Global Scientific JOURNALS

GSJ: Volume 6, Issue 2, February 2018, Online: ISSN 2320-9186
www.globalscientificjournal.com

A THEORETICAL SURVEY OF GRADIENT BASED ALGORITHMS FOR MULTIVARIABLE OPTIMIZATION

Agbedeyi O. D.[1] Adigwe R. U[2] and Nwanjei A.[3]
Delta State Polytechnic, Ogwashi-Uku, Delta State, Nigeria
1. Phone: 08134902214, Email: desmondagbedeyi@gmail.com
2. Phone: 08061349102, Email: hadigwe5@yahoo.com
3. Phone: 07068940280, Email: nwanjeiagiliga@yahoo.com

Abstract:
In this research, we took a theoretical survey of some Gradient Based Algorithms for Multivariate Optimization. Our objective was to make comparison and analyze the desirable characteristics of these methods. The following methods; Descent method, Cauchy's method, Newton's Method, Marguard's Method, Conjugate Gradient Method and Variable-metric Method (DFP) were surveyed. We found out among other things that gradient methods have certain advantage over methods that are none gradient based. In particular, Direct search methods require many function evaluations to converge to the minimum point but Gradient Methods exploit the derivative information of the function and are usually faster search methods.

Keywords: Multivariable Optimization, Cauchy's Method, Newton's Method, gradient based search

1. INTRODUCTION:

In this research, we present and survey theoretically algorithms for optimizing functions having multiple design or decision variables. The algorithms are presented for minimization problems but can also be used for maximization problem by using the duality principle. The algorithms for multivariable optimization can be classified into Direct Search and Gradient-Based methods but the focus of this research is on Gradient- Based Algorithms.

In the quest to address the problems associate with classical deterministic algorithms, several powerful optimization algorithms have been developed. Thus, a user facing a challenging optimization problem meets the task of determining which algorithm is appropriate for a given problem. An inappropriate approach may lead to a large waste of resources, both from the view of wasted efforts in implementation and from the view of the resulting suboptimal solution to the optimization problem of interest.

Hence, there is a need for objective analysis of the relative merits and shortcomings of the various algorithms for optimization problems. This need has certainly been recognized by others, as illustrated in recent conferences, where numerous sessions are devoted to comparing algorithms. Nevertheless, virtually all comparisons have been numerical tests on specific problems. Although sometimes enlightening, such comparisons are severely limited in the general insight they provide. Some comparisons for Several Stochastic Optimization Approaches are given in James et al (1999) and Arnold (2002).

Our objective is to make comparison and analyze the desirable characteristics of Gradient-based methods for multivariable optimization. We will consider six basic algorithms — Descent method, Cauchy's method, Newton's Method, Marguard's Method, Conjugate Gradient Method and Variable-metric Method (DFP).

2. GENERAL ALGORITHM FOR SMOOTH FUNCTIONS

All algorithms for unconstrained gradient-based optimization can be described as follows. We start with iteration number $k = 0$ and a starting point, $x^{(k)}$.

1. **Test for convergence**. If the conditions for convergence are satisfied, then we can stop and $x^{(k)}$ is the solution.
2. **Compute a search direction**. Compute the vector $d^{(k)}$ that defines the direction in n-space along which we will search.
3. **Compute the step length**. Find a positive scalar, $\alpha^{(k)}$ such that $f(x^{(k)} + \alpha^{(k)} d^{(k)}) < f(x^{(k)})$.
4. **Update the design variables**. Set $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$, $\mathrm{k} = \mathrm{k} + 1$ and go back to 1.

$$x^{(k+1)} = x^{(k)} + \underbrace{\alpha^{(k)} d^{(k)}}_{\Delta x^{(k)}} \qquad (1)$$

There are two sub-problems in this type of algorithm for each major iteration: computing the search direction $d^{(k)}$ and finding the step size (controlled by $\alpha^{(k)}$). The difference between the various types of gradient-based algorithms is the method that is used for computing the search direction.

2.1 Optimality Conditions

Consider a function $f(x)$ where x is the n-vector $x = [x_1, x_2, ...x_n]^T$.

The gradient vector of this function is given by the partial derivatives with respect to each of the independent variables,

$$\nabla f(x) \equiv g(x) \equiv \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \tag{2}$$

Therefore, the second-order partial derivatives can be represented by a square symmetric matrix called the Hessian matrix,

$$\nabla^2 f(x) \equiv H(x) \equiv \begin{bmatrix} \frac{\partial^2 f}{\partial^2 x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial^2 x_n} \end{bmatrix} \tag{3}$$

which contains $n(n+1)/2$ independent elements.

If f is quadratic, the Hessian of f is constant, and the function can be expressed as;

$$f(x) = \frac{1}{2} x^T H x + g^T x + \alpha$$

i. **Necessary conditions** (for a local minimum):

$$\|g(x^*)\| = 0 \text{ and } H(x^*) \text{ is positive semi-definite.} \tag{4}$$

ii. **Sufficient conditions** (for a strong local minimum):

$$\|g(x^*)\| = 0 \text{ and } H(x^*) \text{ is positive definite.} \tag{5}$$

2.2 Descent Direction.

A Descent Direction is a search direction $d^{(t)}$ at the point $x^{(t)}$ for which the condition $\nabla f(x^{(t)}) \cdot d^{(t)} \leq 0$ is satisfied in the neighborhood of $x^{(t)}$. The condition for descent search direction can be proved by comparing function values at two point along any descent directions. See Kalyanmoy (2012) for the proof.

How descent a search direction $d^{(t)}$ is, is a function of the magnitude of the vector $\nabla f(x^{(t)})\cdot d^{(t)}$. Thus, the most negative value of $\nabla f(x^{(t)})$ is the most preferred gradient for the search direction $d^{(t)}$. Thus, $-\nabla f(x^{(t)})$ is called the steepest descent direction.

For any linear function, it is worthy of note that the steepest descent direction at any point may not exactly pass through the true minimum point. But, the steepest descent direction is a direction which is a local best minimum direction. Also it is not guaranteed that moving along the steepest descent direction will always take the search closer to the true minimum point.

Gradient based methods work by searching along several directions iteratively. The algorithms vary according to the way the search directions are defined. To perform a search for each search direction $s^{(k)}$, we perform an undirected search to locate the minimum as follow;

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)}$$

where $\alpha^{(k)}$ is the step length. Since, $x^{(k)}$ and $s^{(k)}$ are known, the point $x^{(k+1)}$ can be expressed with only one variable $\alpha^{(k)}$. Thus, a unidirectional search on $\alpha^{(k)}$ can be performed and the new point $x^{(k+1)}$ can be obtained. Thereafter, the search is continued from the new point along another search direction $s^{(k+1)}$. This process continues until the search converges to a local minimum point. If a gradient-based search method is used for unidirectional search, the search can be terminated using the following procedure. By differentiating the expression $f(x^{(k+1)}) = f(x^{(k)} + \alpha s^{(k)})$ with respect to α and satisfying the optimality criterion, it can be shown that the minimum of the unidirectional search occurs when

$$\nabla f(x^{(k+1)}).s^{(k)} = 0.$$

the above criterion can be used to check the termination of the unidirectional search method.

3. CAUCHY'S (STEEPEST DESCENT) METHOD

This method uses the negative of the gradient as a search direction at any point $x^{(t)}$. That is,

$$s^{(k)} = -\nabla f(x^{(k)}).$$

Since the gives maximum descent in function values, it is also known as the steepest descent method. At every iteration, the derivative is computed at the current point and a unidirectional search is performed in the negative to the derivative direction to find the minimum point along that direction. The minimum point become the current point and the search is continued from this point. the algorithm continues until a point having a small enough gradient vector is found. This algorithm guarantees improvement in the function value at every iteration.

3.1 Steepest descent algorithm:

1. Select starting point $x^{(0)}$, a maximum number of iteration to be performed M, two terminal parameters ε_1, ε_2, and set $k = 0$.
2. Compute $\nabla f(x^{(k)})$, the first derivative at the point $x^{(k)}$.
3. If $\|\nabla f(x^{(k)})\| \le \varepsilon_1$, **Terminate;**

   Else, if $k \ge M$; **Terminate;**

   Else go to step 4.
4. Perform a unidirectional search to find $\alpha^{(k)}$ using ε_2 such that $f(x^{(k+1)}) = f[(x^{(k)}) - \alpha^{(k)}\nabla f(x^{(k)})]$ is minimum. One criterion for termination is when $|\nabla f(x^{(k+1)}).\nabla f(x^{(k)})| \le \varepsilon_2$.
5. If $\dfrac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} \le \varepsilon_1$, **Terminate;**

   Else, set $k = k+1$ and go to step 2.

Since the direction $s^{(k)} = -\nabla f(x^{(k)})$ is a descent direction, the function value $f(x^{(k+1)})$ is always smaller than $f(x^{(k)})$ for positive values of $\alpha^{(k)}$. This method works well when $x^{(k)}$ is far away for x^*. When the current point is very close to the minimum, the change in the gradient vector is small. Thus, the new point created by the unidirectional search is also close to the current point. this slows the convergence process near the true minimum. Convergence can be made faster by using the second-order derivative which is discussed in Newton's Method.

4. NEWTON'S METHOD

Newton's method uses second-order derivatives to create search directions. This allows faster convergence to the minimum point. considering the first three terms in Taylor's series expansion of the multivariable function, it can be shown that the first-order optimality condition will be satisfied if a search direction $s^{(k)} = [\nabla^2 f(x^{(k)}]^{-1} \nabla f(x^{(k)})$ is used. It can be shown that if $[\nabla^2 f(x^{(k)})]^{-1}$ is positive semidefinite, the direction $s^{(k)}$ must be descent. But if the matrix $[\nabla^2 f(x^{(k)})]^{-1}$ is not positive-semidefinite, the direction $s^{(k)}$ may or may not be descent, depending on whether the quantity $\nabla f(x^{(k)})^T [\nabla^2 f(x^{(k)})]^{-1} \nabla f(x^{(k)})$ is positive or not. Thus, the above search direction may not always guarantee a decrease in the function value in the neighborhood of the current point. But the second-order optimality condition suggests that $\nabla^2 f(x^{(*)})$ be positive-definite for the minimum point. Thus, it can be assumed that the matrix $\nabla^2 f(x^{(*)})$ is positive-definite in the neighborhood of the minimum point and the above search direction becomes descent near the minimum point. Thus, this method is suitable and efficient when the initial point is close to the optimum point. Since the function value is not guaranteed to reduce at every iteration, occasional restart of the algorithm from a different point is often necessary.

Using this method, demands some knowledge of the optimum point in the search space. Moreover, the computation of the Hessian matrix and its inverse are also computationally expensive.

4.1 Newton's Algorithm.

1. Select starting point $x^{(0)}$, a maximum number of iteration to be performed M, two terminal parameters ε_1, ε_2, and set $k = 0$.

2. Compute $\nabla f(x^{(k)})$, the first derivative at the point $x^{(k)}$.

3. If $\left\|\nabla f(x^{(k)})\right\| \le \varepsilon_1$, **Terminate;**

   Else, if $k \ge M$; **Terminate;**

   Else go to step 4.

4. Perform a unidirectional search to find $\alpha^{(k)}$ using ε_2 such that $f(x^{(k+1)}) = f[x^{(k)} - \alpha^{(k)}\nabla^2 f(x^{(k)})\nabla f(x^{(k)})]^{-1}$ is minimum. One criterion for termination is when $\left|\nabla f(x^{(k+1)}).\nabla f(x^{(k)})\right| \le \varepsilon_2$.

5. If $\dfrac{\left\|x^{(k+1)} - x^{(k)}\right\|}{\left\|x^{(k)}\right\|} \le \varepsilon_1$, **Terminate;**

   Else, set $k = k+1$ and go to step 2.

5. MARGUARDT'S METHOD

Cauchy's method works well when the initial point is far away from the minimum point and Newton's method works well when the initial point is near the minimum point. In any given problem, it is usually not known whether the chosen initial point is away from the minimum or close to the minimum, but wherever be the minimum point, a method can be devised to take advantage of both these methods. In Marguardt's method, Cauchy's method is initially followed. Thereafter, Newton's method is adopted. The transition from Cauchy's method to Newton's method is adaptive and depends on the history of the obtained intermediate solutions, as outline in the following algorithm.

5.1 Marguardt's Method Algorithm.

1. Select starting point $x^{(0)}$, a maximum number of iteration to be performed M, a terminal parameters ε. set $k = 0.$ and $\lambda^{(0)} = 10^4$ (a large number)
2. Compute $\nabla f(x^{(k)})$, the first derivative at the point $x^{(k)}$.
3. If $\left\|\nabla f(x^{(k)})\right\| \leq \varepsilon$, **Terminate;**

   Else, if $k \geq M$; **Terminate;**

   Else go to step 4.
4. Calculate $s(x^{(k)}) = -[H^{(k)} + \lambda^{(k)} I]^{-1} \nabla f(x^{(k)}).$ Set $x^{(k+1)} = x^{(k)} + s(x^{(k)}).$
5. If $f(x^{(k+1)}) < f(x^{(k)})$, go to step 6. Else to step 7.
6. Set $\lambda^{(k+1)} = \frac{1}{2}\lambda^{(k)}$, Set $k = k+1$ and go to step 2.
7. Set $\lambda^{(k+1)} = 2\lambda^{(k)}$ and go to step 4.

A large value of the parameter λ is used initially. Thus, the Hessian matrix has little effect on the determination of the search direction (see step 4). Initially, the search is similar to that in Cauchy's method. After a number of iterations (this is when the current solution has converged to the minimum), the value of λ becomes small and the effect is more like that in Newton's method. The algorithm can be made faster by performing a unidirectional search while finding the new point in step 4: $x^{(k+1)} = x^{(k)} + \alpha^{(k)} s(x^{(k)}).$ Since the computation of the Hessian matrix and its inverse are computationally expensive, the unidirectional search along $s^{(k)}$ is not usually performed. For simpler objective functions, however, a unidirectional search in Step 4 can be achieved to find the new point $x^{(k+1)}$.

6. CONJUGATE GRADIENT METHOD.

The conjugate gradient method is similar to the conjugate direction method. See Kalyanmoy (2012). Assuming the objective function is quadratic (which is a valid assumption at the vicinity of the minimum for many functions), conjugate search direction can be found using only the first-order derivatives. The details of the calculation procedure, see Reklaitis et al. (1983) or Rao (1984).

Fletcher and Reeves (1964) suggested the following conjugate search direction and proved that $s^{(k)}$ is a conjugate to all previous search directions $s^{(j)}$ for $j = 1,2,\cdots,(k-1)$:

$$s^{(k)} = -f(x^{(k)}) + \frac{\left\|\nabla f(x^{(k)})\right\|^2}{\left\|\nabla f(x^{(k-1)})\right\|^2} s^{(k-1),} \tag{6}$$

with $s^{(0)} = -\nabla f(x^{(0)})$. Note that this recursive equation for search direction $s^{(k)}$ requires only first-order derivatives at two points $x^{(k)}$ and $x^{(k-1)}$. The initial search direction $s^{(0)}$ is assumed to be the steepest descent direction at the initial point. Thereafter, the subsequent search directions are found by using the above recursive equation.

6.1 Conjugate Gradient Method Algorithm.

1. Select starting point $x^{(0)}$ and terminal parameters $\varepsilon_1, \varepsilon_2, \varepsilon_3, .$
2. Compute $\nabla f(x^{(0)})$ and set $s^{(0)} = -\nabla f(x^{(0)})$.
3. Find $\lambda^{(0)}$ such that $f(x^{(0)} + \lambda^{(0)} s^{(0)})$ is minimum with termination parameter ε_1. Set $x^{(1)} = x^{(0)} + \lambda^{(0)*} s^{(0)})$ and $k = 1$ calculate $\nabla f(x^{(1)})$.
4. Set $s^{(k)} = -f(x^{(k)}) + \frac{\left\|\nabla f(x^{(k)})\right\|^2}{\left\|\nabla f(x^{(k-1)})\right\|^2} s^{(k-1),}$

5. Find $\lambda^{(k)}$ such that $f(x^{(k)}+\lambda^{(k)}s^{(k)})$ is minimum with termination parameter ε_1. Set $x^{(k+1)} = x^{(k)} + \lambda^{(k)*}s^{(k)})$

6. If $\dfrac{\left\|x^{(k+1)} - x^{(k)}\right\|}{\left\|x^{(k)}\right\|} \le \varepsilon_2$ or $\left\|\nabla f(x^{(k+1)})\right\| \le \varepsilon_3$, Terminate;

   Else set $k = k+1$ and go to Step 4.

For minimization of linear or quadratic objective functions, two iterations of this algorithm are sufficient to find the minimum. But, for other functions, more iterations through Step 4 to 6 may be necessary. It is observed that the search directions obtained using (6) become linearly dependent after a few iterations of the above algorithm. When this happens, the search process becomes slow. In order to make the search faster, the linear dependence of the search directions may be checked at every iteration. One way to compute the extent of linear dependence is to calculate the included angel between two consecutive search directions $s^{(k-1)}$ and $s^{(k)}$. If the included angle is close to zero (less than a small predefined angle), the algorithm is restarted from Step 1 with $x^{(0)}$ being the current point. A restart us usually necessary after every *N* search directions are created.

7. VARIABLE-METRIC METHOD (DFP METHOD)

We have discussed earlier that once the current solution is close to the minimum point of a function, Newton's method is very efficient. But one difficulty with that method is the computation of the inverse of the Hessian matrix. In variable-metric method, an estimate of the inverse of the Hessian matrix at the minimum point is obtained by iteratively using first-order derivatives. This eliminates expensive computation of the Hessian matrix and its inverse. We replace the inverse of the Hessian matrix by a matrix $A^{(k)}$ at iteration k and have a search direction

$$s^{(k)} = -A\nabla f(x^{(k)}). \qquad (7)$$

Starting with an identity matrix, $A^{(0)} = I$, we modify the matrix A at every iteration to finally take the shape of the inverse of the Hessian matrix at the minimum point. Davidon, Fletcher and Powell suggested a way of modifying the matrix A (Davidon (1959), Fletcher and Powell (1986)). Their method is known as the DFP method:

$$A^{(k)} = A^{(k-1)} + \frac{\Delta x^{(k-1)} \Delta x^{(k-1)^T}}{\Delta x^{(k-1)^T} \Delta e x^{(k-1)}} - \frac{A^{(k-1)} \Delta e(x^{(k-1)})(A^{(k-1)} \Delta e x^{(k-1)}))^T}{\Delta e(x^{(k-1)})^T A^{(k-1)} \Delta e x^{(k-1)})} \qquad (8)$$

In the equation above, $e(x^{(k)}) = \nabla f(x^{(k)})$ which is the gradient of the function at the point $x^{(k)}$. While the change in the design variable is denoted by $\Delta x^{(k-1)} = x^{(k)} - x^{(k-1)}$ and the change in gradient vector is denoted by $\Delta e(x^{(k-1)}) = e(x^{(k)}) - e(x^{(k-1)})$.

In the DFP method, the modification of the matrix, using equation (8) preserves the symmetry and the positive-definiteness of the matrix. This property makes the DFP method attractive. Recall that in order to achieve a descent direction, Newton's method requires that the Hessian matrix must be positive definite. Since an identity matrix is symmetric and positive-definite, at every iteration the positive-definiteness of the matrix $A^{(k)}$ is retained by the above transformation and the function value is guaranteed to decrease monotonically. Thus, a monotonic Improvement in function values in every iteration is expected with the DFP method.

7.1 DFP Algorithm

1. Select starting point $x^{(0)}$ and terminal parameters $\varepsilon_1, \varepsilon_2, \varepsilon_3,$.
2. Compute $\nabla f(x^{(0)})$ and set $s^{(0)} = -\nabla f(x^{(0)})$.
3. Find $\lambda^{(0)}$ such that $f(x^{(0)} + \lambda^{(0)} s^{(0)})$ is minimum with termination parameter ε_1. Set $x^{(1)} = x^{(0)} + \lambda^{(0)*} s^{(0)})$ and $k = 1$ calculate $\nabla f(x^{(1)})$.
4. Compute $A^{(k)} = A^{(k-1)} + \frac{\Delta x^{(k-1)} \Delta x^{(k-1)^T}}{\Delta x^{(k-1)^T} \Delta e x^{(k-1)}} - \frac{A^{(k-1)} \Delta e(x^{(k-1)})(A^{(k-1)} \Delta e x^{(k-1)}))^T}{\Delta e(x^{(k-1)})^T A^{(k-1)} \Delta e x^{(k-1)})}$

   and set $s^{(k)} = -A \nabla f(x^{(k)})$.

5. Find $\lambda^{(k)}$ such that $f(x^{(k)}+\lambda^{(k)}s^{(k)})$ is minimum with termination parameter ε_1. Set

   $x^{(k+1)}=(x^{(k)}+\lambda^{(k)*}s^{(k)})$

6. If $\dfrac{\left\|x^{(k+1)}-x^{(k)}\right\|}{\left\|x^{(k)}\right\|}\le\varepsilon_2$ or $\left\|\nabla f(x^{(k+1)})\right\|\le\varepsilon_3$, Terminate;

   Else set $k=k+1$ and go to Step 4.

8. SUMMARY AND CONCLUSION

Optimization algorithms have been discussed to solve multivariable functions. At first, optimality conditions have been presented. Thereafter, five different gradient-based search algorithms were discussed.

In multivariable function optimization, the first-order optimality condition requires that all components of the gradient vector be zero. Any point that satisfies this condition is a likely candidate for the minimum point. The second-order optimality condition for minimization requires the Hessian matrix to be positive-definite. Thus, a point is minimum if both the first and the second-order optimality conditions are satisfied.

All gradient-based methods work on the principle of generating new search direction iteratively and performing a unidirectional search along each direction. The steepest descent direction is a direction opposite to the gradient direction at any point. in Cauchy's method, the steepest descent direction is used as a search direction, thereby ensuring a monotonic reduction in function values in successive iteration. Cauchy's method works well for solutions far away from the true minimum point. The search become slow when the solution is closer to the minimum point. In Newton's method, the second-order derivative information (Hessian) is used to find a search direction. Even though this search direction is not guaranteed to be a descent direction for points close to the true minimum point, this method is very efficient. Marguardt's method uses a compromise of both these methods by adaptively blending the transition from Cauchy's method to Newton's method in creating search directions. The Fletcher-Reeves method creates search directions that are conjugate with respect to the objective function. The DFP method uses the concept of Newton's method, but the Hessian matrix is not directly used. Instead, the Hessian matrix at the minimum point is estimated by successively modifying an initial identity matrix using first-order derivatives.

9. REFERENCES

Arnold, D. V. (2002): *Noisy Optimization with Evolution Strategies*. Kluwer: Boston.

Davidon, W. C. (1959): Variable metric method for minimization (Technical Report No. ANL-599). AEC *Research Development Report.*

Fletcher, R. and Powell, M. J. D. (1963): A rapidly convergent descent method for minimization. *Computer Journal*, 6, 163-168.

Fletcher, R. Reeves C. M. (1964): Function minimization by conjugate gradient. *Computer Journal*, 7, 149-154.

Kalyanmo, D. (2012): *Optimization for Engineering Design*. New Delhi: PHI Learning Private Limited

Recklaitis, G. V., RAvindran, A., and Ragsdell, K. M. (1983): *Engineering Optimization-Methods and Applications*. New York: Wiley