Cuckoo Search Optimization

June 10, 2023

1 Introduction

Nature-inspired metaheuristic algorithms have been used in a wide range of optimization problems. These algorithms imitate the best features in nature especially the selection of the fittest.

Intensification and diversification are the two important characteristics of metaheuristics.

- Intensification intends to search around the current best solutions and select the best solutions.
- Diversification makes sure that the algorithm can explore the search space more efficiently by randomization.

Cuckoo search is a metaheuristic algorithm based on the breeding behavior of certain species of cuckoos.

2 Cuckoo Breeding Behavior

Quite a number of cuckoo species lay their eggs in the nests of other host birds. This is called brood parasitism.

If a host bird discovers the eggs are not its own, it will either throw these alien eggs away or simply abandon its nest and build a nest elsewhere.

Some species have evolved in such a way that female parasitic cuckoos are often very specialized in the mimicry of color and pattern of the eggs of a few chosen host species.

3 Algorithm

We use the following three idealized rules in cuckoo search:

1. Each cuckoo lays one egg at a time and dumps its egg in a randomly chosen nest.
2. The best nests with 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 is discovered by the host bird with a probability $pa \in [0, 1]$

We assume that if the host bird discovers a cuckoo egg it will abandon the nest and replace it with a completely new nest.

Initially, we have n host bird nests. We will use the following representation for the cuckoo search, each egg in a nest represents a solution and a cuckoo represents a new solution, the aim is to use the new and potentially better solutions to replace a not-so-good solution. We generate the new solutions using the levy's flight as shown below:

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha \oplus \textbf{Lévy}(\lambda),$$

Figure 1:

Here α is the step size, $x_i^{(t)}$ is the current solution, $x_i^{(t+1)}$ is the new solution, $\bigoplus$ means entry wise multiplication.

Some of the new solutions should be generated by Levy's walk around the best solution obtained so far(intensification) and a fraction of the new solutions should be generated by far-field randomization and whose locations should be far enough from the current best solution(diversification). This will make sure that system will not be trapped in a local optimum.

The Pseudo code below shows the basic steps of the Cuckoo search:

begin
  Objective function $f(\mathbf{x})$, $\mathbf{x} = (x_1, ..., x_d)^T$
  Generate initial population of
    n *host nests* $\mathbf{x}_i$ $(i = 1, 2, ..., n)$
  while *(t <MaxGeneration) or (stop criterion)*
    Get a cuckoo randomly by Lévy flights
      evaluate its quality/fitness F_i
    Choose a nest among n *(say,* j*) randomly*
    if $(F_i > F_j)$,
      replace j *by the new solution;*
    end
    A fraction (p_a) *of worse nests*
      are abandoned and new ones are built;
    Keep the best solutions
      (or nests with quality solutions);
    Rank the solutions and find the current best
  end while
  Postprocess results and visualization
end

Figure 2: Pseudo code of the Cuckoo Search (CS

4 Comparison of CS with PSO and GA

The tables below compare the Cuckoo search with Genetic Algorithms(GA) and Particle Swarm Optimization(PSO)

Functions/Algorithms	GA	CS
Multiple peaks	$52124 \pm 3277(98\%)$	$927 \pm 105(100\%)$
Michalewicz's (d=16)	$89325 \pm 7914(95\%)$	$3221 \pm 519(100\%)$
Rosenbrock's (d=16)	$55723 \pm 8901(90\%)$	$5923 \pm 1937(100\%)$
De Jong's (d=256)	$25412 \pm 1237(100\%)$	$4971 \pm 754(100\%)$
Schwefel's (d=128)	$227329 \pm 7572(95\%)$	$8829 \pm 625(100\%)$
Ackley's (d=128)	$32720 \pm 3327(90\%)$	$4936 \pm 903(100\%)$
Rastrigin's	$110523 \pm 5199(77\%)$	$10354 \pm 3755(100\%)$
Easom's	$19239 \pm 3307(92\%)$	$6751 \pm 1902(100\%)$
Griewank's	$70925 \pm 7652(90\%)$	$10912 \pm 4050(100\%)$
Shubert's (18 minima)	$54077 \pm 4997(89\%)$	$9770 \pm 3592(100\%)$

Figure 3: Comparision of CS with GA

Functions/Algorithms	PSO	CS
Multiple peaks	$3719 \pm 205(97\%)$	$927 \pm 105(100\%)$
Michalewicz's (d=16)	$6922 \pm 537(98\%)$	$3221 \pm 519(100\%)$
Rosenbrock's (d=16)	$32756 \pm 5325(98\%)$	$5923 \pm 1937(100\%)$
De Jong's (d=256)	$17040 \pm 1123(100\%)$	$4971 \pm 754(100\%)$
Schwefel's (d=128)	$14522 \pm 1275(97\%)$	$8829 \pm 625(100\%)$
Ackley's (d=128)	$23407 \pm 4325(92\%)$	$4936 \pm 903(100\%)$
Rastrigin's	$79491 \pm 3715(90\%)$	$10354 \pm 3755(100\%)$
Easom's	$17273 \pm 2929(90\%)$	$6751 \pm 1902(100\%)$
Griewank's	$55970 \pm 4223(92\%)$	$10912 \pm 4050(100\%)$
Shubert's (18 minima)	$23992 \pm 3755(92\%)$	$9770 \pm 3592(100\%)$

Figure 4: Comparision of CS with PSO

The numbers in the tables are as follows: $927 \pm 105(100\%)$
927 – the average number of function evaluations
105 – standard deviation
100% – success rate of finding the global optima

5 Conclusion

Simulations and comparison show that CS is superior to these existing algorithms for multimodal objective functions. The primary reasons for this are:

- A fine balance of randomization and intensification.
- Less number of control parameters(only n and pa)