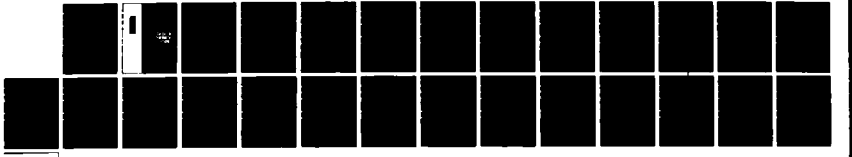
AD-A146 215 A TWO-SEGMENT APPROXIMATION ALGORITHM FOR SEPARABLE CONVEX PROGRAMMING WI. (U) TEXAS UNIV AT AUSTIN CENTER FOR CYBERNETIC STUDIES I ALI ET AL. JAN 84 CCS-RR-476 1/1

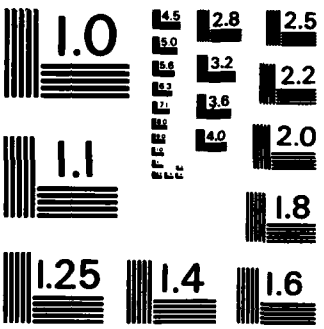
UNCLASSIFIED N00014-81-C-0236 F/G 12/1 NL



END

FILMED

DTIC



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

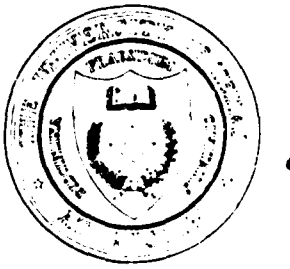
Research Report CCS 476

A TWO-SEGMENT APPROXIMATION ALGORITHM FOR SEPARABLE CONVEX PROGRAMMING WITH LINEAR CONSTRAINTS

by

I. Ali
A. Charnes
T. Song

CENTER FOR CYBERNETIC STUDIES
The University of Texas
Austin, Texas 78712



Research Report CCS 476

A TWO-SEGMENT APPROXIMATION ALGORITHM FOR SEPARABLE CONVEX PROGRAMMING WITH LINEAR CONSTRAINTS

by

I. Ali
A. Charnes
T. Song

January 1984

This research was supported in part by ONR Contracts N00014-81-C-0236 and N00014-82-K-0295 and USARI Contract MDA-903-83-K-0312 with the Center for Cybernetic Studies, The University of Texas at Austin. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Graduate School of Business 4.138
The University of Texas at Austin
Austin, Texas 78712
(512) 471-1821

A Two-Segment Approximation Algorithm for Separable Convex Programming with Linear Constraints

by

I. Ali, A. Charnes, T. Song

Abstract

We discuss a new algorithm for the separable convex programming with linear constraints. This is based on the approximation of the objective function by at most two linear pieces in the neighborhood of the current feasible solution. The two segments will be adaptively defined rather than predecided fixed grids. If, furthermore, the objective function is differentiable, and one introduces a non-Archimedean infinitesimal, the algorithm generates a sequence of feasible solutions every cluster point of which is an optimal solution. Computational tests on the problem with up to 196 non-linear variables is presented.

Key Words

Separable convex programming

Two-segment approximation

A TWO-SEGMENT APPROXIMATION ALGORITHM FOR SEPARABLE CONVEX PROGRAMMING WITH LINEAR CONSTRAINTS

by

I. Ali, A. Charnes, T. Song

1. Introduction

In 1954, Charnes and Lemke, in order to bring to bear the computational power of linear programming methods for computation of non-linear convex programming, developed a new linear programming algorithm which obviated the need to treat explicitly as additional constraints those which were upper bounds on individual variables (see [1]). Specifically, the reduction of the general convex quadratic programming problem to such terms was done by first introducing new variables and constraints in terms of which the objective function was separable in the new variables. Then one approximated the separable functions by piece-wise linear ones, each piece of which corresponded to an additional variable, individually upper bounded, whose sums would be such a new variable. This procedure, although clumsy and sometimes slow in computation to its optimum, an approximation to the optimum desired, was nevertheless stable and was generalized to a large class of differentiable convex programs with convex constraints by Griffith and Stewart by use of first-order Taylor series expansions of the objective and constraint functions. Successive refinement of these separable programming problems was employed to achieve better approximations to the exact optimal solution. See also Charnes and Cooper in [2].

In 1959, Charnes and Lemke developed a non-linear analog of the simplex algorithm which obviated the need to employ such successive linear programming approach. This method was never computationally tested since Lemke and Howson were unable to debug their computer code in what time

remained from their major effort of developing the bimatrix equilibrium solution and Lemke's complementarity theory.

In the intervening years, non-linear programming computation has focused on either constrained gradient methods or the parametric unconstrained methods (of Fiacco and McCormick) which, for computational stability, have required such complications as updating Hessian matrices in Newton-Raphson methods. Recently, efforts have swung back toward the original procedures now designated as SLP, "Successive Linear Programming." But these procedures themselves can be extremely slow and practically unusable if implemented by existing algorithms which are otherwise perfectly suitable for the most frequent usages of these algorithms. Thus, for example, Charnes and Phillips in solving a separable convex programming problem with pure network constraints via an approximating pure network problem with multiple arcs between the same nodes, discovered that the solution time for these with PNET, the Klingman, Karney, Glover, and Stutz world-preeminent primal network code, or with Barr's world-renowned SUPERK code, the world's preeminent out-of-kilter code, was several orders of magnitude slower in solution than anticipated for usual network problems of the same size. This difficulty was overcome with the invention of a new algorithm which required no more than three halves the computation time to be anticipated from PNET on ordinary problems.

The key efforts toward effective methods for non-linear convex programming problems ought therefore to concentrate on development of new algorithms for the special structures and inter-relations between linear programming problems arising in a sequential linear programming approach to their solution. In this paper, we consider convex separable programs with linear constraints and build on previous work by R.R. Meyer [4] to develop a new algorithm which appears thus far to have superior convergence properties (speed) as well as conveniences of representation than does his. In particular, special use is

made of adaptive bounding and grids rather than the fixed a priori grids he employs. Convergence properties of the new method are established and some examples are given of its performance.

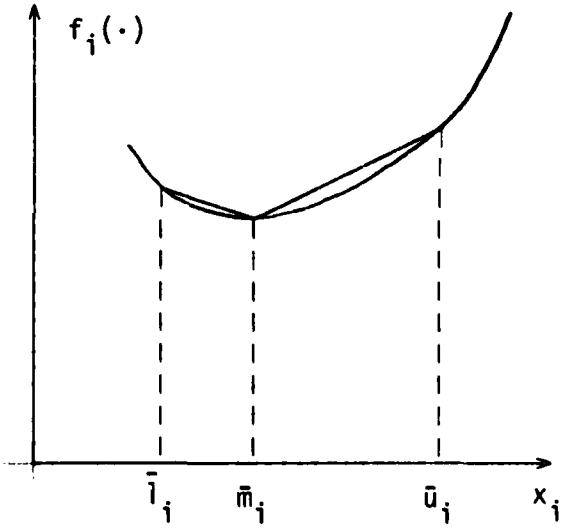
Thus in this paper we present a two segment approximation algorithm with adaptively defined rather than predecided segments. If, further, one introduces a non-Archimedean infinitesimal, the algorithm generates a sequence of feasible solutions every cluster point of which is an optimal solution.

Consider the following program.

$$\text{(P)} \qquad \begin{aligned} \min \quad & z = \sum_{i=1}^{p} f_i(x_i) + g^T y \\ \text{s.t.} \quad & \\ & Ax + Dy = b \\ & l \leqslant x \leqslant u \\ & \underline{y} \leqslant y \leqslant \bar{y} \end{aligned}$$

where $f_i(\cdot)$ is a convex function defined on an interval containing $[l_i, u_i]$; x, l, u are p-vectors with $l_i < u_i$, i = 1,...,p ; y, g, $\underline{y}$, $\bar{y}$ are q vectors and $\underline{y}_i < \bar{y}_i$, i = 1,...,q; A, D are m x p and m x q matrices respectively.

The basic idea is that in a neighborhood $\bar{l} \leqslant x \leqslant \bar{u}$, $\underline{y} \leqslant y \leqslant \bar{y}$ of the current feasible solution $(\bar{m}, y)$, each $f_i(x_i)$ in program (P) will be approximated by at most two linear pieces as in the figure. The currently chosen bounds of this neighborhood are called "artificial bounds." Using this approximation of the object function together with these artificial bounds, we solve the linear programming problem.



If the current point is optimal, or, if the optimal solution of the linear program has no component at an artificial bound, we halve the size of the artificially bounded neighborhood, and, based on this optimal solution we construct another approximating program. Otherwise, we keep the approximation unchanged for those variables that are not at the artificial bound whereas for those variables at artificial bounds we construct a new approximation on the current solution value with the same size interval as before. We then solve the new program. These procedures are repeated until specific criteria to be detailed later are satisfied.

In our procedures, x_i will be split into three parts: a constant part w_i, a decreasable part ξ_i and an increasable part η_i. ξ_i and η_i correspond to the left segment and right segment respectively. ξ_i is initially set at its upper bound and is a non-basic variable. It can only be decreased. Its lower bound 0 corresponds to an artificial bound. In contrast, η_i is initially set at its lower bound and may be a basic variable if x_i is a basic variable of an approximating program. Its upper bound corresponds to an artificial bound. Thus, there is a basic feasible solution available when one starts a new approximating program.

In the next section, we will give the detailed transformations and the algorithms. In section 3, we will prove that usually the algorithm will terminate in a finite number of iterations. If one introduces a non-Archimedean infinitesimal and if the objective function is differentiable, then the algorithm will produce an infinite sequence of feasible solutions every cluster point of which is an optimal solution of the original program. In section 4, some computational results based on network constraints will be presented.

2. Algorithm

We introduce two transformations. The first one, based on a feasible solution of program (P), creates a two segment piecewise-linear approximation program with a certain neighborhood size. The second one only changes part of the approximation problem corresponding to a variable at its artificial bound in the optimal basic solution of the approximating problem.

Suppose (x,y) is a feasible solution of (P) and δ is a positive real number, usually less than $\min_i (u_i - l_i)$. The transformation I gives another program $P(x,y,\delta)$ as follows:

$$
\begin{aligned}
& \min \quad z = d^T\xi + e^T\eta + g^Ty + \sum_{i=1}^{p} f_i(w_i) \\
& \text{s.t.} \\
P(x,y,\delta) \qquad & A\xi + A\eta + Dy = b - Aw \\
& 0 \leqslant \xi \leqslant \alpha \\
& 0 \leqslant \eta \leqslant \beta \\
& \underline{y} \leqslant y \leqslant \bar{y}
\end{aligned}
$$

where $\alpha_i = \min (\delta, x_i - l_i)$, $\beta_i = \min (\delta, u_i - x_i)$,

$w_i = x_i - \alpha_i$,

$$
d_i = \begin{cases} -M & , \text{ if } \alpha_i = 0 \\ (f_i(x_i) - f_i(w_i))/\alpha_i & , \text{ if } \alpha_i > 0 \end{cases}
$$

$$
l_i = \begin{cases} M & , \text{ if } \beta_i = 0 \\ (f_i(x_i+\beta_i) - f_i(x_i))/\beta_i & , \text{ if } \beta_i > 0 \end{cases}
$$

for $i = 1,\ldots,p$ and M is a large positive number.

Clearly, if (x,y) is a feasible solution of (P), then $(\alpha,0,y)$ must be a feasible solution of $P(x,y,\delta)$. Conversely, if (ξ,η,y) is a feasible solution of $P(x,y,\delta)$ then $(\omega+\xi+\eta,y)$ is a feasible solution of (P). As a matter of convenience we say x_i is a basic variable if either ξ_i or η_i is a basic variable. Also, if

x_i is a basic variable, after transformation I η_i would be counted as a basic variable even if it is at its lower bound.

Suppose $(\hat{\xi},\hat{\eta},\hat{y})$ is an optimal basic solution of $P(x,y,\delta)$. Let

$$J_1 = \{i : \hat{\xi}_i = 0 \ , \ \alpha_i > 0\}$$
$$J_2 = \{i : \hat{\eta}_i = \beta_i > 0\}$$

Definition: The optimal solution $(\hat{\xi},\hat{\eta},\hat{y})$ of $P(x,y,\delta)$ is called "regular" if and only if $\hat{\eta}_i > 0 \longrightarrow \hat{\xi}_i = \alpha_i$.

Since $f_i(x_i)$ is convex, $d_i \leqslant e_i$. Thereby, a perturbation can guarantee that the simplex algorithm will give a regular optimal solution. For a regular optimal solution $J_1 \cap J_2 = \phi$, and the following transformation II will change $P(x,y,\delta)$ into another program $P(\hat{x},\hat{y},\delta)$ of the same type, where $\hat{x} = w + \hat{\xi} + \hat{\eta}$.

$$\min \ \bar{z} = \bar{d}^T\xi + \bar{e}^T\eta + \delta^T y + \sum_{i=1}^{p} f_i(\bar{w}_i)$$

s.t.

$$A\xi + A\eta + Dy = b - A\bar{w}$$

$P(\hat{x},\hat{y},\delta)$

$$0 \leqslant \xi \leqslant \bar{\alpha}$$
$$0 \leqslant \eta \leqslant \bar{\beta}$$
$$\underline{y} \leqslant y \leqslant \bar{y}$$

where

$$\bar{\alpha}_i = \begin{cases} \alpha_i & , \ i \notin J_1 \cup J_2 \\ \min\ (\hat{x}_i - l_i \ , \ \delta) & , \ i \in J_1 \\ \delta & , \ i \in J_2 \end{cases} ,$$

$$\bar{\beta}_i = \begin{cases} \beta_i & , \ i \notin J_1 \cup J_2 \\ \min\ (u_i - x_i \ , \ \delta) & , \ i \in J_1 \\ \delta & , \ i \in J_2 \end{cases} ,$$

$$\bar{w}_i = \begin{cases} w_i & , \ i \notin J_1 \cup J_2 \\ \hat{x}_i - \bar{\alpha}_i & , \ i \in J_1 \cup J_2 \end{cases}$$

$$\bar{d}_i = \begin{cases} d_i & , \quad i \notin J_1 \cup J_2 \\ -M & , \quad i \in J_1 \text{ and } \bar{\alpha}_i = 0 \\ [f_i(\hat{x}_i) - f_i(\bar{w}_i)]/\bar{\alpha}_i & , \quad \text{otherwise,} \end{cases}$$

$$\bar{e}_i = \begin{cases} e_i & , \quad i \notin J_1 \cup J_2 \\ M & , \quad i \in J_2 \text{ and } \bar{\beta}_i = 0 \\ [f_i(\hat{x}_i + \bar{\beta}_i) - f_i(\hat{x}_i)]/\bar{\beta}_i & , \quad \text{otherwise.} \end{cases}$$

The variables $\{x_i : i \in J_1 \cup J_2\}$ are said to be at artificial bounds. Actually the difference between problems $P(x,y,\delta)$ and $P(\hat{x},\hat{y},\delta)$ is that the approximation of the objective function for artificial bound variables has different two piece segments. Next, set

$$\xi'_i = \begin{cases} \alpha_i & , \quad i \in J_1 \cup J_2 \\ \hat{\xi}_i & , \quad \text{otherwise,} \end{cases}$$

$$\eta'_i = \begin{cases} 0 & , \quad i \in J_1 \cup J_2 \\ \hat{\eta}_i & , \quad \text{otherwise} \end{cases}$$

It is easy to see that $(\xi',\eta',\hat{y})$ is a basic feasible solution of $P(\hat{x},\hat{y},\delta)$; and following lemma is true.

Lemma 1. Transformation II does not change the corresponding objective function value.

Proof: If $i \in J_1$ then $\hat{\xi}_i = 0$, $\hat{\eta}_i = 0$, $\hat{x}_i = w_i$

$$\begin{aligned} f_i(w_i) + d_i\hat{\xi}_i + e_i\hat{\eta}_i &= f_i(w_i) \\ &= f_i(\hat{x}_i) \\ &= f_i(\bar{w}_i) + \bar{d}_i\bar{\alpha}_i \\ &= f_i(\bar{w}_i) + \bar{d}_i\xi'_i + \bar{e}_i\eta'_i \quad . \end{aligned}$$

For $i \in J_2$, there are $\hat{\xi}_i = \alpha_i$, $\hat{\eta}_i = \beta_i$

$$\begin{aligned} f_i(w_i) + d_i\hat{\xi}_i + e_i\hat{\eta}_i &= f_i(w_i + \alpha_i + \beta_i) \\ &= f_i(\hat{x}_i) \\ &= f_i(\bar{w}_i) + \bar{d}_i\xi'_i + \bar{e}_i\eta'_i \end{aligned}$$

For other variables, the corresponding part of the objective function remains the same as before.

■

Algorithm

Step 0: Initialization. Starting with any usual linear programming software, one can get a basic feasible solution (x^0,y^0) of the program P, or show that the program P has no feasible solution.

Take $\varepsilon > 0$ and $0 < \delta < \min (u_i - l_i)$ where ε may be a non-Archimedean infinitesimal. The transformation I gives a program $P(x^0,y^0,\delta^0)$ denoted by P^1, and its basic feasible solution by $(\bar{\xi}^0,\bar{\eta}^0,y^0)$. Set $k = 1$.

Step 1: If $(\bar{\xi}^{k-1},\bar{\eta}^{k-1},y^{k-1})$ solves the program P^k, set $\delta^k = \frac{1}{2}\delta^{k-1}$, $(\xi^k,\eta^k,y^k) = (\bar{\xi}^{k-1},\bar{\eta}^{k-1},y^{k-1})$ and go to Step 3. If not, go to Step 2.

Step 2: Suppose that by solving program P^k, one gets a regular optimal solution (ξ^k,η^k,y^k). Let

$$J_1^k = \{i : \xi_i^k = 0 \ , \ \alpha_i^k > 0\} \ ,$$
$$J_2^k = \{i : \eta_i^k = \beta_i^k > 0\} \ .$$

If $J_1^k \cup J_2^k \neq \phi$, go to Step 4. Otherwise set $\delta^k = \frac{1}{2}\delta^{k-1}$ and go to Step 3.

Step 3: If $\delta^k < \varepsilon$, terminate; otherwise, set $x^k = \xi^k + \eta^k + w^k$.

By transformation I, one gets program $P(x^k,y^k,d^k)$ denoted P^{k+1} and its feasible basic solution $(\bar{\xi}^k,\bar{\eta}^k,y^k)$. Go to Step 1.

Step 4: Set $\delta^k = \delta^{k-1}$, $x^k = \xi^k + \eta^k + w^k$

By transformation II, one gets program $P(x^k,y^k,\delta^k)$ denoted P^{k+1} and its basic feasible solution $(\bar{\xi}^k,\bar{\eta}^k,\bar{y}^k)$. Go to Step 1.

If we denote the optimal objective function value of program P^k as z^k, following lemma holds:

Lemma 2. $\{z^k\}$ is a non-increasing sequence.

Proof: Clearly it is sufficient to prove that transformations I and II do not increase the corresponding objective function value. By Lemma 1, we need only to prove the assertion holds in the case of transformation I.

If $0 < \xi_i^k < \alpha_i^k$, by regularity of solutions, $\eta_i^k = 0$. Hence

$$x_i^k = \xi_i^k + w_i^k$$

By convexity of $f_i(\cdot)$,

$$\frac{f_i(x_i^k) - f_i(w_i^k)}{\xi_i^k} \leqslant \frac{f_i(w_i^k + \alpha_i^k) - f_i(w_i^k)}{\alpha_i^k} = d_i^k$$

Therefore

$$f_i(x_i^k) \leqslant f_i(w_i^k) + d_i^k \xi_i^k$$
$$= f_i(w_i^k) + d_i^k \xi_i^k + e_i^k \eta_i^k$$

On the other hand, by transformation I

$$f_i^k(w_i^{k+1}) + d_i^{k+1} \bar{\xi}_i^k = f_i(x_i^k)$$

and

$$\bar{\eta}_i^k = 0$$

so

$$f_i(w_i^{k+1}) + d_i^{k+1} \bar{\xi}_i^k + e_i^{k+1} \bar{\eta}_i^k \leqslant f_i(w_i^k) + d_i^k \xi_i^k + e_i^k \eta_i^k$$

By a similar argument, for the case $0 < \eta_i^k < \beta_i^k$, $(\xi_i^k = \alpha_i^k)$, the above inequality also holds.

For $\xi_i^k = 0$, or $\xi_i^k = \alpha_i^k$, $\eta_i^k = 0$, using the same argument as in lemma 1 it can be proved that the above inequality becomes an equality. In conclusion, we have

$$z^{k+1} \leqslant \bar{z}^{k+1} = \sum_{i=1}^{p} f_i(w_i^{k+1}) + d^{k+1}\,\bar{\xi}^k + e^{k+1T}\,\bar{\eta}^k + g^T y^k$$

$$\leqslant \sum_{i=1}^{p} f_i(w_i^k) + d^{kT}\xi^k + e^{kT}\eta^k + g^T y^k = z^k$$

■

3. Convergence

In order to explore the convergence properties of the algorithm, we need the following lemma.

Lemma 3. There are only a finite number of δ^k having the same value.

Proof: Noting that $\delta^{k+1} = \delta^k$ if and only if one is at step 4 and otherwise $\delta^{k+1} < \delta^k$, it is sufficient to prove that the interation between step 2 and step 4 with the same value δ^k can be repeated only finitely often. Suppose that $\delta^{k-1} = \delta^k = \delta^{k+1}$. By lemma 1, transformation II does not change the corresponding objective function value. By step 1, we can assert that the optimal objective value z^{k+1} is strictly less than z^k; otherwise $(\bar{\xi}^k, \bar{\eta}^k, y^k)$ will solve P^{k+1} and $\delta^{k+1} = \frac{1}{2}\delta^k$. This is in contradiction to our assumption $\delta^k = \delta^{k+1}$.

So if $\delta^k = \delta^{k+1} = \delta^{k+2} = \ldots$ then P^{k+1}, P^{k+2}, . . . would all be different approximating programs. On the other hand, the transformation II with the same δ_k, starting from some x^k can generate as we shall see only a finite number of different two segment approximations for $f_i(\cdot)$. As a matter of fact, the junction point of two segments together with δ^k uniquely decides the approximating program. But starting from x_i^k, the transformation II with the same δ^k has only the following finite number of possible junction points.

$$x_i^k - t_i\delta^k \ , \ x_i^k - (t_i - 1)\delta^k \ , \ \ldots \ , \ x_i^k - \delta^k, \ x_i^k + \delta^k \ , \ \ldots \ ,$$

$$x_i^k + s_i\delta^k \ ,$$

$$l_i \ , \ l_i + \delta^k \ , \ \ldots \ , \ l_i + r_i\delta^k \ ,$$

$$u_i \ , \ u_i - \delta^k \ , \ \ldots \ , \ u_i - r_i\delta^k \ .$$

where $t_i = \left\lfloor \frac{x_i^k - l_i}{\delta^k} \right\rfloor$, $s_i = \left\lfloor \frac{u_i - x_i^k}{\delta^k} \right\rfloor$, $r_i = \left\lfloor \frac{u_i - l_i}{\delta^k} \right\rfloor$ are integers.

This completes the proof. ∎

From lemma 3, it is easy to obtain the following theorem.

Theorem 1. For any real number $\varepsilon > 0$ the algorithm must be terminated in a finite number of iterations.

Next if ε is taken to be a non-Archimedean infinitesimal, the stopping criteria would never be satisfied. Therefore, the algorithm would produce an infinite sequence $\{x^k, y^k\}$. Since $l \leqslant x^k \leqslant u$, $\underline{y} \leqslant y^k \leqslant \bar{y}$, the $\{x^k, y^k\}$ must have a cluster point. As we will show next, if all $f_i(\cdot)$ are differentiable then every cluster point of $\{x^k, y^k\}$ is an optimal solution of the program P. To do this we first require the following lemma.

Lemma 4. If all $f_i(\cdot)$ are differentiable and $\{x^k, y^k\}$ is the infinite sequence generated by the procedure, then there exists a subsequence which converges to an optimal solution of program P.

Proof: By lemma 3, since only a finite number of δ^k have the same value, there exists an infinite subsequence $\{x^{n_k}, y^{n_k}\}$ such that $\delta^{n_k+1} = \frac{1}{2}\delta^{n_k}$. Without loss of generality we can simply assume that

$$\{x^{n_k}\} \longrightarrow x^* \ , \ \{y^{n_k}\} \longrightarrow y^* \ , \ \delta^{n_k} \longrightarrow 0 \qquad (k \longrightarrow \infty)$$

If $x_i^* > l_i$, then for k big enough, there are

$$x_i^{n_k} > l_i \ , \ \alpha_i^{n_k} > 0$$

and

$$d_i^{n_k} = \left[f_i(w_i^{n_k} + \alpha_i^{n_k}) - f_i(w_i^{n_k}) \right] / \alpha_i^{n_k}$$

$$= f_i'(\theta_i^{n_k})$$

where $w_i^{n_k} \leqslant \theta_i^{n_k} \leqslant w_i^{n_k} + \alpha_i^{n_k}$.

Hence

$$| \theta_i^{n_k} - x_i^* | \leqslant | \theta_i^{n_k} - x_i^{n_k} | + | x_i^{n_k} - x_i^* |$$

$$\leqslant 2\delta^{n_k} + | x_i^{n_k} - x_i^* | \longrightarrow 0 \quad \text{as } k \longrightarrow \infty .$$

Because $f_i(\cdot)$ is a convex function as well as differentiable, $f_i'(\cdot)$ must be continuous (pp. 246, corollary 25.5 [5]), so

$$d_i^{n_k} \longrightarrow f_i'(x_i^*) \quad \text{as } k \longrightarrow \infty.$$

Similarly, if $x_i^* < u_i$,

$$e_i^{n_k} \longrightarrow f_i'(x_i^*) \quad \text{as } k \longrightarrow \infty.$$

Since the linear programs $\{P^k\}$ have only finitely many different bases, some infinite subsequence necessarily has the same optimal basis B. Without loss of generality we can assume that our subsequence $\{P^{n_k}\}$ is one of these.

Denote the objective function coefficients of P^k corresponding to basis B as C_B^k and $(f_1'(x_1^*), \ldots, f_p'(x_p^*), g_1, \ldots, g_p)$ as C_B.

If $d_i^{n_k}$ is one of $C_B^{n_k}$ for all k, then ξ_i is a basic variable and $\alpha_i^k > 0$. Using the above argument, one can prove that $d_i^{n_k} \longrightarrow f_i'(x_i^*)$. Similarly, if $e_i^{n_k}$ is one of $C_B^{n_k}$ for all k, we have that $e_i^{n_k} \longrightarrow f_i'(x_i^*)$. In conclusion

$$C_B^{n_k} \longrightarrow C_B \qquad \text{as } k \longrightarrow \infty .$$

Since $\delta^{n_k+1} = \frac{1}{2}\delta^{n_k}$, $J_1^{n_k} \cup J_2^{n_k} = \phi$. And, since B is optimal, we have

$$d_i^{n_k} \leqslant C_B^{n_k} B^{-1} a_i$$

$$e_i^{n_k} \geqslant C_B^{n_k} B^{-1} a_i$$

Finally, for $x_i^* > d_i$, $f_i'(x_i^*) \leqslant C_B B^{-1} a_i$

$$x_i^* < u_i \ , \ f_i'(x_i^*) \geqslant C_B B^{-1} a_i$$

Thus

$$f_i'(x_i^*) - C_B B^{-1} a_i \geqslant 0 \quad , \quad \text{if } x_i^* = l_i$$

$$f_i'(x_i^*) - C_B B^{-1} a_i \leqslant 0 \quad , \quad \text{if } x_i^* = u_i$$

$$f_i'(x_i^*) - C_B B^{-1} a_i = 0 \quad , \quad \text{if } l_i < x^*_i < u_i$$

For y* we have similar relations with g. These relations demonstrate that (x*,y*) satisfies the Kuhn-Tucker condition for program P. Because P is a convex program, its Kuhn-Tucker point is also its optimal point.

■

Theorem 2. Under the same assumption as in lemma 4, every cluster point of $\{x^k, y^k\}$ is an optimal solution of P.

Proof: Let

$$f_i^k(x_i) = \begin{cases} f_i(w_i^k) + d_i^k(x_i - w_i^k) & , \ w_i^k \leqslant x_i \leqslant w_i^k + \alpha_i^k \\ f_i(w_i^k + \alpha_i^k) + e_i^k(x_i - (w_i^k + \alpha_i^k)) & , \ w_i^k + \alpha_i^k \leqslant x_i \leqslant \\ & \leqslant w_i^k + \alpha_i^k + \beta_i^k \end{cases}$$

By definition of the d_i^k , e_i^k , recalling $x_i^k = w_i^k + \xi_i^k + \eta_i^k$, it is easy to verify that

$$f_i^k(x_i^k) = f_i(w_i^k) + d_i^k \xi_i^k + e_i^k \eta_i^k$$

Therefore, the optimal objective function value of P^k is

$$z^k = \sum_{i=1}^{p} f_i^k(x_i^k) + g^T y^k$$

Since $f_i(\cdot)$ is a convex function defined on an interval containing $[l_i,u_i]$, there exists a constant L_i such that

$$| f_i(x_i^1) - f_i(x_i^2) | \leq L_i | x_i^1 - x_i^2 | \qquad \forall\ x_i^1 , x_i^2 \in [l_i,u_i]$$

(pp. 237, Thm. 24.7 [5]).

If $\eta_i^k = 0$, $\xi_i^k > 0$, then $\alpha_i^k > 0$ and

$$|f_i^k(w_i^k) - f_i(x_i^k)| \leq |f_i(w_i^k) - f_i(x_i^k)| + \left|\frac{f_i(w_i^k + \alpha_i^k) - f_i(w_i^k)}{\alpha_i^k}(x_i^k - w_i^k)\right|$$

$$\leq 2L_i|x_i - w_i^k|$$

$$\leq 2L_i\delta^k$$

If $\eta_i^k = 0$, $\xi_i^k = 0$, then

$$|f_i^k(x_i^k) - f_i(x_i^k)| = 0 \leq 2L_i\delta^k$$

By an analogous argument we can show that if $\eta_i^k > 0$

$$|f_i^k(x_i^k) - f_i(x_i^k)| \leq 2L_i\delta^k \ .$$

Suppose that there is a subsequence $\{x^{\bar{n}k},y^{\bar{n}k}\}$ such that

$$(x^{\bar{n}k},y^{\bar{n}k}) \longrightarrow (\bar{x},\bar{y}) \quad \text{as } k \longrightarrow \infty$$

and that $\sum_{i=1}^{p} f_i(\bar{x}_i) + g^T\bar{y} > \sum_{i=1}^{p} f_i(x_i^*) + g^T y^*$

Let $$\Delta = \left(\sum_{i=1}^{p} f_i(\bar{x}_i) + g^T\bar{y}\right) - \left(\sum_{i=1}^{p} f_i(x_i^*) + g^T y^*\right) > 0$$

$$L = \max_i \{L_i\}$$

There is an N such that

$$\left|\left(\sum_{i=1}^{p} f_i(x_i^*) + g^T y^*\right) - \left(\sum_{i=1}^{p} f_i(x_i^{n_k}) + g^T y^{n_{ki}}\right)\right| < \frac{\Delta}{8}, \quad \forall\ k1 > N$$

$$\left|\left(\sum_{i=1}^{p} f_i(\bar{x}_i) + g^T\bar{y}\right) - \left(\sum_{i=1}^{p} f_i(x_i^{\bar{n}_k}) + g^T y^{\bar{n}_k}\right)\right| < \frac{\Delta}{8}, \quad \forall\ k2 > N$$

$$0 < \delta^{n_{k1}} < \frac{\Delta}{16\ p \cdot L}, \quad \forall\ k1 > N$$

$$0 < \delta^{\bar{n}_{k2}} < \frac{\Delta}{16\ p \cdot L}, \quad \forall\ k2 > N$$

Hence

$$|z^{n_{k1}} - z^{\bar{n}_{k2}}| = \left|\left[z^{n_{k1}} - \left(\sum_{i=1}^{p} f_i(x_i^{n_{k1}}) + g^T y^{n_{k1}}\right)\right]\right.$$

$$+ \left[\left(\sum f_i(x_i^{n_{k1}}) + g^T y^{n_{k1}}\right) - \left(\sum f_i(x_i^*) + g^T y^*\right)\right] - \Delta$$

$$+ \left[\left(\sum f_i(\bar{x}_i) + g^T\bar{y}\right) - \left(\sum f_i(x_i^{\bar{n}_{k2}}) + g^T y^{\bar{n}_{k2}}\right)\right]$$

$$\left. + \left[\left(\sum f_i(x_i^{\bar{n}_{k2}}) + g^T y^{\bar{n}_{k2}}\right) - z^{\bar{n}_{k2}}\right]\right|$$

$$\geq \Delta - \frac{\Delta}{8} - \frac{\Delta}{8} - \frac{\Delta}{8} - \frac{\Delta}{8} = \frac{\Delta}{2},$$

$$\forall\ k1 > N,\ k2 > N$$

By lemma 2, $\{z^k\}$ is a non-increasing sequence. From lemma 4, it has a subsequence that has a finite limit. Therefore $\{z^k\}$ itself converges to a finite point. But this contradicts the above inequality. The contradiction shows $(\bar{x},\bar{y})$ must be an optimal solution of P.

■

Theorem 3. Under the same assumptions as in Theorem 2, if $f_i(\cdot)$ is a strictly convex function and (x^*,y^*), $(\bar{x},\bar{y})$ are both cluster points of $\{x^k,y^k\}$, then

$$x_j^* = \bar{x}_j$$

Proof: By Theorem 2

$$\sum f_i(x_i^*) + g^T y^* = \sum f_i(\bar{x}_i) + g^T \bar{y}$$

Since $(\alpha\bar{x} + (1 - \alpha)x^* , \alpha\bar{y} + (1 - \alpha)y^*)$ is also a feasible solution of P and $f_i(\cdot)$ are convex, (x^*,y^*) is an optimal solution of P

$$\begin{aligned}\sum f_i(x_i^*) + g^T y^* &\leq \sum f_i(\alpha\bar{x}_i + (1 - \alpha)x_i^*) + g^T(\alpha\bar{y} + (1 - \alpha)y^*) \\ &\leq \alpha\left(\sum f_i(\bar{x}_i) + g^T\bar{y}\right) + (1 - \alpha)\left(\sum f_i(x_i^*) + g^T y^*\right) \\ &= \sum f_i(x_i^*) + g^T y^* \quad .\end{aligned}$$

Therefore, there must hold,

$$f_i(\alpha\bar{x}_i + (1 - \alpha)x_i^*) = \alpha f_i(\bar{x}_i) + (1 - \alpha)f_i(x_i^*) ,$$

$$\text{where} \quad 0 < \alpha < 1 \quad , \quad i = 1,\ldots,P$$

Since $f_i(\cdot)$ is strictly convex, we must have

$$\bar{x}_j = x_j^*$$

■

Corollary: If the objective function of P is strictly convex, then the infinite sequence generated by the algorithm converges to the (unique) optimal solution.

4. Computational Aspects

We have developed a FORTRAN Code to implement the two segment algorithm described in section 2. Based on the network constraints, some problems with up to 169 nonlinear variables have been computed on the CDC-750/130 at the University of Texas at Austin. The computational results are quite encouraging. For example, it took only 0.08 to 0.14 seconds CPU time to solve the 15 nonlinear variable problem A of Meyer in [4].

More generally, since our algorithm does not receive an priori break point set, the preprocessor work of the linearization is no longer neccessary. Again, for ease of proof of convergence in section 3, we stipulated that if a variable hit an artificial bound the length of the segment would remain the same whereas if not, the length would be halved. Our computational experience, however, shows that an alternative rule yields faster convergence. That is, whenever a variable hits an artificial bound, the length of the segment is expanded by a factor of 1.25; otherwise, the segment length is shrunk by a factor of 0.4. This experience confirms that of Meyer's in [4].

Problem 1: This problem is test problem A in [4].

$$\min \sum_{i=1}^{15} w_i(1-q_i)^{x_i}/1000$$

$$\text{s.t.} \sum_{i=1}^{10} x_i = 75{,}000$$

$$\sum_{i=6}^{15} x_i = 67{,}000$$

$$0 \leq x_i \leq u_i \qquad (i = 1,\ldots,15)$$

where w_i, q_i, u_i are constant as in the following table.

i	w_i	q_i	u_i
1	9.2	0.31	16000
2	1.0	0.45	16000
3	7.6	0.23	18000
4	0.6	0.09	10000
5	8.8	0.15	10000
6	4.2	0.21	11000
7	3.2	0.15	17000
8	3.4	0.01	20000
9	8.8	0.79	16000
10	6.6	0.41	15000
11	1.2	0.71	17000
12	4.6	0.77	12000
13	0.8	0.79	13000
14	3.0	0.21	20000
15	1.2	0.07	20000

The problem was solved using different initial lengths Δ_0 and terminal lengths Δ_T of the segment. The following results were obtained.

Δ_o	Δ_T	# of iterations	objective value
10000	100	16	7.738248
8000	100	17	7.738436
8000	50	19	7.738285
6000	100	16	7.738249
4096	100	15	7.738428
1024	100	16	7.738428
1024	1	22	7.738428

Comparing these to a lower bound on the objective value of 7.738140, the results are accurate to four significant figures.

Problem 2: The second example computed is a nonlinear congestion network model with 169 nonlinear function elements for planning the internal movements in the Hajj. The Hajj is the annual pilgrimage involving two million Muslims seeking to achieve one of the five pillars of Islam. The model projects optimal traffic flow during the 12 days of the Hajj in order to make the internal traffic in performance of the rituals of the Hajj as smooth as possible. The network contains 77 nodes and 169 arcs. The measure of traffic congestion on the (i,j)th arc is

$$f(x_{ij}) = \frac{1}{U_{ij} - X_{ij}}$$

where U_{ij} is the capacity of arc (i,j) and X_{ij} is its flow. The objective is to minimize the sum of the congestion over all arcs. The detailed

$$= f_i(\hat{x}_i)$$

$$= f_i(\bar{w}_i) + \bar{d}_i \xi_i' + \bar{e}_i \eta_i'$$

description of the model can be found in [3].

With initial length of segment 50,000 and terminal length 1,000, after 24 iterations (2.85 per second CPU time) a solution was obtained. It is of course far more accurate than that of the approximate once and for all piecewise linearization of [3].

Although the two examples are hardly adequate to make responsible projections, note further that the increase from 15 to 169 nonlinear variables involved a substantially less than quadratic increase in computer time.

References

[1] A. Charnes and W.W. Cooper, Management Models and Industrial Applications of Linear Programming, Vols. I and II, J. Wiley & Sons, Inc., New York, 1961.

[2] A. Charnes and W.W. Cooper, "Nonlinear Network Flows and Convex Programming over Incidence Matrices," Naval Research Logistics Quarterly, Vol. 5, pp. 231-240, 1958.

[3] A. Charnes, S. Duffuaa, and A. Yafi, "A Network Model for Planning the Internal Movements in the Hajj," Center for Cybernetic Studies CCS Report 455, The University of Texas at Austin, 1983

[4] R.R. Meyer, "Two-Segment Separable Programming," Management Science, Vol. 25, No. 4, April 1979, pp. 385-395.

[5] R.T. Rockafellar, Convex Analysis, Princeton University Press, 1970.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CCS 476	2. GOVT ACCESSION NO. AD-A146 215	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A Two-Segment Approximation Algorithm for Separable Convex Programming with Linear Constraints		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* I. Ali, A. Charnes, T. Song		8. CONTRACT OR GRANT NUMBER*(s)* N00014-81-C-0236 N00014-82-K-0295
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Cybernetic Studies The University of Texas at Austin Austin, TX 78712		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research (Code 434) Washington, D.C.		12. REPORT DATE January 1984 13. NUMBER OF PAGES 24
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

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

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Separable convex programming, Two-segment approximation

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

We discuss a new algorithm for the separable convex programming with linear constraints. This is based on the approximation of the objective function by at most two linear pieces in the neighborhood of the current feasible solution. The two segments will be adaptively defined rather than predecided fixed grids. If, furthermore, the objective function is differentiable, and one introduces a non-Archimedean infinitesimal, the algorithm generates a sequence of feasible solutions every cluster point of which is an optimal solution. Computational tests on the problem with up to 196 non-linear variables is presented.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*