A COMBINED VARIABLE METRIC - CONJUGATE GRADIENT ALGORITHM FOR A CLASS OF LARGE SCALE UNCONSTRAINED MINIMIZATION PROBLEMS

Shmuel S. Oren
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California

Abstract

An algorithm is being presented for a special class of unconstrained minimization problems. The algorithm exploits the special structure of the Hessian in the problems under consideration, It is based on applying Bertsekas' [1] Scaled Partial Conjugate Gradient method with respect to a metric that is updated by the Rank One update, using gradients obtained in the preceeding steps. Two classes of problems are presented having the structure assumed in designing the proposed algorithm. In both cases the algorithm uses only first derivative information. Furthermore, it possesses quadratic termination in considerably fewer steps than the number of variables.

1. Introduction

Variable Metric algorithms are considered to be the most advanced methods for solving unconstrained minimization problems of the form: min $f(x)$ where $x \in R^n$ and $f \in C^2$. The basic recursion in these algorithms is analog to the one used in Newton Raphson method having the form:

$$x_{k+1} = x_k - \alpha_k D_k g_k \qquad (1)$$

In this recursion x_k denotes the k^{th} approximation to the minimum, g_k is the gradient at x_k, α_k is a stepsize parameter selected to ensure some convergence criteria, while D_k is an nxn matrix approximating the inverse Hessian $[\nabla^2 f(x)]^{-1}$. The approximations D_k are inferred from the gradients at previous iterations and updated as new gradients become available so as to satisfy the "quasi Newton condition"

$$D_k(g_k - g_{k-1}) = x_k - x_{k-1} \ . \qquad (2)$$

The main motivation underlying such procedures is to capture the second order convergence properties of Newton's method while avoiding the expensive calculation of second derivatives.

The first Variable Metric algorithm was invented by Davidon [4] and further developed and simplified by Fletcher and Powell [5]. Since then a vast literature has been published on this subject. Many of these contributions propose alternative updating procedures for D_k and contain computational results comparing the various computational schemes. However, practically all the theoretical and computational work in this area has been directed toward solving small problems in which the number of variables rarely exceeds fifty.

It is evident even from the above brief description of Variable Metric methods, that the use of such algorithms for large scale problems is limited by the computational and storage requirement involved in maintaining D_k. In such cases it becomes advantageous to use Conjugate Gradient algorithms such as Fletcher Reeves [6] method. These algorithms are usually slower than variable metric methods as they lack the memory features of the later techniques. On the other hand, conjugate gradient methods have the advantage of generating the search directions directly, avoiding the need to store and update an nxn matrix which becomes prohibitive for large n.

The above considerations are relevant as long as *no structural information about* $f(x)$ *is* being utilized. Fortunately, in many of the large scale problem the objective function has some special structure. The expense involved in solving such problems, and computational feasibility considerations, justify the development of special purpose algorithms that exploit the special structure of the objective function. One of the central themes of large scale mathematical

programming has been to develop such special purpose algorithms. This approach, however, has not influenced yet the development of Variable Metric type algorithms for large scale problems.

This paper attempts to follow the aforementioned theme of large scale mathematical programming and proposes an algorithm for a special class of unconstrained minimization problems. More specifically, we focus on problems where the Hessian matrix $\nabla^2 f(x) = M+R$ where M is a block diagonal matrix with blocks of dimension m or less and R is a matrix of rank r, with m and r significantly lower than the dimension of x. Such functions arise for instance from a special class of control problems or in solving certain resource allocation problems by penalty or multiplier methods. Bertsekas [1] who addressed the aforementioned class of optimal control problems, proposed an algorithm in which the directions of search are generated using Fletcher Reeves [6] algorithm with respect to the metric M^{-1} restarted every $r+1$ steps. The matrix M^{-1}, is evaluated in this method at the beginning of each cycle from second derivative information. Bertsekas has shown that this algorithm converges superlinearly and for a quadratic function it terminates in one cycle (i.e. $r+1$ steps).

The algorithm proposed in this paper relates to Bertsekas [1] method in the sense that Variable Metric algorithms relate to Newton's method. The search direction at each step are generated using Fletcher Reeves' [6] algorithm with respect to a metric D, restarted every $r+1$ steps. The nxn matrix D is an approximation to the matrix M^{-1} updated by the Broyden's [2] Rank-One updating formula using the gradients computed at each step. Since M^{-1} is block diagonal we force D to have the same structure which enables us to update and store each block individually. Consequently, for a quadratic function, $D=M^{-1}$ after at most m steps implying "quadratic termination" in significantly fewer steps than n.

Following is an outline for the remainder of this paper. In section 2 we present the theoretical foundation and a conceptual outline of the proposed algorithm. In Sections 3 and 4 we specialize the algorithm to a class of resource allocation problems and to the optimal control problems considered by Bertsekas. Section 5 contains the conclusions and some remarks on the proposed method.

2. Theoretical Foundation and the Conceptual Algorithm

The Fletcher Reeves [6] conjugate gradient algorithm can be described as follows: Starting with an initial point x_0 and $d_0 = -g_0$,

$$x_{k+1} = x_k + \alpha_k d_k \tag{3}$$

where

$$\alpha_k = \arg\min f(x_k+\alpha d_k) \tag{4}$$

$$d_k = -g_k + \beta_{k-1} d_{k-1} \tag{5}$$

and

$$\beta_{k-1} = \| g_k \|^2 / \| g_{k-1} \|^2 \tag{6}$$

It is shown in Luenberger [7] that if $f(x)$ is a positive definite quadratic function and $\nabla^2 f(x)$ has s distinct eigenvalues then the above procedure converges to the minimum of $f(x)$ in at most s steps. When $\nabla^2 f(x) = M+R$ where M is positive definite and R has rank r we can define $y = M^{1/2}x$. Then,

$$\nabla_y f(M^{-1/2}y) = M^{-1/2} g_k \tag{7}$$

and

$$\nabla_y^2 f(M^{-1/2}y) = M^{-1/2} \nabla_x^2 f(x) M^{-1/2} = I+M^{-1/2}RM^{-1/2} \tag{8}$$

Clearly $\nabla_y^2 f(M^{-1/2}y)$ has only $r+1$ distinct eigenvalues. Thus, applying Fletcher Reeves algorithm after changing the variables from x to y will yield the minimum in at most $r+1$ steps. The

above change of variables can be implementd implicitly by writing eq. (3) to (6) for y_k and then substituting $y_k = M^{1/2}x_k$. The resulting algorithm is similar to the original one, but now $d_0=-M^{-1}g_0$ and for $k>0$:

$$d_k = -M^{-1}g_k + \beta_{k-1}d_{k-1} \tag{9}$$

with

$$\beta_{k-1} = (g_k' M^{-1}g_k) / (g_{k-1}'M^{-1}g_{k-1}) \tag{10}$$

The above algorithm can be generalized to non quadratic functions by restarting it every $r+1$ steps with M^{-1} evaluated at the beginning of each cycle and kept fixed during the entire cycle.

The above implementation which has been proposed by Bertsekas [1] results in superlinear convergence but requires the evaluation and inversion of the second derivative matrices composing M at the beginning of each cycle. The alternative approach proposed in this paper avoids the need for second derivative information as well as matrix inversions. In our implementation the matrix M^{-1} is substituted by an approximation D inferred from gradients generated in preceeding iterations and updated successively as new gradients become available.

The following theorem states the properties of Broyden's [2] Rank-One updating formula that forms the basis for the proposed algorithm.

Theorem 1

Let H *be a positive definite symmetric* nxn *matrix and* $\{r_0, ..., r_{n-1}\}$ *and* $\{v_0, ..., v_{n-1}\}$ *sequences of linearly independent vectors such that* $v_k = Hr_k$ *for* $k = 0 ... n-1$. *Let* D_k *be* nxn *matrices such that*

$$D_{k+1} = D_k + (r_k - D_kv_k)(r_k - D_kv_k)' / (r_k - D_kv_k)'v_k \tag{11}$$

and D_0 *is an arbitrary* nxn *positive semi-definite symmetric matrix. Then,* $D_{k+1}v_j = r_j$ *for* $j \le k$.

The above theorem is well known (see for example Luenberger [7]) and its proof will hence be ommited. In particular the theorem implies that $D_n = H^{-1}$; i.e., the n^{th} approximation will be identical to H^{-1} regardless of the initial approximation D_0.

In the specific problem under consideration, assuming the function is quadratic, we have $(M+R)p_k=q_k$ where $p_k = x_k-x_{k-1}$ and $q_k = g_k-g_{k-1}$. Consequently,

$$Mp_k = \hat{q}_k = q_k - Rp_k \tag{12}$$

We shall assume that either Rp_k is available or $\hat{q}_k$ can be obtained directly. In view of (12) and Theorem 1 we can then obtain M^{-1} by repeated application of (11) with p_k and $\hat{q}_k$ taking the role of r_k and v_k respectively. This would require, however, n updates and considerable storage. The computational and storage requirement can be radically reduced by exploiting the fact that M is block diagonal. If we partition the vectors p_k and $\hat{q}_k$ into n segments corresponding to the blocks in M then eq. (12) can be written as

$$M^ip^i_k = \hat{q}^i_k \qquad i = 1, ..., h \tag{13}$$

where M^i is the i^{th} block and p^i_k, $\hat{q}^i_k$ are the corresponding segments of p_k and $\hat{q}_k$. Consequently, we can use (11) to obtain each block $(M^i)^{-1}$ individually. Following this procedure enables us to obtain M^{-1} using only m pairs of vectors (p_k,q_k) where m is the dimension of the largest block in M.

In the remainder of this section we outline a conceptual algorithm based on the above observations. The algorithm is designed to minimize an unconstrained function f(x) whose Hessian $\nabla^2 f(x) = M+R$ where M is a symmetric block diagonal matrix consisting of h blocks having dimensions m^i ($i = 1, ..., h$), while R has rank no greater than r. We use the notation x^i

to denote the segment of the vector x corresponding to M^i. Thus, $x = (x^1, x^2, \ldots x^h)$ and $x^i \in R^{m_i}$.

Algorithm 1.

Start with an initial point $x_0 = (x^1{}_0, x^2{}_0, \ldots x^h{}_0)$ and h positive definite symmetric $m_i x m_i$ matrices $D^i{}_0$ for $i = 1, \ldots, h$.

Step 1: Obtain $g_0 = \nabla f(x_0)$ and set $d^i{}_0 = -D^i{}_0 g^i{}_0$ for $i=1, \ldots, h$

Step 2: Compute

$$x_{k+1} = x_k + \alpha_k d_k \tag{14}$$

where α_k minimize $f(x_k + \alpha d_k)$ with respect to α

Step 3: Obtain $g_{k+1} = \nabla f(x_{k+1})$

$$p_k = x_{k+1} - x_k \tag{15}$$

$$q_k = g_{k+1} - g_k - R p_k \tag{16}$$

Step 4: For $i = 1, \ldots, h$ compute

$$v^i{}_k = p^i{}_k - D^i{}_k q^i{}_k \tag{17}$$

$$\left.\begin{aligned} D^i{}_{k+1} &= D^i{}_k \qquad \text{if } v^i{}_k{}' q^i{}_k \le 0 \\ D^i{}_{k+1} &= D^i{}_k + v^i{}_k v^i{}_k{}' / v^i{}_k{}' q^i{}_k \qquad \text{otherwise} \end{aligned}\right\} \tag{18}$$

Step 5: If $k<r$, compute for $i=1, \ldots, h$

$$d^i{}_{k+1} = -D^i{}_0 g^i{}_{k+1} + \beta_k d^i{}_k \ , \tag{19}$$

where

$$\beta_k = \left(\sum_{i=1}^{h} g^i{}_{k+1}{}' D_0{}^i g^i{}_{k+1}\right) / \sum_{i=1}^{h} g^i{}_k{}' D_0{}^i g_k{}^i \ . \tag{20}$$

Then increment k by 1 and go to Step 2.

Step 6: If $k=r$, reset k to 0, set $x_0 = x_{r+1}$, and $D^i{}_0 = D^i{}_{r+1}$ for $i=1, \ldots, h$, then go to Step 1.

We note that though the matrices $D^i{}_k$ are updated on each iteration, the matrices used in the calculation of β_k are kept fixed during a cycle. This is required in order not to destroy the conjugency of the search direction that is needed to assure quadratic termination. It should be also noted that in Step 4 we do not update $D^i{}_k$ unless the denominator in the rank one correction term is positive. This rule is a crude stabilization device included just to indicate the need for some device that will assure positive definitness of the $D^i{}_k$ matrices. In implementing Algorithm 1, one can use any of the stabilization approaches proposed in the literature for the Rank-One update. Such approaches have been suggested for instance by Murtagh and Sargent [8] and more recently by Cullum and Brayton [3].

3. A Resource Allocation Problem

We now consider the class of resource allocation problem having the form

$$\left.\begin{aligned} &\min f(x) = \sum_{i=1}^{h} f_i(x^i) \\ &\text{subject to} \\ &Ax = b \end{aligned}\right\} \tag{21}$$

where $x = (x^1, x^2, \ldots x^h)$, $x \in R^n$, $x^i \in R^{m_i}$, $b \in R^r$ and A is an $r x n$ matrix.

In this problem the objective function is partially separable in the sense that the decision vector x can be partitioned into segments each of which affects only one term in the objective function. All the decision variables, however, are related through a relatively small number of linear

constraints. This would be a typical formulation in a multiperiod resource allocation problem in which the variables have to satisfy some linear resource or budget constraints.

One possible approach for solving this type of problem is using a multiplier or penalty function method. Such an approach involves the repeated unconstrained minimization of a penalty or penalized Lagrangian function having the form

$$L(x) = \sum_{i=1}^{h} f_i(x^i) + \mu \| Ax - b \|^2 \tag{22}$$

or

$$L(x) = \sum_{i=1}^{h} f_i(x^i) + \lambda' Ax + \mu \| Ax - b \|^2 \tag{23}$$

For either of the functions (22) or (23), the Hessian is:

$$\nabla^2 L(x) = \nabla^2 f(x) + 2\mu A'A$$

The matrix $\nabla^2 f(x)$, however, is block diagonal with the i^{th} block being $\nabla^2 f_i(x^i)$. We thus have the structure assumed in the design of Algorithm 1 with $M_i = \nabla^2 f_i(x^i)$ and $R = 2\mu A'A$. Consequently, the penalty or penalized Lagrangian function given by (22) or (23) can be minimized using Algorithm 1 restarted every $r+1$ steps (i.e., number of constraints plus one).

4. A Class of Optimal Control Problems

We address now the class of optimal control problems considered by Bertsekas [1]. These problems have the form

$$\left.\begin{aligned} &J(u^0, \ldots, u^N) = G(x^N) + \sum_{k=0}^{N-1} l_i(u^i) \quad . \\ &\text{Subject to} \\ &x^{i+1} = A_i x^i + f_i(u^i) \qquad i = 0, 1, \ldots, N-1; \qquad x^0 \text{ given.} \end{aligned}\right\} \tag{24}$$

Here $x^i \in R^n$ denotes the state, $u^i \in R^m$ is the control, A_i is an $n \times n$ matrix, f_i: $R^m \rightarrow R^n$, $G: R^n \rightarrow R$ and $l_i: R^m \rightarrow R$.

The Hamiltonian for this problem is

$$H_i(x^i, u^i, \lambda^{i+1}) = l_i(u^i) + \lambda^{i+1}{}' [A_i x^i + f_i(u^i)], \qquad i = 0, \ldots, N-1. \tag{25}$$

λ^i denotes here the costate and is defined by the adjoint equations

$$\left.\begin{aligned} &\lambda^i(u) = A_i' \lambda^{i+1}(u) \quad , \qquad i = 1, \ldots, N-1 \\ &\lambda^N(u) = \partial G / \partial x^N (x^N(u)) \end{aligned}\right\} \tag{26}$$

The gradient of the cost functional J with respect to the mN dimensional control vector u is given by

$$\nabla J(u) = \left[\frac{\partial H_0}{\partial u^0}(u), \ \ldots, \ \frac{\partial H_{N-1}}{\partial u^{N-1}}(u) \right] \tag{27}$$

where

$$\frac{\partial H_i}{\partial u^i}(u) = \frac{\partial l_i}{\partial u^i}(u^i) + \lambda^{i+1}(u)' \frac{\partial f_i}{\partial u^i}(u^i) \tag{28}$$

In eq. (28) $\partial l_i / \partial u^i$ is a row vector denoting the gradient of l_i with respect to u^i and $\partial f_i / \partial u^i$ is the Jacobian of f_i with respect to u^i. The Hessian of J has the form

$$\nabla^2 J = \frac{\partial^2 H}{\partial u^2}(u) + M(u) \frac{\partial^2 G}{\partial (x^N)^2}(u)\, M(u)' \tag{29}$$

where $\partial^2 H/\partial u^2(u)$ is the block diagonal matrix

$$\frac{\partial^2 H}{\partial u^2}(u) = \begin{bmatrix} \frac{\partial^2 H_0}{\partial (u^0)^2}(u) & 0 & & \\ 0 & \frac{\partial^2 H_1}{\partial (u^1)^2}(u) & 0 & \\ & & & \\ & 0 & & \frac{\partial^2 H_{N-1}}{\partial (u^{N-1})^2}(u) \end{bmatrix} \tag{30}$$

and $M(u)'$ is the nxNm matrix

$$M(u)' = \left[A_{N-1} \cdots A_1 \frac{\partial f_0}{\partial u^0}, \ldots, A_{N-1} \frac{\partial f_{N-2}}{\partial u^{N-2}}, \frac{\partial f_{N-1}}{\partial u^{N-1}} \right]. \tag{31}$$

In view of this structure of the Hessian, the problem described in (24) can be conceptually solved by Algorithm 1 with a cycle length equal to the rank of $\partial^2 G/\partial (x^N)^2$ plus one. Such a procedure, however, would be impractical as it requires the evaluation of $M(u)[\partial^2 G/\partial (x^N)^2]M(u)'$ at each iteration. Fortunately, this can be avoided by replacing (16) with a scheme that evaluates q directly as a first order approximation to $(\partial^2 H/\partial u^2)\Delta u$.

The first order approximation to the change in $\nabla J(u)$ due to a change Δu in the control is given by

$$\nabla J(u+\Delta u) - \nabla J(u) \simeq \Delta u' \frac{\partial^2 H}{\partial u^2}(u) + \left[[\lambda^1(u+\Delta u) - \lambda^1(u)]' \frac{\partial f_0}{\partial u^0}(u^0), \ldots, [\lambda^N(u+\Delta u) - \lambda^N(u)]' \frac{\partial f_{N-1}}{\partial u^{N-1}}(u^N) \right] \tag{32}$$

Consequently q can be obtained by subtracting the second term in the right hand side of (32) from the gradient difference. Due to the special structure of $\partial^2 H/\partial u^2$ and in view of (27) and (28), eq. (32) can be decomposed into N equations of the form

$$\Delta u_i' \frac{\partial^2 H_i}{\partial (u^i)^2}(u) \simeq \frac{\partial l_i}{\partial u^i}(u^i+\Delta u^i) - \frac{\partial l_i}{\partial u^i}(u^i) + \lambda^{i+1}(u+\Delta u)' \left[\frac{\partial f_i}{\partial u^i}(u^i+\Delta u^i) - \frac{\partial f_i}{\partial u^i}(u^i) \right], \qquad i=0, 1, \ldots, N-1 \tag{33}$$

The right hand side in (33) is then an expression for the segment of q corresponding to the i^{th} block in $\partial^2 H/\partial u^2$.

Based on the above considerations we describe now a more specific version of Algorithm 1 designed for the optimal control problems addressed in this section. In this algorithm we use subscripts to denote iteration number and superscripts to denote time period. The rank of the matrix $\partial^2 G/\partial x^N$ is denoted by r. For convenience, we have also changed the order of steps.

Algorithm 2

Start with a positive definite simetric matrix $D^i{}_0$ and any $u^i{}_0$ for i=0, ..., N-1

Step 1: Calculate $x^i{}_k$, i = 0, ..., N-1, using

$$x^{i+1}{}_k = A_i x^i{}_k + f_i(u^i{}_k), \text{ with } x^0{}_k = x^0 \tag{34}$$

and $\lambda^i{}_k$, i = 1, ..., N-1, using

$$\lambda^i{}_k = A_i{}' \lambda^{i+1}{}_k, \text{ with } \lambda^N{}_k = \partial G/\partial x^N(x^N{}_i) \tag{35}$$

Step 2: For i = 0, ..., N-1, calculate

$$L^i{}_k = \frac{\partial l_i}{\partial u^k}(u^i{}_k)', \quad \text{and} \quad F^i{}_k = \frac{\partial f_i}{\partial u^i}(u^i{}_k)' \quad . \tag{36}$$

then obtain

$$\nabla J^i{}_k = L^i{}_k + F^i{}_k \lambda^{i+1}{}_k \tag{37}$$

Step 3: If k=0 then go to Step 6, otherwise for i = 0, ..., N-1, calculate

$$\Delta u^i{}_k = u^i{}_k - u^i{}_{k-1} \tag{38}$$

$$q^i{}_k = L^i{}_k - L^i{}_{k-1} + (F^i{}_k - F^i{}_{k-1})\lambda^{i+1}{}_k \tag{39}$$

and

$$v^i{}_k = \Delta u^i{}_k - D^i{}_k q^i{}_k \tag{40}$$

Step 4: For i = 0, ..., N-1;

$$\left.\begin{array}{l} \text{if} \quad v^i{}_k{}' q^i{}_k \leq 0 \text{ set } D^i{}_k = D^i{}_{k-1} \\ \text{otherwise } D^i{}_k = D^i{}_{k-1} + (1/v^i{}_k{}' q_k i)\, v^i{}_k\, v^i{}_k{}' \end{array}\right\} \tag{41}$$

Step 5: If $k < r+1$, go to Step 6. Otherwise, reset k to 0, then set $D^i{}_0 = D^i{}_{r+1}$, $u^i{}_0 = u^i{}_{r+1}$ and $\nabla J^i{}_0 = \nabla J^i{}_{r+1}$ for i = 0, ..., N-1 .

Step 6: For i = 0, ..., N-1; if k = 0 set $d^i{}_0 = -D^i{}_0 \nabla J^i{}_0$, otherwise

$$d^i{}_k = -D^i{}_0 \nabla J^i{}_k + \beta_{k-1}\, d^i{}_{k-1} \tag{42}$$

where

$$\beta_{k-1} = \Big(\sum_{i=0}^{N-1} \nabla J^i{}_k{}'\, D^i{}_0 \nabla J^i{}_k\Big) \Big/ \sum_{i=0}^{N-1} \nabla J^i{}_{k-1}{}'\, D^i{}_0\, \nabla J^i{}_{k-1} \tag{43}$$

Step 7: For i = 0, ..., N-1 calculate

$$u^i{}_{k+1} = u^i{}_k + \alpha_k\, d^i{}_k \tag{44}$$

where α_k minimizes $J(u^0{}_k + \alpha d^0{}_k, \ldots, u^{N-1}{}_k + \alpha d^{N-1}{}_k)$.

Then, increment k by 1 and return to Step 1.

It should be noted that since D^i_0 is replaced only at the end of $r+1$ steps the updating performed in Step 3 and 4 could be done after the cycle is completed. Such an approach, however, would require more storage since then we have to store Δu^i_k and q^i_k for all $i=0,\ldots,N-1$ and $k=1,\ldots,r+1$. For a linear quadratic problem where $f_i(u^i) = B_i u^i$ and $l_i(u^i) = u^{i'} R_i u^i$ we have $q^i_k = R_i \Delta u^i_k$. Thus, by Theorem 1 it will take m updatings to obtain R_i^{-1}. Assuming that $r+1 \geq m$, the second cycle will be a properly scaled partial conjugate gradient cycle and will thus converge to the exact minimum by the end of that cycle.

5. Conclusions

General purpose Variable Metric algorithms are impractical for large scale optimization problems due to their high computational and storage requirements. These costs, however, can be reduced by specializing such algorithms to specific classes of problems and exploiting the special structure of such problem to reduce computational and storage requirements. The paper implements this philosophy for a class of problems in which the Hessian matrix consist of a sum of a block diagonal matrix and a low rank matrix. We use a rank-one updating formula to approximate the inverse of the first part of the Hessian. We then use that approximation in a Scaled Partial Conjugate Gradient algorithm restarted every $r+1$ steps where r is the rank of the second term in the Hessian. Due to the block diagonal form of the first part of the Hessian being approximated, the block's can be updated and stored individually. This procedure considerably reduces the storage requirements by maintaining sparcity. Furthermore for a quadratic problem the approximation becomes exact after as many steps as the dimension of the largest blocks. The resulting algorithm possesses "quadratic termination" in a number of step significantly lower than the number of variables.

The use of the Rank One update was motivated by the fact that it permits the use of arbitrary independent updating vectors. This property is crucial since the search directions in our approach are different from the vectors obtained by multiplying the current matrix approximation times the gradient. On the other hand, the use of the Rank One formula raises stabilization problems as it does not guarantee the positive definitness of the approximations even when applied to a positive definite quadratic function. Various stabilization schemes have been proposed in the literature and can be used in implementing our approach. Much of the stabilization problems, however, can be avoided by using positive definite initial approximations whose eigenvalues are all below the eigenvalues of the matrix being approximated. It can be shown (see Luenberger [7]) that in the quadratic case such an initial approximation guarantees that all the rank one correction terms will have positive denominators, and hence, the approximations will be positive definite. A simple initial approximation satisfying the above requirement would be, for instance, the identity scaled through division by an upper bound on the norm of the matrix being approximated.

We presented two classes of problems leading to the structure under consideration and discussed the potential implementation of the proposed approach to these problems. In both cases the algorithm uses only first derivatives and doesn't require any matrix inversions. Numerical experiments, however, with this method are limited so far and it still remains to demonstrate practical value of the proposed method.

References:

1. Bertsekas, D. P., "Partial Conjugate Gradient Methods for a Class of Optimal Control Problems", *IEEE Transaction on Automatic Control*, Vol. AC-19, pp. 209-217, 1974.

2. Broyden, C. G., "Quasi-Newton Methods and Their Application to Function Minimization", *Math. Comp.*, Vol. 21, pp 368-381, 1967.

3. Cullum, Jane and Brayton, R. K., Some Remarks on the Symmetric Rank One Update, IBM Research Center Report 6157, Yorktown Heights, New York, 1976.

4. Davidon, W. C., "Variable Metric Method for Minimization," A.E.C. Research and Development Rep. ANL-5990 (Rev.), 1959.

5. Fletcher, R. and Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization", *Comp. Jnl.*, Vol. 6, pp. 163-168, 1963.

6. Fletcher, R. and Reeves, C. M., "Function Minimization by Conjugate Gradients", *Comp. Jnl.*, Vol. 7, pp. 149-154, 1964.

7. Luenberger, D. G., *"Introduction to Linear and Nonlinear Programming"*, Reading, Mass: Addison-Wesley 1973.

8. Murtagh, B. A. and R. W. H. Sargent, Computational Experience with Quadratically Convergent Minimization Methods, *Comp. Jnl.* 13, 185-194, 1970.