Expert Systems With Applications 90 (2017) 484–500

Contents lists available at ScienceDirect

Expert Systems With Applications

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

An improved Opposition-Based Sine Cosine Algorithm for global optimization

Mohamed Abd Elaziz[a,b], Diego Oliva[c,d], Shengwu Xiong[a,e,*]

[a] School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China
[b] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig, Egypt
[c] Departamento de Ciencias Computacionales, Universidad de Guadalajara, CUCEI Av. Revolucion 1500, Guadalajara, Jal, Mexico
[d] Cybernetics Institute, Tomsk Polytechnic University, Lenin Avenue 30, Tomsk, Russian Federation
[e] Hubei Collaborative Innovation Center of Basic Education Information Technology Services, Hubei University of Education, Wuhan, China

ARTICLE INFO

Article history:
Received 5 May 2017
Revised 21 July 2017
Accepted 22 July 2017
Available online 15 August 2017

Keywords:
Sine Cosine Algorithms (SCA)
Opposition-Based Learning (OBL)
Metaheuristic (MH)
Engineering problems

ABSTRACT

Real life optimization problems require techniques that properly explore the search spaces to obtain the best solutions. In this sense, it is common that traditional optimization algorithms fail in local optimal values. The Sine Cosine Algorithms (SCA) has been recently proposed; it is a global optimization approach based on two trigonometric functions. SCA uses the sine and cosine functions to modify a set of candidate solutions; such operators create a balance between exploration and exploitation of the search space. However, like other similar approaches, SCA tends to be stuck into sub-optimal regions that it is reflected in the computational effort required to find the best values. This situation occurs due that the operators used for exploration do not work well to analyze the search space. This paper presents an improved version of SCA that considers the opposition based learning (OBL) as a mechanism for a better exploration of the search space generating more accurate solutions. OBL is a machine learning strategy commonly used to increase the performance of metaheuristic algorithms. OBL considers the opposite position of a solution in the search space. Based on the objective function value, the OBL selects the best element between the original solution and its opposite position; this task increases the accuracy of the optimization process. The hybridization of concepts from different fields is crucial in intelligent and expert systems; it helps to combine the advantages of algorithms to generate more efficient approaches. The proposed method is an example of this combination; it has been tested over several benchmark functions and engineering problems. Such results support the efficacy of the proposed approach to find the optimal solutions in complex search spaces.

1. Introduction

Optimization is present in several fields of science and engineering, this is a process where the best solution of a specific problem is found using a search mechanism. In recent years a group of optimization approaches called metaheuristic has taken the attention of the scientific community. Metaheuristic Algorithms (MA) mimic a natural process to find the optimal solution. MA performs a stochastic search of the best parameters in an optimization problem. The main idea of these methods is the collective behavior that exists between the candidate solutions. These search agents interchange information about their positions in the search space. Using different operators that depend on the metaphor of each algorithm, the search agents are displaced to new positions where the probability to find optimal solutions is increased. The goal is to have a good balance between the exploration of the entire search space and the exploitation of the prominent regions. Several MA techniques have been developed in the last years, this fact is related to the no-free-lunch theorem that states that not all the optimization algorithms can be applied to the same problem (Wolpert & Macreadym, 1997). In other, words it is necessary to find the best MA that can be adapted to the real life problems to be solved. There have been proposed a different classification of MA, however, they commonly are divided into swarm algorithms and evolutionary algorithms (Mirjalili, 2015b). The main difference between them is that evolutionary algorithms use operators that imitate the process of mutation and crossover from the genetic theory (Mirjalili, 2015b).

On the other hand, swarm techniques simulate different behaviors from nature. For example, Particle Swarm Optimization

[*] Corresponding author.
E-mail addresses: abd_el_aziz_m@yahoo.com, meahmed@zu.edu.eg (M. Abd Elaziz), diego.oliva@cucei.udg.mx (D. Oliva), xiongsw@whut.edu.cn (S. Xiong).

http://dx.doi.org/10.1016/j.eswa.2017.07.043
0957-4174/© 2017 Elsevier Ltd. All rights reserved.

(PSO) is inspired by bird flocking and fish schooling (Kennedy & Eberhart, 1995). Artificial Bee Colony is another interesting MA where the operators are bees that are searching for food sources (Karaboga, 2005). New swarm approaches as Grey Wolf Optimizer (GWO) or Whale Optimization Algorithm (WOA) have been recently proposed (Mirjalili, Mirjalili, & Lewis, 2014), (Mirjalili & Lewis, 2016a). These methods mimic the hunting behavior of wolves and whales respectively. A considerable amount of literature has been published on MA. In the state-of-the-art, they have been proposed several approaches that simulate different processes from nature. For example, the Crow Search Algorithm (CSA) that emulates the behavior of crows to hide and stole food (Askarzadeh, 2016). Meanwhile, the Wind Driven Optimization (WDO) is based on the motion of the wind in the atmosphere (Bayraktar, Komurcu, Bossard, & Werner, 2013). Another interesting and popular approach is the Flower Pollination Algorithm (FPA) that is inspired in the process of transfer pollen between flowers (Yang, 2012). In this context, the Tree-Seed Algorithm has been developed by (Kiran, 2015), it is based on the relations between trees and their seeds. In (Cuevas, Dıaz Cortés, & Oliva Navarro, 2016) the Social Spider Optimization (SSO) is introduced as an alternative approach for global optimization. Another recently proposed approach is the Stochastic Fractal Search (Salimi, 2015), different to other MA this algorithm has not inspired in nature, and it considers a mathematic concept called the fractal to optimize complex problems. In this context, an interesting MA proposed in 2016 is the Sine Cosine Algorithm (SCA), it was introduced as an alternative for global optimization (Mirjalili, 2015b). The SCA uses the mathematical functions sine and cosine to perform the exploitation and exploration of the search space. The optimization process of SCA considers two elements of the set of candidate solutions. One of the selected elements affects the next position that the other element will take. In other words, the next position of one of the elements could be inside of a neighbor area of the other candidate solution or outside of this radio. The sine and cosine functions are used to compute the new positions using some variables that permit select one of both mathematical operators (sine or cosine). The SCA has been tested over a big amount of benchmark function showing good performance in comparison with similar approaches (Mirjalili, 2015b). In the same, context SCA has also been applied to the design of airfoil in order to verify its capabilities over real problems (Mirjalili, 2015b). Recently SCA has also been applied for in different problems like binarization of handwritten Arabic text (Mudhsh, Xiong, Abd ElAziz, Hassanien, & Duan, 2017) and for solving the unit commitment problem in energy production (Kaur S, 2016). Moreover and interesting implementation of SCA for detection of galaxies using image retrieval is presented in (Abd ElAziz, Selim, & Xiong, 2017). In this context, SCA has also been modified to solve multi-objective optimization problems (Tawhid & Savsani, 2017). Such methods are based on the standard version of SCA. The main drawback of SCA is that like other MA, its accuracy and convergence are affected by the calibration and randomness of some internal parameters this fact is similar to other MA. Some values should be selected according to the problem to be solved, and other settings are modified in the iterative process.

In the related literature, it has been presented an improved version SCA with an elitism strategy, and it is applied for feature selection in machine learning (Sindhu, Ngadiran, & Yacob, 2017). This version also includes a modified method for update the solutions. The disadvantage of this approach is that it includes an extra parameter that should be tuned, and it is desired that MA has less human interaction. Here is important to mention that since the SCA was introduced in 2015 there exist a few amount of applications, and researchers still looking for problems in which the features of SCA can be useful.

Metaheuristic algorithms are not perfect, some of them have several problems that affect their accuracy and performance. In order to avoid these situations, the Opposition-Based Learning (OBL) has been introduced, the OBL takes a candidate solution and generates their opposite position in search space (Tizhoosh, 2005). Using a single rule OBL verifies if the opposite value or the candidate solution has the best objective function value. Depending on the algorithm this process could be applied at initialization or when an operator modifies the set of feasible solutions. OBL has demonstrated is efficacy improving several MA. In (Bulbul, Pradhan, Roy, & Pal, 2015), the authors proposed the OBL Krill Herd (KH) algorithm for economic load dispatch problem. The Firefly (FF) algorithm has also been modified using OBL for numerical optimization problems (Verma, Aggarwal, & Patodi, 2016). The OBL has also increased the convergence speed of MA, for example, the Electromagnetism-Like Optimization (Cuevas, Oliva, Zaldivar, Cisneros, & Pajares, 2012). The Opposition-Based rule has also been used to estimate parameters in control engineering using the Shuffled Frog Leaping (SFL) algorithm (Ahandani & Alavi-Rad, 2015). The use of OBL has also been extended to multi-objective optimization (Ma et al., 2014). All of these works show that OBL is an interesting mechanism to achieve better results in optimization problems.

The aim of this paper is to introduce a modified version of SCA called Opposition-Based Sine Cosine Algorithm (OBSCA). The use of OBL in combination with the optimization features of SCA improves substantially the accuracy and performance of the standard SCA. Such improvement the affronts the disadvantages of the standard SCA, preserving the good optimization capabilities. The proposed OBSCA has been experimental tested over an extensive set of mathematical benchmark problems. Moreover, in order to prove that OBSCA is able to solve real-life optimization problems, it was tested over benchmark engineering problems. Comparisons with other similar approaches indicate that the OBSCA can provide better results in terms of accuracy and efficacy. The experiments and comparisons are supported by different metrics and statistical validations.

The rest of the paper is organized as follows. Section 2 introduces Preliminaries over, the standard SCA and Opposition-based Learning. The proposed OBSCA is introduced in Section 3. Meanwhile, in Section 4 are presented the experiments and comparisons. Finally Section 5 presents the conclusions.

2. Preliminaries

2.1. Sine Cosine Algorithm

The sine cosine algorithm is a new metaheuristic algorithm (Mirjalili, 2015b), the solutions are updated based on the sine or cosine function as in equations (1) or (2), respectively:

$$X_i = X_i + r_1 \times \sin(r_2) \times |r_3 P_i - X_i| \tag{1}$$

$$X_i = X_i + r_1 \times \cos(r_2) \times |r_3 P_i - X_i| \tag{2}$$

In general, from the previous two functions are combined into one function as in the following equation (Mirjalili, 2015b):

$$X_i = \begin{cases} X_i + r_1 \times \sin(r_2) \times |r_3 P_i - X_i| & if\ r_4 < 0.5 \\ X_i + r_1 \times \cos(r_2) \times |r_3 P_i - X_i| & if\ r_4 \geq 0.5 \end{cases} \tag{3}$$

Where P_i is the destination solution, X_i is the current solution, |.| indicates the absolute value. r_1, r_2, r_3 and r_4 are random variables.

The parameter r_1 is random variable which responsible for determine the area of the next solution, this area may be either outside space between X_i and P_i or inside them. In (Mirjalili, 2015b) the authors update the parameter r_1 using (4) to balance exploration and exploitation.

$$r_1 = a - t\frac{a}{T} \tag{4}$$