ChinaRxiv

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-201901.00203

Postprint of Dynamic Adaptive Cuckoo Search Algorithm Based on Dimension-by-Dimension Opposition-Based Learning

Authors: Huang Minming, He Qing, Wenxi

Date: 2019-01-28T00:00:00+00:00

Abstract

To address the deficiencies of the Cuckoo Search algorithm (CS), such as low optimization accuracy, slow convergence speed, insufficient search vitality in later stages, and dimensional interference when handling high-dimensional optimization problems, a Dynamically Adaptive Cuckoo Search algorithm based on Dimension-wise Opposition-based Learning strategy (DA-DOCS) is proposed. First, dimension-wise opposition-based learning is performed on solutions after selection and update to reduce dimensional interference and expand population diversity; then, an elitist preservation mechanism is used to evaluate the results, enhancing the algorithm' s optimization capability; finally, the information of current solutions is fully utilized for dynamically adaptive scaling factor control to guide solutions toward rapid convergence and improve the algorithm' s search vitality. Experimental results demonstrate that, compared with the standard Cuckoo Search algorithm, the proposed algorithm exhibits improvements in optimization accuracy, convergence speed, and search vitality in later stages, and also possesses certain competitive advantages when compared with other improved algorithms.

Full Text

Dynamically Adaptive Cuckoo Search Algorithm Based on Dimension-by-Dimension Opposition-Based Learning

Huang Minming[1] , He Qing[1] , Wen Xi[2] 1. Guizhou University, a. College of Big Data & Information Engineering; b. Guizhou Provincial Key Laboratory of Public Big Data, Guiyang 550025, China 2. College of AI & Data Science, Hebei University of Technology, Tianjin 300000, China

Abstract: To address the limitations of the Cuckoo Search (CS) algorithm, including low optimization precision, slow convergence speed, insufficient

search vitality in later stages, and dimensional interference when handling high-dimensional optimization problems, this paper proposes a Dynamically Adaptive Cuckoo Search algorithm based on Dimension-by-dimension Opposition-based Learning (DA-DOCS). First, dimension-by-dimension opposition-based learning is applied to selected updated solutions to reduce inter-dimensional interference and expand population diversity. Then, an elite preservation mechanism evaluates these results to enhance the algorithm' s optimization capability. Finally, information from current solutions is fully utilized for dynamically adaptive scaling factor control, guiding rapid convergence and improving search vitality. Experimental results demonstrate that compared with the standard CS algorithm, the proposed algorithm achieves higher optimization precision, faster convergence speed, and greater search vitality in later stages, while also showing competitive advantages over other improved algorithms.

Keywords: Cuckoo search algorithm; opposition-based learning; function optimization; inter-dimensional interference; dynamic adaptation

0 Introduction

The Cuckoo Search algorithm (CS) [**?**], proposed by Yang and Deb in 2009, is a novel global search algorithm inspired by the brood parasitism behavior of cuckoos and the Lévy flights behavior of birds. Due to its high efficiency, few parameters, and ease of implementation, CS has been widely applied to engineering and function optimization problems, becoming a new highlight in heuristic intelligent algorithms [**?**, **?**]. However, the CS algorithm also suffers from weaknesses such as poor convergence capability and insufficient search vitality [**?**, **?**], prompting scholars worldwide to conduct further research and improvements.

Wang et al. [**?**] analyzed the Markov chain model of the CS algorithm to study its state transition process and proved its convergence. Chen et al. [**?**] improved the CS algorithm by incorporating a Logistic model to automatically adjust the step size control factor and discovery probability, thereby enhancing global search ability. Experimental results demonstrated the improved algorithm' s superior performance. Pauline et al. [**?**] proposed a self-adaptive Cuckoo algorithm with two-parent crossover and validated its effectiveness through experiments. Ma et al. [**?**] combined an elite population candidate solution pool with Powell' s local search strategy, showing through experiments that their improved algorithm possesses certain competitive advantages.

These research achievements have enriched the improvement efforts on the CS algorithm and effectively enhanced its convergence speed and precision. However, most improved algorithms adopt a holistic update-then-evaluate strategy, where all dimensional information of a solution is updated first before evaluation via the objective function. This evaluation approach causes interference among dimensions, deteriorating convergence efficiency and speed. Wang et

al. [**?**] proposed a dimension-by-dimension improved CS algorithm that employs a dimension-wise update-and-evaluate strategy to strengthen evolutionary dimension information and reduce inter-dimensional interference. While experimental results validated its effectiveness, the dimension-by-dimension strategy consumes numerous evaluation times in the early convergence stage when handling high-dimensional complex functions, leading to slower convergence efficiency and insufficient vitality in later stages.

Inspired by dimension-by-dimension improvement strategies and various hybrid improvement algorithms [**?**], this paper proposes a Dynamically Adaptive Cuckoo Search algorithm based on Dimension-by-dimension Opposition-based Learning (DA-DOCS) to address the slow convergence, low solution precision, poor search vitality, and difficulty in converging for high-dimensional complex functions in standard CS. First, during algorithm iteration, a dimension-by-dimension opposition-based learning strategy is employed to reduce mutual interference among dimensions and expand the search space through opposition-based learning, thereby improving search efficiency. Second, an elite preservation method evaluates solutions after dimension-by-dimension opposition-based learning to enhance search speed. Finally, information from current solutions is utilized for dynamically adaptive control of the scaling factor to guide solution convergence dynamically and improve optimization capability. This paper selects eight standard test functions for experimental simulation, and results show that compared with the CS algorithm, the improved DA-DOCS algorithm achieves better convergence precision, speed, and efficiency, particularly outperforming CS on high-dimensional optimization problems. Compared with other improved CS algorithms, DA-DOCS demonstrates stronger search vitality and optimization capability.

1.1 Cuckoo Search (CS) Algorithm

The fundamental principle of the Cuckoo algorithm maps the parasitic nest locations to solutions in the algorithm' s population space, using nest location quality as fitness values. To simulate the cuckoo reproduction mechanism, the algorithm establishes three idealized rules [**?**]:

Rule 1: Each cuckoo lays one egg at a time in a randomly chosen host bird nest.
Rule 2: The best nest with the highest quality egg is carried over to the next generation.
Rule 3: The number of available host nests is fixed, and host birds discover cuckoo eggs with probability P_a.

Based on these rules, after initialization, the CS algorithm updates nest positions through Lévy flights as follows [**?**]:

$$x_i^{t+1} = x_i^t + \alpha \otimes \text{Lévy}(\beta)$$

where x_i^{t+1} represents the updated nest position, x_i^t denotes the current nest position, and α is the step size scaling factor, typically set to 0.01 [**?**].

The Lévy flight random search path is calculated as:

$$\text{Lévy}(\beta) \sim \frac{u}{|v|^{1/\beta}} \times \Phi$$

where u and v are standard normal random variables, β is the Lévy flight control factor (usually 1.5 [**?**]), and Φ is computed as:

$$\Phi = \left[\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}} \right]^{1/\beta}$$

After nest updating, a uniformly distributed random number $R \in (0, 1)$ is compared with the egg discovery probability P_a. If $R \geq P_a$, poor nest positions are abandoned and reconstructed using equation (4); otherwise, the current position is retained. The position update [**?**] is:

$$x_i^{t+1} = x_i^t + r \times (x_j^t - x_k^t)$$

where r is a scaling factor uniformly distributed in $(0, 1)$, and x_j^t and x_k^t represent two random solutions in generation t.

1.2 Opposition-Based Learning

Opposition-based learning (OBL), proposed by Tizhoosh in 2005 [**?**], is a novel technique in intelligent computing. Its principle involves simultaneously evaluating a feasible solution and its opposite solution, selecting the better one [**?**].

Definition 1: If x is any real number in interval $[a, b]$, its opposition-based number is defined as [**?**]:

$$x' = a + b - x$$

The distance from x to a is $x - a$, and from x' to b is $b - x'$. According to equation (5), $b - x' = b - (a + b - x) = x - a$, so these distances are equal, as shown in Figure 1.

Definition 2: If $P = (x_1, x_2, ..., x_D)$ is any point in D-dimensional space with $x_i \in [a_i, b_i]$, its opposition-based solution is [**?**]:

$$P' = (x_1', x_2', ..., x_D'), \quad \text{where} \quad x_i' = a_i + b_i - x_i$$

Since feasible solutions and opposition-based points lie on opposite sides of the search space, simultaneously searching two symmetric spaces expands the search space and improves search efficiency.

2.1 Dimension-by-Dimension Opposition-Based Learning Strategy

In standard CS algorithms for high-dimensional complex functions, updating all dimensional information before evaluating fitness via the objective function causes interference among dimensions, masking evolutionary dimension information and affecting convergence speed and optimization precision. The proposed dimension-by-dimension opposition-based learning strategy effectively avoids inter-dimensional interference while expanding population search range, exploring new search spaces, and enhancing population diversity. In each generation, it resembles independent evolution of each dimension, simultaneously searching two symmetric regions in every dimension, improving both convergence efficiency and global optimization capability.

In this strategy, let population size be N and dimension be D. The position matrix when selecting and updating positions during iteration is $x_{nest} \in \mathbb{R}^{N\times D}$. The upper and lower bounds for each dimension are $x_{up} \in \mathbb{R}^{N\times D}$ and $x_{low} \in \mathbb{R}^{N\times D}$, respectively. The opposition-based position matrix for each dimension is:

$$x'_{nest} = x_{up} + x_{low} - x_{nest}$$

During algorithm iteration, after abandoned eggs are reconstructed and replaced, they merge with retained eggs. Instead of using the holistic update-then-evaluate approach from CS, the strategy applies opposition-based learning to each dimension of updated solutions. Updated dimensional values are mapped to their opposition-based numbers for spatial search, eliminating degenerate dimension information. This not only reduces inter-dimensional interference but also expands solution space search range, thereby improving search efficiency and optimization capability. The dimension-wise evolutionary learning strategy considers update information for each dimension. After a dimension' s value undergoes opposition-based learning, it combines with other dimensions' values to form a new solution, which is evaluated via the objective function. If the current solution' s quality improves, the opposition-based update result for that dimension is retained; otherwise, the update is abandoned and the pre-opposition dimension information is preserved. This elite preservation approach continues dimension-by-dimension opposition-based learning updates until all dimensions are processed.

Based on this strategy, evolutionary dimension information is strengthened, reducing evaluation times consumed by random updates in standard CS, improving convergence speed and optimization capability. The elite preservation

method enhances algorithm efficiency.

2.2 Dynamic Adaptation

During standard CS iteration, when random probability R exceeds survival probability P_a, eggs are abandoned and new eggs are constructed using equation (4) to replace them. However, standard CS construction updates based on two random solutions in the solution space exhibit high randomness and instability without fully utilizing current solution information for dynamic adaptive iteration updates, hindering local search in the dimension-by-dimension opposition-based learning strategy. Additionally, the scaling factor r in equation (4) is a uniformly distributed random number in $(0, 1)$, limiting optimization performance. To improve algorithm performance, the scaling factor' s value can be adaptively adjusted according to iteration times, showing a decreasing trend. This enables escaping local optima in early search stages to avoid premature convergence while enhancing local optimization capability in later stages. The scaling factor fully utilizes current solution information for dynamic adaptation, dynamically guiding solution convergence and improving search vitality. Therefore, DA-DOCS modifies equation (4) as:

$$x_i^{t+1} = \begin{cases} x_i^t + r \times (x_i^t - x_k^t) & \text{if } R \geq P_a \\ x_i^t & \text{otherwise} \end{cases}$$

where x_k^t represents a random solution in generation t, and t denotes the current iteration number.

2.3 DA-DOCS Algorithm Flow

The DA-DOCS algorithm procedure consists of five main steps:

a) Initialization: Define the objective function $f(x)$, set parameters including population size N, search space dimension D, maximum discovery probability P_a, maximum iterations T_{max}, and precision threshold δ, then randomly generate N initial nest positions x_i $(i = 1, 2, ..., N)$.

b) Random Walk: Select the objective function and calculate fitness for each nest position, retaining the best nest position. Generate new nest positions via Lévy flight random walk.

c) Selective Update: Compare uniformly distributed random number $R \in (0, 1)$ with discovery probability P_a. If $R \geq P_a$, discard poor nest positions and update using equation (8); otherwise, retain the current solution. Calculate objective function fitness values and carry over better nests to the next generation.

d) Dimension-by-Dimension Opposition-Based Update: For solutions retained in step c), apply dimension-by-dimension opposition-based learning

using equation (7), considering update information for each dimension. After opposition-based learning, combine the current dimension' s value with other dimensions to form a new solution and evaluate it. If current solution quality improves, retain the opposition-based update result for that dimension; otherwise, abandon the current dimension' s update and preserve its pre-opposition value. Continue this elite preservation approach for subsequent dimensions until all dimensions are updated.

e) Termination Criterion: Check if current solution quality meets algorithm termination conditions. If satisfied, terminate and output the final solution; otherwise, return to step b) until termination conditions are met. The improved algorithm flowchart is shown in Figure 2.

2.4 Complexity Analysis of the Improved Algorithm

According to literature [**?**, **?**], the time complexity of the Cuckoo Search algorithm is $O(N \times D \times T_{max})$, where N is population size, D is dimension, and T_{max} is maximum iterations. When the objective function computation time $O(f(n))$ is higher-order, complexity becomes $O(N \times D \times T_{max} \times f(n))$; when same-order or lower-order, complexity is $O(N \times D \times T_{max})$ [**?**]. The proposed dimension-by-dimension opposition-based learning strategy adds $O(N \times D)$ operations. Additionally, to improve search vitality, the dynamic adaptation mechanism introduces an inner iteration layer. Thus, the improved algorithm' s time complexity becomes $O(N \times D \times T_{max} \times (f(n) + D))$. Space complexity S is primarily affected by population size N and search space dimension D, expressed as $S = O(N \times D)$. The proposed algorithm exhibits higher complexity than standard CS, with both time and space complexity increasing with dimension.

3.1 Test Functions and Experimental Parameters

Experiments were conducted using MATLAB R2016a on a 64-bit Windows 7 system with an Intel Core i5-6500 processor. To validate DA-DOCS performance, eight standard test functions were employed (Table 1). Functions F_1-F_3 are unimodal, while F_4-F_8 are multimodal functions containing numerous local minima in the solution space, with different theoretical minima for different dimensions [**?**].

Fitness error metric (12) evaluates algorithm precision convergence capability:

$$\delta = |f(x) - f(x^*)|$$

where $f(x)$ is the obtained solution fitness and $f(x^*)$ is the known optimal solution. Smaller error values indicate better solution quality.

Testing focuses on two aspects: **a)** analyzing solution quality, convergence speed, and dimensional variation for DA-DOCS versus standard CS to test optimization precision, convergence speed, and high-dimensional performance; **b)** com-

paring optimization precision and performance with other improved CS algorithms on fixed-dimension functions.

3.2.1 Solution Quality Analysis

To observe DA-DOCS solution quality, population size $N = 30$, maximum iterations $T_{max} = 5000$, discovery probability $P_a = 0.25$, and Lévy flight step scaling factors $\alpha = 0.1, 1.3$ [**?**] were set. Table 2 shows results from 30 independent runs of DA-DOCS and standard CS, comparing mean fitness error and standard deviation, with best results in bold.

Table 2 indicates DA-DOCS generally improves solution quality over standard CS. For unimodal functions, solution quality improves significantly. For multimodal functions with numerous local optima, DA-DOCS also performs excellently. Notably, CS fails to converge on F_4 and F_5, while DA-DOCS achieves global optima on both. Thus, DA-DOCS demonstrates superior solution precision compared to standard CS.

3.2.2 Convergence Speed Analysis

To analyze DA-DOCS convergence speed, population size $N = 30$, discovery probability $P_a = 0.25$, and step scaling factors $\alpha = 0.1, 1.3$ [**?**] were used. Eight test functions underwent 30 independent experiments comparing mean function evaluation counts (FES) and standard deviations for convergence to specified precision thresholds (Table 3). "-" indicates failure to converge to the threshold, with best results in bold. Figure 3 [Figure 3: see original paper] shows convergence curves.

Standard CS's holistic evaluation-then-update strategy suffers from dimensional interference, wasting evaluation times and yielding slower convergence [**?**]. The proposed dimension-by-dimension opposition-based learning strategy, while consuming certain evaluation times, enhances local refinement capability for better solutions, accelerating convergence. Table 3 shows CS fails to converge on F_3, F_4, F_5 within specified thresholds, while DA-DOCS converges on all functions with significantly lower mean FES and standard deviation. Thus, the improved algorithm achieves faster convergence and better robustness.

Figures 3(a)-(h) plot iteration count versus log fitness values, graphically illustrating the optimization process. For both simple unimodal and complex multimodal functions, DA-DOCS shows stronger early-stage convergence and greater late-stage search vitality. Particularly in Figures 3(d) and 3(e), DA-DOCS rapidly converges to theoretical optima. The improved algorithm compensates for standard CS' s slow convergence and weak late-stage local convergence, demonstrating better exploitation and exploration capabilities, validating Table 3 results.

3.2.3 Dimension Variation Analysis

To analyze dimension variation impact, Table 4 compares mean iteration counts and standard deviations for CS and DA-DOCS across different dimensions converging to identical precision thresholds. Parameters: $N = 30$, $P_a = 0.25$, $\alpha = 0.1, 1.3$ [**?**]. "-" indicates failure to converge within $1000 \times D$ maximum iterations, with best results in bold.

When dimension increases from 50 to 100, CS' s average iterations more than double, while DA-DOCS' s increase remains within 1x. At $D = 100$, CS fails to converge completely, whereas DA-DOCS converges all functions to specified thresholds. The dimension-by-dimension opposition-based strategy and dynamic adaptation better avoid dimensional interference, showing superior global optimization capability in high-dimensional function solving, effectively improving global search, local exploitation, and overall solution effectiveness, efficiency, and search vitality.

3.3 Comparison with Other Improved CS Algorithms

To further validate performance, five test functions were compared against DDICS [**?**] and DSCS [**?**] algorithms (Table 5). Comparisons include mean fitness error, best value, standard deviation, and runtime, with best results in bold. Parameters: $N = 30$, $P_a = 0.25$, $T_{max} = 5000$, $D = 20$, 30 independent runs per algorithm.

Table 5 shows varying performance across different functions. DA-DOCS converges to theoretical minima on F_1, demonstrating strong optimization capability with significant advantages in mean and maximum fitness error. For F_5, DA-DOCS improves by nearly 40 orders of magnitude over other algorithms. While DA-DOCS runtime exceeds DSCS, it reduces time by 69.68% compared to DDICS. Figure 4 [Figure 4: see original paper] shows DA-DOCS achieves better solution precision, convergence speed, and late-stage vitality versus DDICS and DSCS, validating Table 5 results.

4 Conclusion

This paper proposes DA-DOCS to address CS algorithm limitations. Improvements include: **a)** incorporating dimension-by-dimension opposition-based learning with elite preservation evaluation to enhance population diversity, expand search range, strengthen evolutionary dimension information, and reduce dimensional interference, yielding better optimization precision and speed for high-dimensional functions; **b)** utilizing current solution information for dynamically adaptive scaling factor control, enabling previous generation' s optimal solution information to influence current solution update direction and enhance search vitality. Experimental simulations on eight standard test functions demonstrate that DA-DOCS achieves superior convergence speed, optimization capability, and search vitality for high-dimensional problems

compared to standard CS, and exhibits competitive convergence precision and vitality versus other improved CS algorithms.

References

[1] Yang Xinshe, Deb S. Cuckoo Search via Lévy flights [C]// Proc of World Congress on Nature & Biologically Inspired Computing. Piscataway: IEEE Publications, 2009: 210-214.

[2] Jiang Minlan, Luo Jingyuan, Jiang Dingde, et al. A cuckoo search-support vector machine model for predicting dynamic measurement errors of sensors [J]. IEEE Access, 2017, 4 (99): 5030-5037.

[3] Wang Lijing, Zhong Yiwen, Yin Yilong. Nearest neighbor cuckoo search algorithm with probabilistic mutation [J]. Elsevier Science Publishers B. V. Amsterdam, 2016, 49: 498-509.

[4] Lan Shaofeng, Liu Sheng. Overview of research on Cuckoo search algorithm [J]. Computer Engineering and Design, 2015, 36 (4): 1063-1067.

[5] Yang Huihua, Wang Ke, Li Lingqiao, et al. K-means clustering algorithm based on adaptive Cuckoo search and its application [J]. Journal of Computer Applications, 2016, 36 (8): 2066-2070.

[6] Wang Fan, He Xingshi, Wang Yan, et al. Markov model and convergence analysis based on Cuckoo search algorithm [J]. Computer Engineering, 2012, 38 (11): 180-185.

[7] Chen Hua, Zhang Yidan. Adaptive Cuckoo algorithm based on logistic model [J]. Computer Engineering and Applications, 2015, 51 (20): 31-35.

[8] Pauline O, Zarita Z, Chee K S, et al. Adaptive Cuckoo search algorithm with two-parent crossover for solving optimization problems [J]. Advances in Swarm and Computational Intelligence Springer International, 2015, 6 (2): 427-435.

[9] Ma Wei, Sun Zhengxing, Li Junlou. Cuckoo search algorithm based on powell local search method for global optimization [J]. Application Research of Computers, 2015, 32 (6): 1667-1775.

[10] Wang Lijin, Yin Yilong, Zhong Yiwen. Cuckoo search algorithm with dimension by dimension improvement [J]. Journal of Software, 2013, 24 (11): 2687-2698.

[11] Wang Jun, Zhou Bihua, Zhou Shudao. An improved Cuckoo search optimization algorithm for the problem of chaotic systems parameter estimation [J]. Computational Intelligence and Neuroscience, 2016, 8 (2): 1-8.

[12] Gao Yunlong, Yan Peng. Unified optimization based on multi-swarm PSO algorithm and Cuckoo search algorithm [J]. Control and Decision, 2016, 31 (4): 601-608.

[13] Mlakar U, Fister I J, Fister I. Hybrid self-adaptive Cuckoo search for global optimization [J]. Swarm & Evolutionary Computation, 2016, 29: 1-8.

[14] Tizhoosh H R. Opposition-based learning: a new scheme for machine intelligence [C]// Proc of Computational Intelligence for Modelling. Vienna, Austria: Computer Society, 2005, 1: 695-701.

[15] Wei Wenhong, Zhou Jianlong, Fang Chen, et al. Constrained differential evolution using generalized opposition-based learning [J]. Acta Electronica Sinica, 2016, 20 (11): 4413-4437.

[16] Zhang Sen, Luo Qifang, Zhou Yongquan. Hybrid grey wolf optimizer using elite opposition-based Learning strategy and simplex method [J]. International Journal of Computational Intelligence and Applications, 2017, 16 (2): 1-12.

[17] Ahandani M A, Alavi-Rad H. Opposition-based learning in shuffled frog leaping: an application for parameter identification [J]. Information Sciences. 2015, 291 (291): 19-42.

[18] Zhang Yongwei, Wang Lei, Wu Qidi. Dynamic adaptation Cuckoo search algorithm [J]. Control and Decision, 2014, 29 (4): 617-622.

[19] Elkeran A. A new approach for sheet nesting problem using guided Cuckoo search and pairwise clustering [J]. European Journal of Operational Research, 2013, 231 (3): 757-769.

[20] Xiao Hailin, Zhang Wenjuan, Nie Zaiping, et al. User selection based on Cuckoo search algorithm and Interference Alignment [J]. Journal of University of Electronic Science and Technology of China, 2017, 46 (6): 801-806.

[21] Fan Shuaijun. Application research and improvement of Cuckoo search algorithm [D]. Chengdu: Southwest Jiaotong University, 2016.

[22] Cai Zefan, Yang Xiaodong. Cuckoo Search Algorithm with deep search [C]// Proc of the 3rd IEEE International Conference on Computer and Communications. Chengdu: IEEE Publications, 2018: 2241-2246.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.