Expert Systems With Applications 61 (2016) 378–385

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Linkage artificial bee colony for solving linkage problems

Phuc Nguyen Hong, Chang Wook Ahn*

Department of Computer Engineering, Sungkyunkwan University, Suwon, Republic of Korea

ARTICLE INFO

Article history:
Received 17 February 2016
Revised 5 May 2016
Accepted 30 May 2016
Available online 2 June 2016

Keywords:
Artificial bee colony
Meta-heuristics
Order statistics

ABSTRACT

Nature-inspired meta-heuristics have gained popularity for the solution of many real world complex problems, and the artificial bee colony algorithm is one of the most powerful optimisation methods among the meta-heuristics. However, a major drawback prevents the artificial bee colony algorithm from accurately and efficiently finding final solutions for complex problems, whose variables interact with each other. We propose a novel optimization method based on the artificial bee colony algorithm and statistics. The proposed optimization method is evaluated for Pott models and optimization linkage functions, and the proposed method is verified to outperform traditional artificial bee colony and other meta-heuristics for those cases.

1. Introduction and related work

Meta-heuristic optimization has gained significant popularity for solving complex problems that are challenging to solve using derivative-based techniques. Several nature-inspired meta-heuristic optimization algorithms have been developed. These algorithms are also referred to as population-based meta-heuristics or general-purpose algorithms since they can be applied to a wide range of problems. Some popular population-based meta-heuristics are the genetic algorithm (GA) (Holland, 1975), differential evolution (DE) (Storn & Price, 1997), artificial bee colony (ABC) (Karaboga & Basturk, 2007b), and clonal selection algorithm (CSA)(Leandro & Von Zuben, 2002).

Since ABC was introduced (Akay & Karaboga, 2009; Karaboga & Basturk, 2007a, 2007b), it has been used to address several problems in different fields (Akay, 2013; Apalak, Karaboga, & Akay, 2014). A comprehensive survey of ABC applications and its improved versions is presented in (Tsai, Pan, Liao, & Chu, 2009). ABC has also been implemented in parallel processing in order to improve its performance (Narasimhan, 2009; Subotic, Tuba, Chen, & Stanarevic, 2011; Subotic, Tuba, & Stanarevic, 2010; Zou, Zhu, Chen, & Sui, 2010). The improved ABC versions often increase computation costs and are more complicated than the conventional ABC algorithm (Cavdar, Mohammad, & Alavi, 2013).

However, ABC fails to address the linkage problem where variables are dependent and interactive. The ABC algorithm and its variants assume problem variables are independent, and the optimal solution can be obtained by manipulating one variable at a time. This paper presents a novel category of the ABC method, linkage-aware artificial bee colony (LABC) that can overcome this obstacle. LABC exploits statistics measures for the current population to estimate variable dependency, and can detect a group of dependent variables and manipulate them simultaneously. The proposed method is experimentally evaluated for Pott models and the bit trap functions (Ackley, 1987).

Linkage problems have been actively solved for GAs (Hauschild & Pelikan, 2011), and can be classified according to the number of interactive variables. We introduce pairwise linkage problems for ABC algorithms, and propose a novel statistics-based method to measure the relationship between two variables.

We use the Pott model to evaluate LABC. There are pairwise interactions between variables in the Pott model, and it can model many problems in computer vision, such as segmentation, stereo matching, restoration, and optical flow.

The contributions of this paper are three-fold.

- We introduce a novel LABC that can resolve the linkage problems
- We propose a novel statistics measure to compute the dependency of variables
- We evaluate LABC for the Pott model and linkage functions. The qualitative and quantitative experimental results show that LABC is a promising optimization algorithm, and the proposed reduced energy function can be used as a real application for meta-heuristic algorithms

The order Statistics Coefficient (OSC) and Robust Order Statistics Coefficient (ROSC) are presented in Section 2. ABC and LABC are presented and discussed in Section 3, and experimental results are reported in Section 4. Conclusions are presented in Section 5.

* Corresponding author.
E-mail addresses: hongphuc@skku.edu (P.N. Hong), cwan@skku.edu (C.W. Ahn).

http://dx.doi.org/10.1016/j.eswa.2016.05.043
0957-4174/

Fig. 1. Order statistics coefficients for two sets of data points.

2. Statistics measures for variable dependency

2.1. Order statistics coefficient

OSC (Xu, Chang, Hung, Kwan, & Fung, 2007) can measure the monotonically non-linear association of two sets of data points using order statistics and the rearrangement inequality. OSC sets of data points are highly correlated if large or small values of a given set occur in conjunction with large or small values of another. The sets are uncorrelated if large or small values of one set are associated with small or large values, respectively, of another.

Consider two time series $\mathbf{X} = \{x_1, x_2, ..., x_m\}$ and $\mathbf{Y} = \{y_1, y_2, ..., y_m\}$, of length m. The time series pairwise results are the set $\mathbf{L} = \{(x_1, y_1), ..., (x_m, y_m)\}$. Rearranging $\mathbf{L}$ according to the values of x_i, we obtain $\mathbf{L}_1 = \{\left(x_{(1)}, y_{[1]}\right), ..., \left(x_{(m)}, y_{[m]}\right)\}$, where $(x_{(1)} \le ... \le x_{(m)})$ are the order statistics of $\mathbf{X}$, and $y_{[1]}$, ..., $y_{[m]}$ are the associated concomitants (David & Nagaraja, 2003). Rearranging $\mathbf{L}$ according to the values of y_i, we obtain $\mathbf{L}_2 = \{\left(x_{[1]}, y_{(1)}\right), ..., \left(x_{[m]}, y_{(m)}\right)\}$, where $y_{(1)} \le ... \le y_{(m)}$. The OSC is

$$r(\mathbf{X}, \mathbf{Y}) \triangleq \frac{\sum_{i=1}^{m} (x_{(i)} - x_{(m-i+1)}) y_{[i]}}{\sum_{i=1}^{m} (x_{(i)} - x_{(m-i+1)}) y_{(i)}}, \tag{1}$$

where $r(\mathbf{X}, \mathbf{Y})$ is in the interval $[-1, 1]$, and larger values indicate higher correlation between $\mathbf{X}$ and $\mathbf{Y}$. Fig. 1 shows an illustration of OSC distance for two sets of data points.

2.2. Motivation to improve order statistics coefficients

OSC is highly sensitive to changes in association, tolerates monotone nonlinear transformations, and is robust to noise (Xu et al., 2007). This makes OSC suitable to use as a metric for measuring image similarity. Given two series of data points of the same size, set $\mathbf{Z}$ can be constructed from the two series and OSC can be computed using $\mathbf{Z}$ to measure the degree of correlation between the data series. However, a problem limiting OSC usage is that the OSC is undefined for a data series that includes equal values.

2.3. ROSC

The concept of an ROSC is to monotonically increase the value of each element in ordered sets $\mathbf{Z}_1$ and $\mathbf{Z}_2$ by its rank in $\mathbf{Z}_1$ and $\mathbf{Z}_2$. Similar to OSC, an ROSC forms the two time series pairwise first, resulting in a set, $\mathbf{Z} = \{z_1, z_2, ..., z_m\}$, and rearranges $\mathbf{Z}$ according to the values of x_i and y_i to result in new ordered sets, $\mathbf{Z}_1 = \{\left(x_1^1, y_1^2\right), ..., \left(x_m^1, y_m^2\right)\}$ and $\mathbf{Z}_2 = \{\left(x_1^2, y_1^1\right), ..., \left(x_m^2, y_m^1\right)\}$, respectively. Increasing the value of each element in $\mathbf{Z}_1$ and $\mathbf{Z}_2$, we obtain new sets $\mathbf{Z}_1' = \left\{\left(x_1^{1'}, y_1^{2'}\right), ..., \left(x_m^{1'}, y_m^{2'}\right)\right\}$ and $\mathbf{Z}_2' = \left\{\left(x_1^{2'}, y_1^{1'}\right), ..., \left(x_m^{2'}, y_m^{1'}\right)\right\}$, where $x_i^{1'} = x_i^1 \lambda\left(x_i^1\right)$; $y_i^{2'} = y_i^2 \lambda\left(y_i^2\right)$; $y_i^{1'} = y_i^1 \lambda\left(y_i^1\right)$. $\lambda\left(x_i^1\right)$ is the rank of element x_i^1 in $\mathbf{Z}_1$; and $\lambda\left(y_i^1\right)$ and $\lambda\left(y_i^2\right)$ are the ranks of elements y_i^1 and y_i^2 in $\mathbf{Z}_2$, respectively. An ROSC is defined as

$$ROSC(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=1}^{m} \left(x_i^{1'} - x_{m-i+1}^{1'}\right) y_i^{2'}}{\sum_{i=1}^{m} \left(x_i^{1'} - x_{m-i+1}^{1'}\right) y_i^{1'}} \tag{2}$$

where exploiting the sorted sets, $\mathbf{Z}_1$ and $\mathbf{Z}_2$, the ranks of x_i^1 and y_i^1 can be computed as $\lambda\left(x_i^1\right) = i$ and $\lambda\left(y_i^1\right) = i$.

An ROSC has several important properties as follows.

2.3.1. Property 1

An ROSC is within the interval $[-1, 1]$.

Proof. Following (David & Nagaraja, 2003), rearrange as

$$\sum_{i=1}^{m} x_{m-i+1}^{1'} y_i^{1'} \le \sum_{i=1}^{m} x_i^{1'} y_i^{2'} \le \sum_{i=1}^{m} x_i^{1'} y_i^{1'} \tag{3}$$

and

$$\sum_{i=1}^{m} x_{m-i+1}^{1'} y_i^{1'} \le \sum_{i=1}^{m} x_{m-i+1}^{1'} y_i^{2'} \le \sum_{i=1}^{m} x_i^{1'} y_i^{1'}. \tag{4}$$

Then subtracting (3) by (4) and dividing by $\sum_{i=1}^{m} \left(x_i^{1'} - x_{m-i+1}^{1'}\right) y_i^{1'}$, we have $-1 \le ROSC(\mathbf{X,Y}) \le 1$, which completes the proof. □

2.3.2. Property 2

$ROSC(\mathbf{X}, \mathbf{Y}) = -1(+1)$ when X and Y have a monotonic decreasing (increasing) relationship.

Proof. Assume $y_i = \varphi(x_i)$, then if $\varphi(\cdot)$ is a decreasing function, $y_i^{2'} = y_{m-i+1}^{1'}$ for all i. Substituting into (2), $ROSC(\mathbf{X}, \mathbf{Y}) = -1$. Similarly, if $\varphi(\cdot)$ is an increasing function, $y_i^{2'} = y_i^{1'}$ for all i and $ROSC(\mathbf{X}, \mathbf{Y}) = 1$. □

3. Artifical bee colony and linkage artificial bee colony

3.1. Artifical bee colony

The ABC algorithm (Karaboga & Basturk, 2007b) mimics the foraging behavior of real honeybees and includes three groups: employed, onlooker, and scout bees. Employed bees have associations with specific food sources (solutions); onlooker bees within the hive watch the dances of employed bees to select food sources; and scout bees search for food sources randomly.

The general procedure of the original ABC algorithm consists of four steps. In each iteration, employed, onlooker, and scout bees phases are performed in an ordered manner and the iteration repeats until the termination criteria are met.

3.1.1. Initialization

The algorithm generates an initial population randomly according to a uniform distribution within a feasible space, the control parameters are set and the population of food sources is initialized by scout bees using

$$\mathbf{P}_{i,j} = lb_j + rand(0, 1) \times (ub_j - lb_j), \tag{5}$$

where i indicates the ith candidate solution index in the population $\mathbf{P}$, j indicates the jth variable index of a solution, $\mathbf{P}_{i,j}$ is maintained by the lower bound lb_j and the upper bound ub_j, and $rand$(0, 1) generates a number between 0 and 1 with a uniform distribution.

3.1.2. Employed bees

The employed bees search for solutions neighboring the candidate solution, $\mathbf{P}_i$, in their memory to find solutions that have better fitness. A candidate neighboring solution, $\mathbf{s}$, is created by stochastically changing a randomly selected variable j of $\mathbf{P}_i$

$$\mathbf{s}_j = \mathbf{P}_{i,j} + rand(-1, 1) \times (\mathbf{P}_{i,j} - \mathbf{P}_{l,j}) \tag{6}$$