Knowledge-Based Systems 97 (2016) 158–174

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

An effective and efficient fruit fly optimization algorithm with level probability policy and its applications

Lin Wang[a], Rui Liu[a], Shan Liu[b,*]

[a] School of Management, Huazhong University of Science and Technology, Wuhan 430074, China
[b] School of Management, Xi'an Jiaotong University, Xi'an 710049, China

ARTICLE INFO

Article history:
Received 5 May 2015
Revised 3 January 2016
Accepted 4 January 2016
Available online 12 January 2016

Keywords:
Fruit fly optimization algorithm
Level probability
Continuous function problem
Joint replenishment problems

ABSTRACT

An improved fruit fly optimization algorithm (FOA) is proposed for optimizing continuous function problems and handling joint replenishment problems (JRPs). In the proposed FOA, a level probability policy and a new mutation parameter are developed to balance the population diversity and stability. Twenty-nine complex continuous benchmark functions are used to verify the performance of the FOA with level probability policy (LP–FOA). Numerical results show that the proposed LP–FOA outperforms two state-of-the-art variants of FOA, the differential evolution algorithm and particle swarm optimization algorithm, in terms of the median and standard deviations. The LP–FOA with a new and delicate coding style is also used to handle the classic JRP, which is a complex combinatorial optimization problem. Experimental results reveal that LP–FOA is better than the current best intelligent algorithm, particularly for large-scale JRPs. Thus, the proposed LP–FOA is a potential tool for various complex optimization problems.

1. Introduction

The fruit fly optimization algorithm (FOA) is a novel meta-heuristic method proposed by Pan [17] to solve the global optimization problems. The FOA simulates the intelligent collaboration of fruit flies when they swarm in searching for food. A fruit fly can find food easily with its sensitive osphresis and vision abilities. Considering its simple structure, easy implementation, and good performance, the FOA and its variants are applied to handle many practical optimization fields, such as in financial distress situations [17], power load forecasting [10], neural network parameter optimization [2,11], web auction logistics service [13], proportional–integral–derivative (PID) controller parameter tuning [24], and many other applications [4,8,12,34,35,38].

However, the basic FOA often derives a local extreme when solving high-dimensional functions and large-scale combinational optimization problems. Thus, many researchers focused on improving the search ability of the basic FOA. Pan [18] designed a modified FOA by adding a new parameter, wherein the fruit flies search in a three-dimensional space. In contrast to other FOAs with a non-linear generation mechanism of candidate solutions, a linear generation mechanism of candidate solutions was proposed by Shan et al. [23]. Pan et al. [16] improved the basic FOA by adding a new parameter. With its dynamic search radius, the improved FOA (IFOA1) exhibits an excellent performance. Yuan et al. [36] developed a multi-swarm FOA (MFOA), wherein multiple sub-swarms move independently in the search space at the same time and local behavior between sub-swarms is considered. Wang et al. [32] proposed the swarm collaboration and added random perturbation (RP) to improve the performance of FOA (IFOA2). Although these improvements enhanced the performance of FOA, deriving the global optimal solution is still difficult when facing high-dimensional functions and large-scale combinational optimization problems.

This study aims to propose a new and improved FOA to solve the aforementioned complex optimization problems effectively and efficiently. Several advanced techniques are adopted to enhance the effectiveness of the FOA, including a level probability (LP) solution generation method and a novel parameter, to balance the population diversity and stability. The proposed new and improved FOA is called LP–FOA. The LP–FOA is applied to 29 standard functions optimization problems and joint replenishment problems (JRPs) to verify its performance. In fact, JRPs have been extensively investigated because of its significance in the theoretical and practical fields. However, JRPs are typically nondeterministic polynomial hard (NP-hard) problems.

IFOA1 has been proven to be better than the basic FOA, MFOA, and five other existing harmony search algorithms [16] for continuous function optimization problems. In this study, the results of

* Corresponding author. Tel.: +860115807106158.
E-mail addresses: wanglin982@gmail.com (L. Wang), rliuhust316@gmail.com (R. Liu), shan.l.china@gmail.com, 279266384@qq.com (S. Liu).

http://dx.doi.org/10.1016/j.knosys.2016.01.006
0950-7051/

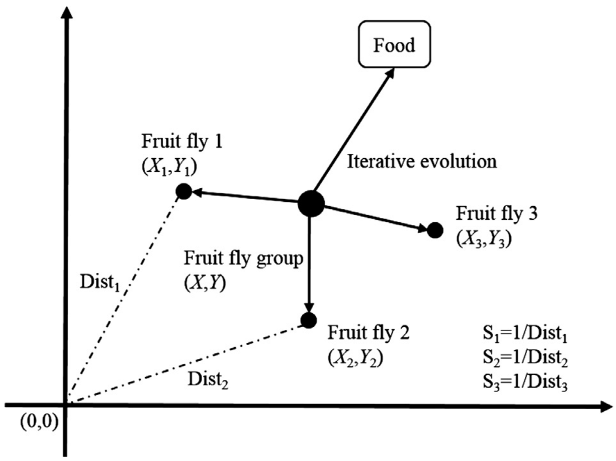

Fig. 1. Food-searching iterative process of the fruit fly swarm (source: Wang et al. [32]).

the benchmark functions test indicate that the proposed LP–FOA outperforms IFOA1, IFOA2, differential evolution (DE), and particle swarm optimization (PSO). Comparative and randomly generated examples also show that LP–FOA is better than the current best algorithm for JRPs. Thus, the proposed LP–FOA can be a strong candidate for solving complex optimization problems. The main contributions of this study are as follows: (a) An improved FOA with a level probability policy and a new mutation parameter is proposed. (b) A delicate LP–FOA-based coding style is designed to solve JRPs, resulting in major improvements in solving large-scale problems.

The remainder of the paper is organized as follows: Section 2 summarizes the related FOA techniques and introduces the basic FOA. Section 3 describes the proposed LP–FOA. In Section 4, two latest variants of FOA are adopted to compare with the LP–FOA by the 29 benchmark function test. In Section 5, the LP–FOA is used to deal with JRPs. Section 6 discusses the results and provides future research recommendations.

2. The basic FOA

The fruit fly is a widely existing insect that has well-developed olfactory and visual organs. The fruit fly can easily find a food source through its scent, even if the source is 40 kilometer away. When a fruit fly swarm hunts for food, each individual can measure the smell concentration in its current location. Then, the smell concentrations of all individuals are compared by fitness. The best fitness that corresponds to the current location is the best current location. Afterward, the fruit fly swarm will move toward the current location by aid of their sensitive vision. Fig. 1 shows the food-searching iterative process of the fruit fly swarm.

The steps of the basic FOA [17] can be described as follows:

Step 1: Initialization. The fruit fly swarm location range (*LR*), fly range (*FR*), maximal central processing unit (CPU) time (*Maxtime*), and population size (*popsize*) are initialized. The initial *LR* is generated as follows:

$$X\_I = rand\ (LR), \quad Y\_I = rand\ (LR). \tag{1}$$

Step 2: Osphresis searching phase.

Step 2.1: When each individual hunts for food, its new location can be generated by using Eq. (2):

$$X_i = X\_I + rand\ (FR), Y_i = Y\_I + rand\ (FR). \tag{2}$$

Step 2.2: Compute the distance of the food source to the origin by using Eq. (3):

$$Dist_i = \sqrt{X_i^2 + Y_i^2}. \tag{3}$$

Step 2.3: Compute the smell concentration judgment value (S_i) and the judgment function ($Smell_i$) of the individual location by using Eqs. (4) and (5), respectively:

$$S_i = 1/Dist_i, \tag{4}$$

$$Smell_i = fitness(S_i). \tag{5}$$

Step 2.4: Select the fruit fly with maximal judgment function and its corresponding location by using the following equation:

$$[bestSmell, bestIndex] = \max\ (Smell). \tag{6}$$

Step 3: Vision searching phase.

Maintain the maximal judgment function value and corresponding *X*, *Y* coordinates. The fruit fly swarm moves toward the location by using their vision.

$$Smellbest = bestSmell, X\_I = X(bestIndex), Y\_I = Y(bestIndex). \tag{7}$$

Step 4: The implementation of **Steps 2** and **3** are repeated. The circulation stops when the CPU time reaches the *Maxtime*.

3. Procedure of the LP–FOA

The FOAs often derive a local extreme for solving high-dimensional functions and large-scale combinational optimization problems. However, an excellent FOA can easily skip a local extreme. Thus, the LP method is adapted to expand the search space and skip local optima. The LP method divides the randomly generated location of the fruit fly into different levels according to probability.

To expand search space and maintain stability simultaneously for the proposed LP method, we set that the fruit fly of level with high probability cannot fly far away, while the fruit fly of level with high probability can fly farther. Fig. 2 shows the fly range of the fruit fly in the LP method. The probability in level 1 is two times that in level 2. However, the fly range of the fruit fly in level 2 is 10 times that of the fruit fly in level 1. The probability in level *c* ($c=1,\ldots, C$, where *C* is the maximal level) is two times that in level $c+1$, similar to levels 1 and 2. We assume that the probability in level *c* is two to four times that in level $c+1$ because the search space will be clearly reduced when the probability of two adjacent levels has an evident difference. In this study, we assume that the probability in level *c* is 10 times that in level $c+1$.

The details are described as follows:

Step 1: Initialization. The swarm location range (*X_I, Y_I*), *Maxtime, sizepop*, mutation factor (*F*), and maximal level (*C*) are initialized, as follows:

$$X\_I = rand(LR),$$
$$Y\_I = rand(LR),$$
$$Smellbest = fitness(S(X\_I, Y\_I)).$$

Step 2: Osphresis searching phase.

Step 2.1: Update the location of the swarm based on the LP method and an adaptive mutation factor. The detailed procedure is provided in Sub-algorithm 1.

Steps 2.2 to 2.4: These steps are similar to **Steps 2.2** to **2.4** in Section 2.

Step 2.5: Swarm collaboration. According to [32], the best 50% individuals are selected to fly toward the best location, whereas the other individuals are randomly generated by initialization. In fact, a slight difference in the performance of the different percentages of individuals that fly to the best location is observed [32]. The detailed procedure is provided in Sub-algorithm 2.

Step 2.6: **Steps 2.1** to **2.5** are repeated. The judgment function values ($Smell_i$) of the new swarm are calculated. The fruit fly with