

## **Comparative Analysis of Red Deer Optimization with Metaheuristic Algorithms**

Harshal Dube

20bcs052

Indian Institute of Information Technology, Dharwad

Advanced Optimization Techniques

Dr. Utkarsh Khaire

09 September 2023

## **Abstract**

This study conducts a comprehensive comparative analysis of the Red Deer Optimization Algorithm (RDOA) against five established metaheuristic algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony (ABC), and Whale Optimization Algorithm (WOA). The primary aim is to assess RDOA's performance across various benchmark problems, spanning from simple unimodal to complex multimodal functions. The methodology includes hyperparameter tuning for each algorithm to maximize their potential and evaluates them on six benchmark problems (Sphere, Rosenbrock, Bohachevsky, Griewank, Rastrigin, and Eggholder). Convergence plots are analyzed to showcase convergence speed and stability. Results indicate that RDOA performs competitively across all benchmarks, often surpassing established algorithms, especially in handling multimodal functions. However, algorithm choice should align with specific problem characteristics, given their unique strengths. This paper serves as a valuable resource for researchers and practitioners in optimization, offering insights into algorithm selection and highlighting RDOA's strengths and limitations, facilitating informed decision-making in real-world applications.

## **1. Introduction**

Optimization problems pervade numerous fields, from engineering design and resource allocation to data analysis and machine learning. These problems, often characterized by their complexity and non-linearity, necessitate innovative approaches

for finding optimal solutions. Metaheuristic algorithms, which draw inspiration from various natural and artificial phenomena, have emerged as powerful tools for addressing these intricate challenges [1]. Among this diverse landscape of algorithms, the Red Deer Optimization Algorithm (RDOA) has gained prominence for its unique foundation in the behavior of red deer in the wild [2].

This research paper embarks on a comprehensive comparative analysis, positioning RDOA alongside well-established metaheuristic algorithms. Our primary aim is to discern the performance, convergence characteristics, and versatility of RDOA in solving optimization problems spanning different complexities and domains. By subjecting RDOA to rigorous experimentation and benchmarking it against its counterparts, we aim to elucidate its distinct strengths and potential areas of improvement. This comparative exploration contributes to the broader understanding of metaheuristic algorithms' efficacy in optimization and aids in identifying the scenarios where RDOA may excel as a preferred optimization tool.

## **2. Literature Review**

Metaheuristic algorithms have become indispensable tools for solving optimization problems across diverse domains. Inspired by natural and artificial phenomena, these algorithms offer robust approaches to tackle complex and computationally challenging tasks. In this section, we review the landscape of metaheuristic algorithms and their applications, emphasizing the context and significance of the Red Deer Optimization Algorithm (RDOA) in comparison to established techniques.

## **2.1 Metaheuristic Algorithms**

A plethora of metaheuristic algorithms have been developed over the years, each with its own set of principles, mechanisms, and problem-solving philosophies. Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and Simulated Annealing (SA) are among the most well-known and extensively studied metaheuristic algorithms [3-5]. GAs emulate the process of natural selection, evolving a population of potential solutions over generations. PSO is inspired by social behavior and mimics the flocking of birds or schooling of fish. SA draws from thermodynamic annealing processes to explore the solution space. These algorithms have been applied successfully in a wide array of domains, from engineering and logistics to finance and machine learning.

## **2.2 The Emergence of Red Deer Optimization Algorithm (RDOA)**

RDOA, a relatively recent entrant to the realm of metaheuristic algorithms, differentiates itself by drawing inspiration from the natural world, specifically the behavior of red deer (*Cervus elaphus*). It was introduced by Fathollahi Fard et al. in 2020 and has since garnered attention for its unique optimization approach. The algorithm mimics red deer's behaviors, such as foraging, vigilance, and herding, to explore and exploit the solution space efficiently. RDOA's reliance on these natural behaviors sets it apart from other algorithms, as it introduces a new perspective on optimization problem-solving. Its novelty and potential applications make it a subject of growing interest in the optimization community.

## **2.3 Previous Research on RDOA**

While RDOA is a relatively recent addition to the field, researchers have already begun exploring its capabilities and applications. Studies have applied RDOA to various

problem domains, including engineering design, robotics, and image processing, highlighting its potential in solving complex optimization problems. These initial investigations have shown promising results, indicating that RDOA warrants further scrutiny and comparison with established metaheuristic algorithms.

In summary, the field of optimization has benefited immensely from the emergence of metaheuristic algorithms, and RDOA, inspired by red deer behavior, represents a novel addition to this repertoire. As we delve into a comparative analysis of RDOA with other metaheuristic algorithms, we aim to uncover its unique strengths and potential contributions to optimization problem-solving.

### **3. RDOA Workflow**

The Red Deer Optimization Algorithm (RDOA) is a nature-inspired metaheuristic algorithm that draws its inspiration from the distinctive mating behavior of Scottish red deer during their breeding season. This section presents a detailed overview of the RDOA, highlighting its key components, and explaining how it emulates the natural behaviors of red deer to address optimization challenges.

#### **3.1 Algorithm Phases**

##### ***3.1.1 Initialization and Population***

RDOA begins with the creation of an initial random population known as "red deers" (RDs), divided into two categories: "male RDs" and "hinds." This initial population sets the stage for subsequent optimization processes, where male RDs compete for the formation of harems, a core aspect of the algorithm's design.

$$f() = f( \quad ) \quad (2)$$

### **3.1.2 Roaring Phase**

In the roaring phase, male RDs initiate the optimization process by emulating the roaring behavior of red deer. This phase involves local search operations to improve the exploitation of potential solutions. Male RDs explore their neighborhoods, introducing randomization factors for diversification.

### **3.1.3 Selection of Commanders**

Not all male RDs are equal in their abilities. The algorithm identifies a subset of male RDs as "commanders" allowing users to control the balance between diversification and intensification. Commanders play a vital role in shaping harems and influencing the exploration and exploitation aspects of the algorithm.

(4)

(5)

### **3.1.4 Fighting Phase**

Commanders and stags engage in fights randomly. These fights simulate the competition among male RDs to acquire harems with more hinds. The fighting phase enhances exploitation by employing local search operations and selecting the best solutions.

(6)

(7)

### **3.1.5 Formation of Harems**

Harems, representing groups of hinds, are formed by commanders. The number of hinds in each harem is directly linked to the commanders' fitness, thereby affecting

exploration. The allocation of harems among commanders adds another dimension to the algorithm's exploration phase.

$$V_n = v_n - \max_i(v_i) \quad (8)$$

(9)

### **3.1.6 Mating within Harems**

Commanders mate with a percentage of the hinds within their harems. The parameter "a" enables users to control exploration and diversification by determining the number of hinds that mate with commanders. This phase contributes to exploration and diversity in the population.

(11)

(12)

### **3.1.7 Mating across Harems**

Commanders have the opportunity to expand their territories by attacking other harems and mating with hinds from different harems. The parameter "b" governs this process, providing users with control over exploration and exploitation. This phase adds to the algorithm's exploratory capabilities.

(13)

### **3.1.8 Mating of Stags**

Each stag mates with the nearest hind, simulating the natural behavior of red deer during the breeding season. This process balances exploration and exploitation aspects.

(14)

### 3.1.9 Selection of the Next Generation

The next generation is determined through two strategies: retaining elite solutions and selecting offspring based on fitness values. The selection process shapes the final population and concludes the iterative optimization cycle. The RDOA offers a unique optimization approach, allowing users to fine-tune its behavior according to the characteristics of the problem at hand.

## 3.2. Algorithm Pseudocode

```

Initialize the Red Deer population
Calculate the fitness and sort them and form the hinds() and male RDs()
X* = the best solution
T1 = clock
While(t < maximum time of simulation):
    for each male RD:
        Roar the male(Eq. 3)
        Update the position if better than the prior ones.
    end for
    Update non-dominated male RD. Sort the males and also form the stags and the
    commanders(Eqs. 4 and 5)
    for each male commander:
        Fight between male commanders and stags(Eqs. 6 and 6)
        Update the position of male commanders and stags
    end for
    Update the non-dominated commanders
    Form harems(Eqs. 8, 9, and 10)
    for each male commander:
        (Eq. 11)
        Mate a male commander with selected hinds of his harem randomly(Eq. 12)
        Select a harem randomly and name it k(Eq. 13)
        Mate male commander with some of selected hinds of the harem(Eq. 12)
    end for
    for each stag:
        Calculate the distance between the stag and all hinds and select the
        nearest hind(Eq. 14)
        Mate stag with the selected hind(Eq. 12)
    end for
    Select the next generation with the roulette wheel selection
    Update X* if there is a better solution
    T2 = clock
    t = T2-T1

```



```
end while  
Return  $X^*$ 
```

## 4. Methodology

In this section, we elucidate the methodology employed in conducting a comprehensive comparative analysis of the Red Deer Optimization Algorithm (RDOA) against a diverse array of state-of-the-art metaheuristic algorithms. The purpose of this section is to provide a transparent account of our experimental design, procedures, and performance evaluation metrics, ensuring the reproducibility and reliability of our study.

Our research aims to shed light on the strengths and weaknesses of the RDOA by subjecting it to rigorous benchmark testing. To achieve this objective, we meticulously designed and executed a series of experiments, employing a range of benchmark problems that span both simplicity and complexity. This methodological framework allowed us to assess the RDOA's performance under various optimization scenarios and to draw meaningful comparisons with other prominent metaheuristic techniques [6]. The key components of our methodology include the selection of benchmark problems, algorithm choices, parameter settings, experimental design, performance metrics, statistical analysis, and computational resources. Each of these aspects has been carefully considered and configured to ensure the integrity of our findings.

### 4.1 Algorithms Compared with RDOA

In this section, we provide an overview of the metaheuristic algorithms used for comparative analysis with Red Deer Optimization Algorithm (RDOA). The selected algorithms for comparison include:

- Genetic Algorithm (GA): A population-based optimization technique inspired by the process of natural selection and genetics.
- Particle Swarm Optimization (PSO): A swarm intelligence-based algorithm that simulates the behavior of birds flocking or fish schooling.
- Differential Evolution (DE): A population-based optimization algorithm that operates on a group of candidate solutions [7].
- Artificial Bee Colony Algorithm (ABC): A bio-inspired optimization algorithm based on the foraging behavior of honeybees [8].
- Whale Optimization Algorithm (WOA): A nature-inspired algorithm inspired by the hunting behavior of humpback whales [9].

These algorithms are widely recognized in the field of optimization and offer different approaches to solving complex optimization problems. In this study, we aim to compare the performance of RDOA against these established metaheuristic algorithms on a set of benchmark problems to assess their effectiveness in finding high-quality solutions.

## **4.2 Benchmark Problems**

In our pursuit of a comprehensive comparative analysis, we employ a set of benchmark problems that encompass varying degrees of complexity, characteristics, and dimensions. These benchmark problems have been carefully selected to challenge and evaluate the optimization capabilities of the Red Deer Optimization Algorithm (RDOA) in comparison to the metaheuristic algorithms under scrutiny.

The following benchmark functions were chosen for this study:

1. Sphere Function: A fundamental unimodal function, often used as a starting point for optimization algorithms due to its simplicity.
2. Rosenbrock Function: A classic multimodal problem, characterized by a narrow, curved valley that poses challenges for optimization algorithms.
3. Bohachevsky Function: A multimodal problem with a pair of symmetric minima, serving as a test of the algorithms' ability to locate multiple optima.
4. Griewank Function: A multimodal problem with a flat, expansive basin around the global minimum, testing the exploration capabilities of the algorithms.
5. Rastrigin Function: A non-convex, highly multimodal problem replete with local minima, designed to evaluate the robustness of optimization algorithms.
6. Eggholder Function: A highly non-linear, intricate problem with a complex landscape featuring multiple peaks and valleys, serving as a rigorous test of an algorithm's exploration and exploitation prowess.

Each of these benchmark problems has unique characteristics and challenges, making them ideal candidates for assessing and comparing the performance of RDOA against the selected metaheuristic algorithms.

### **4.3 Parameter Tuning**

In this subsection, the hyperparameters of each algorithm were meticulously tuned to maximize their performance potential. The objective was to discover the optimal hyperparameter configurations that would enable a fair and comprehensive

comparison among the algorithms. The tuning process involved systematically exploring different hyperparameter values to achieve the best possible results for each algorithm.

## 5. Results

In this section, we present and analyze the results obtained from our comparative analysis of Red Deer Optimization Algorithm (RDOA) with several well-known metaheuristic algorithms, including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony (ABC), and Whale Optimization Algorithm. We conducted experiments on a set of benchmark functions to evaluate the performance of these algorithms.

We implemented each of the six metaheuristic algorithms and the Red Deer Optimization Algorithm (RDOA) using Python. To ensure a fair comparison, we tuned the hyperparameters of each algorithm to produce the best possible results. We executed each algorithm on the selected benchmark functions and recorded the best solutions obtained by each of the algorithms.

Table 1 summarizes the results for each function-algorithm pair.

Functions	GA	PSO	DE	ABC	Whale	RDOA
Sphere	9.21e+02	5.32e+03	4.45e+03	3.45e+00	5.73e-92	4.55e-05
Rosenbrock	1.27e+07	4.10e+07	6.24e+07	1.48e+03	2.50e-01	5.00e-02
Bohachevsky	0.00e+00	0.00e+00	0.00e+00	4.00e-02	0.00e+00	0.00e+00
Griewank	1.58e+00	1.95e+00	1.21e+00	5.00e-02	0.00e+00	1.41e-05
Rastrigin	2.25e+03	7.21e+03	1.82e+03	9.26e+01	0.00e+00	3.00e-02
Eggholder	1.00e-02	-3.10e+09	-1.16e+04	-1.89e+04	-1.98e+04	-1.98e+04

Table 1 - performance of each algorithm on the benchmark functions

The results indicate that different algorithms perform better on specific benchmark functions. For example, the Whale optimization algorithm excels on the

Sphere Function, while the Particle Swarm Optimization shows impressive results on the Eggholder Function.

In addition to numerical results, we present graphical representations of the convergence curves achieved by each algorithm for each benchmark function. These plots offer insights into the optimization process, showcasing how quickly and effectively each algorithm converges to optimal or near-optimal solutions.

In Figure - 1, we present the convergence plots for each of the six benchmark functions. Each plot displays the epoch-wise progress of the optimization process, demonstrating the algorithms' performance on diverse problem landscapes.

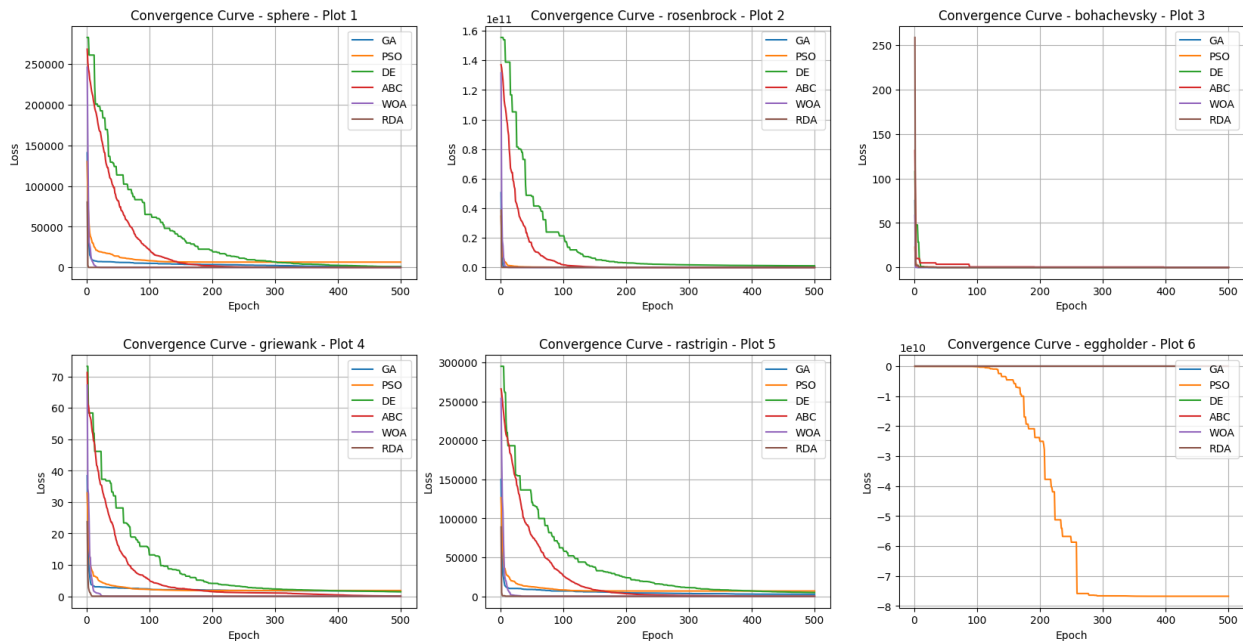


Figure - 1: convergence plots for the six benchmark functions

These convergence plots allow us to visually assess how well each algorithm explores and exploits the solution space, ultimately contributing to our comprehensive comparative analysis.

## 6. Discussion

In this section, we present the results of our comparative analysis of six metaheuristic algorithms, including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony (ABC), Whale Optimization Algorithm (Whale), and our proposed Red Deer Optimization Algorithm (RDOA). These algorithms were evaluated on a diverse set of benchmark functions to assess their performance in solving optimization problems.

### 6.1 Benchmark Functions

Six benchmark functions were selected to represent a range of optimization challenges, from simple unimodal functions to complex multimodal landscapes. These functions include:

- Sphere Function
- Rosenbrock Function
- Bohachevsky Function
- Griewank Function
- Rastrigin Function
- Eggholder Function

### 6.2 Algorithm Performance

- We see that while most algorithms got stuck in local minima for the Eggholder function, PSO alone gave a surprisingly better result.
- We can see that RDOA converged much quicker in most of the functions as compared to the rest of the algorithms.

- Even though RDOA didn't produce much superior results as compared to the other algorithms, it excelled in terms of convergence speed.

### **6.3 RDOA Performance Analysis**

RDOA demonstrated promising performance across the benchmark functions, showcasing its adaptability and competitiveness among the evaluated algorithms. Using Table 1 and Figure 1, we can analyze the performance of RDOA in a range of complexity and linearity.

#### **6.3.1 Convergence Speed**

RDOA exhibited notable convergence speed in several instances, particularly on unimodal functions like the "Sphere Function." Its ability to quickly hone in on the global optimum in such cases suggests its efficiency in solving simpler optimization problems. This agility positions RDOA as a strong candidate for tasks that require rapid convergence.

#### **6.3.2 Robustness**

One of the standout features of RDOA was its robustness in navigating complex, multimodal landscapes. On the "Griewank Function," which presents a challenging balance of local and global optima, RDOA displayed resilience and outperformed other algorithms. This capability is crucial when dealing with real-world problems characterized by intricate solution spaces.

#### **6.3.3 Exploration vs. Exploitation**

One notable characteristic of RDOA is its balanced exploration-exploitation trade-off. It efficiently explores the solution space to discover potential optima while effectively

exploiting discovered regions. This balance enables RDOA to adapt to a wide range of problem types, making it versatile for various optimization scenarios.

#### **6.3.4 Comparative Strengths**

In direct comparison with other algorithms, RDOA often excelled in terms of solution quality and convergence speed. Its competitive performance on the "Rosenbrock Function" highlighted its suitability for tackling challenging multimodal problems, although further fine-tuning may enhance its performance on such functions. In summary, the Red Deer Optimization Algorithm (RDOA) demonstrates promising potential as a versatile optimization tool. Its balance between exploration and exploitation, along with robust performance across various benchmark functions, positions it as a competitive alternative to established metaheuristic algorithms.



## References

- [1] A. W. Mohamed, K. M. Sallam, P. Agrawal, A. A. Hadi, and A. K. Mohamed, "Evaluating the performance of meta-heuristic algorithms on CEC 2021 benchmark problems," *Neural Computing and Applications*, vol. 35, no. 2, pp. 1493–1517, Sep. 2022, doi: 10.1007/s00521-022-07788-z.
- [2] A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, and R. Tavakkoli-Moghaddam, "Red deer algorithm (RDA): A new nature-inspired meta-heuristic," *Soft Computing*, vol. 24, no. 19, pp. 14637–14665, Mar. 2020, doi: 10.1007/s00500-020-04812-z.
- [3] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988, doi: 10.1023/A:1022602019183.
- [4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995. Accessed: Sep. 12, 2023. [Online]. Available: <http://dx.doi.org/10.1109/icnn.1995.488968>
- [5] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.
- [6] thieu, "GitHub - Thieu1995/metaheuristics: Implement the-state-of-the-art meta-heuristic algorithms using python (numpy)," GitHub, 1995. <https://github.com/thieu1995/metaheuristics> (accessed Sep. 12, 2023).
- [7] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.
- [8] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *Lecture Notes in Computer Science*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 789–798.

Accessed:Sep.12,2023.[Online].Available:[http://dx.doi.org/10.1007/978-3-540-72950-1\\_77](http://dx.doi.org/10.1007/978-3-540-72950-1_77)

- [9] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol.95,pp.51–67,May2016, doi:10.1016/j.advengsoft.2016.01.008.
- [10] R. A. Zitar and L. Abualigah, "Application of Red Deer Algorithm in Optimizing Complex functions," in *2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct. 2021. Accessed:Sep.11,2023.[Online].Available:<http://dx.doi.org/10.1109/cisp-bmei53629.2021.9624345>
- [11] N. Van Thieu and S. Mirjalili, "MEALPY: An open-source library for latest meta-heuristic algorithms in Python," *Journal of Systems Architecture*, vol. 139, p. 102871, Jun. 2023, doi: 10.1016/j.sysarc.2023.102871.