Applied Mathematics and Computation 266 (2015) 1083–1092

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Cuckoo search algorithm based on frog leaping local search and chaos theory

Xueying Liu [a,b,*], Meiling Fu [b]

[a] College of Arts and Sciences, Shanghai Maritime University, Shanghai 201306, China
[b] College of Science, Inner Mongolia University of Technology, Hohhot 010051, China

ARTICLE INFO

Keywords:
Cuckoo search
Chaos theory
Frog leaping algorithm
Inertia weight

ABSTRACT

Cuckoo algorithm is a novel optimization algorithm in the field of heuristic intelligence algorithms. Given the strong random leaping in solution space search, careful local searches are susceptible to falling into the local optimum. Thus, the latter phase of the optimization slows down and the accuracy diminishes. To improve the performance of the algorithm, this paper proposes an improved cuckoo search that utilizes chaos theory to enhance the variety of the initial population. Then, this study introduces inertia weight into the Lévy flight random search to improve global searching capability. Finally, it applies the local search mechanism of the frog leaping algorithm to enhance local search and further improve the search speed and convergence precision of the algorithm. Typical test functions are employed to verify the performance of the improved algorithm. Comparison results with other algorithms indicate that the improved algorithm displays strong optimizing accuracy and high speed. Furthermore, this algorithm is confirmed to be convergent.

1. Introduction

Swarm intelligence algorithms are a new type of evolutionary algorithm. These algorithms include genetic algorithms [1], difference evolution (DE) [2], particle swarm optimization [3], shuffled frog leaping algorithms (SFLA) [4], and monkey algorithms [5], which are inspired by natural laws, as well as the intelligent behavior of biological populations. These algorithms have demonstrated their unique capacities, as well as their applicability in science and engineering technology. Each intelligence algorithm corresponds to one practical heuristic source. For example, the biogeography algorithm [6,7] applies a global optimization method according to the distribution characteristics of biology in geography. Glowworm swarm optimization [8] is a random optimization algorithm that simulates natural bioluminescence behavior. Wolf pack algorithm [9] is a leadership strategy-based algorithm that follows the predation tendencies of wolf packs. Therefore, the remarkable concept and wide applicability of intelligent optimization algorithms are consistently explored and developed by numerous researchers.

Cuckoo search (CS) was proposed by Cambridge scholars Yang Xin She and Deb Suash. This global search algorithm [10] is inspired by the behaviors of cuckoos in locating nests and laying eggs [11] and by the Lévy flight of insects [12]. CS is simple and operates under few controlled parameters, optimal search paths, strong optimizing capability, and ease of use. Considering these advantages, CS has been widely applied in practical engineering optimization problems. Nonetheless, fundamental CS exhibits slow convergence rate and low convergence precision. Thus, many scholars have developed various improvement methods. For example, Literature [13] enhanced CS by solving a function optimization problem. Literature [14] initiated a CS based on Gaussian

* Corresponding author. Tel.: +86 13664882077.
E-mail address: xyliu@aliyun.com (X. Liu).

http://dx.doi.org/10.1016/j.amc.2015.06.041
0096-3003/

distribution. Literature [15] presented a CS based on decision-maker and disturbance factors. Literature [16] improved CS for global optimization. Literature [17] enhanced CS by solving an unconstrained optimization problem. Literature [18] presented a new and complicated CS. Literature [19] mixed a DE algorithm with a CS algorithm. Literature [20] proposed a self-adapting, step-length CS. All of these studies served to enhance the search performance of the algorithm.

To improve the performance of the algorithm further, the current study improves CS and utilizes the initial population of chaos sequence to maintain the variety of the population. Then, the study introduces inertia weight into a Lévy flight random search to enhance global searching capability. Finally, this research uses the local search mechanism of the frog leaping algorithm to conduct a local search for local optimal solutions and to accelerate the convergence of the algorithm.

2. Cuckoo search algorithm

In nature, cuckoos randomly seek nests in which to lay their eggs. To simulate this process, the following three ideal hypotheses are developed:

(1) Each cuckoo lays an egg once and allows it to incubate in a randomly chosen nest.
(2) A group of bird nests is selected at random, and the best nests are maintained until the next generation.
(3) The number of available bird nests n is fixed. The possibility that the owner of a nest locates the nest of a foreign bird is denoted by $p_a \in [0, 1]$.

Given the premise of the hypotheses above, the nest-seeking behavior of cuckoos follows the Lévy flight model. The path and location update formula for this behavior is written as follows:

$$x_i^{t+1} = x_i^t + \alpha \oplus L(\lambda)\,, \quad i = 1, 2, 3 \ldots n \tag{1}$$

where x_i^t and x_i^{t+1} represent the locations of nest i at generations t and $t+1$, respectively; $\oplus$ represents point-to-point multiplication; and $L(\lambda)$ represents a random Lévy search path. The length and direction of this path are uncertain. To apply this formula to CS successfully, Literature [10] introduced the step length-controlled variable α, whose value is a constant over zero. This value varies in different cases, but $\alpha = 0.01$ in general.

$L(\lambda)$ is a Lévy distribution function that complicates integration. Therefore, Yang Xin She converted this function into a probability density function [21] in terms of power through simplification and Fourier transformation.

$$\text{Lévy} \sim u = t^{-\lambda}\ (1 < \lambda < 3), \tag{2}$$

where λ represents the power coefficient. To describe this part in simple and programmable mathematical language, Yang Xin She and Deb Suash adopted the formula of Lévy flight leaping path [22], which was proposed by Mantegna in 1992 to realize CS.

$$s = \frac{\mu}{|v|^{\frac{1}{\beta}}}. \tag{3}$$

Literature [21] proved that the Mantegna algorithm can calculate equivalence. In Formula (3), s represents the Lévy leaping flight path $L(\lambda)$. Furthermore, the relationship between β and λ in formula (2) can be expressed as $\lambda = \beta + 1, 0 < \beta < 2\,(\beta = 1.5$ [23] in CS). $\mu \cdot v$ represents the random number of normal distribution that follows the normal distribution in formula (4), and the standard deviation of the corresponding normal distribution in this formula is represented by σ_μ, σ_v. The values are shown in formula (5).

$$\mu \sim N\left(0, \sigma_\mu^2\right), \quad v \sim N\left(0, \sigma_v^2\right). \tag{4}$$

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_v = 1. \tag{5}$$

Suppose $S = \alpha \oplus L(\lambda) = \alpha \oplus s$, where S represents the path of cuckoos from the location of a previous nest x_i^t to a new location x_i^{t+1} as determined randomly in the solution space based on formula (1). s requires the derivation of two random numbers of normal distributions μ and v from formulas (4) and (5). μ and v are uncertain values; therefore, the length and direction of the path that the cuckoo searches at random according to Lévy flight are highly randomized. As a result, leaping from one region to another is easy. These features benefit the global search capability of the algorithm in the early phase of optimization, thereby enhancing the global optimizing capability of CS.

$r \in [0, 1]$ and p_a are compared through location updates. If $r > p_a$, then the located nest is changed at random; if not, the nest remains unchanged. Finally, the best nest location $(x_i^{t+1})$with a superior test value is maintained. At this point, the best $(x_i^{t+1})$is denoted by x_i^{t+1}.

Basic steps of CS:

Step 1: Initialization parameters are set. n best locations are randomly generated, and the corresponding adaptive values of nests are calculated. Then, the current optimal location of the bird and of the global optimal solution are determined and maintained until the next generation.