ON THE EVALUATION OF THE MULTIPARAMETER ADJUSTMENT UNCONSTRAINED NLP METHOD

M. Nazih El-Derini
Computer Science and Automatic Control Department
Faculty of Engineering, Alexandria University
Alexandria, Egypt

ABSTRACT

This paper concerns with multiparameter adjustement method. This method needs to be combined with an other unconstrained algorithm to minimize the objective function w.r.t. its two parameters. The combined multiparameter adjustment methods are tested for eight different Problems, and results are compared to evaluate them. It is found that the DFP algorithm enhances the method and makes it superperforms other multiparameter adjustment methods. The effects of the maximum number of iterations of DFP, the choice of the starting point and the DSC tolerance are also examined. A comparison between other unconstrained NLP algorithms and the combined multiparameter adjustment method with DFP is also discussed. Results put this method in the same category as DFP, Flecher and Rosenbrock methods from the superiority point of view.

INTRODUCTION

Several varieties of penalty function methods have been proposed, but the essence of all the methods is to transform a constrained nonlinear programming problem into an unconstrained problem or a sequence of unconstrained problems. Thus several of the important constrained NLP algorithms require the use of an effective unconstrained minimization procedure. The advantage of minimizing the unconstrained problem rather than the constrained problem is that much simpler algorithm can be used for the optimization. As might be expected, no single one of the nonlinear programming algorithms has proved to be superior for all nonlinear programming problems under all circumstances.

One of the earliest techniques for obtaining the minimum of a function is the steepest-descent method and its modifications[1,2]. the main handicap in using this method is its dependence on the relative scaling of the decision variables. Second-derivative methods, among which the best-known is Newton's method[3]. This method requires the calculation of the inverse of the Hessian matrix of the second derivatives of the function. Newton's method depends heavily on the starting point and it breaks down when the Hessian matrix is singular. For large problems, a major part of the cost can be attributed to the solution of the Newton equations.

For these problems, the conjugate direction methods [4,5] are widely used, due to their low storage requirements. These methods of conjugate directions have been found very satisfactory when the first derivatives of the function are available. For large scale problems, the conjugate direction method is truncated in order to obtain an approximate solution to the Newton equations[6-9]. An other class of methods termed variable metric exists that replaces the inverse of the local Hessian matrix or its inverse by an approximate metric which uses information from only first order derivatives to do so [10-13]. Most of these methods generate set of conjugate directions using gradients.

In some cases, it is laborious or practically impossible to calculate the first derivatives; consequently, there is a definite need for minimization procedures which do not require them. Some of these procedures are based on conjugate direction, Others change one variable at a time in a pattern move search [14-17].

This paper concerns with multiparameter adjustment method which is classified as a conjugate direction method. An other NLP algorithm is needed to be combined with the multiparameter adjustment method to solve the NLP problems in the two parameters of the method. The combined multiparameter adjustment methods are tested for eight different problems. The testing problems have different number of variables and different degrees of nonlinearity. Results are compared to evaluate these methods.

One criterion of evaluation is the robustness which is the success in obtaining an optimal solution for a wide range of problems, i.e. can an algorithm be expected to solve most problems? of course any algorithm can be defeated by a suitably designed problem. Furthermore we cannot expect that an algorithm will pick out the global minimum if the problem has more than one minimum, but it should at lest reach a local minimum to be considered successful. Other evaluation criteria are the number of functional evaluations and the computer time to termination.

It is found that the DFP algorithm enhances the multiparameter adjustment method and makes it superperforms other multiparameter adjustment methods which are combined with Newton method, gradient methods, or other conjugate direction methods.

In this paper, the effect of some parameters is also examined on the performance of the combined multiparameter adjustment method with DFP. These parameters are the maximum number of iterations of DFP, the choice of the starting point and the DSC tolerance used in minimizing along the line within the DFP method.

Then a comparison among the combined multiparameter adjustment method with DFP and other unconstrained NLP algorithms such as steepest descent, Fletcher-Reeves, Newton with and without minimization along the line, Polak-Rebiere, Pearson-2, DFP, and Rosenbrock, is also discussed for the same test functions. The result of this comparison puts this method in the same category as DFP, Fletcher and Rosenbrock methods from the superiority Point of view.

MULTIPARAMETER ADJUSTMENT METHOD

The problem to be considered is

$$\text{Min } f(x) \quad , \quad H \subset R^n \tag{1}$$

$$x \in H$$

where f is the objective function and it is assumed to be a nonlinear unimodal function.

$x = [\, x_1\ x_2\ \dots\ x_n]^T$ where x is the design vector and x_1, x_2, and x_n are the design variables. We wish to find a point x^*, such that, if $\delta > 0$, then

$$f(x^*) < f(x), \text{ for all } x: \; \| x - x^* \| < \delta \tag{2}$$

To solve system (1), the following iteration is considered:

$$x^{(k-1)} = x^{(k)} - t_1^{(k)}\, \Delta f(x^{(k)}) + t_2^{(k)}\, \Delta\, x^{(k-1)} \tag{3}$$

where $\nabla f\,(x^{(k)})$ is the gradient of the objective function f at the point $x^{(k)}$ and $\Delta\, x^{(k-1)}$ is the difference vector between point $x^{(k)}$ and $x^{(k-1)}$.

$$\Delta x^{(k-1)} = x^{(k)} - x^{(k-1)} \tag{4}$$

$t^{(k)}_1$ and $t^{(k)}_2$ are two parameters to be selected in each search direction to minimize $f\,(x^{(k)} - t^{(k)}_1\, \nabla f(x^{(k)}) + t_2^{(k)}\, \Delta x^{(k-1)})$.

Thus an other unconstrained NLP algorithm in two variables is needed to be combined with the multiparamter adjustment method to compute the two parameters $t^{(k)}_1$ and $t_2^{(k)}$. The superscript (k) indicates the iteration number. All unconstrained NLP algorithms in two variables use one-dimensional search for minimization along the line. The quadratic interpolation procedure, DSC (Davidon, Swann, and Compay), is the one-dimensional search methods used in these multi-dimensional search in the present paper.

The major steps in the algorithm are:

1. Given $x^{(0)}$ and take on the first iteration $\Delta x^{(0)} = 0$,

   i.e. $x^{(1)} = x^{(0)}$

2. For the $k^{\underline{th}}$ step compute $x^{(K)}$, $\nabla f(x^{(k)})$ and

   $\Delta\, x^{(k-1)} = x^{(k)} - x^{(k-1)}$

3. Evaluate $t_1^{(k)}$ and $t_2^{(k)}$which minimize the function

   $f\,(x^{(k)} - t_1^{(k)}\, \nabla f\,(x^{(k)}) + t_2 \Delta\, x^{(k-1)})$ by an efficient two dimensional search.

4. Compute x at iteration (k+1)

   $x^{(k+1)} = x^{(k)} - t_1^{(k)}\, \nabla f(x^{(k)}) + t_2^{(k)}\, \Delta x^{(k-1)})$

5. Test for stopping criteria, terminate when

i) $\| \nabla f(x^{(k+1)}) \| < \in$

ii) $\| x^{(k+1)} - x^{(k)} \|$ AND $\| f(x^{(k+1)}) - f(x^{(k)}) \| < \in_1$

where $\in$ and $\in_1$ are the allowed tollerances.
Otherwise Go To Step 2.
Every $(n+1)$ iterations the algorithm is restarted with $x^{(k+1)} = 0$

The degree of precision in the solution depends upon the termination criteria used to end the computation. The same relative precision in the optimal point x^* and in the optimal function $f(x^*)$ is the joint base for stopping the search in each algorithm. The reason for using such stopping criterion is simply because termination could be premature if the change in f (x) is used solely for a flat plateau, or if the change in x is used solely for a steep slope.

EVALUATION OF MULTIPARAMETER ADJUSTMENT METHODS

The multiparameter adjustment algorithm is combined with four different unconstrained algorithms; steepest descent, fletcher-Reeve, Newton, and DFP. These algorithms are chosen to be from different categories; gradient method, conjugate direction method, second derivative method, and variable metric method respectively.

The convergence of the combined methods is tested by applying such methods and other basic optimization techniques to some standard test functions for which exact optimal points are known. These test functions are given in the Appendix. Results are compared to evaluate these methods.

The major and most important criterion of evaluation which is taken at this stage is the success or failure of a given algorithm to solve a given problem. This criterion is chosen because the ability of an algorithm to solve a wide variety of problems is the most valuable feature to the user of a programming code.

Table 1. Robustness of combined multiparameter adjustment methods.

Test function / Combined Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
Steepest Descent	*	0.0	0.0	0.0	*	0.0	0.0	0.0
Fletcher Reeve	0.16	0.25	0.25	0.0	0.40	1.0+	0.0	0.1
Newton	1.0	1.0	0.0	1.0	1.0	0.25	0.0	0.0
DFP	1.0	0.5	0.80	0.50	1.0	1.0+	0.50	0.5+

* Converges only near the optimal
\+ Converges to local minimum

Table 2. The effect of changing the starting point.

Function/ DFP Tol./ DSC Tol.	Initial point $X^{(0)}$	Final point X^*	Final function (X^*)	N	T	N(F)	N(G)	N(FT)
F_1/E-6/ E-3	$[-5, -3]^T$	$[1.000118,1.000222]^T$	3.203294E-8	2	58	956	6	84
	$[10,10]^T$	$[1.000015,1.000021]^T$	7.977799E-9	2	106	1909	6	134
F_1/E-4/ E-2	$[-1,0.1]^T$	$[.99503,.98959]^T$	4.860383E-5	2	32	469	6	56
	$[-10,1]^T$	$[.995118,.990449]^T$	2.740006E-5	3	129	1963	8	218
	$[10,1]^T$	$[.999098,.997999]^T$	4.7297846E-6	8	162	2257	18	276
	$[1,500]^T$	$[.999723,.999381]^T$	5.033286E-7	2	23	396	6	32
F_4/E-3/ E-3	$[3,0]^T$	$[-21.026652226,-36.76]^T$	1,449898E-23	2	62	703	3	72
	$[0.2,0.2]^T$	$[-21.02665226,-36.760008]^T$	1.449898E-23	2	136	2278	3	202
	$[0.2,.001]^T$	$[-21.02665226,-36.760008]^T$	1.449898E-23	2	236	4335	3	202
F_3/E-11/ E-3	$[0.1,4]^T$	$[1.000063,4.98538]^T$	3.958439E-16	1	1	23	2	0
	$[2,2.1]^T$	$[.99954819,1.9632769]^T$	1.604656E-13	2	3	536	3	6
	$[2,2]^T$	$[0.12877,1.73573]^T$	6.7003489E-12	2	1	325	3	6
F_5/E-6/ E-3	$[2,2,2]^T$	$[1.000000,1.000000,1.000000]^T$	3.75655E-23	2	27	309	9	42
	$[2,3,-1]^T$	$[1.000346,1.000349,1.000693]^T$	2.39242 E-7	11	55	3196	33	94
	$[-1,0,5]^T$	$[.998749,.998751,.997494]^T$	3.134087E-6	6	47	1067	18	82
F_6/E-6/ E-1	$[100,100]^T$	$[1.7954,1.377859]^T$	0.1690426	3	2	38	6	10
	$[-1,-2]^T$	$[1.7954,1.377859]^T$	0.1690426	3	2	45	8	10
	$[0.9, 0.7]^T$	global minimum	$-\infty$	-	-	-	-	-

Table 3. The effect of changing the maximum number of DFP iterations.

Function / initial point X(0)/DSC Tol.	MNI	Final point X^*	Final function $f(X^*)$	N	T	N(F)	N(G)	N(FT)
$F_1/[3,5]^T$/ E-3	10	$[0.99899028,0.9972287]^T$	5.77022628E-5	8	86	1279	32	154
	15	$[1.00392176,1.00798349]^T$	1.6932383E-5	4	51	71:9	10	86
	50	$[1.00412054,1.00832435]^T$	1.74181640E-5	2	42	634	6	70
$F_1/[10,1]^T$/ E-2	20	$[0.99909852,0.99799994]^T$	4.729784E-6	8	171	2257	18	196
	30	$[.99776935,0.99534902]^T$	8.76492094E-6	7	214	3335	16	356
	50	$[1.0087048,1.0177666]^T$	8.36820761E-5	4	192	3118	10	292
$F_1/[1,-600]^T$/ E-3	70	$[0.99771307,0.99486982]^T$	3.67645113E-5	5	377	6385	12	534
	150	$[0.99994910,0.99990334]^T$	5.2205022E-9	3	370	6118	8	540

Table 4. The effect of changing the DSC tolerance.

Function/ initial point/ DFP Tol.	DSC Tol.	Final Point X^*	Final Function $f(X^*)$	N	T	N(F)	N(G)	N(FT)
$F_1/[3,5]^T/$ E-4	0.1	$[1.00412054, 1.008324]^T$	1.7418164E-5	2	26	397	6	46
	E-3	$[1.0039387, 1.0074227]^T$	3.762379E-5	2	42	634	6	70
	E-10	$[.9942885, .98881024]^T$	3.66502394E-5	2	182	3393	6	48
$F_6/[100,100]^T/$ E-3	0.1	$[1.7954, 1.377859]^T$	0.1690426	3	2	38	6	10
	0.01	$[1.7954, 1.377859]^T$	0.1690426	3	2	40	6	10
	E-3	$[1.7954, 1.377859]^T$	0.1690426	3	2	40	6	10
	E-7	$[1.7954, 1.377859]^T$	0.1690426	3	3	44	6	10

Table 5. Robustness of Unconstrained NLP methods.

functions / methods	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
Steepest descent	0.0	0.0	1.0	0.33	0.0	0.0	0.0	0.0
Fletcher-Reeves	0.5	0.5	1.0	0.0	0.0	0.0	0.0	0.5
Newton with minimization	1.0	1.0	0.67	0.0	0.0	0.0	0.0	0.5
Newton with fixed step size	1.0	0.0	0.0	0.5	0.0	0.0	0.0	0.5
Polak-Rebiere	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.5
Pearson-2	0.0	0.25	0.75	0.0	0.0	0.5	0.0	0.0
DFP	1.0	1.0	1.0	1.0	1.0	0.5	0.33	1.0
proposed method	1.0	0.5	0.8	0.5	1.0	1.0	0.5	0.5
Rosenbrock	1.0	1.0	0.33	1.0	1.0	1.0	0.0	1.0

Results concerning robustness for the four combined multiparameter adjustment methods are shown in table (1). The factor of the robustness is calculated as the ratio of the number of success in getting the optimal value to the total number of trials with different starting points.

The combined multiparameter adjustment method with DFP has the ability of solving all test functions. Also it reaches the optimal solution in less number of stages than other combined multiparameter adjustment methods. Thus the DFP algorithm enhances the multiparameter adjustment method and makes it superperforms other combined multiparameter adjustment methods.

EFFECT OF THE ALGORITHM'S PARAMETERS

The effects of changing the starting points, the DFP tolerance, and the DSC tolerance are also examined on the performance of the combined multiparameter adjustment method with DFP algorithm. The starting points are the most influenced factor which affects performance of the method. Great changes in the number of function calls; N (F), the number of gradient calls; N(G), the number of the t's function calls; N (FT), the number of stages; N, and the time; T are accompanied the changes in the starting points. A sample of the results to illustrate these changes is shown in table (2). It is clear from this table that the initial point has a great influence on the performance. Also it is important to notice that for f_3 if the starting point is $[2, 2]^T$ the algorithm converges to a saddle point, for F_6 if the starting point is $[0.9,0.7]^T$ the algorithm does not converge to the local minimum $[1.7954, 1.377859]^T$ but it tends to converge to the global minimum which satisfies the equation $x^2_2 = 1-0.25x^2_1$ and the function goes to $-\infty$. It is also important to mention that some starting points in functions f_2 , f_4, f_7 and f_8 will make the algorithm diverges. Such examples of the last point are;

$$x^{(0)} = [1,2,-2,3]^T \text{ for } f_2 , x(0) = [-5,-3]^T$$

for f_4, $x^{(0)} = [0.5,0.5]^T$ for f_7, and $x^{(0)} = [0.1,1,1]^T$ for f_8.

The effect of the maximum allowable number of iterations, MNI, in the DFP algorithm depends on the behaviour of the function of be minimized. This is illustrated in table (3).

As the maximum allowable number of DFP iterations increases the total number of stages decreases and the total number of function evaluation and the time within the early stages increase. This procedure helps the DFP algorithm in getting close to the minimum quickly, hence the improvement of increasing the maximum number of DFP iteration will be very small at the next stages. For this reason the time and the number of function evaluations will decrease for some functions, and some starting points. While for other starting points or other functions these factors will increase then decrease as the maximum number of DFP iterations increases. The improvement in the value of a function and the norm will become constant after certain number of iterations which means that any further iterations will be wast of time without improvement.

Decreasing the value of the DSC tolerance will lead to slower convergence and more number of function evaluations with negligible improvement in the value of the function. Sometimes this will lead to error due to division by small number. For some function like F_6, there is no effect due to changing of the DSC tolerance, this result is illustrated in table (4).

COMPARISON AMONG UNCONSTRAINED METHODS

The convergence of the combined multiparameter adjustment DFP algorithm and eight other unconstrained methods is tested by applying such methods to the eight tested functions. The first criterion comparison which is taken in this paper is the robustness. Results are shown in table (5) for the nine methods.

It is clear from table (5) that the combined multiparameter adjustment with DFp methods is categorized n the same level of superiority of DFP, Fetcher, and Rosenbrock algorithms.

The main disadvantage of Rosenbrock algorithm is that it needs lot of time to prepare the functions to be tested. Also it takes longer time for convergence than the DFP or the combined multiparameter adjustment with DFP algorithms. In mean while Rosenbrock algorithm does not need calculation of the gradient which reduces its complexity at each stage.

By comparing the DFP algorithm and the combined DFP with multiparameter adjustment algorithm form the number of function evaluations and the CPU time, we find that the combined multiparameter adjustment method decreases the number of major stages but more iterations are needed within each stage. Thus the number of derivative evaluations in the proposed algorithm is less than that of the DFP algorithm. While the number of function evaluations in the DFP algorithm is less than that in the proposed one. The CPU time of the proposed algorithm is less a little bit than that of the DFP algorithm for most of the testing functions. It is also worth at the end of this comparison to mention that the proposed algorithm has higher rate of robustness for f_6 and f_7, while the opposite is for f_2, f_3, f_4 and f_8.

CONCLUSIONS

A multiparameter adjustment method is proposed and combined with four different unconstrained algorithms. These algorithms are tested for eight different test functions. It is found that the DFP algorithm enhances the multiparameter adjustment algorithm and make it superperforms the other combined multiparameter adjustment algorithms. The effects of changing the maximum allowable number of iterations of DFP, the DSC tolerance and the initial points are also examined on the performance of the combined multiparameter adjustment method with DFP.

A comparison between other unconstrained NLP algorithms and the combined multiparameter adjustment method with DFP is also discussed. Results put the proposed method in the same category as DFP, fletcher, and Rosenbrock methods from the superiority point of view.

REFERENCES

[1] Curry, H.B., " The method of steepest descent for nonlinear minimization problems", *Quarterly of Applied Mathematics*, Vol.2, pp 258-261, 1944.

[2] Marquardt, D.W., "Solution of nonlinear chemical engineering models", *Chemical Engineering Progress*, Vol. 55, pp 65-70, 1959.

[3] Mangasarian, O.L., "Techniques of Optimization", *Journal of Engineering for industry*, vol. 94,pp 365-372, 1972.

[4] FletBher, R. and Reeves, C.M., "Function minimization by conjugate gradients", *Computer Journal*, vol. 7, pp 149-154, 1964.

[5] Beckman, F.S., "The solution of linear equations by the conjug ate gradient method", *mathematical methods for digital computers*, vol.1, John Wiley and Sons, New York, 1960.

[6] Dembo, R., Eisenstat, S.C. and Steihaug, T., "Inexact Newton methods", *SIAM Journal on Numerical Analysis*, Vol. 19, pp 400-408, 1982.

[7] Dixon, L.C.W. and Price, R.C., "Numerical experience with the truncated Newton method", *Numerical Optimisation center, Hatfield Polytechnic*, Technical report No. 169, 1986.

[8] Nazareth, J.L., "Analogues of Dixon and powell's theorems for unconstrained minimization with inexact line searches", *SIAM Journal on Numerical Analysis*, Vol.23,pp 170-177, 1986.

[9] Dixon, L.C.W. and Price, R.C., "Truncated Newton for sparse unconstrained optimization using automatic differentiation", *Journal of optimization theory and applications*, Vol. 60, No. 2, pp 261-275, 1989.

[10] Davidon, W.C. "Variable metric method for minimization",*Atomic Energy Commission*, Research and development report No. ANL-5990, 1959.

[11] Fletcher, R. and Powell, M.J.D., "A rapidly convergent descent method for minimization", *Computer Journal*, Vol. 6, pp 163-168, 1963.

[12] Flectcher, R. and Sinclair, J.W., "Degenerate values for Broyden methods", *Journal of Optimization theory and applications*, Vol. 33, No.3, pp 311-324, 1981.

[13] Groewank, A. and P.H.L. Toint, "Partitioned variable metric updates for large structured optimization problems", *Numerical Mathematics*, vol.39, pp 119-137, 1982.

[14] Rosenbrock, H.,"Automatic method for finding the greatest or least value of a function", *Computer journal*, vol. 3, pp 175-148, 1960.

[15] Hooke, R. and Jeeves, T.A., "Direct search solution of Numerical and statistical problems", *Journal of the Association for computing Machinery*, vol. 8, pp 212-229, 1961.

[16] Powell, M.J.D., "An efficient methods for finding the minimum of a function of several variables without calculating derivatives", *Computer Journal*, Vol. 7, pp 155-162, 1964.

[17] Nelder, J.A. and R. Mead, R., "A simplex method for function minimization", *Computer Journal*, vol. 7, pp 308-313, 1965.

APPENDIX: TEST FUNCTIONS

The test function used were as follows

1. Min $f_1(x) = 100 (x_2 - x_1^2)^2 + (1 - x_1^2)^2$

   $f_1(x^*) = 0$ at $x^* = [1 \quad 1]^T$

2. Min $f_2(x) = (x_1 + 10 x_2)^2 + 5 (x_3 - x_4)^2 + (x_2 - 2x_3)^4$

   $+ 10 (x_1 - x_4)^4$

   $f_2(x^*) = 0$ at $x^* = [0 \ 0 \ 0 \ 0]^T$

3. Min $f_3(x) = (x_1 x_2)^2 (1 - x_1)^2 (1 - x_1 - x_2 (1 - x_1)^5)^2$

   $f_3(x^*) = 0$ at $x^* = [1 \text{ unbounded}]^T$,

   $x^* = [0 \text{ unbounded}]^T$, or $x^* = [\text{unbounded } 0]^T$

4. Min $f_4(x) = (x_1^2 + 12x_2 - 1)^2 + (49x_1^2 + 49x_2^2 + 84x_1 +$

   $2324x_2 - 681)^2$

   $f_4(x^*) = 0$ at $x^* = [0.252784 \quad 0.280878]^T$ or

   $x^* = [-21.026653 \ - \ 36.76009]^T$

5. Min $f_5(x) = 100 (x_3 - (\frac{x_1 + x_2}{2})^2)^2 + (1 - x_1)^2 + (1 - x_2)^2$

   $f_5(x^*) = 0$ at $x^* = [1 \ 1 \ 1]^T$

6. Min $f_6(x) = (x_1 - 2)^2 (x_2 - 1)^2 + \frac{0.04}{-0.25x_1^2 - x_2^2 + 1} +$

   $5(x_1 - 2x_2 + 1)^2$

   $f_6(x^*) = 0.169043$ at $x^* = [1.7954 \quad 1.3778]^T$ (Local)

   $f_6(x^*) \rightarrow -\infty$ (Global)

   along the curve $x_2^2 = 1 - 0.25 x_1^2$

7. Max $f_7(x) = e^{-x_1 - x_2} (x_1^2 + 3x_2^2)$

   $f_7(x^*) = 1.1036$ at $x^* = [0 \ 1]^T$

8. Min $f_8(x) = 100 \{[x_3 - 10 \phi(x_1, x_2)]^2 +$

   $[\sqrt{x_1^2 + x_2^2 - 1}\,]^2_3 \} + x_3^2$

   where $\phi(x_1, x_2) = 1/2\pi \tan^{-1}(x_2/x_1)$

   for $x_1 > 0$

   $= 1/2 + 1/2\pi \tan^{-1}(x_2/x_1)$

   for $x_1 < 0$

   $f_8(X^*) = 0$ at $X^* = [1 \quad 0 \quad 0]^T$