Applied Soft Computing 52 (2017) 771–794

Contents lists available at ScienceDirect

Applied Soft Computing

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

Snap-drift cuckoo search: A novel cuckoo search optimization algorithm

Hojjat Rakhshani*, Amin Rahati

Department of Computer Science, Faculty of Mathematics, University of Sistan and Baluchestan, Zahedan 98135-674, Iran

ARTICLE INFO

Article history:
Received 21 August 2015
Received in revised form 2 June 2016
Accepted 29 September 2016
Available online 11 October 2016

Keywords:
Global numerical optimization
Cuckoo search
Lévy flights
Nonparametric tests
Parameter sensitivity

ABSTRACT

Cuckoo search (CS) is one of the well-known evolutionary techniques in global optimization. Despite its efficiency and wide use, CS suffers from premature convergence and poor balance between exploration and exploitation. To address these issues, a new CS extension namely snap-drift cuckoo search (SDCS) is proposed in this study. The proposed algorithm first employs a learning strategy and then considers improved search operators. The learning strategy provides an online trade-off between local and global search via two snap and drift modes. In snap mode, SDCS tends to increase global search to prevent algorithm of being trapped in a local minima; and in drift mode, it reinforces the local search to enhance the convergence rate. Thereafter, SDCS improves search capability by employing new crossover and mutation search operators. The accuracy and performance of the proposed approach are evaluated by well-known benchmark functions. Statistical comparisons of experimental results show that SDCS is superior to CS, modified CS (MCS), and state-of-the-art optimization algorithms in terms of convergence speed and robustness.

1. Introduction

In recent years, metaheuristic algorithms have been extensively used to solve complex and highly non-linear optimization problems. Most of these methods are inspired from natural or physical processes. Genetic algorithm (GA) [1], particle swarm optimization (PSO) [2] and artificial bee colony (ABC) [3] are some examples of nature-inspired algorithms. The GA takes inspiration from the phenomenon of natural evolution [1]; the PSO algorithm attempts to mimic social behavior of bird flocking or fish schooling [2]; and the ABC models food searching behavior of honey bees [3]. On the other hand, harmony search (HS) [4], gravitational search algorithm (GSA) [5] and simulated annealing (SA) [6] belong to physics-inspired algorithms. The HS emulates the improvisation of music players to find the solution of optimization problems [4]; the GSA models the law of gravity and the notion of mass interactions [5]; and the SA resembles the process of annealing in melted materials [6]. Compared to gradient-based methods, metaheuristic algorithms do not require differentiable fitness functions and calculation of derivatives [7]. Also, in contrast with problem-specific heuristics algorithms, they have no assumption about the problem that is being optimized. For these reasons, developing metaheuristic algorithms for optimization has grown in recent years. They have shown good performance in a wide range of applications such as function optimization [7–11], data clustering [12,13], design optimization [14,15], economic dispatch [16], and neural networks training [17]. The common problem of metaheuristic algorithms, however, is their slow rate of convergence. This makes metaheuristic algorithms unsuitable for solving real-world optimization problems, especially when objective function evaluation is computationally expensive and time consuming such as protein structure predication problem [18]. In addition, tuning and adjusting the control parameters of these algorithms is itself an optimization problem [19]. Also, real-world problems are usually NP-hard and there is no a well-known general algorithm that can be used for any given problem [20]. Hence, searching for more efficient metaheuristic algorithms remains an open issue.

One of the latest nature-inspired metaheuristic algorithms is cuckoo search (CS) that was developed by Yang and Deb [21]. CS combines the idea of obligate brood parasitism of some cuckoo species with Lévy flights. Long jumps provided by Lévy flights can explore the search space better than algorithms by uniform and Gaussian distributions [19]. A combination of these Lévy flights advantages with local and global search abilities makes CS as one of the most efficient optimization algorithms. A previous study by Yang and Deb [21] reveals the fact that CS has a better performance

* Corresponding author.
E-mail addresses: rakhshani@pgs.usb.ac.ir (H. Rakhshani), a.rahati@cs.usb.ac.ir (A. Rahati).

http://dx.doi.org/10.1016/j.asoc.2016.09.048

in comparison with PSO and GA. Moreover, a conceptual comparison suggests that CS and DE algorithms provide more robust results than PSO and ABC [22]. CS has been applied in various fields of optimization and computational intelligence with promising results. Yang and Deb [23] proposed a multiobjective CS for engineering design problems. Gandomi et al. [24] modified CS to solve truss optimization problems. Layeb [25] proposed a new variant of CS in combination with quantum-based approach to solve Knapsack problems. In [26], a multi-population CS has been introduced for brain tumour images segmentation. Araghi et al. developed an advanced CS algorithm for optimally tuning parameters of traffic signal controllers [27]. In another research, using the modified CS (MCS) a new computational aerodynamic shape optimization algorithm has been developed [28]. M. Sait et al. adopted CS in order to solve the virtual machine placement problem of datacenters [29]. Mishra et al. presented a solution to the optimal power flow using the MCS optimization algorithm [30]. Interestingly, Sun et al. applied CS to predict the yearly foreign tourist arrivals to China [31]. Suresh and Lal introduced a new variant of CS for segmenting satellite images [32]. Wang et al. proposed an intelligent optimized hybrid model based on CS algorithm to forecast solar radiation [33]. In Ref. [34], authors hybridized CS and teaching-learning-based algorithms to address optimization problems in structure designing and machining processes. Moreover, Cobos et al. developed a method based on CS and balanced Bayesian information criterion for clustering web search results [35]. In Ref. [36], authors adopted CS algorithm for solving both convex and nonconvex economic dispatch problems. In another study [37], Bhateja et al. investigated the applicability of CS in cryptanalysis of Vigenere cipher. In Ref. [38], authors applied a CS algorithm for determining the location and depth of cracks in a cantilever beam.

Although CS and its various extensions have been proven successful, a large number of future researches are necessary to achieve more efficient algorithm. Toward this goal, nearly all existing metaheuristic variants share a common characteristic: they attempt to make an appropriate trade-off between the exploration and exploitation. The former mechanism is responsible for generating solutions in new regions of search space, whilst the latter searches at the vicinity of current promising solutions [39,40]. During the optimization process, exploration walk around more extensive search space and consequently increases the required computational time. On the other side, highly favored exploitation may imply a rapid loss of diversity and lead to the so-called premature convergence problem. For this reason, providing a good trade-off between this two conflicting strategies has long been an important issue in evolutionary computation and optimization [39,40]. To this end, a variety of techniques have been applied. Most of these techniques deal with: (1) parameter settings; (2) neighborhood topology; (3) learning strategies; and (4) hybridized methods [38]. Approaches in the first category mainly use memory based adaptation [41,42], linearly increasing/decreasing functions [43], Gaussian adaption [44], non-linear increasing/decreasing functions [45], and fuzzy based methods [46] to control the key parameter settings in metaheuristics. The second category, employs effective neighborhood structures such as fully informed [47], heterogeneous [48], self-adaptive [49], and ring topology [50] to prevent the algorithm from premature convergence. The learning techniques which are classified into the third category, present learning strategies to perform an effective search toward the global optimum. Social learning [51], comprehensive learning [45], orthogonal learning [52], teaching and peer learning [53], opposition based learning [54], and elitist learning [55] are some examples of such considerations. Moreover, hybrid variants incorporate search strategies from different algorithms in order to combine advantages of each individual strategy [56]. Particularly, metaheuristics are successfully hybridized with constraint programming, tree search techniques, problem relaxation, fuzzy methods, and other metaheuristic algorithms [56–60]. Given the aforementioned approaches, however, a key question that comes to mind is which technique is more suitable for making a good balance between the exploration and exploitation.

An appropriate answer to the above question might be found by referring to the *optimal contraction theorem* for exploration–exploitation trade-off [40]. This theorem states that an optimal optimizer is expected to take into the account the best useful information about the problem at hand. To verify this finding, authors conducted a set of experiments using PSO, GA, SA, multi-start stochastic hill climbing, branch-bound, and random sampling algorithms. Although each of the selected optimizers had different exploration-exploitation characteristics, but the obtained results showed that it is promising to adopt a dynamic optimization feature factor which is computed periodically based on the accumulated information of the fitness function to regulate a trade-off [40]. Motivated by this aspect, we introduce a new CS variant, namely snap-drift CS (SDCS), which periodically acquires information about the fitness function of the problem via a learning technique in order to optimize its exploration-exploitation behavior. Indeed, this algorithm uses a form of reinforcement learning which toggles between two snap and drift modes. In snap mode, SDCS reinforces global search ability to mitigate premature convergence problem. Alternatively, in drift mode, the probability of local search increases to enhance the rate of convergence. In fact, SDCS uses a kind of probabilistic adaptation; whereby, the probability of algorithm being adapted reduces as performance increases. In line with this purpose, SDCS also presents new search operators to strengthen global and local search capabilities in snap and drift modes, respectively. To sum up, the contributions of this research are presented as follows:

- A new CS extension, namely SDCS, has been developed which incorporates useful information about the optimization problem in order to improve exploration- exploitation trade-off in CS.
- The proposed SDCS employs a reinforcement learning technique inspired by snap-drift neural network in combination with new search operators to provide an improvement in search process.
- A series of experiments are conducted to investigate the impact of the proposed schema on the performance of the CS algorithm.
- The convergence speed, robustness and computational complexity of the proposed SDCS are evaluated against the traditional CS, state-of-the-art metaheuristics, and several variants of the basic algorithms.

The rest of paper organizes as follows. Section 2 briefly reviews related works in the literature. Section 3 introduces standard CS algorithm. Section 4 elaborates our proposed approach and its technical details. In Section 5, performance of the proposed SDCS is validated by performing some numerical experiments on well-known benchmark functions and comparing the results with standard CS, MCS, and state-of-the-art algorithms. Results and discussion are presented in Section 6, followed by conclusion in the last section.

2. Related works

Up to now, different variants of the standard CS for global optimization have been proposed in the literature. The early work on the topic was initiated by Walton et al., who incorporated an information exchange mechanism and a self-adaptive step size parameter into the CS [61]. Experimental studies showed that the proposed MCS algorithm can lead to an improvement in the convergence rate of basic algorithm. In another study, a new CS extension