

Classical Dance-Metaheuristic: A Metaheuristic Optimization Algorithm Inspired by Classical Dance

Jincheng Zhang ^{1,*} , Jindong Zhang ²

¹ Faculty of Science and Technology, Rajabhat Maha Sarakham University, Maha Sarakham 44000, Thailand

² Independent Researcher, Putian, Fujian 351144, China

Email: ¹ zjc1639834588@gmail.com

*Corresponding Author

Abstract—This paper proposes a metaheuristic optimization algorithm based on classical dance, namely the Classical Dance Metaheuristic (CDMH). The algorithm combines the core elements of ballet, Indian classical dance and Chinese classical dance with modern optimization techniques, providing a new approach to high-dimensional optimization problems. The CDMH algorithm optimizes the search process through three stages of simulation: the posture training stage in ballet, the rhythm and mudra exploration stage in Indian classical dance, and the integration stage of body, rhythm and artistic conception in Chinese classical dance. Experimental results show that CDMH shows good optimization ability in multiple classic optimization problems and can effectively avoid the dilemma of local optimal solutions.

Keywords—Metaheuristic Algorithms, CDMH, Classical Dance, Optimization, Machine Learning, Interdisciplinary Approaches

I. INTRODUCTION

With the rapid development of intelligent algorithms, metaheuristic algorithms, as an innovative solution method, have shown their unique advantages in dealing with complex optimization problems. It allows flexible and effective exploration of possible solutions in a large and complex search space to find the global optimum. Therefore, it is widely used in various fields such as machine learning, data mining, and engineering design. The main advantage of this type of algorithm is that it overcomes the limitations of traditional optimization methods and avoids the dilemma of falling into local optimality, especially when facing large-scale, high-dimensional and complex problems. Despite this, traditional optimization methods still achieve good results for some problems, and their algorithmic mechanisms perform relatively well in some optimization tasks with structural constraints. However, in general, they have problems such as being prone to falling into local optimality and limited scalability. Therefore, how to balance the accuracy and efficiency of the algorithm and how to avoid the limitations of traditional methods in the process of solving optimization problems have become important research topics. In this context, metaheuristic algorithms have attracted the attention of researchers due to their unique global search capabilities and adaptability under multi-constraint conditions, especially when dealing with complex, dynamic and large-scale problems [1]-[25].

Metaheuristic algorithms are flexible enough to find good solutions in multiple fields and can also handle complex

constraints of different problems through their adaptation mechanisms. In addition, by combining multiple strategies, we can achieve a balance between local and global search and avoid premature convergence. With the improvement of computing power, many new variants and optimization strategies continue to emerge, the performance of metaheuristic algorithms continues to improve, and the field continues to develop actively. Among these changes, some algorithms are based on simulating natural phenomena, while others draw on ideas from fields such as biology, sociology, and art to create new heuristic methods. Through interdisciplinary integration and innovation, metaheuristic algorithms have gradually demonstrated powerful problem-solving capabilities and become a powerful tool for solving complex problems [26]-[56].

To solve this complex situation, this paper proposes a new metaheuristic optimization algorithm, namely the classical dance metaheuristic (CDMH), which combines elements of classical dance. The algorithm not only starts from the traditional mathematical model, but also draws on the unique artistic form and training methods of classical dance, absorbs its essence and precision, and forms an artistic and efficient optimization process. The CDMH algorithm combines three dance elements: ballet, Indian classical dance, and Chinese classical dance. It transforms the characteristics of these elements into the key steps of the optimization algorithm, aiming to ensure the global search capability of the algorithm while improving the elegance and efficiency of the solution process. The CDMH algorithm aims to combine the artistry of dance with the rationality of optimization to provide novel solutions to complex optimization problems. This is innovative and can effectively improve the efficiency of solutions in practical applications.

II. INSPIRATION AND DESIGN IDEAS OF THE CDMH ALGORITHM

The design ideas of the CDMH algorithm are derived from the dance training and performance of classical dance art, especially ballet, Indian classical dance, and Chinese classical dance. Each dance form has its own training mode and performance style, but they all emphasize the precision, rhythm and coordination of movements, as well as the internal and external discipline in the performance [57]-[65]. In this paper, we analyze these dance forms in detail and propose to transform their core ideas into three main stages

of the optimization process: solution construction, local search perturbation, and global optimization stabilization adjustment.

A. Ballet (*Corrective Posture, Basic Skills Training*)

Ballet emphasizes the precision of movements and the symmetry of postures. Through a lot of basic training, dancers have mastered extremely perfect body control and precision of movements. In this stage, the CDMH algorithm imitates the training method of ballet and optimizes the structure and form of the solution, aiming to ensure the balance and symmetry of the solution. In this process, the algorithm adjusts the overall orientation and initial position of the solution by optimizing the "pose" to ensure the stability of the optimization process and the initial quality of the solution. The optimization results of this stage lay the foundation for the subsequent search process, allowing us to explore solutions from a good starting point.

B. Indian Classical Dance (*Rhythm, Gesture, Coordination*)

Indian classical dance emphasizes the harmony of rhythm and gesture. Dancers express their inner feelings and stories through precise rhythm control and rich gesture changes. In the second stage of CDMH, the algorithm imitates the rhythm of Indian dance and changes its gestures and positions in the solution space to explore more possible solutions. In this stage, the algorithm tries to find potential solutions to the problem by perturbing the solution space and speeding up the search for solutions by changing the rhythm. The main goal of this stage is to quickly break through the local optimum through local search and expand to a larger solution space.

C. Chinese Classical Dance (*Body Rhythm, Rotation, Breathing*)

Chinese classical dance emphasizes the unity of "form, spirit, and rhythm", and emphasizes the integration of the fluency of movements and the sense of rhythm. In this stage, CDMH adjusts the global and local dynamics of the solution by introducing flexible control mechanisms such as simulated annealing to simulate the breathing and turning skills of dance. In this way, the algorithm gradually adjusts the stability and elegance of the solution during the search process, ensuring that the quality of the solution during the search process is stably close to the global optimal solution, and through reasonable transformation and adjustment, the overall elegance of the final solution is improved. The central idea of this stage is to adjust the global optimization strategy so that the algorithm remains stable throughout the search process and avoids excessive local optimization.

Through the above three-stage design, the CDMH algorithm not only embodies the artistry of classical dance, but also integrates the optimization strategy of the metaheuristic algorithm, providing an elegant and efficient solution for complex optimization problems.

III. CDMH ALGORITHM FRAMEWORK AND WORKING PRINCIPLE

The core structure of the CDMH algorithm consists of several modules, covering all steps from initialization, three optimization stages to evaluation and update. The overall process aims to effectively explore and optimize the solution

space by simulating the essence of classical dance, and finally obtain the optimal solution. This process not only guarantees the calculation process of each stage, but also ensures the efficiency and elegance of the algorithm in complex search space through flexible control mechanisms. The specific steps are as follows:

To better understand the logic behind each stage, the CDMH algorithm draws metaphorical inspiration from three classical dances to simulate optimization dynamics. Each dance style corresponds to a specific optimization strategy, forming a structure that closely combines artistic interpretation and parameter-driven mechanisms.

A. Ballet (*Posture Correction, Basic Training*)

Ballet emphasizes the precision of movements and the symmetry of postures. Through a lot of basic training, dancers have mastered extremely perfect body control and precision of movements. In this stage, the CDMH algorithm imitates the training method of ballet, aiming to optimize the structure and shape of the solution to ensure the balance and symmetry of the solution. In this process, the algorithm adjusts the overall orientation and initial position of the solution by optimizing the "pose" to ensure the stability of the optimization process and the initial quality of the solution. The optimization results of this stage lay the foundation for the subsequent search process, making it possible to find a solution from the right starting point.

B. Indian Classical Dance (*Rhythm, Gesture, Coordination*)

Indian classical dance emphasizes the harmony of rhythm and gesture. Dancers express their inner feelings and stories through precise rhythm control and rich gestures. In the second stage of CDMH, the algorithm imitates the rhythm of Indian dance, changing its gestures and positions in the solution space to explore more possible solutions. In this stage, the algorithm tries to find potential solutions to the problem by perturbing the solution space, changing the speed, and speeding up the solution search. The main goal of this stage is to quickly break through the local optimum through local search and expand to a larger solution space.

C. Chinese Classical Dance (*Body Rhythm, Rotation, Breathing*)

Chinese classical dance emphasizes the unity of "form, spirit, and rhythm", and pays attention to the combination of the fluency of movements and the sense of rhythm. In this stage, CDMH introduces flexible control mechanisms such as simulated annealing to adjust the global and local dynamics of the solution and simulate the dancer's breathing and rotation skills. In this way, the algorithm gradually adjusts the stability and accuracy of the solution during the search process, so that the quality of the solution during the search process steadily approaches the global optimal solution, and through reasonable transformation and adjustment, the overall accuracy of the final solution is improved. The central idea of this stage is to adjust the global optimization strategy so that the algorithm remains stable throughout the search process and avoids excessive local optimization.

Through the above three-stage design, the CDMH algorithm not only embodies the artistry of classical dance, but also integrates the optimization strategy of meta-heuristic

algorithms, providing an elegant and efficient solution to complex optimization problems.

D. Initialization Stage

The initialization stage is the starting point of the CDMH algorithm and lays the foundation for the subsequent optimization process. First, the algorithm randomly generates a set of initial solutions. Each solution represents a dance posture structure, which contains three main vectors: posture vector, rhythm vector and artistic concept vector. Each solution corresponds to the skeleton of a potential dance action, and the combination of these vectors constructs a preliminary definition of the dance posture, similar to the basic movements made by dancers when they first learn to dance.

At this stage, the algorithm also needs to set search space constraints such as posture range, rhythm range and energy limit for each dance posture. Pose limits ensure that the shape of each dance pose does not exceed the specified range of motion, rhythm limits determine the speed and frequency of dance movements, and energy constraints ensure that the algorithm does not over-search during the optimization process and avoid infeasible solutions.

E. Iterative Optimization Phase

In the iterative optimization phase of the CDMH algorithm, each iteration contains three core optimization steps, which simulate the training process of ballet, Indian classical dance, and Chinese classical dance respectively. These three phases refine and optimize the solution from different perspectives, allowing the algorithm to effectively explore the solution space in multiple dimensions. Each phase is independent and complementary, working together to make the solution reach the global optimum.

1) Phase 1: Optimizing Form Constraints

In this phase, the CDMH algorithm ensures the symmetry and structural optimization of the solution by adjusting the "pose encoding". Similar to strict ballet training, pose adjustment emphasizes the precision and symmetry of the movement. In this phase, the algorithm adopts a symmetric loss function and a penalized L2 regularization term, aiming to reduce redundancy and asymmetry in the solution and optimize the overall shape of the solution. This process ensures the elegance and stability of the solution and finds the best point between symmetry and balance.

2) Phase 2: Rhythm and Gesture Changes in Indian Dance (Exploring Rhythm)

Rhythm and gesture are the most important elements of Indian classical dance. Dancers express emotions and intensity by controlling changes in rhythm and gestures. Inspired by this art form, the CDMH algorithm introduces a rhythm perturbation mechanism in this phase to simulate the changes in the dancer's sense of rhythm through a combination of beat-type polynomials and random perturbations. The solutions in the solution space are constantly shifting and changing, exploring multiple possible solution paths, just like dancers switching between movements at different rhythms. This phase not only facilitates local search, but also simulates rhythm changes

through finite state machines, making the search process more dynamic and diverse.

3) Phase 3: Deliberate Traffic Control

Chinese classical dance emphasizes the unity of "form, spirit, and rhyme". Every movement of the dancer contains profound meaning and rhythm. In the third phase of CDMH, the algorithm takes advantage of this idea and introduces a flow mechanism in the simulated annealing algorithm to adjust the global and local dynamics of the solution. The algorithm adopts "breathing regulation". That is, the speed and direction of the search process are adjusted according to the current state of the solution, and the balance between global search and local search is flexibly adjusted to ensure a smooth transition of the optimization process. In this phase, we effectively search for solutions globally while fine-tuning locally to eventually find the optimal solution that satisfies all constraints.

F. Evaluation and Update

After each iteration, the CDMH algorithm evaluates the current solution and updates the solution based on the evaluation results. The evaluation process uses a fusion scoring function that comprehensively considers gesture beauty, rhythmic harmony, and emotional energy expression. The gesture beauty score measures the formal symmetry and elegance of the solution, the rhythmic harmony score evaluates the rhythm of the action and the smoothness of gesture changes in the solution, and the emotional energy is scored based on the inherent potential and expressiveness of the solution. The comprehensive evaluation of these three aspects allows the algorithm to accurately judge the quality of the solution.

In the update phase, the CDMH algorithm uses an elite strategy to retain the current optimal solution and avoids falling into the local optimum through the artistic concept preservation mechanism. The elite strategy records and saves the optimal solution in each round of iteration, and uses the optimal solution after each iteration as the starting point for the next round of search. The artistic concept preservation mechanism allows the algorithm to maintain elegance and stability while pursuing the optimal solution, avoiding over-optimization or falling into the local optimum.

G. Return to the Optimal Solution

After completing all iterations, the CDMH algorithm outputs an optimal solution path that is both formally and psychologically the best solution. The solution represents the best dance posture structure obtained after three stages of optimization under complex constraints. It not only meets the requirements of shape optimization, but also expresses a precise sense of rhythm and perfect artistic conception. The solution is not only numerically optimal, but also reflects the elegance and efficiency of the algorithm in the optimization process.

Here is an explanation of the ballet phase in pseudocode: Mathematically speaking, the "posture" of this phase corresponds to the vector update strategy of the structural parameters of the solution. Specifically, the algorithm uses a symmetric forced loss function and L2 regularization to adjust the direction and position vectors of the solution to achieve a balanced, stable update. This corresponds to

controlling the shape vector of the solution to maintain structural balance in the solution space.

Pseudocode of Algorithm CDMH:
<p>Input:</p> <ul style="list-style-type: none"> - Objective function $f(x)$ - Number of agents N - Number of iterations T - Search space boundaries $[x_{min}, x_{max}]$ <p>Output:</p> <ul style="list-style-type: none"> - Best solution x_{best} <p>1. Initialization:</p> <p>For each agent $i \in \{1, \dots, N\}$ do:</p> <p style="padding-left: 20px;">Randomly initialize posture_vector[i], rhythm_vector[i], concept_vector[i]</p> <p style="padding-left: 20px;">Combine as solution $x[i]$</p> <p style="padding-left: 20px;">Constrain $x[i]$ within $[x_{min}, x_{max}]$</p> <p>EndFor</p> <p>Evaluate all $x[i]$, record best x_{best}</p> <p>2. Iterative Optimization (for $t = 1$ to T):</p> <p>For each agent $i \in \{1, \dots, N\}$ do:</p> <p style="padding-left: 20px;">### Phase 1: Ballet Stage - Posture Optimization</p> <ul style="list-style-type: none"> - Apply symmetry_loss($x[i]$) + L2_regularization($x[i]$) - Adjust posture_vector[i] to enhance balance and symmetry - $x[i] \leftarrow \text{PoseAdjusted}(x[i])$ <p style="padding-left: 20px;">### Phase 2: Indian Classical Dance Stage - Rhythm Exploration</p> <ul style="list-style-type: none"> - Perturb rhythm_vector[i] with rhythmic polynomial & noise - Use rhythm_FSM (finite state machine) to guide changes - $x[i] \leftarrow x[i] + \text{RhythmPerturbation}(x[i])$ <p style="padding-left: 20px;">### Phase 3: Chinese Dance Stage - Global- Local Flow (Simulated Annealing)</p> <ul style="list-style-type: none"> - Compute $\Delta f = f(x[i]) - f(x_{best})$ - If $\Delta f < 0$ or $\exp(-\Delta f / T) > \text{random}(0,1)$: # Breathing control <li style="padding-left: 20px;">Accept new $x[i]$ with higher probability <li style="padding-left: 20px;">Adjust step size via "breathing" strategy: flow_rate(t) <li style="padding-left: 20px;">- $x[i] \leftarrow x[i] + \text{BreathingFlow}(x[i], t)$ <p style="padding-left: 20px;">### Evaluate and Update</p> <ul style="list-style-type: none"> - Evaluate new $f(x[i])$ - If $f(x[i]) < f(x_{best})$: update $x_{best} \leftarrow x[i]$ <p>EndFor</p> <p style="padding-left: 20px;">### Artistic Preservation and Elite Strategy</p> <ul style="list-style-type: none"> - Keep best k solutions from previous generation - Avoid replacing them unless new score is significantly better <p>3. Return x_{best}</p>

Indian classical dance stage explanation: The "rhythm" and "gesture" of this phase directly correspond to the step size and perturbation control during the solution search process. By dynamically adjusting the step size using a rhythmic

perturbation mechanism (modeled with beat-type polynomials and random noise), the algorithm explores a larger solution space and simulates a wider variety of solution trajectories. This means adaptively updating the size and direction of the solution vector to achieve efficient local search and escape from the local optimum.

Chinese classical dance performance explanation: The "breathing" and "rotation" techniques here correspond to the global and local dynamic adjustment mechanisms in the simulated annealing framework. The algorithm controls the temperature and acceptance probability vectors to gradually adjust the intensity and direction of the search. This ensures a smooth balance between exploration and exploitation, and stabilizes convergence to the global optimum by controlling the transition dynamics of the solution vector.

IV. DETAILED DESCRIPTION OF THE CORE MODULES OF THE CDMH ALGORITHM

The three core modules of the CDMH algorithm target different elements of dance art, combining mathematical modeling with artistic expression to ensure the accuracy and elegance of the optimization process.

A. Form Constraints

Inspired by ballet posture control, CDMH combines a symmetric loss function with an L2 regularization term to ensure the symmetry and balance of the solution. During the optimization process, the module continuously adjusts the geometry of the solution to ensure the accuracy and stability of the dance structure.

B. Rhythm Changes

Inspired by the rhythm changes in Indian classical dance, CDMH simulates the rhythm changes in dancers' movements through a combination of beat-type polynomials and random perturbation mechanisms. This module helps the algorithm to conduct richer and more diverse searches in the solution space and explore more potential solution paths.

C. Intentional Flow

CDMH is based on the concept of body rhythm in Chinese dance, and controls the balance between global and local search solutions by introducing a flow mechanism in the simulated annealing algorithm. This module flexibly adjusts the optimization process according to the state of the solution through "breathing regulation" to ensure a smooth and elegant search process.

V. EXPERIMENT AND RESULT ANALYSIS

A. Experimental Setting

This paper uses five common optimization objective functions for testing: quadratic function, Rastrigin function, Ackley function, Rosenbrock function, and Griewank function. Five hundred independent experiments were conducted for each objective function, and the average results and convergence time were calculated.

In this study, the Python code used by the researchers for the experiment is as follows:

```
import numpy as np
import random
```

```

import math
import matplotlib.pyplot as plt
import time

#  Three Objective Functions
def quadratic(x, y):
    return x**2 + y**2

def rastrigin(x, y):
    A = 10
    return A * 2 + (x**2 - A * np.cos(2 * np.pi * x))
+ (y**2 - A * np.cos(2 * np.pi * y))

def ackley(x, y):
    return -20 * np.exp(-0.2 * np.sqrt(0.5*(x**2 +
y**2))) - np.exp(0.5*(np.cos(2*np.pi*x) +
np.cos(2*np.pi*y))) + np.e + 20

#  Rosenbrock Function (newly added)
def rosenbrock(x, y):
    return (1 - x)**2 + 100 * (y - x**2)**2

#  Griewank Function (newly added)
def griewank(x, y):
    return 1 + (x**2 + y**2) / 4000 - np.cos(x) *
np.cos(y / np.sqrt(2))

#  CDMH Algorithm
def CDMH(obj_func, n_iter=100, n_agents=10):
    positions = np.random.uniform(-10, 10,
size=(n_agents, 2))
    best_position = None
    best_score = float('inf')
    score_trace = []

    for t in range(n_iter):
        for i in range(n_agents):
            x, y = positions[i]
            x -= 0.01 * x * (1 + np.sin(t/10))
            y -= 0.01 * y * (1 + np.cos(t/10))

            if random.random() < 0.5:
                x += random.choice([-1, 1]) *
np.random.normal(0, 0.3)
            if random.random() < 0.5:
                y += random.choice([-1, 1]) *
np.random.normal(0, 0.3)

            alpha = np.exp(-t / n_iter)
            x += alpha * np.sin(t + i)
            y += alpha * np.cos(t + i)

            score = obj_func(x, y)
            if score < best_score:
                best_score = score
                best_position = (x, y)
                positions[i] = [x, y]
            score_trace.append(best_score)
    return best_score, best_position, score_trace

```

```

#  PSO
def PSO(obj_func, n_iter=100, n_agents=10):
    positions = np.random.uniform(-10, 10,
(n_agents, 2))
    velocities = np.zeros((n_agents, 2))
    pbest = positions.copy()
    pbest_scores = np.array([obj_func(*p) for p in
positions])
    gbest = pbest[np.argmin(pbest_scores)]
    score_trace = []

    for t in range(n_iter):
        for i in range(n_agents):
            r1, r2 = np.random.rand(), np.random.rand()
            velocities[i] = (0.5 * velocities[i] +
1.5 * r1 * (pbest[i] - positions[i])
+
1.5 * r2 * (gbest - positions[i]))
            positions[i] += velocities[i]
            score = obj_func(*positions[i])
            if score < pbest_scores[i]:
                pbest_scores[i] = score
                pbest[i] = positions[i]
            gbest = pbest[np.argmin(pbest_scores)]
            score_trace.append(obj_func(*gbest))
    return obj_func(*gbest), gbest, score_trace

#  SA
def SA(obj_func, n_iter=100):
    current = np.random.uniform(-10, 10, size=2)
    best = current
    best_score = obj_func(*current)
    T = 1.0
    T_min = 0.001
    alpha = 0.95
    score_trace = []

    for t in range(n_iter):
        new = current + np.random.normal(0, 1,
size=2)
        delta = obj_func(*new) - obj_func(*current)
        if delta < 0 or np.exp(-delta / T) >
np.random.rand():
            current = new
            if obj_func(*current) < best_score:
                best_score = obj_func(*current)
                best = current
            T *= alpha
            score_trace.append(best_score)
    return best_score, best, score_trace

#  GA
def GA(obj_func, n_iter=100, n_agents=10):
    population = np.random.uniform(-10, 10,
size=(n_agents, 2))
    score_trace = []

    def crossover(p1, p2):
        return (p1 + p2) / 2

    def mutate(child, rate=0.1):

```

```

if np.random.rand() < rate:
    child += np.random.normal(0, 0.5, size=2)
return child

for t in range(n_iter):
    scores = np.array([obj_func(*ind) for ind in
population])
    best_idx = np.argmin(scores)
    best_score = scores[best_idx]
    best_individual = population[best_idx]
    score_trace.append(best_score)

    selected =
population[np.argsort(scores)[:n_agents//2]]
    children = []
    for _ in range(n_agents - len(selected)):
        p1, p2 = random.sample(list(selected), 2)
        child = crossover(p1, p2)
        child = mutate(child)
        children.append(child)
    population = np.vstack([selected, children])

return best_score, best_individual, score_trace

#  Unified Test Entry
algorithms = {
    "CDMH": CDMH,
    "PSO": PSO,
    "SA": SA,
    "GA": GA
}

functions = {
    "Quadratic": quadratic,
    "Rastrigin": rastrigin,
    "Ackley": ackley,
    "Rosenbrock": rosenbrock,
    "Griewank": griewank
}

#  Run All Combinations and Calculate Averages
for func_name, func in functions.items():
    print(f"\n🔍 Objective Function: {func_name}")
    plt.figure(figsize=(10, 6))

    # Store results for each algorithm
    all_algo_traces = {algo_name: [] for algo_name
in algorithms.keys()}

    for algo_name, algo_func in algorithms.items():
        avg_result = 0
        avg_trace = np.zeros(100)
        avg_time = 0
        # Run 500 experiments
        for _ in range(500):
            start_time = time.time()
            result, position, trace = algo_func(func)
            avg_result += result
            avg_trace += np.array(trace)
            avg_time += (time.time() - start_time)

```

```

avg_result /= 500
avg_trace /= 500
avg_time /= 500

print(f"{algo_name:<5} Avg Result:
{avg_result:.4f}, Time: {avg_time:.2f}s")
all_algo_traces[algo_name] = avg_trace

# Plot convergence curves
plt.plot(all_algo_traces[algo_name],
label=f"{algo_name}")

plt.title(f"Convergence on {func_name} Function
(Average of 500 runs)")
plt.xlabel("Iterations")
plt.ylabel("Best Score")
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()

```

B. Experimental Results

The following is the output of the Python code used by the researchers for the experiment:

🔍 Objective Function: Quadratic
CDMH Avg Result: 0.2305, Time: 0.01s
PSO Avg Result: 0.0000, Time: 0.02s
SA Avg Result: 0.0290, Time: 0.00s
GA Avg Result: 0.0182, Time: 0.01s

🔍 Objective Function: Rastrigin
CDMH Avg Result: 2.2704, Time: 0.02s
PSO Avg Result: 1.6380, Time: 0.03s
SA Avg Result: 4.9066, Time: 0.00s
GA Avg Result: 4.1570, Time: 0.03s

🔍 Objective Function: Ackley
CDMH Avg Result: 1.7774, Time: 0.02s
PSO Avg Result: 0.1353, Time: 0.02s
SA Avg Result: 1.2958, Time: 0.00s
GA Avg Result: 1.0898, Time: 0.02s

🔍 Objective Function: Rosenbrock
CDMH Avg Result: 0.9961, Time: 0.02s
PSO Avg Result: 0.2505, Time: 0.02s
SA Avg Result: 3.1408, Time: 0.00s
GA Avg Result: 5.1189, Time: 0.01s

🔍 Objective Function: Griewank
CDMH Avg Result: 0.0091, Time: 0.02s
PSO Avg Result: 0.0151, Time: 0.03s
SA Avg Result: 0.0324, Time: 0.00s
GA Avg Result: 0.0113, Time: 0.02s

C. Experimental Results Analysis

The ClassicalDance-Metaheuristic algorithm shows good stability and competitiveness in multiple classic objective function tests. Although CDMH is not the best algorithm for quadratic function (a simple convex optimization problem), it is able to obtain a near-optimal solution in a very short time, proving the robustness of the underlying search function. For

the more challenging Rastrigin function, CDMH outperforms SA and GA. Although its performance is slightly inferior to PSO, it has stable performance, strong global search capabilities in multi-peak functions, and is not easy to fall into local optimality.

CDMH can obtain reasonable solutions even for Ackley functions (complex functions with multiple local minima), showing a certain ability to escape from local optimality. For Rosenbrock functions, CDMH performs significantly better than SA and GA. Although it is not as good as PSO, it is much more stable than the other two algorithms and performs well when dealing with narrow and long search spaces.

It can be seen that for the Griewank function, the CDMH algorithm gives the best results among all algorithms, and has obvious advantages for such slow-changing functions with multiple extreme values. The advantages of CDMH can be summarized as follows. Powerful global search ability: It has powerful search capabilities, especially for multi-peak functions such as Rastrigin and Griewank, and is not easy to fall into local optimality. Stable results: The performance of each objective function changes little, which is suitable for optimization tasks with high stability requirements.

Reasonable execution time: A good solution can be obtained without sacrificing too much speed. In other words, the algorithm strikes a good balance between search efficiency and accuracy. Adaptable to a variety of function types: Whether it is a convex optimization function (such as Quadratic), a multi-peak function (such as Ackley, Rastrigin) or a complex path function (such as Rosenbrock), CDMH performs well and shows good generalization ability. In summary, CDMH is a general and stable meta-heuristic algorithm, which outperforms mainstream algorithms for some complex features. It is suitable for various optimization tasks.

VI. CONCLUSION AND OUTLOOK

The CDMH algorithm imitates the learning process of classical dance, introduces artistic heuristic elements on the basis of traditional optimization algorithms, and provides new optimization ideas. Experimental results show that the CDMH algorithm performs well on several standard test functions, especially in terms of global optimization and convergence speed. Future research may further expand the scope of application of the algorithm and explore its potential in fields such as machine learning, creative generation, and artificial intelligence action learning.

REFERENCES

- [1] 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>.
- [2] J.-S. Pan, L.-G. Zhang, R.-B. Wang, V. Snášel, and S.-C. Chu, "Gannet optimization algorithm: a new metaheuristic algorithm for solving engineering optimization problems," *Mathematics and Computers in Simulation*, vol. 202, pp. 343–373, 2022, <https://doi.org/10.1016/j.matcom.2022.06.007>.
- [3] 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, art. 1126450, 2023, <https://doi.org/10.3389/fmech.2022.1126450>.
- [4] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13187–13257, 2023, <https://doi.org/10.1007/s10462-023-10470-y>.
- [5] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: a survey of one decade of research (2009–2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021, <https://doi.org/10.1109/ACCESS.2021.3056407>.
- [6] J.-S. Pan, P. Hu, V. Snášel, and S.-C. Chu, "A survey on binary metaheuristic algorithms and their engineering applications," *Artificial Intelligence Review*, vol. 56, no. 7, pp. 6101–6167, 2023, <https://doi.org/10.1007/s10462-022-10328-9>.
- [7] 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>.
- [8] Y. Gao, "PID-based search algorithm: a novel metaheuristic algorithm based on PID algorithm," *Expert Systems with Applications*, vol. 232, p. 120886, 2023, <https://doi.org/10.1016/j.eswa.2023.120886>.
- [9] S. M. Almufti, A. A. Shaban, Z. A. Ali, R. I. Ali, and J. D. Fuente, "Overview of metaheuristic algorithms," *Polaris Global Journal of Scholarly Research and Trends*, vol. 2, no. 2, pp. 10–32, 2023, <https://doi.org/10.58429/pgjstr.v2n2a144>.
- [10] Z. Meng, B. S. Yıldız, G. Li, C. Zhong, S. Mirjalili, and A. R. Yildiz, "Application of state-of-the-art multiobjective metaheuristic algorithms in reliability-based design optimization: a comparative study," *Structural and Multidisciplinary Optimization*, vol. 66, no. 8, art. 191, 2023, <https://doi.org/10.1007/s00158-023-03639-0>.
- [11] 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, p. 110011, 2023, <https://doi.org/10.1016/j.knsys.2022.110011>.
- [12] B. Toaza and D. Esztergár-Kiss, "A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems," *Applied Soft Computing*, vol. 148, p. 110908, 2023, <https://doi.org/10.1016/j.asoc.2023.110908>.
- [13] V. C. SS and A. HS, "Nature inspired meta heuristic algorithms for optimization problems," *Computing*, vol. 104, no. 2, pp. 251–269, 2022, <https://doi.org/10.1007/s00607-021-00955-5>.
- [14] V. Tomar, M. Bansal, and P. Singh, "Metaheuristic algorithms for optimization: a brief review," *Engineering Proceedings*, vol. 59, no. 1, p. 238, 2024, <https://doi.org/10.3390/engproc2023059238>.
- [15] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems," *Heliyon*, vol. 8, no. 5, 2022, <https://doi.org/10.1016/j.heliyon.2022.e09399>.
- [16] K. Zolfi, "Gold rush optimizer: a new population-based metaheuristic algorithm," *Operations Research and Decisions*, vol. 33, no. 1, pp. 113–150, 2023, <https://doi.org/10.37190/ord230108>.
- [17] X. Wang, V. Snášel, S. Mirjalili, J.-S. Pan, L. Kong, and H. A. Shehadeh, "Artificial Protozoa Optimizer (APO): a novel bio-inspired metaheuristic algorithm for engineering optimization," *Knowledge-Based Systems*, vol. 295, p. 111737, 2024, <https://doi.org/10.1016/j.knsys.2024.111737>.
- [18] V. Lai, Y.-F. Huang, C.-H. Koo, A. N. Ahmed, and A. El-Shafie, "A review of reservoir operation optimisations: from traditional models to metaheuristic algorithms," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 3435–3457, 2022, <https://doi.org/10.1007/s11831-021-09701-8>.
- [19] A. M. Fathollahi-Fard, G. Tian, H. Ke, Y. Fu, and K. Y. Wong, "Efficient multi-objective metaheuristic algorithm for sustainable harvest planning problem," *Computers & Operations Research*, vol. 158, art. 106304, 2023, <https://doi.org/10.1016/j.cor.2023.106304>.
- [20] L. Xie, T. Han, H. Zhou, Z. R. Zhang, B. Han, and A. Tang, "Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization," *Computational Intelligence and Neuroscience*, vol. 2021, p. 9210050, 2021, <https://doi.org/10.1155/2021/9210050>.
- [21] M. Azizi, M. B. Shishehgarhaneh, M. Basiri, and R. C. Moehler, "Squid game optimizer (SGO): a novel metaheuristic algorithm," *Scientific Reports*, vol. 13, no. 1, p. 5373, 2023, <https://doi.org/10.1038/s41598-023-32465-z>.
- [22] M. Dehghani, E. Trojovská, and P. Trojovský, "A new human-based metaheuristic algorithm for solving optimization problems on the base

- of simulation of driving training process,” *Scientific Reports*, vol. 12, no. 1, p. 9924, 2022, <https://doi.org/10.1038/s41598-022-14225-7>.
- [23] H. Y. Chong, H. J. Yap, S. C. Tan, K. S. Yap, and S. Y. Wong, “Advances of metaheuristic algorithms in training neural networks for industrial applications,” *Soft Computing*, vol. 25, no. 16, pp. 11209–11233, 2021, <https://doi.org/10.1007/s00500-021-05886-z>.
- [24] M. Dehghani, G. Bektemyssova, Z. Montazeri, G. Shaikemelev, O. P. Malik, and G. Dhiman, “Lyrebird optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems,” *Biomimetics*, vol. 8, no. 6, p. 507, 2023, <https://doi.org/10.3390/biomimetics8060507>.
- [25] 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>.
- [26] P. Trojovský and M. Dehghani, “A new bio-inspired metaheuristic algorithm for solving optimization problems based on walrus behavior,” *Scientific Reports*, vol. 13, no. 1, p. 8775, 2023, <https://doi.org/10.1038/s41598-023-35863-5>.
- [27] H. T. Sadeeq and A. M. Abdulazeez, “Giant trevally optimizer (GTO): a novel metaheuristic algorithm for global optimization and challenging engineering problems,” *IEEE Access*, vol. 10, pp. 121615–121640, 2022, <https://doi.org/10.1109/ACCESS.2022.3223388>.
- [28] M. Khan, Z. Ullah, O. Mašek, S. R. Naqvi, and M. N. A. Khan, “Artificial neural networks for the prediction of biochar yield: a comparative study of metaheuristic algorithms,” *Bioresour. Technology*, vol. 355, art. 127215, 2022, <https://doi.org/10.1016/j.biortech.2022.127215>.
- [29] J. Pasha, A. L. Nwodu, A. M. Fathollahi-Fard, G. Tian, Z. Li, H. Wang, and M. A. Dulebenets, “Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings,” *Advanced Engineering Informatics*, vol. 52, p. 101623, 2022, <https://doi.org/10.1016/j.aei.2022.101623>.
- [30] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, “Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for solving optimization problems,” *International Journal of Intelligent Systems*, vol. 36, no. 10, pp. 5887–5958, 2021, <https://doi.org/10.1002/int.22535>.
- [31] M. Dehghani, E. Trojovská, and T. Zušćák, “A new human-inspired metaheuristic algorithm for solving optimization problems based on mimicking sewing training,” *Scientific Reports*, vol. 12, no. 1, p. 17387, 2022, <https://doi.org/10.1038/s41598-022-22458-9>.
- [32] L. Xie, T. Han, H. Zhou, Z. Zhang, B. Han, A. Tang, “Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization,” *Computational Intelligence and Neuroscience*, vol. 2021, p. 9210050, 2021, <https://doi.org/10.1155/2021/9210050>.
- [33] V. Sharma and A. K. Tripathi, “A systematic review of meta-heuristic algorithms in IoT based application,” *Array*, vol. 14, p. 100164, 2022, <https://doi.org/10.1016/j.array.2022.100164>.
- [34] P. Trojovský and M. Dehghani, “Subtraction-average-based optimizer: a new swarm-inspired metaheuristic algorithm for solving optimization problems,” *Biomimetics*, vol. 8, no. 2, p. 149, 2023, <https://doi.org/10.3390/biomimetics8020149>.
- [35] M. Dehghani *et al.*, “Kookaburra optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems,” *Biomimetics*, vol. 8, no. 6, p. 470, 2023, <https://doi.org/10.3390/biomimetics8060470>.
- [36] M. Azizi, M. B. Shishehgharkhaneh, M. Basiri, R. C. Moehler, “Squid game optimizer (SGO): a novel metaheuristic algorithm,” *Scientific Reports*, vol. 13, no. 1, p. 5373, 2023, <https://doi.org/10.1038/s41598-023-32465-z>.
- [37] M. Khan, Z. Ullah, O. Mašek, S. R. Naqvi, M. N. Aslam Khan, “Artificial neural networks for the prediction of biochar yield: a comparative study of metaheuristic algorithms,” *Bioresour. Technology*, vol. 355, p. 127215, 2022, <https://doi.org/10.1016/j.biortech.2022.127215>.
- [38] O. Al-Baik *et al.*, “Pufferfish optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems,” *Biomimetics*, vol. 9, no. 2, p. 65, 2024, <https://doi.org/10.3390/biomimetics9020065>.
- [39] H. Y. Chong, H. J. Yap, S. C. Tan, K. S. Yap, S. Y. Wong, “Advances of metaheuristic algorithms in training neural networks for industrial applications,” *Soft Computing*, vol. 25, no. 16, pp. 11209–11233, 2021, <https://doi.org/10.1007/s00500-021-05886-z>.
- [40] J. Pasha *et al.*, “Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings,” *Advanced Engineering Informatics*, vol. 52, p. 101623, 2022, <https://doi.org/10.1016/j.aei.2022.101623>.
- [41] S. S. Kareem, R. R. Mostafa, F. A. Hashim, H. M. El-Bakry, “An Effective Feature Selection Model Using Hybrid Metaheuristic Algorithms for IoT Intrusion Detection,” *Sensors*, vol. 22, no. 4, p. 1396, 2022, <https://doi.org/10.3390/s22041396>.
- [42] 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>.
- [43] M. A. Akbari, M. Zare, R. Azizpanah-abarghoee, S. Mirjalili, M. Deriche, “The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems,” *Scientific Report*, vol. 12, no. 1, p. 10953, 2022, <https://doi.org/10.1038/s41598-022-14338-z>.
- [44] V. Osuna-Enciso, E. Cuevas, and B. M. Castañeda, “A diversity metric for population-based metaheuristic algorithms,” *Information Sciences*, vol. 586, pp. 192–208, 2022, <https://doi.org/10.1016/j.ins.2021.11.073>.
- [45] F. S. Gharehchopogh, I. Maleki, and Z. A. Dizaji, “Chaotic vortex search algorithm: metaheuristic algorithm for feature selection,” *Evolutionary Intelligence*, vol. 15, no. 3, pp. 1777–1808, 2022, <https://doi.org/10.1007/s12065-021-00590-1>.
- [46] K. Zolfi, “Gold rush optimizer: a new population-based metaheuristic algorithm,” *Operations Research and Decisions*, vol. 33, no. 1, pp. 113–150, 2023, <https://doi.org/10.37190/ord230108>.
- [47] M. Dehghani, E. Trojovská, P. Trojovský, O. P. Malik, “OOBO: a new metaheuristic algorithm for solving optimization problems,” *Biomimetics*, vol. 8, no. 6, p. 468, 2023, <https://doi.org/10.3390/biomimetics8060468>.
- [48] X. Wang *et al.*, “Artificial Protozoa Optimizer (APO): a novel bio-inspired metaheuristic algorithm for engineering optimization,” *Knowledge-Based Systems*, vol. 295, p. 111737, 2024, <https://doi.org/10.1016/j.knsys.2024.111737>.
- [49] J. Priyadarshini, M. Premalatha, R. Čep, M. Jayasudha, K. Kalita, “Analyzing physics-inspired metaheuristic algorithms in feature selection with K-nearest-neighbor,” *Applied Sciences*, vol. 13, no. 2, p. 906, 2023, <https://doi.org/10.3390/app13020906>.
- [50] A. M. Fathollahi-Fard *et al.*, “Efficient multi-objective metaheuristic algorithm for sustainable harvest planning problem,” *Computers & Operations Research*, vol. 158, p. 106304, 2023, <https://doi.org/10.1016/j.cor.2023.106304>.
- [51] M. Kaveh and M. S. Mesgari, “Application of meta-heuristic algorithms for training neural networks and deep learning architectures: a comprehensive review,” *Neural Processing Letters*, vol. 55, no. 4, pp. 4519–4622, 2023, <https://doi.org/10.1007/s11063-022-11055-6>.
- [52] I. Faridmehr, M. L. Nehdi, I. F. Davoudkhani, A. Poolad, “Mountaineering team-based optimization: a novel human-based metaheuristic algorithm,” *Mathematics*, vol. 11, no. 5, p. 1273, 2023, <https://doi.org/10.3390/math11051273>.
- [53] H. I. Shaheen, G. I. Rashed, B. Yang, J. Yang, “Optimal electric vehicle charging and discharging scheduling using metaheuristic algorithms: V2G approach for cost reduction and grid support,” *Journal of Energy Storage*, vol. 90, p. 111816, 2024, <https://doi.org/10.1016/j.est.2024.111816>.
- [54] N. D. Lagaros, V. Plevris, and N. A. Kallioras, “The mosaic of metaheuristic algorithms in structural optimization,” *Archives of Computational Methods in Engineering*, vol. 29, no. 7, pp. 5457–5492, 2022, <https://doi.org/10.1007/s11831-022-09773-0>.
- [55] 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, 2024, <https://doi.org/10.22266/ijies2024.0831.43>.
- [56] C. S. W. Ng, M. N. Amar, A. J. Ghahfarokhi, L. S. Imsland, “A survey on the application of machine learning and metaheuristic algorithms for intelligent proxy modeling in reservoir simulation,” *Computers & Chemical Engineering*, vol. 170, p. 108107, 2023, <https://doi.org/10.1016/j.compchemeng.2022.108107>.

- [57] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, 2018, <https://doi.org/10.1007/s00500-016-2474-6>.
- [58] S. Rahapsari, "The quest of finding the self in the Bedhaya: unravelling the psychological significance of the Javanese sacred dance," *Culture & Psychology*, vol. 28, no. 3, pp. 413–432, 2022, <https://doi.org/10.1177/1354067X211047441>.
- [59] M. S. Dwarika and H. M. Haraldsen, "Mental health in dance: a scoping review," *Frontiers in Psychology*, vol. 14, p. 1090645, 2023, <https://doi.org/10.3389/fpsyg.2023.1090645>.
- [60] J. R. Challapalli and N. Devarakonda, "A novel approach for optimization of convolution neural network with hybrid particle swarm and grey wolf algorithm for classification of Indian classical dances," *Knowledge and Information Systems*, vol. 64, no. 9, pp. 2411–2434, 2022, <https://doi.org/10.1007/s10115-022-01707-3>.
- [61] L. Androschuk, "The evolution of classical dance in the context of the development of a dancer and athlete," *Theory and methodology of physical education and sports*, no. 2, pp. 54–58, 2022, <https://doi.org/10.32652/tmfvs.2022.2.54-58>.
- [62] M. Joshi and S. Chakrabarty, "An extensive review of computational dance automation techniques and applications," *Proceedings of the Royal Society A*, vol. 477, no. 2251, p. 20210071, 2021, <https://doi.org/10.1098/rspa.2021.0071>.
- [63] N. Jain, *et al.*, "An enhanced deep convolutional neural network for classifying Indian classical dance forms," *Applied Science*, vol. 11, no. 14, p. 6253, 2021, <https://doi.org/10.3390/app11146253>.
- [64] R. Wen, "Strategies for cultivating students' performance skills in classical dance teaching from the perspective of aesthetic education," *SHS Web of Conferences*, vol. 174, 2023, <https://doi.org/10.1051/shsconf/202317402003>.
- [65] Q. Zhou, *et al.*, "Application development of dance pose recognition based on embedded artificial intelligence equipment," *Journal of Physics: Conference Series*, vol. 1757, no. 1, 2021, <https://doi.org/10.1088/1742-6596/1757/1/012011>.