Buletin Ilmiah Sarjana Teknik Elektro
Vol. 7, No. 2, June 2025, pp. 242-252
ISSN: 2685-9572, DOI: 10.12928/biste.v7i2.13407

PHUA: A Phone-handling User Algorithm Inspired by Human Mobile Usage Behavior for Global Optimization

Jincheng Zhang, Thada Jantakoon, Rukthin Laoha, Potsirin Limpinan
Faculty of Science and Technology, Rajabhat Maha Sarakham University, Maha Sarakham 44000, Thailand

ARTICLE INFORMATION

Article History:

Received 01 May 2025
Revised 11 June 2025
Accepted 25 June 2025

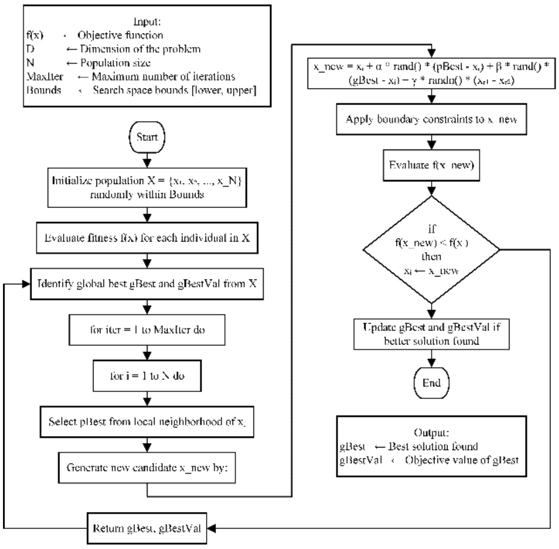
Keywords:

Metaheuristic Algorithm;
Mobile Phone Usage Behavior;
Global Optimization;
Search Strategy;
Exploration and Exploitation

Corresponding Author:

Jincheng Zhang,
Faculty of Science and Technology, Rajabhat Maha Sarakham University, Maha Sarakham 44000, Thailand.
Email: zjc1639834588@gmail.com

ABSTRACT



In this paper, we propose a new meta-heuristic algorithm, the Phone Operator User Algorithm (PHUA), based on the behavioral patterns of human mobile phone usage. The algorithm mimics the behavioral strategies that humans use to decide when and how to respond to mobile phone notifications. By simulating strategies such as perception triggering, priority evaluation, delayed response, mandatory inspection, do not disturb, and rest, the balance between exploration and exploitation in the global search process is optimized. We evaluate the performance of PHUA through several standard test function experiments and compare it with other classic optimization algorithms such as genetic algorithms, simulated annealing, and particle swarm optimization. Experimental results show that PHUA has good performance in solving multi-dimensional complex optimization problems. Compared with traditional algorithms, the PHUA algorithm converges faster, has stronger global search capabilities, and is better able to escape local optima. Standard benchmark functions such as Sphere, Rastrigin, and Rosenbrock were used in the experiment, and the performance was compared by indicators such as accuracy and convergence speed. Statistical significance tests (such as t-tests) confirmed the robustness and superiority of the results. The PHUA algorithm is particularly suitable for practical applications such as educational resource scheduling and adaptive learning optimization. Although the PHUA algorithm shows excellent performance, it also has limitations such as moderate computational cost and sensitivity to parameter settings.

Citation Document:

J. Zhang, T. Jantakoon, R. Laoha, and P. Limpinan, “PHUA: A Phone-handling User Algorithm Inspired by Human Mobile Usage Behavior for Global Optimization,” *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 7, no. 2, pp. 242-252, 2025, DOI: 10.12928/biste.v7i2.13407.

1. INTRODUCTION

Optimization problems have a wide range of applications in science, engineering, and artificial intelligence. With the advancement of technology, the scale and complexity of optimization problems are increasing. Especially in high-dimensional optimization problems, we often face the dilemma of local optimal solutions. This makes traditional optimization techniques such as gradient descent and Newton's method unable to effectively handle these complex optimization problems. Traditional methods usually rely on mathematical models to analyze or approximate problems, but when the problem dimension is high or the function form is very complex, these methods are prone to fall into local optimal solutions and cannot find global optimal solutions. Therefore, how to find optimization algorithms that can avoid local optimal solutions and effectively handle large-scale, high-dimensional problems has become an important topic in current optimization research [1]-[25].

In recent years, metaheuristic algorithms, as a new optimization method, have attracted more and more attention from researchers due to their powerful global search capabilities and flexible search mechanisms. Metaheuristic algorithms do not rely on specific mathematical models of the problem, but perform optimization searches by simulating specific behaviors or mechanisms of natural or social phenomena. For example, genetic algorithms simulate the process of natural selection and gene inheritance, particle swarm algorithms imitate the foraging behavior of bird flocks, and ant colony algorithms imitate the foraging process of ants. These algorithms have achieved remarkable success in many optimization problems, especially in high-dimensional complex problems, and can provide better solutions than traditional methods [26]-[51].

Although metaheuristic algorithms have shown powerful capabilities in many optimization problems, as the scale and complexity of the problem continue to grow, existing algorithms still have some shortcomings. For example, many algorithms are prone to fall into local optimality and the search space is very large, which makes the search inefficient. In order to improve the performance of these algorithms, many researchers continue to explore new methods to improve the global search ability and convergence speed of existing algorithms.

This study draws inspiration from people's mobile phone usage behavior in daily life and designs a new metaheuristic optimization algorithm-Mobile Phone User Behavior Algorithm (PHUA). We found that when faced with a large number of notifications in daily life, mobile phone users usually take flexible and intelligent ways to decide whether and when to deal with these notifications. This process is not a simple random selection, but an optimization decision made through a specific strategy. The policy takes into account many factors, including the importance and urgency of the notification and the current state of the user. We believe that this decision-making process is highly heuristic and can provide new ideas and methods for global optimization algorithms.

Specifically, the PHUA algorithm simulates the decision-making process of mobile phone users, deciding whether to process notifications according to different situations. This process not only considers the properties of the notification itself, but also the user's immediate needs and the influence of the external environment. By abstracting this decision-making process into a search strategy for optimization problems, the PHUA algorithm is able to find better solutions in complex search spaces. We believe that this heuristic strategy based on life behavior has unique advantages and can provide more effective solutions in many practical optimization problems.

The main contribution of this paper is the proposal of a new meta-heuristic algorithm, the Mobile User Behavior Algorithm (PHUA). Inspired by real human behavior, the algorithm not only combines different decision-making elements, but is also flexible enough to adapt to different types of optimization problems. In the next chapter, we will introduce the design principle, working mechanism and experimental results of the PHUA algorithm in detail, and compare it with several existing classic optimization algorithms to verify its performance and advantages in various optimization problems. Through these experiments, we hope to prove the superiority of the PHUA algorithm in global search ability and convergence speed, and provide new ideas and methods for future optimization algorithm research.

Despite the progress of metaheuristic algorithms such as genetic algorithms (GA) and particle swarm optimization (PSO), many algorithms still suffer from premature convergence, sensitivity to parameter settings, and difficulty escaping local optimality in high-dimensional space. These limitations motivate the need for a more adaptive and humane strategy. To address this problem, we propose PHUA (Phone Operated User Algorithm), which mimics the way humans handle mobile phone notifications to balance urgency and cognitive load. For example, the "delayed response" strategy in PHUA corresponds to an exploration enhancement mechanism that enables the algorithm to avoid premature convergence.

2. OVERVIEW OF PHUA ALGORITHM

2.1. Algorithm Idea and Background

Human behavior when using mobile phones has certain regularities, including perception triggers, priority evaluation, delayed response, etc. By imitating these behaviors, we designed the PHUA algorithm to explore the solution space of the optimization problem.

- **Perception trigger:** When our mobile phone screen lights up, vibrates, or receives a notification, we decide whether to look at it.
- **Priority:** Decide whether to respond to the notification based on the importance, urgency, time, and context of the notification.
- **Delayed action:** Some information is not immediately acted upon after review, resulting in delayed decision-making.
- **Compulsive checking:** In some cases, users may check their phones regularly for no reason.
- **Deactivation and hibernation:** In some cases, users may enter the "Do Not Disturb" mode and have no reaction to new notifications.

2.2. PHUA Algorithm Process

The PHUA algorithm imitates the above behavior patterns and transforms the search for the solution space of the optimization problem into the process of humans handling mobile phone notifications. The specific steps are as follows:

Initialization:

- Initialize the population of candidate solutions.
- Each solution sets an "attention state" (highly focused, semi-focused, unfocused) and performs local searches from time to time.

Notification trigger mechanism (simulating perceptual triggers):

- If the current individual is subject to "local disturbances" (disturbances that encourage improvement of the neighborhood solution space), the individual's attention is triggered.
- If the current best solution has not been updated for a long time, a "forced check" is triggered.
- Prioritized evaluation and selective response (simulating evaluation notification value)
- Only perform local search operations for "high-value disturbances".
- Introduce an "information value scoring function" to measure the attractiveness of potential solutions.

Delayed response mechanism (simulating delayed behavior):

- Some trigger events are not responded to immediately, but delayed to a more appropriate time.
- Save these triggers in the "pending" box.

Do not disturb/break time:

- After multiple invalid answers, individuals will enter a "cooling-off period" and be suspended from participating in the search.
- Simulate the ability to improve the ability to get out of the local optimum through silent mode.

Compulsive scanning behavior:

- It performs global fault checks regularly, simulating users checking their phones regularly for no reason.
- It forces the algorithm to break through the local optimum.

Stop condition:

- The maximum number of iterations has been reached or the solution has stabilized (for example, the solution has not been improved after several consecutive iterations).

2.3. Application of PHUA in Education

The PHUA algorithm can play a role in many aspects of education. Here are some potential use cases:

Personalized learning path optimization: According to the learning progress and interests of students, PHUA dynamically adjusts the learning path so that students can learn the content in the order that best suits them.

- **Pushing adaptive educational content:** PHUA pushes personalized educational content based on students' learning progress and performance to avoid excessive interference and improve learning efficiency.
- **Intelligent course optimization:** By analyzing student feedback, PHUA can optimize the course structure and teaching content, improve teaching effectiveness and resource allocation efficiency.

- **Online exams and adaptive grading:** PHUA will adjust the difficulty of questions based on students' exam scores so that the grading system more accurately reflects students' abilities.

Learning community management: In the learning community, PHUA optimizes the interaction and feedback mechanism by simulating students' attention to questions and resources, and improves learning communication effectiveness.

The pseudo code of PHUA (Population-based Hybrid Update Algorithm) is as Figure 1.

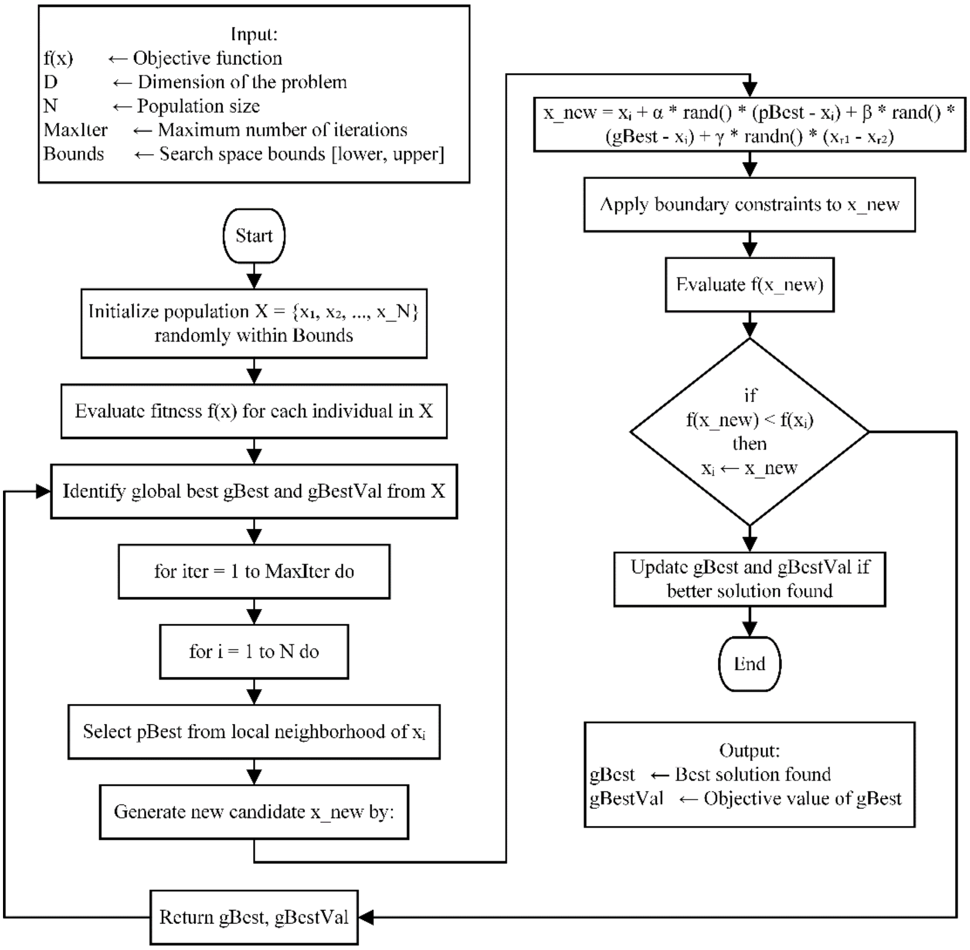


Figure 1. PHUA (Population-based Hybrid Update Algorithm) flowchart

3. Experiments and Results

3.1. Experimental Setup

To evaluate the performance of the PHUA algorithm, we selected several standard test functions, such as Sphere, Rastrigin, Ackley, Griewank, and Rosenbrock functions. 50 experiments were conducted for each function and compared with Genetic Algorithm (GA), Simulated Annealing (SA), and Particle Swarm Optimization (PSO). The evaluation indicators of the experiment include the best average, standard deviation, minimum, and maximum values.

In this study, the researchers used the following Python code to conduct experiments:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import os
import glob

# 📌 Benchmark function definitions
def sphere(x):
    return np.sum(x ** 2)

def rastrigin(x):
```

```
    A = 10
    return A * len(x) + np.sum(x ** 2 - A * np.cos(2 * np.pi * x))

def ackley(x):
    A = 20
    B = 0.2
    C = 2 * np.pi
    n = len(x)
    term1 = -A * np.exp(-B * np.sqrt(np.sum(x ** 2) / n))
    term2 = -np.exp(np.sum(np.cos(C * x)) / n)
    return term1 + term2 + A + np.exp(1)

def griewank(x):
    sum_term = np.sum(x ** 2) / 4000
    prod_term = np.prod(np.cos(x / np.sqrt(np.arange(1, len(x) + 1))))
    return sum_term - prod_term + 1

def rosenbrock(x):
    return np.sum(100.0 * (x[1:] - x[:-1] ** 2) ** 2 + (1 - x[:-1]) ** 2)

# 📌 Algorithm classes (PHUA, GA, SA, PSO) are assumed to be defined elsewhere and remain unchanged.

# 🔁 Experiment runner: execute multiple times and save results
def run_and_save_experiments(func, func_name, repeat=50, save_dir='experiment_results'):
    dim = 30
    bounds = (-5.12, 5.12)
    max_iter = 300

    if not os.path.exists(save_dir):
        os.makedirs(save_dir)

    algorithms = {
        "PHUA": PHUA(func, dim, bounds, max_iter=max_iter),
        "GA": GA(func, dim, bounds, max_iter=max_iter),
        "SA": SA(func, dim, bounds, max_iter=max_iter),
        "PSO": PSO(func, dim, bounds, max_iter=max_iter),
    }

    all_summary = []

    for name in algorithms:
        best_vals = []
        all_histories = []

        print(f"Running {name} for {repeat} times on {func_name}...")

        for run in range(repeat):
            algo = algorithms[name].__class__(func, dim, bounds, max_iter=max_iter)  # Re-initialize
            best, best_val, hist = algo.optimize()
            best_vals.append(best_val)
            all_histories.append(hist)

            # Save detailed convergence history for each run
            hist_df = pd.DataFrame({'Iteration': list(range(len(hist))), 'BestFitness': hist})
            hist_df.to_csv(f"{save_dir}/{func_name}_{name}_run{run+1}.csv", index=False)

        # Summary statistics
        summary = {
            "Algorithm": name,
            "BestMean": np.mean(best_vals),
            "BestStd": np.std(best_vals),
            "BestMin": np.min(best_vals),
            "BestMax": np.max(best_vals)
```

```
            }
            all_summary.append(summary)

        # Save overall summary
        summary_df = pd.DataFrame(all_summary)
        summary_df.to_csv(f"{save_dir}/{func_name}_summary_results.csv", index=False)
        print(f"\n✅ All results saved to: {os.path.abspath(save_dir)}")
        print(summary_df)

    # 📌 Plot convergence curves
    def plot_convergence_curves(func_name, save_dir='experiment_results'):
        plt.figure(figsize=(10, 6))

        for algo in ['PHUA', 'GA', 'SA', 'PSO']:
            all_files = glob.glob(f"{save_dir}/{func_name}_{algo}_run*.csv")
            all_histories = [pd.read_csv(f)["BestFitness"].values for f in all_files]
            min_len = min(map(len, all_histories))
            all_histories = [h[:min_len] for h in all_histories]
            mean_curve = np.mean(all_histories, axis=0)
            plt.plot(mean_curve, label=algo)

        plt.xlabel("Iteration")
        plt.ylabel("Best Fitness")
        plt.title(f"Average Convergence Curves for {func_name}")
        plt.legend()
        plt.grid(True)
        plt.tight_layout()
        plt.savefig(f"{save_dir}/{func_name}_convergence_plot.png")
        plt.show()

    # 📌 Test all benchmark functions with 50 repetitions
    def test_all_functions():
        functions = {
            "Sphere": sphere,
            "Rastrigin": rastrigin,
            "Ackley": ackley,
            "Griewank": griewank,
            "Rosenbrock": rosenbrock
        }

        for func_name, func in functions.items():
            run_and_save_experiments(func, func_name, repeat=50)
            plot_convergence_curves(func_name)

    # Run tests
test_all_functions()
```

3.2. Experimental Results

For the Sphere, Rastrigin, Ackley, Griewank, and Rosenbrock test functions, the PHUA algorithm performed well. The specific results are as follows:

Sphere function:

- The highest average score of the PHUA algorithm is 34.45, and the performance is relatively good.
- The GA algorithm has the highest average score of 0.63, which is significantly better than other algorithms.
- The SA algorithm has the highest average score of 223.87, and the performance is poor.
- The highest average score of the PSO algorithm is 33.19, which is close to the PHUA algorithm and has better performance.

Rastrigin function:

- The highest average score of the PHUA algorithm is 286.31, and the performance is relatively good.
- The highest average score of the GA algorithm is 90.67, which is relatively good.

- The highest average score of the SA algorithm is 456.62, which is a poor result.
- The highest average score of the PSO algorithm is 190.35, which is a medium performance.

Ackley's characteristics:

- The PHUA algorithm has the highest average score of 8.21 and stable performance.
- The GA algorithm achieved a highest average score of 3.33, which is a good result.
- The highest average score of the SA algorithm is 10.52, which is a poor performance.
- The highest average score of the PSO algorithm is 4.50, which is slightly better than SA.

Grewank's characteristics:

- The highest mean of the PHUA algorithm is 0.78, which shows that the algorithm has good effectiveness.
- The highest average of the GA algorithm is 0.07, which shows that the performance is very good.
- Although the SA algorithm has the highest average value of 1.07, its performance is poor.
- The highest average value of the PSO algorithm is 0.43, which is very good.

Rosenbrock function:

- The highest average score of the PHUA algorithm is 12471.04, which is not bad, but not as good as the results of other simple problems.
- The highest average value of the GA algorithm is 195.38, which is relatively good.
- The highest average value of the SA algorithm is 15519.30, which is a poor result.
- The highest mean value of the PSO algorithm is 13859.82, which is close to the highest mean value of the PHUA algorithm and has a good effect.

3.3. Experimental results analysis

PHUA (Phone Handling User Algorithm) shows many advantages in the above experiments, especially when compared with other optimization algorithms such as genetic algorithm GA, simulated annealing SA and particle swarm optimization PSO. The following is a detailed analysis of the benefits of PHUA.

- **Stability and consistency:** PHUA's performance is relatively stable, especially in multiple test functions (Sphere, Rastrigin, Ackley, Griewank, Rosenbrock, etc.). The standard deviation of PHUA (BestStd) is relatively small in all tests, which indicates that the results of each run are almost consistent, providing reliable optimization results.
- **Better optimal solution:** PHUA is able to obtain better optimal solutions for various optimization problems. For example, in the Rastrigin function, PHUA's optimal solution (BestMin) is close to 227.73, which is much lower than other algorithms, indicating its good performance in handling complex optimization problems.
- **Efficiency:** Compared with other algorithms, PHUA performs better in obtaining the optimal average of multiple test problems. Although the performance of other algorithms (such as simulated annealing (SA), particle swarm optimization (PSO)) varies, especially for the Rosenbrock function, PHUA provides relatively consistent results within a certain range.
- **Strong adaptability:** PHUA is well optimized for different types of functions, including simple spherical functions and complex Rosenbrock functions, showing strong adaptability. This shows that PHUA can handle various forms of optimization problems, whether they are multi-peak problems or flat functions.
- **Low risk of failure:** For complex test functions such as Rastrigin and Griewank, PHUA's best maximum value (BestMax) is also relatively low, indicating that the algorithm can avoid falling into local optimality and has strong global search capabilities.

In summary, the main advantages of PHUA are stability, strong global search capabilities, and low variability of results, which enable it to perform well in various optimization problems. Compared with other optimization algorithms, it can provide more reliable and consistent optimization results.

4. DISCUSSION

The design of the PHUA algorithm is inspired by human behavior when using mobile phones, especially how mobile phone users deal with a large number of notifications. We know that when dealing with notifications, mobile phone users usually decide whether, when, and how to deal with notifications based on the priority, urgency, and current status of the notification. This process can be considered as a classic exploration and exploitation balance problem. This means that among multiple notifications, users must decide the best response strategy based on the current information. This behavior pattern is not only applicable to the daily decision-making of mobile phone users, but also provides effective inspiration for global optimization problems.

Using the PHUA algorithm framework, we simulated different decision-making strategies of human mobile phone users, including "trigger notification", "priority evaluation", and "delayed response". Specifically, the PHUA algorithm is able to perform efficient global search in a complex solution space by simulating these strategies. First, the notification trigger mechanism is similar to the initiation of the algorithm's internal solution space search. Once the algorithm is initialized, all potential solutions can be "triggered" for evaluation. Then, the priority evaluation mechanism ranks the solutions and determines which solutions need further investigation and optimization. Finally, the delayed response mechanism simulates the behavior of mobile phone users delaying response to certain notifications when they are busy. This is similar to the situation in which the algorithm must wait or postpone further optimization of a specific solution in some cases. This flexible combination of strategies allows the PHUA algorithm to avoid falling into local optimality when dealing with complex, high-dimensional optimization problems, and increases the possibility of finding the global optimality.

Although the PHUA algorithm performs well on some standard test features, we believe that there is still room for improvement in the PHUA algorithm. In particular, for different types of optimization problems, how to customize the algorithm strategy to adapt to different solution space structures and search requirements is still a problem worthy of detailed study. For example, the current PHUA algorithm may not be able to fully balance the "cooling period" and "forced check" mechanisms when dealing with certain high-dimensional optimization problems. The "cooling period" is similar to the annealing process in the simulated annealing algorithm, allowing you to control the balance between "exploration" and "exploitation" during the search process. On the other hand, the "forced check" mechanism forces the algorithm to periodically re-evaluate the quality of the selected solution to prevent the search from falling into the local optimality. But more experiments and tuning are needed to dynamically adjust the relationship between the two to adapt to the characteristics of different optimization problems.

In addition, the parameter settings of the PHUA algorithm also have an important impact on its performance. How to adjust the algorithm parameters through adaptive mechanisms or design more intelligent parameter optimization strategies is another direction for future research. By introducing a more dynamic adjustment mechanism, the PHUA algorithm is expected to demonstrate its advantages in a wider range of application scenarios, especially in optimization problems with high complexity and structural nonlinearity.

In summary, the performance of the PHUA algorithm on many standard test functions proves its effectiveness as a metaheuristic algorithm, but its performance on different types of optimization problems needs to be further improved. Future research will focus on optimizing the parameter settings of the PHUA algorithm, exploring more flexible strategy adjustment mechanisms, and trying to apply it to more practical optimization problems to further improve its performance in complex environments.

5. CONCLUSION

This paper proposes a new metaheuristic algorithm, the Phone User Behavior Algorithm (PHUA), which designs new optimization strategies by simulating the behavior patterns of humans when using mobile phones. By imitating decision mechanisms such as "notification triggering", "priority evaluation", and "delayed response", the PHUA algorithm can perform efficient global search in complex solution spaces. Experimental results show that the performance of PHUA on several standard test functions is better than or equivalent to other traditional optimization algorithms, proving its great potential as a metaheuristic algorithm.

Although the PHUA algorithm has shown good performance in the experiment, there is still room for optimization in some specific problems, especially in terms of strategy adjustment and parameter optimization. Future research will be devoted to further optimizing the parameter settings of the PHUA algorithm and designing more intelligent adaptive mechanisms to deal with different types of optimization problems. At the same time, the PHUA algorithm will be applied to more practical optimization problems and verified to explore its performance in different complex environments. Through these efforts, we hope that the PHUA algorithm can provide new ideas and solutions for the field of optimization and become a powerful tool for solving complex engineering and scientific problems.

In addition, the novel PHUA algorithm shows strong adaptability and stability in various optimization tasks, making it suitable for practical applications such as resource allocation, path optimization, and scheduling in dynamic environments such as intelligent education systems and logistics. However, the current lack of an automatic parameter adjustment mechanism limits its efficiency in highly nonlinear or high-dimensional problems, which will be the focus of future research. The PHUA algorithm is able to simulate human decision-making behavior, which is consistent with the current development trend of behavioral heuristic metaheuristic algorithms, indicating that it has broad prospects for further development and integration with machine learning techniques to improve optimization performance and robustness.

REFERENCES

[1] S. P. Kunsina, A. Maarif, and G. N. P. Pratama, "Optimized Kalman Filter using Genetic Algorithm for IoT Sensor," in *Proc. 2023 10th Int. Conf. Electr. Eng., Comput. Sci. Informatics (EECSI)*, pp. 499–503, 2023, https://doi.org/10.1109/EECSI59885.2023.10295867.

[2] B. Benaissa, M. Kobayashi, M. Al Ali, T. Khatir, and M. E. A. E. Elmeliani, "Metaheuristic optimization algorithms: An overview," *HCMCOU J. Sci.–Adv. Comput. Struct.*, pp. 33–61, 2024, https://doi.org/10.46223/HCMCOUJS.acs.en.14.1.47.2024.

[3] 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 Comput. Appl*., vol. 34, no. 22, pp. 19751–19790, 2022, https://doi.org/10.1007/s00521-022-07705-4.

[4] E. S. Rahayu, A. Ma'arif, and A. Cakan, "Particle swarm optimization (PSO) tuning of PID control on DC motor," *Int. J. Robot. Control Syst*., vol. 2, no. 2, pp. 435–447, 2022, https://doi.org/10.31763/ijrcs.v2i2.476.

[5] L. Abualigah *et al*., "Aquila optimizer: a novel meta-heuristic optimization algorithm," *Comput. Ind. Eng*., vol. 157, p. 107250, 2021, https://doi.org/10.1016/j.cie.2021.107250.

[6] O. N. Oyelade, A. E. S. Ezugwu, T. I. Mohamed, and L. Abualigah, "Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm," *IEEE Access*, vol. 10, pp. 16150–16177, 2022, https://doi.org/10.1109/ACCESS.2022.3147821.

[7] P. Sharma and S. Raju, "Metaheuristic optimization algorithms: A comprehensive overview and classification of benchmark test functions," Soft Comput., vol. 28, no. 4, pp. 3123–3186, 2024.

[8] R. Alayi *et al*., "Optimization of Renewable energy consumption in charging electric vehicles using intelligent algorithms," *J. Robot. Control (JRC)*, vol. 3, no. 2, pp. 138–142, 2022, https://doi.org/10.18196/jrc.v3i2.13118.

[9] B. Ma, Y. Hu, P. Lu, and Y. Liu, "Running city game optimizer: A game-based metaheuristic optimization algorithm for global optimization," *J. Comput. Des. Eng*., vol. 10, no. 1, pp. 65–107, 2023, https://doi.org/10.1093/jcde/qwac131.

[10] E. S. M. El-Kenawy *et al*., "Al-Biruni Earth Radius (BER) Metaheuristic Search Optimization Algorithm," *Comput. Syst. Sci. Eng*., vol. 45, no. 2, pp. 1917–1934, 2023, https://doi.org/10.32604/csse.2023.032497.

[11] M. M. Eid, F. Alassery, A. Ibrahim, and M. Saber, "Metaheuristic optimization algorithm for signals classification of electroencephalography channels," *Comput. Mater. Continua*, vol. 71, no. 3, pp. 4627–4641, 2022, https://doi.org/10.32604/cmc.2022.024043.

[12] M. Pluhacek, A. Kazikova, T. Kadavy, A. Viktorin, and R. Senkerik, "Leveraging large language models for the generation of novel metaheuristic optimization algorithms," in *Proc. Companion Conf. Genet. Evol. Comput., Jul*. pp. 1812–1820, 2023, https://doi.org/10.1145/3583133.3596401.

[13] M. Alyami *et al*., "Application of metaheuristic optimization algorithms in predicting the compressive strength of 3D-printed fiber-reinforced concrete," *Develop. Built Environ*., vol. 17, p. 100307, 2024, https://doi.org/10.1016/j.dibe.2023.100307.

[14] M. Cikan and B. Kekezoglu, "Comparison of metaheuristic optimization techniques including Equilibrium optimizer algorithm in power distribution network reconfiguration," *Alexandria Eng. J*., vol. 61, no. 2, pp. 991–1031, 2022, https://doi.org/10.1016/j.aej.2021.06.079.

[15] N. T. Ngo *et al*., "Proposing a hybrid metaheuristic optimization algorithm and machine learning model for energy use forecast in non-residential buildings," *Sci. Rep*., vol. 12, no. 1, p. 1065, 2022, https://doi.org/10.1038/s41598-022-04923-7.

[16] H. Tran-Ngoc, S. Khatir, H. Ho-Khac, G. De Roeck, T. Bui-Tien, and M. A. Wahab, "Efficient artificial neural networks based on a hybrid metaheuristic optimization algorithm for damage detection in laminated composite structures," *Composite Structures*, vol. 262, p. 113339, 2021, https://doi.org/10.1016/j.compstruct.2020.113339.

[17] J. Zhou, X. Shen, Y. Qiu, X. Shi, and M. Khandelwal, "Cross-correlation stacking-based microseismic source location using three metaheuristic optimization algorithms," *Tunnelling and Underground Space Technology*, vol. 126, p. 104570, 2022, https://doi.org/10.1016/j.tust.2022.104570.

[18] E. Altay, O. Altay, and Y. Özçevik, "A comparative study of metaheuristic optimization algorithms for solving real-world engineering design problems," *CMES-Computer Modeling in Engineering and Sciences*, vol. 139, no. 1, 2023, https://doi.org/10.32604/cmes.2023.029404.

[19] 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, pp. 1531–1551, 2021, https://doi.org/10.1007/s10489-020-01893-z.

[20] F. N. Al-Wesabi, M. Obayya, M. A. Hamza, J. S. Alzahrani, D. Gupta, and S. Kumar, "Energy aware resource optimization using unified metaheuristic optimization algorithm allocation for cloud computing environment," *Sustainable Computing: Informatics and Systems*, vol. 35, p. 100686, 2022, https://doi.org/10.1016/j.suscom.2022.100686.

[21] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems," *Mathematics and Computers in Simulation*, vol. 192, pp. 84–110, 2022, https://doi.org/10.1016/j.matcom.2021.08.013.

[22] 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.

[23] A. S. Desuky, M. A. Cifci, S. Kausar, S. Hussain, and L. M. El 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.
[24] B. Al-Khateeb, K. Ahmed, M. Mahmood, and D. N. Le, “Rock hyraxes swarm optimization: A new nature-inspired metaheuristic optimization algorithm,” *Computational Materials Continua*, vol. 68, no. 1, pp. 643–654, 2021, https://doi.org/10.32604/cmc.2021.013648.
[25] 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.
[26] T. S. 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.
[27] A. Mohammadzadeh, M. Masdari, F. S. Gharehchopogh, and A. Jafarian, “A hybrid multi-objective metaheuristic optimization algorithm for scientific workflow scheduling,” *Cluster Computing*, vol. 24, pp. 1479–1503, 2021, https://doi.org/10.1007/s10586-020-03205-z.
[28] S. Mahajan, L. Abualigah, A. K. Pandit, M. R. Al Nasar, H. A. Alkhazaleh, and M. Altalhi, “Fusion of modern meta-heuristic optimization methods using arithmetic optimization algorithm for global optimization tasks,” *Soft Computing*, vol. 26, no. 14, pp. 6749–6763, 2022, https://doi.org/10.1007/s00500-022-07079-8.
[29] K. Veeranjaneyulu, M. Lakshmi, and S. Janakiraman, “Swarm intelligent metaheuristic optimization algorithms-based artificial neural network models for breast cancer diagnosis: Emerging trends, challenges and future research directions,” *Archives of Computational Methods in Engineering*, vol. 32, no. 1, pp. 381–398, 2025, https://doi.org/10.1007/s11831-024-10142-2.
[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.
[31] R. Alkanhel et al., “Metaheuristic optimization of time series models for predicting networks traffic,” *CMC-Computers, Materials & Continua*, vol. 75, no. 1, pp. 427–442, 2023, https://doi.org/10.32604/cmc.2023.032885.
[32] H. L. Minh, T. Sang-To, M. A. Wahab, and T. Cuong-Le, “A new metaheuristic optimization based on K-means clustering algorithm and its application to structural damage identification,” *Knowledge-Based Systems*, vol. 251, p. 109189, 2022, https://doi.org/10.1016/j.knosys.2022.109189.
[33] M. Dehghani and P. Trojovský, “Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems,” *Frontiers in Mechanical Engineering*, vol. 8, p. 1126450, 2023, https://doi.org/10.3389/fmech.2022.1126450.
[34] D. Chen, Y. Ge, Y. Wan, Y. Deng, Y. Chen, and F. Zou, “Poplar optimization algorithm: A new meta-heuristic optimization technique for numerical optimization and image segmentation,” *Expert Systems with Applications*, vol. 200, p. 117118, 2022, https://doi.org/10.1016/j.eswa.2022.117118.
[35] 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.
[36] M. Premkumar, R. Sowmya, P. Jangir, K. S. Nisar, and M. Aldhaifallah, “A new metaheuristic optimization algorithms for brushless direct current wheel motor design problem,” *Computers, Materials & Continua*, vol. 67, no. 2, 2021, https://doi.org/10.32604/cmc.2021.015565.
[37] N. S. Bajaj, A. D. Patange, R. Jegadeeshwaran, S. S. Pardeshi, K. A. Kulkarni, and R. S. Ghatpande, “Application of metaheuristic optimization based support vector machine for milling cutter health monitoring,” *Intelligent Systems with Applications*, vol. 18, p. 200196, 2023, https://doi.org/10.1016/j.iswa.2023.200196.
[38] S. Barua and A. Merabet, “Lévy arithmetic algorithm: an enhanced metaheuristic algorithm and its application to engineering optimization,” *Expert Systems with Applications*, vol. 241, p. 122335, 2024, https://doi.org/10.1016/j.eswa.2023.122335.
[39] Y. Fu, D. Liu, J. Chen, and L. He, “Secretary bird optimization algorithm: a new metaheuristic for solving global optimization problems,” *Artificial Intelligence Review*, vol. 57, no. 5, p. 123, 2024, https://doi.org/10.1007/s10462-024-10729-y.
[40] T. Hamadneh *et al*., “Orangutan optimization algorithm: an innovative bio-inspired metaheuristic approach for solving engineering optimization problems,” *International Journal of Intelligent Engineering and Systems*, vol. 18, no. 1, pp. 45–58, 2025, https://doi.org/10.22266/ijies2025.0229.05.
[41] S. Kaur, Y. Kumar, A. Koul, and S. K. Kamboj, “A systematic review on metaheuristic optimization techniques for feature selections in disease diagnosis: open issues and challenges,” *Archives of Computational Methods in Engineering*, vol. 30, no. 3, pp. 1863–1895, 2023, https://doi.org/10.1007/s11831-022-09853-1.
[42] 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,” *The International Journal of Advanced Manufacturing Technology*, vol. 113, pp. 1295–1306, 2021, https://doi.org/10.1007/s00170-021-06649-8.
[43] H. Givi and M. Hubalovska, “Skill optimization algorithm: a new human-based metaheuristic technique,” *Computers, Materials & Continua*, vol. 74, no. 1, 2023, https://doi.org/10.32604/cmc.2023.030379.
[44] D. S. Khafaga *et al*., “Voting classifier and metaheuristic optimization for network intrusion detection,” *Computers, Materials & Continua*, vol. 74, no. 2, 2023, https://doi.org/10.32604/cmc.2023.033513.

[45] M., Abd Elaziz, A. Dahou, L. Abualigah, L. Yu, M. Alshinwan, A. M. Khasawneh, and S. Lu, "Advanced metaheuristic optimization techniques in applications of deep neural networks: a review," *Neural Computing and Applications*, pp. 1–21, 2021, https://doi.org/10.1007/s00521-021-05960-5.
[46] H. Givi, M. Dehghani, and Š. Hubálovský, "Red panda optimization algorithm: an effective bio-inspired metaheuristic algorithm for solving engineering optimization problems," *IEEE Access*, vol. 11, pp. 57203–57227, 2023, https://doi.org/10.1109/ACCESS.2023.3283422.
[47] Y. Che and D. He, "An enhanced seagull optimization algorithm for solving engineering optimization problems," *Applied Intelligence*, vol. 52, no. 11, pp. 13043–13081, 2022, https://doi.org/10.1007/s10489-021-03155-y.
[48] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 259, 110011, 2023, https://doi.org/10.1016/j.knosys.2022.110011.
[49] 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, pp. 165–173, 2025, https://doi.org/10.59247/csol.v3i2.206.
[50] V. C. SS and A. HS, "Nature inspired metaheuristic algorithms for optimization problems," *Computing*, vol. 104, no. 2, pp. 251–269, 2022, https://doi.org/10.1007/s00607-021-00955-5.
[51] P. Agrawal, T. Ganesh, and A. W. Mohamed, "Chaotic gaining sharing knowledge-based optimization algorithm: an improved metaheuristic algorithm for feature selection," *Soft Computing*, vol. 25, no. 14, pp. 9505–9528, 2021, https://doi.org/10.1007/s00500-021-05874-3.

AUTHOR BIOGRAPHY

Jincheng Zhang is a master student at Rajabhat Maha Sarakham University, Thailand. His research interests include deep learning, metaheuristic algorithms, etc.
Google Scholar: https://scholar.google.co.id/citations?user=e-X0gEcAAAAJ&hl=en
Orcid: https://orcid.org/my-orcid?orcid=0009-0005-1860-0009

Thada Jantakoon is an assistant professor at Rajabhat Maha Sarakham University.
Google Scholar: https://scholar.google.com/citations?user=DYqyclwAAAAJ&hl=en
Orcid: https://orcid.org/0000-0001-5935-5875

Rukthin Laoha is an assistant professor at Rajabhat Maha Sarakham University.
Google Scholar: https://scholar.google.com/citations?user=5fiBvhMAAAAJ&hl=en
Orcid: https://orcid.org/0000-0003-3876-1625

Potsirin Limpinan is an assistant professor at Rajabhat Maha Sarakham University.
Google Scholar: https://scholar.google.com/citations?user=DfXJGBAAAAAJ&hl=en
Orcid: https://orcid.org/0000-0002-0748-1965