

AD-A169 847

THE CONTINUOUS PROJECTIVE SUMT METHOD FOR CONVEX
PROGRAMMING(U) GEORGE WASHINGTON UNIV WASHINGTON DC
INST FOR MANAGEMENT SCIE... G P MCCORNICK 04 JUN 86
GWU/INSE/SERIAL-T-507/86

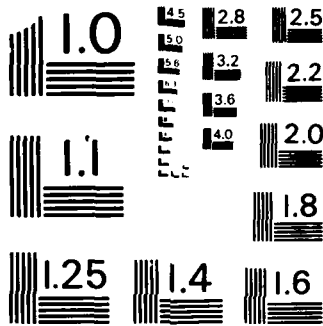
1/1

UNCLASSIFIED

F/G 12/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

12

AD-A169 847

THE CONTINUOUS PROJECTIVE SUMT
METHOD FOR CONVEX PROGRAMMING

by

Garth P. McCormick

THE
GEORGE
WASHINGTON
UNIVERSITY

STUDENTS FACULTY STUDY R
ESEARCH DEVELOPMENT FUT
URE CAREER CREATIVITY CC
MMUNITY LEADERSHIP TECH
NOLOGY FRONTIER DESIGN
ENGINEERING APPREHENSIVE
GEORGE WASHINGTON UNIV

DTIC
SELLING
JUL 15 1986
A



DTIC FILE COPY

Approved
for release
by NSA
on 08-16-2011

SCHOOL OF ENGINEERING
AND APPLIED SCIENCE

12

THE CONTINUOUS PROJECTIVE SUMT
METHOD FOR CONVEX PROGRAMMING

by

Garth P. McCormick

GWU/IMSE/Serial T-507/86 ✓
4 June 1986

The George Washington University
School of Engineering and Applied Science
Department of Operations Research

DTIC
ELECTED
S JUL 15 1986
A

Institute for Management Science and Engineering

Research Sponsored by
Contract N00014-86-K-0052
Office of Naval Research

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

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Department of Operations Research

Institute for Management Science and Engineering

Abstract
of
GWU/IMSE/Serial T-507/86
4 June 1986

THE CONTINUOUS PROJECTIVE SUMT
METHOD FOR CONVEX PROGRAMMING

by

Garth P. McCormick

An algorithm for solving convex programming problems is derived from the differential equations characterizing the trajectory of unconstrained minimizers of the classical logarithmic barrier function method. Convergence of this continuous Projective SUMT method to a global solution of a convex programming problem is proved under minimal assumptions. Extension of the algorithm to a form which handles linear equality constraints produces a differential equation analogue of Karmarkar's projective method for linear programming. The concluding discussion includes a discrete form of the algorithm. *A-*



Research Sponsored by
Office of Naval Research

	<input checked="" type="checkbox"/>	
	<input type="checkbox"/>	
	<input type="checkbox"/>	
By _____		
Distribution/		
Availability Codes		
Dist	Avail and/or	Special

AI

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Department of Operations Research

Institute for Management Science and Engineering

THE CONTINUOUS PROJECTIVE SUMT
METHOD FOR CONVEX PROGRAMMING

by

Garth P. McCormick

1. Derivation of the Continuous
Projective SUMT Method

Without loss of generality, a nonlinear programming problem can be written:

$$\begin{array}{l} \text{minimize } c^T x \\ x \in E^n \end{array}$$

subject to the restrictions

$$x \in R = \{x \mid g_i(x) \geq 0, \text{ for } i = 1, \dots, m\}.$$

(If in the original formulation the objective function $f(x)$ is nonlinear, adding a variable z , an inequality constraint $z - f(x) \geq 0$, and minimizing z creates an equivalent optimization problem.)

The logarithmic barrier function approach for solving the problem is to minimize

$$P(x, r) = c^T x - r \sum_{i=1}^m \ln[g_i(x)]$$

over the strict interior

$$R^0 = \{x \mid g_i(x) > 0, \text{ for } i = 1, \dots, m\}$$

for a sequence of values $r_k \downarrow 0$.

Under certain conditions the unconstrained minimizers of the barrier function approach solutions of the original problem.

The minimizing trajectory, denoted by $x(r)$, is often unique and continuously differentiable as a function of r . When it is (see [Fiacco and McCormick 1968, p. 77]),

$$dx(r)/dr = \nabla P[x(r), r]^{-1} c/r. \quad (1.1)$$

This fact has been very helpful in the development of extrapolation techniques for accelerating convergence to the solution.

To conform to the usual way in which differential equations are written, let $t = 1/r$, and define

$$V(x) = - \sum_{i=1}^m \nabla g_i(x) [1/g_i(x)] + \sum_{i=1}^m \nabla g_i(x) [1/g_i(x)]^2 \nabla g_i(x)^T.$$

Let x_0 be a point in the strict interior R^0 . The projective SUMT method is to apply the differential equation (1.1) without restricting x_0 to be an unconstrained minimizer of the barrier function for some value of r . Making the substitutions above, this leads to the differential equation

$$dx/dt = -V(x)^{-1} c \quad (1.2)$$

which is to be solved from the initial point $x(t_0) = x_0$.

Example

Consider the problem:

$$\text{minimize } x + y \quad \text{subject to } g(x,y) = -x^2 + y \geq 0.$$

Let the starting point be $(0,1)$, where $g_0 = 1$. The differential equation to be solved is

$$\begin{aligned} \begin{pmatrix} dx/dt \\ dy/dt \end{pmatrix} &= - \left[\begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} (-x^2 + y)^{-1} + \begin{pmatrix} -2x \\ 1 \end{pmatrix} (-x^2 + y)^{-2} (-2x, 1) \right]^{-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= - \begin{bmatrix} (1+2x)(-x^2 + y) \\ (2x^2 + 2x + 2y)(-x^2 + y) \end{bmatrix} (1/2) \end{aligned}$$

The solution is (from the point $(0,1)$, $t_0 = 0$)

$$x(t) = -t/[2(t+1)], \quad y(t) = [1/(t+1) + t^2/[4(t+1)^2]].$$

The trajectory of the differential equation is plotted in Figure 1. It converges to the solution $(-1/2, 1/4)$.

2. Convergence to a Global Solution of the Convex Programming Problem

The convergence analysis will assume the following.

Assumption 1. The strict interior R^0 of the feasible region is not empty.

Assumption 2. The problem functions $g_i(x)$ are concave and twice continuously differentiable.

Assumption 3. The set of points such that $x \in R$, and

$$f(x) = \{ \inf c^T x \quad \text{s.t. } x \in R \}$$

is compact.

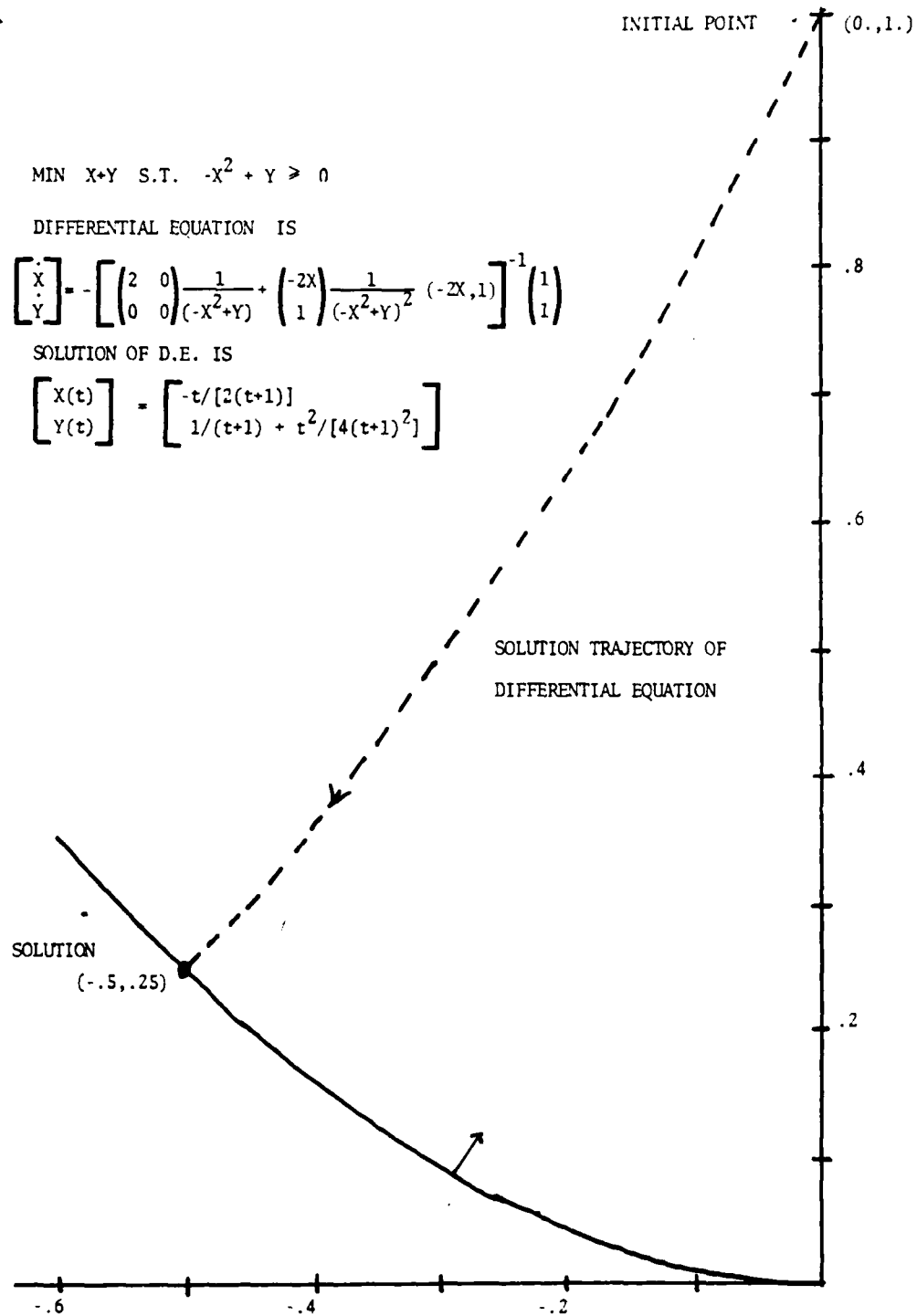


Figure 1. Solution of Example Problem by Continuous Projective Algorithm.

If a problem does not satisfy Assumption 3, then there are global minimizers "at infinity." There would be no reason to expect the algorithm to pick out a finite one at the end of the converging trajectory.

Another interesting consequence can be deduced from Assumption 3.

Corollary 1. Let x_0 be any feasible point. Then

$$\{x \mid f(x) \leq f(x_0), x \in R\}$$

is compact.

Proof: The proof is on page 94 of the book [Fiacco and McCormick 1968].

The importance of this is that the trajectory $x(t)$ remains in a compact set.

Assumption 2 implies that the matrix $V(x)$ is always positive semi-definite. In order for the algorithm to work, positive definiteness is required.

Assumption 4. For every $x \in R^0$, $V(x)$ is positive definite.

This assumption is not limiting and is usually satisfied in practice. For example, if the linear constraints of a problem (including some nonnegativity constraints if they are present) have rows which span E^n , $V(x)$ is positive definite.

The next assumption is traditional in nonlinear programming.

Assumption 5. Let x_0 be any point in R . The gradients of the constraints equal to zero at x_0 are linearly independent.

This assumption is not needed. Use of it facilitates the exposition of the proof of convergence.

The next well-known theorem on bordered matrices is used in the convergence analysis.

Lemma 1. Let A, C be symmetric matrices and assume that the bordered matrix below is positive definite. Then A^{-1}, C^{-1} exist and are positive definite and

$$\begin{pmatrix} A & B^T \\ B & C \end{pmatrix} = \begin{pmatrix} A^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} A^{-1} & B^T \\ A & B \\ -I & \end{pmatrix} (D^{-1}) [BA^{-1}, -I]$$

where D is positive definite and

$$D = (C - BA^{-1}B^T)$$

Proof: omitted.

The following well-known approximations are also needed.

Lemma 2. When t is large,

$$(It + B)^{-1} = I(1/t) + o(1/t).$$

Proof: omitted.

Lemma 3. When A^{-1} exists, and t is large,

$$(A + B(1/t))^{-1} = A^{-1} - A^{-1}BA^{-1}(1/t) + o(1/t).$$

The following notation will be used through the rest of the paper. Let $G = \text{diag}(g_i(x))$ be a diagonal matrix consisting of the first r constraints. Let N^T be the n by r matrix of the gradients of the first r constraints. Let S be an n by $(n-r)$ basis matrix generating the null space of N . (This assumes that N has rank r .)

The matrix M is the residual when $N^T G^{-2} N$ is removed from V . Thus

$$M = - \sum_{i=1}^m \nabla g_i^2(x) (1/g_i(x)) + \sum_{i=r+1}^m \nabla g_i(x) [1/g_i^2(x)] \nabla g_i^T(x).$$

Let $N\#$ denote any r by n matrix such that $N(N\#) = I$.

Note that $S^T \nabla g_i(x) = 0$, for $i = 1, \dots, r$. Note also that $(N\#)^T \nabla g_i(x) = e_i$, for $i = 1, \dots, r$, where e_i is a column vector with a one in the i th position and zeroes elsewhere.

In terms of the quantities defined above, there are two useful explicit expressions for the inverse of V . The first is

$$\begin{aligned} V^{-1} &= \begin{pmatrix} N & G & N+M \\ & & \end{pmatrix}^{-1} = [N\#, S] \begin{bmatrix} G^{-2} + (N\#)^T MN\#, & (N\#)^T MS \\ S^T MN\#, & S^T MS \end{bmatrix}^{-1} \begin{pmatrix} (N\#)^T \\ S^T \end{pmatrix} \\ &= [N\#, S] \left\{ \begin{pmatrix} [G^{-2} + (N\#)^T MN\#]^{-1}, & 0 \\ 0, & 0 \end{pmatrix} \right. \\ &\quad \left. + \begin{pmatrix} [G^{-2} + (N\#)^T MN\#]^{-1} \\ -I \end{pmatrix} [B^{-1}] \begin{pmatrix} [G^{-2} + (N\#)^T MN\#]^{-1}, & -I \end{pmatrix} \right\} \begin{pmatrix} (N\#)^T \\ S^T \end{pmatrix} \end{aligned} \quad (2.1)$$

where $B = [(S^T MS - S^T MN\#[G^{-2} + (N\#)^T MN\#]^{-1} (N\#)^T MS]$. The second is

$$\begin{aligned} [N\#, S] \left\{ \begin{pmatrix} 0 & 0 \\ 0 & (S^T MS)^{-1} \end{pmatrix} \right. \\ \left. + \begin{bmatrix} -I \\ (S^T MS)^{-1} S^T MN\# \end{bmatrix} (E^{-1}) [-I, (N\#)^T MS (S^T MS)^{-1}] \right\} \begin{pmatrix} (N\#)^T \\ S^T \end{pmatrix} \end{aligned} \quad (2.2)$$

where $E = G^{-2} + (N\#)^T MN\# - (N\#)^T MS (S^T MS)^{-1} S^T MN\#$.

These were obtained using the bordered matrix inverse formula from Lemma 1.

The first theorem states that for finite values of t , the form of the differential equation is such that no constraint value can be equal to zero.

Theorem 1. For all finite t , $g_i(t) > 0$, for $i = 1, \dots, m$.

Proof: Assume the contrary. There is a finite value \bar{t} where at least one of the constraints equals zero.

It is easy to show that there is a nonempty set I of the subset of the integers from 1 to m with the properties that

$$\lim_{t \rightarrow \bar{t}} \frac{g_i(t)}{g_j(t)} > 0, \text{ for all } i, j \text{ in } I, \text{ and} \quad (2.3)$$

$$\lim_{t \rightarrow \bar{t}} \frac{g_i(t)}{g_j(t)} = 0, \text{ for all } i \in I, \text{ all } 1 \leq j \leq m, j \notin I.$$

Intuitively the integers in I are associated with the constraints tending to zero at \bar{t} at the same, fastest, rate.

Without loss of generality assume that $I = \{1, \dots, r\}$.

In order to prove the theorem, an approximation to $V(t)^{-1}$ will be made. Now

$$\begin{aligned} & g_1(t)^2 M(t) \\ &= - \sum_{i=1}^m \nabla_{g_i(t)}^2 [g_1(t)^2 / g_i(t)] + \sum_{i=r+1}^m \nabla_{g_i(t)} \frac{g_1(t)^2}{g_i(t)^2} \nabla_{g_i(t)}^T. \end{aligned}$$

This tends to zero as $g_1(t)$ tends to zero because of (2.3). Using the approximation from Lemma 2,

$$[G^{-2}(t) + (N\#(t))^T M(t) N\#(t)]^{-1} = G(t)^2 + o(g_1(t)^2).$$

Combining this with (2.1) and the approximation from Lemma 3,

$$\begin{aligned} B^{-1}(t) = & [[S^T(t)M(t)S(t)]^{-1} \\ & + [S^T(t)M(t)S(t)]^{-1} S^T(t)M(t)(N\#(t))] \\ & \cdot G(t)^2 [(N\#(t))^T M(t)S(t)] [S^T(t)M(t)S(t)]^{-1}] \\ & + o(g_1(t)^2). \end{aligned}$$

The full approximation is then

$$\begin{aligned} V^{-1}(t) &= [N\# - S(S^T MS)^{-1} S^T MN\#] [G]^2 [(N\#)^T - (N\#)^T MS(S^T MS)^{-1} S^T] \\ &+ S(S^T MS)^{-1} S^T + o(g_1(t)^2). \end{aligned} \quad (2.4)$$

From this it follows, for $i = 1, \dots, r$,

$$dg_i/dt = -\lambda_i (g_i)^2 + o(g_1(t)^2), \quad \text{and} \quad (2.5)$$

$$df/dt = - \sum_{i=1}^r (\lambda_i)^2 g_i^2 - c^T S(S^T MS)^{-1} S^T c + o(g_1(t)^2) \quad (2.6)$$

where $\lambda = [(N\#)^T - (N\#)^T MS(S^T MS)^{-1} S^T] c$ is the vector of second order estimates of the K-K-T multipliers.

For the following analysis, the zero order terms are ignored.

Since every term in (2.6) is nonpositive, it follows that

$$f(t) \leq f(t_0) - \int_{t_0}^t [\lambda_1(\tau) g_1(\tau)]^2 d\tau. \quad (2.7)$$

From (2.5) it follows that

$$\ln[g_1(t)] - \ln[g_1(t_0)] = - \int_{t_0}^t \lambda_1(\tau) g_1(\tau) d\tau. \quad (2.8)$$

Since by assumption, f is bounded below in the feasible region, the second term on the right of (2.7) above goes to a finite value.

Therefore the right-hand side of (2.8) goes to a finite value. If $g_1(t)$ went to zero as $t \rightarrow \bar{t}$ the left hand side of (2.7) would go to $-\infty$. This contradiction proves the theorem.

Let \bar{x} denote the limit of $x(t)$ as t goes to infinity. "Bar" notation will be associated with other quantities evaluated at \bar{x} .

Let u_i denote the i th component of $(N\#)^T c$. These will be shown later to converge to the Karush-Kuhn-Tucker multipliers.

Lemma 2. For those $g_i(t)$, for $i = 1, \dots, r$,

$$\lim_t t g_i(t) > 0. \quad (2.9)$$

Proof: Using the previous approximation,

$$dg_i/dt = -g_i^2 [u_i - (c^T S)(S^T M S)^{-1} (S^T M N\#) e_i] + o(g_i^2).$$

The quantity multiplying g_i^2 above has a finite limit as $t \rightarrow \infty$. In order for the above limit to approach zero, l'Hopital's rule implies that

$$(dg_i/dt)/g_i^2$$

goes to $-\infty$. This contradicts the fact that it has a finite limit.

Corollary 2. For all $i = 1, \dots, m$, (2.9) holds.

Proof. The first r constraints go to zero faster than any of the last $m-r$ so the corollary is trivially true.

From now on the matrix N^T will consist of the gradients of all the constraints which are going to zero. Assume that these are g_1, \dots, g_r . The matrix M has the form

$$M = - \sum_{i=1}^m \nabla^2 g_i(x) / g_i(x) + \sum_{i=r+1}^m \nabla g_i(x) (1/g_i(x))^2 \nabla g_i(x)^T.$$

Because of Corollary 2, the matrix M/t has a finite limit. It may tend to singularity, but this doesn't hurt the convergence analysis.

Because of Corollary 1, $f(t)$ converges monotonically to some lower bound. Thus

$$\lim_{t \rightarrow \infty} t c^T V(t)^{-1} c = 0.$$

Two results of this are (see (2.1) and (2.2))

$$\lim_{t \rightarrow \infty} t c^T N^\# [G^{-2} + (N^\#)^T M N^\#]^{-1} (N^\#)^T c = 0 \quad (2.10)$$

and

$$\lim_{t \rightarrow \infty} t c^T S (S M S)^{-1} S^T c = 0.$$

From the latter it follows that

$$c^T S (S M S / t)^{-1} S^T c = 0.$$

Because M/t has a finite limit (recall $(S M S)$ is positive definite) it follows that:

$$\lim_t c^T S(t) = 0. \quad (2.11)$$

Lemma 3. The point \bar{x} is a Lagrangian stationary point.

Proof: Because of (2.11), i.e., that the gradient c of the linear objective function is orthogonal to the basis matrix generating the null space of the binding constraint gradients, it is well known that

$$c = \bar{N}^T \bar{u}$$

for some vector of Lagrange multipliers \bar{u} . Because of the linear independence assumption there is only one vector and it is

$$\bar{u} = (\bar{N}^\#)^T c.$$

It now remains to show that the components of this vector are nonnegative.

Assume the g_i 's are ordered from "fastest" to "slowest" in the rate at which they tend to zero. Within the set of g_i 's tending to zero, there are possibly two groups. The first, $\{g_1, \dots, g_p\}$, has the property that

$$\lim_t t g_i^2(t) = 0.$$

For the second ($i = p+1, \dots, r$),

$$\lim_t t g_i^2(t) > 0 \quad (\text{possibly } \infty).$$

Let G_1 represent the diagonal matrix associated with the first set,

and G_2 the diagonal matrix associated with the second set. Then

$$\begin{aligned} & t c^T (N\#) [G_2^{-2} + (N\#)^T MN\#]^{-1} (N\#)^T c \quad (\text{which goes to zero from (2.10)}) \\ & = \text{a scalar tending to zero} + (u^2)^T [G_2^{-2}/t + D]^{-1} u^2, \end{aligned}$$

where u^2 is the portion of $(N\#)^T c$ corresponding to the constraints with indices $p+1, \dots, r$, and D is a matrix with a finite limit. The matrix $[G_2^{-2}/t + D]$ is positive definite. From this it follows that

$$\bar{u}^2 = 0. \quad (2.13)$$

That is, the multipliers associated with the constraints going to zero at a "slow" rate are zero.

The exact formula for dg_i/dt is (see (2.2))

$$\begin{aligned} & dg_i/dt \\ & = c^T (N\#, S) \left\{ \begin{pmatrix} 0 & 0 \\ 0 & (S^T MS)^{-1} \end{pmatrix} + \begin{bmatrix} -I \\ (S^T MS)^{-1} S^T MN\# \end{bmatrix} [E^{-1}] \right. \\ & \quad \left. \cdot [-I, (N\#)^T MS (S^T MS)^{-1}] \begin{bmatrix} (N\#)^T \nabla g_i \\ S^T \nabla g_i \end{bmatrix} \right\}, \end{aligned}$$

where $E = G_2^{-2} + (N\#)^T MN\# - (N\#)^T MS (S^T MS)^{-1} S^T MN\#$.

Recall, $(N\#)^T \nabla g_i = e_i$, and $S^T \nabla g_i = 0$. Note further that for

$1 \leq i \leq p$,

$$E^{-1} e_i = (0, \dots, 0, t g_i^2, 0, \dots, 0)^T + o(t g_i^2/p).$$

The above becomes

$$dg_i/dt = [u_i + c^T S(S^T MS)^{-1} MN^T]g_i + o(tg_i^2(t)).$$

Since $c^T S$ has been shown to tend to zero (2.11), it follows that the dominant term is $-u_i$. Because g_i is going to zero, this ultimately must have a nonnegative value. Thus

$$\bar{u}_i \geq 0, \quad \text{for } i = 1, \dots, p. \quad (2.14)$$

Theorem 2. The point \bar{x} is a global solution for the convex programming problem.

Proof: Because of (2.12), (2.13), and (2.14), there is a vector \bar{u} with nonnegative elements such that

$$c = \bar{N}^T \bar{u}$$

where \bar{N}^T is the matrix of the binding constraint gradients. Thus the Karush-Kuhn-Tucker sufficiency conditions are satisfied at \bar{x} .

3. Linear Equality Constraints

Consider the problem

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & g_i(x) \geq 0, \quad i = 1, \dots, m \\ & Ax = b. \end{aligned}$$

The modified algorithm to handle the linear equality constraints is given below. Let $n-K$ be the rank of A .

Let B be a basis matrix for the null space of A . The modified algorithm is to solve the differential equation

$$dx(t)/dt = - B[B^T V B]^{-1} B^T c$$

where V is, as before,

$$- \sum_{i=1}^m \nabla^2 g_i(x) (1/g_i(x)) + \sum_{i=1}^m \nabla g_i(x) (1/g_i(x)) \nabla g_i^T(x).$$

It is required that the initial point x_0 satisfy $Ax_0 = b$, and

$g_i(x_0) > 0$, for $i = 1, \dots, m$.

The proof of convergence to a global solution of the problem above when the functions $g_i(x)$ are twice continuously differentiable and concave is a straightforward modification of the proofs just presented. The problem is to be regarded as constrained in a subspace of reduced dimension.

Specifically, the equivalent problem is

$$\min_{z \in \mathbb{R}^K} C^T z \quad \text{s.t.} \quad G_i(z) \geq 0, \quad \text{for } i = 1, \dots, m,$$

where $C^T = c^T B$, and $G_i(z) = g_i(x_0 + B(z - z_0))$, for $i = 1, \dots, m$.

The assumptions which were made on the problem in x space are now to be made on the problem in z space.

The changes in the assumptions are straightforward. The only one which needs explicit clarification is the assumption that the "V" matrix in z space is positive definite. This matrix has the form

$$- \sum_{i=1}^m \nabla^2 G_i(z) / (G_i(z)) + \sum_{i=1}^m \nabla G_i(z) (1/G_i^2(z)) \nabla G_i^T(z)$$

which is (using the chain rule of differentiation twice)

$$- \sum_{i=1}^m B^T \nabla^2 g_i(x) B / g_i(x) + \sum_{i=1}^m B^T \nabla g_i(x) (1/g_i^2(x)) \nabla g_i^T(x) B.$$

This is, under the concavity assumption, always positive semidefinite, but the additional assumption of definiteness is required. This, as noted before, is usually satisfied in practice.

Linear Programming and Karmarkar's Algorithm

One form of a linear programming problem is:

$$\min c^T x \quad \text{s.t.} \quad Ax = b, \quad x_j \geq 0, \quad j = 1, \dots, n.$$

In order to apply the recent projective method of Karmarkar [Karmarkar 1984], the problem must have the form

$$\min c^T x \quad \text{s.t.} \quad Ax = b, \quad e^T x = n, \quad x_j \geq 0, \quad \text{for } j = 1, \dots, n.$$

He further assumes that the minimum value equals zero and that the matrix A has full row rank. He then projects x into another space where the objective function is nonlinear. Attempting to minimize the transformed problem yields a corresponding iteration in the x place of

$$x_{k+1} = \left[x_k - \alpha D \left[I - (DA^T, e) \left[\begin{pmatrix} AD \\ e^T \end{pmatrix} (DA^T, e) \right]^{-1} \begin{pmatrix} AD \\ e^T \end{pmatrix} \right] Dc \right] \beta,$$

where $D = \text{diag}(x_j^k)$, α is a step-size scalar, and β is a scalar chosen to ensure that $e^T x_{k+1} = n$. (Here e is a vector of all ones.)

If the requirement $e^T x = n$ were ignored, Karmarkar's direction would be proportional to

$$s_1 = -D[I - (DA)^T(AD A)^{-2}(AD)]Dc.$$

The direction given by the Projected SUMT algorithm modified for linear equality constraints is

$$s_2 = -D[B D^{-2} B^{-1}] B^{-1} c,$$

where B is a basis for the null space of A .

It will now be shown that these two vectors are the same. (The similarity between directions generated by Karmarkar's algorithm and those given by the projected form of the logarithmic barrier function method was first pointed out in [Gill, et al. 1985]). Now

$$As_1 = -AD^2 c + (AD A)^T(AD A)^{-2} AD^2 c = 0, \text{ and}$$

$$As_2 = -AS[S D^{-2} A^{-1}] S^{-1} c = 0.$$

Furthermore

$$B D^{-2} s_1 = -B c + B A^T (AD A)^{-2} AD^2 c = -B c, \text{ and}$$

$$B D^{-2} s_2 = -(B D^{-2} B)(B D^{-2} B)^{-1} B^{-1} c = -B c.$$

It is necessary to show that each row of A is linearly independent of $B D^{-2}$. If not, there is some row (w.l.o.g. the first) for which a vector d exists such that

$$e_1^T A = d^T B D^{-2}.$$

Multiplying both sides on the right by Bd yields

$$0 = e_1^T A B d = d^T B^T D^{-2} B d.$$

Since $B^T D^{-2} B$ is positive definite this implies $d = 0$, a contradiction.

Now

$$\begin{pmatrix} A \\ B^T D^{-2} \end{pmatrix} \begin{pmatrix} s_1 - s_2 \\ 1 \end{pmatrix} = 0,$$

and the matrix has rank n . Thus the vectors are equal.

4. Discussion

It is not being argued that the Projective SUMT method is exactly equivalent to the algorithm proposed by Karmarkar. But the similarities are striking, and the amount of work required to compute the direction vectors seems equivalent.

Several points are worth noting. First, the convergence of the Projective SUMT method depends on several reasonable assumptions which are usually satisfied in practice. It does not depend on the value of the objective function being equal to zero at the solution, or on the existence of some potential function which is being minimized at each iteration. The reason that the algorithm never encounters the boundary of the feasible region for finite t is that the form of the V matrix, with the $(g_1)^2$'s occurring in the denominator, forces a slower rate in the change of these constraint values as the boundary is approached.

Work is in progress to describe a discrete version of the Projective SUMT method. The most likely form it will take is to

generate at x_k the direction of search

$$s_k = -V(x_k)^{-1} c.$$

The next point is $x_{k+1} = x_k - V(x_k)^{-1} c t_k$, where t_k is the minimizer of the function

$$-2n[f(x_k) - f(x_k(t))] - \sum_{i=1}^m 2n[g_i(x_k(t))],$$

where $x_k(t) = x_k + s_k t$ is restricted to remain in R^0 . (Here, of course, $f(x) = c^T x$.)

The step-size function is a variant of the method of centers function proposed in [Huard 1967]. Because $s_k^T \nabla f(x_k) < 0$, the derivative of the function above gives an infinite impulse to the term associated with the objective function and f decreases for t small. As the point $x_k(t)$ approaches a constraint boundary, the function increases. Because the step-size function is convex (and usually strictly convex) there is a unique solution to the step-size problem.

It is not obvious to the author that any discrete version of this algorithm will be superior to the ordinary logarithmic barrier function method. The main reason for the abandonment of the latter was the ill-conditioning of the Hessian matrix of second derivatives of the barrier function. The same problem arises in inverting the V -matrix of the Projective SUMT algorithm. To date, however, no one has tried the approximate inverse as given by (2.4). The usual approach has been to form the matrix and then invert it. By separating out the g_i 's, which

are small, and computing the associated $N\#$, S , and $(S^T M S)^{-1}$, the numerical problems are no more difficult than those ordinarily encountered in nonlinear programming. The fact that the Hessian matrices are usually dyadic in structure (see [McCormick 1983, Chapter 3]) can be used to advantage also.

It is not obvious that any discrete version of this algorithm is superior to a second order "direct" method for nonlinear programming problems (see [McCormick 1983, Chapter 15] for a development of such a method). Indications from people who have experimented with Karmarkar's method are that in the initial phases, his method can bypass the combinatorial problems arising in movement from vertex to vertex. That is, in his method, as well as in that just described, all the constraints influence motion. In direct methods, only constraints binding, or nearly binding, enter into the computations. If this is the reason for some of the success, it argues for better "active set strategies" in traditional linear and nonlinear programming algorithms.

To answer the question about which algorithms will in the long run be "better" will require much computational testing, and much work on the matrix methods needed to implement them.

References

- [1] A. V. Fiacco and G. P. McCormick. Nonlinear Programming: Sequential Unconstrained Minimization Techniques. New York: John Wiley and Sons, Inc., 1968.
- [2] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright. "On Projected Newton Barrier Methods for Linear Programming and an Equivalence to Karmarkar's Projective Method." Technical Report SOL 85-11, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA. July 1985.
- [3] P. Huard. "Resolution of Mathematical Programming with Nonlinear Constraints by the Method of Centers." In Nonlinear Programming, pp. 207-219. Edited by J. Abadie. Amsterdam: North Holland, 1967.
- [4] N. Karmarkar. "A New Polynomial Time Algorithm for Linear Programming." Combinatorica 2 (1984): 373-395.
- [5] G. P. McCormick. Nonlinear Programming: Theory, Algorithms and Applications. New York: John Wiley and Sons, Inc., 1983.

END

DTIC

8-86