A method for minimization of piecewise quadratic functions with lower and upper bounds

Janusz S. Sosnowski

CP-90-003
July 1990

IIASA

International Institute for Applied Systems Analysis □ A-2361 Laxenburg □ Austria

Telephone: (0 22 36) 715 21*0 □ Telex: 079 137 iiasa a □ Telefax: (0 22 36) 71313

Foreword

One of the main problems in development the decision support software is the availability of efficient optimization algorithms. These algorithms, when applied in decision support systems should possess several features - like robustness, efficiency and high speed. All these facts motivated the System and Decision Sciences Program to investigate all these topics.

The paper presents the result of a collaborative research made at the Systems Research Institute of the Polish Academy of Sciences. This research is being performed upon a contracted study agreement between the IIASA and the Polish Academy of Sciences. The presented algorithm will be included as an option in the HYBRID package implemented on IIASA computers: running UNIX (on Sun Sparc and on VAX 6210) and on PC IBM AT compatible.

Alexander B. Kurzhanski
Chairman
System and Decision Sciences Program.

Contents

A method for minimization of piecewise quadratic functions with lower and upper bounds

*Janusz S. Sosnowski**

1 Introduction

The paper describes a numerically stable method of minimization of piecewise quadratic convex functions subject to lower and upper bounds. The presented approach may be used for solving linear and quadratic programming problems while the multiplier, proximal multiplier or penalty methods are applied to original problem. An active set algorithm which takes into account the form of the objective function and bounds is developed. For solving a sequence of quadratic subproblems generated by the active set algorithm a numerically stable method of updating R- factors in QR factorization is adopted.

In the paper we will deal with the following problem:

$$\min f(x) \tag{1}$$

$$f(x) = cx + (1/2)\|A_1x - b_1\|^2 + (1/2)\|(A_2x - b_2)_+\|^2 \tag{2}$$

$$l \leq x \leq u \tag{3}$$

where: $A_1 \in R^{r\times n}, A_2 \in R^{s\times n}, b_1 \in R^k, b_2 \in R^s, c \in R^n$, and $l, u \in R^n$ are given lower and upper bounds $x \in R^n$

The following notation will be used:

$a_i^{(k)}$ – denotes the i–th rows of matrix A_k

$b_i^{(k)}$ – denotes the i–th components of the vector b_k

x_j – denotes j–th component of x

$\|x\|$ – denotes L_2–norm of x

$(u)_+$ – denotes the vector composed of components $\max(0, u_i)$

A_k^T – denotes the transposition of matrix A_k.

The above formulation generalizes the problems of minimization faced in ordinary multiplier method for linear programming problems. It also covers subproblems in the regularized or proximal multiplier method (Rockafellar, 1976). Note that if we introduce the following notation

$$A_1 = \begin{pmatrix} A_{11} \\ \sqrt{\gamma}I \end{pmatrix} \quad b_1 = \begin{pmatrix} b_{11} \\ \sqrt{\gamma}r \end{pmatrix}$$

then the minimized function will take the form

$$f(x) = cx + (\gamma/2)\|x - r\|^2 + (1/2)\|A_1x - b_1\|^2 + (1/2)\|(A_2x - b_2)_+\|^2$$

*Systems Research Institute, Polish Academy of Sciences, Newelska 6, 01–447 Warsaw, Poland.

The minimized function (2) is convex, piecewise quadratic, and twice differentiable beyond the set:

$$\{x \in R^n : a_i^{(2)}x - b_i^{(2)} = 0, \quad i = 1, \cdots, s\}$$

In the active set algorithm applied to the problem (1) – (3), we solve a sequence of minimization quadratic functions without constraints. After a finite number of steps, a set of indices of constraints which are active at the solution is found. That is based on the observation that at the optimal solution a certain subset of constraints is satisfied with equality. Let

$$J^* = \{j : x_j^* = l_j \quad \text{or} \quad x_j^* = u_j\}$$

where x^* – denotes solution of the problem (1) – (3).

Additionally we introduce the following set of indices

$$I^* = \{i : a_i^{(2)}x^* - b_i^{(2)} > 0\}$$

If the sets J^* and I^* were known, then the variables $x_j, j \in J^*$ could be eliminated and the problem reduced to the following unconstrained optimization problem

$$\min f_{I^*}(x) \tag{4}$$

$$f_{I^*}(x) = cx + (1/2)\|A_1x - b_1\|^2 + (1/2)\sum_{i \in I^*}(a_i^{(2)}x - b_i^{(2)})^2 \tag{5}$$

If we solve the above problem with respect to the free variables $x_j, j \in \{1, 2, \ldots, n\} \setminus J^*$, then we will reach a solution of the problem (1)-(3). The solution meets the Kuhn–Tucker conditions, which can be formulated in the following way.

Let A_2^* be a matrix composed of rows with indices from the set I^*, and b_2^* be the corresponding vector.

The gradient of f in point x^* is defined by

$$\frac{\partial f^*}{\partial x} = c + A_1^T(A_1x^* - b_1) + (A_2^*)^T(A_2^*x^* - b_2^*)$$

From the Kuhn-Tucker optimality condition, the following relations hold for the minimum point x^*

$$\frac{\partial f^*}{\partial x_j} \geq 0 \quad \text{if} \quad x_j^* = l_j, \qquad \frac{\partial f^*}{\partial x_j} \leq 0 \quad \text{if} \quad x_j^* = u_j,$$

and

$$\frac{\partial f^*}{\partial x_j} = 0 \quad \text{if} \quad l_j < x_j^* < u_j$$

In the active set algorithm a sequence of unconstrained quadratic problems are solved to predict the correct sets J^* and I^*.

2 An active set algorithm

Our algorithm differs from the active set algorithm described in (Fletcher,1981) and (Gill, Murray and Wright, 1981), because beside upper and lower bounds, we also take into account the piecewise quadratic form of the minimized function.

We define two types of working sets. At the k-th iteration of the active set algorithm, I_k will be a working set of the function f. That set defines a quadratic function as follows:

$$f_{I_k}(x) = cx + (1/2)\|A_1 x - b_1\|^2 + (1/2) \sum_{i \in I_k} (a_i^{(2)} x - b_i^{(2)})^2 \tag{4}$$

The second working set defines those variables which are fixed at bounds.

$$J_k = \{j : x_j = l_j \text{ or } x_j = u_j\} \tag{5}$$

For given point x, it is also useful to define the following set of indices

$$J(x) = \{\, j : x_j = l_j \text{ and } \partial f(x)/\partial x_j \geq 0 \,\} \cup \{\, j : x_j = u_j \text{ and } \partial f(x)/\partial x_j \leq 0 \,\} \tag{6}$$

and

$$I(x) = \{i : a_i^{(2)} x - b_i^{(2)} > 0\} \tag{7}$$

For the last set the following relation holds

$$f(x) = f_{I(x)}(x) = cx + (1/2)\|A_1 x - b_1\|^2 + (1/2) \sum_{i \in I(x)} (a_i^{(2)} x - b_i^{(2)})^2$$

Additionally, the complements of the working sets will be defined as follows

$$\bar{I}_k = \{1, 2, \ldots, s\} \setminus I_k. \tag{8}$$

$$\bar{J}_k = \{1, 2, \ldots, n\} \setminus J_k. \tag{9}$$

Using the notation defined above, for given working sets I_k and J_k, the following minimization subproblem can be formulated

$$\min f_{I_k}(x) \tag{10}$$

$$x_j = \bar{x}_j \quad j \in J_k \tag{11}$$

where

$$\bar{x}_j = \begin{cases} l_j & \text{if fixed lower bound} \\ u_j & \text{if fixed upper bound} \end{cases} \tag{12}$$

The active set algorithm, in the form described below, solves a sequence of the subproblems. For given working sets I_k and J_k, we minimize the quadratic function f_{I_k} in respect to variables x_j which indices j belong to the set $\bar{J}_k$. This variable will be free. The variables which indices belong to the set J_k are fixed on their bounds. This is an unconstrained quadratic subproblem. Its solution defines a search direction. The step length is determined to provide feasibility. The piecewise quadratic form of the function f is also taken into account while the step length is computed.

2.1 Algorithm

0. (Initialization) For the given initial feasible point x^0 determine I_0, J_0 as follows

$$I_0 = I(x^0), \quad J_0 = J(x^0).$$

Set $k := 0$.

1. (Subproblem optimality test). If

$$\partial f_{I_k}(x^k)/\partial x_j = 0, \quad i \in \bar{I}_k$$

then minimum of the subproblen is found, go to step 2.

Otherwise do to step 4.

2. (Optimality test). If

$$I_k = I(x^k) \quad i \quad J_k = J(x^k)$$

then assume x^k as an optimal solution and stop. Otherwise, continue.

3. (Working sets reduction). From the sets of indices which are defined as the workings sets delete an index for which holds

$$\max\{\max_{j \in J^d} |\partial f_{I_k}(x^k)/\partial x_j|, \max_{i \in J^d} |a_i^{(2)}x - b_i^{(2)}|\}$$

$$J^d = J_k \setminus J(x^k) \quad I^d = I_k \setminus I(x^k)$$

4. (Search direction computing). Solve the unconstrained optimization subproblem (10)–(12) and let $\bar{x}^k$ be a minimizer. Set

$$p^k = \bar{x}^k - x^k$$

as the search direction.

5. (Step length computation). Find $\bar{\alpha}$ - an upper bound for the step length

$$\bar{\alpha} = \min\{\bar{\alpha}_1, \bar{\alpha}_2\}$$

a. Where $\bar{\alpha}_1$ is chosen in such way that $x^k + \bar{\alpha}_1 p^k$ remains feasible:

$$\alpha_1 = \min_{j \in K} (l_j - x_j^k)/p_j^k, \qquad K = \{\, j : j \in \bar{J}, p_j^k < 0\,\}$$

$$\alpha_2 = \min_{j \in K} (u_j - x_j^k)/p_j^k, \qquad K = \{\, j : j \in \bar{J}, p_j^k > 0\,\}$$

$$\bar{\alpha}_1 = \min\{\alpha_1, \alpha_2\}$$

b. Where $\bar{\alpha}_2$ is maximal value which provides

$$I(x^k + \bar{\alpha}_2 p^k) = I(x^k)$$

Thus

$$\bar{\alpha}_2 = \min_{i \in I_k}\{(b_i^{(2)} - a_i^{(2)}x^k)/a_i^{(2)}p^k\}$$

For found $\bar{\alpha}$ compute:

$$\alpha^k = \operatorname*{argmin}_{\alpha} f(x^k + \alpha p^k) \quad \alpha \leq \bar{\alpha}$$

$$x^{k+1} = x^k + \alpha^k p^k$$

6. (Test for working set augmentation). If $\alpha^k < \bar{\alpha}$ then:

$$I_{k+1} = I_k \quad J_{k+1} = J_k$$

Set $k := k + 1$ and go to step 1.

7. (Working sets augmentation).

   a. If $\alpha^k = \bar{\alpha}_1$ and l is the index of the variables which bounded step length, then:

   $$J_{k+1} = J_k \cup \{l\}, \quad I_{k+1} = I_k$$

   b. If $\alpha^k = \bar{\alpha}_2$ and r is the index of the rows in the matrix A_2 which bounded step length, then:

   $$I_{k+1} = I_k \cup \{r\}, \quad J_{k+1} = J_k$$

Set $k := k + 1$ and go to step 1.

Remark: We note that working sets reduction in step 3 based on the observation that the Lagrange multiplier λ_j for the constraints:

$$l_i \leq x_j \leq u_j$$

are

$$\lambda_j = \begin{cases} \partial f_{I_k}(x^k)/\partial x_j & \text{if } x_j = l_j \\ -\partial f_{I_k}(x^k)/\partial x_j & \text{if } x_j = u_j \end{cases}$$

If optimality conditions are fulfilled then we have found the optimal point. If not, the objective function can be decreased by deleting corresponding bound or row of the matrix which defined function (4).

For the sake of simplicity we will drop the index k in the description of the working sets. Let A_2^I and b_2^I be a submatrix and a subvector composed of rows and coordinates corresponded to indices $i \in I$.

$$A^I = \begin{pmatrix} A_1 \\ A_2^I \end{pmatrix} \quad b_1 = \begin{pmatrix} b_1 \\ b_2^I \end{pmatrix}$$

Using the above notation, the problem (10)–(12) can be rewritten as follows

$$\min f_I(x) \tag{13}$$

$$f_I(x) = cx + (1/2)\|A^I x - b^I\|^2 \tag{14}$$

$$x_j = \bar{x}_j \quad j \in J \tag{15}$$

We divide the vector x into two vectors corresponding to the working set J and its complement:

x_J – vector free variables

x_J – vector fixed variables

We have

$$x = (x_{\bar{J}} \quad x_J)$$

Then we divide the matrix A^I into two submatrices which rows correspond to the fix and free variables respectively.

$$A^I = (A^I_{\bar{J}} \quad A^I_J)$$

So we have:

$$f_I(x) = c^{\bar{J}} x^{\bar{J}} + c^J x^J + (1/2)\|A^I_{\bar{J}} x^{\bar{J}} + A^I_J x^J - b^I\|^2 \tag{16}$$

Let us consider the problem of finding free variables x^J as a result of minimization (16) without constraints. We assume that the matrix A^I_J has full column rank. In this case the problem of minimizing function (16) has an unique solution. Such a situation takes place when the considered subproblem is defined for the proximal or for the regularized multiplier method. The minimum of the function (16) can be obtained by solving the following system of equations:

$$(A^I_J)^T A^I_J x^J = (A^I_J)^T (b^I - A^I_{\bar{J}} x^{\bar{J}}) - c^J \tag{17}$$

The classical approach to solving this problem is via the system of normal equations

$$\bar{B} x^J = \bar{b} \tag{18}$$

where $\bar{B}$ is the symmetric positive definite matrix in the form:

$$\bar{B} = (A^I_J)^T A^I_J \tag{19}$$

and

$$\bar{b} = (A^I_J)^T (b^I - A^I_{\bar{J}} x^{\bar{J}}) - c^J \tag{20}$$

In a discussion of methods which can be useful for solving the system (18), one should take into account such features as numerical stability of algorithms, density and the dimension of matrices (Golub and Van Loan, 1983) (Heath, 1984).

Equation (18) can be solved via the conjugate gradient algorithm or by the preconditioned conjugate gradient algorithm. Those methods can be especially useful for large and sparse problems, but unfortunately the algorithms converge slowly when the problem is ill-conditioned.

Another approach for solving the normal equation based on factorization of the matrix $\bar{B}$ using Cholesky's method:

$$\bar{B} = R^T R \tag{21}$$

where R is upper triangular, and then x^J is computed by solving the two triangular systems

$$R^T y = \bar{b} \tag{22}$$

$$R x^J = y \tag{23}$$

Despite many useful features of the normal equation method, the method with direct application of Cholesky's partition to the normal equations also has several drawbacks. We mention some of them

- Necessity of explicitly forming and processing $\bar{B}$ according to (19)
- The condition number of $\bar{B}$ is the square of the condition number of A^I_J.

3 Application the QR decomposition

To simplify the discussion we write (16) in the following form:

$$f_I(x) = c^J x^J + (1/2)\|A^I_J x - h^I_J\|^2 + g^I_J \tag{24}$$

where

$$h^I_J = b^I - A^I_J x^J$$
$$g^I_J = c^J x^J$$

In the orthogonal factorization approach a matrix Q is used to reduce A^I_J to the form

$$Q^T A^I_J = \begin{pmatrix} R^I_J \\ 0 \end{pmatrix} \quad Q^T h^I_J = \begin{pmatrix} p_1 \\ p^I_2 \end{pmatrix}$$

where R^I_J is upper triangular. We have

$$f_I(x) = c^J x^J + (1/2)\|R^I_J x - p_1\|^2 + (1/2)\|p_2\|^2 + g^I_J$$

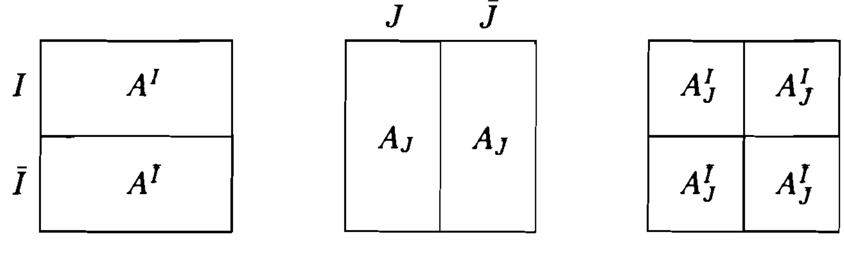
The application of the orthogonal matrix Q does not change L_2–norm and an advantage of such a transformation is that we do not need to save the matrix Q. It can be discarded after it has been applied to the vector h^I_J. Moreover, the matrix R^I_J is the same as the Cholesky's factor of $\bar{B}$ (19) apart from possible sign differences in some rows.

The above Q–R transformation can be carried out by using of Givens rotations which are very attractive for our case (see George and Heath,1980). In our implementation we do not store the orthogonal matrix Q and the obtained matrix R^I_J is used for solution (22)–(23), where the vector $\bar{b}$ is given by (20).

3.1 Update of Q–R decomposition

As we have shown in the description of the active set algorithm, the working sets were changed during sequential steps (compare steps 3, 7). Changes of working sets result in changes of the matrix A^I_J, but only one row or one column can be added or removed from that matrix at a time. This means that the matrix A^I_J which defined the Hessian of minimizing function (24) is changed. Consequently we should update Q–R factorization whenever an index is added to or deleted from the working set. Computing a new factorization ab initio would be much too expensive so we adopted numerically stable methods for updating the Q–R decomposition (e.g. see Golub and Van Loan, 1983, Lawson, Hanson,1974).

To simplify the description we split up the initial matrix A in a way which corresponds to the definition of the working sets (see the Figure below). The contents of matrix A^I_J changes along computation.



We now describe the way of updating in step 7 when an index is added to the working set.

1. If the following holds
$$I_{k+1} = I_k \cup \{r\}, \quad J_{k+1} = J_k$$
then the column r is deleted from $A^{\bar{I}}_J$ and it is added to the matrix A^I_J.

2. If the following holds
$$J_{k+1} = J_k \cup \{l\}, \quad I_{k+1} = I_k$$
then the row l is added to $A^{\bar{I}}_J$ and it is deleted from $A^I_{\bar{J}}$.

Similarly, let us consider changes of the matrix $A^{\bar{I}}_J$ when the working sets are reduced in step 3.

1. If the following holds
$$J_{k+1} = J_k \setminus \{l\}, \quad I_{k+1} = I_k$$
then the column l is deleted from $A^{\bar{I}}_J$ and it is added to $A^{\bar{I}}_{\bar{J}}$.

2. If the following holds
$$I_{k+1} = I_k \setminus \{r\}, \quad J_{k+1} = J_k$$
then the row r is removed from $A^{\bar{I}}_J$ and it is added to A^I_J.

We have just seen that to modify Q–R decomposition of the matrix $A^{\bar{I}}_J$ the following cases should be considered:

1. Adding a column
2. Deleting a column
3. Adding a row
4. Deleting a row

In sequel we shortly describe the above four modifications of the Q–R factorization. Assume that we have the upper triangular matrix $R^{\bar{I}}_J$ which has been obtained after application of the Q–R decomposition to the matrix $A^{\bar{I}}_J$.

3.2 Adding a column

Assume that the column $a^l_{\bar{I}}$ is to be added to the matrix $A^{\bar{I}}_J$.

$$(A^{\bar{I}}_J, a^l_{\bar{I}})$$

We want to obtain a new decomposition with the upper triangular matrix in the form:

$$\begin{pmatrix} R^{\bar{I}}_J & u \\ 0 & \gamma \end{pmatrix}.$$

Where the column vector u is obtained by solving the triangular system of equations

$$(R^{\bar{I}}_J)^T u = (A^{\bar{I}}_J)^T a^l_{\bar{I}}$$

and the scalar γ is calculated in the form

$$\gamma = (\|a^l_{\bar{I}}\|^2 - \|u\|^2)^{1/2}$$

3.3 Deleting a column

Deleting the column l from the matrix A_J^I corresponds to deleting the column l from the matrix R_J^I. Note that the matrix H obtained from R_J^I after deleting l is an upper Hessenberg matrix. This matrix contains some of subdiagonals elements not equal zero. Clearly, the nonzero subdiagonal elements can be zeroed by sequence Givens rotations (Golub and Van Loan, 1983).

3.4 Adding a row

Suppose that we have the upper triangular matrix R_J^I and we wish to obtain an upper triangular of

$$\bar{A} = \begin{pmatrix} a_r^J \\ A_J^I \end{pmatrix}$$

It corresponds to the following Hessenberg matrix

$$H = \begin{pmatrix} a_r^J \\ R_J^I \end{pmatrix}$$

After application a sequence of Gives rotation to the matrix H the nonzero subdiagonal elements can be zeroed.

3.5 Deleting a row

This type of modification of Q–R decomposition is possible in the case when the matrix after removing a row is positive define. Suppose that for an orthogonal matrix Q we have

$$QA_J^I = \begin{pmatrix} R_J^I \\ 0 \end{pmatrix}$$

Note that the matrix Q is not stored. For the deleting row a_r^J we wish to find an upper triangular matrix $\tilde{R}_J^I$, for which we have

$$(\tilde{R}_J^I)^T \tilde{R}_J^I = (R_J^I)^T R_J^I - a_r^J (a_r^J)^T \tag{25}$$

We should determine an orthogonal matrix U as the product of Givens rotations, that the following holds

$$U \begin{pmatrix} R_J^I \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{R}_{J_r}^I \\ a_r^J \end{pmatrix} \tag{26}$$

Note that due to $U^T U = I$ the equation (25) holds.

The matrix U is chosen in such a way, that

$$U \begin{pmatrix} u \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

where u and α are determined as the solution of the system

$$(R_J^I)^T u = a_r^J$$

and

$$\alpha = (1 - \|u\|^2)^{1/2}$$

If the Givens rotations which defines U are then used as in (26), the desired matrix $\tilde{R}_J^I$ will be found.

4 Final remarks

We have developed a stable numerical method for minimization of the piecewise quadratic function with lower and upper bounds. Such problems arise, for example, in application of the multiplier method to linear programming problems. The presented approach can be also useful for problems in which the matrices A_1 and A_2 are large and sparse. In those cases, the methods for symbolic generation of sparse structure for storing the factors R^I_J can be adopted in the similar way as in (Björck, 1988).

References

Björck Å, (1988). A direct method for sparse least squares problems with lower and upper bounds. Numerische Mathematik,Vol.54,19–32.

Flecher, R. (1981). Practical methods of optimization, vol II, Constrained optimization, Wiley, New York.

Gill P.E, W. Murray, M.H. Wright, (1981) Practical Optimization. Academic Press. London, New York.

George A. and M.T. Heath, (1980) Solution of Sparse Linear Least Squares Problem Using Givens Rotations. Linear Algebra and its Application Vol.34,69–83.

Golub, G.H. and C.F. Van Loan (1983). Matrix Computations, Johns Hopkins University Press, Baltimore, Maryland.

Heath, M.T. (1984). Numerical Methods for Large Sparse Linear Least Squares Problems. SIAM J. Sci. Stat. Comput., Vol. 5, No. 3.

Lawson, C.L and R.J. Hadson, (1974). Solving Least Squares Problems. Prentice–Hall. New Jersey.

Rckafellar,R.T, (1976). Augmented lagrangians and Applications of the Proximal Point Algorithm in Convex Programming. Mathematics of Operations Research, Vol. 1, No 3,97–116.