Knowledge-Based Systems 89 (2015) 228–249

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm

Seyedali Mirjalili *

School of Information and Communication Technology, Griffith University, Nathan Campus, Brisbane, QLD 4111, Australia
Queensland Institute of Business and Technology, Mt Gravatt, Brisbane, QLD 4122, Australia

ARTICLE INFO

Article history:
Received 23 May 2015
Received in revised form 25 June 2015
Accepted 11 July 2015
Available online 18 July 2015

Keywords:
Optimization
Stochastic optimization
Constrained optimization
Meta-heuristic
Population-based algorithm

ABSTRACT

In this paper a novel nature-inspired optimization paradigm is proposed called Moth-Flame Optimization (MFO) algorithm. The main inspiration of this optimizer is the navigation method of moths in nature called transverse orientation. Moths fly in night by maintaining a fixed angle with respect to the moon, a very effective mechanism for travelling in a straight line for long distances. However, these fancy insects are trapped in a useless/deadly spiral path around artificial lights. This paper mathematically models this behaviour to perform optimization. The MFO algorithm is compared with other well-known nature-inspired algorithms on 29 benchmark and 7 real engineering problems. The statistical results on the benchmark functions show that this algorithm is able to provide very promising and competitive results. Additionally, the results of the real problems demonstrate the merits of this algorithm in solving challenging problems with constrained and unknown search spaces. The paper also considers the application of the proposed algorithm in the field of marine propeller design to further investigate its effectiveness in practice. Note that the source codes of the MFO algorithm are publicly available at http://www.alimirjalili.com/MFO.html.

1. Introduction

Optimization refers to the process of finding the best possible solution(s) for a particular problem. As the complexity of problems increases, over the last few decades, the need for new optimization techniques becomes evident more than before. Mathematical optimization techniques used to be the only tools for optimizing problems before the proposal of heuristic optimization techniques. Mathematical optimization methods are mostly deterministic that suffer from one major problem: local optima entrapment. Some of them such as gradient-based algorithms require derivation of the search space as well. This makes them highly inefficient in solving real problems.

The so-called Genetic Algorithm (GA) [1], which is undoubtedly the most popular stochastic optimization algorithm, was proposed to alleviate the aforementioned drawbacks of the deterministic algorithms. The key success of GA algorithm mostly relies on the stochastic components of this algorithm. The selection, re-production, and mutation have all stochastic behaviours that assist GA to avoid local optima much more efficient than mathematical optimization algorithms. Since the probability of selection and re-production of best individuals is higher than worst individuals, the overall average fitness of population is improved over the course of generations. These two simple concepts are the key reasons of the success of GA in solving optimization problems. Another fact about this algorithm is that there is no need to have gradient information of the problems since GA only evaluates the individuals based on the fitness. This makes this algorithm highly suitable for solving real problems with unknown search spaces. Nowadays, the application of the GA algorithm can be found in a wide range of fields.

The years after the proposal of the GA were accompanied by the highest attention to such algorithms, which resulted in the proposal of the well-known algorithms like PSO [2] algorithm, Ant Colony Optimization (ACO) [3], Differential Evolution (DE) [4], Evolutionary Strategy (ES) [5], and Evolutionary Programming (EP) [6,7]. The application of these algorithms can be found in different branches of science and industry as well. Despite the merits of these optimizers, there is a fundamental question here as if there is any optimizer for solving all optimization problems. According to the No-Free-Lunch (NFL) theorem [8], there is no algorithm for solving all optimization problems. This means that an optimizer may perform well in a set of problems and fail to solve a different set of problems. In other words, the average performance of optimizes is equal when considering all optimization problems. Therefore, there are still problems that can be solved by new optimizers better than the current optimizers.

* Address: School of Information and Communication Technology, Griffith University, Nathan Campus, Brisbane, QLD 4111, Australia.
E-mail address: seyedali.mirjalili@griffithuni.edu.au

http://dx.doi.org/10.1016/j.knosys.2015.07.006
0950-7051/© 2015 Elsevier B.V. All rights reserved.

This is the motivation of this work, in which a novel nature-inspired algorithm is proposed to compete with the current optimization algorithms. The main inspiration of the proposed algorithm is the navigating mechanism of moths in nature called transverse orientation. The paper first proposes the mathematical model of spiral flying path of moths around artificial lights (flames). An optimization algorithm is then proposed using the mathematical model to solve optimization problems in different fields. The rest of the paper is organized as follows.

Section 2 reviews the literature of stochastic and heuristic optimization algorithms. Section 3 presents the inspiration of this work and proposes the MFO algorithm. The experimental setup, results, discussion, and analysis are provided in Section 4. Section 5 investigates the effectiveness of the proposed MFO algorithm in solving nine constrained engineering design problems: welded beam, gear train, three-bar truss, pressure vessel, cantilever beam, I-beam, tension/compression spring, 15-bar truss, and 52-bar truss design problems. In addition, Section 6 demonstrates the application of MFO in the field of marine propeller design. Eventually, Section 7 concludes the work and suggests several directions for future studies.

2. Literature review

This section first reviews the state-of-the-art and then discusses the motivations of this work.

A brief history of stochastic optimization techniques was provided in the introduction. A general classification of the algorithms in this field is based on the number of candidate solutions that is improved during optimization. An algorithm may start and perform the optimization process by single or multiple random solution(s). In the former case the optimization process begins with a single random solution, and it is iteratively improved over the iterations. In the latter case a set of solutions (more than one) is created and improved during optimization. These two families are called individual-based and population-based algorithms.

There are several advantages and disadvantages for each of these families. Individual-based algorithms need less computational cost and function evaluation but suffer from premature convergence. Premature convergence refers to the stagnation of an optimization technique in local optima, which prevents it from convergence towards the global optimum. In contrary, population-based algorithms have high ability to avoid local optima since a set of solutions are involved during optimization. In addition, information can be exchanged between the candidate solutions, which assist them to overcome different difficulties of search spaces. However, high computational cost and the need for more function evaluation are two major drawbacks of population-based algorithms.

The well-known algorithms in the individual-based family are: Tabu Search (TS) [6,9], hill climbing [10], Iterated Local Search (ILS) [11], and Simulated Annealing (SA) [12]. TS is an improved local search technique that utilizes short-term, intermediate-term, and long-term memories to ban and truncate unpromising/repeated solutions. Hill climbing is also another local search and individual-based technique that starts optimization by a single solution. This algorithm then iteratively attempts to improve the solution by changing its variables. ILS is an improved hill climbing algorithm to decrease the probability of trapping in local optima. In this algorithm, the obtained optimum at the end of each run is perturbed and considered as the starting point in the next iteration. Eventually, the SA algorithm tends to accept worse solutions proportional to a variable called cooling factor. This assists SA to promote exploration of the search space and consequently avoid local optima.

Although different improvements of individual-based algorithms promote local optima avoidance, the literature shows that population-based algorithms are better in handling this issue. Regardless of the differences between population-based algorithms, the common is the division of optimization process to two conflicting milestones: exploration versus exploitation [13]. The exploration milestone encourages candidate solutions to change abruptly and stochastically. This mechanism improves the diversity of the solutions and causes high exploration of the search space. In PSO, for instance, the inertia weight maintains the tendency of particles toward their previous directions and emphasizes exploration. In GA, high probability of crossover causes more combination of individuals and is the main mechanism for the exploration milestone.

In contrast, the exploitation milestone aims for improving the quality of solutions by searching locally around the obtained promising solutions in the exploration milestone. In this milestone, candidate solutions are obliged to change less suddenly and search locally. In PSO, for instance, low inertia rate causes low exploration and high tendency toward to best personal/global solutions obtained. Therefore, the particles converge toward best points instead of churning around the search space. The mechanism that brings GA exploitation is the mutation operators. Mutation causes slight random changes in the individuals and local search around the candidate solutions.

Exploration and exploitation are two conflicting milestones where promoting one results in degrading the other [14]. A right balance between these two milestones can guarantee a very accurate approximation of the global optimum using population-based algorithms. On one hand, mere exploration of the search space prevents an algorithm from finding an accurate approximation of the global optimum. On the other hand, mere exploitation results in local optima stagnation and again low quality of the approximated optimum. Due to the unknown shape of the search space for optimization problems, in addition, there is no clear accurate timing for transition between these two milestones. Therefore, population-based algorithms balance exploration and exploitation milestones to firstly find a rough approximation of the global optimum, and then improve its accuracy.

The general framework of population-based algorithms is almost identical. The first step is to generate a set of random initial solutions $(\vec{X}) = \{\vec{X_1}, \vec{X_2}, \ldots, \vec{X_n}\}$. Each of these solutions is considered as a candidate solution for a given problem, assessed by the objective function, and assigned an objective value: $(\vec{O}) = \{O_1, O_2, \ldots, O_n\}$. The algorithm then combines/moves/updates the candidate solutions based on their fitness values with the hope to improve them. The created solutions are again assessed by the objective function and assigned their relevant fitness values. This process is iterated until the satisfaction of an end condition. At the end of this process, the best solution obtained is reported as the best approximation for the global optimum.

Recently, many population-based algorithms have been proposed. They can be classified to three main categories based on the source of inspiration: evolution, physic, or swarm. Evolutionary algorithms are those who mimic the evolutionary processes in nature. Some of the recently proposed evolutionary algorithms are Biogeography-based Optimization (BBO) algorithm [15], evolutionary membrane algorithm [16], human evolutionary model [17], and Asexual Reproduction Optimization (ARO) [18].

The number of recently proposed swarm-based algorithms is larger than evolutionary algorithms. Some of the most recent ones are Glowworm Swarm Optimization (GSO) [19], Bees Algorithm (BA) [20], Artificial Bee Colony (ABC) algorithm [21], Bat Algorithm (BA) [22], Firefly Algorithm (FA) [23], Cuckoo Search (CS) algorithm [24], Cuckoo Optimization Algorithm (COA) [25], Grey Wolf Opti-