavior of the Tapia indicators in that case we performed numerical experimentation with the Netlib set of linear programming test problems. Details of the numerical experiments are given in Section 4. The results were interesting yet their interpretation was quite challenging and led to some important questions concerning the behavior of the iteration sequence generated by Algorithm 1. They also led to the introduction of new indicators that have a powerful ability to identify several groups of variables.

We first start by showing an example of the behavior of the Tapia indicators for problem ADLITTLE when $\sigma^k = 0.1$. Observe that the convergence of the Tapia indicators corresponding to variables that are positive in the relative interior of the solution set is somewhat blurred??, see Figure 2.1. Observe also that the dual Tapia indicators do not exhibit such behavior, see Figure 2.1. In other words the Tapia indicators, for variables that are positive in the relative interior of the solution set, do not seem to converge to 1. This observation is more evident in problem AGG. In this problem the primal logarithmic Tapia indicator seem actually to deviate from 1 in the last two iteration as seen in Figure 2.1. The behavior can be quite ?? as seen in Figure 2.1. The dual Tapia

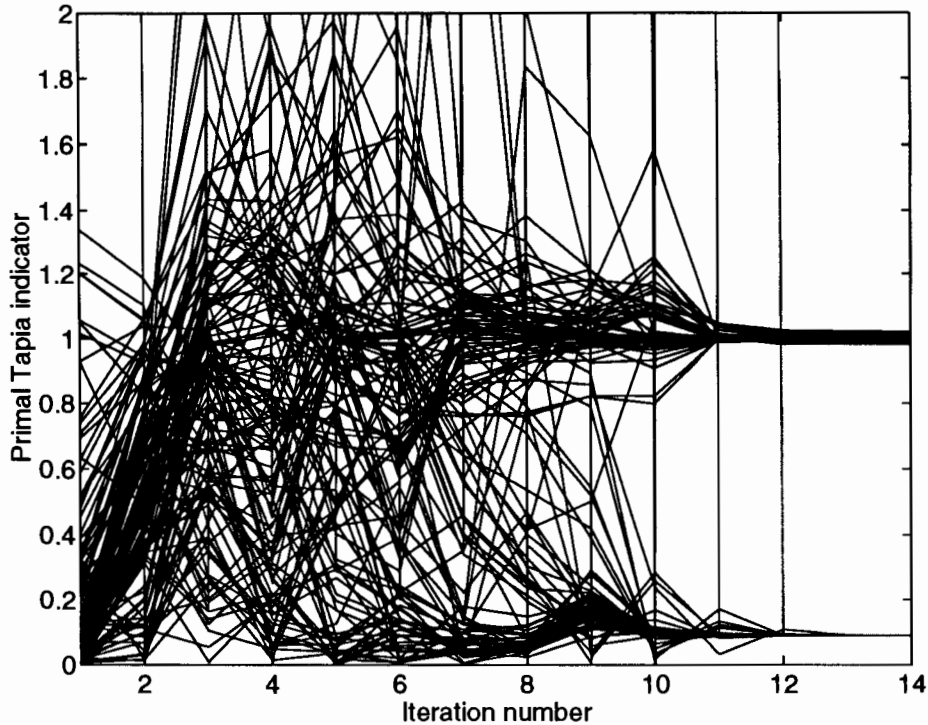


Figure 2: The primal Tapia indicator for all variables in ADLITTLE with $\sigma^k = 0.1$.

indicators corresponding to a subset of positive variables seem to approach values different than 1. We point out that in the case of problem AGG, the problem was solved with tighter stopping criterion than that used in ADLITTLE. We will come back to this point in Section 4. We emphasize that these phenomena appeared in many problems in the Netlib set.

As a consequence of these observations it was evident that studying these phenomena requires different tools than the existing indicators provide. This concern was the motivation for introducing the *logarithmic indicators*.

3 The Logarithmic Indicators

In this section we introduce several logarithmic indicators constructed from the three existing indicators discussed in Section 3. These indicators are

- The *logarithmic primal-dual indicator*

$$LOGPD(x_i^k, y_i^k) = \log\left(\frac{x_i^k}{y_i^k}\right), \quad i = 1, \dots, n,$$

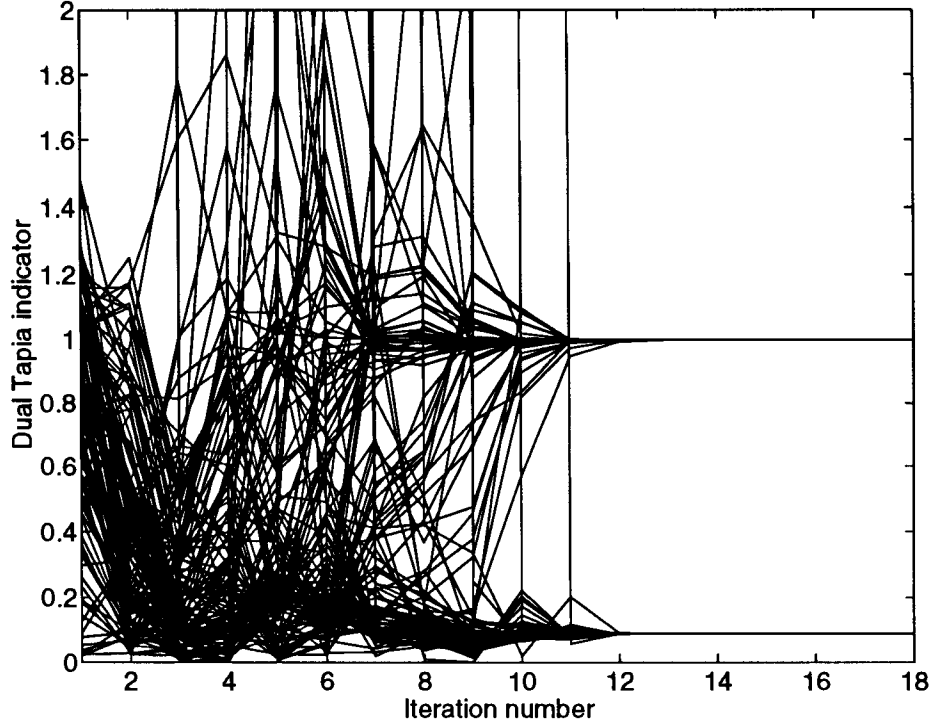


Figure 3: The dual Tapia indicator for all variables in ADLITTLE with $\sigma^k = 0.1$.

- The *logarithmic variable indicators*

$$LOGV_P(x_i^k) = \log(x_i^k),$$

and

$$LOGV_D(x_i^k) = \log(y_i^k).$$

- The *logarithmic Tapia indicators*

$$LOGT_P(x_i^k) = \log | 1 - T_P(x_i^k) |,$$

and

$$LOGT_D(y_i^k) = \log | 1 - T_D(x_i^k) |.$$

The separation properties for these indicators can be easily derived for the case when $\sigma^k \rightarrow 0$ at least R-linearly. For $LOGV_P$ we have

$$\lim_{k \rightarrow \infty} LOGV_P(x_i^k) \rightarrow \begin{cases} -\infty & \text{if } i \in \mathcal{Z} \\ \log(x_i^*) & \text{if } i \notin \mathcal{Z} \end{cases}$$

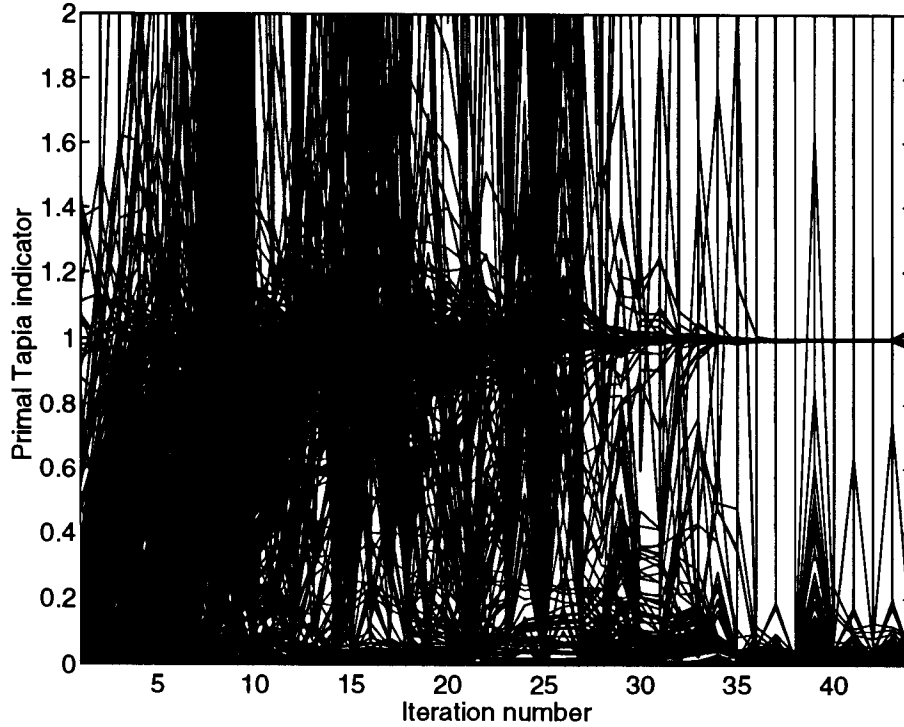


Figure 4: The primal Tapia indicator for all variables in AGG with $\sigma^k = 0.1$.

where x_i^* is the limit point of $\{x^k\}$ which is guaranteed to converge by Theorem ?? of Tapia, Zhang, and Ye [26]. For *LOGPD* we have

$$\lim_{k \rightarrow \infty} \text{LOGPD}(x_i^k, y_i^k) \longrightarrow \begin{cases} -\infty & \text{if } i \in \mathcal{Z} \\ +\infty & \text{if } i \notin \mathcal{Z} \end{cases}$$

Here we used the convention $\log(+\infty) = +\infty$. For the logarithmic Tapia indicators we have

$$\lim_{k \rightarrow \infty} \text{LOGT}_P(x_i^k) \longrightarrow \begin{cases} 0 & \text{if } i \in \mathcal{Z} \\ -\infty & \text{if } i \notin \mathcal{Z} \end{cases}$$

The separation property for the logarithmic primal-dual indicator is the same if the sequence of centering parameters is bounded away from zero. It is not clear, however, what would be the separation property for the logarithmic Tapia indicators, if there is any, in that case. Extensive numerical experimentation is performed in the next section to study the logarithmic Tapia indicators when $\sigma^k \geq \sigma > 0$.

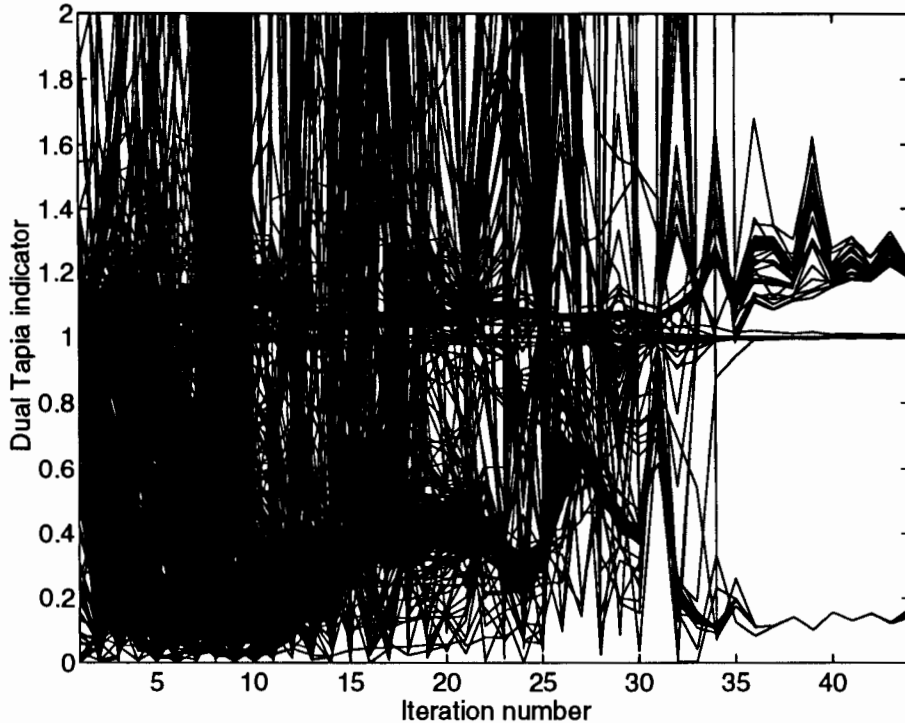


Figure 5: The primal Tapia indicator for all variables in AGG with $\sigma^k = 0.1$.

4 Numerical Experiments

In this section we present several numerical experiments with the logarithmic indicators introduced in Section 3. 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 duality gap $c^T x - b^T y$ to zero. For numerical purposes our stopping criterion is stated in terms of the *relative gap* $\frac{c^T x - b^T y}{1 + |b^T y|}$, rather than the gap. 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

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

In the following experiments all problems are solved to an accuracy of 10^{-8} unless otherwise specified. This choice agrees with the default choice for the stopping criterion in many interior-point codes. The experiments were performed on a Sun 4/490 workstation with 64 Megabytes of memory.

4.1 The Logarithmic Tapia Indicators

The behavior of the logarithmic Tapia indicators was quite surprising. Figure 4.1 shows the behavior of the dual logarithmic Tapia indicator when $\sigma^k = 0.1$ for all k for problem LOTFI. The

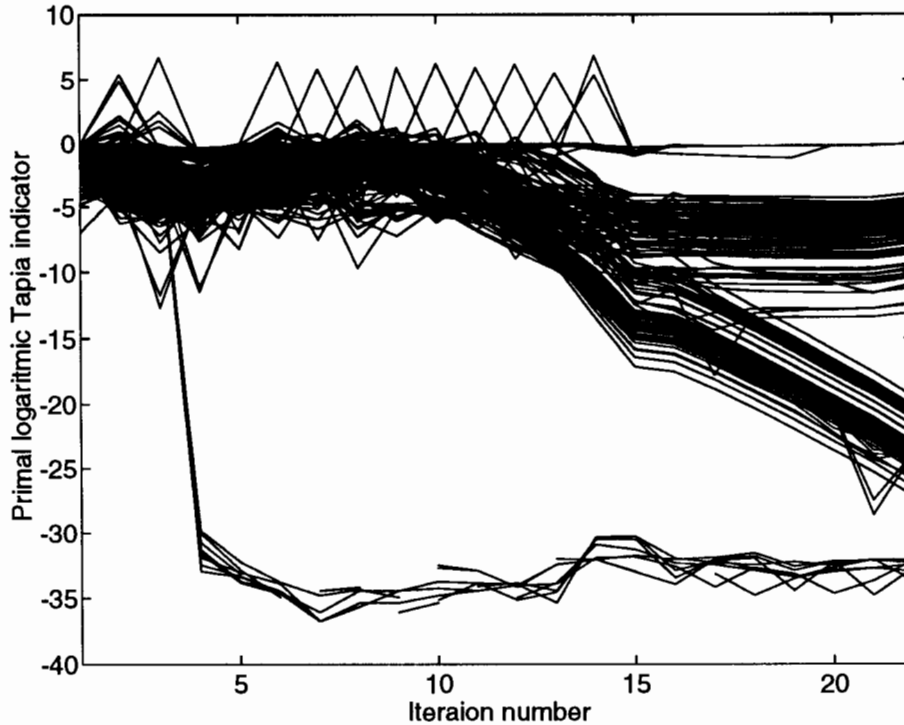


Figure 1: The dual logarithmic Tapia indicator for all variables in LOTFI with $\sigma^k \rightarrow 0.1$.

trajectories of the dual logarithmic Tapia indicators split into four groups instead of the two basic groups (variables that are zero everywhere on the solution set and variables that are positive in the relative interior of the solution set). So it seems that the logarithmic Tapia indicators are able to identify several subgroups of variables instead of only the two basic groups.

In the following we try to explain this behavior.

4.1.1 Variables Invariant on the Solution Set

El-Bakry, Tapia and Zhang [5] pointed out that the convergence of the Tapia indicators does not require the convergence of the iteration sequence, but only the milder condition that the sequence $\{(\Delta x^k, \Delta y^k)\}$ converges to zero. For Algorithm 1, the current theory (see Tapia, Zhang, and Ye [26]) ensures that this will occur when $\sigma^k \rightarrow 0$. In this case the primal Tapia indicator corresponding to variables that are positive in the relative interior of the solution set converges to 1 (even if the

iteration sequence itself does not converge). This in turn implies that the logarithmic Tapia indicator corresponding to such variables approaches infinity in the limit. Observe that the logarithmic Tapia indicators corresponding to some variables seem to approach infinity in Figure 4.1. Several of these indicators do so very early (at iteration 4) while others approach infinity only when the iterates are close to the solution set. On the other hand observe that there is a third group that does not seem to approach infinity at all. To check this phenomenon in other problems we ran the code with $\sigma^k = 0.1$ on problem AFIRO. The same phenomenon occurred again, see Figure 4.1.1. We see

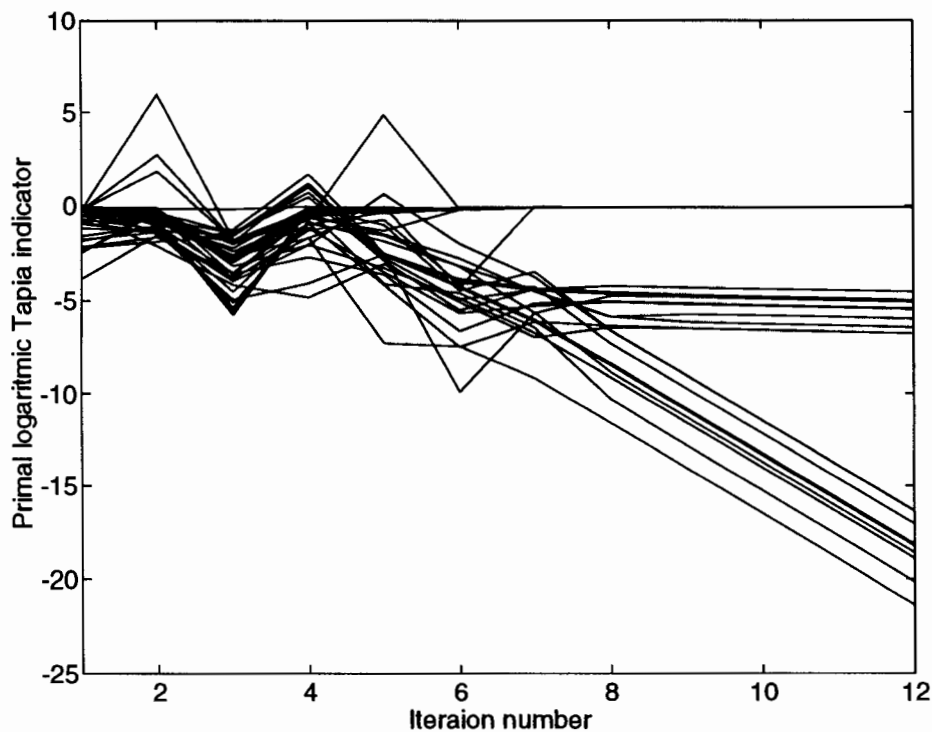


Figure 2: The logarithmic Tapia primal indicator for all variables in AFIRO with $\sigma^k = 0.1$.

that there is a group of variables with their corresponding logarithmic Tapia indicator approaching some intermediate value.

Solving the same problem (AFIRO) with $\sigma^k = O((x^k)^T y^k)$ and also by CPLEX []. Comparing the three approximate solutions we observed the following, see Table (4.1.1)

- (1) Variables with the corresponding primal logarithmic Tapia indicator approaching values in the range $(-7, -4)$ are different in both solutions.
- (2) Variables with the corresponding primal logarithmic Tapia indicator approaching $-\infty$ have

the same value (to at least eight significant digits) in the three approximate solutions. Moreover those variables are *basic* in the CPLEX solution.

Variable Name	\hat{x}_j	$\hat{\hat{x}}_j$	$\hat{x}_j - \hat{\hat{x}}_j$	Status
X01	7.99999999e+01	8.00000000e+01	-9.9999994e-08	basic
X02	2.54999999e+01	2.55000000e+01	-1.0000000e-07	basic
X03	5.45000000e+01	5.45000000e+01	0	basic
X04	8.47999999e+01	8.48000000e+01	-9.9999994e-08	basic
X06	6.70725826e+01	7.04887640e+01	-3.4161813e+00	basic
X07	1.82142856e+01	1.82142857e+01	-1.0000000e-07	basic
X15	4.88582971e+01	5.22744784e+01	-3.4161812e+00	nonbasic
X16	7.10969376e+01	7.47180899e+01	-3.6211522e+00	basic
X17	1.29274174e+01	9.51123596e+00	3.41618143e+00	basic
X22	5.00000000e+02	5.00000000e+02	0	basic
X23	4.75919999e+02	4.75920000e+02	-9.9999999e-07	basic
X24	2.40800000e+01	2.40800000e+01	0	basic
X26	2.15000000e+02	2.15000000e+02	0	basic
X28	1.39125795e+02	2.71341050e+02	-1.3221525e+02	nonbasic
X36	3.39942857e+02	3.39942857e+02	0	basic
X37	2.44817061e+02	1.12601807e+02	1.32215254e+02	basic
X38	5.98240921e+01	1.16676651e+02	-5.6852558e+01	basic
X40	3.60874205e+02	2.28658950e+02	1.32215255e+02	basic
X47	3.36935854e+01	2.26983040e+01	1.09952814e+01	basic
X49	2.24628214e+02	9.13846893e+01	1.33243524e+02	basic
X50	1.02000001e+01	1.02000000e+01	1.00000001e-07	basic
X51	1.69078970e+02	1.08605259e+02	6.04737109e+01	basic

Table 1: All positive variables in the approximate solution for problem AFIRO

This suggests that the logarithmic Tapia indicators have the ability to distinguish between positive variables that are invariant throughout the solution set and those whose values change from solution to solution.

4.1.2 Variables Invariant on the Feasible Set

We now direct our attention back to Figure 4.1 where the dual logarithmic Tapia indicator is plotted against the iteration number. Again the previously discussed subgroups of variables in the three categories of variables that are zero throughout the solution set, positive variables that are invariant on the solution set, and positive variables that have different values in different solutions are clearly

shown.

A new subgroup of variables manifests itself with the property that they reach their terminal values very early in the course of the algorithm. This is reflected in the behavior of the corresponding dual logarithmic Tapia indicator approaching $-\infty$ from iteration 4 and settling down there. The same phenomenon occurred again for SHARE1B, but at a relatively later stage (at iteration 12), see Figure 4.1.2. We observed that at iteration 12 the relative primal feasibility residual became

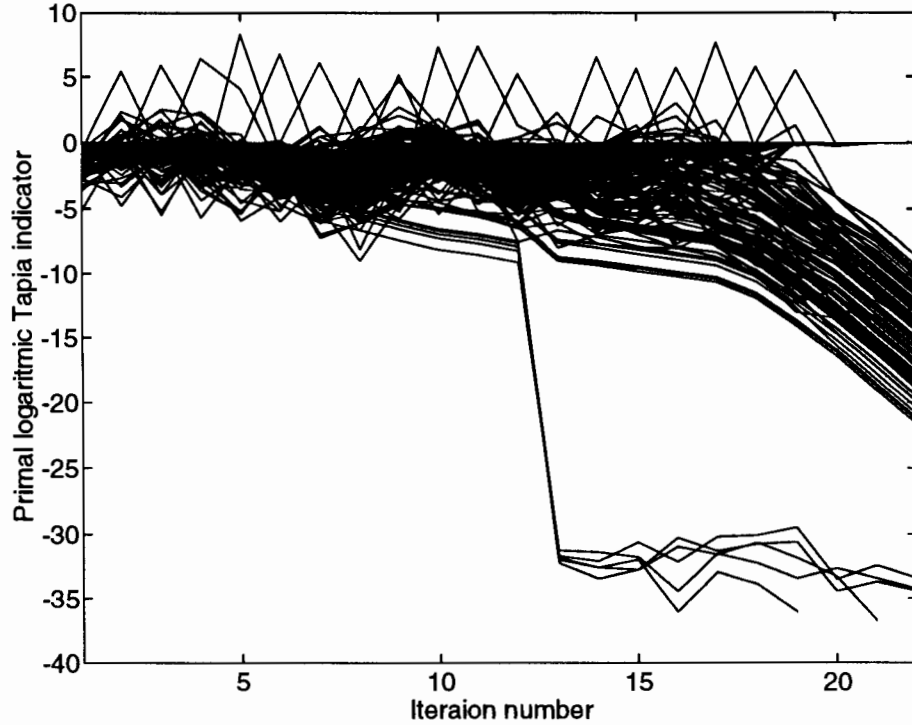


Figure 3: The logarithmic Tapia dual indicator for all variables in SHARE1B with $\sigma^k \rightarrow 0.1$.

2.0×10^{-12} after being 1.0×10^{-2} at iteration 11. So a plausible interpretation for the behavior of the logarithmic Tapia indicator in this case is that the logarithmic Tapia indicator is able to identify positive variables that are invariant (have the same value) throughout the primal feasibility region $\{x : Ax - b = 0, x \geq 0\}$ when the iterate x^k is sufficiently close to $\{x : Ax - b = 0, x \geq 0\}$. It is also interesting to observe that dual feasibility is not needed for such identification (relative dual feasibility for SHARE1B was of order 10^{-3} at iterations 11, 12, and 13).

4.1.3 Variables Unbounded on the Solution Set

Solving problem ADLITTLE with $\sigma^k = 0.1$, we observed that the code terminated with relative feasibility of order 10^{-8} while the dual feasibility $\|A^T \hat{\lambda} + \hat{y} - c\|$, where $(\hat{x}, \hat{\lambda}, \hat{y})$ is the approximate solution given by the code, was 15.6, which is not small. One explanation is that $(\hat{x}, \hat{\lambda}, \hat{y})$ is not close enough to the solution set. To check if this is the case, we resolved the problem to accuracies 10^{-10} and 10^{-13} asking for increasingly more accurate approximate solutions. The results are shown in Table (4.1.3). In all cases $\|A^T \hat{\lambda} + \hat{y} - c\|$ is large although the solution seems more accurate (which

ϵ_{exit}	relative gap	relative primal feasibility	relative dual feasibility	$\ A^T \lambda + y - c\ $
10^{-8}	1.4D-09	6.1D-15	3.1D-18	15.6
10^{-10}	1.1D-11	6.0D-15	1.0D-30	15.6
10^{-13}	9.1D-14	2.0D-15	3.2D-47	15.6

is reflected by the decrease in the relative gap) as we reduced ϵ_{exit} .

Another explanation is that some of the dual variables are unbounded on the solution set at the approximate solution given by the code. Both primal and dual logarithmic Tapia indicators are computed and plotted in Figures 4.1.3 and 4.1.3 respectively. First we note that the three groups of variables discussed earlier for AFIRO appear in Figure 4.1.3. However observe that in Figure 4.1.3, one of the trajectories seems to approach $+\infty$. This suggests that the corresponding dual slack is also approaching $+\infty$, and hence unbounded in the solution set. The value of this variable at \hat{x} , the approximate solution given by the code, is 2.06×10^{19} .

To investigate the validity of such a claim, we solved the same problem with smaller values of σ , decreasing σ an order of magnitude each time. For $\sigma = 0.01, 0.001$, we obtain similar behavior. For smaller values of σ the dual logarithmic Tapia indicator, corresponding to the same variable, does not seem to approach infinity although it still has a very peculiar behavior (see Figure 4.1.3) for $\sigma = 10^{-5}$. Finally, it is interesting to know that if we solve the problem with $\sigma^k = O((x^k)^T y^k)$, the dual logarithmic Tapia indicator behavior becomes similar to any other variable, see Figure 4.1.3. We attempt to explain this behavior in Section 5.

4.1.4 Yet Another Subgroup of Variables

Another subgroup of variables was discovered in some problems of the Netlib set. When we solved problem AGG, again with $\sigma^k = 0.1$, the behavior of the primal logarithmic Tapia indicator for some variables was peculiar, see Figure 4.1.4. The trajectories for several logarithmic Tapia indicators was oscillating between, seemingly, two different values as the corresponding variable approached its

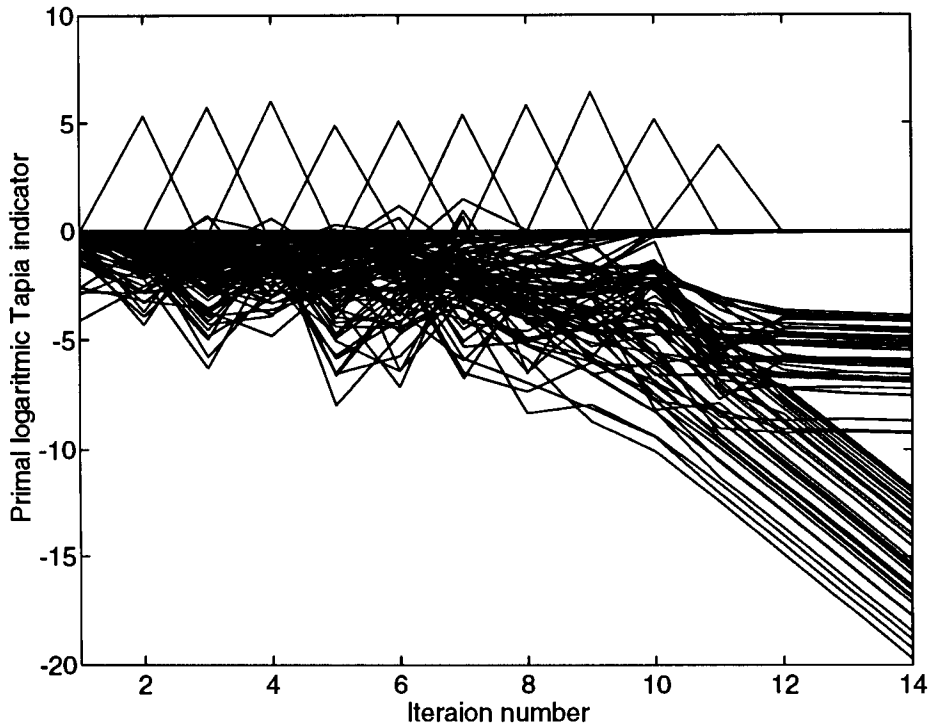


Figure 4: The primal logarithmic Tapia indicator for all variables in ADLITTLE with $\sigma^k = 0.1$.

terminal value. The corresponding primal variable is zero at the approximate solution obtained by the code. The corresponding dual variable has the large value of 10^8 , but its dual logarithmic Tapia indicator does not approach $+\infty$. This behavior was observed in several problems, e.g. SHIP04S and FORPLAN. The interpretation of such behavior is still an open question.

4.2 Other Logarithmic Indicators

In this section we consider the logarithmic variable indicators and the logarithmic primal-dual indicator. Although these indicators have the ability to identify more subgroups of variables, see for example Figures 4.2 and 4.2, than either the primal-dual indicator or the variables as indicators, they cannot compete with the logarithmic Tapia indicators in identifying all the subgroups that the logarithmic Tapia indicator identifies. For example the primal-dual indicator did demonstrate a strong ability to identify members in \mathcal{F}_c^P , see Figure 4.2 for primal-dual indicators for problem SHARE1B and compare to the corresponding Figure 4.1.2 for the primal logarithmic Tapia indicator. The reason is simple. Although variables in \mathcal{F}_c^P approach their terminal values once primal feasibility is reached, the corresponding dual variables, whose terminal values are zero, are not suf-

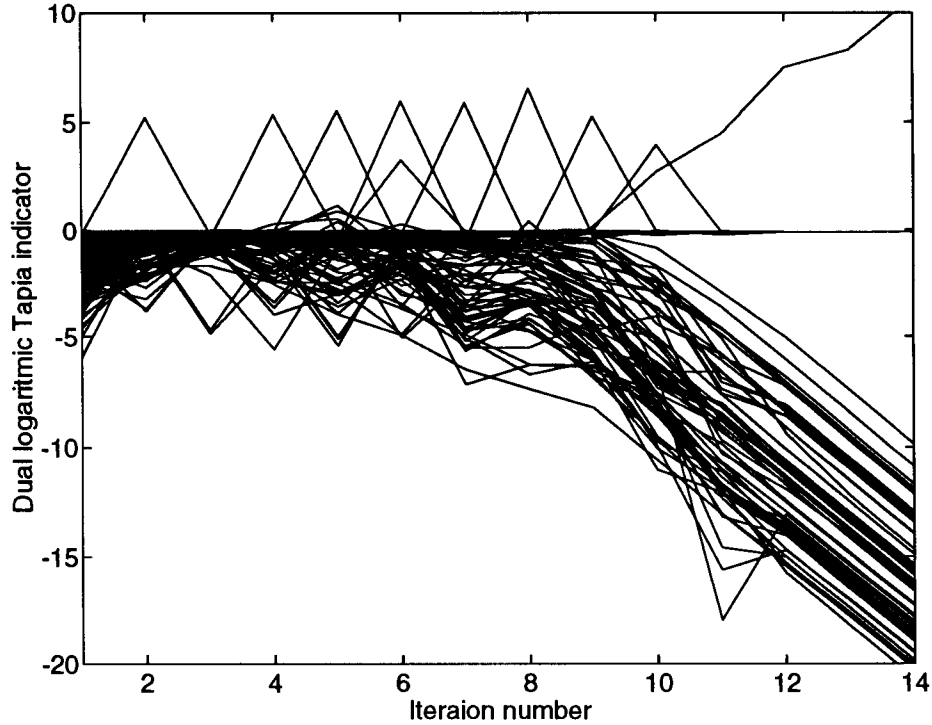


Figure 5: The primal logarithmic Tapia indicator for all variables in ADLITTLE with $\sigma^k = 0.1$.

ficiently close to zero. Moreover both logarithmic primal-dual and logarithmic variable indicators were not able to identify variables in \mathcal{P}_c .

5 Interpretation of the Logarithmic Tapia Indicators Behavior when $\sigma > 0$

Although the surprising ability of the logarithmic Tapia indicator to identify several subgroups of variables is impressive, an attempt to explain thier difference was quite challenging. We first define the following sets

- $\mathcal{F}^P = \{x : Ax = b, x \geq 0\}$ denotes the primal feasible region.
- \mathcal{S}^P denotes the primal solution set.
- $\mathcal{P}_c = \{i : x_i = f_i, \text{ where } f_i \text{ is constant for all } x \in \mathcal{S}^P\}$.
- $\mathcal{P}_{nc} = \{i : x_i = f_i(x) > 0, \text{ where } f_i(x) \text{ is not constant for } x \in \mathcal{S}^P\}$.

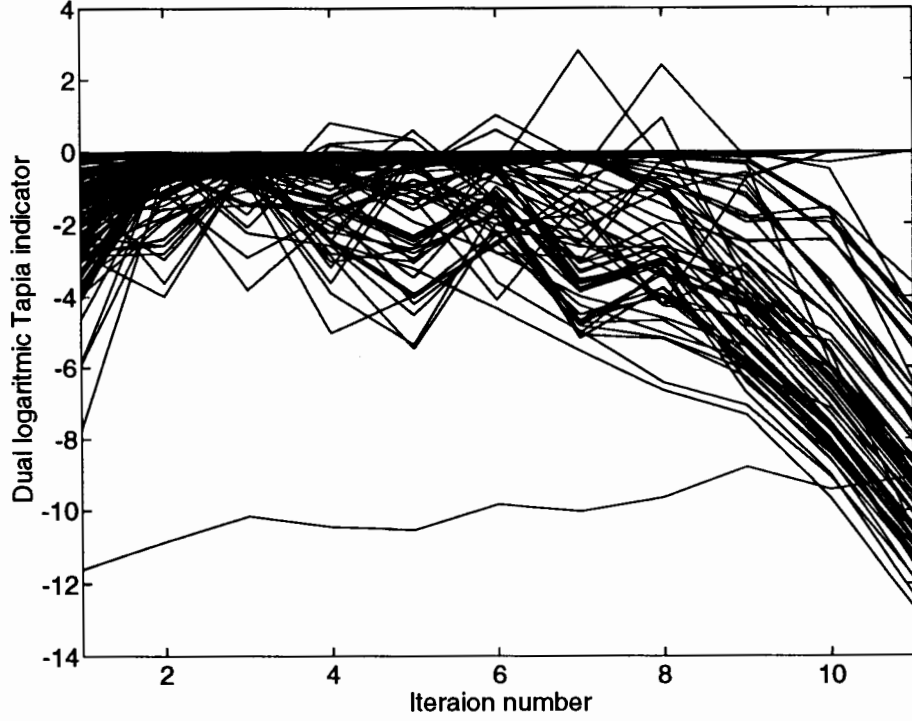


Figure 6: The logarithmic dual Tapia indicator for all variables in ADLITTLE with $\sigma^k = 10^{-5}$.

- $\mathcal{P}_{ri} = \{i : x_i = f_i(x) \geq 0, \text{ where } f_i(x) \text{ is not constant and } f_i(x) = 0 \text{ on } \partial_r \mathcal{S}^P\}$
- $\mathcal{P}_{ub} = \{i : x_i > M \text{ for any positive constant } M \text{ and for at least one } x \in \mathcal{S}^P\}$.
- $\mathcal{F}_c = \{i : x_i = f_i, \text{ where } c_i \text{ is constant for all } x \in \mathcal{F}^P\}$.

We define the corresponding sets for the dual slacks z similarly.

It is easy to establish the separation property for the logarithmic Tapia indicators as follows

Proposition 5.1 *Let the sequence of iterates $\{(x^k, y^k, \lambda^k)\}$ be generated by Algorithm 1. Assume that*

1. \mathcal{S} is bounded.
2. $(x^k)^T y^k \rightarrow 0$.
3. $\frac{\min(X^k Y^k e)}{(x^k)^T y^k} \geq \gamma$ for all k and some positive constant γ .
4. $\sigma^k \rightarrow 0$ and $\tau^k \rightarrow 1$.

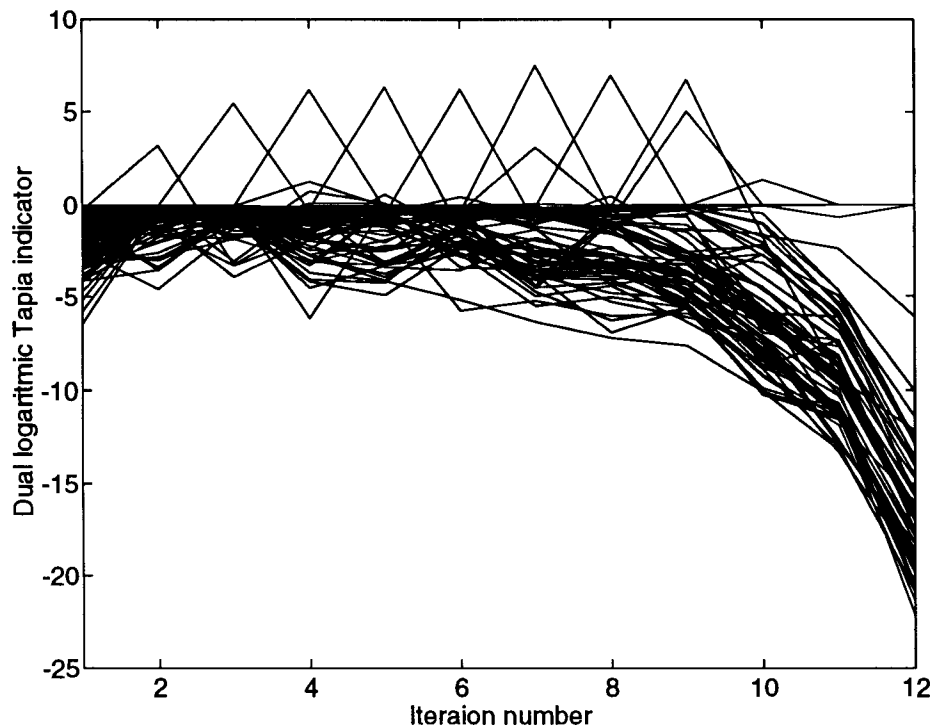


Figure 7: The dual logarithmic Tapia indicator for all variables in ADLITTLE with $\sigma^k \rightarrow 0.1$.

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

$$\lim_{k \rightarrow \infty} LOGT_P(x_i^k) \rightarrow \begin{cases} 0 & \text{if } i \in \mathcal{Z} \\ -\infty & \text{if } i \notin \mathcal{Z} \end{cases}$$

Proof: The proof is straightforward. □

We note here that Conditions 1-4 of Proposition 5.1 ensure the convergence of the step $(\Delta x^k, \Delta y^k)$ to zero, and this is what is needed to ensure the properties of the Tapia (or the logarithmic Tapia) indicators, see El-Bakry, Tapia and Zhang [5]. If condition (iii) is slightly strengthened in the sense that σ^k is required to converge to zero at least R-linearly then these assumption ensures the convergence of the iteration sequence (x^k, y^k) to a solution $(x^*, y^*) \in ri(\mathcal{S})$, see Zhang, Tapia, and Ye [26].

The separation property of the logarithmic Tapia indicator is established in the following proposition.

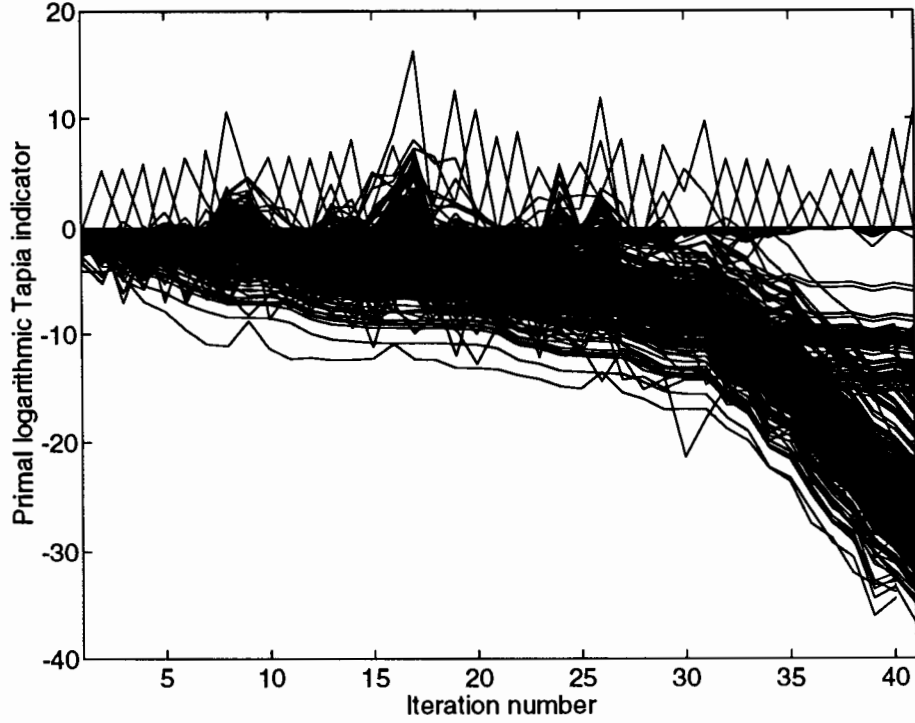


Figure 8: The primal logarithmic Tapia indicator for all variables in AGG with $\sigma^k \rightarrow 0.1$.

5.1 Behavior of the Logarithmic Tapia Indicators when $\sigma^k = \sigma > 0$

First we establish the separation property of the logarithmic Tapia indicator corresponding to the two basic groups of variables.

Proposition 5.2 *Let the sequence of iterates $\{(x^k, y^k, \lambda^k)\}$ be generated by Algorithm 1. Assume that*

1. \mathcal{S} is bounded.
2. $(x^k, y^k) \rightarrow (x^*, y^*)$.
3. $\frac{\min(X^k Y^k e)}{(x^k)^T y^k} \geq \gamma$ for all k and some positive constant γ .
4. $\sigma^k = \sigma > 0$.
5. $\alpha^k \rightarrow \alpha \in (0, 1]$.

Then

$$\lim_{k \rightarrow \infty} LOGT_P(x_i^k) \rightarrow \begin{cases} \eta & \text{if } i \in \mathcal{Z} \\ -\infty & \text{if } i \notin \mathcal{Z} \end{cases}, \quad (5.1)$$

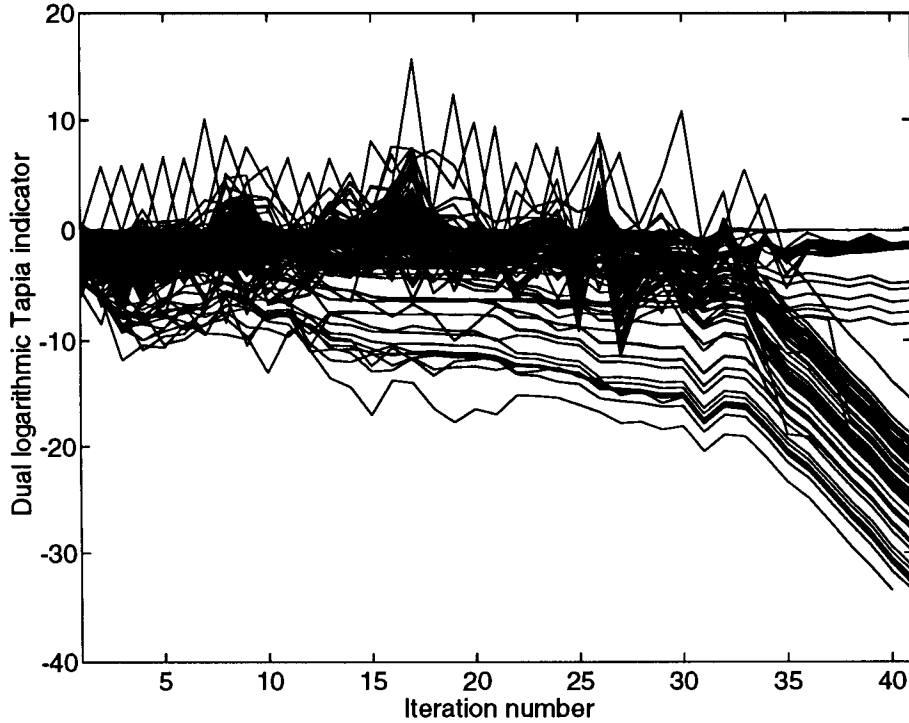


Figure 9: The dual logarithmic Tapia indicator for all variables in AGG with $\sigma^k \rightarrow 0.1$.

where $\eta = \log(1 - \alpha(1 - \sigma))$.

Note that the convergence of the iteration sequence, generated by Algorithm 1, when $\sigma^k \geq \sigma > 0$, has not been established. We also note that it is not clear that $\alpha^k \rightarrow \alpha \in (0, 1)$ under Condition 4 of Proposition 5.2.

5.2 Variables in $\mathcal{Z} \cup \mathcal{P}_c$

We first consider the following proposition that slightly extends conclusion (i) of Theorem (4.1) in Tapia, Zhang and Ye [26]. It basically says that if $\{(x^k, y^k)\}$ is generated by Algorithm 1, then the variables with indices in $\mathcal{Z} \cup \mathcal{P}_c$, i.e. zero variables and variables that are invariant throughout the solution set, converge. The proof is straightforward and will be omitted.

Proposition 5.3 *Let the sequence $\{(x^k, y^k, \lambda^k)\}$ be generated by Algorithm 1. Assume further that*

1. $(x^k)^T y^k \rightarrow 0$.
2. $\min(X^k Y^k e) / (x^k)^T y^k \geq \gamma$ for all k and some positive γ .

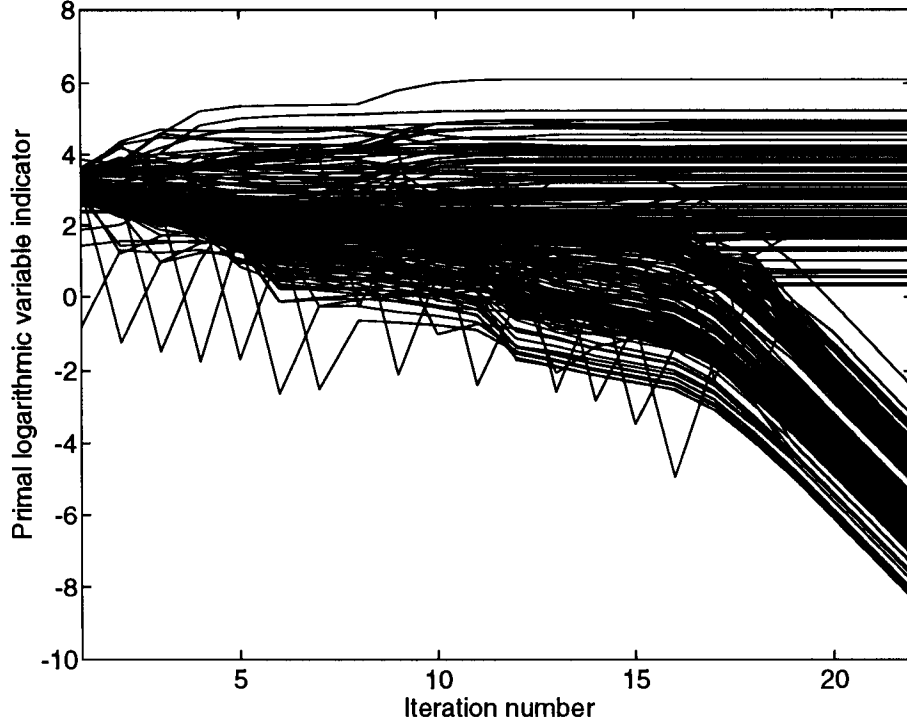


Figure 10: The primal logarithmic variable indicator for all variables in SHARE1B with $\sigma^k \rightarrow 0.1$.

Then for all $i \in \mathcal{Z} \cup \mathcal{P}_c$, the sequence $\{x_i^k\}$ converges to x_i^* .

This proposition explains the behavior of the logarithmic Tapia indicator for variables with $i \in \mathcal{P}_c$. For any of these variables the sequence $\{x_i^k\}$ converges and hence

$$\frac{x_i^{k+1}}{x_i^k} \rightarrow 1, \text{ which implies that } LOGT_P(x_i^k) \rightarrow -\infty. \quad (5.2)$$

For a variable x_j with $j \in \mathcal{Z}$, assuming that the iteration sequence $\{y_j^k\}$ converges to y_j^* , we have

$$LOGT(x_i^k) \rightarrow \log(1 - \alpha(1 - \sigma)). \quad (5.3)$$

Note that we assumed that the dual variable sequence $\{y_j^k\}$ converges. This is guaranteed if y_j^* has a constant value in all dual solutions (a trivial special case is the uniqueness of dual solution). Hence in these two cases the primal logarithmic Tapia indicator behavior is explained. In the following we try to explain their behavior for other subgroups of variables.

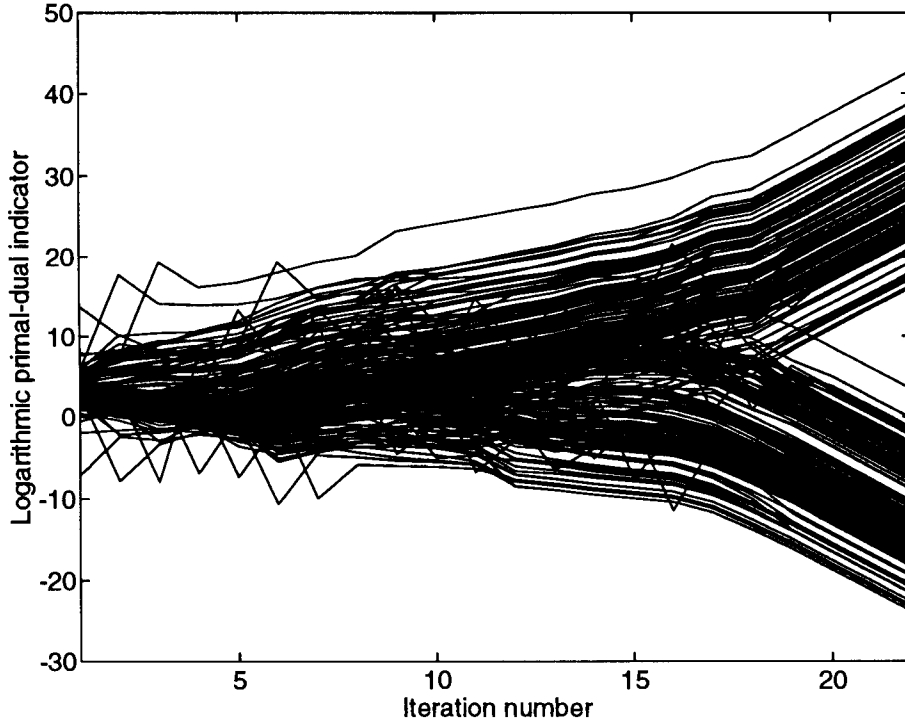


Figure 11: The logarithmic primal-dual indicator for all variables in SHARE1B with $\sigma^k \rightarrow 0.1$.

5.3 Variables in $\mathcal{P} - \mathcal{P}_c$

For variables that have different values at different solutions of problem (1.1), we give two plausible explanations under different assumptions.

First assume that

- (A1) $\sigma^k \geq \sigma > 0$ and $\alpha^k \geq \alpha$ for some $\sigma, \alpha \in (0, 1)$.
- (A2) the iteration sequence $\{(x^k, y^k, \lambda^k)\}$ generated by Algorithm 1 converges to a point (x^*, y^*, λ^*) in the relative interior of the solution set.

By Theorem 2.1 of Zhang and Tapia [32] the limit point is the analytic center of the solution set.

Without loss of generality, consider an arbitrary trajectory of the primal logarithmic Tapia indicator in any of the previous figures for the case $\sigma^k = \sigma = 0.1$. For k sufficiently large this trajectory closely approximates a straight line. Observe that the slope of these straight lines is the same for variables in \mathcal{P}_c and variables in \mathcal{P}_{nc} . For all examples that we have in the previous sections the slope was nonpositive. So assume that the slope for a given curve is $-\xi_i$ where $\xi_i \geq 0$.

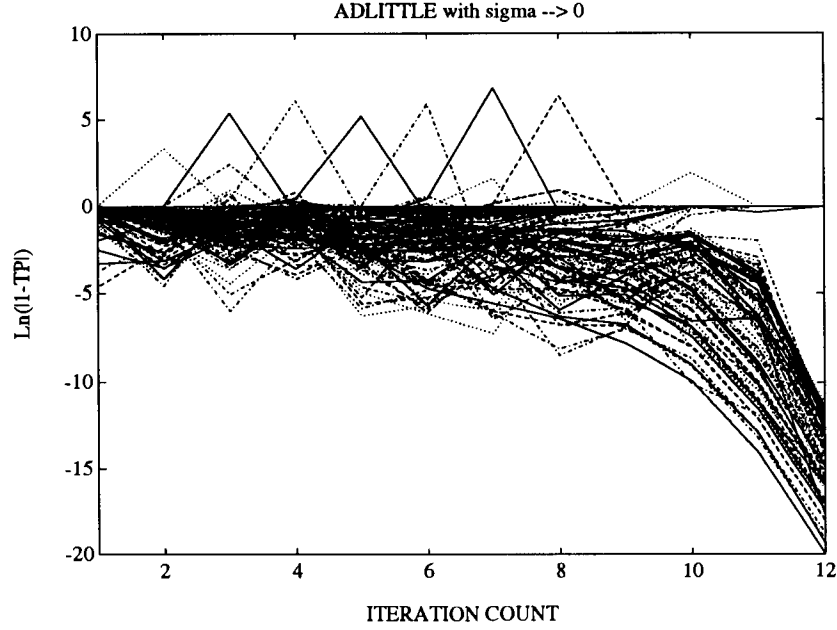


Figure 12: The logarithmic Tapia primal indicator for all variables in ADLITTLE.

Thus

$$\log(\text{LOGT}_P(x_i^{k+1})) - \log(\text{LOGT}_P(x_i^k)) = -\xi_i,$$

hence

$$\log \left| \frac{\alpha^{k+1}}{\alpha^k} \frac{x_i^k}{x_i^{k+1}} \frac{\Delta x_i^{k+1}}{\Delta x_i^k} \right| = -\xi_i.$$

Convergence of the iteration sequence implies

$$\frac{x_i^k}{x_i^{k+1}} \rightarrow 1,$$

and Condition (A1) above implies that $\alpha \leq \alpha^{k+1}/\alpha^k$, hence

$$\log \left| \alpha \frac{\Delta x_i^{k+1}}{\Delta x_i^k} \right| \leq -\xi_i,$$

which implies that

$$\left| \frac{\Delta x_i^{k+1}}{\Delta x_i^k} \right| \leq e^{-\alpha \xi_i}.$$

So although the logarithmic Tapia indicator is not an explicit measure of how fast the components of the iteration sequence converge, the slope of its graph is closely related to the Q_1 factor of the sequence $\{\Delta x_i^k\}$. In light of this interpretation, the behavior of the Tapia indicator for variables in $\mathcal{P} - \mathcal{P}_c$ means that the sequences $\{\Delta x_i^k\}$ for $i \in \mathcal{P} - \mathcal{P}_c$ converge to zero linearly with a Q_1 factor

larger than the Q_1 factor for variables in \mathcal{P}_c . This implies that $\{\Delta x_i^k\}$ for $i \in \mathcal{P} - \mathcal{P}_c$ converges to zero slower than $\{\Delta x_i^k\}$ for $i \in \mathcal{P}_c$.

On the other hand, if the iteration sequence $\{x_i^k\}_{i \in \mathcal{P} - \mathcal{P}_c}$ does not converge then $x_i^{k+1}/x_i^k \not\rightarrow 1$ which results in $LOGT_{\mathcal{P}}(x_i^k) \not\rightarrow -\infty$. So our second interpretation suggests that the iteration sequence generated by Algorithm 1 may not actually converge.

The current convergence theory for the iteration sequence includes Gonzaga and Tapia [10] and [11] and Tapia, Zhang and Ye [26]. Gonzaga and Tapia [10] consider convergence of a modification of the Mizuno-Todd-Ye algorithm which does not fit in the framework considered in this paper. Tapia, Zhang and Ye [26] considered the iteration sequence generated by Algorithm 1 and proved that the sequence converges under the assumption, in addition to other assumptions, that $\{\sigma^k\}$ converges to zero at least R-linearly. Although it is widely believed that the sequence converges for $\sigma^k \geq \sigma > 0$, this has not been proved or disproved.

5.4 Variables in \mathcal{P}_{ub}

If, for some $i \in \mathcal{N}$, we have

$$LOGT_{\mathcal{P}}(x_i^k) \rightarrow +\infty \text{ then } \left| \frac{x_i^{k+1}}{x_i^k} \right| \rightarrow +\infty, \quad (5.4)$$

which indicates that the sequence $\{x_i^k\}$ is unbounded. This in turn, since that sequence is approaching the solution set, indicates that the corresponding variable x_i is unbounded on the solution set \mathcal{S}^P . Note also that (5.4) implies that

$$\alpha^k \left| \frac{\Delta x_i^k}{x_i^k} \right| \rightarrow +\infty \text{ and hence } |\Delta x_i^k| \rightarrow +\infty,$$

since x_i^k is bounded away from zero and $\alpha^k \leq 1$ from Algorithm 1. For an analysis of the limiting behavior of infeasible interior-point algorithms in the presence of unbounded variables, see Mizuno, Todd, and Ye [22].

To explain why this phenomenon is not encountered when σ^k is chosen so that $\sigma^k \rightarrow 0$, we recall the step bounded deterioration property of Tapia, Zhang and Ye [26],

$$|\Delta x_i^k| \leq \beta_1 (x^k)^T y^k + \beta_2 \sigma^k,$$

for some positive constants β_1 and β_2 . So it is clear that, under the assumption that $(x^k)^T y^k \rightarrow 0$, the step Δx_i^k converges to zero as long as $\sigma^k \rightarrow 0$. This demonstrates that if we choose $\sigma^k \rightarrow 0$, the perturbed Newton step is forced to zero, and the volatile behavior of unbounded variables is suppressed, see Figure 5.

5.5 Variables in \mathcal{F}_c

Assume that the sequence $\{x_j^k\}$ generated by Algorithm 1 is bounded. Assume further that the feasible set \mathcal{F}^P of problem (1.1) is closed and bounded. If x_j is invariant throughout \mathcal{F}^P for a given j , then $\{x_j^k\}$ converges to x_j as the iterates (x^k, y^k, λ^k) approach primal feasibility. In that case

$$\frac{x_j^{k+1}}{x_j^k} \rightarrow 1 \text{ as } Ax^k - b \rightarrow 0,$$

which explains the behavior of the Tapia indicator in this case.

6 Dual Variables

The behavior of the Tapia indicator depends essentially on three factors, (i) the limit points of $\{(x^k, z^k)\}$ satisfy strict complementarity, (ii) linearized complementarity is approached, and (iii) the sequence of steps $\{(\Delta x^k, \Delta z^k)\}$ converges to zero. However a surprising phenomenon was observed when the logarithmic Tapia indicators were applied to the sequence of dual variables $\{y^k\}$ (Lagrange multipliers corresponding to the equality constraints $Ax - b = 0$). The Tapia indicator trajectories seem to cluster near two values, with the corresponding variables being zero or positive on the approximate solution obtained from the code. Moreover, the logarithmic Tapia indicator trajectories have the same behavior for the logarithmic Tapia indicators for either $\{x^k\}$ or $\{z^k\}$. The trajectories split into several groups. Two examples are given in Figure 6 for problems BLEND and SCSD1. It is worth mentioning that all groups of logarithmic Tapia indicators discussed in Section 4 were observed for the dual variables y .

The behavior of the logarithmic Tapia indicators corresponding to variables y_j with $y_j^* > 0$ is no surprise if the sequence $\{y^k\}$ converges. But the split between the trajectories corresponding to positive variables as well as the clustering of the ones corresponding to zero dual variables near σ is not expected. The reason being that these variables do not satisfy any complementarity conditions.

We point out that if some of the primal variables x are added as slack variables to transform the primal problem into the form (1.1), then some of the dual variables y_j will be related to a dual slack z_ℓ by the following relation

$$y_j + z_\ell = c_\ell,$$

which may explain the behavior of the Tapia (and logarithmic Tapia indicator) for the dual variables y . This suggests that the behavior of the dual variables y plays a nontrivial role in the behavior of the primal-dual interior-point methods.

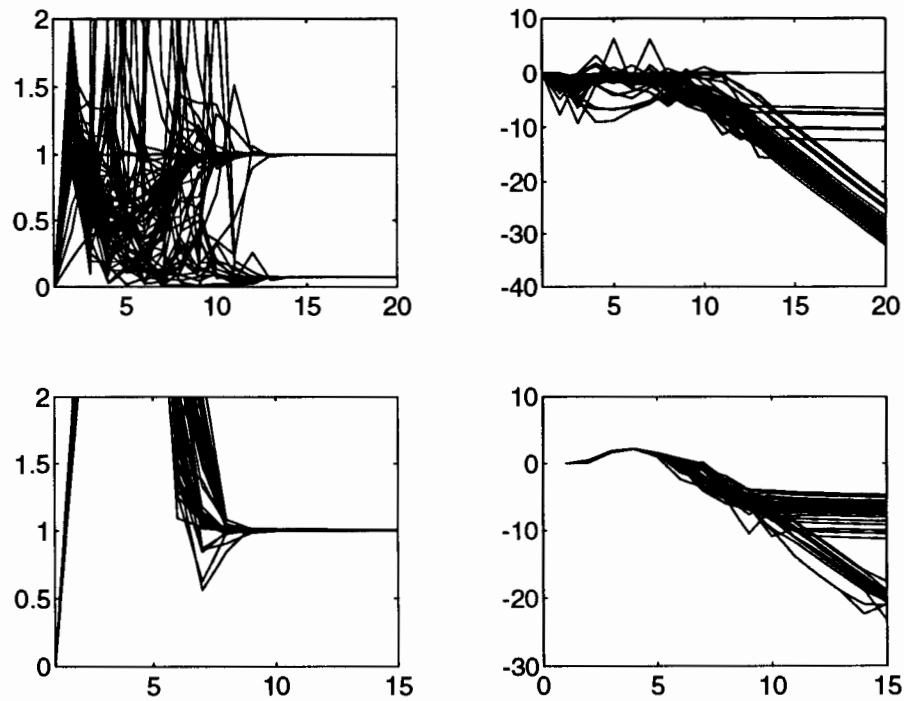


Figure 13: The logarithmic Tapia indicator for all dual variables in BLEND and SCSD1.

7 Conclusion

In this paper, several logarithmic indicators have been introduced. Among these the logarithmic Tapia indicators have proven to be powerful tools for identifying several subgroups of variables in interior-point methods. We have not been able to obtain a definite interpretation for their behavior for some subgroups of variables. We believe that, with the understanding gained from these indicators on the behavior of subgroups of variables, more efficient implementation of finite termination and hybrid methods is possible. We also believe that the behavior of several groups of variables merits further study to analyze its effect on the implementation as well as the theory of interior-point methods.

References

- [1] 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.
- [2] I. Choi and D. Goldfarb. A note on solution containing ellipsoids in linear programming. *To appear in J.O.T.A.*
- [3] 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.
- [4] A. S. El-Bakry. *On the role of indicators in identifying zero variables in linear programming*. PhD thesis, Department of Mathematical Sciences, Rice University, Houston, Texas 77251, 1991.
- [5] A. S. El-Bakry, R. A. Tapia, and Y. Zhang. A study of indicators for identifying zero variables in interior-point methods. *SIAM Review*, ??:??–??, 1993.
- [6] D. M. Gay. Electronic mail distribution of linear programming test problems. *COAL Newsletter*, 13:10–12, 1985.
- [7] D. M. Gay. Stopping tests that compute optimal solutions for interior–point linear programming algorithms. In S. Gomez, J. P. Hennart, and R. A. Tapia, editors, *Advances in Numerical Partial Differential Equations and Optimization : Proceedings of the Fifth Mexico–United States Workshop*, volume 47 of *Proceedings in Applied Mathematics*, pages 17–42. SIAM Publications, Philadelphia, PA, USA, 1991.
- [8] P.E. Gill, W. Murray, D.B. Ponceleon, and Saunders M.A. Preconditioners for indefinite systems arising in optimization. *SIAM J. Matrix Analysis and Applications*, 13:292–311, 1992.
- [9] 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.
- [10] C. C. Gonzaga and R.A. Tapia. On the convergence of the mizuno-todd-ye algorithm to the analytic center of the solution set. Technical Report TR92-41, Department of Computational and Applied Mathematics, Rice University, Houston, Tx 77251, 1992.
- [11] C. C. Gonzaga and R.A. Tapia. On the quadratic convergence of the simplified mizuno-todd-ye algorithm for linear programming. Technical Report TR92-42, Department of Computational and Applied Mathematics, Rice University, Houston, Tx 77251, 1992.

- [12] N. K. Karmarkar and K. G. Ramakrishnan. Computational results of an interior point algorithm for large scale linear programming. *Mathematical Programming*, 52:555–586, 1991.
- [13] M. Kojima. Determining basic variables of optimal solutions in Karmarkar’s new LP algorithm. *Algorithmica*, 1:499–515, 1986.
- [14] M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior point method for linear programming. In Nimrod Megiddo, editor, *Progress in mathematical programming, interior-point and related methods*, pages 29–47. Springer-Verlag, New York, 1989.
- [15] V.V. Kovacevic-Vujcic. Improving the rate of convergence of interior-point methods for linear programming. *Mathematical Programming*, 52:467–479, 1991.
- [16] I. J. Lustig. An implementation of a strongly polynomial time algorithm for basis recovery. Dept. Civil Eng. and O.R., Princeton University, in preparation.
- [17] I.J. Lustig, R.E. Marsten, and D.F. Shanno. Computational experience with a primal-dual interior point method for linear programming. *Linear Algebra and Appl.*, 152:191–222, 1989.
- [18] I.J. Lustig, R.E. Marsten, and D.F. Shanno. On implementing Mehrotra’s predictor-corrector interior point method for linear programming. *SIAM Journal on Optimization*, 2(3):435–449, 1992.
- [19] 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.
- [20] S. Mehrotra. Implementations of affine scaling methods: towards faster implementations with complete Cholesky factor in use. Technical Report 89-15, Dept. Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, 1989.
- [21] S. Mehrotra. On finding a vertex solution using interior point methods. *Linear Algebra and Its Applications*, 152:233–253, 1991.
- [22] S. Mizuno, M.J. Todd, and Y. Ye. A surfaces of analytic centers and infeasible-interior-point algorithms for linear programming. Technical Report No.1037, School of ORIE, Cornell University, Ithaca, NY 14853-3801, 1992.
- [23] M. G. C. Resende and G. Veiga. A dual affine scaling algorithm for minimum cost network flow problems. Technical Report 1.3, Revised February 1991, Mathematical Sciences Research Center, AT & T Bell Laboratories, Murray Hill, NJ 07974, USA, 1990.

- [24] 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.
- [25] R. A. Tapia and Y. Zhang. An optimal-basis identification technique for interior-point linear programming methods. *Linear Algebra and Its Applications*, 152:343–363, 1991.
- [26] R.A. Tapia, Y. Zhang, and Y. Ye. On the convergence of the iteration sequence in primal-dual interior-point methods. Technical Report TR91-24, Department of Mathematical Sciences, Rice University, Houston, TX. 77251-1892, 1991.
- [27] M.J. Todd. Improved bounds and containing ellipsoid in Karmarkar’s linear programming algorithm. *Math. of Operations Research*, 13:650–659, 1988.
- [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] F. C. de Vreede. Indicators as stopping criteria in interior point methods in linear programming. Master’s thesis, The Netherlands, 1993.
- [30] Y. Ye. Recovering optimal basis in Karmarkar’s polynomial algorithm for linear programming. *Mathematics of Operations Research*, 15:564–572, 1990.
- [31] Y. Ye. On the finite convergence of interior-point algorithms for linear programming. Technical Report 91–05, Dept. of Management Sciences, University of Iowa, Iowa City, IA 52242, USA, 1991. To appear in *Mathematical Programming Ser. B*.
- [32] Y. Zhang and R. A. Tapia. On the convergence of interior-point methods to the center of the solution set in linear programming. Technical Report TR91-30, Department of Mathematical Sciences, Rice University, Houston, TX. 77251-1892, 1991.
- [33] Y. Zhang, R. A. Tapia, and J. E. Dennis. On the superlinear and quadratic convergence of primal-dual interior-point linear programming algorithms. *SIAM J. on Optimization*, 2:304–324, 1992.