Applied Soft Computing 33 (2015) 15–23

Contents lists available at ScienceDirect

Applied Soft Computing

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

The continuous artificial bee colony algorithm for binary optimization

Mustafa Servet Kiran*

Department of Computer Engineering, Faculty of Engineering, Selçuk University, 42075 Konya, Turkey

ARTICLE INFO

Article history:
Received 2 January 2015
Received in revised form 19 March 2015
Accepted 2 April 2015
Available online 22 April 2015

Keywords:
Artificial bee colony
Binary optimization
Conversion of continuous values
Uncapacitated facility location problem

ABSTRACT

Artificial bee colony (ABC) algorithm, one of the swarm intelligence algorithms, has been proposed for continuous optimization, inspired intelligent behaviors of real honey bee colony. For the optimization problems having binary structured solution space, the basic ABC algorithm should be modified because its basic version is proposed for solving continuous optimization problems. In this study, an adapted version of ABC, ABC_{bin} for short, is proposed for binary optimization. In the proposed model for solving binary optimization problems, despite the fact that artificial agents in the algorithm works on the continuous solution space, the food source position obtained by the artificial agents is converted to binary values, before the objective function specific for the problem is evaluated. The accuracy and performance of the proposed approach have been examined on well-known 15 benchmark instances of uncapacitated facility location problem, and the results obtained by ABC_{bin} are compared with the results of continuous particle swarm optimization (CPSO), binary particle swarm optimization (BPSO), improved binary particle swarm optimization (IBPSO), binary artificial bee colony algorithm (binABC) and discrete artificial bee colony algorithm (DisABC). The performance of ABC_{bin} is also analyzed under the change of control parameter values. The experimental results and comparisons show that proposed ABC_{bin} is an alternative and simple binary optimization tool in terms of solution quality and robustness.

1. Introduction

In recent years, many swarm intelligence-based meta-heuristic approaches have been proposed in order to solve NP-hard optimization problems. Artificial bee colony (ABC) algorithm, which is one of them, has been proposed for solving continuous optimization problems, inspired waggle dance and foraging behavior of real honey bee colonies [1]. If the solution space of the problem is binary structured, either working of the method should be adapted to run on the binary solution space by using binary vectors or continuous values in the solution vectors in the method should be transformed to binary values. In this study, by inspiring an approach proposed for particle swarm optimization (PSO) by Güner and Şevkli [2], the continuous values in the solutions produced by ABC have been moved to binary space and the results obtained by the proposed approach named as ABC_{bin} have been compared with continuous PSO.

Since invention of ABC, studies on ABC in the literature have increased significantly. The ABC algorithm was used for designing of digital IRR filters by Karaboga [3], Singh [4] used it for solving leaf-constrained minimum spanning tree problem, Rao et al. [5] proposed ABC for optimization of distribution network configuration for loss reduction. The ABC was implemented to solve quadratic minimum spanning tree problem by Sundar and Singh [6]. While a modified ABC for real parameter optimization was proposed by Akay and Karaboğa [7]; Karaboğa and Akay [8] also modified ABC for solving constrained optimization problems by using Deb's rules [9] and Pan et al. [10] developed a discrete model of ABC for lot-streaming flow shop scheduling problem. The ABC was also used for neural networks training [11], solving reliability redundancy allocation problems [12], software test suite optimization [13]. In addition, Alatas [14] proposed a chaotic ABC algorithm for avoiding to get stuck on local solutions and a G_{best}-guided ABC for numerical function optimization was proposed by Zhu and Kwong [15]. DisABC which means discrete ABC algorithm for solving uncapacitated facility location problems (UFLPs) has been also proposed by Kashan et al. [16]. Artificial agents of ABC algorithm is modified to work on binary structured solution space to solve binary optimization problems by using logic operators in [17], and an ABC variant is proposed for binary optimization in [18].

According to above applications and modifications of ABC, the algorithm is highly successful for solving optimization problems. But according to best knowledge of the authors, the usage of ABC algorithm that its artificial agents work on continuous solution in

* Tel.: +90 332 223 1992; fax: +90 332 241 06 35.
E-mail address: mskiran@selcuk.edu.tr

http://dx.doi.org/10.1016/j.asoc.2015.04.007

this study is first effort for solving a binary optimization problem. The paper is organized as follows; Section 1 presents introduction and a brief literature review about the method, the ABC algorithm is given in Section 2. Conversion of continuous values to binary values is presented in Section 3 and the general model of the UFLP is presented in Section 4. The computational experiments are conducted in Section 5. Finally, conclusion and future works are given in Section 6.

2. Artificial bee colony algorithm

In the artificial hive of ABC, there are two types of forager called as employed and unemployed foragers. A new food source is produced for each employed forager in the initialization of the algorithm. Employed foragers move not only nectar foraged from food source but also the position information about food sources to the hive. Onlooker bees which are one of the unemployed foragers try to improve food source positions of employed foragers by considering information moved by employed foragers to the hive. In the algorithm, food sources represent feasible solutions for the optimization problem. The other type of unemployed foragers is scout bee. If a food source could not be improved by the employed or onlooker bees in a certain time (*limit*), the employed bee assigned to this food source becomes a scout bee. ABC algorithm consists of four phases named as initialization, employed bee, onlooker bee and scout bee, which are sequentially realized. It should be also mentioned that ABC algorithm is an iterative algorithm and half of the population in the hive is employed bee and the other half is onlooker bee and only one scout bee can occur at the each iteration. Phases of the algorithm are expressed as follows [8]:

Initialization phase: This phase is realized only one time. The population size is first determined and half of the population consists of employed bee and the other half is onlooker bees. A new food source is produced for each employed bee by using Eq. (1).

$$x_{i,j} = x_j^{\min} + \lambda(x_j^{\max} - x_j^{\min}) \quad i = 1, \ldots, N,\ j = 1, \ldots, D \tag{1}$$

where $x_{i,j}$ is jth dimension of ith employed bee, $x_j^{\min}$ and $x_j^{\max}$ is lower and upper bounds of jth parameters, respectively, λ is a random number in range of [0,1], N is the number of employed bees and D is the dimensionality of the optimization problem. In addition, the abandonment counter (AC) of each employed bee is reset in this phase. And then, the fitness values of the food sources of the employed bees are calculated as follows:

$$fit_i = \begin{cases} \dfrac{1}{1+f_i} & \text{if}\,(f_i \geq 0) \\ 1 + abs(f_i) & \text{otherwise} \end{cases} \tag{2}$$

where fit_i is the fitness value of food source of ith employed bee, f_i is the objective function value specific for the optimization problem of food source of ith the employed bee. In addition, *limit* value is determined and a counter (abandonment counter-AC) is created and reset for each food source in this phase.

Employed bee phase: Each employed bee try to find a new food source in order to improve self-solution by using Eq. (3).

$$v_{i,j} = x_{i,j} + \phi(x_{i,j} - x_{k,j}) \quad i, k \in \{1, 2, \ldots, N\},$$
$$j \in \{1, 2, \ldots, D\} \text{ and } i \neq k \tag{3}$$

where $v_{i,j}$ jth dimension of ith candidate solution, $x_{i,j}$ is jth dimension of ith employed bee, $x_{k,j}$ is jth dimension of kth employed bee, ϕ is a random number in range of [−1,+1], N is the number of employed bee and D is the dimensionality of the optimization problem. Also, the neighbor of candidate solution (k) and dimension of the problem (j) are randomly selected among the employed bee population and between dimensionality of the problem, respectively.

The fitness value of the new food source is calculated by using Eq. (2) and if new one is better than old one, the new food source position is memorized by the employed bee and the AC of the food source is reset. Otherwise, AC is increased by 1.

Onlooker bee phase: The employed bee share the position information about the self-food source by dancing in the dance area of the hive. The onlooker bees watch the dance and select an employed bee by using the fitness values of the food sources of the employed bees and roulette wheel. The selection probability is calculated as follows:

$$p_i = \frac{fit_i}{\sum_{j=1}^{N} fit_j} \tag{4}$$

where p_i is probability of being selected ith employed bee. After the selection, the onlooker bees try to improve the solutions of employed bees by using Eq. (3). If new solution obtained by the onlooker bee is better than the solution of employed bee, the employed bee memorizes the solution of onlooker bee and the AC is reset. Otherwise, AC is increased by 1.

Scout bee phase: In this phase, the AC with highest content is fixed and compared with predetermined *limit* value. If value of the AC with maximum content is higher than the *limit*, the employed bee of food source belongs to this AC with maximum content becomes a scout bee. A new solution is generated for this scout bee by using Eq. (1) and the AC of the new food source is reset. The scout bee generated new solution for itself returns to instatement (employed bee).

3. Solution transformation and proposed algorithm

The continuous values of the food source position in the ABC algorithm should be converted to the binary values in order to solve the binary optimization problem. Despite the fact that Eq. (5) has been proposed for conversion of continuous values to binary solution in Şevkli and Güner [19], we used a bit modified version (Eq. (6)) of Eq. (5) in order to increase search ability of the ABC algorithm and provide quick change for the binary values in this study.

$$z_i = floor(|y_i \ \mathbf{mod}\ 2|) \tag{5}$$

where y_i is the solution vector included continuous values and z_i is the binary vector. The continuous value is first divided by 2 and absolute value of the remainder is floored. Therefore, the binary number is obtained. An example is given as follows:

$$z_i = floor(|-3.724\ \mathbf{mod}\ 2|) = floor(|-1.724|) = floor(|1.724|) = 1$$

According to this conversion, if the absolute value of the remainder is between 0 and 0.9999, the binary number will be 0 and if the absolute value of the remainder is between 1 and 1.9999, the binary number will be 1. When absolute value of the remainder is higher than 0.5, despite the fact that absolute value of the remainder is closer to 1 than 0, it is obtained 0 by Eq. (5). The same manner, when absolute value of the remainder is higher than 1.5, despite the fact that absolute value of the remainder is closer to 2 than 1, it is obtained 1 by Eq. (5). Therefore, we propose round operation instead of floor and double mod process given by Eq. (6).

$$z_i = round(|y_i\ \mathbf{mod}\ 2|) \bmod 2 \tag{6}$$