

# A comparative study of gradient-based and meta-heuristic optimization methods using Griewank benchmark function

Dalla, C. E. R<sup>1\*</sup>; da Silva, W. B.<sup>1</sup>; Dutra, J. C. S.<sup>1</sup>; Colaço, J. M.<sup>2</sup>;

<sup>1</sup>Department of Mechanical Engineering, Universidade Federal do Espírito Santo, Vitória, ES, Brazil

<sup>2</sup>COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brazil

carloseduardodalla@gmail.com

## Abstract

Optimization methods are frequently applied to solve real-world problems such, engineering design, computer science, and computational chemistry. This paper aims to compare gradient-based algorithms and the meta-heuristic particle swarm optimization to minimize the multidimensional benchmark Griewank function, a multimodal function with widespread local minima. Several approaches of gradient-based methods such as steepest descent, conjugate gradient with Fletcher-Reeves and Polak-Ribiere formulations, and quasi-Newton Davidon-Fletcher-Powell approach were compared. The results presented showed that the meta-heuristic method is recommended for function with this behavior because is no needed prior information of the search space. The performance comparison includes computation time and convergence of global and local optimum.

## 1 Introduction

Nowadays, many optimization problems emerge from engineering, physical sciences, and the computation field. Applications of unconstrained optimization algorithms are present in the areas of digital signal processing, automatic control of processes, mechanical systems, robotics, and machine learning.[1]. Therefore, is desirable to have computational effortless and efficient algorithms to minimize or maximize multivariable functions. In this article, we made a comparative study between gradient-based optimization methods and heuristic particle swarm optimization (PSO) to solve a benchmark Griewank function represented in Equation 1.

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

with  $x_i \in [-600, 600]$ , for  $i = 1, \dots, n$  where the global minima is  $f(\mathbf{x}) = 0$  located at  $x_i = [0, \dots, 0]$ , but the Griewank function has many widespread local minima, exponentially increasing with  $n$  which are regularly distributed. The highly increase of the local minima, suggest that the global minimum becomes extremely difficult to detect using deterministic methods. [2]

The complexity for 1D and 2D is shown in the zoomed-in Figure 1. The multimodal and unique global minima turn the Griewank function a typical test optimization function for stochastic algorithms, such as simulated annealing, differential evolution, and particle swarm optimization. [3]

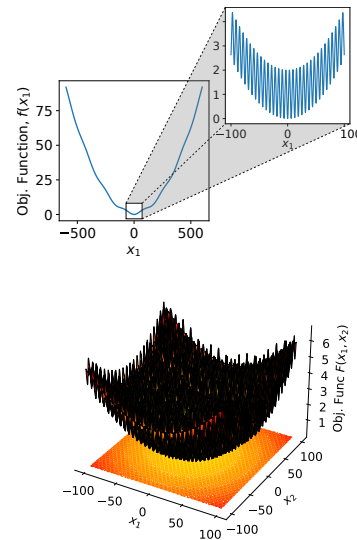


Figure 1: 1D and 2D Griewank function for  $x_1 \in [-600, 600]$  and a zoom at interval  $[-100, 100]$ .

## 2 Gradient based algorithms

The unconstrained gradient-based algorithms are iterative methods that extensively use the gradient information of the objective function as a search direction in the design space during the iterative procedure. [4, 5]

### 2.1 Steepest descent method

The steepest descent method is one of the oldest and most known methods used to minimize functions that have several variables. This method, considered the simplest of the gradient type methods, was created in 1847 by Cauchy and is called a deterministic method, where the results converge to the same point if have the initial guess.[1] The algorithm is shown below,

---

**Algorithm 1:** Steepest descent method

---

```

Initial guess  $\mathbf{x}_0$  at  $k = 0$ ;
while  $\|\nabla f(\mathbf{x}_k)\| > tol$  do
    Find the search direction  $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$ ;
    Solve for  $\alpha_k$  by minimizing  $f(\mathbf{x}_k + \alpha \mathbf{d}_k)$ ;
    Update the results  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$ ;
     $k \leftarrow k + 1$ 
end

```

---

### 2.2 Conjugate gradient method

The conjugate gradient method was originally introduced by Hestenes and Stiefel [6] for the solution of a linear system of equations  $\mathbf{C}\mathbf{x} = -\mathbf{b}$  where  $\mathbf{C}$  is symmetric and positive definite, which is equivalent to minimize a quadratic function. However, if the function to be minimized,  $F(\mathbf{x})$ , is not quadratic, the nonlinear conjugate gradient method is applied to update iteratively the solution vector  $\mathbf{x}$ , until the convergence is reached as shown in Algorithm 2. [7]

---

**Algorithm 2:** Nonlinear conjugate gradient method

---

```

Task: Minimize  $F(\mathbf{x})$ .
Initialize:
    Start with an initial guess  $\mathbf{x}_0$  and compute the gradient  $\mathbf{g}_0 = \nabla F(\mathbf{x}_0)$ .
    Set the initial search direction  $\mathbf{d}_0 = -\mathbf{g}_0$ 
for  $j = 0, 1, 2, \dots, k_{max}$  do
    Compute the step length  $\alpha_j$  by minimizing  $F(\mathbf{x}_j + \alpha \mathbf{d}_j)$ 
    Set new solution vector  $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{d}_j$ 
    Compute new gradient  $\mathbf{g}_{j+1} = \nabla F(\mathbf{x}_{j+1})$ 
    Compute search direction  $\mathbf{d}_{j+1} = -\mathbf{g}_{j+1} + \beta \mathbf{d}_j$ 
end

```

---

The first steps of nonlinear conjugate gradient method are similar to steepest descent method. However, the search direction is updated by assuming the orthogonality between next and previous gradient present in  $\beta$  proposed originally by Fletcher and Reeves [8] like in Equation 2,

$$\beta^{FR} = \frac{\mathbf{g}_{j+1}^T \mathbf{g}_{j+1}}{\mathbf{g}_j^T \mathbf{g}_j} \quad (2)$$

The Polak-Ribiere conjugate gradient [9] method use

$$\beta^{PR} = \frac{(\mathbf{g}_{j+1} - \mathbf{g}_j)^T \mathbf{g}_{j+1}}{\mathbf{g}_j^T \mathbf{g}_j} \quad (3)$$

According to Exl et al.[7] the nonlinear conjugate gradient method that use (3) instead of (2) are believed to have more efficient convergence characteristics due the self-correcting behavior of term  $(\mathbf{g}_{j+1} - \mathbf{g}_j)$ .

### 2.3 Quasi-Newton methods

The quasi-Newton method is an iterative procedure for solving an unconstrained minimization. The method is based on building up curvature information during the iterations. It approximates the curvature of the objective function approximating the Hessian matrix information. This makes the iterations of quasi-Newton computationally more expensive, compared to the nonlinear conjugate gradient. However, stored information in the approximated Hessian might decrease the total number of iterations compared to the conjugate gradient method. [10]

---

**Algorithm 3:** Davidon-Fletcher-Powell quasi-Newton method

---

```

Task: Minimize  $F(\mathbf{x})$ .
Initialize:
    Start with an initial guess  $\mathbf{x}_0$  and compute the gradient  $\mathbf{g}_0 = \nabla F(\mathbf{x}_0)$ .
    Set the initial search direction  $\mathbf{d}_0 = -\mathbf{g}_0$ 
    Hessian matrix  $\mathbf{H}_0 = \mathbf{I}$ 
for  $j = 0, 1, 2, \dots, k_{max}$  do
    Compute the search direction  $\mathbf{d}_j = -\mathbf{H}_j \mathbf{g}_j$ 
    Compute the step length  $\alpha_j$  by minimizing  $F(\mathbf{x}_j + \alpha \mathbf{d}_j)$ 
    Set new solution vector  $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{d}_j$ 
    Calculate  $\mathbf{p}_j = \alpha_j \mathbf{d}_j$ , and  $\mathbf{g}_{j+1}$ 
    Calculate  $\mathbf{q}_j = \mathbf{g}_{j+1} - \mathbf{g}_j$ 
    Update  $\mathbf{H}_{j+1}$  like Equation 4
end

```

---

In the quasi-Newton method, two approaches for calculating the Hessian matrix are mainly discussed, the Broyden, Fletcher, Goldfarb and Shanno method (BFGS) and Davidon, Fletcher and Powell method (DFP). For this work, the DFP method was implemented, as in Equation 4

$$\mathbf{H}_{j+1} = \mathbf{H}_j - \frac{\mathbf{p}_j \mathbf{p}_j^T}{\mathbf{p}_j^T \mathbf{q}_j} + \frac{\mathbf{H}_j \mathbf{q}_j \mathbf{q}_j^T \mathbf{H}_j}{\mathbf{q}_j^T \mathbf{H}_j \mathbf{q}_j} \quad (4)$$

### 3 Particle swarm optimization

As an alternative to gradient-based methods, heuristic approaches were developed to mitigate the need for gradient information required in previous methods related above. The particle swarm optimization was proposed by Eberhart as a stochastic optimization technique inspired by social (swarm) and individual (particle) behavior of movement of bird and fish migration for food sources.

#### 3.1 PSO Algorithm

In the PSO algorithm, the individuals of the population are represented as particles that have position and velocity in the search space  $\mathbb{R}^n$ , where  $n$  is the dimension of space. The evolution process of the method is based on the update of the velocity of particles during the iteration process, modifying their positions. Consider a particle  $i$  in a epoch  $k$  is represented by  $x_k^i$  and the velocity is  $v_k^i$ . The position and velocity of each particle are allocated in vectors with  $n$  dimensions. The update of each particle position during the iterations are based on two information, the best position of particle  $p_k^i$  and the best global position  $g^k$ , having this information, the velocity of each particle can be updated as Eq. below,

$$v_i^{k+1} = \omega v_i^k + c_1 \text{rnd}_1(p_i^k - x_i^k) + c_2 \text{rnd}_2(g^k - x_i^k) \quad (5)$$

where  $v_i^{k+1}$  is the update of velocity,  $\omega$  is the inertia factor, for standard convention this values is maintained between 0.4 and 0.9,  $\text{rnd}_1$  and  $\text{rnd}_2$  are independent random variables with uniform distribution between 0 and 1. The values of the constants  $c_1$  and  $c_2$ , known as the cognition and social acceleration coefficient, respectively, were found to have superior performance when is 2. [11]

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (6)$$

### 4 Numerical experiments

All the numerical algorithms are implemented in Python language, using an Intel® Core TM i5-4210U CPU @ 1.70GHz, 4 cores, 6 GB RAM. We analyze the sensibility of steepest descent, nonlinear conjugate gradients, with the formulation of Fletcher-Reeves and Polak-Ribiere for computing conjugate directions, and the quasi-Newton with Davidon-Fletcher-Powell (DFP) formulations to approximate the Hessian matrix.

The results of the evaluation of the Griewank function at each iteration are shown in Figure 2. Is possible to note that the first step of all methods is equal, they are differentiated after the second iteration, where the conjugate gradient and DFP have an additional procedure in the algorithm. The gradient-based methods were developed to search local minima, thus have a sensibility to initial guess and characteristics of the objective

function. Also in the Figure 2 are the updated coordinates for each iteration on the surface of the objective function and visualize the accelerated convergence of the quasi-Newton methods and conjugated gradients. It is possible to evaluate by the Table 2 that the quasi-Newton method presents a faster convergence than the other methods, mainly because it is a more sophisticated method. Among the formulations of the conjugate gradient method, we can notice that it is a sensitive method to the initial estimate. Of all the tests performed, the Polak-Ribiere formulation proved to be more stable, because it presents a corrective term in the calculation of the search conjugate direction. Already, using the Fletcher-Reeves equation, due to variations in the Griewank function it tends to diverge from the nearest local minimum.

In all deterministic methods, the value of the objective function declines abruptly, and before iteration 5 they have all reached similar values, however, the DFP method reaches the tolerance criterion, with fewer iterations. The stop criteria are defined in terms of the gradient norm. A common procedure between the methods is the search for the step size  $\alpha_k$  in the search direction, this is a sensitive part of the process, Antoniou et al [1], shown that this value can be found analytically or numerically, via cubic interpolation or golden section search. In this work, we calculate it numerically for all deterministic methods.

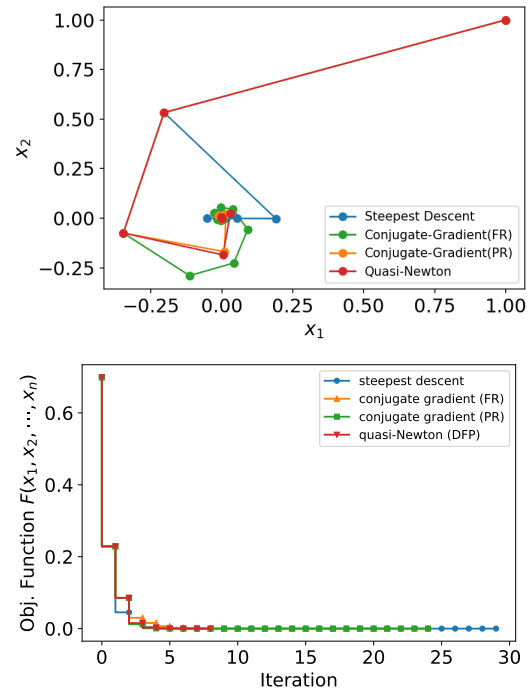


Figure 2: Local minima coordinates and objective function value at each optimization iteration for gradient based methods.

Table 2: Results of optimization of Griewank function with 4 variables using gradient-based algorithms for different initial guesses  $x_0$

Method	$x_0 = [1,1,1,1]$			$x_0 = [100,100,100,100]$		
	Elapsed Time(s)	Iterations	F(x)	Elapsed Time(s)	Iterations	F(x)
Steepest Descent	0.0784	29	1.154e-10	0.0977	30	9.828
CG (Fletcher-Reeves)	0.0698	21	7.588e-11	1.9725	342	9.828
CG (Polak-Ribiere)	0.0909	24	1.301e-10	0.1123	27	9.828
quasi-Newton (DFP)	0.0336	8	2.497e-11	0.0433	10	9.828
$x_0 = [0,0,0,0]$			$x_0 = [100.48, 97.645, 97.797, 100.328]$			

For the heuristic method of the particle swarm the values of the sociability and individuality parameters  $c_1$  and  $c_2$ , respectively, were kept equal to 2, a well-established value in the literature and the inertia parameter  $\omega$  used was equal to 0.5. To obtain the best result for the global minimum and to evaluate the influence of the number of particles in the solution, 200 independent optimization processes were performed for each quantity of individuals (20, 60, 100 particles) with a maximum number of 500 iterations per optimization, this measure was adopted, because in a single stochastic optimization the results may not be satisfactory. The search interval, in this case, was kept at -600, 600 in all  $n$  variables analyzed. The results are shown in Table 3. In the Figure 3, the histograms of the optimizations performed for  $n = 2$  and  $n = 4$  are shown, it is possible to notice that with the increase in the number of particles the distribution range narrows, since there is a greater probability that some of the particles are close to the global minimum. For results with  $n = 2$ , it can be noted that the best optimization reached the global minimum and as the number of particles increased, the average tended to approach the global minimum with smaller standard deviations. Regarding computational time, it is possible to conclude that heuristic methods are more costly for a number of particles over 60. For 20 particles in the population, we note that the elapsed time is similar to that of deterministic methods.

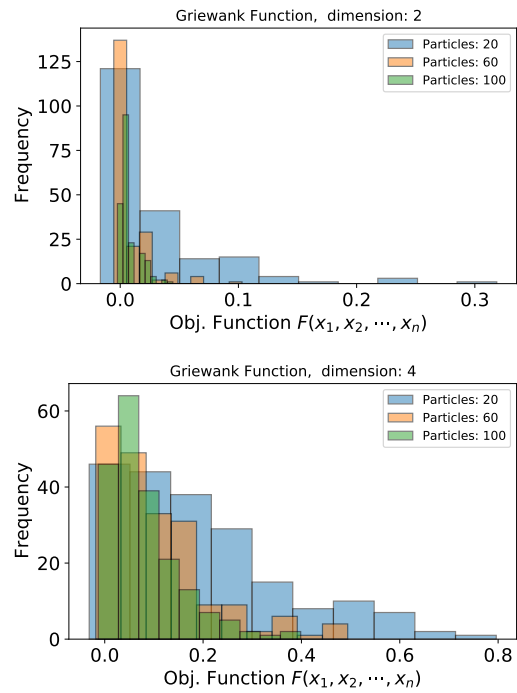


Figure 3: Histogram for optimization procedures using the Particle Swarm Optimization technique.

Table 3: Results for optimization of Griewank function with the heuristic method particle swarm (PSO).

n	Particles	gBest	Mean	Standard Deviation	Elapsed Time (s)
1	20	0.0000	0.00128	3.318e-03	0.015821
	60	0.0000	0.00005	6.958e-04	0.055438
	100	0.0000	0.00000	0.0000	0.077683
2	20	0.0000	0.03823	0.04538	0.03109
	60	0.0000	0.01285	0.01447	0.09956
	100	0.0000	0.00843	0.00830	0.15285
4	20	0.009864	0.23825	0.1763	0.06310
	60	0.009857	0.13132	0.1068	0.15769
	100	0.007396	0.09800	0.0756	0.24793

## 5 Conclusions

A gradient-based and heuristic optimization techniques are compared to solve the Griewank benchmark optimization function. We presented the implementations of the algorithms, based on different formulations from the simplest to sophisticated. The Griewank function has a multimodal behavior, with a large number of local minima. The deterministic method is not recommended for this type of function, because is necessary to have previous information of global minima coordinates, otherwise, the search procedure became extensive. With a different formulation, the heuristic method of particle swarm not require previous information about the model and can sweep large space domains. However, this approach demands a computational effort larger than deterministic methods.

## References

- [1] Antoniou Andreas and Lu Wu-Sheng. *Applications of Unconstrained Optimization*, pages 231–263. Springer US, Boston, MA, 2007.
- [2] M. Locatelli. *Journal of Global Optimization*, 25(2):169–174, 2003.
- [3] Yan Huang, Jian ping Li, and Peng Wang. Unusual phenomenon of optimizing the griewank function with the increase of dimension. *Frontiers of Information Technology & Electronic Engineering*, 20(10):1344–1360, October 2019.
- [4] Jasbir S. Arora. Numerical methods for unconstrained optimum design. In *Introduction to Optimum Design*, pages 411–441. Elsevier, 2012.
- [5] Xin-She Yang. Optimization algorithms. In *Introduction to Algorithms for Data Mining and Machine Learning*, pages 45–65. Elsevier, 2019.
- [6] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409, December 1952.
- [7] Lukas Exl, Johann Fischbacher, Alexander Kovacs, Harald Oezelt, Markus Gusenbauer, and Thomas Schrefl. Preconditioned nonlinear conjugate gradient method for micromagnetic energy minimization. *Computer Physics Communications*, 235:179–186, February 2019.
- [8] R. Fletcher. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, February 1964.
- [9] E. Polak and G. Ribiere. Note sur la convergence de méthodes de directions conjuguées. *Revue française d’informatique et de recherche opérationnelle. Série rouge*, 3(16):35–43, 1969.
- [10] R. E. Ricketts. Practical optimization. *International Journal for Numerical Methods in Engineering*, 18(6):954–954, June 1982.
- [11] Saptarshi Sengupta, Sanchita Basak, and Richard Peters. Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives. *Machine Learning and Knowledge Extraction*, 1(1):157–191, October 2018.