A NOTE ON
NONLINEAR APPROACHES TO LINEAR PROGRAMMING

by

Philip E. Gill, Walter Murray,
Michael A. Saunders and Margaret H. Wright

TECHNICAL REPORT SOL 86–7

April 1986
A NOTE ON
NONLINEAR APPROACHES TO LINEAR PROGRAMMING
by
Philip E. Gill, Walter Murray,
Michael A. Saunders and Margaret H. Wright
TECHNICAL REPORT SOL 86–7
April 1986

Research and reproduction of this report were partially supported by the
U.S. Department of Energy Contract DE-AA03-76SF00326, PA# DE-AS03-76ER72018;

Any opinions, findings, and conclusions or recommendations expressed in this
publication are those of the author(s) and do NOT necessarily reflect the
views of the above sponsors.

Reproduction in whole or in part is permitted for any purposes of the United
States Government. This document has been approved for public release and
sale; its distribution is unlimited.
A NOTE ON
NONLINEAR APPROACHES TO LINEAR PROGRAMMING

Philip E. Gill, Walter Murray,
Michael A. Saunders and Margaret H. Wright

Systems Optimization Laboratory
Department of Operations Research
Stanford University
Stanford, California 94305

Technical Report SOL 86-7
April 1986

ABSTRACT

Certain new approaches to linear programming have recently received considerable publicity because of the promise of substantial improvements in efficiency compared to the simplex method. This note briefly discusses several research directions in methods for solving linear programs using nonlinear problem transformations. In particular, we describe application of a barrier transformation to the dual, and the development of sparse least-squares methods based on the LU factorization of the least-squares matrix or its transpose.

The material contained in this report is based upon research supported by the U.S. Department of Energy Contract DE-AA03-76SF00326, PA No. DE-AS03-76ER72018; and the Office of Naval Research Contract N00014-85-K-0343.

† Presented at the Workshop on Iterative Methods for Mathematical Programs, Stanford University, April 17-18, 1986.
1. Introduction

Linear programming (LP) has been the subject of great interest recently because of publicity surrounding an algorithm (Karmarkar, 1984) that is not only polynomial in complexity, but also is claimed to be much faster than the simplex method, developed nearly 40 years ago by George B. Dantzig. Since the publication of Karmarkar's original algorithm, interest has continued to grow in alternatives to the simplex method (see, e.g., Gay, 1985; Goldfarb and Mehrotra, 1985; Osborne, 1986; Shanno and Marsten, 1985; Todd and Durrell, 1985; Tomlin, 1985; Vanderbei, Meketon and Freedman, 1985). In this connection, Gill et al. (1985) have shown that Karmarkar's projective method is closely related to the classical logarithmic barrier function. In particular, Karmarkar's projective method is a special case of a projected Newton method applied to the logarithmic barrier function. Based on this equivalence, many researchers are re-examining the effectiveness of once-discarded methods that in effect "nonlinearize" a linear program. The proof of polynomial complexity of Karmarkar's original algorithm suggests that a suitable nonlinear transformation may somehow overcome the inherently combinatorial nature of the simplex method.

Because the main interest in nonlinear LP techniques arises from their potential speed, an essential part of the work reported in Gill et al. (1985) involved numerical experiments with an initial implementation of the barrier method on a set of small- to medium-scale test problems. The results were encouraging in that the barrier method was comparable in speed to the simplex method on certain problems. Typically, a barrier algorithm converges in a smaller number of iterations than the simplex method. However, since each iteration of the barrier method requires solution of a linear least-squares subproblem, a barrier method will be faster than the simplex method only if these subproblems can be solved quickly. Thus, the hope is that barrier-type methods will be faster than the simplex method—even substantially so—for problems with structure that allows fast solution of the least-squares subproblems.

The remainder of this note is organized as follows. In Section 2, we summarize some background on the application of a logarithmic barrier function method to a linear program in standard form. A disadvantage of the primal formulation is that the search direction is restricted to lie in a subspace. In order to satisfy this requirement, either the associated linear least-squares subproblem must be solved accurately, or the vector must be projected into the appropriate subspace. As an alternative, Section 3 gives the dual formulation of an LP, which leads to a purely unconstrained subproblem, thereby allowing approximate methods to be used to solve the least-squares subproblem. In Section 4, we discuss techniques for solving the least-squares subproblem, and consider possible uses of the LU factorization.

2. A Barrier-Function Method for the Primal LP

This section is intended to serve as general background on barrier functions and on the work described in Gill et al. (1985).

Barrier-function methods treat inequality constraints by creating a barrier function, which is a combination of the original objective function and a weighted sum of functions with a positive
singularity at the constraint boundary. (Many barrier functions have been proposed; we consider only the logarithmic barrier function, first suggested by Frisch, 1955.) Barrier-function methods require a strictly feasible starting point for each minimization, and generate a sequence of strictly feasible iterates. (For a complete discussion of barrier methods, see Fiacco, 1979; both barrier- and penalty-function methods are described in Fiacco and McCormick, 1968. Brief overviews are given in Fletcher, 1981, and Gill, Murray and Wright, 1981.)

In particular, consider the inequality-constrained problem

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad \phi(x) \\
\text{subject to} & \quad h(x) \geq 0,
\end{align*}
\]

where \( h \) is an \( t \)-vector of nonlinear functions. Let \( x^* \) denote the solution of (2.1). With a barrier-function approach, the inequality constraints of (2.1) are "removed" by including transformed versions in a modified objective function, as follows. Given a barrier parameter \( \mu (\mu > 0) \), let \( z(\mu) \) denote the unconstrained minimum of

\[
B(x, \mu) = \phi(x) - \mu \sum_{i=1}^t \ln(h_i(x)).
\]

Under mild conditions, it can be shown that

\[
\lim_{\mu \to 0} z(\mu) = x^*.
\]

Many results have been proved concerning the asymptotic properties of the sequence \( \{z(\mu)\} \) as \( \mu \to 0 \) (see, e.g., Mifflin, 1972a, b; Jittorntrum, 1978; Jittorntrum and Osborne, 1978).

Now consider applying a barrier-function method to the following linear program in standard form (which will also be called the primal LP):

\[
\begin{align*}
\minimize_{z \in \mathbb{R}^n} & \quad c^Tz \\
\text{subject to} & \quad Az = b \\
& \quad z \geq 0,
\end{align*}
\]

where \( A \) is an \( m \times n \) matrix with \( m \leq n \).

Since the barrier transformation may be applied only to the inequality constraints (2.2c), the subproblem associated with the primal LP treats the linear equality constraints directly and transforms only the bounds:

\[
\begin{align*}
\minimize_{z \in \mathbb{R}^n} & \quad F(z) = c^Tz - \mu \sum_{j=1}^n \ln z_j \\
\text{subject to} & \quad Az = b.
\end{align*}
\]

A standard approach to solving a linearly constrained problem of the form (2.3) is to use a feasible-point descent method (see, e.g., Gill, Murray and Wright, 1981). The current iterate \( z \) always satisfies \( Az = b \), and the next iterate \( \tilde{z} \) is defined as

\[
\tilde{z} = z + \alpha p,
\]
2. A Barrier-Function Method for the Primal LP

where \( p \) is an \( n \)-vector (the search direction), and \( \alpha \) is a positive scalar (the steplength). The computation of \( p \) and \( \alpha \) must ensure that \( A\hat{x} = b \) and \( F(\hat{x}) < F(x) \).

The Newton search direction is defined as the step to the minimum of the quadratic approximation to \( F(x) \) derived from the local Taylor series, subject to retaining feasibility. Thus, the Newton search direction is the solution of the following quadratic program:

\[
\begin{align*}
\text{minimize} \quad & g^Tp + \frac{1}{2}p^THp \\
\text{subject to} \quad & Ap = 0,
\end{align*}
\]

(2.5a)

(2.5b)

where \( g \equiv \nabla F(x) \) and \( H \equiv \nabla^2 F(x) \). The optimality conditions for (2.5) imply that

\[
g + Hp = A^T\pi
\]

(2.6)

for a vector \( \pi \) of Lagrange multipliers. Thus, \( p \) and \( \pi \) satisfy the partitioned linear system:

\[
\begin{pmatrix}
H & A^T \\
A & 0
\end{pmatrix}
\begin{pmatrix}
-p \\
\pi
\end{pmatrix} =
\begin{pmatrix}
g \\
0
\end{pmatrix}.
\]

(2.7)

If \( H \) is nonsingular, one method of solving (2.7) is to solve the equations

\[
AH^{-1}A^T\pi = AH^{-1}g,
\]

(2.8a)

\[
Hp = A^T\pi - g.
\]

(2.8b)

Let \( p_p \) (the Newton direction for the primal LP) denote the solution of subproblem (2.5) when \( F \) is the barrier function in (2.3), with associated Lagrange multiplier \( \pi_p \). The derivatives of \( F \) are:

\[
g(x) = c - \mu X^{-1}e \quad \text{and} \quad H(x) = \mu X^{-2},
\]

(2.9a)

where

\[
X = \text{diag}(x_j), \quad j = 1, \ldots, n,
\]

(2.9b)

and \( e = (1, 1, \ldots, 1)^T \). Since \( H(x) \) is positive definite when \( x > 0 \), \( p_p \) is finite and unique, and is a descent direction for \( F(x) \), i.e., \( (c - \mu X^{-1}e)^T p_p < 0 \). Substituting from (2.9a) in (2.8a), we see that \( \pi_p \) satisfies

\[
AX^2A^T\pi_p = AX(Xc - \mu e).
\]

(2.10)

Recall from standard linear algebra (e.g., Stewart, 1973) that, for any matrix \( C \), the solution of the least-squares problem

\[
\text{minimize} \quad \|f - Cz\|_2^2
\]

(2.11)

satisfies the normal equations

\[
C^T C z = C^T f,
\]

(2.12)

which are always compatible. In referring to (2.11) and (2.12), we shall call \( C \) the least-squares matrix and \( C^T C \) the normal-equation matrix.
Comparing (2.10) and (2.12) and taking $C = XA^T$ in (2.12), it follows that $\pi_p$ solves the least-squares problem

$$\minimize_{\pi} \| Xc - \mu e - XA^T\pi \|_2^2.$$  

(2.13)

The vector $p_p$ is then defined by

$$p_p = -(1/\mu)Xr_p,$$

(2.14)

where $r_p$ is the optimal residual of (2.13):

$$r_p = Xc - \mu e - XA^T\pi_p.$$  

(2.15)

Note that if $m$ variables are nonzero at the solution of (2.2) (i.e., if the primal is nondegenerate), then $XA^T$ will retain full rank even as the iterates approach the solution. (This contrasts with the structural ill-conditioning of the Hessian matrices of the barrier function when fewer than $n$ constraints are active at the solution; see, e.g., Murray, 1971.) However, a well known feature of practical linear programs is that they usually are degenerate; i.e., fewer than $m$ variables are nonzero at the solution. In this case, the matrix $XA^T$ is rank-deficient at the solution, and becomes increasingly ill-conditioned as $\mu \to 0$. Therefore, any method used in practice to solve (2.13) must be able to cope with singularity and extreme ill-conditioning. (Numerical solution of the least-squares subproblems will be discussed in Section 4.)

3. Solving the Dual Problem

With the approach described in Section 2, the search direction is required to lie in a particular subspace in order for the iterates to retain feasibility with respect to the equality constraints (2.2b). In this section, we describe an alternative based on the dual linear program associated with (2.2):

$$\minimize_{y \in \mathbb{R}^m} \ b^T y$$

subject to $A^T y \geq -c.$  

(3.1)

The optimal vector $y^*$ for (3.1) is the negative of the Lagrange multiplier vector $\pi^*$ for the equality constraints of (2.2). (Use of the dual LP in this context has been suggested independently by other researchers, for example at the recent ONR-sponsored Workshop on New Directions in Mathematical Programming, Monterey, California, February 20-21, 1986. It is also closely related to the work of Eriksson, 1981, 1985.)

Since all the constraints of (3.1) are inequalities, applying the barrier transformation to this problem gives a purely unconstrained subproblem:

$$\minimize_{y} \ F(y) = b^T y - \mu \sum_{j=1}^{n} \ln(a_j^T y + c_j).$$  

(3.2)

Let $y(\mu)$ denote the unconstrained minimum of (3.2). Defining

$$D = \text{diag}(d_j) \quad \text{and} \quad X = \text{diag}(x_j) = \mu D^{-1},$$

4
where \( d_j = a_j^Ty + c_j \) and \( z_j = \mu/d_j \), we have

\[
\nabla F = b - Az \quad \text{and} \quad \nabla^2 F = \frac{1}{\mu} AX^2 A^T.
\]

The gradient of \( F \) must vanish at \( y(\mu) \), which implies that

\[
\sum_{j=1}^{n} \frac{\mu}{a_j^Ty(\mu) + c_j} a_j = b,
\]

and hence

\[
\lim_{\mu \to 0} \frac{\mu}{d_j} = z_j^*.
\]

In order to define a Newton method for solving (3.2), we assume that a point \( y \) is available for which \( A^Ty + c > 0 \) (so that \( \nabla^2 F \) is positive definite). Using the above derivatives, \( p_d \) (the Newton search direction for the dual) then satisfies the system

\[
AX^2 A^T p_d = \mu(Az - b).
\]

In contrast to the subproblem (2.5) associated with the primal problem, \( p_d \) need not lie in any particular subspace, and hence (3.4) may be solved only approximately—for example, using a few iterations of a conjugate-gradient method.

Note that if the right-hand side of (3.4) can be expressed in the form \( AXv \) for some vector \( v \), (3.4) will have the form (2.12) of the normal equations for a least-squares problem. (It is advantageous numerically to solve a least-squares problem rather than equations of the form (3.4). However, note that (3.4) always has a bounded solution if a primal-feasible point exists.) We therefore seek \( v \) such that

\[
AXv = Az - b,
\]

whereupon \( p_d \) becomes the solution of the least-squares problem

\[
\minimize_p \| \mu v - XA^T p \|^2,
\]

which may be solved using the techniques to be described in Section 4. Note that \( z > 0 \), whereas \( A(z - Xv) = b \). As convergence occurs, \( v \to 0 \) and \( z \to z^* \).

It might appear that the elements of the least-squares matrix would become unbounded with this approach, in contrast to the primal. However, the relationship (3.3) shows that \( d_j \to 0 \) only as \( \mu \to 0 \), and thus the limiting matrices \( X \) and \( XA^T \) should remain bounded.

An alternative to the exact dual formulation (3.4) is to retain the same "Hessian" for several iterations. In Newton-type methods for nonlinear optimization, it is common to define the search direction with a positive-definite approximation to the Hessian matrix. (For example, the exact Hessian may be indefinite or expensive to compute.) In the present context, any positive-definite
Nonlinear Approaches to Linear Programming

X may be used in (3.4) (3.6) to generate a descent direction. It is essential to retain the definition $z = \mu D^{-1} e$ to obtain the correct gradient $b = A z$, but $X$ need not be defined as $\mu D^{-1}$.

An immediate implication is that the same matrix $X$ could be used for perhaps several iterations of the Newton-type method. (Convergence results for methods of this type are given in, e.g., Dennis, 1970, and Ortega and Rheinboldt, 1970.) Since a sparse factorization of $X A^T$ is needed to compute $v$ and $p$ in (3.5)–(3.6), considerable economies can be made by retaining the factorization from the previous iteration.

4. Solving the Least-Squares Subproblem

It should be emphasized that the least-squares subproblems associated with the primal and dual linear programs ((2.13) and (3.6)) are very similar, and it is crucial that they be solved efficiently. In this section we review techniques that have been suggested or used, and then describe some new approaches based on the LU factorization.

4.1. Background

The major strategies for solving sparse least-squares problems include:

- **direct methods**, based on a sparse QR factorization of $C$ (e.g., George and Heath, 1980) or on the Cholesky factorization of the normal-equation matrix $C^T C$ (e.g., George and Liu, 1981);
- **updating methods** that factorize most of $C$ or $C^T C$ and deal with a few omitted rows or columns by partitioning (e.g., Heath, 1984);
- purely **iterative methods** that perform no factorization, such as a straightforward conjugate-gradient method;
- **hybrid methods** that combine factorization and iterative techniques. For example, in Gill et al. (1985), the subproblem (2.13) is solved with a preconditioned version of the stabilized conjugate-gradient method LSQR (Paige and Saunders, 1982), where the preconditioner is the Cholesky factor of a sparse matrix related to $AX^2 A^T$. (The use of a preconditioned conjugate-gradient method in this context is also discussed by Gay, 1985.) For early work on preconditioners, see Axelsson (1974) and Concus, Golub and O'Leary (1976). Some techniques for computing an approximate ("incomplete") Cholesky factor based on sparsity considerations are given in, e.g., Meijerink and van der Vorst (1977), Manteuffel (1980) and Munksgaard (1980).

The effectiveness of these options varies with the problem.

When using an iterative scheme, it is desirable to be able to use only an approximate solution (obtained, say, from a truncated conjugate-gradient method; see, e.g., Dembo, Eisenstat and Steihaug, 1982). In this context, a disadvantage of the primal formulation is that an approximate solution of (2.13) will not satisfy the equality constraints (2.5b). One means of overcoming this difficulty is to project the search direction into the null space of $A$, for example using a sparse
representation of a basis for the null space. Such a matrix may be constructed from an LU factorization of $A$ (which would need to be computed only once), or alternatively it may be obtained from an LU factorization of $XAT$.

For the dual formulation, a truncated conjugate-gradient method is most promising because an approximate solution of (3.4) can be shown to be a descent direction, as required. However, the speed of convergence of the conjugate-gradient method is crucial to practical success. We now consider methods for preconditioning the conjugate-gradient method using an LU factorization.

4.2. The LU factorization

Given a general sparse matrix $C$ with $n$ rows and $m$ columns, we write

$$C = LU,$$  \hspace{1cm}  (4.1)

where $L$ is square and nonsingular, the dimensions of $U$ are those of $C$, and $L$ and $U$ are (nominally) unit-lower-triangular and upper-triangular respectively. (In general, the rows and columns of $C$ will have been permuted during the factorization, but we shall ignore the permutations for simplicity of notation.) If a suitable form of threshold pivoting (see Section 4.4) is used in computing $L$ and $U$, $L$ tends to remain well-conditioned throughout, so that the condition of $C$ is almost always reflected in $U$.

A set of procedures (LUSOL; see Gill et al., 1986) has recently been developed for computing and updating the $LU$ factors of a general sparse matrix; the techniques in LUSOL are related to the work of Reid (1976, 1982) on computing a sparse $LU$ factorization and performing sparse Bartels-Golub updates (Bartels and Golub, 1969). LUSOL provides a means for implementing several methods to be described below.

In the problems of interest, $C$ is of the form $XAT$ and $n > m$. Hence (4.1) has the form

$$C = XAT = LU = \hat{L}\hat{U}, \quad \hat{L} = L \begin{pmatrix} I \\ 0 \end{pmatrix}, \quad U = \begin{pmatrix} \hat{U} \\ 0 \end{pmatrix},$$  \hspace{1cm}  (4.2)

where $\hat{L}$ denotes the first $m$ columns of $L$ and $\hat{U}$ is an $m \times m$ upper-triangular matrix. (The last $(n - m)$ columns of $L$ will simply be those of the identity.) Again $\hat{L}$ should be well-conditioned, while $\hat{U}$ reflects the condition of $C$. For the present we assume that $C$ has full rank, so that $\hat{U}$ is nonsingular.

Note that (4.2) can be used to find the vector $v$ in (3.5) needed to formulate the dual as the least-squares subproblem (3.6). It is sufficient to solve the two systems

$$\hat{U}^Tu = A\hat{x} - b \quad \text{and} \quad L^Tv = \begin{pmatrix} u \\ 0 \end{pmatrix}.$$  

4.3. LU preconditioning

Following Peters and Wilkinson (1970) and Björck (1976), we note that the least-squares problem

$$\minimize \|f - Cp\|_2^2$$  \hspace{1cm}  (4.3)
may be solved using a factorization of the form (4.2), by solving

\[
\begin{align*}
\text{minimize} & \quad \|f - \hat{L}q\|_2^2, \\
\hat{U}p & = q.
\end{align*}
\]  

(4.4a)

(4.4b)

Because \( \hat{L} \) is well-conditioned, an iterative method may converge more rapidly on problem (4.4a) than on the original problem (4.3) (unless \( C \) itself is well-conditioned or has clustered singular values). Some experiments with this approach are described in Saunders (1979); for related work, see Delves and Barrodale (1979).

In the present context, greater efficiency is likely to result from further preconditioning, particularly as the solution is approached. If \( \tilde{L} \) is the first \( m \) rows of \( \hat{L} \):

\[
\tilde{L} = \begin{pmatrix} \hat{L} \\ M \end{pmatrix},
\]

it follows that \( p \) may be determined by solving

\[
\begin{align*}
\text{minimize} & \quad \|f - \begin{pmatrix} I \\ ML^{-1} \end{pmatrix} s\|_2^2, \\
\tilde{L}q & = s, \quad \tilde{U}p = q.
\end{align*}
\]  

(4.5a)

(4.5b)

The nature of the linear programming problem and the pivoting strategy used in LUSOL are such that \( ML^{-1} \to 0 \) as \( y \to y^* \). Hence, the number of iterations required by a conjugate-gradient algorithm should become small as the solution is approached.

In fact, the factorization \( XA^T = LU \) obtained with the LUSOL pivoting strategy serves to partition \( X \) and \( A \) in the form

\[
X A^T = \begin{pmatrix} X_B \\ X_N \end{pmatrix} \begin{pmatrix} B^T \\ N^T \end{pmatrix} = \begin{pmatrix} X_B B^T \\ X_N N^T \end{pmatrix},
\]

and (4.5) is equivalent to solving the original problem (4.3) with the preconditioner \( X_B B^T = \tilde{L} \tilde{U} \). At all stages, the partitioned factorization pinpoints a nonsingular \( B \) that corresponds to the usual basis matrix in the simplex method, and ultimately it will correspond to an optimal basis.

In general, it may not be necessary to recompute the \( LU \) factorization at every iteration. If the current factors are \( X_k A^T = L_k U_k \), the factors of \( X_{k+1} A^T \) could be taken as \( L_{k+1} = \tilde{D} L_k \) and \( U_{k+1} = U_k \), where \( \tilde{D} = X_{k+1} X_k^{-1} \), as long as \( \tilde{D} \) is reasonably well-conditioned. In some cases it may be efficient to update the \( LU \) factors to account for those elements of \( \tilde{D} \) that introduce ill-conditioning, using some of the update procedures available in LUSOL.

Another approach is to compute the \( LU \) factorization of \( C^T \), so that

\[
C^T = AX = LU,
\]

(4.6)
where \( L \) is nonsingular and \( U \) is an \( m \times n \) trapezoid. In this case, LUSOL will need to be enhanced to use a threshold form of complete pivoting, in order to ensure that the first \( m \) columns of \( U \) point to a suitable basis matrix \( B \). If a matrix \( C \) has full column rank, a reliable factorization \( C = LU \) can almost always be obtained using Gaussian elimination with threshold partial pivoting. By this we mean that when a potential pivot is selected from the remaining rows and columns, it must be reasonably large relative to other elements in the same column, but it need not be compared with elements in other columns. Virtually all existing \( LU \) software uses this strategy.

When computing the factorization (4.6), the column rank will be drastically deficient, and the triangular part of \( U \) will reflect the true rank only if pivots are chosen to be reasonably large relative to all remaining elements; i.e., only if threshold complete pivoting is implemented. Such a strategy requires keeping track of the largest remaining element at each stage without excessive overhead. Any such enhancement to LUSOL would carry a benefit whether \( C \) or \( C^T \) is being factorized, since \( XAT \) is expected to be rank-deficient.

5. References


5. References


Osborne, M. R. (1986). Dual barrier functions with superfast rates of convergence for the linear programming problem, Department of Statistics, Research School of Social Sciences, Australian National University.


**Title:** A Note on Nonlinear Approaches to Linear Programming

**Authors:** Philip E. Gill, Walter Murray, Michael A. Saunders & Margaret H. Wright

**Performing Organization:**
Department of Operations Research - SOL
Stanford University
Stanford, CA 94305-4022

**Controlling Office:**
Office of Naval Research - Dept. of the Navy
800 N. Quincy Street
Arlington, VA 22217

**Report Date:** April 1986

**Number of Pages:** 12 pp.

**Distribution Statement:** This document has been approved for public release and sale; its distribution is unlimited.

**Supplementary Notes:**

**Key Words:**
- Linear programming
- Karmarkar's method
- Barrier methods
- Sparse least-squares
- LU preconditioning

**Abstract:** See next page.
SOL 86-7: "A Note on Nonlinear Approaches to Linear Programming,"
Philip E. Gill, Walter Murray, Michael A. Saunders &
Margaret H. Wright

Certain new approaches to linear programming have recently received considerable publicity because of the promise of substantial improvements in efficiency compared to the simplex method. This note briefly discusses several research directions in methods for solving linear programs using nonlinear problem transformations. In particular, we describe application of a barrier transformation to the dual, and the development of sparse least-squares methods based on the LU factorization of the least-squares matrix or its transpose.
END

Dtic

7-86