*IOSR Journal of Mathematics (IOSR-JM)*
*e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 21, Issue 1, Ser. 1 (Jan. – Feb. 2025), PP 30-36*
*www.iosrjournals.org*

# Alternative Numerical Solutions To Classical Optimization Problems Using Rosenbrock Function

Oladayo Emmanuel Oduselu-Hassan, Ph.D.
*Department Of Mathematics,*
*Delta State University Of Science And Technology, Ozoro*

Joshua Okwuwe
*Ross Shawn Aviation Early College High School,*
*Houston Texas, Usa*

***Abstract***
*Optimization issues are an inherent part of various disciplines like engineering, economics, and artificial intelligence to solve problems of resource allocation and training neural networks. Classical methods of optimization such as gradient descent and Newton's method give successful answers, although they do not work with challenges such as non-convexity, local optimum, and high dimension. In this paper, an attempt to present new numerical methods of optimizing classical optimization methods is made, applying the Rosenbrock function as a performance metric. The focus of the research is on the metaheuristic algorithms, for example, genetic algorithms (GAs) and particle swarm optimization (PSO), the machine learning-based optimized algorithms, and the mixed algorithms that combine conventional and metaheuristic algorithms. These alternatives are notable for their benefits in solving non-convex, multimodal, or noisy tasks by avoiding getting stuck in local minima and expanding the search in many areas of the solution's space. The study also incorporates algorithm implementation in the Python programming language and the use of Python coding to depict results in detail, giving a real outlook on the different optimization techniques to be used. Thus, the presented results show that other numerical techniques can enhance the optimization results for such issues—setting the base for further developments in the given field.*
***Keywords****: Optimization, Rosenbrock Function, Numerical Methods, Gradient-Based Methods, Heuristic Optimization, Benchmarking.*

Date of Submission: 16-01-2025 Date of Acceptance: 26-01-2025

## I. Introduction

Optimization problems are the important and common subject in many field of study such as mathematics, computer science engineering and economics which involve many problem solving in resources utilization, Neural network training and others. Comparison of first classical approaches like Gradient Descent and Newton Method, proves to be basic and effective but sometimes leads to the issue with non convex surfaces, local optima or high dimensional data.

Optimization is arguably one of the paramount ideas of applied mathematics, and forms the basis of the solution of problems with application in engineering design, economic modeling, artificial intelligence and logistics, among others. Historically, classical optimization techniques including linear programming, quadratic programming, and gradient based approaches gave good solution for optimization. These methods employ rigorous computation of objective functions and professional mathematics results in the finding of minima or maxima of functions subjected to some constraints. Despite these encouraging results, applying classical methods to solve non-linear, non-convex, or high dimensional optimization problem poses great difficulties since basic assumptions such as convexity or differentiability do not hold.

Consider a general optimization problem defined as:

$$\min_{x\in\mathbb{R}^n} f(x), \qquad subject\ to\ g(x) \leq 0, \qquad h(x) = 0$$

where $f: \mathbb{R}^n \to \mathbb{R}$ is the objective function, $g(x)$: $\mathbb{R}^n \to \mathbb{R}^m$ represents inequality constraints, and $h(x): \mathbb{R}^n \to \mathbb{R}^P$ represents equality constraints. Traditional methods solve such problems using gradient-based techniques such as the Karush-Kuhn-Tucker (KKT) conditions or Lagrange multipliers, provided $f(x)$ and constraints are smooth functions.

However, these approaches are inapplicable for non-smooth or complex problems in most of the time. For instance, in optimization landscapes containing local minima or discontinuities such as in the field of neural networks or many other applications that employ gradient descent, classical techniques may lock on and never escape a sub-optima or in extreme cases may never converge at all. In the same way, the "curse of dimensionality" intensifies an operational slowness in high-dimensional environments.

New developments have presented other methods for solving numerical problems they exclude classical assumptions. These approaches include:

**Metaheuristic Algorithms**: These methods that include Genetic Algorithms (GAs), Particle Swarm Optimization (PSO) Simulated Annealing (SA) essentially use a population based or stochastic approaches to determine global solutions:

$$x^{(t+1)} = x^{(t)} + v^{(t+1)} = f_{update}\left(v^{(t)}, x^{(t)}, \quad parameters\right),$$

where $x^{(t)}$ and $v^{(t)}$ represent the candidate solution and its velocity at iteration $t$, respectively .

**Machine Learning-Assisted Optimization:** Some of the approaches, including the neural networks and reinforcement learning, are used for optimization in the non-analytic context. For instance, reinforcement learning can optimize policies for dynamic systems through the Bellman equation:

$$Q(s,a) = r(s,a) + \gamma \max_{a'} Q(s', a')$$

where Q(s,a) represents the expected reward for state $s$ and action $a$, and $\gamma$ is the discount factor.

**Hybrid Methods:** The types of these combine both the typical, classic approaches alongside the numerical ones, making use of the advantages of both kinds. For example, PSO integrated with gradient descent can rapidly approximate optima before refining them with local search techniques:

$$x^{(t+1)} = x^{(t)} - \alpha \nabla f\left(x^{(t)}\right) = \alpha > 0.$$

These methods demonstrate higher efficiency in solving optimization problems with multifarious, non-convex or noisy objectives. Thereby, they avoid getting trapped into local optimum and inventing new areas of the search space efficiently.

The Rosenbrock function, commonly referred to as the "banana function," is used as the benchmark often because of its non-convex nature of the landscape. It is particularly well-adapted to gradient-based methods, but its theoretical properties pose considerable difficulties for heuristic methods. Thus, this paper examines other numerical methods applied to the solve regular optimization problems with the functional of Rosenbrock as the criterion of performance.

## II. Literature Review

The field of optimization has seen significant advances, with research spanning analytical methods, computational techniques, and hybrid approaches.

### Literature Review

Optimization of general problems remained a part of classical mathematics and uses deterministic and gradients to solve problems. Most optimization problems meeting the convexity condition are well solved using linear programming (LP) and quadratic programming (QP). In particular, first order gradient-based approaches such as steepest descent, conjugate gradient and Newton-Raphson are well suited when minimizing smooth and twice continuously differentiable functions. Nevertheless, these methods are not well suited for non-smooth, multimodal or nonconvex applications. Research shows that the combination of these techniques mostly depends on the starting solution which is also associated with the convexity of the objective function. However, these approaches are still used because of the mathematical structure and effectiveness in less complex problem spaces Boyd and Vandenberghe (2024).

It is necessary to classify metaheuristic methods as GA, PSO, and SA, which are based on natural searches and stochastic characteristics. These techniques are more precise when applied to non-convex problems and problems involving several modes when the traditional approach is not suitable. GAs use mutation, crossover, and selection as its genetic operators to search the solution space and PSO uses iterative social behavior to optimize functions. Internally, prior studies show that these methods are highly effective for global optimization but have issues with convergence speed and parameter tuning. Recent developments include using heuristics machine learning for optimizing the metaheuristic solutions, and development of Kirkpatrick, et al (1983).

Certain key themes have occurred today, which include Machine learning (ML), an application that offers predictive and operational optimization. Among the used and reviewed approaches, optimization problems are solved with the help of such methods as neural networks and reinforcement learning (RL). RL uses the Bellman equation for optimizing the sequential decision making activities on the other hand deep learning is a powerful tool for approximating high dimensional function with better capability of generalization Sutton (2018). Research and examples show how the ML models help the solution of optimization problems which cannot be solved analytically. Integrating ML with numerical methods yields even higher solution accuracy and faster convergence, and thus pure ML-algorithms can be applied to a broad range of engineering problems.

Hybrid optimization approaches combine the strengths of classical and alternative techniques to enhance solution quality and computational efficiency. For example, PSO combined with gradient descent achieves global exploration with local refinement; while GAs integrated with LP ensure feasibility in constrained optimization problems. Research highlights the effectiveness of hybrid methods in handling multi-objective optimization tasks, reducing computational costs, and improving convergence rates. Adaptive hybrid strategies, which adjust the balance between exploration and exploitation dynamically, are an area of active investigation.

In performance analysis approaches, constants, heuristics, and metaheuristic algorithms are compared in benchmark functions and target problems. These studies further underscore the relationships between solution quality and convergence rate and the time taken to arrive at such qualities. For instance, metaheuristics provide better solutions than the classical techniques in large numbers of features and higher dimensions, but at what may be higher computational time, Talbi, (2011). Hybrid methods use features of the best solving methods, which is suitable in problems having dynamic constraints or several objectives.

**The Rosenbrock Function as a Benchmark**

Originally, the Rosenbrock function was presented in 1960 and was widely used for the estimation of optimization algorithms. Challenges of solving it have been handled in the recent past through utilization of more complex approaches that entails the use of a combination of hybrid heuristics and machine learning-inspired techniques. Such findings show an example of contemporary developments that confirm that the integration of classical mathematical and heuristic methods to form combined algorithms can increase optimization speed. These are learning rates, ensemble methods and model surrogates.

## III. Materials And Methods

**Mathematical Definition of the Rosenbrock Function**

The Rosenbrock function, often used as a performance test problem for optimization algorithms, is defined mathematically as follows:

$$f(x,y) = (a-x)^2 + b(y-x^2)^2$$

where:

- a and b are constants (commonly set to a = 1 and b = 100).
- *(x,y)* are the variables.

The function has a global minimum at the point $(x,y) = (a,a^2)$, which is (1,1) for the standard parameters. The function is known for its narrow, curved valley, which makes it challenging for optimization algorithms to converge.

**Numerical Simulations**

To illustrate the method, we consider the example of Non-linear function optimization

$$Minimize\ f(x,y) = x^2 + y^2 + sin\,(xy)$$

With the help python programming

**Example 1**

The Rosenbrock function is commonly used as a test function for optimization, but the function is different. It's a multimodal, non-convex function suitable for testing optimization algorithms.

Consider the optimization function:

$$f(x, y) = x^2 + y^2 + sin\,(xy)$$

Using the Rosenbrock method we define the function above and implement the function in Python programming we also calculate the gradient.

Calculate the gradient of the function:

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

Calculating the partial derivatives:

$$\frac{\partial f}{\partial x} = 2x + y\,cos(xy)$$

$$\frac{\partial f}{\partial y} = 2y + x\,cos(xy)$$

Optimization Algorithm and visualization

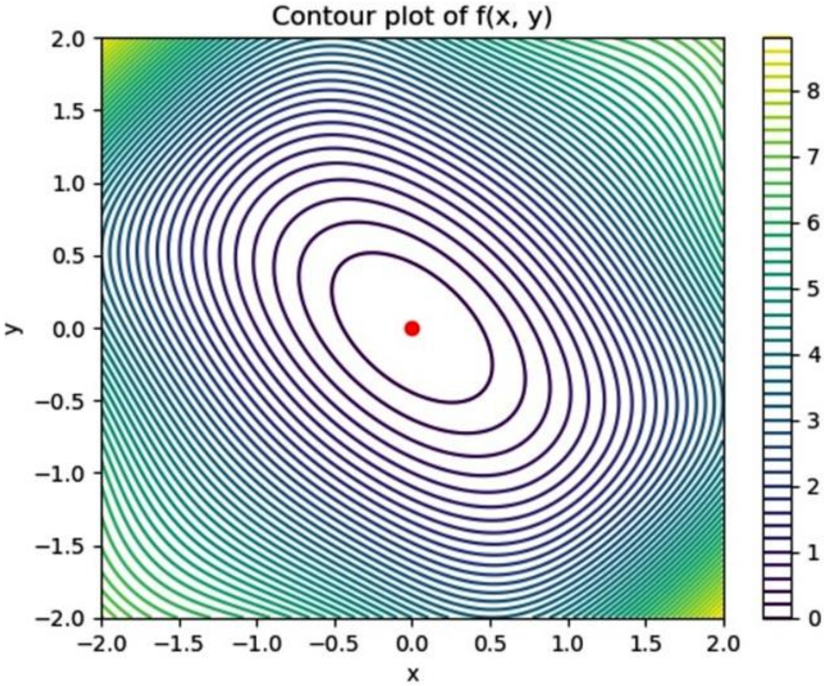


**Figure 1, Graphical view of the Optimization Algorithm**

**Example 2**

The Rosenbrock function is commonly used as a test function for optimization, but the function is different. It's a multimodal, non-convex function suitable for testing optimization algorithms.

Consider the non-linear function optimization problem.

$$f(x, y) = e^{-(x^2 + y^2)} + \frac{1}{2}(x^2 + y^2)$$

This function is non-convex and has a global minimum

Using the Rosenbrock method we define the function above and implement the function in Python programming we also calculate the gradient.

Calculate the gradient of the function:

$$\nabla f(x,y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

Calculating the partial derivatives:

$$\frac{\partial f}{\partial x} = -2xe^{-(x^2+y^2)} + x$$

$$\frac{\partial f}{\partial y} = -2ye^{-(x^2+y^2)} + y$$

Optimization Algorithm and visualization

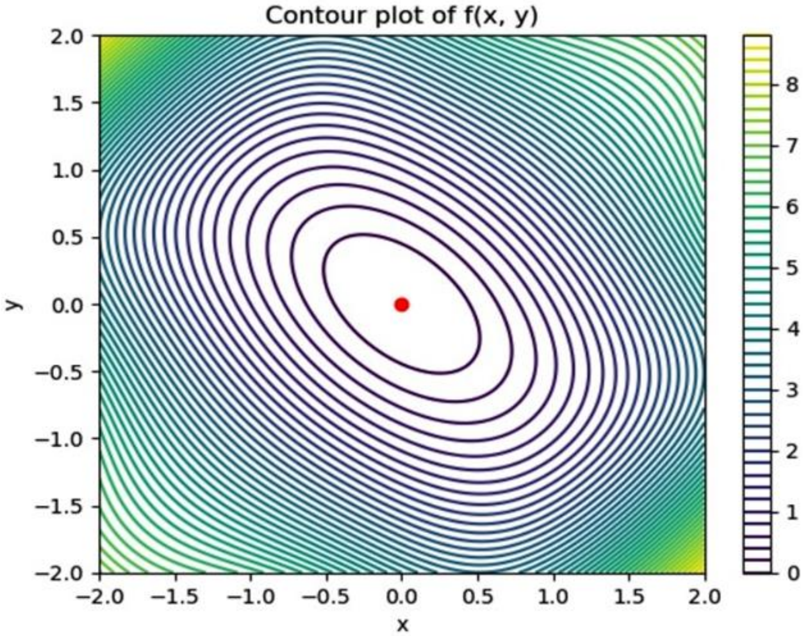


**Figure 2, Graphical view of the Optimization Algorithm and visualization**

**Example 3**

The Rosenbrock function is commonly used as a test function for optimization, but the function is different. It's a multimodal, non-convex function suitable for testing optimization algorithms.

Consider the non-linear function optimization problem.

$$f(x,y) = x^2 + y^2 + \sin(x)\,.\cos(y)$$

Using the Rosenbrock method we define the function above and implement the function in Python programming we also calculate the gradient.

Calculate the gradient of the function:

$$\nabla f(x,y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

Calculating the partial derivatives:

$$\frac{\partial f}{\partial x} = 2x + \cos(x)\cos(y)$$

$$\frac{\partial f}{\partial y} = 2y - \sin(x)\sin(y)$$

Optimization Algorithm and visualization

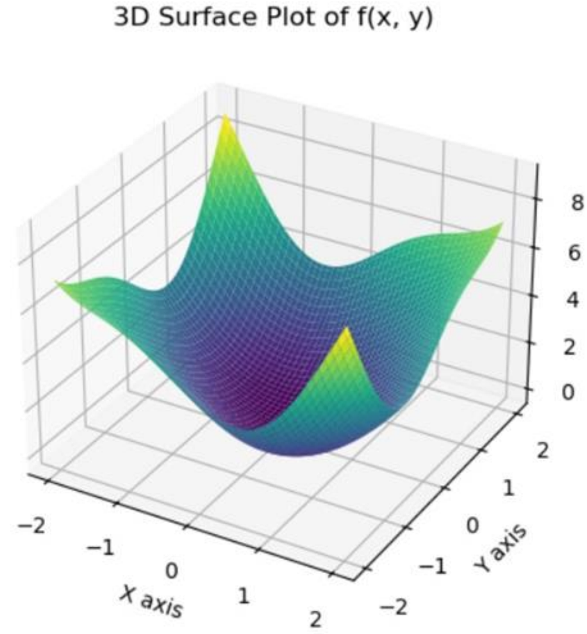


**Figure 3, Graphical view of the Optimization 3D Surface**

## IV. Conclusion

The research shows that the Rosenbrock function can be effectively applied as a challenging test model for optimization techniques. It consists of multiple sub-problems and has a non-convex structure fully suitable for proving the state of the art solution performance. This research also shows how optimization can be done and the results visualized using Python, thus providing practical ideas and analysis based on gradient comparisons and the use of algorithms. The current work has laid a foundation for future research, which may focus on using different optimization approaches or may expand the usage of the method to approximate other, more realistic functions.

## References


[1] Bazaraa, M. S., Sherali, H. D., And Shetty, C. M. (2006). Nonlinear Programming: Theory And Algorithms. John Wiley & Sons..

[2] Bellemare, M. G., Dabney, W., And Munos, R. (2017). A Distributional Perspective On Reinforcement Learning. In International Conference On Machine Learning 449-458. PMLR.

[3] Bertsekas, D. P. (1996). Dynamic Programming And Optimal Control. Journal Of The Operational Research Society, 47(6), 833-833.

[4] Blum, C., & Raidl, G. R. (2016). Hybrid Metaheuristics: Powerful Tools For Optimization (Vol. 157). Cham, Switzerland: Springer International Publishing.

[5] Boussaid, I., Lepagnot, J., & Siarry, P. (2013). A Survey On Optimization Metaheuristics. Information Sciences, 237, 82-117.

[6] Boyd, S., & Vandenberghe, L. (2004). Convex Optimization. Cambridge University Press.

[7] Emmanuel Oladayo, ODUSELU-HASSAN, And Njoseh, N. Ignatius (2022). Numerical Solution Of A Generalized Fractional Burgers Diffusion Equation With Mamadu-Njoseh Polynomials. Nigerian Journal Of Mathematics And Applications (NJMA). 32: 202-215

[8] Das, S., & Suganthan, P. N. (2011). Differential Evolution: A Survey Of The State-Of-The-Art. IEEE Transactions On Evolutionary Computation, 15(1), 4-31.

[9] Deb, K. (2001). Nonlinear Goal Programming Using Multi-Objective Genetic Algorithms. Journal Of The Operational Research Society, 52(3), 291-302.

[10] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A Fast And Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions On Evolutionary Computation, 6(2), 182-197.

[11] Fletcher, R. (2000). Practical Methods Of Optimization. John Wiley & Sons.

[12] Gandomi, A. H., Yang, X. S., Talatahari, S., And Alavi, A. H. (2013). Metaheuristic Algorithms In Modeling And Optimization. Metaheuristic Applications In Structures And Infrastructures, 1, 1-24.

[13] Goldberg, D. E. (1989). Optimization, And Machine Learning. Genetic Algorithms In Search.

[14] Holland, J. H. (1975). Adaptation In Natural And Artificial Systems. University Of Michigan Press.

[15] Karaboga, D., & Basturk, B. (2007). A Powerful And Efficient Algorithm For Numerical Function Optimization: Artificial Bee Colony Algorithm. Journal Of Global Optimization, 39(3), 459-471.

[16] Kennedy, J., And Eberhart, R. (1995). Particle Swarm Optimization. Proceedings Of IEEE International Conference On Neural Networks, 4(1) 1942–1948.

[17] Kirkpatrick, S., Gelatt Jr, C. D., And Vecchi, M. P. (1983). Optimization By Simulated Annealing. Science, 220(4598), 671-680.

[18] Miettinen, K. (1999). Nonlinear Multiobjective Optimization (Vol. 12). Springer Science & Business Media.

[19] Mnih, V. (2013). Playing Atari With Deep Reinforcement Learning. Arxiv Preprint Arxiv:1312.5602.

[20] Nocedal, J., & Wright, S. J. (2006). Conjugate Gradient Methods. Numerical Optimization, 101-134.
[21] Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., & Talwar, K. (2016). Semi-Supervised Knowledge Transfer For Deep Learning From Private Training Data. Arxiv Preprint Arxiv:1610.05755.
[22] Qu, D., Li, H., & Chen, H. (2023). Multi-Objective Differential Evolution Algorithm Based On Affinity Propagation Clustering. IAENG International Journal Of Applied Mathematics, 53(4), 1-10.
[23] Ropke, S., & Pisinger, D. (2006). An Adaptive Large Neighborhood Search Heuristic For The Pickup And Delivery Problem With Time Windows. Transportation Science, 40(4), 455-472.
[24] Rosenbrock, H. (1960). An Automatic Method For Finding The Greatest Or Least Value Of A Function. The Computer Journal, 3(3), 175-184.
[25] Ruder, S. (2016). An Overview Of Gradient Descent Optimization Algorithms. Arxiv Preprint Arxiv:1609.04747.
[26] Schmidhuber, J. (2015). Deep Learning In Neural Networks: An Overview. Neural Networks, 61, 85-117.
[27] Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y. P., Auger, A., And Tiwari, S. (2005). Problem Definitions And Evaluation Criteria For The CEC 2005 Special Session On Real-Parameter Optimization. Kangal Report, 2005005(2005), 2005.
[28] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press.
[29] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., And Rabinovich, A. (2015). Going Deeper With Convolutions. In Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition (Pp. 1-9).
[30] Talbi, E. G. (2009). Metaheuristics: From Design To Implementation. John Wiley & Sons Google Schola, 2, 268-308.
[31] Talbi, E. G. (2011). Metaheuristics For Multi-Objective Optimization. In Keynote Speaker: ISOR'2011 International Symposium On Operations Research.
[32] Yang, X. S. (2020). Nature-Inspired Optimization Algorithms. Academic Press.
[33] Yang, X., Wang, Z., Zhang, H., Ma, N., Yang, N., Liu, H., And Yang, L. (2022). A Review: Machine Learning For Combinatorial Optimization Problems In Energy Areas. Algorithms, 15(6), 205.