Applied Soft Computing 11 (2011) 3021–3031

Contents lists available at ScienceDirect

Applied Soft Computing

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

A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems

Dervis Karaboga*, Bahriye Akay

Erciyes University, The Department of Computer Engineering, 38039 Melikgazi, Kayseri, Turkiye

ARTICLE INFO

Article history:
Received 1 September 2008
Received in revised form 2 September 2010
Accepted 1 December 2010
Available online 17 December 2010

Keywords:
Swarm intelligence
Modified Artificial Bee Colony algorithm
Constrained optimization

ABSTRACT

Artificial Bee Colony (ABC) algorithm was firstly proposed for unconstrained optimization problems on where that ABC algorithm showed superior performance. This paper describes a modified ABC algorithm for constrained optimization problems and compares the performance of the modified ABC algorithm against those of state-of-the-art algorithms for a set of constrained test problems. For constraint handling, ABC algorithm uses Deb's rules consisting of three simple heuristic rules and a probabilistic selection scheme for feasible solutions based on their fitness values and infeasible solutions based on their violation values. ABC algorithm is tested on thirteen well-known test problems and the results obtained are compared to those of the state-of-the-art algorithms and discussed. Moreover, a statistical parameter analysis of the modified ABC algorithm is conducted and appropriate values for each control parameter are obtained using analysis of the variance (ANOVA) and analysis of mean (ANOM) statistics.

1. Introduction

Structural optimization, engineering design, VLSI design, economics, allocation and location problems are just a few examples of fields in which constrained optimization problems are encountered [1]. A constrained optimization problem (1) is defined as finding parameter vector $\vec{x}$ that minimizes an objective function $f(\vec{x})$ subject to inequality and/or equality constraints:

$$\begin{array}{lll} \text{minimize } f(\vec{x}), & \vec{x}=(x_1,\ldots,x_n)\in\mathbb{R}^n & \\ & l_i \le x_i \le u_i, & i=1,\ldots,n \\ \text{subject to}: & g_j(\vec{x}) \le 0, & \text{for } j=1,\ldots,q \\ & h_j(\vec{x})=0, & \text{for } j=q+1,\ldots,m \end{array} \quad (1)$$

The objective function f is defined on a search space, $\mathbb{S}$, which is defined as a n-dimensional rectangle in $\mathbb{R}^n$ ($\mathbb{S}\subseteq\mathbb{R}^n$). Domains of variables are defined by their lower and upper bounds. A feasible region $\mathbb{F}\subseteq\mathbb{S}$ is defined by a set of m additional constraints ($m\ge 0$) and $\vec{x}$ is defined on feasible space ($\vec{x}\in\mathbb{F}\in\mathbb{S}$). At any point $\vec{x}\in\mathbb{F}$, constraints g_j that satisfy $g_j(\vec{x})=0$ are called active constraints at $\vec{x}$. By extension, equality constraints h_j are also called active at all points of $\mathbb{S}$ [2]. Constrained optimization problems are hard to optimization algorithms and also no single parameter (number of linear, nonlinear and active constraints, the ratio $\rho=|\mathbb{F}|/|\mathbb{S}|$, type of the function, number of variables) is proved to be significant as a major measure of difficulty of the problem [3].

Since most of the optimization algorithms have been primarily designed to address unconstrained optimization problems, constraint handling techniques are usually incorporated in the algorithms in order to direct the search towards the feasible regions of the search space. Methods dealing with the constraints were grouped into four categories by Koziel and Michalewicz [4]: (i) methods based on preserving feasibility of solutions by transforming infeasible solutions to feasible ones with some operators [5,6]; (ii) methods based on penalty functions which introduce a penalty term into the original objective function to penalize constraint violations in order to solve a constrained problem as an unconstrained one. [7–10]; (iii) methods that make a clear distinction between feasible and infeasible solutions [11–15]; (iv) other hybrid methods combining evolutionary computation techniques with deterministic procedures for numerical optimization [16–19].

Koziel and Michalewicz investigated a decoder based constraint handling approach which incorporates a homomorphous mapping, between n-dimensional cube and a feasible search space for solving constrained numerical optimization problems by evolutionary algorithms [4]. The method uses a decoder which transforms a constrained problem to an unconstrained problem via a homomorphous mapping (HM).

Another study related with evolutionary algorithms is Adaptive Segregational Constraint Handling Evolutionary Algorithm (ASCHEA) proposed by Hamida and Schoenauer for constrained optimization problems based on a population level adaptive

* Corresponding author. Tel.: +90 352 437 49 01x32577; fax: +90 352 437 57 84.
E-mail addresses: karaboga@erciyes.edu.tr (D. Karaboga), bahriye@erciyes.edu.tr (B. Akay).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.12.001

penalty function to handle constraints [20]. ASCHEA employs a constraint-driven mate selection for recombination and a segregational selection that favors a given number of feasible individuals and utilizes an equality constraint handling strategy which starts a large feasible domain and tightens it progressively [20].

Runarsson and Yao also investigated the performance of an evolution strategy using different constraint handling methods [21]. They described the over-penalty approach (OPA) which ranks feasible individuals according to their objective function value, followed by the infeasible solutions ranked according to penalty function value.

Runarsson and Yao introduced an approach which balances the objective and penalty functions by stochastic ranking (SR) in Evolution Strategy (ES) [22]. The approach avoids setting a hard-to-set parameter penalty factor and treats constrained optimization as multi-objective optimization where constraints are regarded as an additional objective function. Moreover, they improved the performance of the evolution strategy (Improved Stochastic Ranking, ISR) by employing a search mechanism to overcome the problem of a search bias aligned with the coordinate axis [21].

Mezura-Montes and Coello Coello proposed an evolution strategy which is called Simple multi-membered Evolution Strategy (SMES) to solve global nonlinear optimization problems [23]. The approach uses a diversity mechanism based on allowing infeasible solutions to remain in the population instead of using a penalty function. A feasibility-based comparison mechanism is used to guide the process toward the feasible region of the search space. Also, the initial step size of the evolution strategy is reduced in order to perform a finer search and a combined panmictic recombination technique is applied to improve its exploitation capabilities [23].

In this work, ABC algorithm described by Karaboga [24] inspiring from the foraging behaviour of honey bees is modified to solve constrained optimization problems and its performance is investigated on constrained problems. ABC algorithm is extended by replacing the selection mechanism of the simple ABC algorithm with Deb's selection mechanism [15] in order to cope with the constraints and introducing a probabilistic selection scheme that assigns probability values to feasible solutions based on their fitness values and to infeasible individuals based on their violations. Although it is a known fact that the handling technique employed may influence the performance of the algorithm, a handling technique using Deb's rules, in the category of methods searching for feasible solutions on assumption that any feasible solution is better than any infeasible one [15] was adopted, since it has advantages in terms of simplicity, computational cost, fine tuning requirement etc. over other techniques.

The performance of ABC algorithm has been tested on 13 well-known constrained optimization test problems described in [22] and compared with those of other previous studies [4,20–23].

The remainder of the paper is organized as follows. In Section 2, the modified ABC algorithm adapted for solving constrained optimization problems is introduced and then in Section 3, the proposed algorithm is compared with other algorithms. Finally, in Section 4, conclusions and future works are provided.

2. Modified Artificial Bee Colony algorithm

Artificial Bee Colony algorithm is a recently proposed optimization algorithm that simulates the foraging behaviour of a bee colony [24]. In a real bee colony there are some tasks done by specialized individuals. Bees try to maximize the nectar amount unloaded to the food stores in the hive by this division of labour and self-organization. Division of labor and self-organization are essential components of swarm intelligence. The minimal model of swarm-intelligent forage selection in a honey bee swarm consists of three kinds of bees: employed bees, onlooker bees, and scout bees [25]. Employed bees are responsible from exploiting the nectar sources explored before and they give information to the other waiting bees in the hive about the quality of the food source which they are exploiting. Onlooker bees wait in the hive and establish food source to exploit depending on the information shared by the employed bees. Scouts search environment in order to find a new food source depending on an internal motivation or external clues or randomly.

ABC algorithm uses this minimal model to simulate the collective intelligence in foraging behaviour of a honey bee swarm. In ABC algorithm, each food source is exploited by only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. Half of the colony comprises employed bees and the other half includes the onlooker bees. The employed bee whose food source has been abandoned by its bee becomes a scout. In ABC algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees or the onlooker bees is equal to the number of food sources in the population.

Pseudo-code of the modified ABC algorithm proposed for solving constrained problems is given in Algorithm 1:

Algorithm 1 *(Pseudo-code of main body of ABC algorithm).*

Initialization
Evaluation
cycle = 1
repeat
  Employed Bees Phase
  Calculate Probabilities for Onlookers
  Onlooker Bees Phase
  Scout Bees Phase
  Memorize the best solution achieved so far
  cycle = cycle + 1
until cycle = Maximum Cycle Number

In the first step, ABC algorithm generates a randomly distributed initial population of $sn/2$ solutions (food source positions), where sn denotes the size of population. Each solution $\vec{x}_i$ ($i = 1, 2, \ldots, sn/2$) is a D-dimensional vector. Here, D is the number of optimization parameters. Because initialization with feasible solutions is very time consuming process and in some cases it is impossible to produce a feasible solution randomly, ABC algorithm does not consider the initial population to be feasible. In initialization phase, random values between the lower and the upper boundaries of the parameters are assigned for the parameters of solutions. $failure_i$ is the non-improvement number of the solution $\vec{x}_i$, used for the abandonment. Initialization procedure is given in Algorithm 2.

Algorithm 2 *(Initialization step of ABC algorithm).*

for $i = 1$ to $sn/2$ **do**
  for $j = 1$ to D **do**
    Generate $\vec{x}_i$ solution
$$x_i^j = x_{min}^j + rand(0,1)(x_{max}^j - x_{min}^j) \quad (2)$$
    where x_{min}^j and x_{max}^j are lower and upper bound of the parameter j, respectively.
  end for
  $failure_i = 0$
end for

After initialization, the population is evaluated and is subjected to repeated cycles of the search processes of the employed bees, the onlooker bees and scout bees. Employed bee procedure of the ABC algorithm is presented in Algorithm 3.