Applied Soft Computing 32 (2015) 72–79

Contents lists available at ScienceDirect

Applied Soft Computing

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

Ions motion algorithm for solving optimization problems

Behzad Javidy [a], Abdolreza Hatamlou [a,*], Seyedali Mirjalili [b]

[a] Department of Computer Science, Khoy Branch, Islamic Azad University, Khoy, Iran
[b] School of Information and Communication Technology, Griffith University, Nathan, Brisbane, QLD 4111, Australia

ARTICLE INFO

Article history:
Received 24 June 2014
Received in revised form 18 March 2015
Accepted 19 March 2015
Available online 28 March 2015

Keywords:
Ions motion algorithm
Optimization problems
Meta-heuristic algorithms

ABSTRACT

This paper proposes a novel optimization algorithm inspired by the ions motion in nature. In fact, the proposed algorithm mimics the attraction and repulsion of anions and cations to perform optimization. The proposed algorithm is designed in such a way to have the least tuning parameters, low computational complexity, fast convergence, and high local optima avoidance. The performance of this algorithm is benchmarked on 10 standard test functions and compared to four well-known algorithms in the literature. The results demonstrate that the proposed algorithm is able to show very competitive results and has merits in solving challenging optimization problems.

1. Introduction

Over the last two decades, numerous algorithms with inspiration from the nature have been proposed for solving various optimization problems. Some of the most popular are Genetic Algorithm (GA) [1–5], Differential Evolution Algorithm (DE) [6–10], Particle Swarm Optimization (PSO) [11–14], and Artificial Bee Colony (ABC) [15–19]. Optimization algorithms have advantages and disadvantages compared to each other and may show different performances when solving discrete and continuous problems.

There are two conflicting criteria when assessing two algorithms: convergence rate versus quality of the final solution. Fast convergence speed may result in premature convergence and entrapment in local optima. On the other hand, favoring quality of solutions may results in more extensive search of the search space and consequently slower convergence. To address these two issues, the researchers improve the current algorithms or propose new techniques. In this paper, a new optimization algorithm called Ions Motion Optimization (IMO) is proposed as a competitive algorithm in this field.

The IMO algorithm is a population-based algorithm inspired from properties of ions in nature. Our main objectives when designing this algorithm are to require IMO to have the least number of tuning parameters, low computational complexity, fast convergence, and high local optima avoidance. The main inspirations of the IMO algorithm are two different ions: anion (a negative charged particle) and cation (a positive charged particle). The IMO algorithm divides the population of candidate solutions to two sets of negative charged ions and positive charged ions, and improve them according to the important characteristics of the ions "*ions with the same charges repel each other, but with opposite charges attract each other*" [20].

In liquid state, the ions have greater freedom of motion compared to the solid phase (crystal) where high attraction forces between prevent ions from moving around freely. In fact, ions face minor motion and mostly vibrate in their position in solid phase. The IMO algorithm also mimics these two phases to perform diversification and intensification during optimization. The rest of the paper is organized as follows:

Section 2 provides the literature review of the recent meta-heuristics algorithms. Section 3 proposes the IMO algorithm. The results and discussion of the test functions are provided in Section 4. Eventually, Section 5 concludes the works and opens some avenues for future studies.

2. Literature review

Stochastic optimization techniques refer to the set of approaches that generate random solutions for an optimization problem. Based on the mechanism of the algorithm, the random solutions are combined in order to improve the oval quality of the initial solutions. This process is iterated until the satisfaction of a termination condition. A taxonomy of stochastic optimization algorithms here is based on the number of random solutions generated in each iteration. An algorithm may create single or multiple random solutions in every iteration.

* Corresponding author. Tel.: +98 9144627893.
E-mail address: hatamlou@iaukhoy.ac.ir (A. Hatamlou).

http://dx.doi.org/10.1016/j.asoc.2015.03.035
1568-4946/

Simulated Annealing (SA) [21] is an example of the algorithms with single solution in every iteration. The optimization starts with one solution and it is improved over the course of iterations. The solution tends to randomly change its position based on a cooling factor. The higher cooling factor, the more sudden random changes in the solution. The cooling factor is increased over the iterations which results in convergence of the solution around an optimum. The limitation of such algorithm is that they are very likely to be trapped in local optima although the computational complexity is low.

Population-based algorithms belong to the stochastic optimization techniques with multiple solutions in each iteration. The optimization process starts with creating a set of random solutions (population). These solutions are then merged to create a new population. In order to guarantee improvement of the whole population, best solutions are usually selected for improving the quality of the solutions with poor quality. Obviously, the main advantage of these approaches is high local optima avoidance since a population is employed to search the search space. However, the computational complexity of the population-based algorithms is much higher than algorithms with single candidate solution.

The population-based algorithms themselves can be divided to three groups based on inspiration: swarm-inspired, evolution-inspired, and physics-inspired algorithms [22]. The swarm-based algorithms mostly mimic the social and individual behavior of swarm, herds, schools, or groups of creatures in nature. The PSO algorithm is the most popular swarm-inspired algorithm in this class, which imitate the collective behavior of birds. In this algorithm, candidate solutions are able to save and retrieve the best solutions they obtained so far as well as the best solution achieved by the whole swam. The convergence is guaranteed by moving toward the best positions obtained so far and the guidance by the best solution of the swarm.

Another well-known swarm-inspired algorithm is Ant Colony Optimization (ACO) proposed by Dorigo [23]. As its name suggests, this algorithm simulates the social and collective behavior of an ant colony. The main inspiration of this algorithm is the mechanism that ants utilize pheromone to find the shortest path from nest to foods. At every iteration, search agents of ACO record the history of solutions and qualities in order to fill out a pheromone matrix and eventually improve other solutions. The Artificial Bee Colony (ABC) [15–19] is a similar method that mimics the social life style of bees in a bee colony. In this algorithm, the search agents are divided into different groups to explore (scout bees) and exploit (onlooker and employed) the search space. Some of the other algorithms in this class are Bat Algorithm proposed by Yang [24], Glowworm Swarm Optimization (GSO) proposed by Krishnanand and Ghose [25], Multi-swarm optimization [26], Gray Wolf Optimizer [22], Artificial Fish-Swarm Algorithm (AFSA) [27], Firefly Algorithm (FA) [28], Cuckoo Search (CS) [29], Krill Herd (KH) [30], and Heart Algorithm [31].

The second class of algorithms is evolution-inspired algorithm. Such algorithms mostly simulate evolutionary phenomena in nature. Similar to other population-based algorithm, the optimization process starts with a set of random solutions. Then, three main operators evolve the initial population: selection, re-production, and mutation. The selection operator is responsible for choosing proper individuals based on their fitness values. The re-production operator combines the selected individuals by the selection operator. Eventually, the mutation operator randomly changes the re-produced new individuals in order to maintain diversity of the whole population. The most well-known algorithm in this class is GA [32]. This algorithm considers candidate solutions as chromosomes and the parameters as genes. In every generation, the chromosomes with higher fitness values have higher chance to crossover with other chromosomes. Therefore, the overall fitness of all chromosomes is increased over the course of iterations. Some of the other algorithms in this class are Differential Evolution (DE) [33], Evolution Strategy (ES) [34], Genetic Programming (GP) [35], and Biogeography-based Optimizer (BBO) [36].

The last class of algorithms is physics-based algorithms where the main inspiration mostly originates from physical rules and phenomena in nature. Similar to the other two classes, optimization is done by a group of candidate solutions called search agents. The key difference here is that the search agents are moved/combined based on physics-inspired concepts. For instance, Magnetic Optimization Algorithm (MOA) [36] simulates the electromagnetic forces between electromagnetic particles to move the search agents around the search space. Since the electromagnetic force is proportional to the fitness of particles, search agents tend to be attracted toward the fittest particles. Therefore, the search agents of this algorithm are improved by moving toward the best solutions. A similar algorithm to MOA is Gravitational Search Algorithm (GSA) [37]. The GSA algorithm considers the search agent as masses that attract each other based on the gravitational forces between them, which are again proportional to their fitness functions. Regarding to the movement of masses, Newtonian law of motion is also utilized by the GSA algorithm. Some of the other algorithms in this class are Ray Optimization (RO) [38], States of Matter Search (SMS) [39], Big-Bang Big-Crunch (BBBC) [40], Black Hole (BH) [41], Artificial Chemical Reaction Optimization Algorithm (ACROA) [42], and Kinetic Gas Molecules Optimizer [43].

All the algorithms in three classes have their own advantages and disadvantages. According to no-free-lunch theorem [44], none of them is able to solve all optimization problems. Regardless of differences in the mechanisms of population-based algorithms in this field, the common is the division of the search process to two main milestones: diversification versus intensification. Diversification refers to the milestone where candidate solutions tend to be merged more frequently and find promising areas of the search space. In other words, candidate solutions face sudden changes in diversification milestone in order to explore the search space as broad as possible. Contradictory, candidate solutions are prone to very little changes in the intensification milestone. In fact, intensification milestone promotes convergence toward the best solutions obtained in the diversification milestone. As discussed in Section 1, these two phases are in conflict. Favoring diversification result in higher local optima avoidance, whereas emphasizing intensification yields to faster convergence rate. The following section proposes a new physics-based algorithm with two specific milestones for diversification and intensification.

3. Ions motion optimization (IMO) algorithm

This section first discusses the inspirations of the IMO algorithm. The mathematical model and the algorithm are then presented.

3.1. Inspirations

The word "*ion*" is a Greek term. English physician Michael Faraday introduced this term in 1834. Generally speaking, charged particles are called ion and can be divided to two types:

- Anion: ions with negative (−) charge.
- Cation: ions with positive (+) charge.

The conceptual model of anions and cations are illustrated in Fig. 1.

The main inspiration of the IMO algorithm is the fact that ions with similar charges tend to repel, whereas ions with opposite