Measurement 68 (2015) 225–235

Contents lists available at ScienceDirect

Measurement

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

Investigation of cricket behaviours as evolutionary computation for system design optimization problems

Murat Canayaz [a,*], Ali Karcı [b]

[a] Computer Science Research and Application Center, Yuzuncu Yil University Campus, 65080 Van, Turkey
[b] Department of Computer Engineering, Faculty of Engineering, Inonu University Campus, 44280 Malatya, Turkey

ARTICLE INFO

Article history:
Received 19 December 2014
Received in revised form 18 February 2015
Accepted 23 February 2015
Available online 6 March 2015

Keywords:
Nature-inspired algorithm
Meta-heuristic algorithm
Optimization
Cricket
Design problems

ABSTRACT

In this study, the behaviours of an insect species called cricket were investigated and tried to develop a new meta-heuristic algorithm approach that may be used in solving optimization problems by modelling these behaviours. These insect species make a sound by flapping their wings and attract the other crickets around them. While creating this algorithm, the physics laws related to propagation of sound as well as the crickets ability to predict the temperature with the number of flaps were also considered. The approach performance was tried to be shown by applying the developed approach at the end of the study to both numeric problems and cantilever stepped and welded beam that are of system design optimization problems.

1. Introduction

Optimization is known as the studies for obtaining optimum results under certain conditions. For example, in a commercial company, it is aimed at keeping costs and expenses minimum while maximizing the profit. In other words, they are required to be at optimum values. Optimization comes into play in solution of such problems and helps achieving optimum results. In addition, in the mathematical sense, optimization can be defined as generating solutions for a problem by sending values to a function from suitable ranges of value, in order to minimize or maximize the function. This suggests that optimization can also be used in solution of problems that are difficult to solve using traditional mathematical methods. In obtaining these optimum results, the algorithms based on heuristics, which is one of optimization techniques, have significant contribution. These algorithms are used to solve optimization problems and thus it is aimed at achieving optimum or near optimum results as result of the operations. Heuristic algorithms are algorithms aimed at finding a good result without necessitating proof. We frequently encounter these algorithms in daily life. For example, we make heuristic choices about which way to use while going to work. In addition to heuristic algorithms, there is a mechanism that works on these algorithms and decides the best method for solving a problem amongst such algorithms. These decision mechanisms are also known as meta-heuristic algorithms in the literature.

The known meta-heuristic algorithms include:

- Simulated annealing [1].
- Tabu search [2].
- Genetic algorithms [3].
- Ant colony optimization [4].
- Artificial bee colony [5].
- Particle swarm optimization [6].

There are certain common items that must be present in these algorithms. These items include:

* Corresponding author.
E-mail addresses: mcanayaz@yyu.edu.tr (M. Canayaz), ali.karci@inonu.edu.tr (A. Karcı).

http://dx.doi.org/10.1016/j.measurement.2015.02.052
0263-2241/

- Object function $f(x)$.
- Design variable x.
- Constraints $g(x)$.
- E.g.: $g(x) < 0$.

Size to be optimized (maximum or minimum) is defined, in order to find object function and optimum value, and the parameters they take with changed values are defined as design variables. Certain constraint functions must be found for controlling value taking parameters in specified ranges. These functions are called as constraints in the literature [7,8].

In recent years, algorithms such as firefly and bat developed by Yang are also taking place amongst meta-heuristic algorithms and contribute to the solution of optimization problems. As suggested by their name, these algorithms are inspired by the behaviours of living creatures in the nature. In other words, in the literature, they take place as algorithms that are realized through modelling of creatures' behaviours in nature, while hunting or moving. This study presents a new algorithm that is revealed through inspiration from an insect species known as cricket, which attracts other crickets with its sound. While creating the algorithm, the natural physics laws in connection with the propagation of sound in nature were tried to follow. The algorithm also includes cricket characteristics such as the ability to estimate temperature from the number of flaps, which is suggested by Dolbear [9]. Furthermore, aforementioned in Yang's bat algorithm, during the creation phase of the algorithm, it has been sought, if it is possible to take the good aspects of other algorithms and create a new algorithm. For this purpose, bat, firefly and PSO [6,10] algorithms were used and the draft version was presented [11]. The suggested algorithm, like these algorithms, is based on population and tries to offer new candidate solutions that converge into optimum result. In this study, the bat and firefly algorithms will be introduced respectively. Later on, the information about behaviours of the crickets, which are the source of inspiration for this study will be given, and the developed algorithm be introduced. The aim of this study is to use the behaviours of cricket as an evolutionary computation technique approach and demonstrate to the researchers that it can be applied to engineering problems as well as the other areas that require optimization. At consequence, it was attempted to show the performance of the algorithm applied on numeric problems and system design problems. Numerical results were compared with the results obtained from the problems derived from the Bat and the ABC algorithm.

2. Bat algorithm

This algorithm was developed through inspiration from bats' ability to find their direction and preys in nature [12]. As known, bats find their direction through reflection of high sound waves they propagate from surrounding objects and find their prey using this method. Therefore, the sound signals, they generate vary depending on their hunting strategy and location of the preys around. This algorithm contains basic principles that are accepted by using these characteristics of bats.

These principles;

- Any bats use echolocation to detect the distance. Thus, they know the difference between preys and obstacles of around them.
- Bats fly randomly with speed v_i at location x_i to seek for prey. Furthermore, they fly a constant frequency $f_{\min}$, varying wavelength λ and loudness A_0. They can arrange the ratio of pulse emission, r must be in the range $[0, 1]$.
- Although the sound of intensity changes by different ways, in this algorithm it changes the range of A_0 and $A_{\min}$.

The pseudo code pertaining to this algorithm is shown in Fig. 1 [12]. There are many studies in optimization and other areas that use this algorithm [13,14].

3. Firefly algorithm

The name of the algorithm comes from firefly, which was modelled in its development. The firefly that emits more light attracts the other fireflies. In other words, it can be said that there is a direct proportion between the light emitted by a firefly and its level of attractiveness [15,16]. In this algorithm, brightness of fireflies is considered as the object function of the heuristic algorithm. As of its structure, the firefly algorithm can be classified under particle swarm optimization algorithms. This algorithm has basic principles of its own. Fig. 2 shows the pseudo-code pertaining to this algorithm [15]. This algorithm continues being used in many areas including optimization [17–19].

- All fireflies are unisexual. All fireflies could attract another one regardless of gender.

Bat Algorithm Pseudo Code

```
Objective function f(x), x=(x1,....,xd)^T
Initialize the bat population xi(i=1,2,...n) ve vi
Define pulse frequency fi at xi
Initialize pulse rates ri and loudness Ai
while(t<Max number of iterations)
Generate new solutions by adjusting frequency, and
updating velocities and locations/solutions
if(rand>ri)
Select a solution among the best solutions
Generate a local solution around the selected best solution
end if
Generate a new solution by flying randomly
if(rand<Ai & f(xi)<f(x*))
Accept the new solutions
Increase ri and reduce Ai
end if
Rank the bats and find the current best x*
end while
Postprocess results and visualization
```

Fig. 1. Pseudo code of the bat algorithm.