Knowledge-Based Systems 000 (2016) 1–12

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A novel multi-scale cooperative mutation Fruit Fly Optimization Algorithm

Yiwen Zhang[a,*], Guangming Cui[a], Jintao Wu[a], Wen-Tsao Pan[b], Qiang He[c]

[a] the Key Laboratory of Intelligent Computing & Signal Processing, Ministry of Education, Anhui University, Anhui 230031, China
[b] Department of Information Management, Oriental Institute of Technology, Taiwan, ROC
[c] School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia

ARTICLE INFO

Article history:
Received 10 June 2016
Revised 26 September 2016
Accepted 27 September 2016
Available online xxx

Keywords:
Fruit fly optimization algorithm
Multi-scale cooperative mutation
Global optimization
Convergence

ABSTRACT

The Fruit Fly Optimization Algorithm (FOA) is a widely used intelligent evolutionary algorithm with a simple structure that requires only simple parameters. However, its limited search space and the swarm diversity weaken its global search ability. To tackle this limitation, this paper proposes a novel Multi-Scale cooperative mutation Fruit Fly Optimization Algorithm (MSFOA). First, we analyze the convergence of FOA theoretically and demonstrate that its convergence depends on the initial location of the swarm. Second, a Multi-Scale Cooperative Mutation (MSCM) mechanism is introduced that tackles the limitation of local optimum. Finally, the effectiveness of MSFOA is evaluated based on 29 benchmark functions. The experimental results show that MSFOA significantly outperforms the improved versions of FOA presented in recent literature, including IFFO, CFOA, and CMFOA, on most benchmark functions.

1. Introduction

Optimization calculation finds an optimal solution that fulfils a set of constraints and achieves an optimal goal. When solving complex optimization problems, such as those with high dimensions, severe constraints, and multi-polarity, traditional optimization techniques, such as integer programming [1], dynamic programming [2], linear programming [3], and graph algorithms [4,5] suffer from computational complexity. In recent years, Swarm Intelligence (SI) based optimization algorithms have become more and more popular, bringing new vitality to optimization calculation.

Compared to traditional optimization algorithms, such as Genetic Algorithm (GA) [7], SI features simplicity and effectiveness in solving complex optimization problems [8]. Many SI-based optimization algorithms have been proposed, e.g., Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). PSO simulates the foraging behavior of bird swarms [8,20]. PSO depends strongly on parameter settings and is prone to converge into a local optimum. ACO models the foraging behavior of ants and is inspired by its information exchange mechanism [9]. However, ACO converges slowly due to stagnation. Such optimization algorithms find a near-optimal solution for a complex problem with a large number of variables and constraints [22]. In addition to SI based optimization algorithms that simulate animals behaviors, there are also optimization algorithms that simulate physical phenomena, e.g., Gravitational Search Algorithm (GSA) and Kinetic Gas Molecule Optimization (KGMO). GSA [10] simulates the characteristics of gravity. It is an intelligent algorithm that combines vector calculation with the law of universal gravitation. Kinetic Gas Molecule Optimization (KGMO) algorithm [11] simulates the dynamics of gas molecular. It models the variation of gaseous molecular kinetic energy with temperature and the irregular motion of gas molecules to seek the optimal value within the searching space. However, such algorithms are not very popular due to their high complexity.

The Fruit Fly Optimization Algorithm (FOA), proposed by Pan in 2011 [12], is a new SI based optimization algorithm inspired by the foraging behavior of fruit flies. Compared with other SI-based optimization algorithms, FOA is easy to understand and computationally inexpensive. As a novel optimization algorithm, FOA has attracted a lot of researchers attention and has been applied successfully in many areas in recent years. To name a few, J. Li et al. applied FOA to solving the hybrid flow-shop rescheduling problem in steelmaking systems and obtained convincing experimental results [13]. Y. Zhang et al. introduced FOA into service computing, and experimentally analyzed its performance [14,15]. H. Li et al. proposed a hybrid annual power load-forecasting model that incorporates FOA based on generalized regression neural network [16]. S. Lin et al. employed FOA to optimize an artificial neural network

* Corresponding author.
E-mail addresses: zhangyiwen@ahu.edu.cn (Y. Zhang), cgm133123@163.com (G. Cui), ahwjt@yahoo.com (J. Wu), teacherp0162@yahoo.com.tw (W.-T. Pan), qhe@swin.edu.au (Q. He).

http://dx.doi.org/10.1016/j.knosys.2016.09.027
0950-7051/

model for improving logistics quality, service satisfaction classification and prediction [17]. X. Zheng et al. designed a new encoding scheme and multi-group techniques that improve the effectiveness and efficiency of solving the semiconductor final testing scheduling problem [23].

However, similar to other SI based optimization algorithms, FOA also has its limitations. One of its major limitations is that it is prone to converge into a local optimum because it often fails to traverse the entire solution space [21]. To address this issue, many researchers have attempted to improve FOA in recent years. For example, L. Wu et al. proposed an improved FOA named CMFOA based on cloud model, and evaluated its performance on 33 benchmark functions [6]. However, CMFOA still converges slowly. M. Mitis et al. proposed a chaotic FOA (CFOA) based on chaotic operations and systematically analyzed its performance [18]. The major limitation of CFOA is that the analysis was performed on only 6 benchmark functions, and thus failed to fully evaluate the performance of CFOA. Q. Plan et al. proposed an improved fruit fly optimization algorithm (IFFO) by improving the method of individual generation [19]. The performance of IFFO was analyzed comprehensively on 29 benchmark functions. However, its convergence is still very slow. J. Niu et al. proposed an improved FOA based on differential evolution (DFOA). They modified the expression of the smell concentration judgment value and introduced a differential vector to replace the stochastic search [24]. The convergence of DFOA was analyzed based on 12 benchmark functions. However, the influence on different differential vectors was not considered. X. Yuan et al. proposed an improved FOA based on multi-swarm (MFOA) [25]. The application of MFOA to several benchmark functions and parameter identification of synchronous generator shows an improvement in its performance over the original FOA technique. However, the parameters and the sub-swarm number of MFOA usually impact its search performance.

In order to tackle the limitation of FOA, this paper presents an improved FOA named MSFOA based on a Multi-Scale Cooperative Mutation (MSCM) mechanism with the following main contributions:

(1) The limitations and convergence of the original FOA are analyzed theoretically. We demonstrate that the convergence of FOA depends on the initial swarm location;
(2) An improved fruit fly optimization algorithm, named MSFOA, is developed. MSFOA can escape from local optimum and achieve much better global optimum by employing the MSCM mechanism to mutate the position of the swarm;
(3) Through extensive experiments, we evaluate the convergence, stability, and global optimization ability of MSFOA in comparison with the most recent improved versions of FOA, including IFFO, CFOA and CMFOA, based on 29 benchmark functions.

The remainder of the paper is organized as follows. In Section 2, FOA and its limitation are analyzed. The convergence and the convergence condition of FOA are analyzed in Section 3. Section 4 discusses the MSCM mechanism and MSFOA in detail. MSFOA are evaluated experimentally in Section 5. Finally, Section 6 concludes this paper.

2. Fruit Fly Optimization Algorithm

This section first introduces the original FOA. It then analyzes its limitations theoretically.

2.1. FOA basics

FOA was inspired by the foraging behaviors of fruit flies in the nature [12]. A fruit fly determines the location of food with its unique osphresis, its vision, and the smell concentration. The optimization process of FOA consists of the following 8 steps.

Step 1: Initialization the location of fruit fly swarm.

$$\begin{cases} x\_axis = rand(LR) \\ y\_axis = rand(LR) \end{cases} \tag{1}$$

where LR represents the location parameter of the initial swarm.

Step 2: Generation of the location of individual fruit flies in the swarm.

$$\begin{cases} x_i = x\_axis + rand(V) \\ y_i = y\_axis + rand(V) \end{cases} \tag{2}$$

where V represents the range parameter generated by the swarm, x and y represent the location coordinates.

Step 3: Calculation of the distance between individual fruit fly and origin.

$$Dist_i = \sqrt{x_i^2 + y_i^2} \tag{3}$$

Step 4: Calculation of the smell concentration judgment value of each individual fruit fly.

$$S_i = \frac{1}{Dist_i} \tag{4}$$

Step 5: Calculation of the smell concentration value of each individual fruit fly.

$$Smell_i = Smell\_function(S_i) \tag{5}$$

Step 6: Identification of the optimal smell concentration value in the swarm, denoted by the maximum value.

$$[bestSmell \quad bestindex] = max(Smell_i) \tag{6}$$

Step 7: Reservation of the identified optimal smell concentration value and replacement of the swarm location.

$$\begin{cases} Smellbest = bestSmell \\ x\_axis = x_{bestindex} \\ y\_axis = y_{bestindex} \end{cases} \tag{7}$$

Step 8: Termination of the algorithm if the maximum number of generation is reached; otherwise, go back to **Step 2**.

2.2. Analysis of FOA

Compared with other SI based optimization algorithms, FOA has the advantages of simple structure and low computational complexity. However, FOA cannot solve complex optimization problems efficiently, due to its following limitations.

Limitation 1. FOA is prone to converge into a local optimum because it cannot traverse the entire solution space.

Proof. The calculation of distance ($Dist_i$) in FOA is $Dist_i = \sqrt{x_i^2 + y_i^2}$ and the smell concentration judgment value is $S_i = \frac{1}{Dist_i}$. Therefore, $Dist_i$ and S_i are always greater than 0. Thus, the search space cannot reach the negative threshold. □

Limitation 2. Fruit fly individual values are relatively monotonous and tend towards zero.

Proof. Step 2 to 4 presented in Section 2.1 describe the calculation of the distance and the smell concentration judgment value of individual fruit flies. The calculation process is as follow.

According to Eqs. (3) and (4), when the problem has a large search space and the individual value of x or y are large, S_i always tends toward zero, driving the algorithm to converge into a local optimum. While the algorithm has an initial advantage at the extreme points $X^* = 0$, it can result in weak swarm variation and limited global optimization capacity.

To conclude, FOA is prone to converge into a local optimum inherently, except for the $X^* = 0$ benchmark functions. □