J. Austral. Math. Soc. **20** (Series B) (1978), 487–494

ON FINDING A POINT IN THE RELATIVE INTERIOR OF A POLYHEDRAL SET AND DEGENERACY IN GEOMETRIC PROGRAMMING

T. R. JEFFERSON and C. H. SCOTT

(Received 5 May 1978)

(Revised 11 September 1978)

Abstract

Geometric programming is now a well-established branch of optimization theory which has its origin in the analysis of posynomial programs. Geometric programming transforms a mathematical program with nonlinear objective function and nonlinear inequality constraints into a dual problem with nonlinear objective function and linear constraints. Although the dual problem is potentially simpler to solve, there are certain computational difficulties to be overcome. The gradient of the dual objective function is not defined for components whose values are zero. Moreover, certain dual variables may be constrained to be zero (geometric programming degeneracy).

To resolve these problems, a means to find a solution in the relative interior of a set of linear equalities and inequalities is developed. It is then applied to the analysis of dual geometric programs.

1. Introduction

It is common in mathematics to find problems of solving sets of linear equalities and inequalities. Most first year courses in linear algebra deal with the solution of linear equations. Orden [6] has dealt with the more general problem of solving linear equality and inequality systems. When some of the inequalities are also required to be strict the occurrences are less common but they are nevertheless important in certain analysis. Geometric programming is one such area.

https://doi.org/10.1017/S1446181100001814 Published online by Cambridge University Press

2. Geometric programming

Geometric programming originated with the analysis of a class of mathematical programs called posynomial programs [2]. This class of programs has the form:

$$\begin{aligned} &\mathit{min} \quad g_0(t) \\ &\mathit{subject\ to} \quad g_k(t) \leqslant 1, \quad k = 1, 2, \ldots, p. \end{aligned}$$

The decision vector $t \in R^m$, and each component is positive. The functions $g_k(t)$ have the form

$$\sum_{i \in [k]} c_i \prod_{j=1}^{m} t_j^{a_{ij}}, \quad k = 0, 1, \ldots, p.$$

The c_i are positive and the a_{ij} are arbitrary real constants. The index sets $[k]$, $k = 0, 1, \ldots, p$, form a sequential partition of the integers 1 to n, that is

$$[0] = \{1, \ldots, n_1\}, \quad [1] = \{n_1 + 1, \ldots, n_2\}, \ldots, [p] = \{n_p + 1, \ldots, n\}.$$

Thus the functions $g_k(t)$ have the form of polynomials but with non-integer powers and positive coefficients; hence the name posynomials. The example given in Section 4 is a typical posynomial program.

Geometric programming analysis generates a dual program which is potentially easier to solve. It is perhaps worth mentioning that the name geometric programming was chosen because of the importance of arithmetic-geometric inequality in the derivation of the dual mathematical program. The dual program is

$$\max \nu(\delta) = \prod_{i=1}^{n} (c_i/\delta_i)^{\delta_i} \prod_{k=1}^{p} \lambda_k^{\lambda_k}$$

subject to orthogonality conditions

$$\sum_{i=1}^{n} a_{ij} \delta_i = 0, \quad j = 1, 2, \ldots, m, \tag{1}$$

a normality condition

$$\sum_{i \in [0]} \delta_i = 1 \tag{2}$$

and non-negativity conditions

$$\delta_i \geqslant 0, \quad i = 1, 2, \ldots, n. \tag{3}$$

The dimensions m, n and constants c_i and a_{ij} are defined as before. The variables λ_k, $k = 1, 2, \ldots, p$, are defined by

$$\lambda_k = \sum_{i \in [k]} \delta_i, \quad k = 1, \ldots, p.$$

The sets $[k]$, $k = 0, 1, 2, \ldots, p$, are defined as before.

https://doi.org/10.1017/S1446181100001814 Published online by Cambridge University Press

At optimality the primal and dual variables are related by

$$c_i \prod_{j=1}^{m} t_j^{a_{ij}} = \begin{cases} \delta_i \nu(\delta), & i \in [0] \\ \delta_i / \lambda_k, & i \in [k], \quad \lambda_k > 0, \\ & k = 1, 2, \ldots, p \end{cases} \tag{4}$$

and the primal and dual objective functions are equal.

It can easily be seen that the dual program has a nonlinear objective function subject to linear equalities and inequalities. This problem is often easier to solve than the primal program. Moreover, equations (4) are linear in the variables $\log t_j$, thus providing an easy means of obtaining the primal optimal solution from the dual optimal solution.

There are certain difficulties in solving the dual programs. The gradient of $\log \nu(\delta)$ ($\log \nu(\delta)$ is easier to work with than $\nu(\delta)$) has components

$$\frac{\partial \log \nu(\delta)}{\partial \delta_i} = \begin{cases} \log \dfrac{c_i}{\delta_i} - 1, & i \in [0], \\[2ex] \log \dfrac{\lambda_k c_i}{\delta_i}, & i \notin [0]. \end{cases}$$

Should any particular $\delta_i = 0$, $\partial \log \nu(\delta)/\partial \delta_i = \infty$. To apply a gradient method to the problem we require a feasible δ such that $\delta_i > 0$, $i = 1, 2, \ldots, n$.

The other major problem is degeneracy. A geometric program is said to be degenerate when there does not exist a feasible δ with all components positive. Duffin, Peterson and Zener analyse degenerate geometric programs in Chapter 6 of [2]. Degenerate programs are equivalent to canonical programs where the terms, associated with components δ_i which cannot be made positive, are deleted.

Both problems can be resolved by the approach presented in the next section. An example of the method appears in Section 4.

3. Cascading simplex method

Let us consider the problem of finding a point which requires that all inequalities are satisfied strictly:

$$\left.\begin{aligned} Ax &= d, \\ Bx &< c, \\ x &> 0. \end{aligned}\right\} \tag{5}$$

https://doi.org/10.1017/S1446181100001814 Published online by Cambridge University Press

In order to solve (5) we relax the problem to one which we can handle by the simplex method:

$$\left.\begin{aligned} Ax &= d, \\ Bx + Ix_s &= c, \\ x \geqslant 0, \quad x_s &\geqslant 0, \end{aligned}\right\} \tag{6}$$

where x_s are slack variables.

In order to obtain a vector where the inequalities are satisfied strictly we will try to make x and x_s positive. We do this by first trying to solve (6) with an objective function

$$\begin{aligned} &max \quad \langle e, (x, x_s)\rangle, \\ &where \quad e = (1, 1, \ldots, 1) \quad initially. \end{aligned}$$

Phase I

Phase I is exactly the same as the standard simplex method [3]. We first try to obtain a feasible point for (6). Since (6) is a relaxed version of (5), if we should find that (6) is inconsistent it would imply that (5) is too. Hence suppose that (6) is consistent then at the end of phase I we will have a point which satisfies (6).

Phase II

Phase II is based on the observation that if one judiciously selects basic feasible points, a simple average will lie in the relative interior of the region (provided the region has a relative interior). In general we are looking for a point which is interior relative to only certain inequalities. We of course only need one basic feasible point to hold strictly with respect to a given inequality for the average to hold strictly. With this in mind we are ready to start phase II.

The objective function is designed to make those components positive that we wish to be positive. As soon as we find one basic feasible point for which a desired component is positive we set the associated component of the e vector to zero as one point is all we need. The steps are as follows.

Step 1. If $\langle e, (x, x_s)\rangle > 0$, go to step 2. Otherwise, go to step 3.

Step 2. Add (x, x_s) to the set of feasible points which we shall average later. For each component of (x, x_s) which is positive set the corresponding component of e to zero. Return to the tableau generated from phase I (this is to minimize the number of pivots on the matrix to obtain any basic feasible point). If e is a vector of zeros go to step 5. Otherwise, go to step 3.

https://doi.org/10.1017/S1446181100001814 Published online by Cambridge University Press

Step 3. Do a simplex pivot. If optimal (in the linear programming sense) and $\langle e,(x,x_s)\rangle = 0$, stop. An interior point cannot be obtained and for each component of e which is one the associated component of (x,x_s) cannot be made positive. If the solution of (6) is unbounded (in the linear programming sense) go to step 4. Otherwise go to step 1.

Step 4. In the case of the linear programming problem being unbounded, the column to be pivoted on was found to have non-positive elements. Thus the associated variable can take any positive value and still maintain feasibility. Let us assume the variable is x_j and that we fix its value at 1. To do this we delete the jth column from the simplex tableau and subtract this column from the basic feasible solution. Continue on with this adjusted tableau. Go to step 1.

4. Example

As an example we obtain a point with all components strictly positive for the dual of a geometric program. The problem is a multiphase chemical equilibrium developed by Dantzig, De Haven and Sams [1].

Primal formulation

$$\textit{min} \quad x_{11}^{-.00119}\, x_1^{-.001332}\, x_2^{-.002271}\, x_3^{-.002485}\, x_4^{-4.67}\, x_5^{-4.672}\, x_6^{-.00814}\, x_7^{-.008092}\, x_8^{-.005}$$

$$x_{10}^{-.00088}\, x_9^{-.000909}$$

subject to

$$.05418x_1+.02186x_2+.09773x_3+.006694x_5x_4\leqslant 1$$

$$10^{-6}x_1+10^{-5}x_2+10^{-6}x_3+10^{-10}x_4+10^{-8}x_5+10^{-3}x_6$$

$$+10^{-3}x_7+.109x_5x_6\leqslant 1,$$

$$.0001611x_5x_2+10^{-23}x_5x_2x_4+.193\times 10^{-5}x_5x_2x_4^{-1}$$

$$+10^{-4}x_{10}+10^{-6}x_1+10^{-5}x_2+10^{-6}x_3+10^{-10}x_4+10^{-8}x_5$$

$$+10^{-3}x_6+10^{-3}x_8+.109x_5x_4+.0001611x_5x_2$$

$$+10^{-23}x_5x_2x_4+.193\times 10^{-5}x_5x_2x_4^{-1}+10^{-5}x_9$$

$$+.0001118x_9x_1+10^{-4}x_{11}\leqslant 1.$$

This problem can be seen to be highly nonlinear.

https://doi.org/10.1017/S1446181100001814 Published online by Cambridge University Press

In the dual program, however, all the nonlinearity is in the objective function.

Dual formulation

$$\begin{aligned}
max\quad & \left(\frac{1}{\delta_1}\right)^{\delta_1}\left(\frac{.054181}{\delta_2}\right)^{\delta_2}\left(\frac{.02186}{\delta_3}\right)^{\delta_3}\left(\frac{.09773}{\delta_4}\right)^{\delta_4}\left(\frac{.006694}{\delta_5}\right)^{\delta_5}\left(\frac{10^{-6}}{\delta_6}\right)^{\delta_6}\left(\frac{10^{-5}}{\delta_7}\right)^{\delta_7}\\
& \left(\frac{10^{-6}}{\delta_8}\right)^{\delta_8}\left(\frac{10^{-10}}{\delta_9}\right)^{\delta_9}\left(\frac{10^{-8}}{\delta_{10}}\right)^{\delta_{10}}\left(\frac{10^{-3}}{\delta_{11}}\right)^{\delta_{11}}\left(\frac{10^{-3}}{\delta_{12}}\right)^{\delta_{12}}\left(\frac{.109}{\delta_{13}}\right)^{o_{13}}\left(\frac{.0001611}{\delta_{14}}\right)^{\delta_{14}}\\
& \left(\frac{10^{-23}}{\delta_{15}}\right)^{\delta_{15}}\left(\frac{.00000193}{\delta_{16}}\right)^{\delta_{16}}\left(\frac{10^{-4}}{\delta_{17}}\right)^{\delta_{17}}\left(\frac{10^{-6}}{\delta_{18}}\right)^{\delta_{18}}\left(\frac{10^{-5}}{\delta_{19}}\right)^{\delta_{19}}\left(\frac{10^{-6}}{\delta_{20}}\right)^{\delta_{20}}\left(\frac{10^{-10}}{\delta_{21}}\right)^{\delta_{21}}\\
& \left(\frac{10^{-8}}{\delta_{22}}\right)^{\delta_{22}}\left(\frac{10^{-3}}{\delta_{23}}\right)^{\delta_{23}}\left(\frac{10^{-3}}{\delta_{24}}\right)^{\delta_{24}}\left(\frac{.109}{\delta}\right)^{\delta_{25}}\left(\frac{.0001611}{\delta_{26}}\right)^{\delta_{26}}\left(\frac{10^{-23}}{\delta_{27}}\right)^{\delta_{27}}\\
& \left(\frac{.00000193}{\delta_{28}}\right)^{\delta_{28}}\left(\frac{10^{-5}}{\delta_{29}}\right)^{\delta_{29}}\left(\frac{.0001118}{\delta_{30}}\right)^{\delta_{30}}\left(\frac{10^{-4}}{\delta_{31}}\right)^{\delta_{31}}
\end{aligned}$$

subject to

$$\left.\begin{array}{r}
\delta_1 = 1.\\
-.00119\,\delta_1+\delta_{31}=0,\\
-.001332\,\delta_1+\delta_2+\delta_{18}+\delta_{30}=0,\\
-.002271\,\delta_1+\delta_3+\delta_7+\delta_{14}+\delta_{15}+\delta_{16}+\delta_{19}+\delta_{26}+\delta_{27}+\delta_{28}=0,\\
-.002485\,\delta_1+\delta_4+\delta_8+\delta_{20}=0,\\
-4.67\,\delta_1+\delta_5+\delta_9+\delta_{13}+\delta_{15}-\delta_{16}+\delta_{21}+\delta_{25}+\delta_{27}-\delta_{28}=0,\\
-4.672\,\delta_1+\delta_5+\delta_{10}+\delta_{13}+\delta_{14}+\delta_{15}+\delta_{16}+\delta_{22}+\delta_{25}+\delta_{26}+\delta_{27}+\delta_{28}=0,\\
-.00814\,\delta_1+\delta_{11}+\delta_{23}=0,\\
-.008092\,\delta_1+\delta_{12}=0,\\
-.005\,\delta_1+\delta_{24}=0,\\
-.00088\,\delta_1+\delta_{17}=0,\\
-.000909\,\delta_1+\delta_{29}+\delta_{30}=0,
\end{array}\right\}\qquad(7)$$

where $\delta_i \geqslant 0$, $i = 1, 2, \ldots, 31$.

Applying the cascading simplex method to the dual formulation we initially obtain the linear program

$$max\ \sum_{i=1}^{31}\delta_i$$

subject to equations (7).

https://doi.org/10.1017/S1446181100001814 Published online by Cambridge University Press

The program is consistent and phase I yields the following x and e vectors,

(1, .000423, .001271, .002485, 4.671, 0, 0, 0, 0, 0, .00814, .008092, 0, 0, 0,
.001, .000888, 0, 0, 0, 0, 0, 0, .005, 0, 0, 0, 0, 0, .000909, .00119)

and

(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
1, 1, 0, 0).

The new e vector gives rise to an associated linear program to start off the cascade in phase II. The sequence of x vectors for phase II continues with

(2, .000423, .002542, .00497, 9.342, .000423, 0, 0, 0, 0, .01628, .016184,
0, 0, 0, .002, .00176, 0, 0, 0, 0, 0, 0, .01, 0, 0, 0, 0, 0, .001818, 0.0238)

and e_6 is set to zero.

The cascade continues until finally the last component is made positive. The final x vector is

(20, .008523, .017794, .04473, 84.069, .000423, .001271, .002485, .002542,
.002, .15466, .16184, 4.571, .002, .001271, .017452, .0176, .00423,
.001271, .002485, .002542, .002, .00814, .1, 4.671, .002, .001271, .001,
.000909, .017271, .0238)

and e_{29} is set to zero. Thus the required feasible point with positive compoents is

(1, .00042615, .0008897, .0022365, 4.20345, .00002115, .00006355,
.00012425, .0001271, .0001, .007733, .008092, .23355, .0001, .00006355,
.0008771, .00088, .00002115, .00006355, .00012425, .0001271, .0001,
.000407, .005, .23355, .0001, .00006355, .00005, .00004545, .00086355,
.00006355).

5. Conclusions

Depending on the matrix (a_{ij}) certain δ_i can be constrained to zero by (1), (2) and (3); or possibly (1), (2) and (3) are inconsistent. In either case we have a degenerate geometric program. Degenerate programs are most easily dealt with by eliminating the degeneracy. When (1), (2) and (3) are inconsistent then the dual problem is totally degenerate which implies by Lemma 1, p. 168 of [2] that the primal problem has a limit point for which the objective function is zero. Since a posynomial function is by construction always non-negative we can therefore try to minimize some other aspect of the problem or add another constraint to the problem. In other words, we should reformulate the problem. Inconsistency of (1), (2) and (3) can be found by phase I of the cascading simplex method.

The other possibility is that some variables δ_i are restricted to be equal to zero by (1), (2) and (3). By applying the cascading simplex method we wish to find a point for which $x>0$ (δ being associated with x). The cascading simplex method will find an interior point if the program is canonical ($\delta>0$). If the problem is degenerate then the cascading simplex method will find which components of δ are zero. The point found will be an interior point to the reduced form of the dual problem as defined in [2].

To apply gradient or second-order methods to solve the problem we require an interior point since the gradient can be infinite at the boundary. The cascading simplex method finds such a point when checking for degeneracy.

The cascading simplex algorithm has been implemented as part of a geometric programming algorithm [4, 5].

References

[1] G. B. Dantzig, J. C. De Haven and C. P. Sams, "A mathematical model of the chemistry of the external respiratory system", in *Proc. 4th Berkeley Symp. on Math. Stat. and Probab.* (ed. J. Neyman) (Berkeley: University of California Press), 4 (1960), 181–196.

[2] R. J. Duffin, E. L. Peterson and C. Zener, *Geometric programming—theory and application* (Wiley, 1967).

[3] G. Hadley, *Linear programming* (Addison-Wesley, 1962).

[4] T. R. Jefferson, "A dual-primal algorithm for geometric programming" (to appear).

[5] T. R. Jefferson, *Geometric programming with an application to transportation planning* (Ph.D. Dissertation, Northwestern University, 1972).

[6] A. Orden, "On the solution of linear equation/equality systems", *Math. Programming* 1 (1971), 137–152.

School of Mechanical and Industrial Engineering
University of N.S.W.
P.O. Box 1, Kensington, N.S.W. 2033
Australia

https://doi.org/10.1017/S1446181100001814 Published online by Cambridge University Press