Journal of Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.754738

ON THE SOLUTION OF SOME OPTIMIZATION PROBLEMS USING FREE DERIVATIVE BASED ALGORITHM

T. E. Olaosebikan[1*], K. J. Adebayo[2], A. O. Dele-Rotimi[3], O. J. Ayodele[4], O. E. Popoola[5], T. E. Akinwumi[6], F. O. Egbon[7],

[1,3,4,5,6,7]Department of Mathematical Sciences, Bamidele Olumilua University of Education, Science and Technology, Ikere-Ekiti, Nigeria. [2]Department of Mathematics, Ekiti State University, Ado-Ekiti, Nigeria.

(Email Address: [1*]olaosebikan.temitayo@bouesti.edu.ng, [2] kayode.adebayo@eksu.edu.ng, [3]dele-rotimi.adejoke@bouesti.edu.ng, [4]ayodele.oluwasola@bouesti.edu.ng, [5]popoola.odunayo@bouesti.edu.ng, [6]akinwumi.taiwo@bouesti.edu.ng and [7]egbon.friday@bouesti.edu.ng)

***Corresponding Author:** T. E. Olaosebikan

*Department of Mathematical Sciences, Bamidele Olumilua University of Education, Science and Technology, Ikere-Ekiti, Nigeria.

Email Address: olaosebikan.temitayo@bouesti.edu.ng,

ABSTRACT

If f is continuous and nonlinear on R^n, its application has proven evident throughout time. Minimizers are points where f is not differentiable. This work specifically focuses on the scenario when it is challenging to calculate the gradient and Hessian matrices for each given value of x. This study introduces a novel technique that utilizes derivatives to solve optimization problems. Specifically, it focuses on employing finite difference representations of the gradient and Hessian in the Quasi Newton method and Derivative Free Trust Region methods. If it is proven that f has a unique solution, it may be demonstrated that the step length (h) generated converges globally. Two test problems were employed for actual implementation utilizing MATLAB software. The numerical outcomes demonstrated the efficacy and resilience of the algorithms, which exhibited favourable comparisons to certain preexisting methods.

Keywords: Continuous functions, Differentiability, Quasi-Newton Method, Trust Region Method, Free Derivative, Optimization Problems,

1.0. Introduction

Derivative-based algorithms that are available at no cost are a highly efficient approach for tackling optimization problems [1][11]. This can be attributed to several variables, which include: Derivative-based algorithms that are available for free can produce solutions that are equally, if not more, accurate compared to conventional methods like gradient descent or Newton's Method [9][17][22] . These algorithms frequently necessitate fewer iterations and hence demand fewer computer resources compared to classic approaches that require numerous iterations to get optimal values. The versatility of these optimizations enables their application across different frameworks and problem sets, simplifying the process by offering consistent solution paths without requiring customization or adjustments specific to each model [15][19]. Unlike other numerical techniques that rely on iterative assessment of known points using ODE solvers, free derivatives only compute precise values at specific locations during each iteration cycle [18][20]. This approach greatly reduces computing time compared to convolutional neural networks or deep learning applications, which often require extensive backtracking through samples to find the true global optimum value(s)[5].

Conventionally, optimization is the process of finding the most effective strategies to meet specific objectives while adhering to predetermined constraints [14][17]. Recent developments in digital computer technology, applied mathematics and operations research, have led to the emergence of intricate industrial challenges in fields like engineering, management, and economics [11][12]. Consequently, there is a continuous development of novel and more potent optimization techniques to address these problems.

The field of optimization has significantly contributed to various disciplines such as science, social sciences, and management. This contribution is primarily shown through the creation of algorithms that effectively address real-world problems [8][10]. The origins of optimization can be attributed to the work of Newton, Lagrange, and Cauchy, who pioneered differential methods for the purpose of optimization [21]. The birth of the calculus of variations is credited to Bernoulli, Euler, Lagrange, and Weierstrass [6][11]. Although numerous contributions were made, significant progress was not achieved until the 20th century, when the introduction of computer technology facilitated the execution of optimization processes, leading to multiple advances thereafter [12][13].

2.0 QUASI-NEWTON ALGORITHM

Min $\{ f^0 (z) | Z \in R^n\}$, for $f^0(.)$ twice differentiable

Step 1: Select $a\ Z_0 \in R^n, select\ a,\ \beta \varepsilon\ (0.5, 0.8), and\ set\ i = 0$

Step 2: Compute $\nabla f^0(z_i)$

Step 3: If $\nabla f^0(z_i) = 0$, stop; else go to step 4

Step 4: Compute $\mathrm{H}(z_i) = \frac{\partial f^2 (z_i)}{\partial z^2}$

Step 5: If $\mathrm{H}(z_i)^{-1}$ exists, compute $\mathrm{h}(z_i)$ by solving $\mathrm{H}(z_i)\,\mathrm{h}(z_i) = -\nabla f^0 (z_i)$ and go to step 6; else set $\mathrm{h}(z_i) = -\nabla f^0 (z_i)$ and goto step 6.

It is observed that, when the Hessian matrices, $\mathrm{H}(z_i)$, are singular, then, Quasi-Newton algorithm is not applicable; and when is not positive definite, the search direction may fail to be a descent direction. Consequently, the problem will have no solution. Henceforth, gradient and Hessian methods in R^n must be employed for computing the step size and continuing with the following process.

Step 6: Set $\lambda = 1$

Step 7: Compute $\Delta = \mathrm{f}^0\big(z_i + \lambda \mathrm{h}(z_i)\big) - \mathrm{f}^0 (z_i) - \frac{\lambda}{3} < \nabla \mathrm{f}^0(z_i),\ \mathrm{h}(z_i) >$

Step 8: If $\Delta \leq 0, set\ \lambda i = \lambda,$ and go to step 9, else, set $\lambda i = \lambda\beta$ and go to step 7

Step 9: Set $Z_i + 1 = Z_i + \lambda_i h(Z_i)$, set $i = i + 1$, and goto step 2

Above algorithm work with derivative from step 2 and 4 respectively.

3.0 TRUST-REGION ALGORITHM

step 0: given $\Delta > 0$. Initialize the Trust-Region Method size to Δ_0 $(0, \Delta)$, and $\eta \in [0, ¼)$.Set $k = 0$
step 1: Approximately solves the trust-region problem to obtain P_k
step 2: Evaluate Q_k
step 3: if $Q_k < ¼$, $\Delta_k + 1 = ¼\, II\ Pk\ II$; $if\ Q > ¾\ and\ II\ Pk\ II = \Delta_k, \Delta_k + 1 = min(2\Delta_k, \Delta)$; $else\ \Delta_k = \Delta_k + 1$
step 4: $Q_k > \eta$, $x_{k+1}^5 = x_k^5 + p_k$, else $x_{k+1}^5 = x_k^5$. Set $k + 1$. go to step 1 [3][16]

4.0 NUMERICAL RESULTS

4.1 Problem P1
$Min\ f(x, y) = 100(y - x^2)^2 + (1 - x)^2$ [2]
Starting points: $z_0 = (-1.2, 1.0)$, $h = 0.0001$

Table 4.1: Table of result for P1 using Quasi-Newton method

N	z_n	h	$g(z_n)$	$h(z_n)$
0	$(-1.2000, 1.0000)$	0.0001	$(-215.6000, -88.000)$	$(0.0248, 0.3806)$
1	$(-1.1752, 1.3806)$	0.0001	$(-4.6375, -0.1221)$	$(1.8681, -4.3903)$
2	$(0.6929, -3.0097)$	0.0001	$(996.6181, -697.9610)$	$(0.0005, 3.4905)$
3	$(0.6934, 0.4808)$	0.0001	$(-0.6132, -4.105 \times 10^{-6})$	$(0.3145, 0.4361)$
4	$(1.0078, 0.9169)$	0.0001	$(39.8698, -19.7719)$	$(-0.0003, 0.0982)$
5	$(1.0075, 1.0151)$	0.0001	$(0.0151, -2.857 \times 10^{-5})$	$(-0.0078, -0.0157)$
6	$(0.9997, 0.9994)$	0.0001	$(0.0240, -0.0123)$	$(0.02900, 0.6413)$
7	$(1.0000, 1.0000)$	0.0001	$(3.8043, -1.1597) \times 10^{-5}$	$(-0.0770, -0.5747)$
8	$(1.0000, 1.0000)$	0.0001	$(-0.2523, -0.1505)$	$(0.2870, 0.5747)$
9	$(1.0000, 1.0000)$	0.0001	$(0.1038, 0.0515)$	$(-0.1072, -0.2148)$
10	$(1.0000, 1.0000)$	0.0001	$(-0.3865, -0.1931)$	$(0.4008, 0.8025)$

The solution converges at iteration 7 which is the minimum,
$Z^* = (1.0000, 1.00000)$

Table 4.2: Table of result for P1 using Trust Region method

N	z_n	h	$g(z_n)$	$h(z_n)$
0	$(-1.2000, 1.0000)$	0.0001	$(-215.600, -88.00)$	$(0.0248, 0.3806)$
1	$(-1.1752, 1.3806)$	0.0001	$(-4.6374, -0.1221)$	$(1.8607, -4.3727)$
2	$(0.6854, -2.9924)$	0.0001	$(948.4878, -692.3707)$	$(0.0005, 3.4625)$
3	$(0.6859, 0.4705)$	0.0001	$(-0.6281, -3 \times 10^{-8})$	$(0.3229, 0.4430)$
4	$(1.0088, 0.9135)$	0.0001	$(42.0876, -20.8505)$	$(-0.0004, 0.1035)$
5	$(1.0085, 1.0170)$	0.0001	$(0.0170, 0.0000)$	$(-0.088, -0.0178)$
6	$(0.9996, 0.9992)$	0.0001	$(0.0309, -0.158)$	$(0.3662, 0.8114)$
7	$(1.0000, 1.0000)$	0.0001	$(0.5828, -0.1950)$	$(-0.1004, -0.1998)$
8	$(1.0000, 1.0000)$	0.0001	$(-0.3590, -0.2209)$	$(0.4171, 0.8354)$
9	$(1.0000, 1.0000)$	0.0001	$(0.1678, 0.0831)$	$(-0.1739, -0.3483)$
10	$(1.0000, 1.0000)$	0.0001	$(-0.6965, -0.3479)$	$(0.0725, 0.1452)$

P1 converges at the 7th iteration, to the minimum, $Z* = (1.0000, 1.00000)$

4.2 Problem P2

(2)$Min\ f(x,y) = x^2 - 4x + y^2 - y - xy$ [4]

Using starting points: $z_0 = (1.2, 0.8)$, $h = 0.0001$

Table 4.3: Table of result for P2 using Quasi-Newton method

N	z_n	h	$g(z_n)$	$h(z_n)$
0	$(1.2, 0.8)$	0.0001	$(-2.4000, -0.6000)$	$(1.8000, 1.2000)$
1	$(3.0, 2.0)$	0.0001	$(0.1744, 0.1102)$	$(-0.1530, -0.1316)$
2	$(3.0, 2.0)$	0.0001	$(0.0000, -0.4441)$	$(0.1480, 0.2961)$
3	$(3.0, 2.0)$	0.0001	$(0.8882, 0.0000)$	$(-0.5921, -0.2961)$
4	$(3.0, 2.0)$	0.0001	$(0.0000, 0.4441)$	$(-0.1480, -0.2961)$
5	$(3.0, 2.0)$	0.0001	$(-0.8882, 0.0000)$	$(0.5921, 0.2961)$
6	$(3.0, 2.0)$	0.0001	$(0.0000, -0.4441)$	$(0.1480, 0.2961)$

Table 4.3 shows that P2 converged at the 3rd iteration, minimum, $Z^* = (3, 2)$

Table 4.4: Table of result for P2 using Trust Region method

N	z_n	h	$g(z_n)$	$h(z_n)$
0	$(1.2, 0.8)$	0.0001	$(-2.40001, -0.60002)$	$(1.80001, 1.200)$

1	$(3.0, 2.0)$	0.0001	$(-2.40001, -0.60002)$	$(1.80001, 1.2000)$
2	$(1.2, 0.8)$	0.0001	$(-2.40001, -0.60002)$	$(1.8001, 1.2004)$
3	$(3.0, 2.0)$	0.0001	$(-2.40001, -0.60002)$	$(1.8001, 1.20004)$
4	$(3.0, 2.0)$	0.0001	$(-2.40001, -0.60002)$	$(1.8001, 1.20004)$
5	$(3.0, 2.0)$	0.0001	$(-2.40001, -0.60002)$	$(1.8001, 1.20004)$
6	(3.0, 2.0)	0.0001	(-2.400001, -0.600002)	(1.8001, 1.20004)

P2 converged after 3rd iteration, with minimum, $Z^* = (3, 2)o$

Therefore, it's obvious that the Quasi-Newton method is faster, even though they converge at the same number of iteration because Quasi Newton method is cheaper than that of Trust-Region method. However, to compare their performance we must consider both cost and speed of convergence. An algorithm that converges quickly but takes a few seconds per iteration may take far more time than an algorithm which converges more slowly but only requires a few milliseconds per iteration. Quasi-Newton method is expected to be more efficient when the derivatives of the function are not available.

5.0 CONCLUSION

This paper aimed to research the domain based and derivative free optimization algorithms in an effort to identify their range of potential applications with respect to specific problem. We have investigated the range of problems which can be solved through certain algorithms. This paper has reviewed Quasi Newton and Trust Region techniques as two commonly used solutions, while specifically focusing on issues involving only two variables. Further research is needed to assess if these methods would prove still efficient when dealing with a larger number of parameters.

6.0 REFERENCE

1. A.R. Conn and N. Luis Vicente, (2009); Introduction to derivation free optimization. Siam, Philadelpha.
2. A.R. Conn, K. Scheinberg, P. l. Toint (1998); On the convergence of derivation-free method for unconstrained optimization, An Approximation theory and optimization tribute to Powell M.J.D.'' Iserles A; Buhmanu M. (pp.83-108) Cambridge university press.
3. A.R. Conn, N. I. M. Gould, and PH. L. Toint, (2000): Trust-Region Methods, MPSSIAM Series on Optimization, SIAM, Philadelphia.
4. A.R. Nelder and R. Mead, (1965), A simplex method for function Minimization, Comput.J.[vol.7, pp. 308–313].
5. C. LagariasJ, J. A. Reeds, M. H. Wright, and P. E. Wright(1998); Convergence properties of the Nelder–Mead simplex method in low dimensions, [9, pp.112–147] *SIAM J. Optim.*
6. C.T.Kelley, (1999), Iterative method for optimization, *SIAM J.*

7. Emerson Butyn, Elizabeth W. Karas and Welington de Oliveira (2022). A derivative-free trust-region algorithm with copula-based models for probability maximization problems. *European Journal of Operational Research*. 298(1), 59-75
8. G.J. Kolda, R.M. Lewis and V.J. Torczon (2003), Optimization by direct search: New perspectives on some classical and modern methods, SIAM Review 45, 385–482.
9. G. John, (1978); Revision of a Derivative-Free Quasi-Newton Method. Math. Comp. volume 32, number 141, pages 201-221.
10. H.H. Rosenbrock (1960); An automatic method finding the greatest or least value of function; *comput. J.,* vol.3, pp.175.
11. J. Nocedal and S.J. Wright, (1999): Numerical optimization, Operation Research, Springer Verlag, Newyork.
12. J. Nocedal andS.J. Wright (2014): Derivative-Free unconstrained optimization for theoretical stand point.
13. M.J.D. Powell (2008); Development of NEWUOA for Minimization without derivative IMA *Journal of numerical Analysis* 28 (pp. 649-664).
14. M.J.D. Powell (1964); An efficient method of finding the minimum of a function of several variables without calculating derivative; *comput. J.*, vol.7, pp. 155.
15. N. Pham, Bogdam M, Wilamowski (2011), Improved Nelder Mead's Simplex method and application, *Journal of computing vol. 3.*
16. N.I.M. Gould, D. Orban, A. Sartenaer and P.L. Toint, (2005); "Sensitively of the Trust-Region Algorithms to their parameters",40R: *A Quarterly Journal of operations Research* .vol.3, No. 3, pp.227-241.
17. Issam A.R. Moghrabi (2022). A non-Secant quasi-Newton Method for Unconstrained Nonlinear Optimization. Cogent Engineering. .https://doi.org/10.1080/23311916.2021.2018929
18. Ph.R. Sampaio and Ph.L. Toints, (2014): A Derivative-Free Trust-funnel method for Equality- constrained non-linear optimization.
19. S.C. Nash and A. Sofer, (1996) *Linear and Nonlinear Programming*, McGraw-Hill.
20. V.J. Torczon (1991); On the convergence of multidirectional search algorithms, *SIAM Journal on Optimization.*
21. X. Want (2003); Derivative –free optimization Algorithm, report, department of computing and software, MC Master University.
22. X. Wang, Dong Liang, Xingdong Feng and Lu Ye (2006); A derivative free optimization algorithm based on conditional moment. Department of Mathematics and Statistics, Yort University, Canada.
23. W. Huyer and A. Neumaier (1999), Global Optimization by Multilevel Coordinate Search. *Journal of Global Optimization* 14, 331–355.