Applied Thermal Engineering 76 (2015) 535–537

Contents lists available at ScienceDirect

Applied Thermal Engineering

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

Letter to Editor

A short discussion about "Economic optimization design of shell-and-tube heat exchangers by a cuckoo-search-algorithm"

Keywords:
Cuckoo search
Cuckoo optimization algorithm
Swarm intelligence

A B S T R A C T

This short discussion concerns some inaccuracy in the recently published paper "M. Asadi et al. Economic optimization design of shell-and-tube heat exchangers by a cuckoo-search-algorithm. Applied thermal engineering, 2014". There was a mix-up between the cuckoo search (CS) with the cuckoo optimization algorithm (COA), which may cause further confusion if uncorrected. In this discussion, we intend to clarify the differences between both algorithms so as to provide a more accurate description of both algorithms.

1. Introduction

During the last two decades, nature-inspired algorithms, especially those based on swarm intelligence, have become one of the most powerful tools for optimization. These algorithms can usually cope with both discrete and continuous optimization problems. Many studies in the literature also showed their potential in various real-world applications.

Recently, an interesting paper by Asadi et al. [1] was published in Applied Thermal Engineering where the authors stated that they used a cuckoo search (CS) algorithm. However, if we look at the paper in more detail, it is easily seen that the authors did not use the original CS developed by Yang and Deb in 2009 [2], but it was confused with another algorithm cuckoo-inspired algorithm, called cuckoo optimization algorithm (COA) developed by Rajabioun in 2011 [3]. In fact, there is no direct link between CS and COA, except that both algorithms were inspired by cuckoo behavior. Therefore, this paper intends to briefly discuss the main differences between both cuckoo-inspired algorithms, and also tries to show the primary causes leading to similar mistakes that are increasingly appearing in the published literature.

The structure of this discussion is as follows: Section 2 briefly discusses the fundamental ideas of the CS algorithm, while Section 3 illustrates the basics of the COA algorithm. In Section 4, the increasingly growing causes of these kinds of mistakes in published papers are briefly discussed. The paper finishes with some conclusions in Section 5.

2. Cuckoo search

The Cuckoo Search (CS) algorithm, or simply cuckoo search, was developed by Xin-She Yang and Suash Deb in 2009 [2,4], and CS is the first algorithm that is inspired by the characteristics and behaviour of the so-called brooding parasitism of cuckoo species. This CS is a swarm-intelligence-based algorithm [5]. According to Yang and Deb, the following three idealized rules were used in the CS [2]:

1. Each cuckoo lays one egg at a time, and dump its egg in randomly chosen nest.
2. The best nests with the high-quality eggs will carry over to the next generation.
3. The number of available host nests is fixed, and the egg laid by a cuckoo can be discovered by the host bird with a probability $p_a \in [0,1]$. In this case, the host bird can either throw the egg away or abandon the nest, and then build a completely new nest at a new location.

The original CS algorithm is presented in Algorithm 1. As can be seen from the pseudocode, the first step of the CS algorithm is to generate an initial population, where the host nests are positioned within the search space randomly (line 1). In the main loop that follows, the algorithm randomly obtains a new position of i-th cuckoo using Lévi flights (line 5), and then evaluates its fitness (line 6). Then, a certain random solution j is selected that can be replaced when the i-th solution is better (lines 9–11). The worst nest can be abandoned and a new one built in place of it (lines 12–14). However, tracking the best solution is performed in lines 15–17.

The original CS algorithm consists of the following components:

- The population of nests is represented as $\mathbf{x}_i = (x_{i1}, \ldots, x_{iD})^T$ for $i = 1 \ldots \mathrm{Np}$, where Np denotes the population size and D is the dimension of the problem.
- The initialization of host nests is performed randomly obeying the equation $x_{ij}^{(0)} = U(0,1) \cdot (\mathrm{Ub}_j - \mathrm{Lb}_j) + \mathrm{Lb}_j$.
- The new solution (of a cuckoo) is obtained according to equation $x_i^{(t+1)} = x_i^{(t)} + \alpha \otimes L(s,\lambda)$, where α is the step size scaling factor, and $L(s,\lambda)$ corresponds to a random step size s, which is a random number drawn from the Lévy distribution with the exponent λ.
- The fitness function $f(\mathbf{x}_i)$ is then evaluated.
- Select the j-th nest via $j = \lfloor U(0, \mathrm{Np}) \rfloor \wedge j \neq i$.

http://dx.doi.org/10.1016/j.applthermaleng.2014.11.009
1359-4311/

Algorithm 1 The original cuckoo search algorithm.

Input: Population of nests $\mathbf{x_i} = (x_{i1}, \ldots, x_{iD})^T$ for $i = 1 \ldots Np$, MAX_FE.
Output: The best solution $\mathbf{x}_{best}$ & its corresponding value $f_{min} = \min(f(\mathbf{x}))$.

```
generate_initial_host_nest_locations();
eval = 0;
while termination_condition_not_met do
  for i = 1 to Np do
    x_i = generate_new_solution(x_i);
    f_i = evaluate_the_new_solution(x_i);
    eval = eval + 1;
    j = ⌊rand(0,1) * Np + 1⌋;
    if f_j < f_i then
      x_i = x_j; f_i = f_j; // replace i-th solution
    end if
    if rand(0,1) < p_a then
      init_nest(x_worst);
    end if
    if f_i < f_min then
      x_best = x_i; f_min = f_i;    // save the local best solution
    end if
  end for
end while
```

- Replace the *i*-th solution with the *j*-th solution, if a better solution is found; i.e., $\mathbf{x}_i^{(t+1)} = \mathbf{x}_j^{(t)}$ if $f(\mathbf{x}_j) < f(\mathbf{x}_i)$.
- Keep the best solution, i.e., $\mathbf{x}_{\text{best}}^{(t+1)} = \mathbf{x}_i^{(t)}$ if $f(\mathbf{x}_i^{(t)}) < f(\mathbf{x}_{\text{best}}^{(t+1)})$.
- Stop when the termination condition is satisfied.

Currently, the CS has been applied to many optimization problems in practice. Readers are invited to refer to some reviews regarding the cuckoo search [6–8], where they can appreciate the breadth of applicability of the CS algorithms.

3. Cuckoo optimization algorithm (COA)

Cuckoo optimization algorithm (COA) was published in 2011 by R. Rajabioun [3], about two years after the publication of the original cuckoo search. COA was based on the egg-laying behaviour of cuckoo birds, and it can be considered as an evolutionary algorithm, though it is impropriate to be labeled as swarm-intelligence-based because it does not use interacting/swarming characteristics.

The main calculation in the algorithm is the egg-laying radius (ELR), and this ELR is a function of the total number of eggs, current cuckoo's eggs, and variable limits. That is

$$\text{ELR} = \alpha \times \frac{\text{number of current cuckoo's eggs}}{\text{total number of eggs}} \times (U_i - L_i), \qquad (1)$$

where U_i and L_i are the upper limit and lower limit of the variable, respectively. Here, α is an integer related to the maximum value of ELR. In essence, ELR is a pseudorandom variable as a fraction between the lower and upper limits.

The basic steps of the COA can be summarized as the pseudo-code 3 shown in Algorithm LABEL:alg:coa.

It is worth pointing out that apart from the definition of ELR, there is no explicit updating equation in COA. The implementation is guided by described procedure, much like genetic algorithms. Therefore, it is more proper to classify it as an evolutionary algorithm.

Algorithm 2 The cuckoo optimization algorithm.

```
Initialize cuckoo habitats with random points on the profit function;
Dedicate some eggs to each cuckoo;
Define ELR for each cuckoo;
Let cuckoos to lay eggs inside their corresponding ELR;
Kill those eggs that are recognized by host birds;
Let eggs hatch and chicks grow;
Evaluate the habitat of each newly grown cuckoo;
Limit cuckoos' maximum number in the environment and kill those who
  live in the worst habitats;
Cluster cuckoos and find the best group and select goal habitat;
Let the new cuckoo population migrate toward the goal habitat;
if (stopping condition is satisfied) then stop else go to 2 endif
```

Table 1
Differences between CS and COA.

Component	CS	COA
Representation of individuals	Nest & egg positions	Habitat (cuckoo position)
Number of eggs laid	1	5–10
Move operator	Lévi flights	Move within ELR
Local search	Random walk	Variable neighborhood search
Global search	Re-initialization of worst solutions	Migration of cuckoos
Replacement	One-to-one	Replace worst

4. Differences between the two cuckoo-based algorithms

There is no direct link between these two cuckoo-based algorithms, and the main steps are very different. However, confusion and mix-up often occur due to the naming similarity and careless writing. Though COA was published 2 years after the publication of the CS paper, it was quite surprising that the COA paper did not mention any previous literature.

To clarify any possible confusion, this section intends to emphasize the differences between both cuckoo-inspired algorithms, i.e., CS and COA. The main differences can be summarized in Table 1.

Although the representations of a solution are different in both algorithms (i.e., nests in CS and habitats in COA), it is essentially the same thing in real-world cuckoos. Initial solutions are the same dimensionality and are initialized randomly. A cuckoo in the CS algorithm lays one egg in a nest, while five to twenty eggs can be laid by a cuckoo in the COA. Furthermore, the move operator represents a great difference between both algorithms, because CS generates positions of new solutions in a more global manner. In contrast, the solutions in COA are generated in the neighborhood of the parent solutions. The local search improvement is performed by CS using the randomly selected solution from the population that can replace the current solution, while the COA generates a variable neighborhood of solutions scattered in the neighborhood of the current solution. The global search is represented by occasional re-initialization of the worst solution in the CS, while the step into undiscovered regions in the search space is realized by the migration operator in the COA. Finally, the solutions are replaced in the one-to-one manner in CS, while replacing the worst solutions is done in the COA.

As it can be seen from the above procedures, the seemingly similar behaviour and steps make it difficult for new researchers to see the differences clearly. However, differences are major and one algorithm should not be confused with another.

5. Discussion and conclusions

This brief discussion intends to clarify the confusion and mix-up concerning the two cuckoo-based algorithms in the literature. We highlighted the main differences between both algorithms. Due to the naming similarity and the similar source of inspiration, unfortunate confusion may occur. Therefore, both readers and authors should be careful when describing these algorithms and the literature accuracy should be ensured.

References

[1] M. Asadi, Y. Song, B. Sunden, G. Xie, Economic optimization design of shell-and-tube heat exchangers by a cuckoo-search-algorithm, Appl. Therm. Eng. 71 (1) (2014) 1030–1038.
[2] X.-S. Yang, S. Deb, Cuckoo search via lévy flights, in: Nature & Biologically Inspired Computing, NaBIC 2009. World Congress on, IEEE, 2009, pp. 210–214.
[3] R. Rajabioun, Cuckoo optimization algorithm, Appl. soft Comput. 11 (2011) 5508–5518.
[4] X.-S. Yang, S. Deb, Engineering optimisation by cuckoo search, Int. J. Math. Model. Numer. Optim. 1 (2010) 330–343.
[5] I. Fister Jr., X.-S. Yang, I. Fister, J. Brest, D. Fister, A brief review of nature-inspired algorithms for optimization, Elektrotehniški Vestnik. 80 (2013) 116–122.
[6] X.-S. Yang, S. Deb, Cuckoo search: recent advances and applications, Neural Comput. Appl. 24 (2014) 169–174.
[7] I. Fister Jr., X.-S. Yang, D. Fister, I. Fister, Cuckoo search: a brief literature review, in: Cuckoo Search and Firefly Algorithm, Springer, 2014, pp. 49–62.
[8] I.J. Fister, D. Fister, I. Fister, A comprehensive review of cuckoo search: variants and hybrids, Int. J. Math. Model. Numer. Optim. 4 (2013) 387–409.

Iztok Fister Jr.[a,*], Iztok Fister[a], Xin-She Yang[b]
[a] *University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia*

[b] *School of Science and Technology, Middlesex University, London NW4 4BT, UK*

[*] Corresponding author.
E-mail address: iztok.fister1@um.si (I. Fister).

22 September 2014
Available online 15 November 2014