Applied Soft Computing 27 (2015) 28–37

Contents lists available at ScienceDirect

Applied Soft Computing

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

Opposition versus randomness in binary spaces

Z. Seif, M.B. Ahmadi *

Department of Mathematics, College of Sciences, Shiraz University, 71454, Shiraz, Iran

ARTICLE INFO

Article history:
Received 24 April 2012
Received in revised form
24 September 2014
Accepted 29 October 2014
Available online 6 November 2014

Keywords:
Opposition-based learning
Binary spaces
Randomness
Binary gravitational search algorithm

ABSTRACT

Evolutionary algorithms start with an initial population vector, which is randomly generated when no preliminary knowledge about the solution is available. Recently, it has been claimed that in solving continuous domain optimization problems, the simultaneous consideration of randomness and opposition is more effective than pure randomness. In this paper it is mathematically proven that this scheme, called opposition-based learning, also does well in binary spaces. The proposed binary opposition-based scheme can be embedded inside many binary population-based algorithms. We applied it to accelerate the convergence rate of binary gravitational search algorithm (*BGSA*) as an application. The experimental results and mathematical proofs confirm each other.

1. Introduction

Many real-world problems such as feature selection and dimensionality reduction, data mining, unit commitment, and cell formation, are formulated as optimization problems with binary variables. In addition, problems defined in the real space may be considered in the binary space since binary coding provides some of the algorithms with a lot of flexibility by decomposing each real value and allowing the implicit parallelism to take advantage of this. In the past decades, different kinds of nature-inspired evolutionary optimization algorithms have been designed and applied to solve binary-encoded optimization problems, e.g., binary differential evolution [1], binary particle swarm optimization [2], binary ant colony optimization [3], genetic algorithm [4], and binary gravitational search algorithm [5]. This algorithms start with an initial population vector, which is randomly generated when no preliminary knowledge about the solution space is available. The computation time is directly related to the distance of initial guesses from the optimal solution.

We can improve our chance to start with a closer (fitter) solution by checking the opposite solution simultaneously [6]. The concept of opposition-based learning (*OBL*) was originally introduced by Tizhoosh [6]. It has been utilized in a wide range of learning and optimization fields. *OBL* was first proposed as a machine intelligence scheme for reinforcement learning [6–8]. Afterward, it has been employed to enhance soft computing methods such as fuzzy systems [9,10] and artificial neural networks [11–14]. *OBL* has proven to be an effective method for solving optimization problems by combining it with differential evolution [15–17], particle swarm optimization [18–21], ant colony optimization [22,23], simulated annealing [24], and gravitational search algorithm [25] in a wide range of fields from image processing [10,26,27] to system identification [28,29]. It has been also applied to assist evolutionary algorithms in solving discrete and combinatorial optimization problems [30]. It has been shown that in terms of convergence speed, utilizing random numbers and their opposite is more beneficial than using the pure randomness to generate initial estimates in absence of a prior knowledge about the solution of a continuous domain optimization problem [31]. In this paper, it will be mathematically proven that this fact can be extended to binary optimization problems. It is noticeable that the proofs in [31] are not suitable for binary spaces.

Binary gravitational search algorithm, introduced by Rashedi et al., is a stochastic search algorithm based on the law of gravity and mass interactions [5]. In *BGSA*, the search agents are a collection of masses which interact with each other based on the Newtonian theory that postulates every particle in the universe attracts every other particle with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. *BGSA* is able to optimize both real and binary optimization problems. The effectiveness of this algorithm in solving a set of nonlinear benchmark functions has been proven [5].

This paper is organized as follows. In Section 2 the concept of opposition-based learning in continuous and binary spaces are

* Corresponding author. Tel.: +98 9177170280; fax: +98 7132277116.
E-mail addresses: zseif@shirazu.ac.ir (Z. Seif), mbahmadi@shirazu.ac.ir (M.B. Ahmadi).

http://dx.doi.org/10.1016/j.asoc.2014.10.038
1568-4946/

introduced. It also covers the theorems and proofs corresponding to opposition in binary domains. Section 3 provides a brief review of binary gravitational search algorithm. The outline of opposition-based binary gravitational search algorithm is presented in Section 4. Then, an experimental study is given in Section 5, where *OBL* is applied to accelerate the convergence rate of *BGSA* as an application and the performance of the new algorithm will be evaluated on nonlinear benchmark functions. Finally, a conclusion is given in Section 6.

2. Opposition-based learning

In this section, first, the concept of opposition-based learning in the continuous space is reviewed. Afterward, it will be defined in binary domains and its corresponding theorems will be proven.

2.1. Opposition in continuous domain

Definition 1. Let $x \in [a, b]$ be a real number. The opposite number, denoted by $\breve{x}$, is defined by $\breve{x} = a + b - x$.

This definition can be extended to higher dimensions [6].

Definition 2. Let $X(x_1, x_2, \ldots, x_d)$ be a point in d-dimensional space, where $x_1, x_2, \ldots, x_d$ are real numbers and $x_i \in [a_i, b_i]$, $i = 1, 2, \ldots, d$. The opposite point of X is denoted by $\breve{X}(\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_d)$ where $\breve{x}_i = a_i + b_i - x_i, \quad i = 1, 2, \ldots, d$.

2.2. Opposition in binary domain

Although, opposition-based learning was created for accelerating a continuous search space, it can be applied alongside binary space. In this section we prove that a binary opposite point is more likely to be closer to the solution than a random one. To follow this purpose, first we modify the previous definitions as follows.

Definition 3. Let $x \in \{0, 1\}$. The opposite number, denoted by $\breve{x}$, is defined by $\breve{x} = 1 - x$.

Similarly, this definition can be extended to higher dimensions.

Definition 4. Let $X(x_1, x_2, \ldots, x_d)$ be a point in d-dimensional binary space $S = \{0, 1\}^d$ where $x_i \in \{0, 1\}$, $i = 1, 2, \ldots, d$. The opposite point of X is denoted by $\breve{X}(\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_d)$ where $\breve{x}_i = 1 - x_i, \quad i = 1, \quad 2, \ldots, d$.

In this paper, the distance between two points is computed based on the Hamming distance which is defined as follows.

Definition 5. The *Hamming distance* between two binary vectors $x, y \in \{0, 1\}^d$ is defined by

$$HD(x, y) = \sum_{i=1}^{d} x(i) \oplus y(i), \tag{1}$$

where "$\oplus$" is the *XOR*, and $x(i)$ and $y(i)$ are the ith bits of x and y. In other words, Hamming distance gives the number of positions in which two binary vectors differ. It is obvious that $x(i) \oplus y(i)$ can be replaced by the absolute value of the arithmetic subtraction $\left|x(i) - y(i)\right|$.

Now, we are going to prove our theorems.

Theorem 1. *Every point $X(x_1, x_2, \ldots, x_d)$ in the d-dimensional binary space with $x_i \in \{0, 1\}$, $i = 1, 2, \ldots, d$, has a unique opposite point $\breve{X}(\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_d)$ defined by $\breve{x}_i = 1 - x_i, \quad i = 1, \quad 2, \ldots, d$.*

Proof. Let both $\breve{X}(\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_d)$ and $X'(x'_1, x'_2, \ldots, x'_d)$ be the opposite points of $X(x_1, x_2, \ldots, x_d)$. According to the definition of opposite point, for each i, $1 \le i \le d$, we have $\breve{x}_i = 1 - x_i$ and $x'_i = 1 - x_i$ and so $\breve{x}_i = x'_i$. This means that $\breve{X} = X'$. □

Theorem 2. *Let $X(x_1, x_2, \ldots, x_d)$ be a point in d-dimensional binary space, where $x_i \in \{0, 1\}$, $i = 1, 2, \ldots, d$, and $\breve{X}(\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_d)$ is its opposite point. Then, for each $Y \in \{0, 1\}^d$ we have*

$$HD(X, Y) = d - HD(\breve{X}, Y).$$

where *HD* denotes the Hamming distance.

Proof. It is clear that for each $x \in \{0, 1\}$, $x^2 = x$. So,

$$HD(X, Y) = \sum_{i=1}^{d} x(i) \oplus y(i) = \sum_{i=1}^{d} \left|x_i - y_i\right| = \sum_{i=1}^{d} \left|1 - \breve{x}_i - y_i\right|$$

$$= \sum_{i=1}^{d} (1 - \breve{x}_i - y_i)^2 = \sum_{i=1}^{d} (\breve{x}_i^2 - 2\breve{x}_i + y_i^2 - 2y_i + 2\breve{x}_i y_i + 1).$$

Since $\breve{x}_i = \breve{x}_i^2$ and $y_i = y_i^2$, we have

$$\sum_{i=1}^{d} (\breve{x}_i^2 - 2\breve{x}_i + y_i^2 - 2y_i + 2\breve{x}_i y_i + 1) = \sum_{i=1}^{d} (-\breve{x}_i^2 - y_i^2 + 2\breve{x}_i y_i + 1)$$

$$= \sum_{i=1}^{d} (1 - (\breve{x}_i - y_i)^2) = d - \sum_{i=1}^{d} \left|\breve{x}_i - y_i\right| = d - HD(\breve{X}, Y).$$

Hence, $HD(X, Y) = d - HD(\breve{X}, Y)$. □

Definition 6. Let $X(x_1, x_2, \ldots x_d)$ be a candidate solution for a binary optimization problem. Assume $f(X)$ is a fitness function which is used to measure candidates optimality. According to opposite point definition, $\breve{X}(\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_d)$ is the opposite point of $X(x_1, x_2, \ldots, x_d)$. Now, if $f(\breve{X}) \ge f(X)$, then point X can be replaced with $\breve{X}$. Otherwise, we continue with X Hence, the point and its opposite are evaluated simultaneously to continue with the fitter one.

When evaluating a solution X to a given problem, simultaneously computing its opposite solution will provide another chance for finding a candidate solution closer to the global optimum. The following theorem answers this significant question: why is an opposite number more effective than an independent random number.

Theorem 3. *Assume $y = f(X)$ is an arbitrary function with at least one solution at $X_s(x_{s_1}, x_{s_2}, \ldots, x_{s_d})$, $x_{s_i} \in \{0, 1\}, \quad i = 1, 2, \ldots, d$. Suppose $X(x_1, x_2, \ldots, x_d)$ and $X_r(x_{r_1}, x_{r_2}, \ldots, x_{r_d})$ are the first and second random guesses in the solution space respectively. Then*

$$(i) \quad Pr(\left\|\breve{X}, X_s\right\| \le \min\{\left\|X, X_s\right\|, \left\|X_r, X_s\right\|\}) > Pr(\left\|X_r, X_s\right\| \le \min\{\left\|X, X_s\right\|, \left\|\breve{X}, X_s\right\|\}), \tag{2}$$

where $d \neq 1$. In other words, The probability that the distance between $\breve{X}$ and X_s be less than or equal to the distance between $\{X, X_r\}$ and X_s is more than the probability that the distance between X_r and X_s be less than or equal to the distance between $\{X, \breve{X}\}$ and X_s. In fact, $\breve{X}$ is more probable than X_r to be the closest to X_s among $\{X, \breve{X}, X_r\}$. Equality holds in Eq. (2) when $d = 1$.

$$(ii) \quad Pr(\min\{\left\|X, X_s\right\|, \left\|\breve{X}, X_s\right\|\} \le \min\{\left\|X, X_s\right\|, \left\|X_r, X_s\right\|\}) > Pr(\min\{\left\|X, X_s\right\|, \left\|X_r, X_s\right\|\} \le \min\{\left\|X, X_s\right\|, \left\|\breve{X}, X_s\right\|\}), \tag{3}$$