Expert Systems With Applications 95 (2018) 384–420

Contents lists available at ScienceDirect

Expert Systems With Applications

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

New cuckoo search algorithms with enhanced exploration and exploitation properties

Rohit Salgotra[a,*], Urvinder Singh[a], Sriparna Saha[b]

[a] *Electronics & Communication Engineering Department, Thapar University, Punjab, India*
[b] *Department of Computer Science, Indian Institute of Technology Patna, India*

ARTICLE INFO

Article history:
Received 11 May 2017
Revised 17 November 2017
Accepted 21 November 2017
Available online 22 November 2017

Keywords:
Cuckoo search
Cuckoo Version 1.0
Cuckoo Version 2.0
Cuckoo Version 3.0
Numerical optimization
Benchmark problems

ABSTRACT

Cuckoo Search (CS) algorithm is nature inspired global optimization algorithm based on the brood parasitic behavior of cuckoos. It has proved to be an efficient algorithm as it has been successfully applied to solve a large number of problems of different areas. CS employs Lévy flights to generate step size and to search the solution space effectively. The local search is carried out using switch probability in which certain percentages of solutions are removed. Though CS is an effective algorithm, still its performance can be improved by incorporating the exploration and exploitation during the search process. In this work, three modified versions of CS are proposed to improve the properties of exploration and exploitation. All these versions employ Cauchy operator to generate the step size instead of Lévy flights to efficiently explore the search space. Moreover, two new concepts, division of population and division of generations, are also introduced in CS so as to balance the exploration and exploitation. The proposed versions of CS are tested on 24 standard benchmark problems with different dimension sizes and varying population sizes and the effect of probability switch has been studied. Apart from this, the best of the proposed versions is also tested on CEC 2015 benchmark suite. The modified algorithms have been statistically tested in comparison to the state-of-the-art algorithms, namely grey wolf optimization (GWO), differential evolution (DE), firefly algorithm (FA), flower pollination algorithm (FPA) and bat algorithm (BA). The numerical and statistical results prove the superiority of the proposed versions with respect to other popular algorithms available in the literature.

1. Introduction

Cuckoo search (CS) algorithm (Yang & Deb, 2009) is a recently introduced meta-heuristic algorithm and is based on the obligate brood parasitic behavior of cuckoo species found in nature. The algorithm starts by dividing the search process into two phases which are a global and a local phase. In the global phase, the formation of new nests takes place while in the local phase, removal of a fraction of worst nests is followed. Here global phase refers to the exploration where as local phase corresponds to the exploitation. The global phase is governed by Lévy flight based random walks rather than simple Brownian or Gaussian walks (Pavlyukevich, 2007). The main reason for the use of Lévy flight is because of their heavy tail, infinite mean and variance, which helps in exploring the search space in a potentially more efficient way. The local phase is governed by selecting two random solutions from the search space with a certain probability, which controls the extent of exploitation. So overall there are three parameters which control the working capability of CS algorithm (Yang & Deb, 2009). The first parameter is the Lévy flight component which controls the exploration search equation, second is the exploitation or local search equation controlled by two random solutions and third is the probability which decides the extent of exploration and exploitation.

The CS algorithm, because of its simple structure, has gained attraction from the research community in the recent years and a large number of articles have been proposed to improve its working capability such as Zhang, Wang, and Wu (2012), Kanagaraj, Ponnambalam, and Jawahar (2013), Ilunga-Mbuyamba et al. (2016), Long, Liang, Huang, and Chen (2014), Wang and Zhong (2015) and others. A detailed discussion about the recent advances on CS is given in Section 3. It has been found that CS algorithm lacks proper balance between the exploration and exploitation phase and more work is required to be done to achieve the balance (Zhang et al., 2012). Also, probability, which decides the extent of exploration and exploitation, is a very important factor and no proper study has been done to find an appropriate value

* Corresponding author.
E-mail addresses: rohit.salgotra@thapar.edu (R. Salgotra), sriparna@iitp.ac.in (S. Saha).

https://doi.org/10.1016/j.eswa.2017.11.044

of this parameter. Further, a very little work has been done to test what set of parameters gives the best performance of CS algorithm. Most of the works done till date on CS has focused only on its application in different domains and very limited work has been done to improve its performance. The algorithm though is very efficient in exploring the search space; much is required to improve its exploitative tendencies. Several other problems are also associated with CS. Those are discussed in detail in Section 3. Motivated by these, in the current work at first we developed some new improved versions of CS which have good exploration and exploitation properties. A thorough sensitivity study of different parameters namely probability, population size, and dimension of CS is also conducted and parameter combinations which give the best results for a particular set of problems are reported.

To achieve the above-said objectives, three new variants have been proposed in Section 3 in this a new concept of division of population and generations. By division of generations, the authors aim to divide the population into two halves. This is done because the exploitation is weak in CS, so it requires a better search equation which can provide better results. There are two advantages of dividing the population into two halves. For the first half of population, the algorithm is able to perform extensive exploration using the original equations and for the second half of the population, intensive exploitation can be performed using the new search equation. Also division of population has been done to help the members of the population to change their positions abruptly and converge faster towards the end of iterations. And both division of population and generations is done to achieve a balance between exploration and exploitation phase. More discussion about how the division of population and generation is applied is given in the consequent subsections. For improving the exploration, instead of using Lévy flight based random walks, Cauchy based walks are employed and to improve the exploitation of the local search, different searching strategies are used to evaluate the final solution. More justification for the use of Cauchy based random walks has been presented in subsequent subsections. Further to analyze the performance of CS and the proposed variants experimentally, a set of twenty-four benchmark functions (Suganthan et al., 2005) has been used and the effects of different parameters like switching probability, population size and dimension have been discussed. The effect of switching probability has been analyzed by choosing five different probability values and the best fit probability is estimated. To analyze the effect of population size, three different population sizes are used. In order to show the effect of population size and probability, only CS and proposed variants are compared and best among the proposed variants are analyzed. The best among the proposed variants is then compared with the well-known state of art algorithms. Also, five different dimension sizes varying from 30 to 1000 are used for analyzing the applicability of proposed variant on higher dimensional problems. Statistical tests are also done to validate the end results. Further to validate the best-proposed algorithm in solving some highly challenging datasets, it is applied to CEC 2015 benchmark problems (Liang, Qu, Suganthan, & Chen, 2014) and comparative study with respect to other well-known algorithms is presented.

The article is divided into six sections and is outlined as follows: Section 1 details about the introduction part, dealing with the motivation, objectives, and contribution of this research. Section 2 gives details of the basic CS algorithm while in Section 3, the literature review or the related work is presented. Section 4 outlines the proposed approaches along with their mathematical models and theoretical analysis. This section also presents the complexity analysis of proposed versions. In Section 5, extensive results are presented. Here detailed results with respect to the parameters of CS and the proposed variants along with a comparative analysis of state-of-the-art algorithms are presented. In the subsequent subsections of Section 5, main findings of research, limitations and insightful implications from the proposed work are also presented. The final Section 6 presents the concluding remarks and future scope. The outline of the article is given in Fig. 1.

2. Cuckoo search algorithm

Cuckoo Search (CS) is a new heuristic algorithm inspired from the obligate brood parasitic behavior of some cuckoo species as they lay their eggs in the nests of host birds. Some cuckoos have a specialty of imitating colors and patterns of eggs of a few chosen host species. This reduces the probability of eggs being abandoned. If host bird discovers foreign eggs, they either abandon the eggs or throw them away. Parasitic cuckoos choose a nest where the host bird just lays its eggs. Eggs of cuckoo hatch earlier than their host eggs and when it hatches, it propels the host eggs out of the nests. Hence cuckoo chicks get a good share of food and sometimes they even imitate the voice of host chicks to get more food (Payne, 2005). Mostly cuckoos search food by a simple random walk, where the random walk is a Markov chain whose next position is based on current position and transition probability of next position. Using Lévy flights instead of simple random walks improve the search capabilities. Lévy flight is a random walk in step-lengths following a heavy-tailed probability distribution (Yang & Deb, 2009). Each cuckoo acts as a potential solution to the problem under consideration. The main aim is to generate a new and potentially better solution (cuckoo) to be replaced with a not so good solution. Each nest has one egg but as the problem complexity increases, multiple eggs can be used to represent a set of solutions. There are three basic idealized rules of CS. First rule says that each cuckoo lays one egg and dumps it in a random nest. The second rule is that the nest with the highest fitness will carry over to next generations whereas the final rule defines that the number of available host nests is kept fixed and the egg laid by cuckoo is discovered by host bird with a probability $p \epsilon$ [0, 1]. And depending on p, the host bird either throws the egg away or abandons the nest. It is assumed, that only a fraction p of nests is replaced by new nests.

Based on the three rules, the cuckoo search has been implemented. To generate a new solution $x_i^{t+1}$for ith cuckoo, Lévy flight is performed. This step is called global random walk and is given by

$$x_i^{t+1} = x_i^t + \alpha \otimes Le'vy(\lambda)\left(x_{best} - x_i^t\right) \tag{1}$$

The local random walk is given by:

$$x_i^{t+1} = x_i^t + \alpha \otimes H(p-\epsilon) \otimes \left(x_j^t - x_k^t\right) \tag{2}$$

where x_i^t is the previous solution, $\alpha > 0$ is the step size related to problem scales and $\otimes$ is entry wise multiplication. Here x_j^t *and* x_k^t are randomly selected solutions and x_{best} is the current best solution. In present work, the random step length via Lévy flight is considered due to more efficiency of Lévy flights in exploring the search space and is drawn from a Lévy distribution having infinite variance and mean.

$$Le'vy \sim \left\{ \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda,}} (s \gg s_0 > 0) \right. \tag{3}$$

As some fractions of new solutions are generated by Lévy flights, so the local search speeds up. Here some of the solutions should be generated by far field randomization which will keep the system away from getting trapped in local optimum, $\Gamma(\lambda)$ is the gamma function, p is the switch probability, ε is a random number and $(1 < \lambda \leq 3)$. The step length in cuckoo search is heavy-tailed and any large step is possible due to large scale randomization. The pseudo-code for CS is given in Algorithm 1.