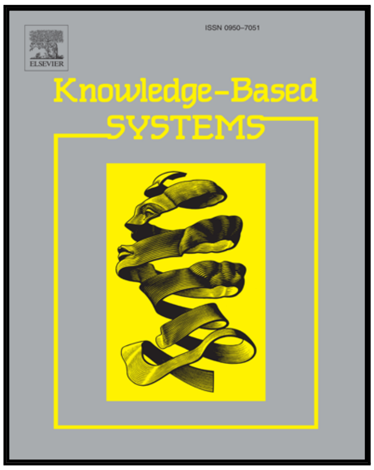
Accepted Manuscript



Emperor Penguin Optimizer: A Bio-inspired Algorithm for Engineering Problems

Gaurav Dhiman, Vijay Kumar

PII: S0950-7051(18)30296-X
DOI: 10.1016/j.knosys.2018.06.001
Reference: KNOSYS 4365

To appear in: *Knowledge-Based Systems*

Received date: 16 October 2017
Revised date: 30 May 2018
Accepted date: 1 June 2018

Emperor Penguin Optimizer: A Bio-inspired Algorithm for Engineering Problems

Gaurav Dhiman[a*], Vijay Kumar[b]

[a*,b] *Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab, INDIA*

[*] ***Corresponding author email:*** *gdhiman0001@gmail.com, gaurav.dhiman@thapar.edu*

Abstract

This paper proposes a novel optimization algorithm, called Emperor Penguin Optimizer (EPO), which mimics the huddling behavior of emperor penguins (*Aptenodytes forsteri*). The main steps of EPO are to generate the huddle boundary, compute temperature around the huddle, calculate the distance, and find the effective mover. These steps are mathematically modeled and implemented on 44 well-known benchmark test functions. It is compared with eight state-of-the-art optimization algorithms. The paper also considers for solving six real-life constrained and one unconstrained engineering design problems. The convergence and computational complexity are also analyzed to ensure the applicability of proposed algorithm.The experimental results show that the proposed algorithm is able to provide better results as compared to the other well-known metaheuristic algorithms.

Keywords: Optimization techniques; Metaheuristics; Constrained optimization; Unconstrained optimization; Benchmark test functions.

1. Introduction

During the last few decades, various algorithms have been proposed to solve the variety of real-life engineering optimization problems [1, 2]. These optimization problems are very complex in nature because they have more than one local optimum solution. These problems are divided into various categories whether they are constrained or unconstrained, discrete or continuous, static or dynamic, single or multi-objective.

In order to increase the efficiency and accuracy of these problems, researchers have encouraged to rely on metaheuristic optimization algorithms [3]. Metaheuristics become more popular in various field because they do not require gradient information, easy to implement, and bypass the local optima problem.

Generally, metaheuristics are divided into two categories such as single-solution and multiple-solution based problems.

In single-solution based algorithms the searching process starts with one candidate solution whereas in multiple-solution based algorithm the optimization performs using a set of solutions (i.e., population). Multiple-solution or population based metaheuristics have advantages over single-solution based metaheuristics. These are as follows:

- The searching process starts with random generated population i.e, a set of multiple solutions.
- The multiple solutions can share the information between each other around the search space and avoid local optimal solutions.
- The exploration capability of multiple-solution or population based metaheuristics have better than the single-solution based techniques.

Multiple-solution based metaheuristic algorithms are further classified into three categories such as evolutionary-based, physics-based, and swarm-based methods. The first category is generic population based metaheuristic which is inspired from biological evolution i.e., mutation, recombination, and selection. These methods do not make any assumptions about the basic fitness landscape.

The second category is physics-based algorithms in which each search agent can communicate and move throughout the search space according to some physics rules such as gravitational force, electromagnetic force, inertia force, and many more.

The last category is swarm-based algorithms which are inspired by the collective behavior of social creatures. This collective intelligence is based on the interaction of swarm with each other. Swarm-based algorithms are easier to implement than the evolutionary-based algorithms due to include the less number of operators (i.e., selection, crossover, mutation). Apart from this, there are some advantages of swarm-based algorithms which are as follows:

- Swarm-based algorithms can maintain the information about the search space during course of iterations whereas evolutionary-based algorithms can eliminate the information of the previous generations.
- Swarm-based algorithms have few input parameters as compared to the evolutionary techniques.
- Swarm-based algorithms utilize less memory space for saving the best optimal solutions.

The key phases of metaheuristic algorithms are exploration and exploitation [4, 5]. The exploration phase ensures that algorithm investigates the different promising regions in a given search space whereas exploitation ensures the searching of optimal solutions around the promising regions which is obtained in the exploration phase [6]. However, it is difficult to balance between these phases due to its stochastic nature. Therefore, the fine tuning of these two phases is required to achieve the near optimal solutions for a given optimization problem.

This is the one fact which can motivates us to develop a novel metaheuristic algorithm for solving real-life optimization problems. Another fact of our motivation is the set of given problem in which the performance of one optimizer does not guarantee to solve other optimization problem with different nature [7].