A Comparative Analysis of Optimization Algorithms Using Griewank Benchmark Function

Yiğit Ateş
Dept. of Mathematical Engineering
Yildiz Technical University
Istanbul, Turkey
yigit.ates@std.yildiz.edu.tr

Assoc. Prof. Hale Köçken
Dept. of Mathematical Engineering
Yildiz Technical University
Istanbul, Turkey
hgonce@yildiz.edu.tr

Lect. PhD. Gökhan Göksu
Dept. of Mathematical Engineering
Yildiz Technical University
Istanbul, Turkey
gokhan.goksu@yildiz.edu.tr

***Abstract*—This study presents a comparative analysis of five optimization algorithms, namely Newton-Raphson, Hestenes-Stiefel, Polak–Ribière, Fletcher-Reeves, and Steepest Descent using the Griewank function as a benchmarking tool. The evaluation compares the optimal values achieved, execution time, and number of iterations required for convergence and highlights the importance of selected initial points. The results of this benchmarking study will provide valuable insights into the suitability of these algorithms for solving problems with characteristics similar to the Griewank function.**

Keywords—optimization methods, gradient, conjugate gradient, hessian, benchmark,, griewank, newton-raphson, hestenes-stiefel, polak-ribière, fletcher-reeves, steepest descent

I. Introduction

By today's standards, problems in many different domains, especially engineering, have grown remarkably complicated. Problems can no longer fit into simple quadratic or unimodal solutions. Instead, the objective functions exhibit a wide range of trends and behaviors, resembling intricate patterns rather than simple paths. Because of this intricacy, it is extremely difficult, if not impossible, to determine the optimal solutions. However, global optimization techniques offer an alternative for identifying and solving these issues. Through strategic utilization of samples and approximation methods, they navigate towards the global optimum in a stochastic manner [1].

This study aims to compare the effectiveness of three conjugate gradient methods (Hestenes-Stiefel, Polak-Ribière, and Fletcher-Reeves) and other gradient based methods (Newton-Raphson, Steepest Descent) on two dimensional Griewank (GW) function. By testing these methods on this function with the purpose of minimizing, the comparing metrics such as the optimal values they find, their execution time, and the total number of iterations required will be presented and used for an evaluation.

II. Objective Function

$$f_{min}(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (1)$$

Domain of the GW function is subject to $-600 \leq x_i \leq 600$ where $i \in \{1, 2, \ldots, n\}$ and n is the number of independent variables.

The number of local minima for an arbitrary n is unknown, although in this study where the dimension is $n = 2$, there are 500 local minima [1].

$$f_{min}(x) = 1 + \frac{1}{4000}(x_1^2 + x_2^2) - \cos(x_1) \cdot \cos\left(\frac{x_2}{\sqrt{2}}\right) \quad (2)$$

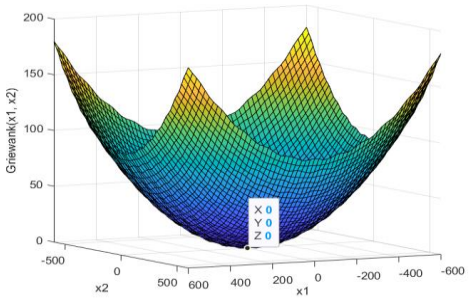

Figure 2.1: Surface plot of the GW function in domain

The global minima, denoted by a black point in Figure 2.1, is found at $x^* = (0, 0, \ldots, 0)$ where $f(x^*) = 0$.

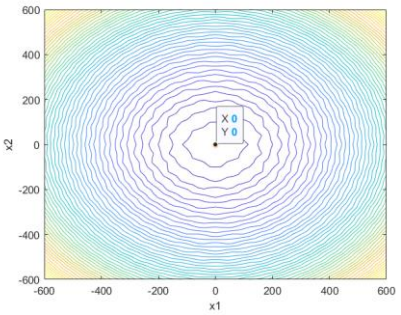

Figure 2.2: Contour of the GW function

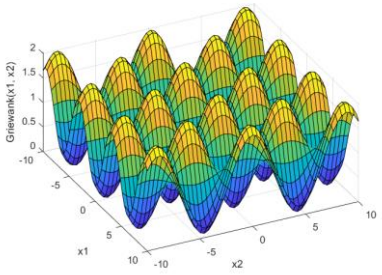

Figure 2.3: Close inspection of local minima

Two-dimensional GW function features a large number of regularly and widely distributed local minima (Figure 2.3) that exponentially increase as shown in Figure 2.2.

III. METHODOLOGY

The MATLAB script will execute the algorithms with the following three uniformly distributed initial starting points selected randomly within the domain interval of the GW function using the MATLAB function $randn()$.

- $x_0^1 = [82.5884; -36.7312]$
- $x_0^2 = [520.8128; -444.1125]$
- $x_0^3 = [-405.3812; 353.1414]$

This process will continue until the points satisfy the stopping criterion, with $\epsilon = 10^{-4}$:

$$\|\nabla f(x_i)\| \leq \epsilon \quad \text{and} \quad |\nabla f(x_i) - \nabla f(x_{i+1})| \leq \epsilon$$

A. Benchmarking Procedure

For benchmarking, each algorithm will be executed 200 times, initiated from 200 uniformly distributed distinct initial points randomly chosen within the domain interval. However, to maintain consistency, each algorithm is constrained to a maximum iteration limit of 100; if this threshold is surpassed, the execution will be considered complete. This approach aims to ensure a thorough evaluation of each algorithm's robustness and efficiency across diverse regions of the function's domain.

The source code for all algorithms and benchmark tests is available at this GitHub repository.

B. Hestenes-Stiefel Method

Choosing an initial starting point x_0
Set $d_0 = -\nabla f(x_0)$

while $\|\nabla f(x_i)\| \leq \epsilon$ && $|\nabla f(x_i) - \nabla f(x_{i+1})| \leq \epsilon$

$$\alpha_i = -\frac{\nabla f(x_i)^T \cdot d}{d^T \cdot H \cdot d} \quad (3)$$

$$x_{i+1} = x_i + \alpha \cdot d$$

$$\beta_i = \frac{\nabla f(x_{i+1})^T \cdot (\nabla f(x_{i+1}) - \nabla f(x_i))}{d^T \cdot (\nabla f(x_{i+1}) - \nabla f(x_i))} \quad (4)$$

$$d_{i+1} = -\nabla f(x_{i+1}) + \beta \cdot d$$

Update $i = i + 1$

end

at point x_i,

- x_i is the current point
- x_{i+1} is the updated approximation point
- $\nabla f(x_i)$ is the gradient of the function
- $\nabla^2 f(x_i)$ or H is the Hessian matrix of the function
- α_i is the step size
- d_i is the direction search vector
- β_i is a coefficient

C. Polak–Ribière Method

The Polak-Ribière method follows the fundamental structure of the conjugate gradient algorithm outlined in Hestenes-Stiefel. The key difference lies in the calculation of the β coefficient (Equation 4) used to update the direction vector.

$$\beta_i = \frac{\nabla f(x_{i+1})^T \cdot (\nabla f(x_{i+1}) - \nabla f(x_i))}{\nabla f(x_i)^T \cdot \nabla f(x_i)} \quad (5)$$

D. Fletcher-Reeves Method

Similar to the Polak-Ribière method, the Fletcher-Reeves approach inherits the foundational structure of the conjugate gradient algorithm, and it distinguishes itself by offering an alternative computation method for the β coefficient.

$$\beta_i = \frac{\nabla f(x_{i+1})^T \cdot \nabla f(x_{i+1})}{\nabla f(x_i)^T \cdot \nabla f(x_i)} \quad (6)$$

E. Steepest Descent

The Steepest Descent algorithm differs from conjugate gradient methods, notably in its approach to determining step size (comparing Equation 3 and Equation 7). Instead of utilizing a direction search vector it predominantly relies on the gradients of the objective function at the achieved points.

Choosing an initial starting point x_0

while $\|\nabla f(x_i)\| \leq \epsilon$ && $|\nabla f(x_i) - \nabla f(x_{i+1})| \leq \epsilon$

$$\alpha_i = \frac{\nabla f(x_i)^T \cdot \nabla f(x_i)}{\nabla f(x_i)^T \cdot H \cdot \nabla f(x_i)} \quad (7)$$

$$x_{i+1} = x_i - \alpha \cdot \nabla f(x_i)$$

Update $i = i + 1$

end

F. Newton-Rapshon Method

The Newton-Raphson method presents a distinct strategy for optimization in contrast to Steepest Descent and other conjugate gradient algorithms, as it integrates second-order information through the utilization of the Hessian matrix directly.

Choosing an initial starting point x_0

while $\|\nabla f(x_i)\| \leq \epsilon$ && $|\nabla f(x_i) - \nabla f(x_{i+1})| \leq \epsilon$

$$x_{i+1} = x_i - \nabla^2 f(x_i)^{-1} \cdot \nabla f(x_i)$$

Update $i = i + 1$

end

IV. RESULTS

A. *Convergence Patterns and Initial Point Sensitivity*

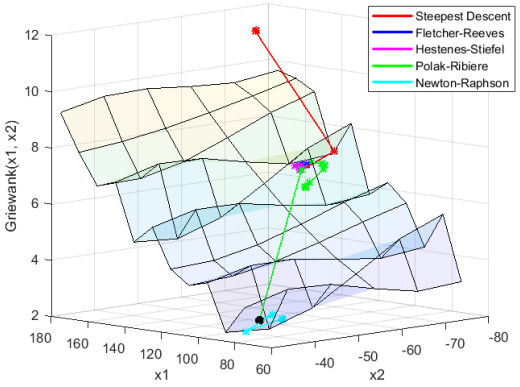

Figure 4.1: Paths of Algorithms on GW Surface from x_0^1

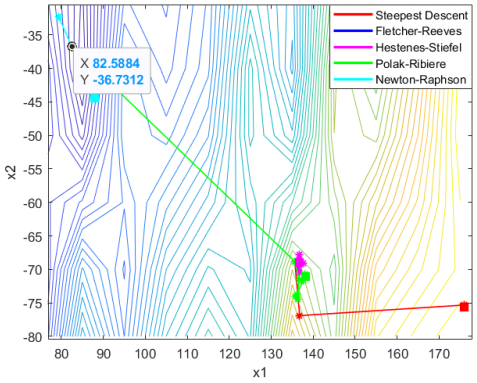

Figure 4.2: Paths of Algorithms on GW Contour from x_0^1

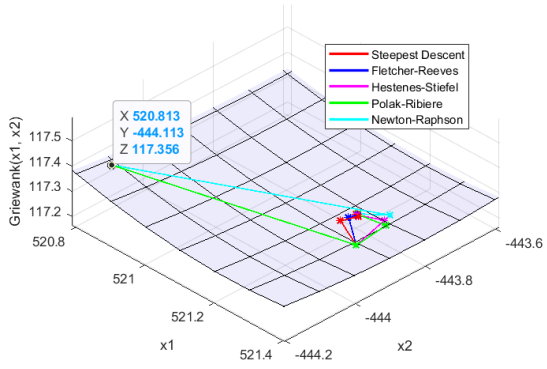

Figure 4.3: Paths of Algorithms on GW Surface from x_0^2

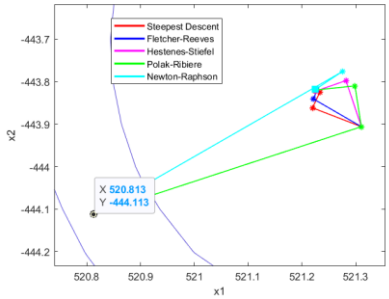

Figure 4.4: Paths of Algorithms on GW Contour from x_0^2

In the case of the GW function, which has 500 local minima in a two-dimensional space, the convergence behavior of optimization algorithms becomes even more complex. Despite starting from the same initial point x_0^1, the algorithms may converge to different optimal values. Notably, Polak-Ribière, Hestenes-Stiefel, and Fletcher-Reeves methods follow a similar or even identical initial path before converging in different trends due to the differences in the β coefficient calculation. These three methods tend to make a significant initial step, followed by smaller steps to converge to their optimal values. These behaviors are illustrated in Figure 4.1 and Figure 4.2, where the algorithms exhibit varying convergence paths with initial point. Noticeably, except for the Newton-Raphson method, the function values at the optimal points increase instead of decreasing. Remarkably, the Newton-Raphson method stands out as the only algorithm resulting in a decrease with the least amount of iteration and time. This unexpected behavior of the remaining 4 methods highlights the challenging nature of optimizing functions with numerous local minima, such as the GW function and will be investigated further in *B. Benchmarking* subchapter.

Table 4.1: Statistics for initial point x_0^1 where $f(x_0^1) = 2.6313$

Alg.	(x_1^*, x_2^*)	$f(x_1^*, x_2^*)$	n	*Time* (s)
H-S	(136.7081, -68.9615)	6.8578	11	0.6950
P-R	(138.1609, -71.0149)	6.0365	12	0.7682
F-R	(136.7081, -68.9615)	6.8578	10	0.6137
S-D	(176.0174, -75.6049)	11.1692	10	0.6940
N-R	(87.9206, -44.3844)	2.4264	6	0.3548

Figure 4.3, Figure 4.4, and Table 4.2 reveal that with the same initial point x_0^2, all algorithms may converge to nearly identical optimal points and values and minimize the objective function when the initial point is convenient. Unlike the previous initial point x_0^1, this convergence demonstrates a consistent minimization of the objective function across different optimization methods. Additionally, when algorithms find the optimal value during minimization, they tend to converge towards the nearest local minimum. Following this logic, to enhance the likelihood of identifying the global minimum, selecting initial points proximal to $(0,0)$ is highly recommended.

Table 4.2: Statistics for initial point x_0^2 where $f(x_0^2) = 117.3561$

Alg.	(x_1^*, x_2^*)	$f(x_1^*, x_2^*)$	n	*Time* (s)
H-S	(521.2250, -443.8180)	117.2539	5	0.2581
P-R	(521.2250, -443.8180)	117.2539	6	0.3350
F-R	(521.2251, -443.8180)	117.2539	6	0.3118
S-D	(521.2251, -443.8180)	117.2539	8	0.5105
N-R	(521.2250, -443.8180)	117.2539	4	0.1819

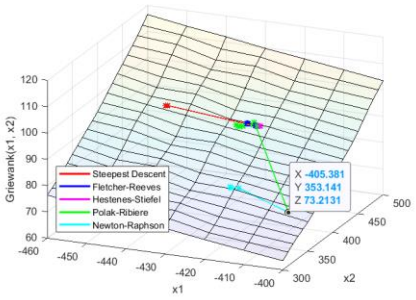

Figure 4.5: Paths of Algorithms on GW Surface from x_0^3

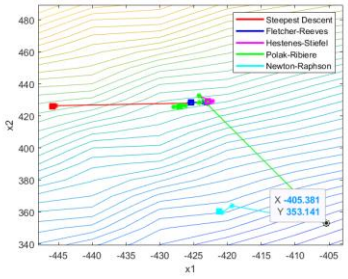

Figure 4.6: Paths of Algorithms on GW Contour from x_0^3

Conversely, starting from the initial point x_0^3, as shown in Figures 4.5 and 4.6, presents a contrasting scenario, where all the algorithms' performance falters, an unexpected increase in the initial function value is observed across all methods, diverging from the primary objective of function minimization. This underscores the crucial importance of initial point selection, particularly for optimizing sophisticated objective functions.

Table 4.3: Statistics for initial point x_0^3
where $f(x_0^3) = 73.2131$

Alg.	(x_1^*, x_2^*)	$f(x_1^*, x_2^*)$	n	*Time (s)*
H-S	(-422.8607, 429.0555)	91.6557	9	0.4846
P-R	(-427.0301, 426.0724)	91.0464	12	0.6563
F-R	(-425.3697, 428.4191)	92.1901	10	0.5396
S-D	(-445.8695, 426.0712)	95.1598	11	0.6858
N-R	(-421.1933, 360.2470)	78.7375	6	0.3005

B. Benchmarking

As the GW function is discussed to be a sophisticated objective function in the previous subchapter and it is not always certain that the methods used in this study can minimize the function, to evaluate the performance of each algorithm across the function's domain, each algorithm are executed 200 times with different uniformly distributed initial points for benchmarking. Performance metrics, such as the percentage of runs that achieve a smaller value than the initial value, iteration count, and elapsed time, are presented.

Table 4.4: Value comparison at initial and optimal points

Algorithms	*Number of*	
	$f(x^*) < f(x_0)$	$f(x^*) > f(x_0)$
Hestenes-Stiefel	103	97
Polak–Ribière	108	92
Fletcher-Reeves	109	91
Steepest Descent	93	107
Newton-Raphson	106	94

Table 4.4. shows that each method minimizes the objective function nearly 50% of the time, with Steepest Descent performing slightly below this benchmark. Since there is no significant difference in the minimization success rate among the methods, it is advisable to examine other statistics for a more comprehensive evaluation.

Table 4.5: Iteration counts of the algorithms

Algorithms	*Iterations*		
	Min	*Max*	*Mean*
Hestenes-Stiefel	4.0000	18.0000	8.1650
Polak–Ribière	3.0000	100.0000	9.9200
Fletcher-Reeves	4.0000	100.0000	26.6300
Steepest Descent	5.0000	37.0000	10.8200
Newton-Raphson	3.0000	9.0000	4.7200

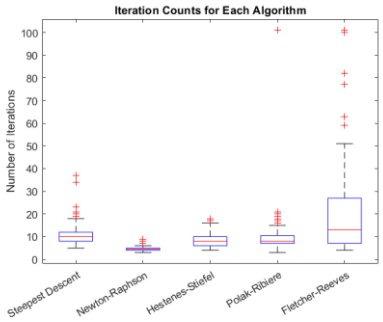

Figure 4.5: Iteration Count Boxplot of the Algorithms

In terms of iteration and execution time numbers, the Newton-Raphson method has the fewest attempts, the lowest mean, and the lowest median iteration numbers, as well as the lowest execution time statistics. Even the outliers in iteration numbers are below 10 attempts, indicating a consistent pattern and more reliable performance compared to other methods. Although the Fletcher-Reeves method has the best minimization success in Table 4.4, it exhibits multiple iteration outliers around 100, with most values above the median and the highest time-consuming statistics. This performance could have been even worse if the iteration number had not been constrained to 100, indicating that it is not a reliable and consistent method.

Table 4.6: Execution times of the algorithms

Algorithms	*Time Elapsed (s)*		
	Min	*Max*	Mean
Hestenes-Stiefel	0.1510	0.8914	0.3696
Polak–Ribière	0.1027	5.0604	0.4593
Fletcher-Reeves	0.1496	5.2935	1.3254
Steepest Descent	0.2325	2.2094	0.5919
Newton-Raphson	0.0979	0.4094	0.1920

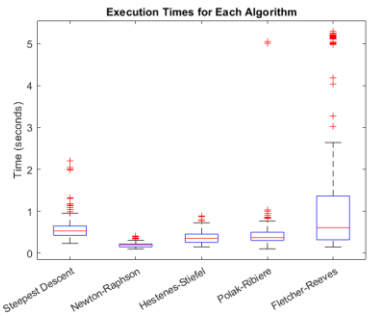

Figure 4.6: Elapsed Time Boxplot of the Algorithms

V. CONCLUSION

The optimization of the Griewank (GW) function poses significant challenges due to its 500 local minima in a two-dimensional space. The investigation into the convergence behavior of optimization algorithms reveals diverse patterns, particularly in response to different initial points.

Benchmarking across the function's domain reveals that each method achieves nearly a 50% success rate in objective function minimization; however, in terms of iteration and execution time, the Newton-Raphson method emerges as the most reliable. It demonstrates the fewest attempts, lowest mean, and lowest median iteration numbers, along with the lowest execution time statistics. The Newton-Raphson method consistently and efficiently converges, underscoring its suitability for handling sophisticated objective functions like the GW function.

The number of steps to find the minimum of the GW function varies among the algorithms. The Newton-Raphson method has the fewest attempts, with most outliers below 10 iterations, indicating highly efficient performance. In contrast, the Fletcher-Reeves method exhibits a mean of 26.6300 with multiple iteration outliers around 100, which is the benchmark threshold. The Polak-Ribière, Hestenes-Stiefel, and Steepest Descent methods behave similarly to each other, resulting in considerably similar iteration numbers, but not lower than Newton-Raphson.

The execution times of the algorithms correlate with their iteration counts as it can be seen in Table 4.5 and 4.6. The Newton-Raphson method has the lowest execution time statistics, reflecting its efficiency in terms of the number of iterations required. Conversely, the Fletcher-Reeves method has the highest time-consuming statistics due to the large number of iterations needed. This correlation makes sense because algorithms requiring fewer iterations generally have shorter execution times, whereas those needing more iterations, like Fletcher-Reeves, naturally take longer to execute.

Convergence of these algorithms depends significantly on the initial conditions. With the initial point x_0^2, all algorithms converge to nearly identical optimal points and values, demonstrating consistent minimization when the initial point is conveniently chosen. However, with the initial point x_0^3, all algorithms exhibit poor performance, with an unexpected increase in the function value. It can be concluded that initial points influence where the algorithms move towards, which is crucial for functions with many local minima like the GW function.

The trade-off between iteration count and execution time arises from the balance between algorithm efficiency and initial condition suitability. Algorithms like Newton-Raphson are efficient in both iteration count and execution time, particularly with favorable initial conditions. In contrast, while Fletcher-Reeves may eventually find a good minimization, it requires more iterations and consequently more execution time. This trade-off highlights the importance of choosing the right initial conditions and algorithms to optimize both time and iteration efficiency.

In a case where the stopping criterion and the absolute error bound change, both the number of steps and execution times will likely be affected. Tighter stopping criteria or a smaller epsilon value generally increase the number of iterations and execution time, as the algorithm works harder to meet the stricter requirements and achieve a more accurate result. In contrast, a looser stopping criterion or larger epsilon value decreases the number of iterations and execution time, as the algorithm can stop earlier, accepting a less precise solution.

In summary, while the optimization of the GW function presents challenges, careful algorithm selection, initial point consideration, and stopping criteria can significantly impact performance and accuracy.

REFERENCES

[1] Ali, M. M., Khompatraporn, C., & Zabinsky, Z. B. (2005). A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of global optimization*, *31*, 635-672.

[2] Dalla, C. E. R., da Silva, W. B., Dutra, J. C. S., & Colaço, M. J. (2021). A comparative study of gradient-based and meta heuristic optimization methods using Griewank benchmark function. *Braz J Dev*, *7*, 55341-55350