Global Journal of Science Frontier Research: F
Mathematics and Decision Sciences
Volume 18 Issue 7 Version 1.0 Year 2018
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 2249-4626 & Print ISSN: 0975-5896

Results and Conclusion of an Algorithm for Solving Indefinite QR-Programming Problems

By Awatif M. A. El Siddieg
Prince Sattam Bin Abdul-Aziz University

Abstract- In this paper we have two sections. In section (1), we write a Matlab program and apply it to solve chosen problems in general QP –problems, we use sub programs[11]. Section (2) conclude our work reported in this paper gave no account to the *special* structures that the matrix of constraints A might have. The *work* is ideal when A is dense, that is, full of non-zero elements. [19].

GJSFR-F Classification: FOR Code: MSC 2010: 00A69

Strictly as per the compliance and regulations of:

Results and Conclusion of an Algorithm for Solving Indefinite QR-Programming Problems

Awatif M. A. El Siddieg

Abstract- In this paper we have two sections. In section (1), we write a Matlab program and apply it to solve chosen problems in general QP –problems, we use sub programs[11]. Section (2) conclude our work reported in this paper gave no account to the *special* structures that the matrix of constraints A might have. The *work* is ideal when A is dense, that is, full of non-zero elements. [19].

I. Introduction

We solve a general quadratic programming problems[[15],[4],[17]] , obtaining a local minimum of a quadratic function subject to inequality constraints. The method terminates at a KKT-point in finite steps[8]. No effort is needed when the function is non- convex[[10],[8] ,[19]]. We give the general description of the matrices that uses in the program and tested the program by a number of problems.

Section(1)

II. Results

In this section, we write a Matlab program and apply it to solve the chosen problems. The program uses *sub programs*:-

1. htu(G,A): to evaluate the inverse of the active Lagrangian matrix, using the QR-factorization of the matrix of constraints when the tableau is complementary).[[13],[14]]. (We know that H,U and T define the inverse of the upper left partition of the basis matrix).This calls for making them available at every complementary tableau[[2],[19]].

2. init(A,G) to obtain an initial feasible point to the main algorithm.
3. solver (A,b), is used to solve a subsystem in the main algorithm.
4. lufactors (A), is used by solver(the above program). [17]

The Program

The program is designed to start with the Hessian matrix G, which is an $n \times n$ symmetric matrix, and A is an $n \times m$ matrix of the constraints, g the gradient of *f,* and b, the vector of right-hand coefficients b_i.[[6],[7],[9]].

Chosen Problems

The above program has been tested by some problems and proved to work adequately.

* Minimize : $-8x_1 - 16x_2 + x_1^2 + 4x_2^2$

Subject to: $-x_1 - x_2 \geq -5$

Ref

19. http://www.iiste.org/journals/index.php/MTM.

Author: Prince Sattam Bin Abdul-Aziz University Faculty of Sciences and Humanilities Studies Math. Dep. Hotat Bani Tamim Kingdom of Saudi Arabia, P.O.Box 13. e-mails: wigdan@hotmail.com, a.elsiddieg@psau.edu.sa

$$-x_1 \geq -3$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

$$G = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & -1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, b = \begin{bmatrix} -5 \\ -3 \\ 0 \\ 0 \end{bmatrix}, \quad g = \begin{bmatrix} -8 \\ -16 \end{bmatrix}$$

x =

3

0

la =

2

-16

>> x =

x

3

2

>> la

la =

2

*. Minimize: $0.5x_1^2 + x_2^2 + 1.5x_3^2 + x_1x_2 + 25.5x_1 + 18x_2 + 29.875x_3$

Subject to : $-14x_1 - 2x_2 - 4x_3 \geq -12$

$$-x_1 - 2x_2 - x_3 \geq -4$$
$$-x_3 \geq -2$$
$$x_1 \geq 0$$
$$x_2 \geq 0, \; x_3 \geq 0$$

$$A = \begin{bmatrix} -14 & -2 & -4 \\ -1 & -2 & -1 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} -12 \\ -4 \\ -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Notes

Notes

$$G = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \qquad g = \begin{bmatrix} 25.5 \\ 18 \\ 29.875 \end{bmatrix}$$

x =
-0.0000
1.5000
-0.0000
la =
-14.7500
4.6563
6.3333
x =
-0.0000
1.5000
-0.0000
la =
-14.7500
4.6563
6.3333
x =
0.0000
0
1.0000
la =
-1.2500
5.5000
4.7500
x =
0.0000
0
1.0000
la =
-1.2500
5.5000
4.7500
x =
0
5
0
2

```
la =
      1.7500
     -8.2500
     16.0000
x =
      0
      0
      2
la =
      1.7500
     -8.2500
     16.0000
>> x
x =
              0
       1.5000
       2.0000
>> la
la =
       11.2188
              6
       20.5000
        0.6563
```

Section(2)

III. Conclusion

The work reported in this paper gave no account to the special structures that the matrix of constraints A might have. The work is ideal when A is dense, that is, full of non-zero elements. In many problems the unknown variables) $x_i (i = 1,\dots,n)$ are required to satisfy-bound restrictions, in which case we start the problem as follows:

$$\begin{aligned} &\min imize \;\; 0.5\, \underline{x}^T G \underline{x} + g^T \underline{x} \\ &subject\ to \\ &A^T \underline{x} \geq \underline{b} \\ &I_i \leq x_i \leq u_i \end{aligned} \tag{2.1}$$

Where I_i and u_i are respectively the lower and upper bounds for the variable x_i, A is $n \times m$ and assumed to be dense, $\underline{b}$ is an m vector, G is an $n \times n$ symmetric matrix and g is an n-vector. In (2.1) except in very *special* situations. A is dense since the bound constraints are separately considered. In this section we give our trial in treating, the case when $I_i = 0$ and u_i is infinite, that is when $x_i \geq 0 \quad \forall\, i$, we do not give general proofs here, nor do we present a compact description of an algorithm. Instead we will show the steps to be followed in a similar way similar to those given in our work reported in this paper [19].

The problem to be treated is

Ref

19. http://www.iiste.org/journals/index.php/MTM.

$$\begin{aligned} \min imize \;\; & 0.5\, \underline{x}^T G \underline{x} + g^T \underline{x} \\ subject\ to \;\; & A^T \underline{x} \geq \underline{b} \\ & \underline{x} \geq 0 \end{aligned} \tag{2.2}$$

let $\underline{\lambda}$ be the vector of multipliers corresponding to $A^T \underline{x} - \underline{b} \geq \underline{0}$ and $\underline{y}$ be the vector of multipliers[1], corresponding the bound constraints $\underline{x} \geq 0$.

The KKT – conditions to (2.2) we get

$$\begin{aligned} & G\underline{x} + \underline{g} - A\underline{\lambda} - \underline{y} = \underline{0} \\ & \underline{b} - A^T \underline{x} + \underline{v} = \underline{0} \\ & \underline{V}^T \underline{\lambda} = 0\,, \quad \underline{y}^T \underline{x} = 0 \\ & \underline{x}\,,\, \underline{y} \geq \underline{0},\, \underline{v},\, \lambda \geq \underline{0} \end{aligned} \tag{2.3}$$

where $\underline{v}$ is the vector of slack variables (2.3) could be put in the form: $M_1 \underline{W} + M_2 \underline{Z} = \underline{q}$

$$\underline{W} = \begin{bmatrix} \underline{Y} \\ \underline{V} \end{bmatrix} \;,\; \underline{q} = \begin{bmatrix} -\underline{g} \\ -\underline{b} \end{bmatrix} \;,\; M_1 = \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix}$$

$$M_2 = \begin{bmatrix} G & -A \\ -A^T & 0 \end{bmatrix} \;,\; \underline{Z} = \begin{bmatrix} \underline{x} \\ \underline{\lambda} \end{bmatrix} \tag{2.4}$$

$$\begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \underline{Y} \\ \underline{V} \end{bmatrix} + \begin{bmatrix} G & -A \\ -A^T & 0 \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{\lambda} \end{bmatrix} = \begin{bmatrix} -\underline{g} \\ -\underline{b} \end{bmatrix}$$

$$\underline{g} - G\underline{x} - A\underline{\lambda} - \underline{y} = \underline{0}$$

$$\underline{b} - A^T \underline{x} + \underline{v} = 0$$

$$\underline{v}^T \underline{\lambda} = 0$$

$$\underline{y}^T \underline{x} = 0$$

$$\underline{x}, \underline{y} \geq 0 \quad , \quad \underline{v}, \underline{\lambda} \geq 0$$

The general complementary tableau [2], will have the *form*:

$[M_B : M_N]$ with M_B having the *form*: $M_B = \begin{bmatrix} G_{12} & -A_{11} & -I & 0 \\ G_{22} & -A_{21} & 0 & 0 \\ -A_{21}^T & 0 & 0 & 0 \\ -A_{22}^T & 0 & 0 & I \end{bmatrix}$

and M_N having the *form*: $M_N = \begin{bmatrix} 0 & 0 & G_{11} & -A_{12} \\ -I & 0 & G_{22}^C & -A_{22} \\ 0 & I & -A_{11}^T & 0 \\ 0 & 0 & -A_{12}^T & 0 \end{bmatrix}$

Ref

1. AMO (2015). Advanced Modeling and Optimization, Volume 17, Number 2.

Here G_{11}, G_{12} and G_{22} define the following partition of G.

$$G = \begin{matrix} n_1 \\ n_2 \end{matrix}\begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix}, \quad n_1 + n_2 = n$$

and A_{11}, A_{12}, A_{21}, A_{22} define the following partition of A.

$$A = \begin{matrix} m_1 \\ m_2 \end{matrix}\begin{bmatrix} A_{11} & A_{12} \\ A_{12} & A_{22} \end{bmatrix}, \quad m_1 + m_2 = m$$

corresponding : $\underline{x}, \underline{\lambda}, \underline{y}, \underline{v}$, $\underline{g}$ and $\underline{b}$ were partitioned to

$$\underline{x} = \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix}\begin{matrix} n_1 \\ n_2 \end{matrix}, \ \underline{\lambda} = \begin{matrix} n_1 \\ n_2 \end{matrix}\begin{bmatrix} \underline{\lambda}_1 \\ \underline{\lambda}_2 \end{bmatrix}, \quad \underline{v} = \begin{bmatrix} \underline{v}_1 \\ \underline{v}_2 \end{bmatrix}\begin{matrix} m_1 \\ m_2 \end{matrix} \ \underline{g} = \begin{bmatrix} \underline{g}_1 \\ \underline{g}_2 \end{bmatrix}\begin{matrix} n_1 \\ n_2 \end{matrix}$$

$$\underline{b} = \begin{bmatrix} \underline{b}_1 \\ \underline{b}_2 \end{bmatrix}\begin{matrix} m_1 \\ m_2 \end{matrix}, \quad \underline{y} = \begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \end{bmatrix}\begin{matrix} n_1 \\ n_2 \end{matrix}$$

Accordingly the *basic* variables are $\underline{x}_2$, $\underline{\lambda}_1, \underline{y}_1$ and $\underline{v}_2$. Their respective non-basic complements are $\underline{y}_2, \underline{v}_1, \underline{x}_1$ and $\underline{\lambda}_2$.

Omitting the superscripts, let q solve

$$\min \ \{y_{q1}, \lambda_{q2}\} \tag{2.5}$$

$$q \in \{q_1, q_2\}$$

Where q_1 and q_2 satisfy :

$$y_{q1} = \min_{1 \le i \le n_1} \ y_i \tag{2.6}$$

$$\lambda_{q2} = \min \ \lambda_i \tag{2.7}$$

$$1 \le i \le m_1$$

To carry on the description let $q = q_2$. If $\lambda_{q2} \ge 0$, then we are at a KKT- point. Otherwise the complement v_{q2} is chosen to be increased.

Accordingly the *basic* variables change by:

$$\underline{x}_2 = \underline{x}_2^{\backslash} - \underline{d}x v_{q2} \tag{2.8}$$

$$\underline{\lambda}_1 = \underline{\lambda}_1^{\backslash} - \underline{d}_x \lambda v_{q2} \tag{2.9}$$

$$y_1 = y_1^{\backslash} - \underline{d}_y v_{q2} \tag{2.10}$$

$$\underline{v}_1 = \underline{v}_1^{\backslash} - \underline{d}_x v_{q2} \tag{2.11}$$

Notes

where the dashes indicate the current values $\underline{d}_x, \underline{d}_\lambda, \underline{d}_y$ and $\underline{d}_v$ are the

$$\text{solution of:} \begin{bmatrix} G_{12} & -A_{11} & -I & 0 \\ G_{22} & -A_{11} & 0 & 0 \\ -A_{21} & -A_{21} & 0 & 0 \\ -A_{22} & 0 & 0 & I \end{bmatrix} \begin{bmatrix} \underline{d}_x \\ \underline{d}_\lambda \\ \underline{d}_y \\ \underline{d}_v \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{0} \\ \underline{e}_{q2} \\ 0 \end{bmatrix}$$

$$\text{is solved in two steps:} \begin{bmatrix} G_{22} & -A_{12} \\ -A_{22}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{dx} \\ \underline{d\lambda} \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{e}_{q2} \end{bmatrix}$$

$$\begin{aligned} G_{22}\underline{dx} - A_{11}\underline{d}_\lambda &= \underline{0} \\ -A_{22}^T \underline{dx} &= \underline{e}_{q2} \end{aligned} \qquad (2.1\ 2)$$

$$\underline{dy} = G_{22}\underline{d}_x - A_{11}\underline{d}_\lambda \qquad (2.1\ 3)$$

$$\underline{dv} = A_{22}^T \underline{d}_x \qquad (2.15)$$

The increase of v_{q2} is continued until either λ_{q2} *increase* to zero or v_{q2} is blocked by either a basic x_{P1} decreasing to zero. The next step is to restart again if λ_{q2} decreases to zero first, in which case we are at another [19] complementary tableau[2]. Or one of the complements y_{P1} of x_{P1} or λ_{P2} of v_{P2} is to be changed in a similar way to that described in the main work of the paper. The process will keep on going until the solution is located. Also we point out another two incomplete features of our algorithm. They are:

1) It did not give any account to degeneracy.

2) Updating the factors of $G_A^{(K)}$ is not carried in all cases.

So, according to [[14],[15],[16]], is equivalent to active set methods in convex problems. When solving non-convex problems the method is more systematic than the variants of the active set methods[[8],[10]].

The latter methods need to change the strategy of choosing the direction of search from time to time, and some of them have no clue of what to do in the negative definite case[11]. In our work no change in the strategy is needed. In fact no check of indefiniteness of the reduced (generalized) Hessian is required.

Still we believe that our work should be tested in all aspects against the (modified) active set methods to reflect the major advantages and disadvantages of our work (i.e the active set methods) which dominated the scene for the last twenty years (of course to our knowledge). Also our work need to be compared with Beal's method [11] , [14]], since they are both constrained as simplex-like methods, although we feel that the general behavior of our work looks different. However, [14] referenced to the equivalence between the active set methods and Beale's method in convex problems. Orthoganalization methods are well known in the numerical analysis community for their numerical stability. Conversely, normal equation methods are known for their lack of numerical stability. QR- factorizations[[12], [17] ,[18]],can make very good use of sparsity of the problem.

Ref

8. Gould N. I. M. and Tiont, P. L.(2002). An iterative working set method for large scale nonconverx quadratic programming Applied Numerical Mathematics, 43, pp. 109 – 128. 13

Acknowledgements

I gratefully acknowledge the head of Mathematics Department, College of Mathematical Sciences, Khartoum University Sudan. Pof. Mohsen Hashim for his support to do this work.

References Références Referencias

1. AMO (2015). Advanced Modeling and Optimization, Volume 17, Number 2.
2. ANITESCU, M. (2005). On Solving Mathematical Programs with Complementarity Constraints as Nonlinear Programs. SIAM Journal on optimization, 15, k pp.1203-1236.
3. Bertsekas D . P. (1991). Linear Network Optimization: Algorithms and codes MIT Press Cambridge, M.A.
4. David G. Luenberger (2003). Linear and nonlinear programming 2nd Edition. Pearson Education, Inc. Publishing as Addison-Wesley.
5. Dennis and Schnabel (1996). Numerical Methods for unconstrained Optimization and nonlinear equation classics in applied Mathematics. SIAM.
6. Fletcher R. (1987). Practical Methods of Optimization, 2nd Ed. John Wiley and Sons.
7. Fletcher R. and Leyffer, S.(2002). Nonlinear programming without a penalty function, Mathematical programming series A, 91, pp. 239-269.
8. Gould N. I. M. and Tiont, P. L.(2002). An iterative working set method for large scale nonconverx quadratic programming Applied Numerical Mathematics, 43, pp. 109 – 128. 13
9. Horst, Pardalos, and Thoai.(1995). Introduction to global optimization.
10. Kocava. M and Stingl. M., Pennon.(2003). A code for non-covex non-linear and semi definite programming optimization methods nonlinear and semi-definite programming optimization methods and software 18, pp. 317-333.
11. Lasunon, P. And Remsungnen, T.(2011). “A new method for solving Tri-level Linear Programming Problems”, International Journal of Pure and Applied Sciences and Technology, 7(2), 149-157.
12. Liu, Baoding and Yao, Kai (2014). “Uncertain multilevel programming: Algorithm and Applications”, Computers and Industrial Engineering, online.
13. Michael J.Best.(2017). Quadratic programming with computer programs .CRC Taylor and Francis Group. LLC Achampan and Hall Books.
14. qp OASES Joachim Ferreau· Christian Kirches. Andreas Potschka.(2014). A parametric active – set algorithm for quadratic programming. Hans George Book. Moritz Diehl.
15. QPSchur. (2005). A dual, active – set, Schur- complement method for large – scale and structured convex quadratic programming.
16. Reha Tutuncu. (2006). Optimization Methods in Finance, General Comuejols-Cambridge University Press.
17. Stephen G. Nash and Ariela Sofer. (1996). Linear and non –liner programming, McGraw Hill, New York.
18. ZDENEK DOSTAL. (2009). Optimal Quadratic programming Algorithms. Department of Applied. Mathematics. VSB-Technical University of Ostrava.
19. http://www.iiste.org/journals/index.php/MTM.

Notes