6-1-1969

A Fixed Point Algorithm for Approximating the Optimal Solution of a Concave Programming Problem

Terje Hansen

COWLES FOUNDATION FOR RESEARCH IN ECONOMICS

AT YALE UNIVERSITY

Box 2125, Yale Station
New Haven, Connecticut

COWLES FOUNDATION DISCUSSION PAPER NO. 277

A FIXED POINT ALGORITHM FOR APPROXIMATING THE OPTIMAL SOLUTION OF A CONCAVE PROGRAMMING PROBLEM

Terje Hansen

June 19, 1969

A FIXED POINT ALGORITHM FOR APPROXIMATING THE OPTIMAL SOLUTION OF A CONCAVE PROGRAMMING PROBLEM*

by

Terje Hansen**

Introduction

In a series of papers Hansen [1], Kuhn [3] and Scarf [5, 6, 7] have described some applications of a combinatorial algorithm, including the approximation of a fixed point of a continuous mapping. A survey of applications has been given in [2]. The purpose of the present paper is to present and illustrate by numerical examples the application of this combinatorial algorithm to the approximation of the optimal solution of a concave programming problem.

The algorithm is based upon the technique devised by Lemke and Howson [4] for the numerical calculation of a Nash equilibrium point for a two person non-zero sum game and on the concept of a primitive set.

*I am indebted to Herbert Scarf for reading a preliminary draft of this paper and making valuable comments. The research described in this paper was carried out under grants from the National Science Foundation and from the Ford Foundation.

**Norwegian School of Economics and Business Administration.

The Concept of a Primitive Set

In order to review the definition of a primitive set,[1] let

$$\Pi = \{\pi^1, \ldots, \pi^d\}$$

comprise all vectors of the form $(k_1/D, \ldots k_n/D)'$, such that the k's and D are integers and $-1 \leq k_i \leq D+1$ with the k's summing to D.

Definition: Let $x = (x_1, \ldots x_n)$ and $y = (y_1, \ldots y_n)$. We define x_i to be larger than y_i, $x_i \underset{i}{>} y_i$, if and only if

$(x_i, \ldots, x_n, x_1, \ldots, x_{i-1})$ is lexicographically larger than

$(y_i, \ldots, y_n, y_1, \ldots, y_{i-1})$. Observe that for x_i to equal y_i by this ordering we must have $x = y$.

In formulating the definition of a primitive set it is convenient to adopt the convention that if $x, y, \ldots z$ are vectors, then by

$$\min_i(x_i, y_i, \ldots z_i)$$

we understand the smallest i^{th} coordinate according to the above ordering.

Definition: A set of n distinct vectors $\pi^{j_1}, \ldots, \pi^{j_n}$ in Π is defined to be a primitive set if there are no vectors $\pi^j \in \Pi$ with

---

[1] The definition of a primitive set that is used in this paper differs somewhat from the one used in [2].

$$\pi_i^j > \min_i(\pi_i^{j_1}, \ldots, \pi_i^{j_n})$$

for all i.

Given any n vectors in Π let us form a matrix K whose columns are the numerators of these vectors

$$K = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ \vdots & & & \\ k_{n1} & k_{n2} & \cdots & k_{nn} \end{bmatrix}$$

We may prove the following theorem[1] which gives necessary and sufficient conditions for such a matrix to represent a primitive set.

<u>Theorem 1.</u> Let K be an $n \times n$ matrix with integral entries and such that $-1 \leq k_{ij} \leq D+1$ for all i and j and $\sum_i k_{ij} = D$ for all j. The columns represent the vectors of a primitive set, if and only if there is a permutation $I(j)$ of the integers $(1, 2, \ldots, n)$, and a rearrangement of the columns of K such that the j^{th} column of K is identical with column $j-1$, excepting the entry in row $I(j)$ which is one unit smaller than the entry in row $I(j) - 1$ which is one unit larger. If $j = 1$, $j-1$ is to be interpreted as n, and similarly for $I(j)$.

As an example of columns of

---

[1]The proof of this theorem follows from the proof of Theorem 3 in [2].

$$\begin{bmatrix} 10 & 10 & 10 & 11 & 11 \\ 20 & 20 & 21 & 20 & 20 \\ 30 & 31 & 30 & 30 & 30 \\ 10 & 9 & 9 & 9 & 10 \\ 30 & 30 & 30 & 30 & 29 \end{bmatrix}$$

with $D = 100$ satisfy the conditions of the theorem if the permutation is given by

j	I(j)
1	1
2	4
3	3
4	2
5	5

The basic combinatorial theorem to be applied in this paper and whose proof follows from the discussion in [1] and [2] may now be stated.

<u>Theorem 2.</u> Let

$$A = \begin{bmatrix} a_{11}, & \cdots & a_{1d} \\ a_{n1}, & \cdots & a_{nd} \end{bmatrix}$$

be a matrix and $b = (b_1, \ldots b_n)'$ a non-negative vector, such that the j^{th} column of A is associated with the j^{th} vector in Π. Specifically we assume that if $\pi_i^j < 0$ for some i; say the first coordinate that is strictly negative is the p^{th}, then the j^{th} column of A consists of $n-1$ zeros, and a single 1, with the entry 1 located in the p^{th} row. Assume that the set of non-negative vectors x satisfying, $Ax = b$, is bounded. Then there exists a primitive set $\pi^{j_1}, \ldots \pi^{j_n}$, so that the columns $j_1, \ldots j_n$ form a feasible basis for $Ax = b$.

The proof of Theorem 2 also provides us with an algorithm supplying a primitive set $\pi^{j_1}, \ldots \pi^{j_n}$, so that the columns $j_1, \ldots j_n$ form a feasible basis for $Ax = b$.

The Optimal Solution of a Concave Programming Problem

Consider the concave programming problem given by

$$\max g(z)$$

$$\text{subject to} \quad f_k(z) \leq 0 \qquad k = 1, \ldots, m$$

$$z \geq 0$$

$g(z)$ is assumed to be concave and the functions $f_k(z)$ are assumed to be convex. All functions are assumed to be twice differentiable. For technical reasons that will become apparent later z_1 is a dummy variable such that

$$\frac{\partial g}{\partial z_1} = \frac{\partial f_1}{\partial z_1} = \ldots = \frac{\partial f_m}{\partial z_1} = 0$$

for any z.

We shall assume that the feasible region is bounded in the sense that no non-negative vector z with $z_1 = 0$ and whose coordinates sum to q, with q a positive number, is in the feasible region. Furthermore we assume that there exists a non-negative vector z^0 whose coordinates sum to q satisfying $f_k(z^0) < 0$ for all k. This is the familiar constraint qualification typically assumed in non-linear programming problems.

Now a necessary and sufficient condition for z^* to represent the optimal solution of the above programming problem is that there exists a non-negative vector λ^* such that

$$\frac{\partial g}{\partial z_i^*} - \Sigma\lambda_k^* \frac{\partial f_k}{\partial z_i^*} \leq 0$$

with equality if $z_i^* > 0$, and

$$f_k(z^*) \leq 0$$

with equality if $\lambda_k^* > 0$.

By an approximation of the optimal solution of a concave programming problem we shall understand a pair of non-negative vectors $(\hat{z}, \hat{\lambda})$ such that

$$\frac{\partial g}{\partial \hat{z}_i} - \Sigma\hat{\lambda}_k \frac{\partial f_k}{\partial \hat{z}_i}$$

is "close" to 0 if $\hat{z}_i > 0$ and is non-positive if $\hat{z}_i = 0$, and

$$f_k(\hat{z})$$

is "close" to 0 if $\hat{\lambda}_k > 0$ and is non-positive if $\hat{\lambda}_k = 0$.

A survey of non-linear programming algorithms is given in [8]. One can describe the present state of non-linear programming succinctly by saying that no efficient algorithm exists for general concave programming problems. As far as the gradient method is concerned the presence of non-linear constraints considerably complicates the algorithm as well as reducing its efficiency. The decomposition procedure requires the maximization of functions of the form

$$g(z) = u_0 - \sum_k u_k f_k(z)$$

with u_k $(k = 0, \ldots m)$ non-negative numbers, which may itself be difficult.

A Fixed Point Algorithm for Approximating the Optimal Solution of a Concave Programming Problem

In order to apply Theorem 2 to approximate the optimal solution of a concave programming problem, we shall define a correspondence between the vectors in Π and the columns of A . We shall distinguish between 3 cases.

Case 1. $\pi_i^j < 0$ for some i . If the first coordinate that is strictly negative is the p^{th} coordinate then the j^{th} column of A consists of $n-1$ zeros, and a single 1, with the entry one located in the p^{th} row.

<u>Case 2.</u> $\pi_i^j \geq 0$ for all i and $f_k(z^j) > 0$, with $z^j = q \cdot \pi^j$, for some k. If the first constraint that is violated is the p^{th} then

$$A_j = \begin{bmatrix} -\frac{\partial f_p}{\partial z_1^j} + 1 \\ \vdots \\ -\frac{\partial f_p}{\partial z_n^j} + 1 \end{bmatrix}$$

<u>Case 3.</u> $\pi_i^j \geq 0$ for all i and $f_k(z^j) \leq 0$, with $z^j = q \cdot \pi^j$, for all k. Then

$$A_j = \begin{bmatrix} \frac{\partial g}{\partial z_1^j} + 1 \\ \vdots \\ \frac{\partial g}{\partial z_n^j} + 1 \end{bmatrix}$$

If the vector $b = (1, \ldots 1)'$, the hypotheses of Theorem 2 are clearly satisfied since by assumption

$$\frac{\partial g}{\partial z_1} = \frac{\partial f_1}{\partial z_1} = \ldots = \frac{\partial f_m}{\partial z_1} = 0 ,$$

implying that all the columns associated with the derivatives of the functions $g(z)$ and $f_k(z)$ have an entry 1 in the first row.

We therefore conclude that there exists a primitive set $\pi^{j_1}, \ldots \pi^{j_n}$, such that the columns $j_1, \ldots j_n$ form a feasible basis for $Ax = b$. The algorithm underlying Theorem 2 will therefore provide us with a non-negative solution to the equations $Ax = b$ which we may write

$$u_i + \Sigma y_j \left(\frac{\partial g}{\partial z_i^j} + 1 \right) + \Sigma x_j a_{ij} = 1$$

with a_j a column of A associated with the derivatives of the f functions and y_j and $x_j > 0$ only for the columns corresponding to the non-negative vectors in the primitive set. Finally $u_i > 0$ only if

$$\min(\pi_i^{j_1}, \ldots \pi_i^{j_n}) < 0 \ .$$

Suppose that $\Sigma x_j + \Sigma y_j = 1$, which is true unless $u_1 > 0$, and also that $\Sigma y_j > 0$. Let $\hat{\pi}$ be a convex combination of the non-negative vectors in the final primitive set such that the vector, $\hat{z} = q \cdot \hat{\pi}$, satisfies the constraints.[1] Further let $x_i(k)$ refer to that x_i associated with the derivatives of $f_k(z^i)$. Define a vector $\hat{\lambda}$ with

$$\hat{\lambda}_k = \frac{\Sigma x_i(k)}{\Sigma \ y_j} \ .$$

---

[1]One such convex combination, which is used in the subsequent numerical examples, is given by

Then $\hat{z}$ and $\hat{\lambda}$ may be taken as an approximation to the optimal solution of the concave programming problem.

Rather than providing precise bounds for the degree of approximation we shall show that as D passes to the limit a subsequence may be selected with the above equations reducing to the Kuhn-Tucker conditions.

We therefore let D pass to the limit and select a subsequence with the u_i's, the Σy_j and $\Sigma x_i(k)$ and the non-negative vectors in the primitive set all converging to u_i, y, x_k and π^* respectively.[1] The above equations then reduce to

$$\hat{z} = q \circ \sum_{i;\, \pi^{j_i} \geq 0} \hat{\alpha}_i \pi^{j_i}$$

with the $\hat{\alpha}_i$'s the optimal solution of the linear programming problem

$$\max \sum_{i;\, \pi^{j_i} \geq 0} \alpha_i g(q \cdot \pi^{j_i})$$

$$\text{subject to} \quad \sum_{i;\, \pi^{j_i} \geq 0} \alpha_i f_k(q \cdot \pi^{j_i}) \leq 0 , \quad k = 1, \ldots m$$

$$\sum_{i;\, \pi^{j_i} \geq 0} \alpha_i = 1$$

$$\alpha > 0$$

The problem obviously has a feasible solution since for some i, $z^{j_i} = q \cdot \pi^{j_i}$, satisfies all the constraints. From the convexity assumption we finally have that $f_k(\hat{z}) \leq 0$ for all k.

[1]An implication of the definition of a primitive set is that the difference between the smallest and largest i^{th} coordinate of the n vectors in the primitive set is equal to $1/D$, for all i, i.e. as D tends to infinity the vectors in any primitive set all converge to a common vector.

$$u_i + y\left(\frac{\partial g}{\partial z_i^*} + 1\right) + \Sigma x_k\left(\frac{\partial f_k}{\partial z_i^*} + 1\right) = 1$$

where $z^* = q \cdot \pi^*$, and y is the limit of $\Sigma\ y_j$.

If we recall the construction of the matrix A we have that $y > 0$ implies that the vector z^* satisfies all the constraints and that $x_k > 0$ implies that $f_k(z^*) \geq 0$. We shall argue that $y > 0$ and that $y + \Sigma x_k = 1$; the desired result will then follow. By assumption the vector z^0 whose coordinates sum to q , satisfies all the constraints. Now if $y = 0$ and we multiply the i^{th} equation by $z_i^0 - z_i^*$, and sum, we obtain

$$\Sigma u_i(z_i^0 - z_i^*) + \Sigma x_k \Sigma(z_i^0 - z_i^*)\left(-\frac{\partial f_k}{\partial z_i^*}\right) = \Sigma(z_i^0 - z_i^*) = 0$$

This will lead to a contradiction if we can show that both terms on the left hand side are non-negative, with the second strictly positive. But the first term is surely non-negative, since if $u_i > 0$ then $\pi_i^* = z_i^* = 0$. In order to see that the second term is positive, we observe that if $x_k > 0$ (which must be true for at least one k), then $f_k(z^*) \geq 0$, and from the convexity assumption

$$0 > f_k(z^0) - f_k(z^*) \geq \Sigma(z_i^0 - z_i^*) \frac{\partial f_k}{\partial z_i^*} .$$

This demonstrates that $y > 0$, and as a consequence, that all of the constraints are satisfied by z^*.

To complete the argument observe that $\Sigma x_k + y = 1$ unless $u_1 > 0$. If $u_1 > 0$, however, $\pi_1^* = z_1^* = 0$ and by assumption z^* does not satisfy all of the constraints, a contradiction.

If we define $\lambda_k^* = x_k/y$, then

$$\frac{\partial g}{\partial z_i^*} - \Sigma\lambda_k^* \frac{\partial f_k}{\partial z_i^*} \leq 0$$

with equality if $z_i^* > 0$ and

$$f_k(z^*) \leq 0$$

with equality if $\lambda_k^* > 0$.

A Numerical Example

To illustrate the working of the algorithm and to compare the approximation with the optimal solution we chose the quadratic programming problem

$$\max g(z) = -z'Bz$$

$$\text{subject to } f_k(z) \leq 0 \quad k = 1, \ldots m$$

$$z \geq 0$$

with B a positive semidefinite matrix

$$B = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.5 & 0.3 & 0.7 & 0.6 & 0.8 \\ 0.0 & 0.5 & 2.0 & 1.0 & 1.5 & 0.8 & 1.2 \\ 0.0 & 0.3 & 1.0 & 3.0 & 2.0 & 1.0 & 0.5 \\ 0.0 & 0.7 & 1.5 & 2.0 & 4.0 & 0.2 & 3.1 \\ 0.0 & 0.6 & 0.8 & 1.0 & 0.2 & 5.0 & 2.6 \\ 0.0 & 0.8 & 1.2 & 0.5 & 3.1 & 2.6 & 6.0 \end{bmatrix}$$

and $f_k(z)$ linear functions of z such that

$$f_1(z) = z_2 - z_3 - z_4 - z_5 - z_6 - z_7 + 1$$

$$f_2(z) = z_4 - z_5 - z_6 - z_7 + 0.5$$

$$f_3(z) = 0.2z_2 + 0.3z_3 + 0.4z_4 + 0.6z_5 + 0.2z_6 + 0.8z_7 - 0.5$$

$$f_4(z) = z_2 - 0.1$$

$g(z)$ and $f_k(z)$ $(k = 1, \ldots 4)$ obviously satisfy all the conditions of the algorithm.

The algorithm was run with 3 different values of D, namely $D = 100$, $D = 300$ and finally $D = 500$. The columns of the matrices K_{100}, K_{300} and K_{500} represent the numerators of the vectors in the final primitive set and are given below

$$K_{100} = \begin{bmatrix} 50 & 50 & 51 & 50 & 50 & 50 & 50 \\ 6 & 6 & 5 & 5 & 5 & 5 & 5 \\ 19 & 20 & 20 & 20 & 20 & 20 & 20 \\ 16 & 15 & 15 & 15 & 15 & 16 & 16 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 6 & 6 & 6 & 6 & 7 & 7 & 6 \\ 3 & 3 & 3 & 4 & 3 & 3 & 3 \end{bmatrix}$$

$$K_{300} = \begin{bmatrix} 150 & 151 & 150 & 150 & 150 & 150 & 150 \\ 16 & 15 & 15 & 15 & 15 & 15 & 16 \\ 60 & 60 & 60 & 60 & 60 & 60 & 59 \\ 45 & 45 & 45 & 45 & 46 & 46 & 46 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 18 & 18 & 18 & 19 & 19 & 18 & 18 \\ 11 & 11 & 12 & 11 & 11 & 11 & 11 \end{bmatrix}$$

$$K_{500} = \begin{bmatrix} 251 & 250 & 250 & 250 & 250 & 250 & 250 \\ 25 & 25 & 25 & 25 & 25 & 26 & 26 \\ 100 & 100 & 100 & 100 & 100 & 99 & 100 \\ 75 & 75 & 75 & 76 & 76 & 76 & 75 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 30 & 30 & 31 & 31 & 30 & 30 & 30 \\ 19 & 20 & 19 & 19 & 19 & 19 & 19 \end{bmatrix}$$

$\hat{z}$ and $\hat{\lambda}$ were then calculated as indicated above. The results are given below.

Comparison of the Optimal Solution
and the Approximation for Different Values of D

Text	True Value	Approximation D = 100	Approximation D = 300	Approximation D = 500
z_2	0.1000	0.1000	0.1000	0.1000
z_3	0.4000	0.4000	0.4000	0.4000
z_4	0.3005	0.3000	0.3000	0.3000
z_5	0.	0.	0.	0.
z_6	0.1221	0.1200	0.1200	0.1240
z_7	0.0774	0.0800	0.0800	0.0760
λ_1	2.6821	2.6821	2.6821	2.6821
λ_2	0.3024	0.3024	0.3024	0.3024
λ_3	0.	0.	0.	0.
λ_4	1.6315	1.6315	1.6315	1.6315
$g(z)$	-1.33506	-1.33512	-1.33512	-1.33510
$f_1(z)$	0.	0.	0.	0.
$f_2(z)$	0.	0.	0.	0.
$f_3(z)$	-0.1535	-0.1520	-0.1520	-0.1544
$f_4(z)$	0.	0.	0.	0.
h_2	0.	-0.0013	0.0013	0.0003
h_3	0.	-0.0019	-0.0019	0.0013
h_4	0.	0.0045	0.0045	0.0005
h_5	-0.0863	-0.0995	-0.0995	-0.0763
h_6	0.	0.0085	0.0085	-0.0107
h_7	0.	-0.0195	-0.0195	0.0077

h_i is defined as

$$h_i = \frac{\partial g}{\partial z_i} - \Sigma\lambda_k \frac{\partial f_k}{\partial z_i}$$

Some observations on D and q . Sometimes it may be difficult to tell a priori the values of q that should be used. If $u_1 > 0$ this suggests that q has been selected too small and should be increased. Similarly if D is specified too small the algorithm may terminate with $\Sigma y_j = 0$. D should then be increased and the algorithm rerun.

Finally some observations on computer time and number of iterations. The examples above were run on an IBM 1130 as well as on an IBM 360-50 computer. In the table below the number of vectors[1] in Π for the three examples and the number of iterations required for the algorithm to terminate are given.

Number of Vectors in Π and Number of Iterations
for Different Values of D

	D = 100	D = 300	D = 500
Number of Vectors in Π	$0.2526 \cdot 10^{10}$	$0.1245 \cdot 10^{13}$	$0.2459 \cdot 10^{14}$
Number of Iterations	820	2491	4121

Observe that the number of iterations required for the algorithms to terminate is approximately linearly related to D . The algorithm did approximately 300 iterations a minute on the IBM 1130 and 5000 iterations a minute on the IBM 360-50.

---

[1]The number of vectors in Π is equal to

$$\binom{D + 2n - 1}{n - 1} - n\binom{2n - 4}{n - 3}$$

REFERENCES

[1] Hansen, T., "On the Approximation of a Competitive Equilibrium," Ph.D. Dissertation, Yale University.

[2] Hansen, T. and H. Scarf, "On the Applications of a Recent Combinatorial Algorithm," Cowles Foundation Discussion Paper No. 272.

[3] Kuhn, H.W., "Simplicial Approximation of Fixed Points," Proceedings of the National Academy of Sciences, 61 (1968).

[4] Lemke, C.E. and J.T. Howson, "Equilibrium Points of Bi-Matrix Games," SIAM Journal, 12 (1964).

[5] Scarf, H., "The Core of an N Person Game," Econometrica, 35 (1967).

[6] ________, "The Approximation of Fixed Points of a Continuous Mapping," SIAM Journal of Applied Math., 15 (1967).

[7] ________, "On the Computation of Equilibrium Prices," in Ten Economic Studies in the Tradition of Irving Fisher, Wiley (1967).

[8] Wolfe, P., "An Outline of Nonlinear Programming," in Mathematics of the Decision Sciences, Part 1, American Math. Soc. (1968).