Control Systems and Optimization Letters, Vol. 3, No 3, 2025
ISSN: 2985-6116, DOI: 10.59247/csol.v3i3.214

Human Breakfast Selection Algorithm (HBSA): A Human-Inspired Metaheuristic for Constrained Optimization

Jincheng Zhang [1*], Jindong Zhang [2]
[1] Faculty of Science and Technology, Rajabhat Maha Sarakham University, Maha Sarakham, Thailand
[2] Independent Researcher, Putian, Fujian, China
Email: [1] zjc1639834588@gmail.com, 172825661@qq.com
*Corresponding Author

***Abstract*—In this paper, we propose a new metaheuristic algorithm inspired by human daily breakfast choice behavior, namely the human breakfast choice algorithm (HBSA). When deciding what to eat for breakfast, people often consider multiple goals, constraints, and personal preferences. The algorithm simulates the memory mechanism, preference guidance, contextual adaptation, and hybrid decision-making strategies of human breakfast choices to achieve more effective exploration capabilities in solving combinatorial optimization problems. We apply the algorithm to a typical 0-1 knapsack problem and conduct comparative experiments with genetic algorithms (GA) and particle swarm optimization algorithms (PSO). The results show that the improved HBSA performs better in terms of solution quality and stability.**

Keywords*—Breakfast Decision-Making; Multi-Objective Optimization; Human Behavior Simulation; HBSA Algorithm; Metaheuristic Algorithms

I. INTRODUCTION

In modern society, breakfast is not only an important meal of the day, but also a daily, complex and diverse decision-making behavior. Every morning, people must decide what to eat for breakfast, taking into account busy schedules, limited food resources, health considerations and personal preferences. This seemingly simple action actually involves a lot of information processing, evaluating advantages and disadvantages, and screening conditions [1]-[4]. For example, an office worker who is in a hurry may choose quick and simple bread, while a health-conscious fitness enthusiast may prefer a high-protein, low-sugar combination of cereals and fruits. Further complicating factors include budget constraints (breakfast cannot be too expensive), taste fatigue (I had eggs yesterday and don't want to eat them today), and external environmental factors (it's cold, so I prefer hot food), all of which affect people's thinking and behavior patterns when choosing breakfast.

From an optimization perspective, this daily behavior is essentially a classic multi-objective, multi-constrained combinatorial optimization problem. Objectives usually include maximizing nutritional value, minimizing time and optimizing taste satisfaction. At the same time, multiple constraints must be met, including food supply, budget constraints, time constraints, etc. Different people have different selection strategies and behavioral habits due to life experiences, physical conditions, cultural backgrounds, environmental changes, etc. The flexibility and adaptability shown by humans in breakfast choices are an inspiration to all of us. Is it possible to systematically abstract the mechanisms of memory, preference, constraint identification, and exploration-exploitation trade-offs involved in human breakfast choices and incorporate them into new general meta-heuristic optimization algorithms?

Based on this idea, we propose human breakfast choice algorithm (HBSA), a meta-heuristic optimization algorithm inspired by human breakfast choice behavior. The algorithm does not rely on physical or biological phenomena in nature, and simulates the behavioral characteristics shown by humans when facing complex decision-making problems, including memory-driven, preference-guided, situational adaptive, and hybrid decision-making. Our goal is to develop adaptive, robust, and efficient intelligent algorithms for various combinatorial optimization scenarios, and explore the potential value of behavior-driven optimization models in complex real-world problems.

At first glance, choosing breakfast may seem trivial. However, from the perspective of decision theory and combinatorial optimization, it is a rich and structured metaphor for complex problem solving. Breakfast selection begins with selecting a subset of available options (foods) and optimizing multiple, often conflicting objectives, such as taste satisfaction, health value, and cooking time. At the same time, multiple constraints such as time, nutritional requirements, cost, and availability must be met. This naturally maps to a typical multi-objective, multi-constrained optimization problem.

- Item/Decision: Individual ingredients (e.g., bread, milk, fruit). Each ingredient has associated attributes (value and cost).
- Constraints: Available time, calorie restrictions, dietary restrictions, or availability of ingredients.
- Goal: Maximize satisfaction (subjective utility), nutritional value, or minimize cooking time.
- Search Space: All possible combinations of breakfast ingredients.
- Behavioral Dynamics: Changing preferences and experiential learning reflect adaptive heuristics. Moreover, breakfast decisions are repeated daily under changing conditions and retain a partial memory of past outcomes, resulting in a dynamic, feedback-driven decision environment. These properties make the model an ideal abstract model for studying adaptive, experience-

based situational optimization, and it is well suited for many real-world problems that require flexible heuristics under uncertainty.

While many existing behavioral metaheuristics (e.g., decision-based metaheuristics and cognitive metaheuristics) incorporate human-like features, HBSA stands out in that it explicitly models the dynamic cognitive flexibility inherent in human breakfast decision-making. Unlike static or natural heuristics, HBSA integrates multiple aspects of human behavior (e.g., memory-based avoidance of choice fatigue, adaptive preference adjustment, context-dependent decision-making, and the delicate balance between exploration and exploitation) into a unified framework. This enables HBSA to effectively capture the real-time adaptability and experiential learning characteristics of human cognition, providing a new perspective for improving search efficiency and solution quality.

This study focuses on simulating the dynamics of human cognition in multi-constraint, multi-objective scenarios, distinguishing HBSA from other behavioral heuristics and contributing to the development of more "human" and flexible metaheuristics. We believe that this innovation will open up new avenues for designing intelligent optimization algorithms to solve complex and rapidly changing real-world problems.

II. Related Work

In the past few decades, meta-heuristic algorithms have made great progress as an important tool for solving complex optimization problems. Representative methods include genetic algorithms (GA), particle swarm optimization (PSO), simulated annealing (SA), ant colony optimization (ACO), and differential evolution (DE). Most of these methods perform optimization search by simulating certain physical processes, biological mechanisms, or social behaviors in nature. Due to its powerful global search capability and parameter flexibility, it is widely used in engineering design, path planning, graph optimization, parameter adjustment and scheduling problems [5]-[17].

Although existing metaheuristic methods have achieved great success, most of them focus on imitating natural phenomena. For example, GA is based on the selection, crossover and mutation mechanisms in biological evolution. PSO simulates the cooperative behavior of bird flocks or fish schools. SA originates from the annealing process in physics. Although these methods have certain theoretical versatility, their operating models are relatively static and cannot directly simulate "human cognitive and behavioral patterns". In fact, humans often show strong dynamic adaptability, memory dependence and environmental responsiveness when facing decision-making problems. For example, they may avoid unsuccessful choices based on past experience, explore new and slightly more novel choices through repeated trials, or quickly adjust strategies according to sudden changes in the environment (such as time pressure or food shortage). These cognitive behavioral characteristics have not been systematically incorporated into the current mainstream metaheuristic algorithms [18]-[30].

In recent years, research on constructing algorithm models from the perspectives of social behavior, psychological mechanisms, human cognition, etc. has begun to attract attention, such as "whale optimization algorithm" and "bee swarm algorithm". Although they introduce more human characteristics, they still cannot cope with the high-frequency, complex and multi-constrained decision-making process in human daily life. Breakfast selection, as a behavioral pattern with the characteristics of "multi-objective, multi-constraint, and strong experience-driven", provides an ideal prototype for building optimization algorithms driven by human behavior.

In this context, we first proposed to use "human breakfast selection behavior" as the inspiration source of the algorithm and developed the HBSA algorithm model. This method combines memory mechanism (avoiding local traps and fatigue), preference adjustment mechanism (simulating individual adaptability and enhancing selection), context perception mechanism (simulating the influence of external constraints) and exploration-exploitation balance mechanism (controlling the switch between global search and local optimization). Local search and memory update strategies are also introduced to further improve search efficiency and convergence quality. We believe that this optimization method based on human behavior model will not only bring new vitality to the research of metaheuristic algorithms, but also provide a more "humanized" and "intelligent" strategic framework for solving practical complex optimization problems.

III. HBSA Algorithm Design Principle

A. Inspiration

When choosing breakfast, humans show the following decision-making behaviors:

- Memory driven: avoid repeatedly choosing the same food.
- Preference: Tend to choose foods that you have eaten frequently in the past.
- Context: Adjust choices based on dynamic conditions such as time of day, weather, and materials.
- Exploration and exploitation: Find a balance between familiar choices and trying new things.

B. Mapping Operation Mechanism

Mechanism of human behavior algorithm. Avoid repeated choices, as repeated choices will cause "choice fatigue" and reduce your success rate. Adjust choices based on preferences Dynamically adjust preference distribution (a concept of reinforcement learning). Added context-aware filtering based on weather/time/material constraints. Try new foods from time to time and introduce random perturbations to improve exploration.

The pseudocode of human breakfast selection algorithm (HBSA) is as follows:

```
Input:
MaxIter     // Maximum number of iterations
PopSize     // Population size
Dim         // Dimension of solution vector
Fitness()   // Fitness evaluation function
ContextConstraints    // External constraints (time, weather, materials, etc.)

Initialize:
Population ← randomly generated PopSize solutions of dimension Dim
PersonalBest ← copy of Population
PersonalBestFitness ← evaluate fitness of each individual in Population
GlobalBest ← individual with best fitness in Population
```

```
GlobalBestFitness ← best fitness value in Population
Memory ← empty set to record previous selections (avoid fatigue)
PerturbProb ← initial perturbation probability

For iter = 1 to MaxIter do:

FitnessValues ← evaluate fitness of all individuals in Population

  // Update personal bests
  For each individual i in Population do
    If FitnessValues[i] > PersonalBestFitness[i] then
      PersonalBest[i] ← Population[i]
      PersonalBestFitness[i] ← FitnessValues[i]
    EndIf
  EndFor

  // Update global best
  currentBestIndex ← index of max FitnessValues
  If FitnessValues[currentBestIndex] > GlobalBestFitness then
    GlobalBest ← Population[currentBestIndex]
    GlobalBestFitness ← FitnessValues[currentBestIndex]
  EndIf

  // Adaptive perturbation probability
  If no improvement in GlobalBestFitness over several iterations then
    Increase PerturbProb to encourage exploration
  Else
    Decrease or reset PerturbProb to balance exploration/exploitation
  EndIf

  // Update each individual in Population
  For each individual i in Population do

    // Memory-driven avoidance of repeated selections
    If Population[i] is similar to any solution in Memory then
      Apply local perturbation to Population[i] to avoid fatigue
    EndIf

    // Preference adjustment mechanism
    Adjust individual's selection probabilities based on historical
success

    // Context-aware filtering
    Modify or filter Population[i] to satisfy ContextConstraints

    // Exploration vs exploitation balance
    With probability PerturbProb:
      Apply random perturbation to Population[i] (exploration)
    Else:
      Guide update using PersonalBest[i] or GlobalBest (exploitation)

  EndFor

  // Local search around the current GlobalBest
  For each neighbor of GlobalBest do
    If neighbor's fitness > GlobalBestFitness then
      GlobalBest ← neighbor
      GlobalBestFitness ← neighbor's fitness
    EndIf
  EndFor

  // Update Memory with current GlobalBest to prevent future fatigue

  Output iteration number, GlobalBestFitness

EndFor

Return GlobalBest, GlobalBestFitness
```

IV. Improved Implementation of HBSA Algorithm

Based on the above ideas, we implemented an improved version of HBSA and added the following mechanisms:

- Individual memory mechanism: record individual optimal solutions and update the global optimal solution.
- Adaptive perturbation probability: If the population evolution stagnates, increase the perturbation probability to increase search diversity.
- Local search mechanism: Perform neighborhood perturbations (bit flips) on the current best solution to fine-tune the search results.
- Guided update mechanism: Learn how to update PSO "based on individual experience" to improve convergence speed.

The specific Python implementation code is as follows:

```
import numpy as np

# Fitness function
def fitness(solution, values, weights, capacity):
    total_value = np.sum(solution * values)
    total_weight = np.sum(solution * weights)
    return total_value if total_weight <= capacity else 0

# Improved HBSA with local search and memory mechanism
def improved_HBSA(num_iter=100, pop_size=50, values=None,
weights=None, capacity=50):
    dim = len(values)
    population = np.random.randint(2, size=(pop_size, dim))
    personal_best = population.copy()
    personal_best_fitness = np.array([fitness(ind, values, weights,
capacity) for ind in population])

    best_solution =
personal_best[np.argmax(personal_best_fitness)].copy()
    best_fitness = np.max(personal_best_fitness)

    for generation in range(num_iter):
        fitness_values = np.array([fitness(ind, values, weights,
capacity) for ind in population])

        # Update individual best history
        for i in range(pop_size):
            if fitness_values[i] > personal_best_fitness[i]:
                personal_best[i] = population[i].copy()
                personal_best_fitness[i] = fitness_values[i]

        # Global best
        current_best_index = np.argmax(fitness_values)
        if fitness_values[current_best_index] > best_fitness:
            best_fitness = fitness_values[current_best_index]
            best_solution = population[current_best_index].copy()

        # Adaptive update probability
        if generation > 0 and fitness_values[current_best_index] ==
best_fitness:
            update_prob = 0.3  # Increase mutation rate if stagnant
        else:
            update_prob = 0.2

        # Update population
        for i in range(pop_size):
            if np.random.rand() < update_prob:
                flip_index = np.random.randint(dim)
                population[i][flip_index] = 1 -
population[i][flip_index]

            # Guided update using personal best (PSO-like)
            rand_idx = np.random.randint(dim)
            population[i][rand_idx] = personal_best[i][rand_idx]

        # Local search: flip each bit of current best to try
improvement
        local_best = best_solution.copy()
        for j in range(dim):
            neighbor = best_solution.copy()
            neighbor[j] = 1 - neighbor[j]
```

```
            if fitness(neighbor, values, weights, capacity) > fitness(local_best, values, weights, capacity):
                local_best = neighbor.copy()
            best_solution = local_best
            best_fitness = fitness(best_solution, values, weights, capacity)

            print(f"Improved HBSA Generation {generation + 1}/{num_iter}, Best fitness: {best_fitness}")

        return best_solution, best_fitness

    # Simple GA implementation
    def GA(num_iter=100, pop_size=50, values=None, weights=None, capacity=50):
        dim = len(values)
        population = np.random.randint(2, size=(pop_size, dim))
        best_solution = None
        best_fitness = -1

        for generation in range(num_iter):
            fitness_values = np.array([fitness(ind, values, weights, capacity) for ind in population])
            probs = fitness_values / (sum(fitness_values) + 1e-6)
            selected = population[np.random.choice(range(pop_size), size=pop_size, p=probs)]

            # Crossover
            next_gen = []
            for i in range(0, pop_size, 2):
                if i + 1 < pop_size:
                    point = np.random.randint(1, dim)
                    next_gen.append(np.concatenate([selected[i][:point], selected[i + 1][point:]]))
                    next_gen.append(np.concatenate([selected[i + 1][:point], selected[i][point:]]))

            population = np.array(next_gen)

            # Mutation
            for i in range(pop_size):
                if np.random.rand() < 0.1:
                    m_point = np.random.randint(dim)
                    population[i][m_point] = 1 - population[i][m_point]

            gen_best_fitness = np.max(fitness_values)
            if gen_best_fitness > best_fitness:
                best_fitness = gen_best_fitness
                best_solution = population[np.argmax(fitness_values)]

            print(f"GA Generation {generation + 1}/{num_iter}, Best fitness: {best_fitness}")

        return best_solution, best_fitness

    # PSO implementation
    def PSO(num_iter=100, pop_size=50, values=None, weights=None, capacity=50):
        dim = len(values)
        population = np.random.randint(2, size=(pop_size, dim))
        velocities = np.random.randn(pop_size, dim)
        personal_best = population.copy()
        personal_best_fitness = np.array([fitness(ind, values, weights, capacity) for ind in population])
        global_best = personal_best[np.argmax(personal_best_fitness)].copy()
        global_best_fitness = np.max(personal_best_fitness)

        for generation in range(num_iter):
            for i in range(pop_size):
                fitness_val = fitness(population[i], values, weights, capacity)
                if fitness_val > personal_best_fitness[i]:
                    personal_best[i] = population[i].copy()
                    personal_best_fitness[i] = fitness_val
                if fitness_val > global_best_fitness:
                    global_best = population[i].copy()
```

```
                    global_best_fitness = fitness_val

                # Update velocity and position
                r1, r2 = np.random.rand(), np.random.rand()
                velocities = 0.5 * velocities + 1.5 * r1 * (personal_best - population) + 1.5 * r2 * (global_best - population)
                population = (population + np.sign(velocities)).clip(0, 1)

                print(f"PSO Generation {generation + 1}/{num_iter}, Best fitness: {global_best_fitness}")

            return global_best, global_best_fitness

        # Compare algorithm performance
        def compare_algorithms(num_iter=100, pop_size=50, num_items=10, values=None, weights=None, capacity=50, num_experiments=10):
            improved_hbsa_fits, ga_fits, pso_fits = [], [], []

            for _ in range(num_experiments):
                print("\nRunning Improved HBSA...")
                _, fit = improved_HBSA(num_iter, pop_size, values, weights, capacity)
                improved_hbsa_fits.append(fit)

                print("\nRunning GA...")
                _, fit = GA(num_iter, pop_size, values, weights, capacity)
                ga_fits.append(fit)

                print("\nRunning PSO...")
                _, fit = PSO(num_iter, pop_size, values, weights, capacity)
                pso_fits.append(fit)

            print("\n=== Average Fitness Comparison ===")
            print(f"Improved HBSA Average Fitness: {np.mean(improved_hbsa_fits):.2f}")
            print(f"GA Average Fitness:     {np.mean(ga_fits):.2f}")
            print(f"PSO Average Fitness:    {np.mean(pso_fits):.2f}")

        # Set parameters and run
        if __name__ == "__main__":
            np.random.seed(42)
            values = np.random.randint(10, 100, size=10)
            weights = np.random.randint(1, 50, size=10)
            capacity = 100

            compare_algorithms(
                num_iter=100,
                pop_size=50,
                num_items=10,
                values=values,
                weights=weights,
                capacity=capacity,
                num_experiments=10
            )
```

V. Comparative Experiment with Conventional Algorithm

A. *Experimental Setup*

- Problem type: 0-1 knapsack problem
- Number of products: 10
- Weight limit: 100
- Number of tests: 10
- Number of iterations per round: 100
- Population size: 50

B. *Experimental Results (Average Of 10 Rounds)*

- Average fitness of the algorithm
- Improvement of HBSA 438.70
- Genetic algorithm (GA) 438.40
- Particle swarm optimization (PSO) 418.40

The results show that the average fitness of HBSA is slightly higher than that of GA and significantly better than that of PSO. This shows that it has good stability and exploration ability in combinatorial optimization tasks.

VI. Algorithm Advantages and Applicability Analysis

A. Algorithm Overview

The human breakfast selection algorithm (HBSA) is a metaheuristic algorithm that simulates the human breakfast selection process. The algorithm solves the optimization problem by imitating the multi-objective decision-making behavior of humans when choosing breakfast (such as nutrition, time, budget, etc.). By combining the four core mechanisms of memory mechanism, preference guidance, external condition response, and hybrid decision strategy, the optimal choice under multi-objective and multi-constraint conditions is achieved.

B. Algorithm Advantages

- Simulating human decision-making behavior: The HBSA algorithm is inspired by the multi-objective decision-making of human breakfast selection and can better simulate the multi-dimensional decision-making process in practical problems. When faced with multiple constraints (time, budget, health, etc.), HBSA provides reasonable decision results by balancing these factors.
- High flexibility: The algorithm is not only suitable for optimizing specific problems, but also has wide applicability to other complex optimization tasks. Its versatility is reflected in the adjustable parameters in the algorithm design, such as preference guidance mechanism, external condition response mechanism, etc., which can be flexibly adjusted to meet various practical needs.
- Adaptation mechanism: HBSA dynamically adjusts decision-making strategies based on past choices through "preference guidance" and "memory-driven" mechanisms to avoid falling into local optimality. This allows you to not only make the best decisions under fixed conditions, but also adapt to different situations in a changing environment.
- Multi-objective optimization: By comprehensively considering multiple objectives (such as nutrition, speed, and preferences), the algorithm can better meet the needs of multi-objective optimization in real-world decision-making. This enables HBSA to balance multiple aspects at the same time when dealing with complex problems and avoid the deviations that may be caused by single-objective optimization.
- Enhanced balance between exploration and utilization: By combining the exploration and utilization balance mechanism, HBSA finds the best balance between exploring new options and leveraging existing experience, avoiding making a single choice due to over-reliance on past decisions.

C. Algorithm Applicability Analysis

- Healthy Diet Zone: HBSA is particularly suitable for application in healthy diet recommendation systems, which take into account the health needs of users and constraints such as time and budget, and can provide personalized dietary recommendations based on user preferences and environmental conditions.
- Intelligent Supply Chain Management: In the fields of material management and resource scheduling, HBSA can be used to optimize product selection, delivery routes, etc. It can efficiently find the optimal solution, especially in the context of multi-objective optimization and complex constraints.
- Personalized Recommendation System: The HBSA algorithm can be applied to personalized recommendation systems, dynamically adjust user preference models, combine historical behavior data with current situations, and provide personalized recommendation results, such as movie recommendations or product recommendations.
- Financial Investment Decision: In the field of financial investment, HBSA can simulate the decision-making process of investors and optimize investment portfolios by comprehensively considering multiple objectives such as risk, return, and market conditions.
- Automated Design and Optimization: In industrial design and engineering optimization, HBSA can solve complex design constraints, combine multiple decision-making strategies for automated design and optimization, and adapt to changing design requirements.

D. Conclusion

The HBSA algorithm shows strong adaptability and versatility by simulating the multi-objective decision-making process of human breakfast selection. HBSA can provide flexible and efficient solutions for optimization problems with multiple constraints and complex objectives, and has been widely used in many fields such as health management, personalized recommendation and supply chain management.

VII. Conclusion

When choosing breakfast, people face multiple trade-offs, including nutrition, time, budget and personal preferences. To simulate this complex decision-making process, we propose a novel meta-heuristic algorithm, namely the human breakfast selection algorithm (HBSA), which is designed to handle multi-objective and multi-constrained optimization problems. The algorithm combines memory mechanism, preference induction, situational adaptation and hybrid decision strategy to simulate human selection strategies in various situations, making it have broad application prospects.

A. Core idea

- Memory-based selection: By simulating human memory of "what to eat recently", we avoid frequently choosing the same option and solve the problem of "choice fatigue".
- Preference-guided sampling: Similar to the "preference update" mechanism in reinforcement learning, we dynamically adjust our selection strategy based on personal preferences and past feedback.
- Context-aware adaptation: It makes the algorithm more flexible and adaptable by dynamically adjusting the selection according to the external environment (weather, time, and material availability).

- Hybrid decision mode: By balancing random exploration and experience utilization, we simulated the human psychological state of "I want to try something new today".
- Improved HBSA: We improved HBSA by introducing local search and memory mechanisms, significantly improving search efficiency and solution quality. HBSA continuously updates the individual best history and global optimal solution, and gradually approaches the optimal solution through multi-generation iterative processing.

B. Comparison and performance

Compared with traditional genetic algorithm (GA) and particle swarm optimization algorithm (PSO), HBSA shows obvious advantages. Experimental results show that in multiple experiments, the average fitness of HBSA is higher than that of GA and PSO, and HBSA shows better convergence and stability, especially when dealing with complex multi-objective optimization problems.

C. Future challenges

As a general metaheuristic algorithm, HBSA may be extended to more practical problems in the future, such as complex optimization tasks such as path planning and resource scheduling. At the same time, further improving the efficiency and exploration ability of the algorithm and combining more environmental recognition mechanisms will be the future development direction of HBSA.

REFERENCES

[1] S. C. Donnelly, "Breakfast - The most important meal of the day!!," *QJM: An International Journal of Medicine*, vol. 111, no. 10, p. 681, 2018, https://doi.org/10.1093/qjmed/hcy201.

[2] M. J. Gibney *et al.*, "Breakfast in human nutrition: The international breakfast research initiative," *Nutrients*, vol. 10, no. 5, p. 559, 2018, https://doi.org/10.3390/nu10050559.

[3] S. E. Middlestadt, L. D. Stevenson, C.-L. Hung, M. L. Roditis, A. D. Fly, and J. L. Sheats, "Beliefs underlying the decision to eat breakfast: The role of theory-based behavioral analysis in the development of policy, communication and educational interventions for healthy eating," *Food studies*, vol. 1, no. 4, pp. 45–54, 2011, https://pmc.ncbi.nlm.nih.gov/articles/PMC3786227/

[4] S. O. Oladejo, S. O. Ekwe, and S. Mirjalili, "The Hiking Optimization Algorithm: A novel human-based metaheuristic approach," *Knowledge-Based Systems*, vol. 296, p. 111880, 2024, https://doi.org/10.1016/j.knosys.2024.111880.

[5] A. S. Desuky, M. A. Cifci, S. Kausar, S. Hussain, and L. M. E. Bakrawy, "Mud Ring Algorithm: A New Meta-Heuristic Optimization Algorithm for Solving Mathematical and Engineering Challenges," *IEEE Access*, vol. 10, pp. 50448–50466, 2022, https://doi.org/10.1109/ACCESS.2022.3173401.

[6] E. Pira, "City councils evolution: a socio-inspired metaheuristic optimization algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 9, pp. 12207–12256, 2023, https://doi.org/10.1007/s12652-022-03765-5.

[7] D. Ke, "Network Intrusion Detection Based on Feature Selection and Transformer," *Proceedings - 2023 International Conference on Intelligent Communication and Computer Engineering, ICICCE 2023*, vol. 74, no. 2, pp. 23–28, 2023, https://doi.org/10.1109/ICICCE61720.2023.00010.

[8] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, no. 3, pp. 1531–1551, 2021, https://doi.org/10.1007/s10489-020-01893-z.

[9] T. S. L. V. Ayyarao *et al.*, "War Strategy Optimization Algorithm: A New Effective Metaheuristic Algorithm for Global Optimization," *IEEE Access*, vol. 10, pp. 25073–25105, 2022, https://doi.org/10.1109/ACCESS.2022.3153493.

[10] T. Hamadneh *et al.*, "Sculptor Optimization Algorithm: A New Human-Inspired Metaheuristic Algorithm for Solving Optimization Problems," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 4, pp. 564–575, 2024, https://doi.org/10.22266/IJIES2024.0831.43.

[11] M. A. Elaziz *et al.*, "Advanced metaheuristic optimization techniques in applications of deep neural networks: a review," *Neural Computing and Applications*, vol. 33, no. 21, pp. 14079–14099, 2021, https://doi.org/10.1007/s00521-021-05960-5.

[12] Ş. Ay, E. Ekinci, and Z. Garip, "A comparative analysis of meta-heuristic optimization algorithms for feature selection on ML-based classification of heart-related diseases," *The Journal of Supercomputing*, vol. 79, no. 11, pp. 11797–11826, 2023, https://doi.org/10.1007/s11227-023-05132-3.

[13] S. Ghani and S. Kumari, "Liquefaction behavior of Indo-Gangetic region using novel metaheuristic optimization algorithms coupled with artificial neural network," *Natural Hazards*, vol. 111, no. 3, pp. 2995–3029, 2022, https://doi.org/10.1007/s11069-021-05165-y.

[14] O. O. Akinola, A. E. Ezugwu, J. O. Agushaka, R. A. Zitar, and L. Abualigah, "Multiclass feature selection with metaheuristic optimization algorithms: a review," *Neural Computing and Applications*, vol. 34, no. 22, pp. 19751–19790, 2022, https://doi.org/10.1007/s00521-022-07705-4.

[15] M. M. Eid, F. Alassery, A. Ibrahim, and M. Saber, "Metaheuristic Optimization Algorithm for Signals Classification of Electroencephalography Channels," *Computers, Materials and Continua*, vol. 71, no. 2, pp. 4627–4641, 2022, https://doi.org/10.32604/cmc.2022.024043.

[16] B. Abdollahzadeh *et al.*, "Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning," *Cluster Computing*, vol. 27, no. 4, pp. 5235–5283, 2024, https://doi.org/10.1007/s10586-023-04221-5.

[17] J. Zhang and J. Zhang, "Classical Dance-Metaheuristic: A Metaheuristic Optimization Algorithm Inspired by Classical Dance," *Control Systems and Optimization Letters*, vol. 3, no. 2, p. 2025, 2025, https://doi.org/10.59247/csol.v3i2.206.

[18] P. Sharma and S. Raju, "Metaheuristic optimization algorithms: a comprehensive overview and classification of benchmark test functions," *Soft Computing*, vol. 28, no. 4, pp. 3123–3186, 2024, https://doi.org/10.1007/s00500-023-09276-5.

[19] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. A. Al-qaness, and A. H. Gandomi, "Aquila Optimizer: A novel meta-heuristic optimization algorithm," *Computers & Industrial Engineering*, vol. 157, p. 107250, 2021, https://doi.org/10.1016/j.cie.2021.107250.

[20] O. N. Ovelade and A. E. Ezugwu, "Ebola Optimization Search Algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems," *International Conference on Electrical, Computer, and Energy Technologies, ICECET 2021*, vol. 10, pp. 16150–16177, 2021, https://doi.org/10.1109/ICECET52533.2021.9698813.

[21] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer," *Neural Computing and Applications*, vol. 35, no. 5, pp. 4099–4131, 2023, https://doi.org/10.1007/s00521-022-07854-6.

[22] R. Alkanhel, E. S. M. El-Kenawy, D. L. Elsheweikh, A. A. Abdelhamid, A. Ibrahim, and D. S. Khafaga, "Metaheuristic Optimization of Time Series Models for Predicting Networks Traffic," *Computers, Materials and Continua*, vol. 75, no. 1, pp. 427–442, 2023, https://doi.org/10.32604/cmc.2023.032885.

[23] L. Abualigah, K. H. Almotairi, and M. A. Elaziz, "Multilevel thresholding image segmentation using meta-heuristic optimization algorithms: comparative analysis, open challenges and new trends," *Applied Intelligence*, vol. 53, no. 10, pp. 11654–11704, 2023, https://doi.org/10.1007/s10489-022-04064-4.

[24] R. Cadario and C. K. Morewedge, "Why do people eat the same breakfast every day? Goals and circadian rhythms of variety seeking in meals," *Appetite*, vol. 168, p. 105716, 2022, https://doi.org/10.1016/j.appet.2021.105716.

[25] A. Daliri, A. Asghari, H. Azgomi, and M. Alimoradi, “The water optimization algorithm: a novel metaheuristic for solving optimization problems,” *Applied Intelligence*, vol. 52, no. 15, pp. 17990–18029, 2022, https://doi.org/10.1007/s10489-022-03397-4.

[26] I. Matoušová, P. Trojovský, M. Dehghani, E. Trojovská, and J. Kostra, “Mother optimization algorithm: a new human-based metaheuristic approach for solving engineering optimization,” *Scientific Reports*, vol. 13, no. 1, p. 10312, 2023, https://doi.org/10.1038/s41598-023-37537-8.

[27] M. Pluhacek, A. Kazikova, T. Kadavy, A. Viktorin, and R. Senkerik, “Leveraging Large Language Models for the Generation of Novel Metaheuristic Optimization Algorithms,” in *GECCO 2023 Companion - Proceedings of the 2023 Genetic and Evolutionary Computation Conference Companion*, pp. 1812–1820, 2023 https://doi.org/10.1145/3583133.3596401.

[28] E.-S. M. El-Kenawy and A. Ibrahim, “Football Optimization Algorithm (FbOA): A Novel Metaheuristic Inspired by Team Strategy Dynamics,” *Journal of Artificial Intelligence and Metaheuristics*, vol. 8, no. 1, pp. 21–38, 2024, https://doi.org/10.54216/JAIM.080103.

[29] D. Balderas, A. Ortiz, E. Méndez, P. Ponce, and A. Molina, “Empowering Digital Twin for Industry 4.0 using metaheuristic optimization algorithms: case study PCB drilling optimization,” *International Journal of Advanced Manufacturing Technology*, vol. 113, no. 5–6, pp. 1295–1306, 2021, https://doi.org/10.1007/s00170-021-06649-8.

[30] I. Zelinka *et al.*, “Impact of chaotic dynamics on the performance of metaheuristic optimization algorithms: An experimental analysis,” *Information Sciences*, vol. 587, pp. 692–719, 2022, https://doi.org/10.1016/j.ins.2021.10.076.S. C. Donnelly, “Breakfast - The most important meal of the day!!,” *QJM: An International Journal of Medicine*, vol. 111, no. 10, p. 681, 2018, https://doi.org/10.1093/qjmed/hcy201.