AFOSR-TR- 78-0681

AD A052370

AD No.
DDC FILE COPY

2
B.S.

SOUTHERN METHODIST UNIVERSITY

DEPARTMENT OF OPERATIONS RESEARCH AND ENGINEERING MANAGEMENT
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
DALLAS, TEXAS 75275

9

14

Technical Rep[redacted] OREM-78004

6

A SPECIAL QUADRATIC PROGRAM

10

R. Helgason,

J. Kennington

H. Lall

15 AFOSR-77-3151

16 2304

17 A6

Department of Operations Research
and
Engineering Management
Southern Methodist University
Dallas, TX 75275

18 AFOSR

11 Febr[redacted]978

12 14p.

19 78-0681

Comments and criticisms from interested readers are cordially invited.

410 597

ABSTRACT

This paper presents a characterization of the solutions of a special quadratic program. This characterization is then used in the development of an efficient algorithm for this class of problems.

ACKNOWLEDGEMENT

This research was supported in part by the Air Force Office of Scientific Research under Grant Number AFOSR 77-3151.

I. INTRODUCTION

Consider the following quadratic program,

$$\min \frac{1}{2} x'Dx - a'x \tag{1}$$

$$\text{s.t.} \quad \Sigma_j x_j = c \tag{2}$$

$$0 \leq x \leq b \tag{3}$$

where D is a positive diagonal matrix and b is a nonnegative vector. Quadratic programs of this type arise when one applies a resource-directive decomposition procedure using subgradient optimization to solve multicommodity network flow problems, (see [1,2,3]). The basic approach is to distribute the arc capacity among the commodities and then solve a set of single commodity problems. This solution is feasible for the multicommodity problem. If in addition this solution is within ε% (where the user selects ε) of a lower bound, then the procedure terminates. Otherwise, a subgradient is used to determine a new proposed allocation. Generally the proposed allocation is not feasible (i.e. exceeds the original arc capacity). When this occurs, one must project this proposed allocation back onto the set of feasible allocations. This projection involves solving a quadratic program of the form (1) - (3) for each arc whose proposed allocation exceeds the arc capacity. Since the subgradient procedure has been found to be at least twice as fast as competing algorithms for this class of problems (see [1]) there is great motivation for developing a fast algorithm for (1) - (3).

Held, Wolfe, and Crowder [2] present an algorithm for solving (1) - (3) when D is an identity matrix and (3) is replaced by $x \geq 0$. The objectives of this exposition are (i) to extend the theory of [2] to the more general program (1) - (3) and (ii) to present an efficient algorithm for (1) - (3) based on this theory.

II. CHARACTERIZATION OF SOLUTIONS

Let us rewrite (1) - (3) in a slightly different form as follows:

$$\min \frac{1}{2} x'Dx - a'x \tag{4}$$

$$\text{s.t.} \quad \Sigma_j x_j - c = 0 \qquad (\lambda) \tag{5}$$

$$x_j - b_j \leq 0 \qquad (u_j) \tag{6}$$

$$-x_j \leq 0 \qquad (v_j), \tag{7}$$

where λ, u_j, and v_j are the Kuhn-Tucker multipliers associated with the three types of constraints. The Kuhn-Tucker conditions for (4) - (7) may be stated as follows:

$$d_j x_j - a_j + u_j - v_j + \lambda = 0, \text{ (for all } j) \tag{8}$$

$$u_j(x_j - b_j) = 0, \text{ (for all } j) \tag{9}$$

$$v_j x_j = 0, \text{ (for all } j) \tag{10}$$

$$u_j, v_j \geq 0, \text{ (for all } j) \tag{11}$$

plus (5), (6), and (7),

where d_j is the j^{th} diagonal element of D. Consider the following solution as a function of λ.

$$\left.\begin{aligned} x_j(\lambda) &= \text{MAX}\left\{\text{MIN}\left(\frac{a_j - \lambda}{d_j}, b_j\right), 0\right\} \\ u_j(\lambda) &= \text{MAX}\,\{a_j - \lambda - d_j b_j, 0\} \\ v_j(\lambda) &= \text{MAX}\,\{\lambda - a_j, 0\}. \end{aligned}\right\} \tag{12}$$

For any selection of λ, the above solution clearly satisfies (6), (7), and (11). We now show that this solution will also satisfy (8), (9), and (10).

Proposition 1.

The solution given by (12) satisfies (9).

Proof. Case 1: $a_j - \lambda - d_j b_j < 0 \Rightarrow u_j = 0 \Rightarrow u_j(x_j - b_j) = 0$

Case 2: $a_j - \lambda - d_j b_j \geq 0 \Rightarrow \frac{a_j - \lambda}{d_j} \geq b_j \Rightarrow x_j = b_j \Rightarrow u_j(x_j - b_j) = 0.$

This completes the proof of proposition 1.

Proposition 2.

The solution given by (12) satisfies (10).

Proof. Case 1: $a_j - \lambda < 0 \Rightarrow \frac{a_j - \lambda}{d_j} < 0 \Rightarrow x_j = 0 \Rightarrow v_j x_j = 0.$

Case 2: $a_j - \lambda \geq 0 \Rightarrow v_j = 0 \Rightarrow v_j x_j = 0.$

This completes the proof of proposition 2.

Proposition 3.

The solution given by (12) satisfies (8).

Proof: Case 1: $\frac{a_j - \lambda}{d_j} \geq b_j \Rightarrow x_j = b_j$, $a_j - \lambda - d_j b_j \geq 0$, and $a_j - \lambda > 0$.

$a_j - \lambda - d_j b_j \geq 0 \Rightarrow u_j = a_j - \lambda - d_j b_j$.

$a_j - \lambda > 0 \Rightarrow v_j = 0$. Thus $d_j x_j - a_j + u_j - v_j + \lambda$

$= d_j b_j - a_j + a_j - \lambda - d_j b_j - 0 + \lambda = 0.$

<u>Case 2</u>: $0 < \frac{a_j - \lambda}{d_j} < b_j \Rightarrow x_j = \frac{a_j - \lambda}{d_j}$, $a_j - \lambda - d_j b_j < 0$, and $a_j - \lambda > 0$.

$a_j - \lambda - d_j b_j < 0 \Rightarrow u_j = 0$.

$a_j - \lambda > 0 \Rightarrow v_j = 0$. Thus $d_j x_j - a_j + u_j - v_j + \lambda$

$$= d_j\left(\frac{a_j - \lambda}{d_j}\right) - a_j + 0 - 0 + \lambda = 0.$$

<u>Case 3</u>: $\frac{a_j - \lambda}{d_j} \leq 0 \Rightarrow x_j = 0$ and $a_j - \lambda \leq 0$.

$a_j - \lambda \leq 0 \Rightarrow a_j - \lambda - d_j b_j \leq 0$ and $v_j = \lambda - a_j$.

$a_j - \lambda - d_j b_j \leq 0 \Rightarrow u_j = 0$. Thus $d_j x_j - a_j + u_j - v_j + \lambda$

$$= 0 - a_j + 0 - (\lambda - a_j) + \lambda = 0.$$

This completes the proof of proposition 3.

Hence to solve (1) - (3) one need only find the appropriate λ such that (5) is satisfied.

$$\text{Let } g(\lambda) = \Sigma_j x_j(\lambda) = \Sigma_j \text{ MAX}\left\{\text{MIN}\left(\frac{a_j - \lambda}{d_j}, b_j\right), 0\right\}.$$

Then we must find λ^* such that $g(\lambda^*) = c$. Note that since $d_j > 0$ and $b_j \geq 0$, $x_j(\lambda)$ may be expressed as follows:

$$x_j(\lambda) = \begin{cases} b_j & , \lambda \leq a_j - d_j b_j \\ \dfrac{a_j - \lambda}{d_j}, & a_j - d_j b_j < \lambda \leq a_j \\ 0 & , \lambda > a_j . \end{cases}$$

Clearly each $x_j(\lambda)$ is piece-wise linear and monotone nonincreasing. Since the sum of such functions preserves this property, $g(\lambda)$ is piece-wise linear and monotone nonincreasing. A typical g is illustrated in Figure 1.

Figure 1 About Here

III. ALGORITHM

Suppose there are n $x_j(\lambda)$, then the breakpoints for the piece-wise linear function $g(\lambda)$ occur at the 2n points $a_j - d_j b_j$ and a_j for $j = 1, \ldots, n$. Let $y_1, \ldots, y_{2n}$ denote these breakpoints where $y_1 \leq y_2 \leq \cdots \leq y_{2n}$. Then for $\lambda \leq y_1$, $g(\lambda) = \Sigma_j b_j$ and for $\lambda \geq y_{2n}$, $g(\lambda) = 0$. Hence (1) - (3) has a feasible solution if and only if $0 \leq c \leq \Sigma_j b_j$.

We now present an algorithm for obtaining the value λ^*, such that $g(\lambda^*) = c$. The procedure consists of a binary search to bracket λ^* between two breakpoints followed by a linear interpolation.

ALGORITHM FOR λ^*

0. *Initialization*

   If $c > \Sigma_j b_j$ or $c < 0$, terminate with no feasible solution; otherwise, set

   $\ell \leftarrow 1$, $r \leftarrow 2n$,

   $L \leftarrow \Sigma_j b_j$ and $R \leftarrow 0$.

1. *Test For Bracketing*

   If $r - \ell = 1$, go to 4; otherwise, set $m \leftarrow [(\ell + r)/2]_I$, where $[k]_I$ is the greatest integer $\leq k$.

2. *Compute New Value*

   Set $C \leftarrow \Sigma_j \text{ MAX} \left\{ \text{MIN} \left(\dfrac{a_j - y_m}{d_j}, b_j \right), 0 \right\}$.

3. *Update*

   If $C = c$, terminate with $\lambda^* \leftarrow y_m$.

   If $C > c$, set $\ell \leftarrow m$, $L \leftarrow C$, and go to 1.

   If $C < c$, set $r \leftarrow m$, $R \leftarrow C$, and go to 1.

4. *Interpolate*

Terminate with

$$\lambda^* \leftarrow y_\ell + \frac{(y_r - y_\ell)\cdot(c - L)}{(R - L)} .$$

REFERENCES

1. Ali, A., R. Helgason, J. Kennington, and H. Lall, "Solving Multicommodity Network Flow Problems," Technical Report IEOR 77015, Department of Operations Research, Southern Methodist University (1977).

2. Held, M., P. Wolfe, and H. Crowder, "Validation of Subgradient Optimization," Mathematical Programming, 6, 62-88, (1974).

3. Kennington, J., and M. Shalaby, "An Effective Subgradient Procedure for Minimal Cost Multicommodity Flow Problems," Management Science, 23, 9, 994-1004, (1977).

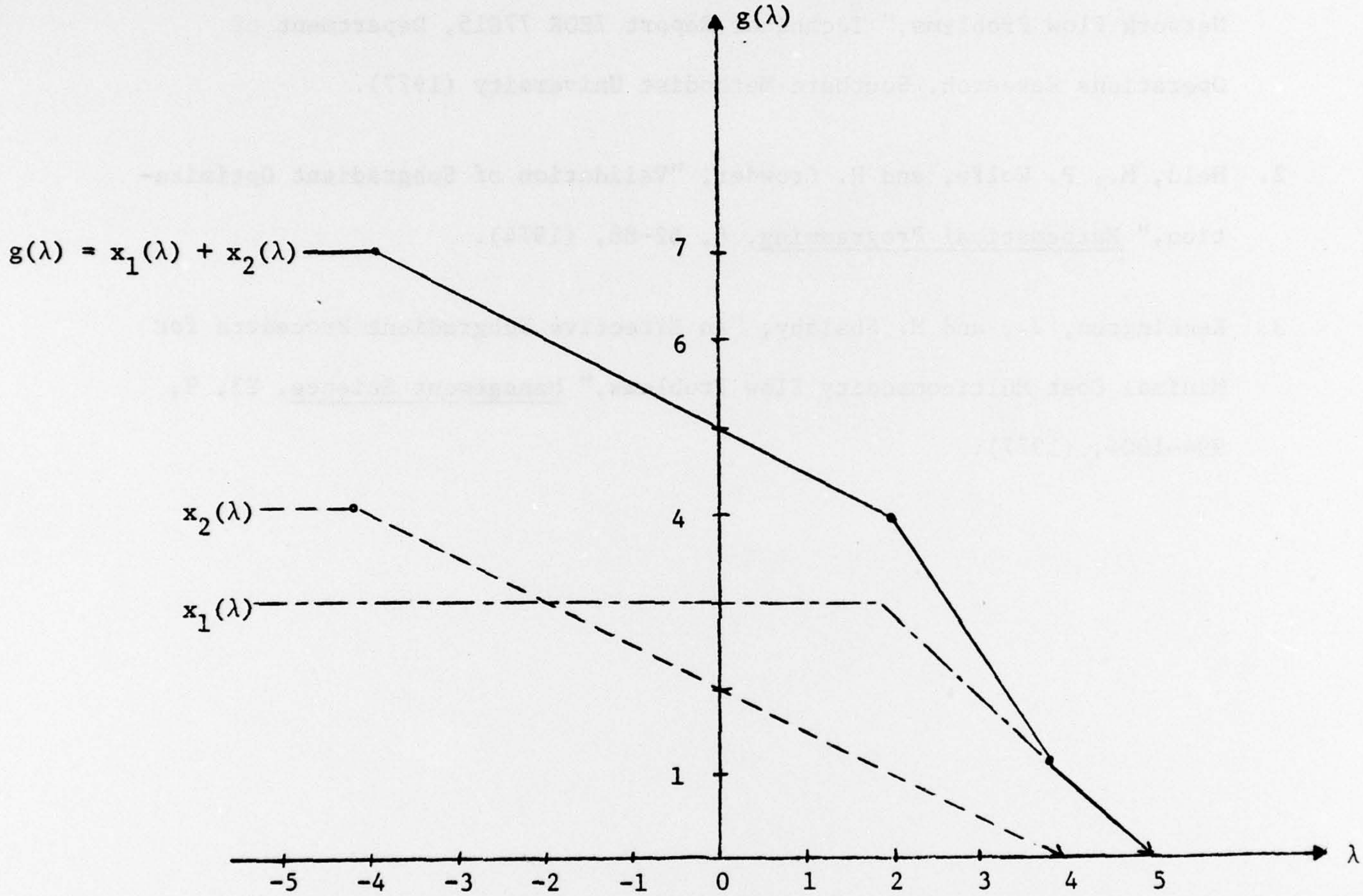

Figure 1. Illustration of $g(\lambda)$ ($a_1 = 5$, $d_1 = 1$, $b_1 = 3$, $a_2 = 4$, $d_2 = 2$, $b_2 = 4$)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 78-0681	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A SPECIAL QUADRATIC PROGRAM		5. TYPE OF REPORT & PERIOD COVERED Interim
		6. PERFORMING ORG. REPORT NUMBER Tech Rpt OREM 78004
7. AUTHOR(s) R. Helgason, J. Kennington, H. Lall		8. CONTRACT OR GRANT NUMBER(s) AFOSR 77-3151
9. PERFORMING ORGANIZATION NAME AND ADDRESS Southern Methodist University Department of Operations Research and Engineering Management Dallas, Texas 75275		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A6
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/NM Bolling AFB, DC 20332		12. REPORT DATE February 1978
		13. NUMBER OF PAGES 11
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)

Approved for public release; distribution 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)

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

This paper presents a characterization of the solutions of a special quadratic program. This characterization is then used in the development of an efficient algoritym for this class of problems.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)