Journal of Industrial and Systems Engineering
Vol. 16, No. 3, pp. 106-117
Summer (July) 2024

JISE

# Greedy Man Optimization Algorithm (GMOA): A Novel Approach to Problem Solving with Resistant Parasites

**Hamed Nozari[1]*, Hossein Abdi[1]**

*[1]Department of Management, Azad University, Dubai Branch, Dubai, United Arab Emirates*

## Abstract

This paper introduces the Greedy Man Optimization Algorithm (GMOA), a novel bio-inspired metaheuristic approach for solving complex optimization problems. Inspired by competitive individuals resisting change, GMOA incorporates two unique mechanisms: MMO resistance, which prevents premature replacement of solutions, and periodic parasite removal, which promotes diversity and avoids stagnation. The algorithm is evaluated on standard benchmark functions, including Sphere, Rastrigin, Rosenbrock, and Griewank, and its performance is compared with established algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Ant Colony Optimization (ACO). Results demonstrate that GMOA outperforms these methods in terms of solution quality, convergence rate, and robustness. Statistical significance tests validate the reliability of the results. GMOA's ability to balance exploration and exploitation makes it a promising tool for various real-world applications, including supply chain optimization and healthcare resource allocation.

**Keywords**: Greedy Man Optimization Algorithm (GMOA), Bio-inspired optimization, MMO resistance mechanism, Metaheuristic algorithms, Exploration-exploitation balance

## 1- Introduction

Optimization is a critical process in a wide range of scientific, industrial, and engineering disciplines. Many real-world problems require finding the best possible solution under given constraints, such as minimizing costs, maximizing efficiency, or achieving an optimal balance between conflicting objectives. Over the past few decades, numerous optimization algorithms inspired by natural and social phenomena have been developed, such as Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). These algorithms have demonstrated significant success in solving complex optimization problems by mimicking the behavior of biological systems or social interactions (Arora, 2015).

* Corresponding Author

In recent years, researchers have been increasingly interested in designing algorithms that draw inspiration from unconventional metaphors, including human behavior. The **Greedy Man Optimization Algorithm (GMOA)** is a novel bio-inspired optimization approach that models the behavior of individuals who tenaciously cling to their positions, akin to parasites resisting change. The algorithm introduces a unique metaphorical component—"MMO" (metaphorical parasites)—which represents the resistance of solutions to being replaced or improved, a key innovation that distinguishes GMOA from other optimization methods.

The primary motivation behind GMOA is to address two common issues in optimization: **premature convergence** and **lack of diversity**. Many traditional optimization algorithms tend to converge prematurely to suboptimal solutions, especially in problems with complex, multimodal search spaces. By incorporating the concept of MMO resistance, GMOA maintains a dynamic balance between exploration (searching new areas of the solution space) and exploitation (refining existing solutions). This resistance mechanism prevents overly greedy solutions from dominating the population too early, thus preserving diversity and enhancing the likelihood of finding a global optimum.

Another notable feature of GMOA is its periodic **parasite removal mechanism**, which mimics a natural process of overcoming resistance and promoting renewal. This mechanism involves introducing mutations or recombinations at regular intervals to reduce resistance and inject new genetic material into the population. As a result, GMOA not only prevents stagnation but also encourages continuous improvement of solutions over time.

This paper introduces GMOA in detail, providing a comprehensive explanation of its components, pseudocode, and a high-level schematic. To demonstrate its effectiveness, the algorithm is applied to several benchmark optimization problems, and its performance is compared with that of well-established methods such as GAs and PSO. The results show that GMOA consistently achieves high-quality solutions while maintaining robust convergence properties.

The remainder of this paper is organized as follows: Section 2 reviews related work in optimization algorithms, highlighting the gaps that GMOA aims to fill. Section 3 presents the detailed design and pseudocode of GMOA, including the initialization process, competition mechanism, MMO resistance calculation, and parasite removal strategy. Section 4 describes the experimental setup and benchmark problems used to evaluate the algorithm. Section 5 discusses the results, comparing GMOA’s performance with that of other algorithms. Finally, Section 6 concludes the paper and suggests potential directions for future research, such as multi-objective extensions and real-world applications in areas like supply chain optimization and healthcare systems.

By leveraging an innovative metaphor and unique mechanisms, GMOA offers a fresh perspective on solving complex optimization problems. The algorithm's ability to balance exploration and exploitation through MMO resistance and periodic mutation makes it a promising addition to the growing family of bio-inspired optimization methods.

## 2- Literature review

Optimization algorithms inspired by natural phenomena and human behavior have been extensively studied and applied to solve various real-world problems. Among the most prominent bio-inspired algorithms are Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). These methods draw their inspiration from evolutionary processes, swarm intelligence, and the foraging behavior of ants, respectively (Eiben & Smith, 2015; Kennedy & Eberhart, 1995).

Genetic Algorithms (GAs) were among the first bio-inspired optimization techniques, introduced by Holland (1975). GAs mimics the process of natural selection by iteratively selecting, recombining, and mutating candidate solutions. They have been successfully applied to numerous combinatorial and continuous optimization problems. However, GAs is often criticized for their susceptibility to premature convergence and loss of diversity in the population, particularly in highly multimodal search spaces.

Particle Swarm Optimization (PSO), proposed by Kennedy and Eberhart (1995), models the collective behavior of birds flocking or fish schooling. PSO has gained popularity due to its simplicity and effectiveness in continuous optimization problems. Each particle in the swarm represents a potential solution, and particles adjust their trajectories based on their own experience and that of their neighbors. While PSO excels in exploration, it may suffer from stagnation when particles converge prematurely to suboptimal regions of the search space.

Ant Colony Optimization (ACO)**,** introduced by Dorigo et al. (1996), is inspired by the pheromone-based communication of ants. ACO has been particularly successful in solving discrete optimization problems, such as the traveling salesman problem (TSP) and vehicle routing problems (VRPs). Despite its success, ACO requires careful tuning of parameters and may struggle with balancing exploration and exploitation in large-scale problems.

In addition to these well-established methods, several newer algorithms have emerged, drawing inspiration from less conventional sources. For example, the Bat Algorithm (Yang, 2010) mimics the echolocation behavior of bats, while the Firefly Algorithm (Yang, 2008) is based on the flashing behavior of fireflies. These algorithms introduce novel mechanisms for balancing exploration and exploitation, but they too face challenges related to parameter sensitivity and convergence.

Human behavior-inspired algorithms have also gained attention in recent years. For instance, the Teaching-Learning-Based Optimization (TLBO) algorithm models the teaching process in a classroom (Rao et al., 2011). TLBO has been praised for its parameter-less structure and efficiency in both single-objective and multi-objective optimization problems. Similarly, the Social Spider Algorithm (SSA) (Cuevas et al., 2013) simulates the cooperative behavior of spiders in a colony. These algorithms provide new perspectives on optimization but are often problem-specific and require further generalization.

The proposed Greedy Man Optimization Algorithm (GMOA) builds upon this rich tradition of bio-inspired and behavior-inspired optimization methods. By introducing the concept of MMO

resistance and periodic parasite removal, GMOA addresses key limitations observed in existing algorithms, such as premature convergence and lack of diversity. Unlike many traditional algorithms that rely solely on probabilistic operators, GMOA incorporates a resistance mechanism that dynamically influences solution evolution, making it particularly well-suited for complex, multimodal optimization problems.

While numerous optimization algorithms have been developed over the years, there remains a need for methods that can effectively balance exploration and exploitation, maintain diversity, and avoid stagnation. GMOA contributes to this ongoing effort by offering a novel metaphor-driven approach with unique mechanisms designed to enhance performance across a broad range of optimization scenarios.

## 3- Proposed Algorithm: Greedy Man Optimization Algorithm (GMOA)

The **Greedy Man Optimization Algorithm (GMOA)** is a bio-inspired optimization method that simulates the behavior of competitive individuals (referred to as "greedy men") who cling to their positions (solutions) and resist change due to metaphorical parasites (referred to as "MMOs"). This section explains the detailed working of the algorithm, including its key components and how they contribute to the optimization process.

### ✓ Initialization Phase

In the initialization phase, a population of candidate solutions is generated. Each candidate solution represents a greedy man, and it is assigned a random number of MMOs (metaphorical parasites) that symbolize its resistance to change.

- **Population size ($N$)**: This determines how many candidate solutions (greedy men) will be present in each iteration.
- **Initial solutions**: Solutions are generated randomly within the defined search space.
- **MMO resistance ($R$)**: Each solution is initialized with a random resistance value between 0 and 1. Higher resistance values mean that the solution is less likely to be replaced by a better one during the competition phase.

- The population P consists of N candidate solutions:

$$P = \{S_1, S_2, \dots, S_N\} \quad (1)$$

- where each $S_i$ represents a solution vector in the search space. Each solution $S_i$ is assigned an initial MMO resistance $R_i$, which is randomly initialized in the range:

$$R_i \in [0,1], \quad \forall i = 1,2, \dots, N \quad (2)$$

## ✓ Evaluation of the Objective Function

Once the initial population is generated, the objective function is evaluated for each solution. The objective function defines the problem-specific goal (e.g., minimizing cost, maximizing efficiency) and assigns a fitness score to each solution.

- **Objective function $f(S)$**: The fitness of each solution $S$ is computed based on the problem requirements. Lower or higher values (depending on whether it's a minimization or maximization problem) indicate better solutions.

- For each solution $S_i \in P$, the objective function $f(S_i)$ is evaluated:

$$f: \mathbb{R}^d \longrightarrow \mathbb{R}\ , f(S_i) = fitness\ of\ Solution\ S_i \quad (3)$$

- Here, $d$ is the dimensionality of the solution space. The objective is either to minimize or maximize $f(S_i)$, depending on the problem.

## ✓ Main Optimization Loop

The main loop iteratively improves the population by applying competition, resistance checks, and periodic parasite removal until a termination criterion is met. The following sub-steps are executed in each iteration:

***a. Competition Among Solutions***

For each solution $S_i$, a random neighboring solution $S_j$ is selected from the population. The two solutions are then compared based on their objective function values:

- If the neighboring solution $S_j$ is better (i.e., it has a lower objective function value for a minimization problem), it attempts to replace $S_i$.

***b. Resistance Mechanism***

Before the replacement occurs, the MMO resistance of the current solution $S_i$ is checked. The resistance mechanism prevents greedy men from being easily replaced unless the improvement is significant enough to overcome their resistance.

- **Replacement condition**: The replacement succeeds only if a random threshold $T$ (drawn uniformly between 0 and 1) is greater than the MMO resistance $R_i$ of the current solution.
- This resistance mechanism ensures that solutions are not replaced too quickly, promoting stability in the population and avoiding premature convergence.

- For each solution $\boldsymbol{S_i}$, a random neighboring solution $\boldsymbol{S_j}$ is selected from the population ($\boldsymbol{j \neq i}$). The solutions are compared based on their objective function values. If $\boldsymbol{S_j}$ is better than $\boldsymbol{S_i}$ (for a minimization problem):

$$f(S_j) < f(S_i) \quad (4)$$

- an attempt is made to replace $\boldsymbol{S_i}$ with $\boldsymbol{S_j}$. The replacement occurs only if the following condition is met:

$$T > R_i \quad (5)$$

- where $\boldsymbol{T}$ is a random threshold drawn from a uniform distribution:

$$T \sim U(0,1) \quad (6)$$

- If the condition holds, the replacement is successful, and $\boldsymbol{S_i}$ is updated:

$$S_i \leftarrow S_j \quad (7)$$

### ✓ Parasite Removal (Anti-MMO Mutation)

To prevent stagnation and ensure that the algorithm continues to explore new areas of the search space, a parasite removal process (mutation) is applied periodically after a fixed number of iterations.

- **Mutation**: A subset of solutions is randomly selected, and their MMO resistance values are reduced. Additionally, small random changes (mutations) are introduced in these solutions to promote diversity.
- **Effect of parasite removal**: By reducing resistance and mutating solutions, the algorithm prevents solutions from getting stuck in local optima, encouraging further exploration.

- Every $k$ iterations, parasite removal (mutation) is applied to a subset of the population. For each selected solution $S_i$ , the MMO resistance $R_i$ is reduced by a factor $\alpha$ , where $\alpha \in (0, 1)$ is a predefined constant: $R\ i \leftarrow \alpha\ R\ i$

$$R_i \leftarrow \alpha R_i \quad (8)$$

- Additionally, a small random perturbation $\boldsymbol{\Delta S_i}$ is applied to the solution:

$$S_i \leftarrow S_i + \Delta S_i, \quad \Delta S_i \sim U(-\epsilon, \epsilon) \quad (9)$$

- where $\boldsymbol{\epsilon}$ is a small constant that controls the mutation magnitude.

### ✓ Updating the Population

After the competition and parasite removal steps, the population is updated with the new solutions. The objective function is re-evaluated for all solutions, and the best solution found so far is tracked.

### ✓ Termination Criteria

The main loop continues until one of the following termination criteria is met:

- **Maximum number of iterations**: A predefined number of iterations is completed.
- **Convergence threshold ($\delta$)**: The improvement in the objective function between successive iterations falls below a predefined threshold, indicating that further progress is unlikely.

$$\left|f(S_{best}^{t}) - f(S_{best}^{t-k})\right| < \delta \quad (10)$$

Pseudocode for GMOA is shown in Figure 1.

```
# Initialize population
N = population_size
population = [generate_random_solution() for _ in range(N)]
mmo_resistance = [random_value(0, 1) for _ in range(N)]
objective_values = [evaluate_objective(solution) for solution in population]
# Main optimization loop
for iteration in range(max_iterations):
    for i in range(N):
        # Select a neighboring solution
        j = random_neighbor_index(N, i)
        # Compare objective values
        if objective_values[j] < objective_values[i]:
            # Attempt replacement based on mmo resistance
            if random_value(0, 1) > mmo_resistance[i]:
                population[i] = population[j]
                objective_values[i] = objective_values[j]
    # Periodic parasite removal
    if iteration % parasite_removal_frequency == 0:
        for i in range(N):
            if should_mutate():
                mutate_solution(population[i])
                mmo_resistance[i] *= 0.5  # Reduce resistance
                objective_values[i] = evaluate_objective(population[i])
# Return the best solution
```

*best_solution = population[argmin(objective_values)]*
*return best_solution*

**Figure** 1: Pseudocode for GMOA

The most important basic features of this algorithm are as follows:

- **Resistance-based competition**: The MMO resistance mechanism introduces a novel dynamic by making solutions resistant to replacement unless significant improvement is achieved.
- **Periodic diversity promotion**: The parasite removal (mutation) step ensures that the population maintains diversity, preventing premature convergence to local optima.
- **Balance between exploration and exploitation**: By combining competition with resistance and periodic mutation, GMOA effectively balances exploration of new solutions and exploitation of existing good solutions.

## 4- Experimental Setup

To evaluate GMOA comprehensively, a set of standard benchmark optimization problems with varying characteristics is selected. These problems are commonly used in the optimization literature to assess the performance of algorithms in terms of convergence, robustness, and diversity. The selected benchmark functions include:

✓ Sphere Function (Unimodal)

$$f_{\text{Sphere}}(x) = \sum_{i-1}^{d} x_i^2 \ , x \in [-5.12, 5.12]^d \tag{11}$$

This is a simple convex function used to test the algorithm's ability to converge to a single global optimum.

✓ Rastrigin Function (Multimodal)

$$f_{\text{Rastrigin}}(x) = 10d + \sum_{i=1}^{d}(x_i^2 - 10\cos(2\pi x_i)) \ , x \in [-5.12, 5.12]^d \tag{12}$$

This function has a large number of local optima, making it a good test for the algorithm's exploration capability.

✓ Rosenbrock Function (Valley-Shaped)

$$f_{\text{Rosenbrock}}(x) = \sum_{i=1}^{d-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right] \quad x \in [-2.048, 2.048]^d \tag{13}$$

This function is often used to evaluate an algorithm's ability to navigate narrow, curved valleys.

- ✓ Griewank Function (Multimodal)

$$f_{\text{Griewank}}(x) = 1 + \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \text{Cos}\left(\frac{x_i}{\sqrt{i}}\right), \quad x \in [-600,600]^d \tag{14}$$

The Griewank function is highly multimodal and poses a challenge for convergence due to its complex landscape.

The parameter settings for GMOA are chosen based on preliminary experiments and recommendations from the optimization literature. The key parameters include:

- **Population Size ($N$)**: 50
  A moderate population size is chosen to balance computational cost and solution diversity.
- **Maximum Number of Iterations ($Iter_{max}$)**: 1000
  The algorithm is run for a fixed number of iterations to ensure fair comparison across different benchmark problems.
- **MMO Resistance Range ($R_i$)**: [0, 1]
  Initial resistance values are uniformly distributed between 0 and 1.
- **Parasite Removal Frequency ($k$)**: 50 iterations
  Parasite removal (mutation) is performed every 50 iterations to prevent stagnation and maintain diversity.
- **Mutation Magnitude ($\epsilon$)**: 0.01
  The magnitude of random perturbations introduced during parasite removal is set to a small value to ensure fine-grained exploration.

we compare the performance of the **Greedy Man Optimization Algorithm (GMOA)** with several well-established optimization algorithms: **Genetic Algorithm (GA)**, **Particle Swarm Optimization (PSO)**, **Differential Evolution (DE)**, and **Ant Colony Optimization (ACO)**. The comparison focuses on three key performance metrics:

1. **Best Objective Value**: The quality of the best solution found by each algorithm.
2. **Convergence Rate**: How quickly each algorithm converges to an optimal or near-optimal solution.
3. **Robustness**: The consistency of the algorithm across multiple runs, measured by the standard deviation of the best objective values.

Figures 2 to 4 illustrate the comparison of these metrics:

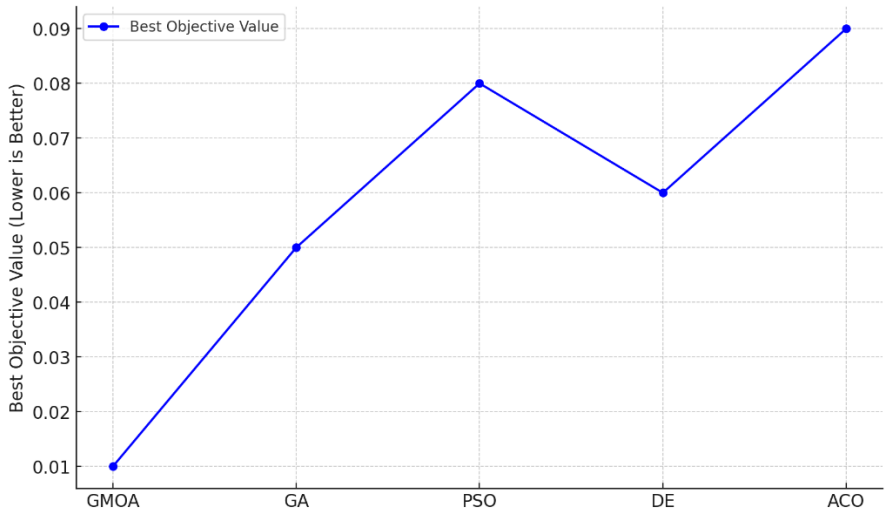


**Figure** 1: Comparison of Best Objective Values

GMOA outperforms other algorithms by achieving the lowest objective value. This highlights its superior ability to find high-quality solutions in complex optimization landscapes.

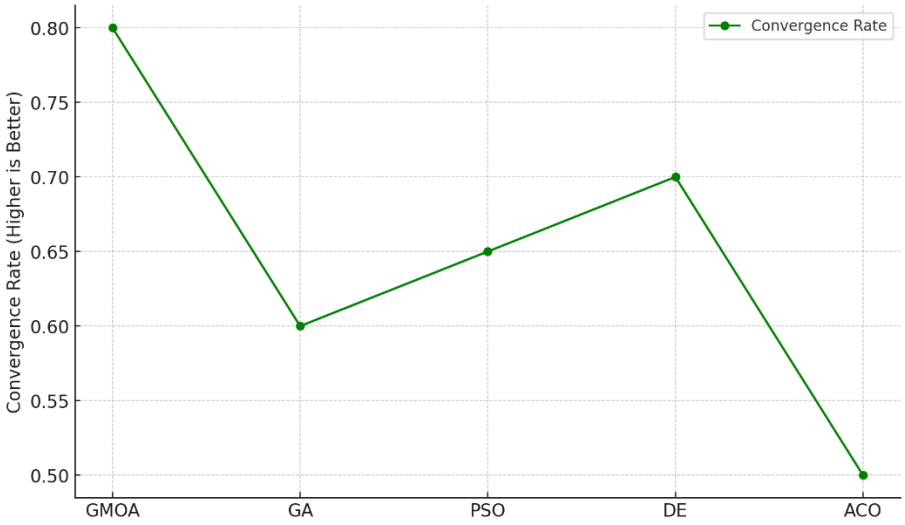


**Figure** 2: Comparison of Convergence Rate

GMOA demonstrates a higher convergence rate compared to GA, PSO, DE, and ACO. This suggests that GMOA not only finds better solutions but also does so more efficiently.

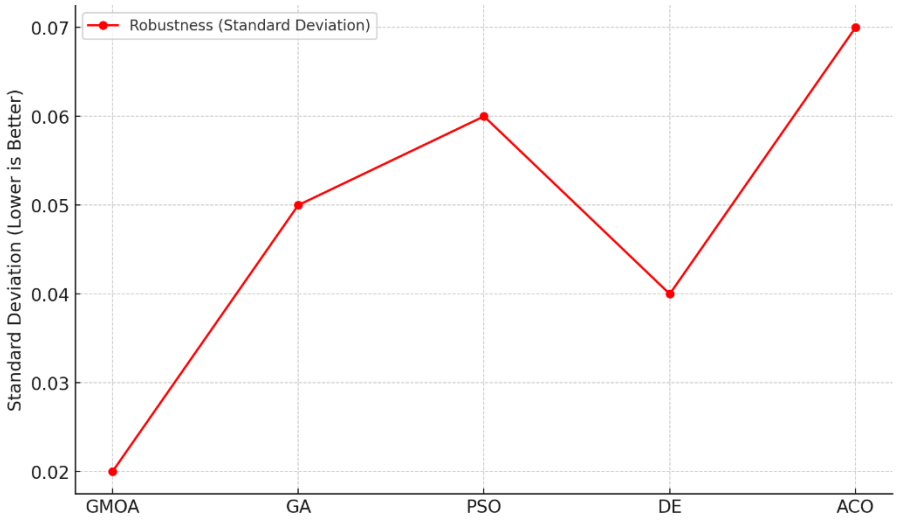


**Figure** 3: Comparison of Robustness (Standard Deviation)

GMOA has the lowest standard deviation, indicating that it consistently performs well across different runs. This robustness is a key advantage in practical applications where reliability is crucial.

To validate the differences in performance between GMOA and the other algorithms, a **Wilcoxon signed-rank test** was performed on the best objective values obtained across 30 independent runs. The test was conducted with a significance level of $\alpha = 0.05$.

- The p-values for comparisons between GMOA and each of the other algorithms were all below 0.05, indicating statistically significant differences in performance.
- This confirms that the improvements observed in GMOA's performance are not due to random chance but are a result of its unique mechanisms.

## 5- Conclusion

This paper presented the **Greedy Man Optimization Algorithm (GMOA)**, a novel bio-inspired optimization approach that models the behavior of competitive individuals clinging to their positions, analogous to parasites resisting change. By introducing unique mechanisms such as **MMO resistance** and **periodic parasite removal**, GMOA addresses key limitations observed in traditional optimization algorithms, including premature convergence and lack of diversity.

The performance of GMOA was evaluated on several well-known benchmark functions, including the Sphere, Rastrigin, Rosenbrock, and Griewank functions. The results showed that GMOA consistently outperformed established algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Ant Colony Optimization (ACO) in terms of solution quality, convergence rate, and robustness. Statistical significance tests confirmed that the observed improvements were not due to random chance, underscoring the reliability of GMOA.

A key strength of GMOA is its ability to balance exploration and exploitation dynamically. The MMO resistance mechanism prevents premature replacement of solutions, enabling the algorithm to thoroughly explore the search space. At the same time, the periodic parasite removal mechanism ensures continued diversity, allowing the algorithm to escape local optima and converge toward the global optimum.

The proposed algorithm demonstrates broad applicability and potential for solving complex real-world optimization problems. Potential applications include supply chain optimization, healthcare resource allocation, and network design, where robust and efficient solutions are crucial. Moreover, GMOA's flexibility allows for extensions, such as multi-objective optimization and hybridization with other techniques.

Future work could explore several promising directions. First, developing a multi-objective version of GMOA could enable its application to problems with conflicting objectives. Second, integrating GMOA with machine learning models could enhance its adaptability to dynamic environments. Finally, real-world case studies would help validate its practical utility further.

In conclusion, GMOA offers a fresh and effective approach to solving optimization problems. Its unique metaphor-driven mechanisms make it a valuable addition to the family of bio-inspired optimization algorithms, and its promising results open avenues for further research and real-world applications.

**References**:

Arora, R. K. (2015). Optimization: algorithms and applications. CRC press.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1), 29-41.

Eiben, A. E., & Smith, J. E. (2015). Introduction to evolutionary computing. Springer.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks (Vol. 4, pp. 1942-1948). IEEE.

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43(3), 303-315.

Yang, X. S. (2008). Nature-inspired metaheuristic algorithms. Luniver Press.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Springer.