A

Cuckoo Search for Optimization and Computational Intelligence

Xin-She Yang
School of Science and Technology, Middlesex University, UK

Suash Deb
Cambridge Institute of Technology, India

INTRODUCTION

In almost all applications in engineering and industry, we are always trying to optimize something -- whether to minimize the cost and energy consumption, or to maximize the profit, output, performance and efficiency (Yang, 2010b; Yang & Koziel, 2011). The optimal use of available resources of any sort requires a paradigm shift in scientific thinking, this is because most real-world applications have far more complicated factors and parameters to affect how the system behaves.

Optimization algorithms are the tools and techniques of achieving the optimality of the problem of interest. This search for optimality is complicated further by the fact that uncertainty almost always presents in the real-world systems. Therefore, we seek not only the optimal design but also robust design in engineering and industry. Optimal design solutions, which are not robust enough, are not practical in reality. Suboptimal solutions or good robust solutions are often the choice in such cases because they are more robust and less sensitive to the uncertainty in the material properties in the real systems.

Optimization problems can be formulated in many ways. For example, the commonly used method of least-squares is a special case of maximum-likelihood formulations. By far the most widely formulation is to write a nonlinear optimization problem as

$$\text{minimize} \quad f_i(x), \quad (i = 1, 2, ..., M), \tag{1}$$

subject to the constraints

$$h_j(x) = 0, \quad (j = 1, 2, ..., J), \tag{2}$$

$$g_k(x) \leq 0, \quad (k = 1, 2, ..., K), \tag{3}$$

where f_i, h_j and g_k are in general nonlinear functions. Here the design vector

$$x = (x_1, x_2, ..., x_d)$$

can be continuous, discrete or mixed in a d-dimensional space. The functions f_i are called objectives or cost functions, and when $M > 1$, the optimization problem is multiobjective or multicriteria (Yang, 2008, 2010a; Yang, 2010b). It is possible to combine different objectives into a single objective, though multiobjective optimization can give far more information and options to the decision-makers with more insight into the problem. It is worth pointing out that here we write the problem as a minimization problem, it can also be written as a maximization problem by simply replacing $f_i(x)$ by $-f_i(x)$.

When all functions are nonlinear, we are dealing with nonlinear constrained problems. In some special cases when f_i, h_j, g_k are linear, the problem becomes linear, and we can use the widely used linear programming techniques such as the simplex method. When some design variables can only take discrete values (often integers), while other variables are real continuous, the problem is of mixed type, which is often difficult to solve, especially for large-scale optimization problems. A very special class of optimization is the convex optimization, which has guaranteed global optimality. Any optimal solution is also the global optimum, and most importantly, there are efficient algorithms of polynomial time to solve such problems.

DOI: 10.4018/978-1-4666-5888-2.ch014

However, nonlinear problems are usually more challenging to solve. For NP-hard problems such as the well-known travelling salesman problem (TSP), there is no efficient algorithm at all. In this case, we have to use specific methods, often heuristic methods by trial and error. There is a major trend in optimization and computational intelligence in recent years to use nature-inspired metaheuristic algorithms to solve these tough NP problems. For example, cuckoo search has been successfully used to solve a class of TSP (Ouaarab et al., 2013).

BACKGROUND

From the optimization point of view, an efficient optimizer (i.e., algorithm) is very important to ensure the optimal solutions are reachable. The essence of an optimizer is a search or optimization algorithm implemented correctly so as to carry out the desired search (though not necessarily efficiently). It can be integrated and linked with other modelling components. There are many optimization algorithms in the literature and no single algorithm is suitable for all problems, as dictated by the No Free Lunch Theorems (Wolpert & Macready, 1997).

Algorithms can be classified as deterministic or stochastic. If an algorithm works in a mechanically deterministic manner without any random nature, it is called deterministic. For such a deterministic algorithm, it will reach the same final solution if it starts with the same initial point. Hill-climbing and downhill simplex methods are good examples of deterministic algorithms. On the other hand, if there is some randomness in the algorithm, the algorithm will usually reach a different point every time the algorithm is executed, even though it starts with the same initial point. Genetic algorithms, firefly algorithm and hill-climbing with a random restart are good examples of stochastic algorithms. In the current literature, stochastic algorithms are often called metaheuristic algorithms.

Analyzing stochastic algorithms in more detail, we can single out the type of randomness that a particular algorithm is employing. For example, the well-known hill-climbing with random restart is a good example. This simple strategy is both efficient in most cases and easy to implement in practice. A more elaborate way to introduce randomness to an algorithm is to use randomness inside different components of an algorithm, and in this case, we often call such an algorithm heuristic, or more often metaheuristic (Yang, 2008; Yang & Koziel, 2011; Yang, 2010b). A very good example is the popular genetic algorithms which use randomness for crossover and mutation in terms of a crossover probability and a mutation rate (Holland, 1975). Here, heuristic means to search by trial and error, while metaheuristic is a higher level of heuristics. However, modern literature tends to refer all new stochastic algorithms as metaheuristic. Here, we will use metaheuristic to mean either. It is worth pointing out that metaheuristic algorithms form a hot research topics and new algorithms appear almost yearly (Yang, 2008; Yang, 2010b).

Metaheuristic algorithms are often nature-inspired, and they are now among the most widely used algorithms for optimization. They have many advantages over conventional optimization algorithms. There are a few recent books and review articles which are solely dedicated to metaheuristic algorithms (Blum & Roli, 2003; Yang, 2008; Yang, 2010a; Yang, 2010b). Metaheuristic algorithms are very diverse, including ant and bee algorithms, bat algorithm (Yang, 2010c; Yang & Gandomi 2012), particle swarm optimization (Kennedy & Eberhart, 1995), firefly algorithm (Yang, 2008, 2009), cuckoo search (Yang & Deb, 2009), flower algorithm (Yang et al., 2013), and others (Koziel & Yang, 2011; Sayadi et al., 2010; Parpinelli & Lopes, 2011; Gandomi et al., 2011). Here we will introduce cuckoo search in great detail.

CUCKOO SEARCH

Cuckoo search (CS) is one of the latest nature-inspired metaheuristic algorithms, developed in 2009 by Xin-She Yang and Suash Deb (Yang & Deb, 2009). CS was based on the brood parasitism of some cuckoo species. In addition, this algorithm is enhanced by the so-called Lévy flights (Pavlyukevich, 2007), rather than by simple isotropic random walks. Recent studies show that CS is potentially far more efficient than PSO and genetic algorithms (Yang & Deb, 2010).

Cuckoo are fascinating birds, not only because of the beautiful sounds they can make, but also because of their aggressive reproduction strategy. Some species such as the *ani* and *Guira* cuckoos lay their eggs