Neural Comput & Applic
DOI 10.1007/s00521-016-2464-8

REVIEW

A comprehensive study of cuckoo-inspired algorithms

Mohamed Abdel-Basset[1] · Abdel-Naser Hessin[1] · Lila Abdel-Fatah[1]

Received: 24 May 2016 / Accepted: 1 July 2016

Abstract Nature-inspired metaheuristic algorithms are considered as the most effective techniques for solving various optimization problems. This paper provides a briefly review of the key features of the cuckoo-inspired metaheuristics: cuckoo search (CS) and cuckoo optimization algorithm (COA). In addition, it discusses some of their important and emerging studies, investigates their applications in several fields, and finally clarifies the differences between both algorithms so as to remove confusion between them.

Keywords Metaheuristic · Cuckoo search · Cuckoo optimization algorithm

1 Introduction

The adoption of artificial intelligence in solving different optimization problems is one of the most interesting trends in the last years especially for what concerns research on nature-inspired metaheuristics. Glover has introduced the term metaheuristic in 1986 [1] to define not problem-specific heuristic method. There are two major characteristics of metaheuristic [2]: the first one is intensification which tends to search around the current best solution, and the second is diversification which makes sure that the algorithm can explore the search space more efficiently, often by randomization. Metaheuristic will be successful in solving a given optimization problem if it can provide a balance between the search space exploitation and exploration to guarantee to find a high-quality solution for a given problem. The nature-inspired metaheuristic algorithms have simulated natural phenomena or natural forces such as bird flocking, ant or bee colonies, animal herding, water drops, and gravity, for example, genetic algorithm (GA) [3], ant colony optimization (ACO) [4], particle swarm optimization (PSO) [5], water wave optimization (WWO) [6], and gravitational search algorithm (GSA) [7].

One of these fabulous phenomena is the cuckoo bird's proliferation. The other side of the wonderful cuckoo chirp is the aggressive brood parasitism; i.e., the cuckoo uses a cruel deception strategy to lay its own egg in the other birds nest after removing the host bird egg. The cuckoo has the ability to mimic the color and the pattern of the host bird eggs to avoid the detection of its egg. If the host bird exposes the cuckoo's egg dumped in its own nest, it either throw out the exotic egg or abandons the nest to start afresh. After the egg hatches, the chick is cared and fed by the host bird until the chick growing. In addition, the chick instinct motivates it to throw out the host eggs or to propel the eggs out of the nest to ensure the food from the host bird.

The representation of the same phenomena varies according to the developer perspective, i.e., one phenomena can be inspected into various ways. For example, Yang and Deb [8] have introduced a metaheuristic "cuckoo search (CS)." Also, Rajabioun [9] has developed an emerging metaheuristic "cuckoo optimization algorithm (COA)" with a different narration.

In a completely different field, Pagh and Rodler [10] use the cuckoo inspiration to develop an algorithm for creating large hash tables with high space utilization and guaranteed constant access times which was called "cuckoo hashing."

✉ Mohamed Abdel-Basset
analyst_mohamed@yahoo.com

[1] Faculty of Computers and Informatics, Zagazig University, Zagazig, Egypt

Published online: 15 July 2016

Because this paper interested in the cuckoo-inspired metaheuristic, it focuses on CS and COA. The structure of this paper is as the following: Sect. 2 briefly outlines the main CS mechanism, Sect. 3 presents the main CS variants, and Sect. 4 reviews the major CS applications. Also for COA, Sect. 5 outlines the main COA mechanism followed by the presentation of main COA variants in Sect. 6 then the review of COA applications in Sect. 7. Finally, a comparison between CS and COA is conducted in Sect. 8 followed by conclusion in Sect. 9.

2 Cuckoo search algorithm (CS)

Cuckoo search is among the most widely used algorithms for optimization. Beside the cuckoo's behavior shown before, it based on the technique of Lévy flights rather than the simple isotropic random walks. Lévy flight is characterized by infinite mean and variance which cause much longer step from its current position, so it is more efficient in exploring the search space.

From the CS implementation attitude, each egg is a solution, and only one egg "solution" can be laid by each cuckoo. The aim is to substitute a not-so-good solution by potentially better solution. The CS is based on three basic rules as following:

1. One cuckoo egg is laid at a time and placed in a randomly chosen nest.
2. The best nests with a high quality of eggs "solutions" will be reserved to the next generations.
3. A host bird can detect a dumped egg in his nest with a probability of Pa $\in$ (0, 1).

The CS strength lies in combining a local search "intensification" with a global search "diversification" controlled by a switching parameter pa. The global search is performed when the cuckoo is searching for a nest to replace an egg with a new one "solution," while the local search is performed when the nest with the strange egg has been detected by the host bird and it should be replaced by a new one. If the egg of a cuckoo in the host bird's nest is very similar to the eggs of the host bird, and then, this cuckoo's egg is less likely to be discovered. Thus, the fitness should be related to the difference in solutions. Figure 1 shows how CS works.

Obviously, in the simple view of CS algorithm, there is no difference between cuckoo, egg or nest; all of them represent a potential solution in the search space. Another case of CS can be obtained where each nest has multiple eggs representing a set of solutions. So, CS has many advantages over conventional algorithms as it has been convenient to different problems even the multi-objective problems with less parameter usage.

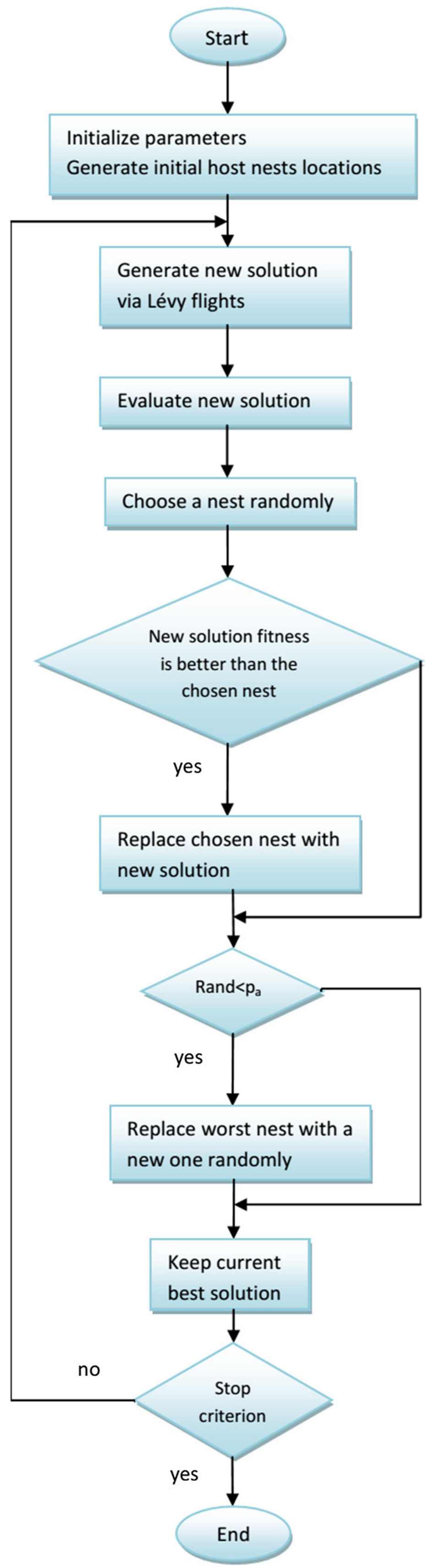


Fig. 1 Cuckoo search flowchart

In order to gain insight into CS search mechanisms, the following formulation has been introduced [11]. The local search can be expressed mathematically, as:

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \in) \otimes (x_j^t - x_k^t), \tag{1}$$

where x_j^t and x_k^t are two different solutions selected randomly by random permutation, $H(\mathrm{u})$ is a Heaviside function, $\in$ is a random number drawn from a uniform distribution, and s is the step size. For the global search, it can be formulated as:

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda), \tag{2}$$

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

where, $\alpha > 0$ is the step size scaling factor that should be related to the scales of the problem. In most cases, $\alpha = O(L/10)$ can be used, where L is the characteristic scale of the problem while in some cases $\alpha = O(L/100)$ can be more effective and avoid flying away. Note that the step size scaling factor has been written as α in Eqs. (3) and (4) regardless the difference in value instead of using two various parameters α_1 and α_2. For simplicity, it has assumed that ($\alpha_1 = \alpha_2 = \alpha$). In addition, "$\sim$" has been written instead of "=" so as to denote drawing random numbers from the distribution; not calculating by using a power law.

3 Cuckoo search variants

Since the original CS appearance in 2009, hundreds of CS studies have been developed. Many researchers have introduced several CS variants which include modification in CS parameters for increasing its efficiency and robust; or hybridization of it with other algorithms to merit different algorithms capabilities. Next, some selected variants will be briefly discussed.

3.1 Modified cuckoo search

3.1.1 Modified cuckoo search (MCS)

Walton et al. [12] have applied two modification to the CS algorithm: the first modification is made to the size of the Lévy flight step size α which has been constant $(\alpha) = O(1)$ has been used by Yang and Deb [8]. The second modification is the information exchanging between the eggs.

In MCS, the eggs are sorted according to their fitness and categorized in two different groups where the high fitness eggs are put into "the top egg group" and the others are put in "the abandoned group". For the abandoned group, α is calculated at each generation by the following formula:

$$\alpha = A/\sqrt{G} \tag{4}$$

where A is the Lévy flight step size (initially $A = 1$) and G is the number of generation.

Obviously, α is inversely related to the number of generations that lead to more local search when getting closer to the solution. For each top egg, another top egg is chosen randomly and a new egg position is generated on the line connecting these two top eggs. This line distance is calculated using the inverse of the golden ratio as:

$$\varphi = (1 + \sqrt{5})/2 \tag{5}$$

If the same egg is picked twice, a local Lévy flight search is done as:

$$\alpha = A/G^2 \tag{6}$$

Also, if the chosen top eggs have the same fitness, the new egg is generated at the midpoint.

3.1.2 Modified adaptive cuckoo search (MACS)

Modified adaptive cuckoo search [13] is another modified version of CS which incorporates the original CS with the following strategies: grouping, parallel, incentive, adaptive, and information sharing. The grouping strategy means that dividing the population into subgroups in order to increase the survival rate of some low-fitness eggs for population diversity, i.e., the grouping strategy prevent some new low-fitness eggs with better gene fragments from easily abandoned in the early stage after their appearance. The grouping strategy can be expressed as:

$$x_j = (x_1, \ldots, x_D)^T \tag{7}$$

$$P_{\mathrm{sk}} = \left\{x_j | 1 \le j \le N_e \text{ and } \quad j \in Z\right\} \tag{8}$$

$$P = \{P_{\mathrm{sk}} | 1 \le k \le N \quad \text{and} \quad k \in Z\} \tag{9}$$

where x_j denotes the jth subgroup member, P_{sk} is the kth subgroup, N_{e} is the subgroup size, and P is the population set.

On each subgroup, cuckoos search the neighborhood space of subgroup for the best solution. If the new solution is better than the worst subgroup solution, cuckoos substitute it with the new one and the successful flight wins a successive flight, until the flight fails to increase the exploitation rate of promising areas "incentive," i.e., there are equal numbers of cuckoos doing a parallel search which enhances the solution space exploration. This strategy searches not only the neighborhoods of the whole population best solution but also the neighborhoods of some worse solutions as well. At the end of each generation, the population is sorted by their fitness value and divided into new subgroups so that the individuals can be exchanged between different subgroups to achieve information sharing. In addition, MACS has used an adaptive Lévy flights step size to adjust search range in different stages.

$$\alpha = \alpha_{\min} + \left(\frac{\mathrm{FE}_{\max} - \mathrm{FE}}{\mathrm{FE}_{\max}}\right)^{m} \cdot (\alpha_{\max} - \alpha_{\min}) \tag{10}$$

$$\alpha_{\min} = \frac{(\mathrm{ub} - \mathrm{lb})}{100} \tag{11}$$

$$\alpha_{\max} = \frac{(\mathrm{ub} - \mathrm{lb})}{4000} \tag{12}$$

where FE is the number of function evaluation, $\mathrm{FE}_{\max}$ is the maximum number of FE, $\alpha_{\max}$ and $\alpha_{\min}$ are maximum and minimum step size, respectively, m is the nonlinear coefficient, and ub and lb are the upper and lower bounds of the selected problem.

3.2 Chaotic cuckoo search

Chaos maps are evolution functions that produce a deterministic bounded sequence of random numbers depending on initial condition. They can have discrete or continuous time domain. There are different types of maps like logistic map, Chebyshev map, tent map (see Table 1).

Chaotic metaheuristics are recently used for more randomness and escaping from local optima. The main idea of integrating chaos maps with the procedures of metaheuristics is availing the chaos maps features. As well, chaos maps can be used instead of the random sequence generator because of being random like, non-period, and non-converging for parameter adaptation.

3.2.1 Emotional chaotic cuckoo search (ECCS)

Lin and Lee [14] have introduced an integration between the Lévy flight and the chaos maps in order to improve the search process. As the Lévy flight has infinite mean and variance, it can lead to a premature search process. To avoid that, the chaotic Lévy flight has used the logistic map to generate a chaotic sequence for the Lévy flight parameter, as:

$$c_s(t+1) = 4.0 \times c_s(t) \times (1 - c_s(t)), \quad 0 \le c_s(0) \le 1 \tag{13}$$

where c_s is the chaotic sequence.

Then, the α parameter in Eq. (1) will be replaced by the generated c_s. Also, a new emotional acceptance criterion is added to prevent getting trapped into local optima. For ECCS, there are two cuckoos' emotions (positive and negative) which correspond to two reactions as follow:

$$e_s = -k \ln\left|\frac{S(F(x_i) - F(x_j))}{S_0}\right|$$

$$\mathrm{IF}(c_s < e_s) \quad \begin{array}{l} \text{THEN positive} \\ \text{ELSE negative} \end{array} \tag{14}$$

where e_s is the cuckoo's emotion, k is a constant factor, S is the stimulus function, S_0 is a stimulus threshold, and F is the fitness function.

3.2.2 Chaotic cuckoo search (CCS)

Like ECCS, the proposed CCS [15] has replaced the value of α, but with 12 different one-dimensional chaotic maps. In addition, an elitism strategy is introduced to protect the best cuckoos from being corrupted by cuckoo updating operator. So, even if cuckoo updating operation destroys its corresponding cuckoo, the best cuckoos can be reverted back if needed. The authors have shown that the performance of CCS is more effective with the sine map and sinusoidal map than others. Regarding these two maps, on average, the sinusoidal map is significantly better than the sine map on most benchmarks. Also, it can provide more information for CS to guide its search while other chaotic maps cannot. When multiple runs are made, it yields a slight difference output with the optimal value. So, it has been selected as the final optimal map to be used for chaotic CS (CCS).

3.3 Improved cuckoo search (ICS)

The main idea behind ICS [16] is adjusting the value of two fixed parameters P_a and α; i.e., in the original CS, the values of P_a and α are fixed which may lead to significant increase in the iteration number if the value of pa is small and the value of α is large. Also, if the value of P_a is large and the value of α is small, it may lead to a premature convergence. The ICS algorithm has identified big enough values of P_a and α in the early generations to increase the diversity of solution vectors. However, these values are decreased in final generations to result in a better fine tuning of solution vectors. The values of P_a and α are dynamically changed with the number of generation as following:

$$P_a(N_G) = P_a^{\max} - \frac{(P_a^{\max} - P_a^{\min})}{N_{\mathrm{iter}}} \times N_G \tag{15}$$

$$\alpha(N_G) = \alpha_{\max} \exp(c N_G) \tag{16}$$

$$c = \frac{\ln(\alpha_{\min}/\alpha_{\max})}{N_{\mathrm{iter}}} \tag{17}$$

where N_{iter} denotes the number of total iterations and N_G is the current iteration.

Table 1 Chaos maps

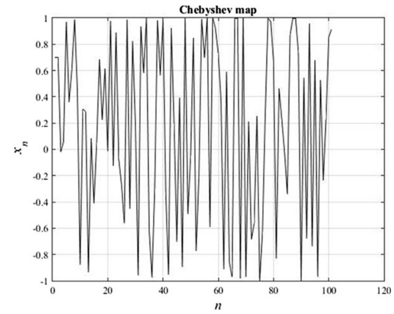
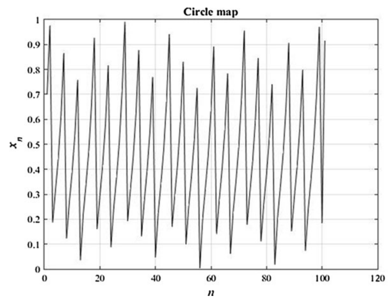
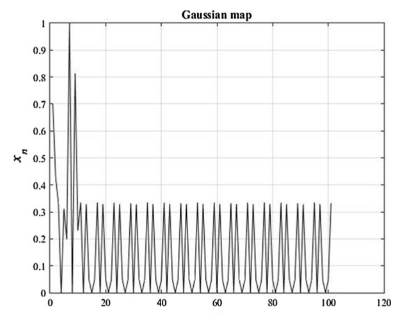
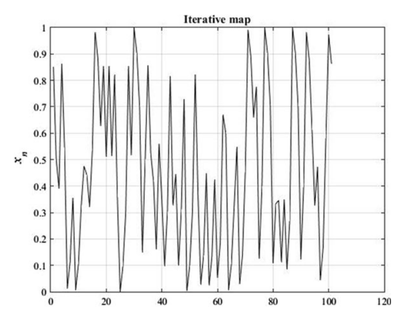
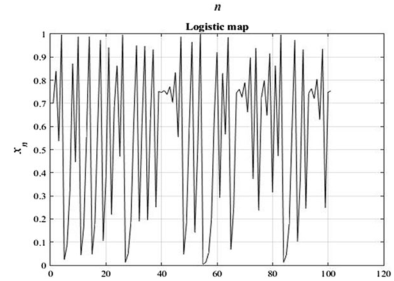
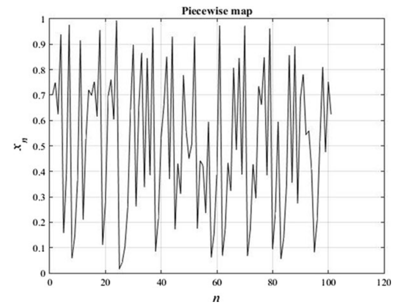
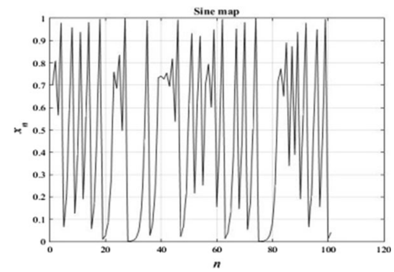
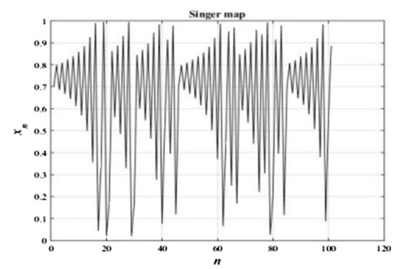
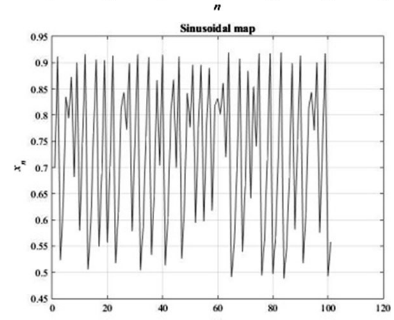
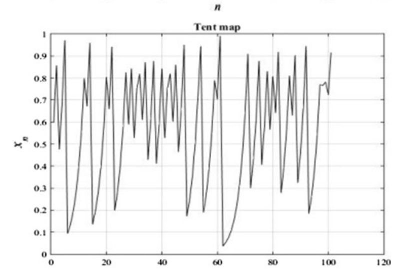
No.	Map name	Definition	Modality
1.	Chebyshev	$x_{n+1} = \cos(k \cos^{-1}(x_n)), \quad x \in (-1, 1)$, where k is a constant	
2.	Circle	$x_{n+1} = x_n + b - (a/2\pi)\sin(2\pi x_n) \bmod (1)$ where $a = 0.5$ and $b = 0.2$, with range (0, 1)	
3.	Gaussian	$x_{n+1} = f\,(x) = \begin{cases} 0, & x_n = 0 \\ \dfrac{1}{x_n} \bmod (1), & \text{otherwise} \end{cases}$	
4.	Iterative	$x_{n+1} = \sin\left(\frac{a\pi}{x_n}\right), \quad a \in (0, 1)$	
5.	Logistic	$x_{n+1} = a x_n (1 - x_n), \quad x \in (0, 1), \quad 0 < a \leq 4$	

Table 1 continued

No.	Map name	Definition	Modality
6.	Piecewise	$x_{n+1} = \begin{cases} \frac{x_n}{p}, & 0 \le x_i < p \\ \frac{x_n - p}{0.5} - p, & p \le x_n < 0.5 \\ \frac{1 - x_n - p}{0.5} - p, & 0.5 \le x_n < 1 - p \\ \frac{1 - x_n}{p}, & 1 - p \le x_n < 1 \end{cases}$	
7.	Sine	$x_{n+1} = \frac{a}{4}\sin(\pi x_n), x \in (0,1), 0 < a \le 4$	
8.	Singer	$x_{n+1} = \mu\left(7.86x_n - 23.31x_n^2 + 28.75x_n^3 - 13.302875x_n^4\right)$, $\mu = 1.07$	
9.	Sinusoidal	$x_{n+1} = ax_n^2 \sin(\pi x_n), \quad a = 2.3$	
10.	Tent	$x_{n+1} = f(x) = \begin{cases} \frac{x_n}{0.7}, & x_n < 0.7 \\ \frac{10}{3}(1 - x_n), & x_n \ge 0.7 \end{cases}$	

3.4 Multi-objectives cuckoo search (MOCS)

The use of traditional single-objective metaheuristics has become insufficient for multiple conflicting objectives of a wide range of optimization problems. Therefore, the need of multi-objective optimization has been raised. Multi-objective optimization can reconcile between a set of alternatives instead of a single/best solution of single-objective metaheuristic "Pareto front" which subject to a set of constraints. Also, it could be also called "vector optimization," "multicriteria optimization," "multiattribute optimization" or "Pareto optimization." The proposed algorithm [17] emulates some of the evolutionary computation concepts which make MOCS more efficient due to diversifying of the search. MOCS based on three main rules:

1. A number K of cuckoo eggs are laid at a time and placed in a randomly chosen nest.
2. The best solutions will be reserved to the next generations "elitism".
3. Each nest will be abandoned with a probability pa, and a new nest with K eggs will be built, according to the similarities or differences of the host eggs. Some random mixing can be used to generate diversity. This implies that n nests are replaced by new nests randomly with the fraction pa. "mutation."

Obviously, the first and last rules should be modified to incorporate multi-objective criteria which has K different objectives. The authors have shown that according to the first rule, the new solutions are generated by performing random walks or Lévy flight. Simultaneously, a form of crossover is performed over solutions through a localized random permutation. For each nest, there can be K solutions which are generated using Eq. (18). In addition, the mutation in the last rule is a vectorized operator that combines Lévy flight and differential quality of the solutions.

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Le vy}(\beta) \tag{18}$$

$$\alpha = \alpha_0 (x_j^t - x_i^t) \tag{19}$$

$$\text{Lévy} \sim u = t^{-1-\beta}, \quad (0 < \beta \leq 2) \tag{20}$$

where $(x_j^t - x_i^t)$ is the difference between two randomly solutions to denote the difference between solution quality.

Also, the generation of step size s samples is a very important operation which is performed using Lévy flight as:

$$s = \alpha_0 \left(x_j^t - x_i^t\right) \oplus \text{Le vy}(\beta) \sim 0.01 \frac{u}{|v|^{1/\beta}} \left(x_j^t - x_i^t\right) \tag{21}$$

where μ and v follow the normal distribution as:

$$u \sim N(0, \sigma_\mu^2) \quad v \sim N(0, \sigma_v^2) \tag{22}$$

with

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left[\frac{1+\beta}{2}\right] \beta 2^{(\beta-1)/2}} \right\}, \quad \sigma_v = 1 \tag{23}$$

where Γ is the standard gamma function.

3.5 Hybridization of CS

The hybridization of CS with other techniques (optimization algorithms, machine learning techniques, heuristics, etc.,) is a prevailing research area because the high efficiency of these algorithms which makes it possible to apply them for solving a wide range of optimization problems. In each part of CS, the hybridization can be carried out. For example:

3.5.1 CS/PSO algorithm

Ghodrati and Lotfi [18] have proposed hybridization between CS and the particle swarm optimization (PSO) algorithm [5]. The proposed algorithm has mimic PSO swarm intelligence to add a communication between cuckoos to inform each other about their positions and help each other to immigrate to a better place.

3.5.2 CS–GA algorithm

Kanagaraj et al. [19] have hybridized CS with the genetic algorithm (GA) [3]. In each CS–GA generation, every two-parent cuckoos will reproduce two eggs instead of one cuckoo egg at a time; the crossover and mutation are performed, respectively. Also, some form of elitism is applied after each operation. Before the generation end, the best solution in the current population will perform Lévy flight to increase the search space exploration for the next generation. At each generation, the algorithm creates a new population via genetic principles "natural selection."

3.5.3 CS/ACO algorithm

In [20] study, the main disadvantage of ant colony optimization (ACO) [4] has been handled by CS, i.e., the ants deposit chemical trail called pheromone on the ground in order to guide the other ants toward the source food. The pheromone quantity left by one ant relies on the amount of food found and the number of ants using this trail. Moreover, this trail has a decreasing action over time "trail evaporation." To accelerate the search process, CS can perform the local search more efficient and faster. Also, there is only a single parameter apart from the population size in CS.

3.5.4 MKF-Cuckoo algorithm

Binu et al. [21] have introduced a new hybrid algorithm "MKF-Cuckoo" which is the hybridization of cuckoo search algorithm with the multiple kernel-based fuzzy c means algorithm (MKFCM) [22]. The objective of MKF-Cuckoo is to find the best centroid in the search space. The proposed algorithm depends on three main phases: (1) the initial solution encoding process by choosing random centroid taken from the input dataset that has represented as $m * k$ matrix, (2) defining the fitness function as minimization function which makes use of kernel-based distance and fuzzy membership function to find the minimum distance between the data points with its nearest neighbor cluster centroid, (3) applying CS approach.

3.5.5 TLCS algorithm

A new hybrid algorithm named teaching–learning-based cuckoo search (TLCS) is proposed by Huang et al. [23]. TLCS has hybrid CS with teaching–learning-based optimization (TLBO) [24]. The main TLCS idea is dividing the population into two subgroups (top solutions P1 and abandoned solutions P2), then applying TLBO to P1 group and the Lévy flight to P2 group.

3.5.6 CS/ABC algorithm

Singla et al. [25] have introduced a hybridization between CS and artificial bee colony optimization (ABC) [26] algorithms to improve the search strategy of CS by using the ABC global search strategy which depends on filtering the solution from the all the bees (onlooker and Scout bees) to give the quality of the food Nectar.

3.5.7 CSHS algorithm

Sheikholeslami et al. [27] have combined CS with a harmony search algorithm (HS) [28] to introduce a new double stage optimization algorithm (CSHS). The framework of CSHS has been used to get rid of CS drawbacks: premature convergence due to random walk and no communication between individuals; i.e., first CS is used for the initial exploration of the search space then a hybridization of CS and HS approaches is employed in the exploitation phase.

3.5.8 CS–IDE algorithm

Zhang et al. [29] have combined CS with the improved differential evolution (IDE) [30] algorithm to merge CS exploration and IDE exploitation capabilities. IDE is an improved version of the traditional differential evolution (DE) [31] algorithm which has the benefits of evolutionary algorithms (mutation, crossover, selection, and elitism). IDE was chosen instead of DE for the sake of higher accuracy as IDE has an adaptable step size which is updated dynamically according to the various generation results.

3.5.9 CCS based on SFLA

Liu and Fu [32] have proposed a new improved CS that utilizes three enhancement methodologies: (1) chaos maps (the Logistic and Tent maps) have been used for generating initial population for more randomization, (2) inertia weight has been used in the Lévy flight to improve exploration, and (3) a hybridization of CS and the local search technique of shuffled frog leaping algorithms (SFLA) [33] has been used to improve the convergence rate.

3.6 Discretizing of CS

There are numerous optimization problems that cannot be solved by using continuous metaheuristic. Therefore, the researchers have to develop different discretization methods to accommodate discrete and combinatorial problems such as traveling salesman problem, facility location problem, make span scheduling problem, and knapsack problem.

For CS, several discretization methods have been proposed. Table 2 shows the main methods that have been used for discretizing CS.

4 Cuckoo search applications

Due to the CS simplicity and flexibility, it has been applied for solving many optimization problems in various fields. In this section, a small fraction of the relevant applications will be listed in Table 3.

5 Cuckoo optimization algorithm (COA)

The COA basic idea is the survival of cuckoo population (mature cuckoos and eggs) in different societies which lead to the pursuit for immigration to a better environment. As discussed before, cuckoos dump eggs into different host nests. These eggs have the chance to grow up if they are more similar to the host bird's eggs, if not they will be killed. After eggs growing, they become mature cuckoo and begin to make societies. Each society has its habitat

Table 2 Major CS discretization methods

Authors and year	Discretization method	References
Zheng et al. (2012)	Transform a COA continuous search space into a binary one by using the sigmoid function (SF): $x_{ij} = f(x) = \begin{cases} 1, & \text{rand}() \leq \dfrac{1}{1+\exp(-x_{ij})} \\ 0, & \text{otherwise} \end{cases}$	[34]
Burnwal and Deb (2013)	The nearest integer (NI):	[35]
	Real values are converted to NI by rounding or truncation	
Ouaarab et al. (2015)	Random key (RK) encoding scheme:	[36]
	uses a vector of random numbers then assigning a weight to each element in the vector (in ascending order). These weights are used to generate one combination as a solution	
Dasgupta and Das (2015)	The smallest position value (SPV):	[37]
	SPV is similar to RK, but it depends on element permutation; i.e., it places the weight of the lowest valued element as the first item on a permutated solution, the next lowest as the second and so on	
Li and Yin (2013)	The largest ranked value (LRV) rule-based random key:	[38]
	LRV is similar to SPV, but the permutation is carried out according to the largest element value	
Guo et al. (2015)	Redefine Lévy flight operations as:	[39]
	The subtraction is replaced by a comparison strategy	
	The addition is replaced by a number of swap moves	
	The multiplication: For each solution vector, a random number is generated for each element, and then it will be compared with the scalar $\sigma \sim = t^{-\lambda}$ to take the corresponding decision	

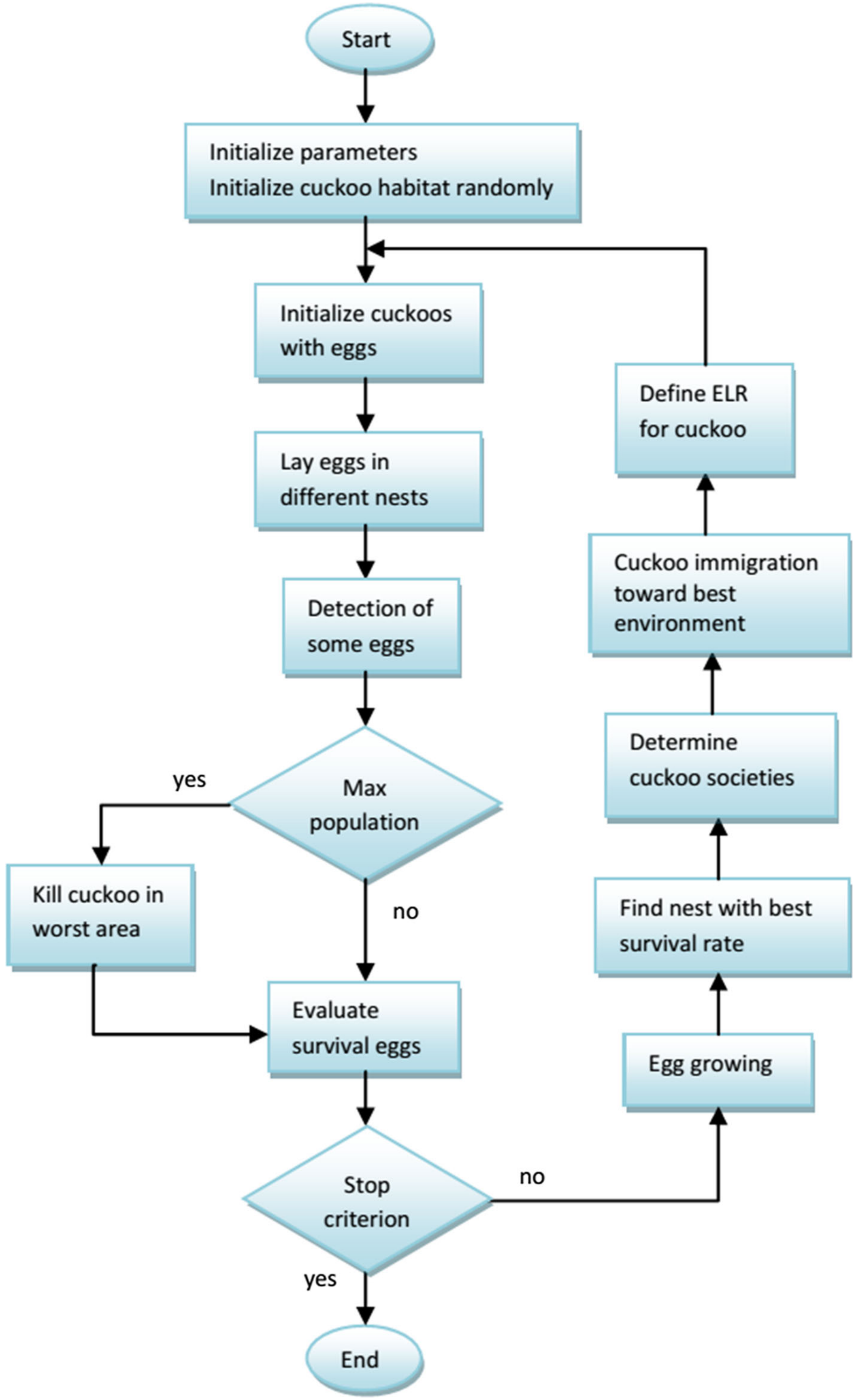
Table 3 CS main applications

Field	Applications
Bioinformatics	Training Artificial Neural Networks [40], Cloud Computing Optimization [41]
Computer Science	Web Document Clustering [42, 43]
Engineering	Structure Designing And Machining Processes [23], Design Of Water Distribution Systems [27], Spectrum Parameter Estimation In Brillouin Scattering Distributed Temperature Sensor [29], Load Frequency Control (LFC) [44], Flexible Single-Link Manipulator [45], Wind Farm Reliability Optimization [46], The Design Process Of Integrated Power Systems For Modern And Energy Self-Sufficient Farms [47], Integrated Circuits (ICS) [48], Network Distribution [49], Performance Analysis [50]
Operations Research	Reliability–Redundancy Allocation Problems [19], Job Scheduling [20], Knapsack Problem [34], Traveling Salesman Problem [36], Scheduling Optimization of Flexible Manufacturing System [35], Flow Shop Scheduling Problems [37] [38], Parallel Machine Scheduling with Step-Deteriorating Jobs and Setup Times [39], Economic Dispatch Problems [51] [52], The Capacitated Vehicle Routing Problem (CVRP) [53], Forecast The Annual Foreign Tourist [54]
Information Science	Image Segmentation [55–57], Satellite Image Classification [25], Noise Suppression and Enhancement of Speech Signal [58], Image Compression [59], Tsallis Entropy Image Processing [60], Face Recognition [61], Cryptanalysis of Vigenere Cipher [62], Optimum Wavelet Based Masking (OWBM) [63]
Geolocation	Mobile Object Tracker [64]
Mathematics	Parameter Estimation of Takagi–Sugeno Fuzzy System [65]

region to live in. Cuckoos are always searching for the best habitat which has more eggs survival rate to be their destination for laying eggs. Then each cuckoo starts to lay an egg inside her egg laying radius (ELR) in some nests randomly. The algorithm stops when all the survived cuckoos converge to the same position (see Fig. 2). To apply COA, the following rules should be considered:

1. The values of any optimization problem variables should be formed as an array called "habitat;" i.e., In a N_{var}—dimension optimization problem, a habitat can be expressed as an array of $1 \times N_{\text{var}}$, which representing the current location of the survival cuckoos.

Fig. 2 COA flowchart



$$\text{Habitat} = [x_1, x_2, \ldots, x_{N_{var}}] \tag{24}$$

2. At different iterations, the lower and upper bounds of cuckoo eggs are 5 and 10, respectively.
3. Each cuckoo lay eggs within a maximum distance from their habitat called "Egg Laying Radius (ELR)." ELR depends on the total number of eggs, number of current cuckoo's eggs, and also variable limits (see Fig. 3).

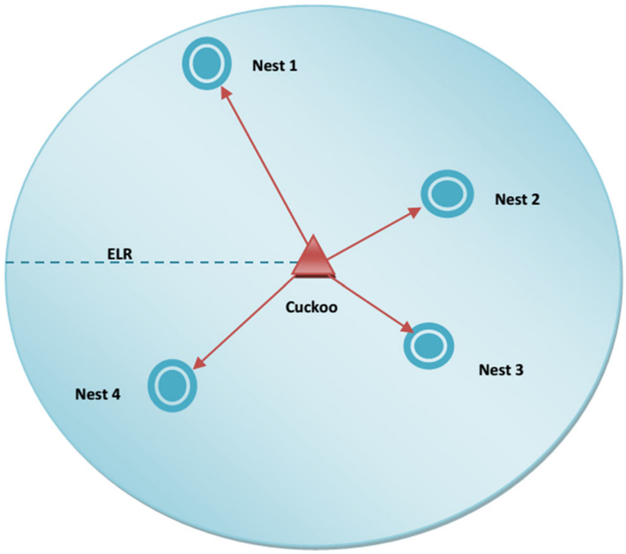


Fig. 3 Random egg laying in ELR

ELR can be defined as:

$$\mathrm{ELR} = \alpha \times \frac{\text{number of current cuckoos eggs}}{\text{total number of eggs}} \times (\mathrm{var}_{\mathrm{hi}} - \mathrm{var}_{\mathrm{low}}) \tag{25}$$

where α is an integer for controlling ELR maximum value and $\mathrm{var}_{\mathrm{hi}}$ and $\mathrm{var}_{\mathrm{low}}$ are variable limits.

4. After egg laying, p % of all eggs (usually 10 %), with less profit (more cost) values, will be killed.
5. The grouping of cuckoos' societies is done with K-means clustering method (a k of 3–5 seems to be sufficient in simulations).
6. For immigration of cuckoos, each cuckoo only flies λ% of all distance toward goal habitat and also has a deviation of φ radians. These two parameters λ and φ help cuckoos search much more positions in all environment (see Fig. 4). For each cuckoo, λ and φ are defined as follows:

$$\lambda \sim U(0, 1) \tag{26}$$

$$\varphi \sim U(-\omega, \omega) \tag{27}$$

where λ is a random number and ω is a parameter for deviation controlling ($\omega = \frac{\pi}{6}$ seems to be enough).

6 COA variants

As COA is less popular than CS, few COA variants have been proposed, for example:

6.1 A modified cuckoo optimization algorithm (MCOA)

Instead of the fixed ELR in the original COA, Kahramanli [66] has used a modified ELR equation to gradually reduce the large ELR value, it can be formulated as:

$$t = \frac{\mathrm{max\_iter}}{c} \tag{28}$$

$$e = \mathrm{var}_{\mathrm{hi}} - \mathrm{var}_{\mathrm{low}} \tag{29}$$

$$a = \lfloor \mathrm{iteration}/t \rfloor + 1 \tag{30}$$

$$e_{\mathrm{new}} = \frac{e_{\mathrm{old}}}{a} \tag{31}$$

$$\mathrm{ELR} = \alpha \times \frac{\text{number of current cuckoo's eggs}}{\text{total number of eggs}} \times e_{\mathrm{new}} \tag{32}$$

where max_iter is the maximum number of iterations defined for the selected problem, c is constant inside the interval (0, 0.20]. (or it can be adjusted to fit the selected problem), and the bracts $\lfloor\,\rfloor$ means the truncation of the output.

6.2 Hybridization of COA

6.2.1 *COA/TS algorithm*

Dejam et al. [67] have combined COA with the tabu search algorithm (TS) [68] concept; i.e., the basic idea of TS is the prohibition "tabu or taboo" of already visited solution space from being visited again to increase diversification, i.e., TS applies any local search procedure intensively to explore the solution space and avoid getting trapped in local optima. During the local search, the best solution in the neighborhood is selected in each iteration as the new current solution by managing a memory of solutions or moves recently applied, called the tabu list to avoid the siege within cycles.

6.2.2 *COGSA algorithm*

Naseri [69] has introduced a hybridization between COA and gravitational search algorithm (GSA) [70]. GSA is a meta-heuristic optimization algorithm motivated by the Newton's laws of gravity and motion. According to GSA, solutions are considered as objects and their performance is measured by their masses which proportional to their value of fitness function. Every object attracts every other object with gravitational force. In the proposed algorithm, the author has used GSA for exploiting and COA for global exploration the search space.

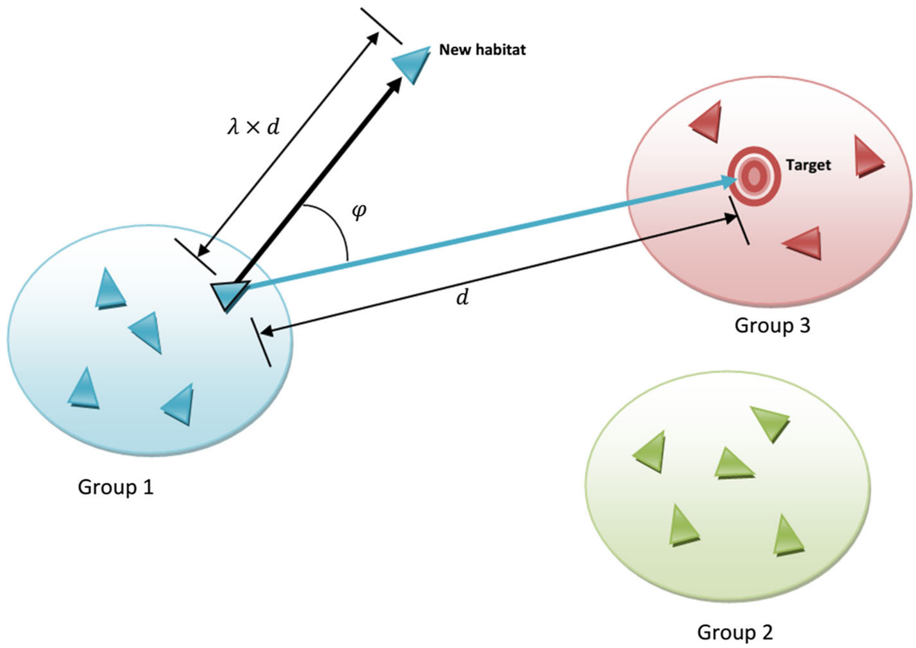


Fig. 4 Explanation of cuckoo Immigration

6.2.3 COA/PSO based on ES

Sarvi et al. [71] have proposed a combination between PSO and COA based on the Eagle strategy (ES) [72] approach; i.e., ES is a two-stage optimization algorithm that is conventional for applying to different purposes with a balanced combination of optimization algorithms. In the first stage, PSO performs initial exploration of the search space and the best solution of the first stage is transmitted to the second stage. In the next stage, COA searches for the global best solution around the first stage best solution.

6.3 Discretizing of COA

Table 3 summarizes the main COA discretization method (Table 4).

7 COA applications

Recently, COA has been applied in many areas of optimization because of the increased awareness about its promising efficiency. For the sake of conciseness, only a sample of COA main applications will be mentioned in Table 5.

8 Comparison between CS and COA

There is some confusion and misunderstanding of CS and COA in the literature, Fister and Yang [113] have provided a brief comparison between CS and COA so as to remove ambiguity and confusion between the concept of both algorithms. Also, Chen et al. [114] have made a comparison between the two algorithms, but this comparison was based on an "artificial neural networks (ANN)" case study. The results show that ANN–COA obtained slightly better results in this case. Table 6 shows the main difference between the two algorithms. In the last 2 years, the awareness of COA effectiveness has reflected on the number of publications that interested in employing COA for solving various problems. Figure 5 shows the percentage of CS and COA publications (forecast).

9 Conclusion

Although the nature-inspired metaheuristics are not well understood regarding why these algorithms work efficiently, they are the most efficient and widely used solution techniques, especially, the evolutionary algorithms such as the previously discussed algorithms CS and COA. However, metaheuristics are not problem specific, choosing the appropriate metaheuristic for the given problem should be considered, i.e., the more adaptation to the given problem, the more efficient metaheuristic. This is, what characterizes CS since it owns a few parameters that need to be adjusted and as shown before CS has many variants which have been applied to numerous application areas. For COA, it has proved its efficiency in solving several problems, especially ANN. But it is still promising and needs more research.

This paper gives an overview of CS and COA main structures, variants, and applications which has been

Table 4 Major COS discretization methods

Authors and year	Discretization method	Reference
Mahmoudi et al. (2013)	transform a COA continuous search space into a binary one by using the sigmoid function $(\mathrm{SF}){:}x_{ij} = f(x) = \begin{cases} 1, & \mathrm{rand}() \leq \dfrac{1}{1+\exp(-x_{ij})} \\ 0, & \text{otherwise} \end{cases}$	[73]
Kazemi and Dejam (2014)	Random-key (RK) encoding scheme: Uses a vector of random numbers then assigning a weight to each element in the vector (in ascending order). These weights are used to generate one combination as a solution. Then, the concepts of distance and geometric rules should be considered by redefining: The new solution generation: Three different operators are defined and for each solution vector, one operator is selected randomly and applied to determine the new position. the three operators are: (1) Permutation: replacing two points selected randomly (2) Sequence order Inversion: several sequential positions are selected and reversed (3) Shifting The migration operator: find the positional Difference between two habitats (the goal point and the current habitat)	[74]
Mousavirad and Ebrahimpour-Komleh (2014)	The nearest integer (NI): A real values are converted to NI by rounding or truncation	[75]

Table 5 CS main applications

Field	Applications
Bioinformatics	Artificial Neural Networks [76–78], Cloud Computing Optimization [79], Feature Selection [80]
Computer Science	Soft Sensor [81], Clustering [82, 83], The Graph Coloring Problem (GCP) [84]
Engineering	Connecting Rod [85], Liquid Level Control [86], Energy Management [87], The Crack Detection Problem [88], Circular Antenna Arrays Design [89], The Selective Harmonic Elimination (She) Problem [90], Design A Hybrid Wind Turbines/Solar Arrays/Fuel Cells System [91], Machining [92], Power Point Control [71, 93], Optimal Power Flow (OPF) Problem [94], Optimal Synchronization Of Teleoperation Systems [95], Optimum Operation Of Reservoir [96], Estimation Of Induction Motor Efficiency Value [97], Inverse Kinematics In Robotics [98]
Operations Research	Quadratic Assignment Problem [67, 74], Production Planning Problem [99], Urban Bus Terminal Location Problem [100], Economic Dispatch [101, 102], Optimal Replacement of Dependability Activities [103], Job Scheduling [104], Coordination of Overcurrent and Distance Relays Optimization [105], Facility Layout Problem (FLP) [106]
Geophysics	Gravity Method [107, 108]
Information Science	Image Segmentation [109, 110], Image Enhancement [111], Image Compression [112]

Table 6 Major difference between CS and COA

Concept	CS	COA
Candidate solution representation	Nest and Egg position	Cuckoo position in habitat
Number of eggs laid	1	5–10
Movement	Lévy flights	ELR
Local search	Random walk	Variable neighborhood search (VNS) [115]
Global search	Redefining the worst solutions	Migration
grouping	No grouping involved	K-means clustering method
Replacement	One-to-one	Replace worst

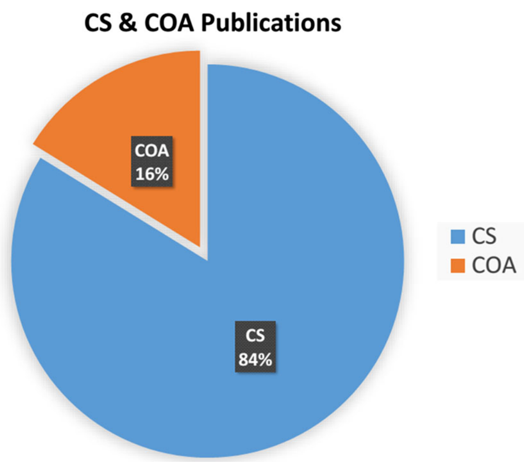


Fig. 5 Percentages of CS and COA publications

supported by a comparison between both algorithms to help the researchers deeply understand the difference between CS and COA algorithms in a simple way.

For future works, there are many applications that can be solved potentially by CS and COA, such as robotics and geographic information system (GIS). Hybridization of cuckoo-inspired algorithms and variants with other methods can be also useful.

References

1. Glover F (1986) Future paths for integer programming and links to artificial intelligence. Comput Oper Res 13(5):533–549
2. Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv (CSUR) 35(3):268–308
3. Sampson JR (1976) Adaptation in natural and artificial systems (John H. Holland). SIAM Rev 18(3):529–530
4. Dorigo M (1992) Optimization, learning and natural algorithms. Ph. D. Thesis, Politecnico di Milano, Italy
5. Kennedy J, R Eberhart (1995) Particle swarm optimization 1
6. Zheng YJ (2015) Water wave optimization: a new nature-inspired metaheuristic. Comput Oper Res 55:1–11
7. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
8. Yang XS, Deb S (2009, December) Cuckoo search via Lévy flights. In: Nature and biologically inspired computing, 2009. NaBIC 2009. World Congress on IEEE, pp 210–214
9. Rajabioun R (2011) Cuckoo optimization algorithm. Appl Soft Comput 11(8):5508–5518
10. Pagh R, Rodler FF (2001) Cuckoo hashing. Springer, Berlin, pp 121–133
11. Yang XS, Deb S (2014) Cuckoo search: recent advances and applications. Neural Comput Appl 24(1):169–174
12. Walton S, Hassan O, Morgan K, Brown MR (2011) Modified cuckoo search: a new gradient free optimisation algorithm. Chaos Solitons Fractals 44:710–718
13. Zhang Y, Wang L, Wu Q (2012) Modified adaptive cuckoo search (MACS) algorithm and formal description for global optimisation. Int J Comput Appl Technol 44(2):73–79
14. Lin JH, Lee HC. (2012). Motional chaotic cuckoo search for the reconstruction of chaotic dynamics. In: source: 11th WSEAS international conference on computational intelligence, man-machine systems and cybernetics (CIMMACS'12), pp 123–128
15. Wang GG, Deb S, Gandomi AH, Zhang Z, Alavi AH (2014, September) A novel cuckoo search with chaos theory and elitism scheme. In: Soft computing and machine intelligence (ISCMI), 2014 international conference on IEEE, pp 64–69
16. Valian E, Mohanna S, Tavakoli S (2011) Improved cuckoo search algorithm for global optimization. Int J Commun Inform Technol 1(1):31–44
17. Yang XS, Deb S (2013) Multiobjective cuckoo search for design optimization. Comput Oper Res 40(6):1616–1624
18. Ghodrati A, Lotfi S (2012) A hybrid CS/PSO algorithm for global optimization. In: Intelligent information and database systems. Springer, Berlin, pp 89–98
19. Kanagaraj G, Ponnambalam SG, Jawahar N (2013) A hybrid cuckoo search and genetic algorithm for reliability–redundancy allocation problems. Comput Ind Eng 66(4):1115–1124
20. Raju R, Babukarthik RG, Dhavachelvan P (2013) Hybrid ant colony optimization and cuckoo search algorithm for job scheduling. In: Advances in computing and information technology. Springer Berlin, pp 491–501
21. Binu D, Selvi M, George A (2013) MKF-cuckoo: hybridization of cuckoo search and multiple kernel-based fuzzy C-means algorithm. AASRI Procedia 4:243–249
22. Chen L, Chen CP, Lu M (2011) A multiple-kernel fuzzy C-means algorithm for image segmentation. IEEE Trans Syst Man Cybern Part B Cybern 41(5):1263–1274
23. Huang J, Gao L, Li X (2015) An effective teaching-learning-based cuckoo search algorithm for parameter optimization problems in structure designing and machining processes. Appl Soft Comput 36:349–356
24. Rao RV, Savsani VJ, Vakharia DP (2012) Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 183(1):1–15
25. Singla S, Jarial P, Mittal G. Hybridization of cuckoo search and artificial bee colony optimization for satellite image classification
26. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, vol 200
27. Sheikholeslami R, Zecchin AC, Zheng F, Talatahari S (2015) A hybrid cuckoo–harmony search algorithm for optimal design of water distribution systems. J Hydroinform 174
28. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68
29. Zhang Y, Yu C, Fu X, Liu W, Bi W (2015) Spectrum parameter estimation in Brillouin scattering distributed temperature sensor based on cuckoo search algorithm combined with the improved differential evolution algorithm. Opt Commun 357:15–20
30. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417
31. Price K, Storn R (1997) Differential evolution: a simple evolution strategy for fast optimization. Dr. Dobb's J 22(4):18–24
32. Liu X, Fu M (2015) Cuckoo search algorithm based on frog leaping local search and chaos theory. Appl Math Comput 266:1083–1092
33. Eusuff MM, Lansey KE (2001) Water distribution network design using the shuffled frog leaping algorithm. In World Water Congress, pp 1–8
34. Zheng H, Zhou Y, He S, Ouyang X (2012) A discrete cuckoo search algorithm for solving knapsack problems. Adv Inform Sci Serv Sci 4(18)

35. Burnwal S, Deb S (2013) Scheduling optimization of flexible manufacturing system using cuckoo search-based approach. Int J Adv Manuf Technol 64(5–8):951–959
36. Ouaarab A, Ahiod B, Yang XS (2015) Random-key cuckoo search for the travelling salesman problem. Soft Comput 19(4):1099–1106
37. Dasgupta P, Das S (2015) A discrete inter-species cuckoo search for flowshop scheduling problems. Comput Oper Res 60:111–120
38. Li X, Yin M (2013) A hybrid cuckoo search via Lévy flights for the permutation flow shop scheduling problem. Int J Prod Res 51(16):4732–4754
39. Guo P, Cheng W, Wang Y (2015) Parallel machine scheduling with step-deteriorating jobs and setup times by a hybrid discrete cuckoo search algorithm. Eng Optim 47(11):1564–1585
40. Ljouad T, Amine A, Rziza M (2014) A hybrid mobile object tracker based on the modified Cuckoo search algorithm and the kalman filter. Pattern Recogn 47(11):3597–3613
41. Nguyen TT, Vo DN (2015) The application of one rank cuckoo search algorithm for solving economic load dispatch problems. Appl Soft Comput 37:763–773
42. Garg A, Sahu OP (2015) Cuckoo search based optimal mask generation for noise suppression and enhancement of speech signal. J King Saud Univ Comput Inform Sci 27(3):269–277
43. Ding X, Xu Z, Cheung NJ, Liu X (2015) Parameter estimation of Takagi-Sugeno fuzzy system using heterogeneous cuckoo search algorithm. Neurocomputing 151:1332–1342
44. Abdelaziz AY, Ali ES (2015) Cuckoo Search algorithm based load frequency controller design for nonlinear interconnected power system. Int J Electr Power Energy Syst 73:632–643
45. Teymourian E, Kayvanfar V, Komaki GM, Zandieh M (2016) Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem. Inf Sci 334:354–378
46. Pradeep SA, Manavalan R (2013) Analysis of cuckoo search with genetic algorithm for image compression
47. Al-Khafaji AA, Darus IZM (2014) Controller optimization using cuckoo search algorithm of a flexible single-link manipulator. Optimization (PSO) 9(10):11
48. Chen JF, Do QH, Hsieh HN (2015) Training artificial neural networks by a hybrid PSO-CS algorithm. Algorithms 8(2):292–308
49. Meziane R, Boufala S, Amar H, Amara M Wind farm reliability optimization using cuckoo search algorithm
50. Zaw MM, Mon EE (2013) Web document clustering using cuckoo search clustering algorithm based on levy flight. Int J Innov Appl Stud 4(1):182–188
51. Zineddine M (2015) Vulnerabilities and mitigation techniques toning in the cloud: a cost and vulnerabilities coverage optimization approach using Cuckoo search algorithm with Lévy flights. Comput Secur 48:1–18
52. Sun X, Sun W, Wang J, Zhang Y, Gao Y (2016) Using a Grey–Markov model optimized by Cuckoo search algorithm to forecast the annual foreign tourist arrivals to China. Tour Manag 52:369–379
53. Piechocki J, Ambroziak D, Palkowski A, Redlarski G (2014) Use of modified Cuckoo search algorithm in the design process of integrated power systems for modern and energy self-sufficient farms. Appl Energy 114:901–908
54. Agrawal S, Panda R, Bhuyan S, Panigrahi BK (2013) Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. Swarm Evol Comput 11:16–30
55. El-Maleh AH, Sait SM, Bala A (2015) State assignment for area minimization of sequential circuits based on cuckoo search optimization. Comput Electr Eng 44:13–23
56. Naik MK, Panda R (2016) A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition. Appl Soft Comput 38:661–675
57. Nguyen TT, Truong AV, Phung TA (2016) A novel method based on adaptive cuckoo search for optimal network reconfiguration and distributed generation allocation in distribution network. Int J Electr Power Energy Syst 78:801–815
58. Ilunga-Mbuyamba E, Cruz-Duarte JM, Avina-Cervantes JG, Correa-Cely CR, Lindner D, Chalopin C (2016) Active contours driven by Cuckoo Search strategy for brain tumour images segmentation. Expert Syst Appl 56:59–68
59. Cobos C, Muñoz-Collazos H, Urbano-Muñoz R, Mendoza M, León E, Herrera-Viedma E (2014) Clustering of web search results based on the cuckoo search algorithm and balanced Bayesian information criterion. Inf Sci 281:248–264
60. Bhateja AK, Bhateja A, Chaudhury S, Saxena PK (2015) Cryptanalysis of Vigenere cipher using Cuckoo search. Appl Soft Comput 26:315–324
61. Basu M, Chowdhury A (2013) Cuckoo search algorithm for economic dispatch. Energy 60:99–108
62. Bhandari AK, Singh VK, Kumar A, Singh GK (2014) Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. Expert Syst Appl 41(7):3538–3560
63. Nandy S, Yang XS, Sarkar PP, Das A (2015) Color image segmentation by cuckoo search. Intell Autom Soft Comput 21(4):673–685
64. Dash P, Saikia LC, Sinha N (2015) Comparison of performances of several FACTS devices using Cuckoo search algorithm optimized 2DOF controllers in multi-area AGC. Int J Electr Power Energy Syst 65:316–324
65. Daniel E, Anitha J (2016) Optimum wavelet based masking for the contrast enhancement of medical images using enhanced cuckoo search algorithm. Comput Biol Med 71:149–155
66. Kahramanli H (2012) A modified cuckoo optimization algorithm for engineering optimization. Int J Future Comput Commun 1(2):199
67. Dejam S, Sadeghzadeh M, Mirabedini SJ (2012) Combining cuckoo and tabu algorithms for solving quadratic assignment problems. J Acad Appl Stud 2(12):1–8
68. Glover F, McMillan C (1986) The general employee scheduling problem. An integration of MS and AI. Comput Oper Res 13(5):563–573
69. Naseri KN (2014) A hybrid cuckoo-gravitation algorithm for cost-optimized QFD decision-making problem. J Math Comput Sci 9:342–351
70. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
71. Sarvi M, Parpaei M, Bagheri H, Kojoori MR (2014) Optimal operation and output oscillations reduction of PEMFC by using an intelligent strategy. Int J Electrochem Sci 9:4172–4189
72. Yang XS, Deb S (2010). Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization. In Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Berlin, pp. 101–111
73. Mahmoudi S, Rajabioun R, Lotfi S (2013) Binary cuckoo optimization algorithm. Nature
74. Kazemi E, Dejam S (2014) Discretization of cuckoo optimization algorithm for solving quadratic assignment problems. J World's Electr Eng Technol 2322:5114
75. Mousavirad SJ, Ebrahimpour-Komleh H (2014) Wrapper feature selection using discrete cuckoo optimization algorithm
76. Hadian M, AliAkbari N, Karami M (2015) Using artificial neural network predictive controller optimized with Cuckoo Algorithm for pressure tracking in gas distribution network. J Nat Gas Sci Eng 27:1446–1454

77. Khajeh M, Golzary AR (2014) Synthesis of zinc oxide nanoparticles–chitosan for extraction of methyl orange from water samples: cuckoo optimization algorithm–artificial neural network. Spectrochim Acta Part A Mol Biomol Spectrosc 131:189–194
78. Addeh J, Ebrahimzadeh A, Azarbad M, Ranaee V (2014) Statistical process control using optimized neural networks: a case study. ISA Trans 53(5):1489–1499
79. Shahdi-Pashaki S, Teymourian E, Kayvanfar V, Komaki GM, Sajadi A (2015) Group technology-based model and cuckoo optimization algorithm for resource allocation in cloud computing. IFAC-PapersOnLine 48(3):1140–1145
80. Moghadasian M, Hosseini SP (2014) Binary cuckoo optimization algorithm for feature selection in high-dimensional datasets. In: International conference on innovative engineering technologies (ICIET'2014), pp 18–21
81. Behnasr M, Jazayeri-Rad H (2015) Robust data-driven soft sensor based on iteratively weighted least squares support vector regression optimized by the cuckoo optimization algorithm. J Nat Gas Sci Eng 22:35–41
82. Asadi1, S., & Rafe, V. (2014). Presenting a method for clustering using cuckoo optimization algorithm. Int Adv Eng Technol (IAET) 22
83. Amiri E, Mahmoudi S (2016) Efficient protocol for data clustering by fuzzy Cuckoo optimization algorithm. Appl Soft Comput 41:15–21
84. Mahmoudi S, Lotfi S (2015) Modified cuckoo optimization algorithm (MCOA) to solve graph coloring problem. Appl Soft Comput 33:48–64
85. Moezi SA, Zakeri E, Bazargan-Lari Y, Zare A (2015) 2&3-Dimensional optimization of connecting rod with genetic and modified cuckoo optimization algorithms. Iran J Sci Technol Trans Mech Eng 39(M1):39
86. Balochian S, Ebrahimi E (2013) Parameter optimization via cuckoo optimization algorithm of fuzzy controller for liquid level control. J Eng 2013
87. Berrazouane S, Mohammedi K (2014) Parameter optimization via cuckoo optimization algorithm of fuzzy controller for energy management of a hybrid power system. Energy Convers Manag 78:652–660
88. Moezi SA, Zakeri E, Zare A, Nedaei M (2015) On the application of modified cuckoo optimization algorithm to the crack detection problem of cantilever Euler-Bernoulli beam. Comput Struct 157:42–50
89. Singh U, Rattan M (2014) Design of linear and circular antenna arrays using cuckoo optimization algorithm. Progress Electromagn Res C 46:1–11
90. Ajami A, Mohammadzadeh B, Oskuee MRJ (2014) Utilizing the cuckoo optimization algorithm for selective harmonic elimination strategy in the cascaded multilevel inverter. ECTI Trans Electr Eng Electron Commun 12(1):7–15
91. Rezaei M, Ghanbari M (2015) Optimization of hybrid Pv/Wind/Fc system considering reliability indices using cuckoo search algorithm. Indian J Fundam Appl Life Sci 5(S1):3304–3320
92. Esfandiari A (2014) Cuckoo optimization algorithm in cutting conditions during machining. J Adv Comput Res 5(2):45–57
93. Sarvi M, Parpaei M (2013) Maximum power point tracking of wind energy conversion system using fuzzy-cuckoo optimization algorithm strategy. Int J Smart Electr Eng 2(4):195–200
94. Le Anh TN, Vo DN, Ongsakul W, Vasant P, Ganesan T (2015) Cuckoo optimization algorithm for optimal power flow. In: Proceedings of the 18th Asia pacific symposium on intelligent and evolutionary systems. Springer International Publishing, vol 1, pp 479–493
95. Shokri-Ghaleh H, Alfi A (2014) Optimal synchronization of teleoperation systems via cuckoo optimization algorithm. Nonlinear Dyn 78(4):2359–2376
96. Hosseini-Moghari SM, Morovati R, Moghadas M, Araghinejad S (2015) Optimum operation of reservoir using two evolutionary algorithms: imperialist competitive algorithm (ICA) and cuckoo optimization algorithm (COA). Water Resour Manage 29(10):3749–3769
97. Babaei, H., & Zayandehroodi, H. (2015). Estimation of Induction Motor Efficiency Value by Nature Based Optimization Algorithm. Cumhuriyet Sci J 1(1)
98. Bayati M (2015) Using cuckoo optimization algorithm and imperialist competitive algorithm to solve inverse kinematics problem for numerical control of robotic manipulators. Proceedings of the Institution of Mechanical Engineers, Part I Journal of Systems and Control Engineering, 0959651814568364
99. Akbarzadeh A, Shadkam E (2015) The study of cuckoo optimization algorithm for production planning problem. arXiv preprint arXiv:1508.01310
100. Fayyazi A, Eskandari MJ, Poueinak MM, Nia PM (2014) Solving urban bus terminal location problem using the metaheuristic cuckoo optimization algorithm. Int J Basic Sci Appl Res 3:46–53
101. Mehdinejad M, Mohammadi-Ivatloo B, Dadashzadeh-Bonab R (2016) Energy production cost minimization in a combined heat and power generation systems using cuckoo optimization algorithm. Energy Efficiency 1–16
102. Mellal MA, Williams EJ (2015) Cuckoo optimization algorithm with penalty function for combined heat and power economic dispatch problem. Energy 93:1711–1718
103. Mellal MA, Adjerid S, Williams EJ, Benazzouz D (2012) Optimal replacement policy for obsolete components using cuckoo optimization algorithm based-approach: dependability context. J Sci Ind Res 71(11):715–721
104. Rabiee M, Sajedi H (2013). Job scheduling in grid computing with cuckoo optimization algorithm. Int J Comput Appl 62(16)
105. Namdari F, Samadinasab S, Shojaei N, Bakhshipour M (2015) Optimal coordination of overcurrent and distance relays using cuckoo optimization algorithm. TELKOMNIKA Indones J Electr Eng 16(3):389
106. Maadi M, Javidnia M, Ghasemi M (2016) Two new algorithms of cuckoo and forest optimization to solve single row facility layout problem. J AI Data Min 4(1):35–48
107. Toushmalani R, Esmaeili A, Parsa Z (2014) Comparison result of inversion of gravity data of a fault by particle swarm optimization and cuckoo optimization methods. Res J Pharm Biol Chem Sci 5(1):428
108. Toushmalani R, Saibi H (2015) 3D inversion of gravity data using Cuckoo optimization algorithm
109. Mousavirad SJ, Ebrahimpour-Komleh H (2015) Entropy based optimal multilevel thresholding using cuckoo optimization algorithm. In Innovations in information technology (IIT), 2015 11th international conference on IEEE, pp 302–307
110. Kouhi SZ, Nejati F, Karimpour J (2013) Solving the graph partitioning based on cuckoo optimization algorithm (COA)
111. Saini MK, Narang D (2013) Cuckoo optimization algorithm based image enhancement
112. Anisheh SM, Asemani D (2013) Image compression using improved wavelet shrinkage and artificial neural network. In International conference on image processing and electronics engineering (ICIPEE'2013), pp 111–113
113. Fister I, Yang XS (2015) A short discussion about "Economic optimization design of shell-and-tube heat exchangers by a cuckoo-search-algorithm". Appl Therm Eng 76:535–537

114. Chen JF, Hsieh HN, Do QH (2014) Predicting student academic performance: a comparison of two meta-heuristic algorithms inspired by cuckoo birds for training neural networks. Algorithms 7(4):538–553

115. Mladenović N, Hansen P (1997) Variable neighborhood search. Comput Oper Res 24(11):1097–1100