Discussion Paper No. 16

ROW DROPPING PROCEDURES FOR CUTTING PLANE ALGORITHMS*

by

Frederic H. Murphy

October 16, 1972

Graduate School of Management
Northwestern University
Evanston, Illinois

I wish to thank Professors Harvey Wagner, Gordon Bradley, and Robert Mifflin of Yale University for their helpful comments.

* This work was supported in part by National Science Foundation Grant GS - 3032.

Abstract

Row dropping procedures are provided for cutting plane algorithms. An example is provided to show that if nonbinding rows are dropped at every iteration while disregarding degeneracy in the linear program of the subproblem, the algorithms will not converge. Also, the results of Topkis [8] and Eaves and Zangwill [3] are reproduced using different techniques.

Topkis [8] and Eaves and Zangwill [3] have shown that constraints may be dropped at each iteration of certain cutting plane algorithms, provided that the objective function to be maximized is strictly quasi-concave. We show that the strict quasi-concavity condition can be modified to quasi-concavity in the case of the Veinott [9] algorithm and to concavity in the case of the Kelly [5] algorithm, provided that nonbinding rows are dropped only when the trial solution satisfies certain conditions. We also will show how to achieve the results in the strictly quasi - concave case by an alternative method of proof.

Consider the nonlinear programming problem

(1) $$\text{maximize } f(x_1,\ldots,x_n)$$

subject to

(2) $$g_i(x_1,\ldots,x_n) \leq 0 \qquad \text{for } i = 1,\ldots,m$$

(3) $$x_j \geq 0 \qquad \text{for } j = 1,\ldots,n,$$

with $f(\cdot)$ quasi-concave and $g_1(\cdot),\ldots,g_m(\cdot)$ quasi-convex and $g_i(0) < 0$ for $i = 1,\ldots,m$. To ensure that the origin is interior to the feasible region, we need only assume Slaters constraint qualification and translate the coordinates of the x_j's. Also, assume the feasible region is contained in a compact set X, and that $f(\cdot),g_1(\cdot),\ldots,g_m(\cdot)$ are continuous, finite and possess continuous gradients for all points in X.

First we reformulate the problem as (NLP).

(4) $$\text{maximize } x_0$$

subject to

(5) $$g_0(x) \equiv x_0 - f(x) - c \leq 0$$

(6) $$g_i(x) \leq 0 \text{ for } i = 1,\ldots,m$$
$$x_j \geq 0 \text{ for } j = 0,\ldots,n,$$

with c any constant such that $f(0) + c > 0$, and setting $x = (x_0, x_1, \ldots, x_n)$. We augment X appropriately to include the added dimension for x_0, and we define S as the feasible region to NLP.

Before we state what a cutting plane algorithm is, we define a central concept to cutting plane algorithms.

Definition of a Cutting Plane Function

In the manner of Topkis [8], we define a limiting cutting plane function.

Definition 1 A point to set mapping $(a(x), b(x))$ from $X \sim S$ into E^{n+2} with $a(x) \in E^{n+1}$ and $b(x) \in E^1$ is a limiting cutting plane function if

(7) $$S \subset H(x) \equiv \{y\colon\ a(x)\cdot y \leq b(x)\}$$

for all $x \in X \sim S$, $(a(x), b(x))$ is bounded on $X \sim S$, and, for any $\{x_k \mid k = 1,2,\ldots\}$, with $\lim_{k\to\infty} x_k = \bar{x} \in X \sim S$ the limit point $(\bar{a}, \bar{b})$ of any convergent subsequence of $\{(a(x_k), b(x_k))\}$ satisfies $\bar{a}\cdot\bar{x} \sim \bar{b}$.

Let $G(x) = \max\,\{g_0(x), \ldots, g_m(x)\}$ and $u(x)$ be the gradient of a

$g_h(x) = G(x)$. If we assume each $g_i(x)$ is convex, then $G(x)$ is a convex function, and we have

$$(8) \qquad G(y) \geq G(x) + u(x)\cdot(y-x)$$

for all $x, y \in X$. Topkis [8] shows that

$$(9) \qquad [u(x),\ u(x)\cdot x - G(x)]$$

is a limiting cutting plane function, which is the one used by Kelley [5]. Note that convexity is important. Without convexity, if, for some y, $G(y) \leq 0$ and $G(x) + u(x)\cdot(y-x) > 0$, then part of the feasible region might be cut off at some iteration. By (8) we are assured that this will not happen.

We have assumed that $G(0) < 0$. For $x \in X \sim S$, define $\lambda(x) = \max\{\lambda: \ \lambda x \in S\}$ and set $w(x) = \alpha(x)x$ for any function $\alpha(x) \in [\lambda(x), 1]$ with $\frac{G(w(x))}{G(x)} \in [0,1]$. Topkis [8] shows that

$$(10) \qquad [u(w(x)),\ u(w(x))\cdot w(x) - G(w(x))]$$

is a family of cutting plane functions. By setting $\alpha(x) = \lambda(x)$, (10) becomes the cutting plane function proposed by Veinott [9]. For using Veinott's cutting plane function, the appropriate assumption on the constraints is quasi-convexity with $\nabla g_i(x) \neq 0$ when $g_i(x) = 0$. Otherwise, we need the assumption of convexity to use (10) when $\lambda(x) < \alpha(x) \leq 1$. Note that (9) is the special case of (10) where $\alpha(x) = 1$.

The Algorithm

Assume that X is a bounded convex polyhedron and, for convenience, that we can specify $0 < b_j < \infty$ such that we can let $X = \{x \mid b_j \geq x_j \geq 0$ for $j = 0.\dots.n\}$. At iteration k we have a linear program (LP1) of the form

(11) $$\text{maximize } x_o \equiv p_k$$

subject to

(12) $$\sum_{j=0}^{n} a_j(x_k^i)x_j - b(x_k^i) \leq 0 \text{ for } i = 1,\dots,r_k$$

(13) $$b_j \geq x_j \geq 0 \text{ for } j = 0,\dots,n,$$

where the vectors x_k^i are solutions from previous iterations of LP1 (at iteration 0 LP1 consists of (11) and (13)).

For the algorithm choose either the cutting plane function in (9) or one from the family in (10). Evaluating the cutting plane function at x_k^i, we have $a_0(x_k^i),\dots,a_n(x_k^i)$, and $b(x_k^i)$, the coefficients in LP1. Also, r_k is the number of constraints in the linear program at iteration k, and b_j is the bound on the $j^{\underline{th}}$ variable in the definition of X.

The dual of LP1 (DLP1) is

(14) $$\text{minimize } \sum_{i=1}^{r_k} b(x_k^i)y_i + \sum_{j=0}^{n} b_j z_j$$

subject to

(15) $$\sum_{i=1}^{r_k} a_0(x_k^i)y_i + z_0 - s_0 = 1$$

(16) $$\sum_{i=1}^{r_k} a_j(x_k^i)y_i + z_j - s_j = 0 \text{ for } j = 1,\dots,n.$$

$$y_i,\ z_j,\ s_j \geq 0 \text{ for } i = 1,\dots,r_k \text{ and } j = 0,\dots,n,$$

with the values y_i^k, for $i = 1,\dots,r_k$, and z_j^k and s_j^k for $j = 0,\dots,n$, given by an optimal basic solution.

Solve DLP1. If $X_{[k]}$ is feasible, $X_{[k]}$ is optimal and we stop. Otherwise, we proceed as follows. If Condition 1 below is satisfied, we drop all nonbasic columns not associated with the z_j and add a new column to DLP1 determined by evaluating the cutting plane function at $X_{[k]}$, the optimal solution to LP1 associated with our solution to DLP1, setting $x_{k+1}^{r_{k+1}} = X_{[k]}$.

<u>Condition 1</u> (a) In DLP1 $y_i^k \geq \epsilon$, $z_j^k \geq \epsilon$ and $s_j^k \geq \epsilon$ for all basic y_i^k, z_j^k, and s_j^k,

(b) the absolute value of the determinant of the basis is greater than ϵ,

where $\epsilon > 0$ and fixed for all iterations.

If Condition 1 is violated, our iteration consists of adding the new column to DLP1 without dropping any nonbasic columns. As soon as Condition 1 is again satisfied we resume dropping nonbinding cutting planes. We show that if $X_{[k]}$ is not feasible in NLP at some finite iteration then any convergent subsequence of $X_{[k]}$ converges to an optimal solution of NLP.

The following example illustrates the need to avoid degeneracy in DLP1, which corresponds to multiple optima in LP1.

(17) $$\text{maximize } x_1 + x_2$$

subject to

(18) $$(x_1-c_1)^2 + (x_2-c_2)^2 \leq c_3$$

(19) $$x_1, x_2 \geq 0.$$

Because (17) is a linear function we need not reformulate the nonlinear program in the form (4), (5) and (6). This allows the following diagram to be two dimensional instead of three dimensional to better illustrate the degeneracy problem. Therefore, let LP1 at iteration 1 be

(20) $$\text{maximize } x_1 + x_2$$

subject to

(21) $$x_1 + x_2 \leq b^1$$

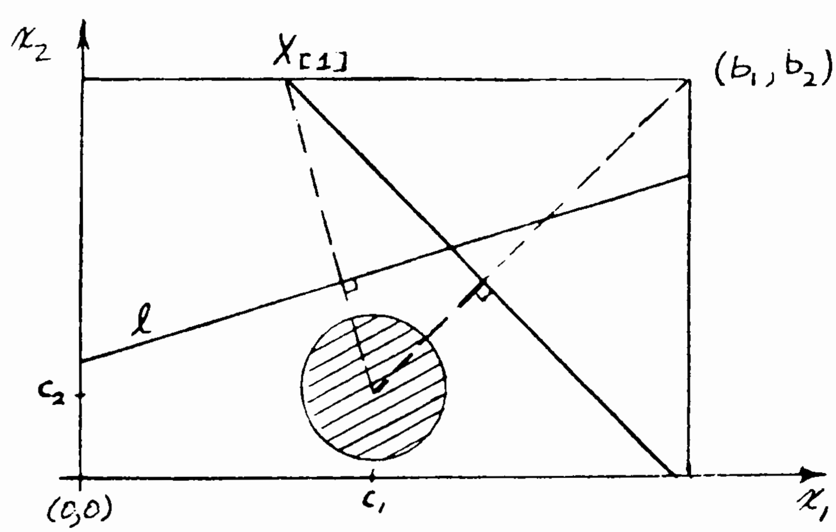
(22) $$x_1 \leq b_1$$

(23) $$x_2 \leq b_2$$

(24) $$x_1, x_2 \geq 0,$$

where $b^1 = b(X_{[0]})$.

Pictorially we have



The cross-hatched area is the feasible region of NLP; and the line connecting $X_{[1]}$ and x is determined by (21). The feasible region of (20),...,(24) is the portion of the rectangle to the left of this line. Observe that all points between x and $X_{[1]}$ are alternative optima in LP1. Suppose that the point $X_{[1]}$ is the solution produced by the simplex algorithm. The cutting plane provides a boundary, ℓ, to the new linear programming feasible region that is perpendicular to the line connecting $X_{[1]}$ and (c_1,c_2), by the symmetry of $(x_1-c_1)^2 + (x_2-c_2)^2$. The portion of the feasible region defined by (21),...,(24) above ℓ is now cut off. If the optimal solution at iteration 2 is $X_{[2]} = x$, dropping all nonbinding constraints makes $X_{[1]}$ feasible again, allowing the algorithm to alternate between $X_{[1]}$ and x as trial solutions, never converging to an optimal solution to NLP.

Part (b) of Condition 1 is used to keep the constraint matrix coefficients uniformly bounded in DLP1 at an optimal solution. To see this, first note that the coefficients in DLP1 are uniformly bounded because they are determined by continuous functions evaluated at points in a compact set. To get to the optimal simplex tableau of DLP1, we perform multiplications and additions involving the uniformly bounded coefficients of DLP1 and the inverse of our optimal basis. That the elements of inverse matrix are uniformly bounded is seen by calculating the inverse by using the matrix of cofactors, whose elements are bounded, and dividing by the determinant of the basis matrix which is bounded away from zero by (b).

Condition 1 in theory is stronger than a condition that allows column dropping whenever there are no multiple optima in LP1. However, from a computational point of view the two conditions are equivalent. If we set ϵ at the closest number to zero that the computer can handle, then part (a) of Condition 1 translates into "degeneracy" within numerical tolerances of the computer. Part (b) of Condition 1 becomes a requirement that the coefficients of DLP1 are within the capability of the computer to handle while performing simplex iterations.

If the simplex or any related algorithm is not used to solve DLP1 and the practical issue of coefficients increasing in an unbounded fashion due to linear dependence is not faced, we can provide an altered version of Condition 1.

<u>Condition 2</u> (a) In DLP1 $y_i^k \geq \varepsilon$, $z_i^k \geq \varepsilon$ and $s_j^k \geq \varepsilon$ for all basic y_i^k, z_j^k, and s_j^k, where $\varepsilon > 0$ and fixed for all iterations.

(b) Once (a) is violated columns are no longer dropped from DLP1.

Either Condition 2 is violated at some finite iteration and convergence is guaranteed by the proofs in Topkis [8] or Condition 2 is satisfied at every iteration and convergence is guaranteed by algorithm IV in [3]. There is still another condition which allows us to drop columns from DLP1.

<u>Condition 3</u> In DLP1, start with the optimal solution determined at iteration k-1 as the initial trial solution at iteration k. If the new column added by the cutting plane function can be pivoted into the basis at a level at least $\varepsilon > 0$, ε fixed for all iterations, then all nonbasic columns in DLP1 associated with the solution y_i^{k-1}, z_j^{k-1}, s_j^{k-1}, for $i = 1,\ldots,r_k$ and $j = 0,\ldots,n$, retained from iteration k-1 may be dropped.

Let

$$\bar{X}_k = \begin{cases} \bar{X}_{k-1} \text{ if } \sum_{g_i(\bar{X}_k)>0} g_i(\bar{X}_k) < \sum_{g_i(X_{[k]})>0} g_i(X_{[k]}) \\ X_{[k]} \text{ if } \sum_{g_i(\bar{X}_k)>0} g_i(\bar{X}_k) \geq \sum_{g_i(X_{[k]})>0} g_i(X_{[k]}) \end{cases} \quad (25)$$

where $\bar{X}_1 = X_1$. For this condition we will show that any convergent subsequence of $\bar{X}_k$ is an optimal solution to NLP if the algorithm does not terminate finitely. To apply Condition 3 as it is stated, we have to start the succeeding iteration before we know whether we can drop nonbasic columns from the present iteration. Also, all we show is the existence of a subsequence converging to an optimal solution of NLP. This condition is, however, a generalization of Condition 1. If Condition 1 holds, since all coefficients are uniformly bounded and we have a basic solution uniformly bounded away from zero, we can pivot in our new column at a level greater than or equal to some fixed number greater than zero.

Convergence Using Condition 1

If there is no uniform bound on the number of columns in DLP1, for every iteration, then convergence is guaranteed by the original convergence proofs. Therefore, we need only consider the case where there is a uniform bound on the number of columns in DLP1.

For a sequence of points $x_k \in X$ with $x_k \to \bar{x}$ let $g_h(x_k) = G(x_k)$ for some fixed h. By construction $G(x)$ is continuous. Consequently, $g_h(\bar{x}) = G(\bar{x})$. Now,

$$h_k(x) = \nabla g_h(x_k) \cdot (x-x_k) + g_h(x_k) \quad (26)$$

is a linear equation that defines a hyperplane tangent to $g_h(x)$ at x_k and is an appropriate cutting plane at x_k. By the continuity of the gradient of $g_h(x)$,

(27) $$\lim_{k\to\infty} \nabla g_h(x_k) = \nabla g_h(\bar{x}),$$

that is:

Lemma 1

(28) $$h(x) = \nabla g_h(\bar{x})\cdot(x-\bar{x}) + g_h(\bar{x})$$

defines a tangent hyperplane to $g_h(x)$ at $\bar{x}$ and is an appropriate cutting plane at $\bar{x}$.

Lemma 2 For every $h(x) = \sum_{j=0}^{n} a_j x_j - b$ that defines a tangent hyperplane to a constraint function $g_i(x)$ at an $\hat{x} \in X$, if

(a) $g_i(x)$ is convex, or

(b) $g_i(x)$ is quasi-convex, and for $\hat{x}$ with $g_i(\hat{x}) = 0$ $\nabla g_i(\hat{x}) \neq 0$,

then there is a uniform bound $\delta > 0$ such that $b \geq \delta$ for all $\hat{x} \in X$.

Proof: By assumptions (a) or (b) and since $0 > G(0)$, $b \geq 0$. Assuming the convexity of $g_i(x)$,

(29) $$G(x) \geq g_i(x) > \sum_{j=0}^{n} a_j x_j - b,$$

(30) $$0 > G(0) \geq -b.$$

Setting $G(0) = -\delta$ we have proved the lemma under hypothesis (a).

Let us assume for all $\delta > 0$ there is an x^δ satisfying (b) for $g_i(x)$ with $h_\delta(x) = \sum_{j=0}^{n} a_j^\delta x_j - b^\delta$, where $(a_0^\delta,\ldots,a_n^\delta) = \nabla g_i(x^\delta)$, having $0 \leq b_\delta < \delta$. Let $\bar{x}$ be the limit of a sequence of x^δ's for a sequence of δ's converging to zero. Note that by assumption $g_i(\bar{x}) = 0$, and so $\bar{x} \neq 0$. Because $\delta \to 0$,

$$(31) \qquad \sum_{j=0}^{n} a_j^\delta x_j^\delta \to \sum_{j=0}^{n} \bar{a}_j \bar{x}_j = 0 \text{ as } \delta \to 0,$$

with $(\bar{a}_0, \ldots, \bar{a}_n) = \nabla g_i(\bar{x})$.

Since $g_i(x)$ is pseudo-convex, $g_i(0) < g_i(\bar{x})$ implies $\sum_{j=0}^{n} \bar{a}_j(0-\bar{x}_j) < 0$,

which contradicts (31).

Therefore, there is a $\delta > 0$ such that

$$(32) \qquad b > \delta.$$

Now we can bound the dual variables in DLP1.

<u>Lemma 3</u> In DLP1 y_i^k and z_j^k are uniformly bounded for all i and j and k.

Proof:

$$(33) \qquad b(x_k^i) \geq \delta > 0$$

and with b_0 one of the upper bounds defining X

$$(34) \qquad b_0 \geq x_0,$$

or, since p_k is the optimal value of LP1,

$$(35) \qquad b_0 \geq p_k,$$

which means that

$$(36) \qquad b_0 \geq \sum_{i=1}^{r_k} b(x_k^i) y_i^k + \sum_{j=0}^{n} b_j z_j^k.$$

Since $b(x_k^i) > 0$ for all i and $b_j > 0$ for all j,

$$(37) \qquad b_0 \geq b(x_k^i) y_i^k \quad \text{for } i = 1, \ldots, r_k.$$

Therefore,

(38) $$y_i^k \leq \frac{b_0}{\delta} \text{ for } i = 1,\ldots,r_k.$$

Similarly

(39) $$z_j^k \leq \frac{b_0}{b_j} \text{ for } j = 0,\ldots,n.$$

Before continuing, we clarify a point about the construction of DLP1. At each iteration we drop all nonbasic columns associated with the constraints in (12). After doing that,we reindex all our columns that remain retaining the columns in the same order from left to right and add our new column to the rightmost position in the portion of the matrix for columns associated with the constraints in (12). Doing this means that the x_k^i that determines column i at iteration k is different from the x_{k+1}^i at iteration k+1 if for some $h \leq i$ the column determined by x_k^h is dropped. The reason for making this clear is that we shall use a subsequence on which the x_k^i's that determine column i on this subsequence will converge to a limit. All that we need to know about these x_k^i's for the subsequence to exist is that they are in the compact space X. We don't need to know anything as to how they came to determine column i for the purpose of taking a subsequence.

By the definition of limiting cutting plane function, $a_j(x_k^i)$ for $b(x_k^i)$ are bounded for all i, j and k. Therefore, we may take a subsequence indexed by k_w with the following properties:

(1) Condition 1 is satisfied

(2) $a_j(x_{k_w}^i) \to a_{ij}^\infty$ for $i = 1,\ldots,r$ and $j = 0,\ldots,n$

(3) $b(x^i_{k_w}) \to b^i_\infty$ for $i = 1,\ldots,r$

(4) $X_{[k_w]} \to X_\infty$

(5) the indices of the basic columns in DLP1 corresponding to constraints (12) in LP1 at k_w are the same for all k_w, with the number of these columns being $r' - 1$.

(6) $a_j(X_{[k_w]}) \to a^{\infty+1}_{r'j}$ for $j = 0,\ldots,n$

(7) $b(X_{[k_w]}) \to b^{r'}_{\infty+1}$

(8) $y_i^{k_w} \to y_i^\infty$ and $z_j^{k_w} \to z_j^\infty$ for all i and j.

We may now construct a limiting linear program (LLP1) to the subsequence indexed by k_w:

(40)
$$\text{maximize } x_0 = p_\infty$$

subject to

(41)
$$\sum_{j=0}^{N} a^\infty_{ij} x_j - b^i_\infty \leq 0 \quad \text{for } i = 1,\ldots,r$$

(42)
$$b_j \geq x_j \geq 0 \quad \text{for } j = 0,\ldots,n.$$

Since p_k is monotonically decreasing and bounded from below by $f(0) + c$, we have $\{p_k\}$ converging to some limit $\bar{p}$. By the Hoffman and Karp [4] result on the continuity of linear programs, we conclude $p_\infty = \bar{p}$.

As in Murphy [6], we construct a limiting linear program for the successor subsequence [10], that is, the subsequence of iterations indexed by $k_w + 1$. Let $|Q| = r'-1$. Then the number of columns in DLP1 at iteration $k_w + 1$ is $r' + n + 1$, since we save active columns in DLP1 corresponding to constraints (12) in LP1, all active and inactive columns corresponding to upper bounds, and add a new column. In LP1 constraint i at iteration k_w+1 is constraint $\ell(i)$ at iteration k_w.

We have shown that the coefficients of the constraint added for iteration k_w+1 converge to a limit, and the constraints $\ell(i)$ converge by the construction of k_w. Noting lemma 1, we can perform a cutting plane iteration on LLP1 using the limits of the coefficients of the constraints added at iteration k_w+1. Also, nonbinding constraints are dropped since the limiting solution to DLP1 satisfies the row dropping rule. Therefore, we arrive at LLP2 as a limit of a subsequence of linear programs and as a cutting plane iteration to LLP1 where LLP2 is:

(43) $$\text{maximize } x_0$$

subject to

(44) $$\sum_{j=0}^{n} a_{ij}^{\infty+1} x_j - b_{\infty+1}^{i} \leq 0 \quad \text{for } i = 1,\ldots,r'$$

(45) $$b_j \geq x_j \geq 0 \text{ for } j = 0,\ldots,n.$$

The value of the optimal LLP2 is also p_∞ since any subsequence of the convergent sequence p_k has the same limit and because of the continuity of linear programs [4].

<u>Lemma 4</u> Let $y_i^{\infty+1} = y_{\ell(i)}^{\infty}$ for $i = 1,\ldots,r'-1$ and $y_{r'}^{\infty+1} = 0$. Then $y_i^{\infty+1}$, for $i = 1,\ldots,r'$, is an optimal solution to the dual of LLP2 with

$z_j^{\infty+1} = z_j^{\infty}$ for all j.

Proof: Note that $y_i^{\infty+1}$, for $i = 1,\ldots,r'$, is a feasible nondegenerate solution to the dual of LLP2 since, y_i^{∞}, for $i = 1,\ldots,r$, is feasible in the dual of LLP1. Also, $y_i^{\infty+1}$, for $i = 1,\ldots,r'$, is optimal because the optimal value of LLP2 is p_{∞}, and this solution has the value p_{∞} in the dual of LLP2.

Therefore, we have:

<u>Lemma 5</u> The point X_{∞} is feasible in NLP.

Look at the optimal simplex tableau of the dual of LLP1 using the solution $y_1^{\infty},\ldots,y_r^{\infty},z_1^{\infty},\ldots,z_n^{\infty}$. The components of X_{∞} are the coefficients of the slack variables in the objective function. Using the cutting-plane function, we determine a new column to be added to the dual of LLP1. Dropping all nonbasic columns still leaves the solution to the dual of LLP1 feasible. Since the dual of LLP2 has the same value for an optimal solution, p_{∞}, our optimal nondegenerate basic feasible solution to the dual of LLP1 is optimal in the dual of LLP2. Since the solution is nondegenerate, if the added column has a negative relative price in the simplex tableau [1], the value of the optimal solution would decrease by pivoting in the new column, contradicting the Hoffman and Karp result [4]. This means the associated primal solution to LLP1 is feasible in LLP2; that is, X_{∞} is feasible and optimal in LLP2. Or, since X_{∞} is not cut off by our cutting plane function, it is feasible in NLP.

By the construction of the subsequence k_w we see that any convergent subsequence of $X_{[k]}$'s is feasible in NLP as long as it is a subsequence where Condition 1 is always satisfied. Let k_v index any convergent subsequence of $X_{[k]}$'s with $X_{[k_v]} \to \hat{X}$ and k_z index the last iteration before iteration k_v where cutting planes are dropped. We are still assuming that Condition 1 is satisfied infinitely often, so that $\{k_z\}$ is infinite. Taking an appropriate convergent subsequence of $X_{[k_z]}$ without reindexing, we have $X_{[k_z]} \to X_\infty$, and a limiting linear program (LLP3) can be constructed as before from the subsequence k_z. We take the appropriate convergent subsequence of $X_{[k_v]}$, again without reindexing, so that with our new subsequence the index k_z is still the last iteration where cutting planes are dropped before iteration k_v. Now $X_{[k_v]}$ is feasible in LP1 at iteration k_z and $p_{k_z} \geq p_{k_v} \geq p_\infty$. Therefore, by taking limits, $\hat{X}$ is a feasible and optimal solution to LLP3; and $\hat{X} = X_\infty$ by Condition 1, which insures there are no multiple optima to LLP3.

Noting that convergence is guaranteed by the proofs of convergence in Topkis [8] when there is no uniform bound on the number of constraints in LP1, we can say

<u>Theorem 1</u> The limit point of any convergent subsequence of optimal solutions to LP1 is feasible in NLP, and, therefore, optimal.

Proof of Convergence using Condition 3.

Again, as with Condition 1 we need only concern ourselves with the case where there is a uniform bound on the number of columns in DLP1.

<u>Lemma 6</u> Let $X_{[k_u]}$ be an infinite subsequence of iterations where Condition 3 is satisfied and $X_{[k_u]} \to X_\infty$. Then X_∞ is a feasible solution to NLP.

Proof: Let

$$d^{k_u} = \sum_{j=0}^{n} a_j(X_{[k_u]})X_{[k_u]} - b(X_{[k_u]}). \tag{46}$$

Assume we can take a subsequence of $X_{[k_u]}$, retaining our index k_u, such that

$$\lim_{k_u \to \infty} d^{k_u} \geq 2\gamma > 0. \tag{47}$$

For (47) to hold, by the definition of limit, there exists a K such that $d^{k_u} \geq \gamma$ for $k_u \geq K$. But $-d^{k_u}$ is the coefficient in the objective function of the new column in DLP1 relative to the basis retained from iteration k_u-1 in DLP1 at iteration k_u. Pivoting into the basis this new column at a level greater than ε decreases the objective function by an amount greater than $d^{k_u} \cdot \varepsilon$, or $\gamma \cdot \varepsilon$. Having an improvement of $\gamma \cdot \varepsilon$ infinitely often means the values of the optimal solutions to DLP1 are unbounded below, because the values of the optimal solutions to the DLP1's are monotonically decreasing with the decrease at iterations k_u greater than $\gamma \cdot \varepsilon$. This contradicts the existence of a feasible solution to NLP. Therefore, $d^{k_u} \to 0$, or X_∞ is feasible in LLP1.

Theorem 2 Any convergent subsequence of $\bar{X}_k$ as defined in (25) is optimal in NLP.

Proof: Note that

$$\sum_{g_i(\bar{X}_k)>0} g_i(\bar{X}_k) \geq \sum_{g_i(\bar{X}_{k+1})>0} g_i(\bar{X}_{k+1}). \tag{48}$$

By the definition of $\bar{X}_k$

$$0 \leq \sum_{g_i(\bar{X}_k)>0} g_i(\bar{X}_k) \leq \sum_{g_i(X_{[k]})} g_i(X_{[k]}). \tag{49}$$

By lemma 6,

$$\sum_{g_i(X_{[k_w]})>0} g_i(X_{[k_w]}) \to 0. \tag{50}$$

Therefore, by (48) (49), and (50), we have

$$\sum_{g_i(\bar{X}_k)>0} g_i(\bar{X}_k) \to 0, \tag{51}$$

or, any convergent subsequence of $\bar{X}_k$ is feasible in the limit and the theorem holds.

Convergence with a Strictly Quasi-concave Objective Function

The following is an alternative method of proof of the results of Topkis [8] using the concept of the continuity of mathematical programs [2].

Here we solve a sequence of mathematical programs consisting of a nonlinear objective function and linear constraints (MP1)

(52) $$\text{maximize } f(x) \equiv p_k$$

subject to

(53) $$\sum_{j=0}^{n} a_j(x_k^i)x_j - b(x_k^i) \leq 0 \quad \text{for } i = 1,\ldots,r$$

(54) $$b_j \geq x_j \geq 0 \text{ for } j = 0,\ldots,n.$$

Let $X_{[k]}$ be an optimal solution to MP1 at iteration k. At each iteration we use a cutting-plane function to add a constraint to cut off the optimal solution to MP1. At the same time we drop all nonbinding constraints (53) in MP1.

<u>Lemma 7</u> The value of the optimal solution of MP1 is monotonically decreasing, that is, $p_k > p_{k+1}$ if $X_{[k]}$ is not feasible in NLP.

Proof: Since $X_{[k]}$ is the optimal solution to MP1 at iteration k, $X_{[k]}$ is still the optimal solution to MP1 with nonbinding constraints removed. By adding the new cut to MP1 with nonbinding constraints removed, the feasible region is reduced in size. Therefore, $p_k \geq p_{k+1}$.

Assume $p_k = p_{k+1}$. By strict quasi concavity

(55) $$p_k < f(\lambda X_{[k]} + (1-\lambda)X_{[k+1]}) \quad \text{for } 0 < \lambda < 1.$$

Since only a finite number of nonbinding constraints are dropped at any iteration, there is a $\lambda^k < 1$ such that $\lambda^k X_{[k]} + (1-\lambda^k)X_{[k+1]}$ is feasible in MP1 at iteration k. Using (54), this contradicts the optimality of $X_{[k]}$ in MP1 at iteration k, which means $p_k > p_{k+1}$.

Now we must investigate the continuity of nonlinear programs of the form MP1. This issue is treated with more generality in [2]. First we introduce new notation. Let T_k be a subsequence of sets. We define

$$(56) \qquad \underline{\lim}\ T_k = \{x | x_k \to x \text{ for } x_k \in T_k\}$$

and

$$(57) \qquad \overline{\lim}\ T_k = \{x | x_{k_w} \to x \text{ for } x_{k_w} \in T_{k_w}\}.$$

The lim inf of T_k, $\underline{\lim}\ T_k$, is the set of limits of sequences and the lim sup of T_k, $\overline{\lim}\ T_k$, is the set of limits of subsequences. This means $\underline{\lim}\ T_k \subset \overline{\lim}\ T_k$. If $\underline{\lim}\ T_k = \overline{\lim}\ T_k$, then the limit of a sequence of sets is defined as

$$(58) \qquad \lim T_k = \underline{\lim}\ T_k = \overline{\lim}\ T_k.$$

<u>Lemma 8</u> Let A_k be a sequence of m x n matrices with $A_k \to A$ and b_k be a sequence of m dimensional vectors such that $b_k \to b$. Let X be a compact set. Also, let $T_k = \{x | A_k x \le b_k,\ x \in X\}$ and $T = \{x | A_x \le b,\ x \in X\}$. Assume there exists an x_0 with $A_k x_0 \le b_k - \delta \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ with $\delta > 0$ for all k. Then $\lim\limits_{k \to \infty} T_k = T$.

Proof: Let x be in T, so that $ax \le b$. Since $A_k \to A$, $b_k \to b$, and the components of x are bounded, because $x \in X$, then for all $\epsilon > 0$, there exists a K_ϵ such that

$$(59) \qquad A_k\, x \le b_k + \epsilon_k \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \text{ for } k \ge K_\epsilon$$

where $\varepsilon_k \geq 0$ and $\varepsilon_k \to 0$. This means

(60) $$A_k[(1-\frac{\varepsilon_k}{\delta})\, x + \frac{\varepsilon_k}{\delta}\, x_0] \leq (1 - \frac{\varepsilon_k}{\delta})b_k + (1 - \frac{\varepsilon_k}{\delta})\, \varepsilon_k \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} + \frac{\varepsilon_k}{\delta}\, b_k - \frac{\varepsilon_k}{\delta}\, \delta \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

$$= b_k - \frac{\varepsilon_k^2}{\delta} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \leq b_k$$

Therefore, $(1 - \frac{\varepsilon_k}{\delta})\, x + \frac{\varepsilon_k}{\delta}\, x_0$ is feasible in T_k and converges to x, which implies $T \subseteq \underline{\lim}\, T_k$.

Now let $x_{k_w} \in T_{k_w}$ and $\lim_{k_w \to \infty} x_{k_w} = \bar{x}$. Since X is compact, $\bar{x} \in X$. Since $x_{k_w} \in T_{k_w}$, $A_{k_w} x_{k_w} \leq b_{k_w}$ with the result that

(61) $$\lim_{k_w \to \infty} (A_{k_w} x_{k_w}) \leq \lim_{k_w \to \infty} b_{k_w} = b.$$

by Rudin [7]

(62) $$\lim_{k_w \to \infty} (A_{k_w} x_{k_w}) = (\lim_{k_w \to \infty} A_{k_w})\, (\lim_{k_w \to \infty} x_{k_w}),$$

that is,

(63) $$A\, \bar{x} \leq b,$$

or

(64) $$T \supseteq \overline{\lim}\, T_k.$$

Therefore we have

(65) $$T = \lim T_k.$$

<u>Theorem 3</u> Assume we have a sequence of mathematical programs

(66)
$$\underset{x \in X}{\text{maximize}} \quad c_k(x) \equiv p_k$$

subject to

(67)
$$A_k x \leq b_k$$

with nonempty feasible regions, $c_k(x) \to c(x)$, a continuous function for $x \in X$, where X is a compact space, $A_k \to A$, and $b_k \to b$, consider the mathematical program

(68)
$$\underset{x \in X}{\text{maximize}} \quad c(x) \equiv p$$

subject to

(69)
$$Ax \leq b.$$

Then $\lim p_k = p$, if there exists an x_0 with $A_k x_0 \leq b_k - \delta \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ with $\delta > 0$ for all k.

Proof: Using Lemma 8 and setting $T_k = \{x | A_k x \leq b, x \in X\}$ and $T = \{x | Ax \leq b, x \in X\}$, we have $\lim\limits_{k \to \infty} T_k = T$. Let Y_k be the set of

optimal solutions to (66) and (67) and let Y be the set of optimal solutions to (68) and (69). For every $\varepsilon > 0$, there is a K_ε such that for $k \geq K_\varepsilon$ and any $x \in T$, there exists an $x_k \in T_k$ where $|x_k - x| < \varepsilon_j$ and, for any $x_k \in T_k$, there exists an $x \in T$ where $|x_k - x| < \varepsilon$, and

$|c(x) - c_k(x)| < \epsilon$ for $x \epsilon X$. For $X_{[k]} \epsilon Y_k$ we have

(70) $$|c(X_{[k]}) - c_k(X_{[k]})| < \epsilon \text{ for } k \geq K_\epsilon,$$

and there exists an $x \epsilon T$ with

(71) $$|X_{[k]} - x| < \epsilon \text{ for } k \geq K_\epsilon.$$

By the continuity of $c(x)$ on X,

(72) $$|c(X_{[k]}) - c(x)| < \epsilon + \lambda,$$

where $\lambda \to 0$ as $\epsilon \to 0$.

Since $c(X_{[k]}) = P_k$ and $p \geq c(x)$,

(73) $$p_k \leq p + \epsilon + \lambda \text{ for } k \geq K_\epsilon.$$

In like manner,

(74) $$p \leq p_k + \epsilon + \lambda \text{ for } k \geq K_\epsilon.$$

Hence

(75) $$|p - p_k| \to 0 \text{ as } k \to \infty,$$

completing the proof.

We have here the same behavior as before with the limiting linear programs. We want to construct a subsequence of MP1's that converge to a limiting mathematical program. Therefore, we take a subsequence, indexed by k_w, with the following properties:

(1) $a_j(x^i_{k_w}) \to a^\infty_{ij}$ for $i = 1,\ldots,r$, and $j = 0,\ldots,n$

(2) $b(x^i_{k_w}) \to b^i_\infty$ for $i = 1,\ldots,r$

(3) $X_{[k_w]} \to X_\infty$

(4) the binding constraints in MP1 are the same for all k_w, with the number of these being $r'-1$

(5) $a_j(X_{[k_w]}) \to a^{\infty+1}_{r'j}$

(6) $b(X_{[k_w]}) \to b^{r'}_{\infty+1}$.

We may now construct a limiting mathematical program (LMP1) to the subsequence indexed by k_w:

(76)
$$\underset{x \in X}{\text{maximize}} \quad c(x) \equiv p$$

subject to

(77)
$$\sum_{j=0}^{n} a^\infty_{ij} x_j - b^i_\infty \leq 0 \quad \text{for } i = 1,\ldots,r.$$

By using Lemma 2 we see that $A_k \cdot 0 \leq b_k - \delta \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$, $\delta \geq 0$ and the conditions of Theorem 3 are satisfied. With a proof similar to that of Lemma 4, we can show that X_∞ is an optimal solution to LMP1. Performing a cutting-plane iteration on LMP1, we arrive at a second mathematical program (LMP2)

(78)
$$\underset{x \in X}{\text{maximize}} \quad c(x)$$

subject to

(79)
$$\sum_{j=0}^{n} a^{\infty+1}_{ij} x_j - b^i_{\infty+1} \leq 0 \quad \text{for } i = 1,\ldots,r',$$

where $a_{ij}^{\infty+1} = a_{\ell(i)j}^{\infty}$ and where constraint i at iteration k_w+1 is constraint $\ell(i)$ at iteration k_w. Note also that LMP2 is the limiting mathematical program to the subsequence k_{w+1}. Since the value of an optimal solution, p_k, is monotonically decreasing by Theorem 3 both LMP1 and LMP2 have the same optimal solution values, p, which leads us to

<u>Theorem 4</u> The point X_∞ is an optimal solution to NLP.

Proof: Assume X_∞ is not optimal in NLP. By Lemma 7 the value of an optimal solution to LMP2 should be strictly less than p if X_∞ is not optimal in NLP, contradicting the fact that p is the value of an optimal solution in LMP2.

References

(1) Dantzig, G., Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1963.

(2) Dantzig, G., Folkman and Shapiro, J., "On the Continuity of the Minimum Set of a Continuous Function," Journal of Mathematical Analysis and Applications, Vol. 17 (1967), pp. 519-549.

(3) Eaves, B. C. and Zangwill, W. I., "Generalized Cutting Plane Algorithms," Working Paper No. 274, Center for Research in Management Science, University of California, Berkeley, July 1969.

(4) Hoffmann, A. J. and Karp, "On Nonterminating Stochastic Games," Management Science, Vol. 12 (1966), pp. 359-370.

(5) Kelley, J. E., Jr., "The Cutting-Plane Method for Solving Convex Programs," Journal of the Society for Industrial and Applied Mathematics, Vol. 8 (1960), pp. 703-712.

(6) Murphy, F., "Column Dropping Procedures for the Generalized Programming Algorithm," submitted to Management Science.

(7) Rudin, W., Principles of Mathematical Analysis, 2nd Edition, McGraw, 1964.

(8) Topkis, D., "Cutting Plane Methods Without Nested Constraint Sets," Operations Research, Vol. 18 (1970), pp. 404-413.

(9) Veinott, A. F., Jr., "The Supporting Hyperplane Method for Unimodal Programming," Operations Research, Vol. 15 (1967), pp. 147-152.

(10) Zangwill, W., Non-linear Programming -- A Unified Approach, Prentice-Hall, 1969.