Computers and Structures 169 (2016) 1–12

Contents lists available at ScienceDirect

Computers and Structures

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

A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm

Alireza Askarzadeh *

Department of Energy Management and Optimization, Institute of Science and High Technology and Environmental Sciences, Graduate University of Advanced Technology, Kerman, Iran

ARTICLE INFO

Article history:
Received 27 September 2015
Accepted 3 March 2016

Keywords:
Metaheuristic optimization
Crow search algorithm
Constrained engineering optimization

ABSTRACT

This paper proposes a novel metaheuristic optimizer, named crow search algorithm (CSA), based on the intelligent behavior of crows. CSA is a population-based technique which works based on this idea that crows store their excess food in hiding places and retrieve it when the food is needed. CSA is applied to optimize six constrained engineering design problems which have different natures of objective functions, constraints and decision variables. The results obtained by CSA are compared with the results of various algorithms. Simulation results reveal that using CSA may lead to finding promising results compared to the other algorithms.

1. Introduction

Engineering design is defined as a decision making process to build products that satisfy specified needs. Most often, engineering design problems include complicated objective functions with a large number of decision variables. The feasible solutions are the set of all designs characterized by all possible values of the design parameters (decision variables). An optimization technique tries to find the optimal solution from all available feasible solutions.

Conventional search methods have long been applied to solve engineering design problems. Although these methods find promising results in many real problems, they may fail in more complex design problems. In real design problems, the number of decision variables can be very large and their effect on the objective function can be very complicated. The objective function may have many local optima, whereas the designer is interested in the global optimum. Such problems cannot be handled by conventional methods that only find local optima. In these cases, efficient optimization methods are needed.

Metaheuristic algorithms have shown promising performance for solving most real-world optimization problems that are extremely nonlinear and multimodal. All metaheuristic algorithms use a certain tradeoff of randomization and local search [1]. These algorithms can find good solutions for difficult optimization problems, but there is no guarantee that optimal solutions can be reached. It is hoped that these algorithms work most of the time, but not all the time. Metaheuristic algorithms could be suitable for global optimization [2]. Based on Glover's convention, all the modern nature-inspired methods are called metaheuristics [3].

Current trend is to utilize nature-inspired metaheuristic algorithms to tackle difficult problems and it has been shown that metaheuristics are surprisingly very efficient [1,2]. For this reason, the literature of metaheuristics has expanded tremendously in the last two decades [4,5]. Some of the well-known metaheuristic algorithms are as follows: genetic algorithm (GA) based on natural selection [6], particle swarm optimization (PSO) based on social behavior of bird flocking and fish schooling [7], harmony search (HS) based on music improvisation process [8], cuckoo search algorithm based on the brood parasitism of some cuckoo species [9], bat algorithm (BA) based on echolocation behavior of microbats [10], group search optimizer (GSO) based on animal searching behavior [11], firefly algorithm (FA) based on the flashing light patterns of tropic fireflies [12], etc. To date, researchers have only used a very limited characteristics inspired by nature and there is room for development of more algorithms. One of the main motivations of this paper is to develop a user-friendly (simple concept and easy implementation) metaheuristic technique by which we may obtain promising results when solving optimization problems.

Crows are widely distributed genus of birds which are now considered to be among the world's most intelligent animals [13,14]. As a group, crows show remarkable examples of intelligence and often score very highly on intelligence tests. They can memorize faces, use tools, communicate in sophisticated ways and hide and retrieve food across seasons [13,15].

* Tel./fax: +98 342 6233176.
E-mail addresses: a.askarzadeh@kgut.ac.ir, askarzadeh_a@yahoo.com

http://dx.doi.org/10.1016/j.compstruc.2016.03.001
0045-7949/

In a crow flock, there is a behavior which has many similarities with an optimization process. According to this behavior, crows hide their excess food in certain positions (hiding places) of the environment and retrieve the stored food when it is needed. Crows are greedy birds since they follow each other to obtain better food sources. Finding food source hidden by a crow is not an easy work since if a crow finds another one is following it, the crow tries to fool that crow by going to another position of the environment. From optimization point of view, the crows are searchers, the environment is search space, each position of the environment is corresponding to a feasible solution, the quality of food source is objective (fitness) function and the best food source of the environment is the global solution of the problem. Based on these similarities, CSA attempts to simulate the intelligent behavior of the crows to find the solution of optimization problems.

2. Crow search algorithm

Crows (crow family or corvids) are considered the most intelligent birds. They contain the largest brain relative to their body size. Based on a brain-to-body ratio, their brain is slightly lower than a human brain. Evidences of the cleverness of crows are plentiful. They have demonstrated self-awareness in mirror tests and have tool-making ability. Crows can remember faces and warn each other when an unfriendly one approaches. Moreover, they can use tools, communicate in sophisticated ways and recall their food's hiding place up to several months later [13–16].

Crows have been known to watch other birds, observe where the other birds hide their food, and steal it once the owner leaves. If a crow has committed thievery, it will take extra precautions such as moving hiding places to avoid being a future victim. In fact, they use their own experience of having been a thief to predict the behavior of a pilferer, and can determine the safest course to protect their caches from being pilfered [17].

In this paper, based on the above-mentioned intelligent behaviors, a population-based metaheuristic algorithm, CSA, is developed. The principles of CSA are listed as follows:

- Crows live in the form of flock.
- Crows memorize the position of their hiding places.
- Crows follow each other to do thievery.
- Crows protect their caches from being pilfered by a probability.

It is assumed that there is a d-dimensional environment including a number of crows. The number of crows (flock size) is N and the position of crow i at time (iteration) iter in the search space is specified by a vector $x^{i,\mathrm{iter}}$ $(i = 1, 2, \ldots, N;\ \mathrm{iter} = 1, 2, \ldots, \mathrm{iter}_{\max})$ where $x^{i,\mathrm{iter}} = \left[x_1^{i,\mathrm{iter}},\ x_2^{i,\mathrm{iter}},\ \ldots,\ x_d^{i,\mathrm{iter}}\right]$ and $\mathrm{iter}_{\max}$ is the maximum number of iterations. Each crow has a memory in which the position of its hiding place is memorized. At iteration iter, the position of hiding place of crow i is shown by $m^{i,\mathrm{iter}}$. This is the best position that crow i has obtained so far. Indeed, in memory of each crow the position of its best experience has been memorized. Crows move in the environment and search for better food sources (hiding places).

Assume that at iteration iter, crow j wants to visit its hiding place, $m^{j,\mathrm{iter}}$. At this iteration, crow i decides to follow crow j to approach to the hiding place of crow j. In this case, two states may happen:

State 1: Crow j does not know that crow i is following it. As a result, crow i will approach to the hiding place of crow j. In this case, the new position of crow i is obtained as follows:

$$x^{i,\mathrm{iter}+1} = x^{i,\mathrm{iter}} + r_i \times fl^{i,\mathrm{iter}} \times \left(m^{j,\mathrm{iter}} - x^{i,\mathrm{iter}}\right) \tag{1}$$

where r_i is a random number with uniform distribution between 0 and 1 and $fl^{i,\mathrm{iter}}$ denotes the flight length of crow i at iteration iter.

Fig. 1 shows the schematic of this state and the effect of fl on the search capability. Small values of fl leads to local search (at the vicinity of $x^{i,\mathrm{iter}}$) and large values results in global search (far from $x^{i,\mathrm{iter}}$). As Fig. 1(a) shows, if the value of fl is selected less than 1, the next position of crow i is on the dash line between $x^{i,\mathrm{iter}}$ and $m^{j,\mathrm{iter}}$. As Fig. 1(b) indicates, if the value of fl is selected more than 1, the next position of crow i is on the dash line which may exceed $m^{j,\mathrm{iter}}$.

State 2: Crow j knows that crow i is following it. As a result, in order to protect its cache from being pilfered, crow j will fool crow i by going to another position of the search space.

Totally, states 1 and 2 can be expressed as follows:

$$x^{i,\mathrm{iter}+1} = \begin{cases} x^{i,\mathrm{iter}} + r_i \times fl^{i,\mathrm{iter}} \times \left(m^{j,\mathrm{iter}} - x^{i,\mathrm{iter}}\right) & r_j \geqslant AP^{j,\mathrm{iter}} \\ \text{a random position} & \text{otherwise} \end{cases} \tag{2}$$

where r_j is a random number with uniform distribution between 0 and 1 and $AP^{j,\mathrm{iter}}$ denotes the awareness probability of crow j at iteration iter.

Metaheuristic algorithms should provide a good balance between diversification and intensification [2]. In CSA, intensification and diversification are mainly controlled by the parameter of awareness probability (AP). By decrease of the awareness probability value, CSA tends to conduct the search on a local region where a current good solution is found in this region. As a result, using small values of AP, increases intensification. On the other hand, by increase of the awareness probability value, the probability of searching the vicinity of current good solutions decreases and CSA tends to explore the search space on a global scale (randomization). As a result, use of large values of AP increases diversification.

3. CSA implementation for optimization

Pseudo code of CSA is shown in Fig. 2. The step-wise procedure for the implementation of CSA is given in this section.

Step 1: Initialize problem and adjustable parameters

The optimization problem, decision variables and constraints are defined. Then, the adjustable parameters of CSA (flock size (N), maximum number of iterations ($\mathrm{iter}_{\max}$), flight length (fl) and awareness probability (AP)) are valued.

Step 2: Initialize position and memory of crows

N crows are randomly positioned in a d-dimensional search space as the members of the flock. Each crow denotes a feasible solution of the problem and d is the number of decision variables.

$$\mathrm{Crows} = \begin{bmatrix} x_1^1 & x_2^1 & \ldots & x_d^1 \\ x_1^2 & x_2^2 & \ldots & x_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^N & x_2^N & \ldots & x_d^N \end{bmatrix} \tag{3}$$

The memory of each crow is initialized. Since at the initial iteration, the crows have no experiences, it is assumed that they have hidden their foods at their initial positions.