Mathematical Theory and Modeling
ISSN 2224-5804 (Paper) ISSN 2225-0522 (Online)
Vol.3, No.14, 2013

www.iiste.org

Results of the Implementation of the Extended Dantzig-Wolfe Method

Awatif M. A .Elsiddieg

Faculty of Mathematical Sciences Elneilain University Math . Department Khartoum Sudan
Present address : Selman Bin Abdul-Aziz University Faculty of Sciences and Humanilities Studies Math .Dep. .Hotat Bani Tamim Kingdom of Saudi Arabia, P.O.Box 13
* *E-mail address :* wigdan1965@hotmail.com

Abstract:

In this work we write a Matlab program and apply it to solve chosen problems. The program uses sub programs such as : 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. (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. Also, it uses: **init(A,G)** to obtain an initial feasible point to the main algorithm. **solver (A,b),** is used to solve a subsystem in the main algorithm. **lufactors (A)**, is used by solver.

1.Introduction:-

In this work we solve a general quadratic programming problems , obtaining a local minimum of a quadratic function subject to inequality constraints .The method terminates at a **KT** point in a finite number of steps. No effort is needed when the function is non- convex. In section (1) we give we give a general description of the matrices that uses in the program and tested the program by a number of problems. In section (2) we give a compact description of the algorithm , and that if the matrix of the constraints A is dense , so the work is ideal. In section (3) , we give a compact description to the algorithm and we show that 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.

2. The Program and Chosen Problems:-

The program is designed to start with the Hessian matrix G, which is an nxn symmetric matrix , and A is an nxm matrix of the constraints , g is the gradient of f , and b ,is the vector of right hand coefficient b_i. The program was tested by a number

of problems and proved to work adequately.

Step 1.

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

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

$$-x_1 \geq -3$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$G = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & -1 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} -5 \\ -3 \\ 0 \\ 0 \end{bmatrix}, \quad g = \begin{bmatrix} -8 \\ -16 \end{bmatrix}$$

Step 2.

Minimize

$$0.5x_1^2 + x_2^2 - 0.5x_3^2 + 5x_1x_2 - 3x_2x_3 + 1.75x_1 - 2.25x_2 - 14x_3$$

Subject to $-3x_1 - 8x_2 \geq -12$

$$3x_1 + 8x_2 + 12x_3 \geq 12$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$-x_3 \geq -2$$

$$A = \begin{bmatrix} -3 & -8 & 0 \\ 3 & 8 & 12 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} -12 \\ 12 \\ 0 \\ 0 \\ -2 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 5 & 0 \\ 5 & 2 & -3 \\ 0 & -3 & -1 \end{bmatrix}, \quad g = \begin{bmatrix} 1.75 \\ -2.25 \\ -14 \end{bmatrix}$$

Step 3.

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 $-14x_1 - 2x_2 - 4x_3 \geq -12$

$$-x_1 - 2x_2 - x_3 \geq -4$$

$$-x_3 \geq -2$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

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

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

3. Compact Description Of The Algorithm:-

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 $x_i (i = 1,...,n)$ are required to satisfy bound restrictions, in which case the problem is started as follows:

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

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, b is an m vector, G is an $n \times n$ symmetric matrix and g is an n-vector. In section 2 except in very special situations. A is dense since the bound constraints are considered separately.

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 away similar to those given in the main work reported in this thesis.

The problem to be treated is

$$\begin{aligned} &\min imize \;\; 0.5\,\underline{x}^T G\underline{x} + g^T\underline{x} \\ &subject\ to \quad A^T\underline{x} \geq \underline{b} \qquad\qquad (1) \\ &\qquad\qquad\quad \underline{x} \geq 0 \end{aligned}$$

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 corresponding the bound constraints $\underline{x} \geq 0$.

The **KT** – conditions to (1) we get

$$\begin{aligned} &G\underline{x} + \underline{g} - A\underline{\lambda} - \underline{y} = \underline{0} \\ &\underline{b} - A^T\underline{x} + \underline{v} = \underline{0} \\ &\underline{V}^T\underline{\lambda} = 0\,, \quad \underline{y}^T\underline{x} = 0 \qquad\qquad (2) \\ &\underline{x}\,,\,\underline{y} \geq \underline{0},\,\underline{v},\,\lambda \geq \underline{0} \end{aligned}$$

where $\underline{v}$ is the vector of slack variables (2) could be put in form $M_1\underline{W} + M_2\underline{Z} = \underline{q}$

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

$$M_2 = \begin{bmatrix} G & -A \\ -A^T & 0 \end{bmatrix}, \quad \underline{Z} = \begin{bmatrix} \underline{x} \\ \underline{\lambda} \end{bmatrix} \qquad \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \underline{Y} \\ \underline{V} \end{bmatrix} + \begin{bmatrix} G & -A \\ -A^T & 0 \end{bmatrix}\begin{bmatrix} \underline{x} \\ \underline{\lambda} \end{bmatrix} = \begin{bmatrix} -\underline{g} \\ -\underline{b} \end{bmatrix}$$

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

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

$$\begin{aligned} &\underline{v}^T\underline{\lambda} = 0 \\ &\underline{y}^T\underline{x} = 0 \\ &\underline{x}, \underline{y} \geq 0 \quad , \quad \underline{v}, \underline{\lambda} \geq 0 \end{aligned}$$

The general complementary tableau 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

$$M_N = \begin{bmatrix} 0 & 0 & G_{11} & -A_{12} \\ -I & 0 & G_{22}^y & -A_{22} \\ 0 & I & -A_{11}^T & 0 \\ 0 & 0 & -A_{12}^T & 0 \end{bmatrix}$$

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

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

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

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

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

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

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

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

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

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

where q_1 and q_2 satisfy $y_{q1} = \min\limits_{1 \le i \le n_1} y_i$

$$\lambda_{q2} = \min \lambda_i$$
$$1 \le i \le m_1$$

To carry on the description let $q = q_2$. If $\lambda_{q2} \ge 0$, then we are at a **KT** point. Otherwise the complement v_{q2} is chosen to be increased. Accordingly the basic variables change by

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

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

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

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

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

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

is solved in two steps

$$\begin{bmatrix} G_{22} & -A_{12} \\ -A_{22}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{d}x \\ \underline{d}\lambda \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{e}_{q2} \end{bmatrix}$$

$$G_{22}\underline{d}x - A_{11}\underline{d}_\lambda = \underline{0}$$
$$-A_{22}^T\underline{d}x = \underline{e}_{q2}$$

$$\underline{d}y = G_{22}\underline{d}_x - A_{11}\underline{d}_\lambda$$

$$\underline{d}v = A_{22}^T\underline{d}_x$$

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 complementary tableau. Or one of the complements y_{P1} of x_{P1} or λ_{P2} of v_{P2} is to be changed in away similar to that described in the main work of the thesis. 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.

As we mentioned in **[1]** that the method is exactly the Dantzig-Wolfe method when solving convex QP problems. So, according to Fletcher in **[4]** 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. 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. In our method no change in the strategy is needed. In fact no check of indefiniteness of the reduced (generalized) Hessian is required. **[5].**

Still we believe that the method should be tested in all aspects against the (modified) active set methods to reflect the major advantages and disadvantages of the method (i.e the active set methods) which dominated the scene for the last twenty years (of course to our knowledge).

Also the method need to be compared with Beal's method

in **[2]** since they are both constrained as simplex-like methods, although we feel that the general behavior of our method looks different. However, Fletcher in **[4]** 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 can make very good use of sparsity of the problem**.[3].**

4.C onclusion:

In this paper , the Matlab problem is applied to solve some chosen problems . It used sub programs such as, **htu(G,A)** to obtain the initial feasible point to the main algorithm , **solver(A,b)** , it is used to solve a subsystem in the main algorism and **lufactors(A)** is used by solver.

References

[1] Awatif M. .A. Elsiddieg Implementation to the Extended Dantzig –Wolfe Method .Mathematical Theory and Modeling.Vol.3 No .2 pp 12-29 (2013).

[2] Beale , E .M .L (1967),. Numerical Methods," in nonlinear programming (J . Abadie, ed).North – Holland ,
Amsterdam.

[3] David G.Luenbberger Linear and non linear programming 2^{nd}Edition . Pearson Education, Inc. Publishing as Addison –Wesley,(2003).

[4] Fletcher R. (1987) Practical Methods of Optimization, 2^{nd} Ed. John Wiley and Sons.

[5] Mohsin .H. A.Hashim An Extension to the Dantzig –Wolfe Method for general quadratic programming Ph.D.thesis, University of Khartoum..(1996).