# Combining interior-point and pivoting algorithms for linear programming

Erling D. Andersen *
Yinyu Ye †

October 19, 1995

**Abstract**

We propose a new approach to combine linear programming (LP) interior-point and simplex pivoting algorithms. In any iteration of an interior-point algorithm we construct a related LP problem, which approximates the original problem, with a known (strictly) complementary primal-dual solution pair. Thus, we can apply Megiddo's [27] pivoting procedure to compute an optimal basis for the approximate problem in strongly polynomial time. We show that, if the approximate problem is constructed from an interior-point iterate sufficiently close to the optimal face, then any optimal basis of the approximate problem is an optimal basis for the original problem. If the LP data are rational, the total number of interior-point iterations to create such a sufficient approximate problem is bounded by a polynomial in the data size. We develop a modification of Megiddo's procedure and discuss several implementation issues in solving the approximate problem. We also report encouraging computational results for this combined approach.

## 1 Introduction

Since Karmarkar published his paper [21] in 1984, there has been a rapid development in interior-point methods for linear programming (LP). The computational issues of this development are summarized by several researchers in a recent volume of *ORSA Journal Computing*. According to Lustig, Marsten and Shanno

*Department of Management, Odense University, DK-5230 Odense M, Denmark. E-mail: eda@busieco.ou.dk. The author is currently visiting Department of Management Sciences, The University of Iowa.

†Department of Management Sciences, The University of Iowa, Iowa City, IA 52242. E-mail: bbuyin@vaxa.weeg.uiowa.edu. This author is supported in part by NSF Grant DDM-9207347 and the University of Iowa Business School Summer Grant.

[26], few open questions are left open in practice. They are: a) detect infeasibility and unboundedness, b) warm-start, and c) generate an optimal basis. In this paper we will address the last issue.

Contrary to pivoting algorithms, interior-point algorithms will in general return an approximate solution in the interior of the optimal face. Except for non-degenerate cases, this solution will not be basic. In practice users sometimes want basic solutions for a couple of reasons. First of all, a basic solution has generally less nonzero components than a solution in the interior of the optimal face, which is desirable when LP relaxations of integer programming problems are solved. Secondly, an optimal basic solution provides easy warm-start for the simplex method when a sequence of closely related problems are solved. Of course, it should be mentioned that there exist many cases when an optimal interior-point solution is more desirable than a basic solution, see, Christiansen and Kortanek [11] and, more recently, Jansen et al. [19], Adler[1] and Greenberg, [16].

It has been sought to combine interior-point and simplex pivoting algorithms:

- Kortanek and Zhu [24] (also see Charnes and Kortanek [10]) proposed a "purification" pivoting procedure from an interior-point to a corner-point having objective value no worse. The number of pivot steps to generate an optimal basic solution is finite but may not be polynomial.

- Given an optimal primal and dual solution pair, Megiddo proposed a strongly polynomial pivoting procedure to compute an optimal basis. One practical difficulty to use his procedure in interior-point algorithms is the fact that interior-point algorithms never generate an exact optimal solution in finite time.

- A cross-over scheme, to run an interior-point algorithm first and later switch to the simplex method, was successively implemented in a recent large-scale case study by Bixby et al. [7] and also by Bixby and Saltzman [6]. It uses a heuristic, based on information from interior-point algorithm iterates, to construct an initial basis for the simplex method to begin with. No theoretical result is established for this approach.

The goal of our paper is to develop an approach to combine interior-point and pivoting algorithms that possesses **both** theoretical guarantee and practical efficiency. We accomplish this goal using two ideas. The first idea is to construct an artificial LP problem, which approximates the original LP problem, in any iteration of an interior-point algorithm. A key feature of the approximate problem is that it has a known (strictly) complementary solution pair. Moreover, we show that, if the approximate problem is constructed from an interior-point iterate sufficiently close to the optimal face, then every optimal basis of this approximate problem is an optimal basis of the original problem. If the LP data are rational, the total number of interior-point iterations to create such a sufficient approximate problem is bounded above by $O(\sqrt{n}L)$, where $L$ is the data size and $n$ is the number of variables in the standard primal form.

Since a complementary solution pair is known for this approximate problem, we could simply apply Megiddo's procedure [27] to compute an optimal basis of the approximate problem in $n$ pivot steps. Our second idea is to develop a modification (which is also strongly polynomial) of Megiddo's and Bixby and Saltzman's procedures. In fact, our modification allows either cross-over to a pivoting algorithm completely or to execute both interior-point and pivoting algorithms concurrently.

This paper is organized as follows. In the next section we introduce some relevant technical results. In Section 3, we show how to construct an LP problem, which approximates the original LP problem, using information from interior-point iterates. Afterwards, in Section 4 we develop a purification pivoting procedure, based on Megiddo and Bixby and Saltzman, for obtaining an optimal basis from a known complementary solution. In Section 5 we discuss implementation issues and propose a criterion on when to start the purification procedure for solving the approximate problem. Finally in Section 6 and 7 we present computational results and final remarks.

# 2 Technical review

Any LP problem can be stated in the following form

$$\begin{array}{rl} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \tag{1}$$

where $c, x \in R^n$, $A \in R^{m\times n}$ and $b \in R^m$. Without lost of generality, we assume that $A$ has full row rank. We will denote (1) by $LP(A, b, c)$. The dual to (1) is

$$\begin{array}{rl} \max & b^T y \\ \text{s.t.} & A^T y + s = c \\ & s \geq 0 \end{array} \tag{2}$$

where $y \in R^m$ and $s \in R^n$. Let the support

$$\sigma(v) = \{j : v_j \neq 0,\ v \in R^n\}. \tag{3}$$

We will use the following notation

$$\bar{\sigma}(v) = \{j : v_j = 0,\ v \in R^n\}. \tag{4}$$

It is well-known that if there exists an optimal solution $x^*$ and $(y^*, s^*)$ to (1) and (2), respectively, then the duality gap

$$(x^*)^T s^* = c^T x^* - b^T y^* = 0. \tag{5}$$

Moreover there always exists a strictly complementary solution such that

$$x^* + s^* > 0 \quad \text{or} \quad \sigma(x^*) = \bar{\sigma}(s^*). \tag{6}$$

This result was proven by Goldman and Tucker [15]. Also see a recent interesting proof by Jansen et al. [20] based on interior-point methods. Furthermore for any strictly complementary solution $(x^*, y^*, s^*)$ we have

$$\sigma(x^*) = P^* \quad \text{and} \quad \sigma(s^*) = Z^* \tag{7}$$

where $P^* \cup Z^* = \{1, ..., n\}$. Thus, the partition is invariant with respect to the solution. Let us call it the optimal partition. In what follows, $P$ and $Z$ will also be used as matrices obtained from $A$ corresponding to their index sets. Thus, the optimal face for the primal problem is

$$\Theta_P = \{x : \ Ax = b, \ x \geq 0, x_j = 0 \text{ for } j \in Z^*\}, \tag{8}$$

and the one for the dual is

$$\Theta_D = \{(y, s) : \ A^T y + s = c, \ s \geq 0, s_j = 0 \text{ for } j \in P^*\}. \tag{9}$$

These faces have interior points, because $\tilde{\Theta}_P = \{x : Ax = b, \ x_{P^*} > 0, \ x_j = 0 \text{ for } j \in Z^*\}$ and $\tilde{\Theta}_D = \{(y, s) : \ A^T y + s = c, \ s_{Z^*} > 0, \ s_j = 0 \text{ for } j \in P^*\}$ are nonempty. Now we define

$$\xi_P = \min_{j \in P^*} \{\max \ x_j : x \in \Theta_P\} \tag{10}$$

and

$$\xi_D = \min_{j \in Z^*} \{\max \ s_j : (y, s) \in \Theta_D\}, \tag{11}$$

and let

$$\xi = \min(1, \xi_P, \xi_D). \tag{12}$$

We must have $\xi > 0$. It can be verified that, if $LP(A, b, c)$ has rational data, then $\xi \geq 2^{-L}$, where L is the bit size of $A$, $b$ and $c$. Throughout this paper, we assume that $LP(A, b, c)$ has a solution.

Many polynomial (and practical) interior-point algorithms (e.g. Kojima et al. [23], Mizuno et al. [31], Monteiro and Adler [33] and Lustig et al. [25]) generate a solution sequence (or subsequence) $(x^k, s^k)$ such that

$$(x^{k+1})^T s^{k+1} \leq (x^k)^T s^k \text{ and } \frac{\min(X^k s^k)}{(x^k)^T s^k} \geq \Omega(1/n). \tag{13}$$

Given an iterate $(x^k, s^k)$ of the sequence satisfying (13), define

$$P^k = \{j : x_j^k \geq s_j^k\} \ \text{ and } \ Z^k = \{1, ..., n\} \setminus P^k. \tag{14}$$

Based on the result of Güler and Ye [17] (also see Mehrotra and Ye[30] [36]), we have

**Theorem 2.1** *Let $\{x^k, s^k\}$ be generated by an $O(\sqrt{n}L)$ interior-point algorithm satisfying (13) and starting from $x^0 = e$ and $s^0 = e$. Then,*

1.
$$0 < e^T x^k + e^T s^k \leq 2n, \ \forall \ k. \tag{15}$$

2.
$$x_j^k \geq \Omega(\xi/n) \ \textit{and} \ s_j^k \leq (x^k)^T s^k/\xi, \ \forall \ k \ \textit{and} \ j \in P^* \tag{16}$$

*and*
$$s_j^k \geq \Omega(\xi/n) \ \textit{and} \ x_j^k \leq (x^k) s^k/\xi, \ \forall \ k \ \textit{and} \ j \in Z^*. \tag{17}$$

3. *When $(x^k)^T s^k \leq O(\xi^2/n)$, then*
$$s_j^k < O(\xi/n) \ \forall \ j \in P^* \ \textit{and} \ x_j^k < O(\xi/n) \ \forall \ j \in Z^*, \tag{18}$$

*so that*
$$P^k = P^* \ \textit{and} \ Z^k = Z^*. \tag{19}$$

**Proof:** Since

$$(x^k - x^0)^T (s^k - s^0) = (x^k - e)^T (s^k - e) = 0,$$

we have
$$e^T x^k + e^T s^k = n + (x^k)^T s^k \leq 2n$$

which proves the first statement.

Also for any complementary solution $(x^*, s^*)$, we have

$$(x^k - x^*)^T (s^k - s^*) = 0.$$

Thus,
$$\sum_{j \in Z^*} x_j^k s_j^* + \sum_{j \in P^*} s_j^k x_j^* = (x^k)^T s^k$$

and
$$\sum_{j \in Z^*} \frac{s_j^*}{s_j^k} \frac{x_j^k s_j^k}{(x^k)^T s^k} + \sum_{j \in P^*} \frac{x_j^*}{x_j^k} \frac{s_j^k x_j^k}{(x^k)^T s^k} = 1.$$

Therefore, for any $j \in P^*$ and any complementary solution $(x^*, s^*)$

$$s_j^k x_j^* \leq (x^k)^T s^k \quad \text{and} \quad \frac{x_j^*}{x_j^k} \frac{s_j^k x_j^k}{(x^k)^T s^k} \leq 1.$$

Given an $j \in P^*$, we let $x^* = \arg\max \{x_j : x \in \Theta_P\}$. Then, using the above fact, the definition of $\xi$, and (13), we obtain

$$s_j^k \xi \leq s_j^k x_j^* < (x^k)^T s^k \quad \text{and} \quad \frac{\xi}{x_j^k} \leq \frac{x_j^*}{x_j^k} < \frac{(x^k)^T s^k}{x_j^k s_j^k} \leq O(n),$$

which proves the second statement for $j \in P^*$. Similarly, we can prove the statement for other $j \in P^*$ and for all $j \in Z^*$.

The third statement of the theorem directly follows from the second statement when $(x^k)^T s^k \leq O(\xi^2/n)$. $\square$

The above theorem indicates that the partition $(P^k, Z^k)$ will be optimal after $O(\sqrt{n}\log(n/\xi))$ iterations. If $LP(A,b,c)$ has rational data, this will happen in $O(\sqrt{n}L)$ iterations. It also states that as $(x^k)^T s^k \to 0$, zero variables uniformly go to zero and positive variables are uniformly bounded away from zero. Hence, interior-point methods will converge to a solution in the interior of the optimal primal and dual faces.

Now let $(B, N)$ denote a partition of variables into basic and non-basic variables. We actually let $(B, N)$ denote both index sets and the corresponding matrices obtained from $A$. Then, $B$ is an optimal basis if and only if $B$ is non-singular,

$$x_B = B^{-1}b \geq 0, \tag{20}$$

and

$$y = B^{-T}c_B, \; s_N = c_N - N^T y \geq 0. \tag{21}$$

If (1) is consistent and bounded, then an optimal basis always exists. This is the foundation of the simplex pivoting algorithm. A basic solution is said to be primal degenerate if at least one component in $x_B$ is zero. Similarly the dual solution is degenerate if at least one of the components in $s_N$ is zero.

# 3 An approximate LP problem

We make an assumption that we have an $O(\sqrt{n}L)$ interior-point algorithm which generates a sequence of iterates $(x^k > 0, y^k, s^k > 0)$ for $LP(A,b,c)$. (Actually we do not need to assume that these iterates are feasible, see such an algorithm in Ye et al. [38].) In addition, we assume that the interior-point algorithm generates $(P^k, Z^k)$ by (14) which is an estimate for the optimal partition $(P^*, Z^*)$.

Using these assumptions, we define a perturbed LP problem

$$\begin{array}{rl} \min & (c^k)^T x \\ \text{s.t.} & Ax = b^k \\ & x \geq 0 \end{array} \tag{22}$$

where

$$b^k = P^k x^k_{P^k}, \; c^k_{P^k} = (P^k)^T y^k, \; c^k_{Z^k} = (Z^k)^T y^k + s^k_{Z^k}. \tag{23}$$

We can now state the following Proposition

**Proposition 3.1** *The solutions $x = (x^k_{P^k}, 0)$ and $(y,s) = (y^k, (0, s^k_{Z^k}))$ are an optimal primal-dual pair to (22). In fact, they are strictly complementary.*

**Proof:** Feasibility follows from construction, optimality from $x^T s = 0$, and strict complementarity from $x^k_{P^k} > 0$ and $s^k_{Z^k} > 0$. $\square$

Due to Theorem 2.1, we know that $LP(A, b^k, c^k)$ will eventually share the same optimal partition with $LP(A, b, c)$. We would like to emphasize that this fact does not hold in general. If $b^k$ and $c^k$ are not constructed from (14), (23) and Proposition 3.1, then $b^k \to b$ and $c^k \to c$ does not necessarily guarantee $(P^k, Z^k)$ converges to $(P^*, Z^*)$ (see Theorem 3 and an example of Mizuno et al. [32]). In other words, the optimal partition for the approximate LP and the one for the original LP may be different, even if $(b^k, c^k)$ is arbitrarily close to $(b, c)$.

For simplicity, we assume that the iterate $(x^k, y^k, s^k)$ is feasible. Let $\eta = \max(1, \|a_j\|)$ where $a_j$ is the $j$'th column in A. (Again, if $LP(A, b, c)$ has rational data, $\eta \le 2^L$.) Then, after $(P^k, Z^k) = (P^*, Z^*)$, we have from Theorem 2.1 that

$$\left\|b^k - b\right\| = \left\|Z^k x^k_{Z^k}\right\| \le n\eta (x^k)^T s^k / \xi \tag{24}$$

and

$$\left\|c^k - c\right\| = \left\|s^k_{P^k}\right\| \le \sqrt{n}(x^k) s^k / \xi \tag{25}$$

which shows that $b^k \to b$ and $c^k \to c$, as $(x^k)^T s^k \to 0$. Moreover, if $(x^k)^T s^k \le \min\{O(\xi^2/n), 2^{-t}\xi/(n\eta)\}$, we have

$$(P^k, Z^k) = (P^*, Z^*), \ \left\|b^k - b\right\| \le 2^{-t} \ \text{ and } \ \left\|c^k - c\right\| \le 2^{-t}. \tag{26}$$

Now we state the following theorem.

**Theorem 3.1** *Let $B$ be an optimal basis for $LP(A, b^k, c^k)$ constructed by (22) and (23). Then, there is $0 < \bar{t} < \infty$ such that $B$ must be also an optimal basis for the original $LP(A, b, c)$ when $(x^k)^T s^k \le 2^{-\bar{t}}$. Furthermore, if $LP(A, b, c)$ has rational data, then $\bar{t} \le O(L)$.*

**Proof:** Let $B$ be any basis of $A$ with $B\bar{x}_B = b$ and $B^T\bar{y} = c_B$. Among those nonzero components in $\bar{x}_B$ and $\bar{s} = c - A^T\bar{y}$, let the value of the smallest absolute-value component be $\tau_B$, which is positive. Then, we define $\tau = \min(1, \min_B(\tau_B))$. $\tau$ must be positive since the total number of possible bases in A is finite. Also let $\zeta = \max(1, \max_B(\|B^{-1}\|))$. Again, if $LP(A, b, c)$ has rational data, $\tau \le 2^{-L}$ and $\zeta \le 2^L$.

We first prove the following lemma

**Lemma 3.1** *Let $B$ be an optimal basis for $LP(A, b^k, c^k)$, where $\left\|b^k - b\right\| < \tau/\zeta$ and $\left\|c^k - c\right\| < \tau/(\eta\zeta)$. Then, $B$ is also an optimal basis for $LP(A, b, c)$, i.e., both solutions $\bar{x}_B$ of $B\bar{x}_B = b$ and $\bar{s} = c - A^T\bar{y}$, where $B^T\bar{y} = c_B$, are nonnegative.*

**Proof:** Let the corresponding optimal primal basic solution be $\bar{x}_B^k \geq 0$ and the dual basic solution be $(\bar{y}^k, \bar{s}^k = c - A^T\bar{y}^k)$ for $LP(A, b^k, c^k)$, i.e., $B\bar{x}_B^k = b^k$ and $B^T\bar{y}^k = c_B^k$. Then, if $x_B \not\geq 0$, each of its negative components must be less than $-\tau$. Let the $i$'th component be negative, i.e., $\bar{x}_i \leq -\tau$. On the other hand, we have

$$\bar{x}_B^k - x_B = B^{-1}(b^k - b)$$

or

$$\left\|\bar{x}_B^k - \bar{x}_B\right\| = \left\|B^{-1}(b^k - b)\right\| \leq \left\|B^{-1}\right\| \left\|b^k - b\right\|.$$

Thus,

$$|\bar{x}_i^k - \bar{x}_i| \leq \left\|B^{-1}\right\| \left\|b^k - b\right\| < \tau$$

or

$$\bar{x}_i^k < \tau + \bar{x}_i \leq \tau - \tau = 0.$$

This contradicts that $\bar{x}_B^k \geq 0$.

Similarly, if $\bar{s}_j \leq -\tau$, then

$$|\bar{s}_j^k - \bar{s}_j| = |a_j^T(\bar{y}^k - y)| \leq |a_j^T B^{-T}(c_B^k - c_B)| \leq \left\|a_j^T\right\| \left\|B^{-1}\right\| \left\|c^k - c\right\| < \tau.$$

This implies that

$$\bar{s}_j^k < \tau + \bar{s}_j \leq \tau - \tau = 0,$$

which is also a contradiction. □

Now, we just need to choose $\bar{t}$ big enough but finite such that $(x^k)^T s^k \leq 2^{-\bar{t}}$ to establish

$$(P^k, Z^k) = (P^*, Z^*)$$

and

$$\left\|b^k - b\right\| < \tau/\zeta \text{ and } \left\|c^k - c\right\| < \tau/(\eta\zeta).$$

From relations (24) and (25), this can be accomplished if

$$2^{-\bar{t}} \leq \min\{O\left(\frac{\xi^2}{n}\right), \frac{\xi\tau}{n\eta\zeta}\}.$$

If $LP(A, b, c)$ has rational data, one can verify that $\bar{t} \leq O(L)$. Thus, the theorem is proved. □

From the above theorem, we have

**Corollary 3.1** *Let $\{x^k, y^k, s^k\}$ be generated by an $O(\sqrt{n}L)$ interior-point algorithm satisfying (13) and starting from $x^0 = e$ and $s^0 = e$. Then, after at most $O(\sqrt{n}L)$ iterations, an optimal basis of $LP(A, b^k, c^k)$ constructed by (22) and (23) is also optimal for $LP(A, b, c)$.*

# 4 Purification pivoting procedures

In this section, we discuss two purification pivoting procedures to compute an optimal basis for (22) from a complementary solution pair. For convenience we will assume that a full set of artificial variables have been added to the problem (22), and let $V = \{n+1, \ldots, n+m\}$ represent the set of artificial variables. Thus, $A$ has been augmented with an identity matrix, i.e., $A := [A, I]$. Using results described in previous sections, we now assume the following for the problem $LP(A, b^k, c^k)$:

(A1) The optimal partition $(P, Z)$ is known and $V \subseteq Z$.

(A2) A primal optimal solution $\tilde{x}$ is known such that $A\tilde{x} = b^k$, $\tilde{x}_Z = 0$ and $\tilde{x}_P > 0$.

(A3) A dual optimal solution $(\tilde{y}, \tilde{s})$ is known such that $A^T\tilde{y} + \tilde{s} = c^k$, $\tilde{s}_{Z\setminus V} > 0$ and $\tilde{s}_P = 0$.

Here, the dual slack variable $\tilde{s}_V$, corresponding to the fixed primal artificial variables, are free of sign and can assume any value. Note that the pivoting procedures discussed below work even if $\tilde{x}_P > 0$ and $\tilde{s}_{Z\setminus V} > 0$ are relaxed to $\tilde{x}_P \geq 0$ and $\tilde{s}_{Z\setminus V} \geq 0$ in (A2) and (A3).

## 4.1 A complementarity pivoting procedure

For simplicity, we replace $(b^k, c^k)$ by $(b, c)$ in the rest of this section. The first pivoting procedure generates iterates which satisfy the complementary condition, but not the non-negativity constraint. Therefore we call the procedure a complementarity pivoting procedure, which is related to the method of Kortanek and Zhu [24].

The idea of the primal phase of the procedure can be outlined as follows. Let $(B, N)$ be a basic and non-basic partition such that

$$x_B = B^{-1}b, \quad x_N = 0, \quad \text{and} \quad x_Z = 0.$$

Such a basis is easy to find, see Algorithm 4.1 below. In general $x_B \not\geq 0$; otherwise $B$ is primal optimal. Therefore, there must be at least one negative component in $x_B$. Now define a "purified" solution

$$\hat{x} = \alpha x + (1-\alpha)\tilde{x} \tag{27}$$

where

$$\alpha = 1/(1-\rho) \quad \text{and} \quad \rho = \min_{B_i \in B}\{x_{B_i}/\tilde{x}_{B_i} : \ \tilde{x}_{B_i} > 0\}.$$

Clearly $\alpha \in (0,1)$ and by construction $A\hat{x} = b$, $\hat{x} \geq 0$ and $\hat{x}_Z = 0$. Hence $\hat{x} \in \Theta_P$. Moreover, $\hat{x}$ has at least one more zero component than $\tilde{x}$ does. Let

$$I = \{B_i : x_{B_i}/\tilde{x}_{B_i} = \rho, \ \tilde{x}_{B_i} > 0\}. \tag{28}$$

It follows that at least one of the basic variables $B_i \in I$ can be replaced with one variable $j$ where $j \in P \cap N$ and $\tilde{x}_j > 0$. If we let $\hat{x}$ replace $\tilde{x}$ then this procedure can be repeated. After at most $|P|$ pivots we must have $A\tilde{x} = Ax = b$ and $\tilde{x}_N = x_N = 0$. Thus, $\tilde{x}_B = x_B \geq 0$ is primal feasible and optimal.

We can now formally state the following procedure.

**Algorithm 4.1**

1. *Choose a basis* $B$, $k = 0$.
2. *while* $(\ \exists B_i \in Z : x^k_{B_i} \neq 0\ )$
3. *Exchange the ith basic variable with a variable* $j \in P \cap N$
4. $k := k + 1$
5. *end while*
6. *while* $(\ \exists B_i \in P : x^k_{B_i} < 0\ )$
7. $\rho = min\{x^k_{B_i}/\tilde{x}^k_{B_i} : \tilde{x}^k_{B_i} > 0\}$
8. $I = \{i : x^k_{B_i}/\tilde{x}^k_{B_i} = \rho, \tilde{x}^k_{B_i} > 0\}$
9. *Exchange one basic variables* $B_i \in I$ *with a variable* $j \in P \cap N, \quad \tilde{x}^k_j > 0$
10. $\alpha = 1/(1-\rho), \quad \tilde{x}^{k+1} = \alpha x^k + (1-\alpha)\tilde{x}^k$
12. $k := k + 1$
13. *end while*

In the above procedure we have used the convention that $x^k_B = B^{-1}b$ where $B$ is the current basis. Steps 2 to 5 construct a basic solution such that $x_Z = 0$. Steps 6 to 13 purify the interior-point solution $\tilde{x}$ to an optimal primal basic solution using the idea outlined above.

Starting from the primal optimal basis $B$ obtained in the primal phase, we can develop a similar dual purification procedure to purify the interior-point dual solution $(\tilde{y}, \tilde{s})$ to an optimal dual basic solution. It can be verified that these two phases together generate an optimal primal-dual basis using less than $n$ pivots.

## 4.2 A superbasic pivoting procedure

Megiddo [27] proposed a procedure to generate an optimal basis from an optimal solution pair. His procedure was slightly modified by Bixby and Saltzman [8]. Their modification generates a basic solution that maintains non-negativity in all pivots and move iterates towards complementarity.

This procedure can be described as follows. Let $(B, N)$ be any basic and non-basic partition of variables. Note

$$\tilde{x}_B = B^{-1}(b - N\tilde{x}_N). \tag{29}$$

Usually $(\tilde{x}_B, \tilde{x}_N)$ is called a primal superbasic solution, since some of its non-basic variables are not zero (e.g., Nazareth [34, p. 31]).

The primal phase of Megiddo's procedure either moves one non-basic variable in $\tilde{x}_N$ to zero, if possible, or pivots them into the basis one at a time. Each move or pivot reduces the number of non-zero and non-basic variables by one, and therefore the number of total moves and pivots is bounded by $|P|$.

This procedure is formally stated as follows:

**Algorithm 4.2**

1. *Choose a basis* $B$ *and let* $x^0 = \tilde{x},\ k = 0$.
2. *while*$(\exists\ j \in N : x_j^k \neq 0)$
3. *Use the primal ratio test to move variable* $x_j^k$ *to zero if possible or pivot it into the basis*
4. *Update* $(B, N),\ x^k,\ k := k + 1$
5. *end while*

The resulting $B$ will be a primal optimal basis. One possible choice of $B$ in Step 1 is $V$, i.e., the set of artificial variables.

Similarly, one can purify the superbasic dual solution $(\tilde{y}, \tilde{s})$ to a basic solution. The dual phase can be stated as follows:

**Algorithm 4.3**

1. *Choose a basis* $B$ *and let* $y^0 = \tilde{y},\ s^0 = c - A^T\tilde{y},\ k = 0$
2. *while*$(\exists\ j \in B : s_j^k \neq 0)$
3. *Use the dual ratio test to move variable* $s_j^k$ *to zero if possible or pivot it out of the basis*
4. *Update* $(B, N),\ y^k,\ s^k,\ k := k + 1$
5. *end while*

The resulting $B$ will be a dual optimal basis. If the initial basis of the dual phase is primal optimal, then it will remain primal optimal, because all pivots in Algorithm 4.3 are primal degenerate. Therefore, the final basis will be both primal and dual optimal. Furthermore the number of pivots in the dual phase is bounded by $|Z|$. In summary the superbasic pivoting procedure generates an optimal basis in less than $n$ pivots from a complementary solution.

# 5 Implementation issues

We now discuss several implementation issues related to the purification pivoting procedures described in the preceding section. One issue is how to estimate the optimal partition and when to start the purification procedure. The theory of Section 3 indicates that, if the interior-point iterate is close enough to the optimal face, then we have an accurate estimate to the optimal partition and we should likely be successful in finding an optimal basis for the original LP problem.

We propose to use the primal-dual Newton direction to determine whether or not the iterate is close to the optimal face. Mehrotra [29] and Ye et al. [37] have shown that, for any LP problem,

$$P^k = \{j : |\Delta x_j^k|/x_j^k \leq |\Delta s_j^k|/s_j^k\} \tag{30}$$

converge towards the optimal partition $P^*$ where $(\Delta x, \Delta y, \Delta s)$ is the primal-dual Newton direction. This technique is closely related to Tapia's indicator, see [12]. Note that this definition of $P^k$ is scaling invariant.

In addition, results in [29] and [37] show that, if the duality gap $(x^k)^T s^k$ is sufficiently small or the iterate is sufficiently close to the optimal face, the iterates based on using the primal-dual Newton direction will reduce the duality gap quadratically. Conversely, when fast or superlinear convergence is about to occur, we anticipate that the iterate is close to the optimal face and the partition $P^k$ will likely equal $P^*$. Thus, we start purification process when the duality gap is reasonably small and the fast convergence is observed.

More precisely, we use the following criteria. Let $\alpha^{\max}$ be the maximum possible step length to the boundary along the primal-dual Newton direction. If the fast convergence occurs, this step length will approach one. We therefore try to start purification if a solution is close to being primal and dual feasible, and

$$\alpha^{\max} \geq (1-\beta^1), \quad \frac{c^T x^k - b^T y^k}{1+|b^T y^k|} \leq \beta^2 \tag{31}$$

or

$$\frac{(c^T x^k - b^T y^k)}{1+|b^T y^k|} \leq \beta^3, \tag{32}$$

where $\beta^1$, $\beta^2$ and $\beta^3$ are suitably chosen constants. (The second stopping criterion will be used for problems where fast convergence is hardly obtained.)

The optimal basis of $LP(A, b^k, c^k)$ may not be optimal for the original LP problem. One option is to continue one or more interior-point iterations and to obtain a more accurate approximate problem. In this case, we should use the basis resulted from the latest purification try as the initial basis for the new try. We call this option a retry. Another option is to use this basis as the starting basis for the simplex algorithm, such as the cross-over procedure proposed by Bixby and Saltzman [8]. Moreover, this basis could be used as a preconditioner for an iterative method for solving the least squares problem in interior-point algorithms. In fact, how to choose such a precondition basis and how to update it are active research topics (Gill et al. [14]).

The complementarity and superbasic pivoting procedures are fairly straight forward to implement. However, we would like to the emphasize few issues. Step 1 in both the complementarity and the superbasic pivoting procedures is to choose an initial basis. Obviously, it is desirable that the initial basis is an accurate estimate to an optimal basis. Thus, we like to choose initial basic variables from $P^k$. Furthermore, we also like to make the initial basis numerically stable. Hence, our initial basic variables are chosen from $P^k$ and $V$ (the set of artificial variables), which is similar to that described in [8]. In our implementation, we have not scaled LP problems as it has been done in [8]. Scaling might improve the efficiency and stability of purification procedures further.

In both the primal and dual phases, we choose the pivot using P.M.J. Harris's idea to enlarge the pool of possible pivots, see Harris [18] and Nazareth [34, pp. 142-143]. For difficult problems we found that this strategy is important. During the dual phase we need to compute the reduced cost, where we have used a trick described by Bixby, see [6, p. 17]. We use $LU$ factorization to represent the basis inverse. Our factorization is a modified version of Suhl and Suhl ([35]), and the $LU$ update is a variant of Bartels-Golub ([34, p. 118]).

The interior-point code we use is an implementation of the primal-dual barrier method. The method contains Mehrotra's predictor-corrector technique [28] and Kojima et al.'s global convergence safeguard, see [22] and [25]. Define

$$pval^k = \frac{\min(x_j^k s_j^k)}{(x^k)^T s^k / n} \tag{33}$$

which is a measure of closeness to the central path. Lustig et al. keep this value higher than

$$\theta_C = 1.0e - 8. \tag{34}$$

Because we need high accuracy to determine $LP(A, b^k, c^k)$, we use instead $\theta_C = 1.0e - 6$. This value may restrict the primal and dual step-length too much, if the choice of step-length is chosen as in [25, pp. 126-127]. To overcome this restriction, we calculate the primal and dual step-length as follows. Let $\alpha_P^{max}$ and $\alpha_D^{max}$ be the maximum possible step-lengths along the primal and dual directions $(\Delta x, \Delta s)$, respectively. Then, we choose maximal $\gamma^{max}$ such that

$$\frac{\min(x_j^k + \gamma^{max}\alpha_P^{max}\Delta x_j)(s_j^k + \gamma^{max}\alpha_D^{max}\Delta s_j)}{(x^k + \gamma^{max}\alpha_P^{max}\Delta x)^T(s^k + \gamma^{max}\alpha_D^{max}\Delta s)/n} \geq \theta_C, \ \forall\, j, \tag{35}$$

and then select the primal and dual step-length given by

$$(\alpha_P, \alpha_D) = \min(\gamma^{\max}, 0.99995)(\alpha_P^{\max}, \alpha_D^{\max}). \tag{36}$$

Using this method we find that a relatively large $\theta_C$ rarely hurts the speed of convergence. Note that finding $\gamma^{max}$ reduces to solving $n$ quadratic inequalities in one variable, because $\alpha_P^{max}$ and $\alpha_D^{max}$ are fixed scalars.

The most expensive operation in interior-point algorithms is the computation of a least-squares problem in each iteration. We solve the problem using a supernodal (push) Cholesky decomposition and an ordering that is obtained with the multiple minimum-degree heuristic, see Andersen et al. [4] for details. Dense columns in the constraint matrix are handled by a technique discussed in Andersen [5], and all test problems are presolved by a presolver described in [3, 2].

# 6 Computational results

We now present computational results for our combined approach. Our source code is written in ANSI C and is compiled on a HP 715/75 workstation using the HP C compiler and +O3 optimization option. The workstation has 64 MB RAM. All running times are obtained with the C procedure "clock" which measures CPU time in seconds (CPU sec).

We set parameters

$$\beta^1 = 10^{-1}, \quad \beta^2 = 10^{-8} \text{ and } \beta^3 = 10^{-12}.$$

Our first set of test problems is the set of NETLIB problems ([13]). We have tried both the complementarity and superbasic pivoting procedures, and found that the latter is much faster than the former in general. Therefore, we will report results based on using the superbasic pivoting procedure. In this test we try purification once using the criteria outlined in the preceding section.

Our results are shown in Tables 1 and 2. The column RGAP shows the relative duality gap of the interior-point iterate and ITE shows the number of interior-point iterations, right before the purification procedure starts. PBI and DBI represent the maximal primal and dual bound infeasibility errors, respectively. For example, PBI is calculated as:

$$x_B = B^{-1}b$$

and

$$\text{PBI } = \max_j \{0, -x_j\}$$

where $B$, generated by the purification procedure, is an optimal basis to the approximate problem, and $b$ is the right hand side of the original problem. (DBI is defined in the same manner for the dual problem.) Therefore, PBI and DBI show how good the basis is to the original problem. Clearly, if PBI and DBI are zero, then $B$ is an exact optimal basis for the original problem. TOPT is the total CPU time spent in interior-point iterations and purification pivots. TCRO is the total CPU time spent in the purification procedure. Thus, TCRO is included in TOPT. Finally, RATIO shows the ratio TCRO/TOPT.

From Tables 1 and 2 we see that an optimal basis to the original problem, with very low PBI and DBI, is obtained for each of the test problems. PBI and DBI are usually below $10^{-10}$, which is very good. Other accuracy measures are even better. Only in few cases, namely DFL001, NESM, ETAMACRO, PILOT.WE, SCSD6, we achieve an accuracy less than $10^{-8}$.

| | Name | RGAP | ITE | PBI | DBI | TOPT | TCRO | RATIO |
|---|---|---|---|---|---|---|---|---|
| 1 | 25FV47 | 6.2E-09 | 25 | 0.0E+00 | 3.1E-13 | 8.27 | 0.45 | 0.05 |
| 2 | 80BAU3B | 8.5E-14 | 43 | 8.3E-17 | 2.8E-14 | 33.54 | 0.75 | 0.02 |
| 3 | ADLITTLE | 9.5E-10 | 12 | 0.0E+00 | 1.5E-12 | 0.13 | 0.02 | 0.15 |
| 4 | AFIRO | 9.3E-11 | 9 | 0.0E+00 | 5.6E-17 | 0.05 | 0.01 | 0.20 |
| 5 | AGG | 1.2E-10 | 21 | 0.0E+00 | 6.0E-12 | 0.61 | 0.03 | 0.05 |
| 6 | AGG2 | 4.9E-11 | 18 | 0.0E+00 | 1.2E-12 | 1.31 | 0.06 | 0.05 |
| 7 | AGG3 | 3.2E-09 | 17 | 0.0E+00 | 1.2E-12 | 1.27 | 0.04 | 0.03 |
| 8 | BANDM | 3.0E-10 | 17 | 0.0E+00 | 2.4E-13 | 0.79 | 0.07 | 0.09 |
| 9 | BEACONFD | 1.5E-12 | 9 | 0.0E+00 | 1.3E-15 | 0.28 | 0.02 | 0.07 |
| 10 | BLEND | 5.9E-11 | 11 | 0.0E+00 | 1.3E-15 | 0.15 | 0.02 | 0.13 |
| 11 | BNL1 | 1.2E-11 | 27 | 1.8E-14 | 2.7E-13 | 3.78 | 0.43 | 0.11 |
| 12 | BNL2 | 5.0E-12 | 37 | 2.1E-14 | 9.7E-15 | 30.46 | 1.75 | 0.06 |
| 13 | BOEING1 | 5.1E-10 | 25 | 0.0E+00 | 7.1E-15 | 1.54 | 0.08 | 0.05 |
| 14 | BOEING2 | 2.1E-10 | 17 | 0.0E+00 | 3.6E-15 | 0.37 | 0.02 | 0.05 |
| 15 | BORE3D | 1.6E-10 | 17 | 0.0E+00 | 7.1E-15 | 0.32 | 0.02 | 0.06 |
| 16 | BRANDY | 1.0E-11 | 17 | 0.0E+00 | 5.7E-14 | 0.59 | 0.06 | 0.10 |
| 17 | CAPRI | -3.2E-13 | 19 | 0.0E+00 | 1.4E-13 | 0.81 | 0.05 | 0.06 |
| 18 | CYCLE | 1.8E-11 | 36 | 1.1E-13 | 2.7E-16 | 17.64 | 1.98 | 0.11 |
| 19 | CZPROB | 1.7E-11 | 32 | 0.0E+00 | 5.7E-14 | 4.84 | 0.15 | 0.03 |
| 20 | D2Q06C | -7.2E-13 | 30 | 4.7E-14 | 2.0E-09 | 53.63 | 2.48 | 0.05 |
| 21 | D6CUBE | 4.5E-11 | 23 | 1.6E-13 | 4.8E-14 | 24.94 | 2.82 | 0.11 |
| 22 | DEGEN2 | 7.2E-09 | 12 | 2.2E-16 | 5.1E-15 | 2.68 | 0.68 | 0.25 |
| 23 | DEGEN3 | 2.6E-11 | 18 | 1.1E-15 | 7.8E-15 | 36.21 | 7.19 | 0.20 |
| 24 | DFL001 | 4.5E-10 | 44 | 6.5E-11 | 1.1E-07 | 3142.71 | 37.54 | 0.01 |
| 25 | E226 | 1.3E-11 | 22 | 0.0E+00 | 2.7E-15 | 0.81 | 0.05 | 0.06 |
| 26 | ETAMACRO | 1.6E-15 | 36 | 0.0E+00 | 3.0E-08 | 2.71 | 0.07 | 0.03 |
| 27 | FFFFF800 | 5.9E-13 | 31 | 0.0E+00 | 1.5E-11 | 3.14 | 0.08 | 0.03 |
| 28 | FINNIS | -1.8E-13 | 23 | 0.0E+00 | 1.1E-13 | 1.31 | 0.08 | 0.06 |
| 29 | FIT1D | 6.8E-09 | 22 | 0.0E+00 | 7.1E-15 | 2.44 | 0.05 | 0.02 |
| 30 | FIT1P | 1.6E-10 | 18 | 0.0E+00 | 7.3E-12 | 3.75 | 0.12 | 0.03 |
| 31 | FIT2D | 2.1E-13 | 22 | 0.0E+00 | 7.5E-14 | 41.95 | 0.49 | 0.01 |
| 32 | FIT2P | 5.2E-13 | 19 | 0.0E+00 | 1.8E-11 | 32.89 | 1.18 | 0.04 |
| 33 | FORPLAN | 2.6E-10 | 26 | 0.0E+00 | 2.6E-14 | 1.08 | 0.04 | 0.04 |
| 34 | GANGES | 3.7E-12 | 16 | 0.0E+00 | 8.9E-14 | 4.36 | 0.42 | 0.10 |
| 35 | GFRD-PNC | 3.2E-09 | 18 | 0.0E+00 | 1.5E-11 | 1.27 | 0.10 | 0.08 |
| 36 | GREENBEA | 1.9E-10 | 42 | 9.0E-14 | 3.1E-10 | 33.66 | 1.49 | 0.04 |
| 37 | GREENBEB | 3.2E-13 | 41 | 4.2E-12 | 3.6E-13 | 28.73 | 1.71 | 0.06 |
| 38 | GROW15 | 1.9E-10 | 15 | 0.0E+00 | 1.9E-09 | 2.27 | 0.54 | 0.24 |
| 39 | GROW22 | 1.4E-10 | 16 | 0.0E+00 | 9.1E-09 | 4.47 | 1.70 | 0.38 |
| 40 | GROW7 | 2.8E-09 | 14 | 0.0E+00 | 3.7E-14 | 0.85 | 0.17 | 0.20 |
| 41 | ISRAEL | 1.2E-13 | 25 | 0.0E+00 | 2.0E-13 | 1.24 | 0.04 | 0.03 |
| 42 | KB2 | 6.4E-10 | 17 | 0.0E+00 | 7.1E-15 | 0.12 | 0.01 | 0.08 |
| 43 | LOTFI | 1.7E-11 | 20 | 2.1E-11 | 1.5E-16 | 0.37 | 0.02 | 0.05 |
| 44 | MAROS-R7 | 2.0E-11 | 13 | 0.0E+00 | 2.0E-10 | 112.91 | 3.21 | 0.03 |
| 45 | MAROS | 1.0E-10 | 26 | 3.5E-10 | 6.3E-12 | 4.16 | 0.26 | 0.06 |
| 46 | MODSZK1 | -2.3E-07 | 24 | 9.1E-13 | 5.7E-14 | 3.21 | 0.16 | 0.05 |
| Sum | | | 1042 | | | 3654.62 | 68.71 | |

Table 1: Results for the NETLIB problems.

| | Name | RGAP | ITE | PBI | DBI | TOPT | TCRO | RATIO |
|---|---|---|---|---|---|---|---|---|
| 45 | NESM | 1.6E-11 | 40 | 0.0E+00 | 2.0E-08 | 11.33 | 0.34 | 0.03 |
| 46 | PEROLD | 3.5E-13 | 37 | 1.3E-09 | 2.7E-11 | 7.92 | 0.60 | 0.08 |
| 47 | PILOT | -7.1E-12 | 44 | 0.0E+00 | 1.0E-09 | 130.73 | 5.79 | 0.04 |
| 48 | PILOT.JA | 2.0E-12 | 38 | 1.4E-14 | 1.5E-09 | 19.59 | 2.00 | 0.10 |
| 49 | PILOT.WE | -2.0E-13 | 43 | 0.0E+00 | 1.7E-07 | 10.06 | 0.46 | 0.05 |
| 50 | PILOT4 | 2.3E-12 | 39 | 0.0E+00 | 4.4E-10 | 5.83 | 0.30 | 0.05 |
| 51 | PILOT87 | 2.2E-14 | 50 | 0.0E+00 | 1.8E-08 | 493.85 | 22.98 | 0.05 |
| 52 | PILOTNOV | 1.9E-09 | 23 | 2.9E-12 | 4.4E-16 | 14.60 | 2.86 | 0.20 |
| 53 | RECIPE | 1.4E-09 | 10 | 0.0E+00 | 2.2E-16 | 0.13 | 0.01 | 0.08 |
| 54 | SC105 | 1.4E-09 | 11 | 0.0E+00 | 1.1E-16 | 0.15 | 0.02 | 0.13 |
| 55 | SC205 | 1.5E-11 | 13 | 0.0E+00 | 2.5E-16 | 0.30 | 0.04 | 0.13 |
| 56 | SC50A | 8.6E-13 | 10 | 0.0E+00 | 1.7E-16 | 0.08 | 0.01 | 0.13 |
| 57 | SC50B | 6.7E-09 | 8 | 0.0E+00 | 2.8E-17 | 0.07 | 0.01 | 0.14 |
| 58 | SCAGR25 | 1.7E-10 | 20 | 0.0E+00 | 1.6E-12 | 0.99 | 0.06 | 0.06 |
| 59 | SCAGR7 | 8.1E-12 | 16 | 0.0E+00 | 2.7E-13 | 0.20 | 0.01 | 0.05 |
| 60 | SCFXM1 | 8.7E-13 | 18 | 0.0E+00 | 1.7E-14 | 0.93 | 0.05 | 0.05 |
| 61 | SCFXM2 | 5.0E-09 | 18 | 0.0E+00 | 1.9E-14 | 2.05 | 0.13 | 0.06 |
| 62 | SCFXM3 | 3.2E-10 | 19 | 0.0E+00 | 2.0E-14 | 3.44 | 0.22 | 0.06 |
| 63 | SCORPION | 7.8E-09 | 10 | 0.0E+00 | 3.4E-13 | 0.44 | 0.05 | 0.11 |
| 64 | SCRS8 | 1.2E-09 | 20 | 0.0E+00 | 9.1E-13 | 1.64 | 0.12 | 0.07 |
| 65 | SCSD1 | 7.2E-09 | 9 | 0.0E+00 | 2.6E-09 | 0.40 | 0.04 | 0.10 |
| 66 | SCSD6 | 2.5E-10 | 11 | 1.9E-16 | 2.0E-08 | 0.90 | 0.09 | 0.10 |
| 67 | SCSD8 | 2.0E-10 | 10 | 0.0E+00 | 3.5E-13 | 2.19 | 0.45 | 0.21 |
| 68 | SCTAP1 | 5.7E-11 | 15 | 0.0E+00 | 8.5E-14 | 0.69 | 0.11 | 0.16 |
| 69 | SCTAP2 | 3.9E-09 | 13 | 0.0E+00 | 1.4E-14 | 2.92 | 0.26 | 0.09 |
| 70 | SCTAP3 | 2.4E-09 | 15 | 0.0E+00 | 3.6E-15 | 4.46 | 0.34 | 0.08 |
| 71 | SEBA | 1.3E-11 | 7 | 0.0E+00 | 0.0E+00 | 0.44 | 0.03 | 0.07 |
| 72 | SHARE1B | 2.6E-11 | 23 | 0.0E+00 | 1.2E-13 | 0.40 | 0.03 | 0.08 |
| 73 | SHARE2B | 4.5E-12 | 12 | 0.0E+00 | 5.7E-14 | 0.21 | 0.02 | 0.10 |
| 74 | SHELL | 9.3E-12 | 23 | 0.0E+00 | 0.0E+00 | 2.04 | 0.09 | 0.04 |
| 75 | SHIP04L | 1.5E-10 | 14 | 0.0E+00 | 6.8E-13 | 1.42 | 0.08 | 0.06 |
| 76 | SHIP04S | 1.9E-11 | 15 | 2.8E-14 | 4.5E-13 | 0.99 | 0.06 | 0.06 |
| 77 | SHIP08L | 5.5E-11 | 17 | 4.4E-16 | 1.3E-12 | 3.32 | 0.18 | 0.05 |
| 78 | SHIP08S | 5.8E-11 | 15 | 3.6E-15 | 9.1E-13 | 1.70 | 0.11 | 0.06 |
| 79 | SHIP12L | 1.8E-09 | 19 | 2.8E-15 | 5.5E-13 | 5.00 | 0.27 | 0.05 |
| 80 | SHIP12S | 6.8E-11 | 18 | 4.6E-15 | 5.5E-13 | 2.32 | 0.16 | 0.07 |
| 81 | SIERRA | 9.3E-10 | 18 | 2.3E-13 | 7.3E-12 | 4.19 | 0.21 | 0.05 |
| 82 | STAIR | 5.1E-12 | 17 | 0.0E+00 | 4.7E-15 | 2.45 | 0.29 | 0.12 |
| 83 | STANDATA | 5.8E-13 | 15 | 0.0E+00 | 0.0E+00 | 0.81 | 0.06 | 0.07 |
| 84 | STANDMPS | 4.6E-10 | 22 | 0.0E+00 | 1.8E-15 | 1.51 | 0.10 | 0.07 |
| 85 | STOCFOR1 | 3.6E-10 | 15 | 1.3E-17 | 5.7E-13 | 0.19 | 0.01 | 0.05 |
| 86 | STOCFOR2 | 1.2E-12 | 21 | 1.8E-15 | 2.1E-13 | 7.38 | 0.73 | 0.10 |
| 87 | STOCHFOR | 6.9E-11 | 33 | 3.2E-14 | 6.4E-14 | 128.72 | 32.65 | 0.25 |
| 88 | TRUSS | 2.6E-12 | 18 | 5.7E-14 | 7.9E-12 | 16.38 | 1.91 | 0.12 |
| 89 | TUFF | 2.8E-10 | 23 | 0.0E+00 | 1.7E-17 | 1.72 | 0.09 | 0.05 |
| 90 | VTP.BASE | 5.6E-11 | 11 | 0.0E+00 | 2.3E-13 | 0.15 | 0.01 | 0.07 |
| 91 | WOOD1P | 2.3E-09 | 14 | 1.3E-16 | 1.7E-13 | 11.39 | 0.50 | 0.04 |
| 92 | WOODW | 4.6E-10 | 23 | 2.3E-18 | 3.6E-15 | 11.41 | 0.62 | 0.05 |
| Sum | | | 953 | | | 920.03 | 75.56 | |
| Total | | | 1995 | | | 4574.65 | 144.27 | |

Table 2: Results for the NETLIB problems continued.

| Name | ROWS | COLS | RROWS | RCOLS | $\|P\|$ | $\|P \cap B\|$ | $\|Z\|$ | $\|Z \cap B\|$ |
|---|---|---|---|---|---|---|---|---|
| CRE-A | 3517 | 4067 | 2781 | 3938 | 822 | 701 | 5662 | 2080 |
| CRE-B | 9649 | 72447 | 5316 | 31818 | 901 | 751 | 35469 | 4565 |
| CRE-C | 3069 | 3678 | 2323 | 3249 | 823 | 719 | 4542 | 1604 |
| CRE-D | 8927 | 69980 | 4081 | 25188 | 933 | 743 | 27655 | 3338 |
| KEN-07 | 2427 | 3602 | 1378 | 2603 | 1236 | 1235 | 1367 | 143 |
| KEN-11 | 14695 | 21349 | 9946 | 16715 | 7182 | 7182 | 9533 | 2764 |
| KEN-13 | 28633 | 42659 | 22356 | 36552 | 15395 | 15394 | 21157 | 6962 |
| OSA-07 | 1119 | 23949 | 1047 | 23015 | 1049 | 1042 | 23013 | 5 |
| OSA-14 | 2338 | 52460 | 2266 | 50457 | 2345 | 2261 | 50378 | 5 |
| OSA-30 | 4351 | 100024 | 4279 | 96119 | 4440 | 4274 | 95958 | 5 |
| PDS-02 | 2954 | 7535 | 2594 | 7172 | 1617 | 1078 | 5718 | 1516 |
| PDS-06 | 9882 | 28655 | 9121 | 27852 | 5415 | 3736 | 23033 | 5385 |
| PDS-10 | 16559 | 48763 | 15598 | 47729 | 8104 | 5872 | 40637 | 9726 |
| Sum | 108120 | 479168 | 83086 | 372407 | | | | |

Table 3: The Kennington problems.

In addition to accuracy, the time spent in the purification procedure is important. Again from Tables 1 and 2 we see that $TCRO$ is less than 10 % of the total solution time $TOPT$. It is more than 20% in very few cases, where these problems are either very primal or dual degenerate, or their bases are dense and expensive to factor.

NETLIB problems are relative small and may not really represent today's large-scale LP problems. Our second set of test problems is the so-called Kennington problems, see Carolan et al. [9]. (We have excluded three largest problems, KEN-18, OSA-60 and PDS-20, in our test because they require memory beyond the capacity of our workstation.) Table 3 shows some problem related statistics.

The columns ROWS and COLS show the numbers of rows and columns before presolving. RROWS and RCOLS show the numbers of rows and columns after presolving. $|P|$ and $|Z|$ show the numbers of variables in $P$ and $Z$, respectively, in the optimal partition. $|P \cap B|$ and $|Z \cap B|$ show how many variables from $P$ and $Z$ included in the final optimal basis. An important observation from Table 3 is that the optimal basic solution is typically highly degenerate. This can be observed from the fact that a large portion, up to 30% to 50%, of the basic variables come from $Z$. We should therefore expect that the dual purification phase is quite expensive for solving these problems. Table 4 shows the computational results for solving the Kennington problems. We see once again that measures PBI and DBI are very good. Although the problems are highly degenerate, the purification is done in less than 15% of the total solution time.

Finally in Table 5 we have compared our purification procedure to the cross-over technique implemented in the state of the art LP code CPLEX ([6]). CPLEX results were produced by R.E. Bixby on our HP 715 workstation in April 1995 using the most recent development code. CROWS and CCOLS show the numbers of rows and columns after CPLEX's presolve, and CTIME is the cross-over time reported by CPLEX. AROWS and ACOLS show the numbers of rows and columns

| Name | RGAP | ITE | PBI | DBI | TOPT | TCRO | RATIO |
|---|---|---|---|---|---|---|---|
| CRE-A | 1.1E-09 | 34 | 1.4E-12 | 5.8E-11 | 21.19 | 1.98 | 0.09 |
| CRE-B | 2.4E-13 | 46 | 4.2E-14 | 5.8E-11 | 291.11 | 15.84 | 0.05 |
| CRE-C | 4.4E-11 | 30 | 2.8E-14 | 1.4E-10 | 16.30 | 1.48 | 0.09 |
| CRE-D | 2.2E-12 | 46 | 1.1E-13 | 4.4E-11 | 248.29 | 10.19 | 0.04 |
| KEN-07 | 2.0E-09 | 13 | 0.0E+00 | 3.6E-12 | 4.24 | 0.33 | 0.08 |
| KEN-11 | 3.6E-09 | 22 | 0.0E+00 | 3.6E-12 | 63.95 | 5.86 | 0.09 |
| KEN-13 | 5.5E-12 | 25 | 0.0E+00 | 3.6E-12 | 192.09 | 28.63 | 0.15 |
| OSA-07 | 1.6E-11 | 17 | 0.0E+00 | 6.8E-13 | 32.33 | 0.73 | 0.02 |
| OSA-14 | -2.9E-14 | 20 | 0.0E+00 | 9.1E-13 | 84.62 | 2.36 | 0.03 |
| OSA-30 | -2.1E-14 | 22 | 0.0E+00 | 8.0E-13 | 183.97 | 8.36 | 0.05 |
| PDS-02 | 4.2E-12 | 29 | 0.0E+00 | 0.0E+00 | 22.66 | 1.99 | 0.09 |
| PDS-06 | 8.3E-13 | 38 | 0.0E+00 | 0.0E+00 | 507.41 | 28.24 | 0.06 |
| PDS-10 | 4.4E-10 | 46 | 0.0E+00 | 0.0E+00 | 2953.51 | 79.16 | 0.03 |
| Sum | | 388 | | | 4621.67 | 185.15 | 0.04 |

Table 4: Results for the Kennington problems.

| Name | CROWS | CCOLS | CTIME | AROWS | ACOLS | ATIME |
|---|---|---|---|---|---|---|
| PILOT87 | 1890 | 4511 | 17.30 | 1966 | 4592 | 22.98 |
| CRE-B | 5529 | 31723 | 10.30 | 5316 | 31818 | 15.84 |
| CRE-D | 4027 | 25126 | 7.00 | 4081 | 25188 | 10.19 |
| KEN-13 | 11024 | 24900 | 10.40 | 22356 | 36552 | 28.63 |
| PDS-06 | 7141 | 25861 | 9.80 | 9121 | 27852 | 28.24 |
| STOCHFOR | 10740 | 9786 | 10.40 | 15336 | 14382 | 32.65 |

Table 5: A comparison between CPLEX and APOS.

after our presolve, and ATIME is the purification time of our code. Considering the difference in problem sizes after presolve, it seems that CPLEX and our code produce optimal bases in a compatible amount of time.

# 7 Conclusions

We have in this paper proposed a new approach to combine interior-point and simplex pivoting algorithms. In any iteration of an interior-point algorithm we construct a related LP problem, which approximates the original problem, with a known (strictly) complementary primal-dual solution pair. Thus, we can apply several purification pivoting procedures to compute an optimal basis for the approximate problem in strongly polynomial time. Theoretically, we have shown that, if the approximate problem is constructed from an interior-point iterate sufficiently close to the optimal face, then any optimal basis of the approximate problem is an optimal basis for the original problem. Moreover, if the LP data are rational, the total number of interior-point iterations to create such a sufficient approximate problem is bounded by a polynomial in the data size.

Practically, we have developed a modification of Megiddo's procedure and discussed several implementation issues in solving the approximate problem. We have also reported computational results for our combined method, which indicate that the method works well in practice.

An interesting topic for future research is whether or not the method could be used to do warm-start for an interior-point LP algorithm. One possibility is to iteratively solve the least-squares problem of interior-point algorithms, using the optimal basis generated from the purification procedure as a preconditioner.

# References


[1] I. Adler and R.D.C. Monteiro. A geometric view of parametric linear programming. *Algorithmica*, 8(2):161–176, 1992.

[2] E. D. Andersen. Finding all linearly dependent rows in large-scale linear programming. Technical report, 1994. to appear Optimization Methods and Software.

[3] E.D. Andersen and K.D. Andersen. Presolving in Linear Programming. Preprint 35, Dept. of Math. and Computer Sci., Odense University, 1993. To appear in Math. Programming.

[4] E.D. Andersen, K.D. Andersen, and X. Xu. Improved Cholesky decompositions for work-stations and vector computers. Technical report, 1994. In preparation.

[5] K.D. Andersen. A modified Schur complement method for handling dense columns in interior point methods for linear programming . Technical report, Odense University, 1994. Submitted to ACM Transaction on Mathematical Software.

[6] R. E. Bixby. Progress in linear programming. *ORSA J. on Comput.*, 6(1):15–22, 1994.

[7] R. E. Bixby, J. W. Gregory, I. J. Lustig, R. E. Marsten, and D. F. Shanno. Very large-scale linear programming: A case study in combining interior point and simplex methods. *Oper. Res.*, 40(5):885–897, 1992.

[8] R.E. Bixby and M.J. Saltzman. Recovering an optimal basis from an interior point solution. *Oper. Res. Lett.*, 15(4):169–178, 1993.

[9] W.J. Carolan, J.E. Hill, J.L. Kennington, S. Niemi, and S.J. Wichman. An empirical evaluation of the KORBX algorithms for military airlift applications. *Oper. Res.*, 38(2):240–248, 1990.

[10] A. Charnes and K.O. Kortanek. An opposite sign algorithm for purification to an extreme point solution, 1963. Memorandum No. 89. Office of Naval Research.

[11] E. Christiansen and K.O. Kortanek. Computation of the collapse state in limit analysis using the LP primal affine scaling algorithm. *J. Comput. Appl. Math.*, 34:47–63, 1991.

[12] A.S. El-Bakry, R.A. Tapia, and Y. Zhang. A study of idicators for identifying zero variables in interior-point methods. *SIAM Rev.*, 36(1):45–72, 1994.

[13] D.M. Gay. Electronic mail distribution of linear programming test problems. *COAL Newsletter*, 13:10–12, 1985.

[14] P.E. Gill, W. Murray, D.B. Ponceleón, and M.A. Saunders. Preconditioners for indefinite systems arising in optimization. *SIAM J. Mat. Anal. Appl.*, 13:292 – 311, 1992.

[15] A.J. Goldman and A.W. Tucker. Theory of linear programming. In H.W. Kuhn and A.W. Tucker, editors, *Linear Inequalities and related Systems*, pages 53–97, Princeton, New Jersey, 1956. Princeton University Press.

[16] H.J. Greenberg. The use of the optimal partition in a linear programming solution for postoptimal analysis. *Oper. Res. Lett.*, 15(4):179–186, 1994.

[17] O. Güler and Y. Ye. Convergence behaviour of interior-point algorithms. *Math. Programming*, 60(2):215–228, 1993.

[18] P.M.J. Harris. Pivot selection methods in the Devex LP code. *Math. Programming Study*, 4:30–57, 1975.

[19] B. Jansen, C. Roos, and T. Terlaky. An interior point approach to postoptimal and parametric analysis in linear programming. In *Interior point methods.* Eövös Loránd University, Department of Operations Research, H-1088 Budapest, Múzeum krt. 6-8., Hungary, 1992.

[20] B. Jansen, T. Terlaky, and C. Roos. The theory of linear programming: skew symmetric self-dual problems and the central path. *Optimization*, 29:225–233, 1994.

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

[22] M. Kojima, N. Megiddo, and S. Mizuno. A primal-dual infeasible-interior-point algorithm for linear programming. *Math. Programming*, 61:263–280, 1993.

[23] M. Kojima, S. Mizuno, and A. Yoshise. A polynomial-time algorithm for a class of linear complementarity problems. *Math. Programming*, 44:1–26, 1989.

[24] K.O. Kortanek and J. Zhu. New purification algorithms for linear programming. *Naval Res. Logist. Quart.*, 35:571–583, 1988.

[25] I. J. Lustig, R. E. Marsten, and D. F. Shanno. Computational experience with a globally convergent primal-dual predictor-corrector algorithm for linear programming. *Math. Programming*, 66(1):123–135, 1994.

[26] I. J. Lustig, R. E. Marsten, and D. F. Shanno. Interior point methods for linear programming: Computational state of the art. *ORSA J. on Comput.*, 6(1):1–15, 1994.

[27] N. Megiddo. On finding primal- and dual- optimal bases. *ORSA J. on Comput.*, 3(1):63–65, 1991.

[28] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. on Optim.*, 2(4):575–601, 1992.

[29] S. Mehrotra. Quadratic convergence in primal-dual methods. *Math. Oper. Res.*, 18(3):741–751, 1993.

[30] S. Mehrotra and Y. Ye. Finding an interior point in the optimal face of linear programs. *Math. Programming*, 62(3):497–515, 1993.

[31] S. Mizuno, M.J. Todd, and Y. Ye. On adaptive-step primal-dual interior-point algorithms for linear programming. *Math. Oper. Res.*, 18:964–981, 1994.

[32] S. Mizuno, M.J. Todd, and Y. Ye. A surface of analytic centers and primal-dual infeasible-interior-point algorithms for linear programming. *Math. Oper. Res.*, 20(1):135–162, 1995.

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

[34] J.L. Nazareth. *Computer solutions of Linear Programs.* Oxford University Press, New York, 1987.

[35] U.H. Suhl and L.M. Suhl. Computing sparse LU factorizations for large-scale linear programming bases. *ORSA J. on Comput.*, 2(4):325–335, 1990.

[36] Y. Ye. On the finite convergence of interior-point algorithms for linear programming. *Math. Programming*, 57:325–335, 1992.

[37] Y. Ye, O. Güler, R.A. Tapia, and Y. Zhang. A quadratically convergent $O(\sqrt{n}L)$-iteration algorithm for linear programming. *Math. Programming*, 59(2):151–162, 1993.

[38] Y. Ye, M.J. Todd, and S. Mizuno. An $O(\sqrt{n}L)$ - iteration homogeneous and self-dual linear programming algorithm. *Math. Oper. Res.*, 19:53–67, 1994.