Guided artificial bee colony algorithm

Milan TUBA, Nebojsa BACANIN, Nadezda STANAREVIC
Faculty of Computer Science
University Megatrend Belgrade
Bulevar umetnosti 29
SERBIA
tuba@ieee.org, nbacanin@megatrend.edu.rs, srna@stanarevic.com

Abstract: Artificial bee colony algorithm is an optimization algorithm based on a particular intelligent behaviour of honeybee swarms. In this paper we present a novel algorithm named GABC which integrates artificial bee colony algorithm (ABC) with self-adaptive guidance adjusted for engineering optimization problems. The novel algorithm speeds up the convergence and improves the algorithm's exploitation. We tested our guided algorithm on four standard engineering benchmark problems. The experimental results show that GABC algorithm can outperform ABC algorithm in most of the cases.

Key-Words: - Artificial bee colony, Constrained optimization, Swarm intelligence, Metaheuristic optimization

1 Introduction

A branch of nature inspired algorithms which are called swarm intelligence is focused on insect behavior in order to develop some meta-heuristics which can mimic insect's problem solution abilities [1], [2], [3], [4]. Optimization algorithms are capable of finding optimal solutions for numerous test problems for which exact and analytical methods do not produce optimal solutions within a reasonable computational time. Their ability to provide many near-optimal solutions at the end of an optimization run enables to choose the best solution according to given criteria. The artificial bee colony (ABC) algorithm is a metaheuristic optimization technique that mimics the process of food foraging of honeybees. Originally the ABC algorithm was developed for continuous function optimization problems, but it can also be successfully applied to various optimization problems.

A majority of industrial engineering optimization problems are constrained problems. The presence of constraints significantly affects the performance of any optimization algorithm. Michalewicz and Fogel [5] describe the following characteristics that make it difficult to solve an optimization problem in the real world:

1. The number of possible solutions (search space) is too large.
2. The problem is so complicated that, with the aim of obtaining a solution, simplified models of the same problem must be used. Thus, the solution is not useful.
3. The evaluation function that describes the quality of each solution in the search space varies over time or it has noise.
4. Possible solutions are highly restricted, making it difficult even generating at least one feasible solution (i.e., satisfy the constraints of the problem).

The constrained optimization problem can be represented as the following nonlinear programming problem [6]:

$$\text{minimize } f(x),\ x=(x_1,\ \dots,\ x_n) \in R^n \tag{1}$$

where $x \in F \subset S$. The objective function f is defined on the search space $S \in R^n$ and the set $F \subset S$ defines the feasible region. Usually, the search space S is defined as an n-dimensional rectangle in R^n (domains of variables defined by their lower and upper bounds):

$$lb_i \leq x_i \leq ub_i, \quad 1 \leq i \leq n \tag{2}$$

the feasible region $F \subset S$ is defined by a set of m additional constraints:

$$g_j(x) \leq 0, \text{ for } j = 1, \dots, q$$

$$h_j(x) = 0, \text{ for } j = q + 1, \dots, m. \tag{3}$$

This research is supported by Ministry of Science, Republic of Serbia, Project No. 44006

Although the original ABC algorithm is a well-performing optimization algorithm, we have noticed that the solution search method of ABC algorithm can be improved by better guided exploration. In order to improve the exploration phase we decided to use the information of the global best solution in the process of producing new candidate solution in the scout phase and to change a random approach. It should be pointed out that the use of global best solution has also been utilized by DE and PSO algorithms [7], [8].

In this paper, we present enhancements of the artificial bee colony algorithm proposed by Karaboga and Basturk [9]. The organization of the remaining paper is as follows. Section 2 details the original ABC algorithm; Section 3 describes the basic theory of the constrained optimization. In Section 4 our modification is proposed and explained in detail. In Section 5 well-known constrained engineering problems are discussed and in Section 6 comparison experiments on the engineering optimization problems are performed to verify efficiency of our proposed approach over the traditional ABC algorithm. Our conclusions and future work are contained in the final Section 7.

2 Pure ABC algorithm

In the ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. The number of employed bees is equal to the number of food sources and an employed bee is assigned to one of the sources. In the ABC algorithm, while onlookers and employed bees carry out the exploitation process in the search space, the scouts control the exploration process. The scouts are characterized by low search costs and a low average in food source quality [9]. 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. An important difference between ABC and other swarm intelligence algorithms is that in the ABC, the solutions of the problem are represented by the food sources, not by the bees. The food source of which the nectar is abandoned by the bees is replaced with a new food source by the scouts which involves calculating a new solution at random. The employed bee of an abandoned food source becomes a scout.

An onlooker bee chooses a food source depending on the probability value associated with that food source, p_i, calculated by the following expression

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{4}$$

where fit_i is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i.

In order to produce a candidate food solution from the old one in memory, the ABC uses the following expression

$$\upsilon_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \tag{5}$$

where $k \in \{1, 2,.., SN\}$ and $j \in \{1, 2,...,D\}$ are randomly chosen indexes. If a solution cannot be improved further through a predetermined number of cycles, the food source will be abandoned. The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called "*limit*" [9]. There are three main control parameters used in the ABC: the number of food sources which is equal to the number of employed or onlooker bees (*SN*), the value of *limit*, and the maximum cycle number.

3 Constrained optimization problems

In case of problems with constraints, a desired solution must be located in the feasible space $F \subset S$ where feasibility means that the solution satisfies all the constraints. Most of the methods to solve constrained problems start with solutions that are outside of the feasible area and it is expected that, after some computational time these solutions reach the feasible area.

The first proposal to extend the ABC algorithm [9] to constrained spaces, used a constraint handling technique originally proposed for a genetic algorithm by Deb [10], [11]. Penalty function method is the most common approach in handling constraints. By adding a penalty term to the objective function, a constrained optimization problem is transformed into an unconstrained one. Based on the penalty function method, Deb has developed a constraint handling approach which does not require any penalty parameter. Deb's method uses a tournament selection operator, where two solutions are compared at a time, and the following criteria are always enforced:

1. Any feasible solution is preferred to any infeasible solution,
2. Among two feasible solutions, the one having better objective function value is preferred,
3. Among two infeasible solutions, the one having smaller constraint violation is preferred.

ISBN: 978-960-474-297-4

In order to adapt the ABC algorithm Karaboga has accepted Deb's constrained handling method instead of the selection process (greedy selection) of the ABC algorithm. Pseudo-code [1] for the ABC algorithm for constrained optimization problems is:

1. Initialize the population of solutions
2. Evaluate the population
3. cycle=1
4. repeat
5. Produce new solutions for the employed bees by using Eq. (6) and evaluate them

$$\upsilon_{i,j} = \begin{cases} x_{i,j} + \phi * (x_{i,j} - x_{k,j}), & R_j < MR \\ x_{i,j} & otherwise \end{cases} \quad (6)$$

6. Apply selection process based on Deb's method
7. Calculate the probability values $P_{i,j}$ for the solutions $x_{i,j}$ using fitness of the solutions and the constraint violations (CV) by Eq. (7)

$$pi = \begin{cases} 0.5 + \left(\frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \right) * 0.5 & if \ \ solution \ \ is \ \ feasible \\ \left(1 - \frac{CV}{\sum_{i=1}^{SN} CV} \right) * 0.5 & if \ \ solution \ \ is \ \ infeasible \end{cases} \quad (7)$$

where CV is defined by Eq. (8)

$$CV = \sum_{1}^{q} g_j(x) \ + \ \sum_{q+1}^{m} h_j(x) \quad (8)$$

8. For each onlooker bee, produce a new solution υ_i by (4) in the neighborhood of the solution selected depending on p_i and evaluate it
9. Apply selection process between υ_i and x_i based on Deb's method
10. Determine the abandoned solutions by using "limit" parameter fr the scout. If it exists, replace it with a new randomly produced solution by step 5

$$x_i^j = x_{\min}^j + rand(0,1) * (x_{\max}^j - x_{\min}^j) \quad (9)$$

11. Memorize the best solution achieved so far
12. cycle = cycle+1
13. until cycle = MCN

4 Guided ABC algorithm

It is well known that both exploration and exploitation should be well balanced in any population-based optimization algorithm [12]. In ABC algorithm, the process of replacing abandoned food source is simulated by randomly producing a new solution, as seen in Eq. (5). The new solutions in scout phase of ABC algorithm are not based on the information of previous solutions or the global best solution. In practice, we noticed that after a certain number of cycles, solutions will approach the optimum value, hence the use of random solution given by Eq. (5) will be a step backwards. Inspired by original proposal in SAVPSO [8], we modify the solution search equation by applying the global best solution to guide the search of scout in order to improve the exploration. To handle constraints, in SAVPSO authors adopt their proposed dynamic-objective constraint-handling method.

The following three characteristics of the feasible region, which can be considered as some kind of knowledge about the feasible region, are responsible for the impact on the search behaviour of the particles [8]:

1. The position of the feasible region with respect to the search space;
2. The connectivity and the shape of the feasible region;
3. The ratio $|F|/|S|$ of feasible region to the search space.

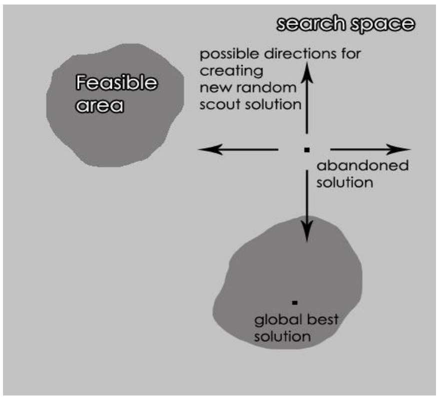
According to the characteristics above, in SAVPSO, the swarm is manipulated according to the following self-adaptive velocity equations:

$$v_{id}(t+1) = \omega |p_{i'd}(t) - p_{id}(t)| sign(v_{id}(t)) + r(p_{id}(t) - x_{id}(t)) + (1-r)(p_{gd}(t) - x_{id}(t)) \quad (10)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)$$

where $r \in U[0, 1]$, $i' \in intU[1, N]$, ω is a scaling parameter, and $sign(v_{id}(t))$ is the sign of $v_{id}(t)$. The self-adaptive velocity formula consists of three parts. The first part is velocity of the particle. The second part is the "cognitive" part which represents personal thinking of itself - learning from its own flying experience. The third part is the "social" part which represents the collaboration among particles - learning from group flying experience [8]. In the original ABC in scout phase new solution is generated by using a random approach, thus it is very difficult to generate a new solution that could be placed in the promising region of the search space. Our modified algorithm uses a different approach based on proposal utilized in [8]. Instead of generating a random solution based on Eq. (5), the scout will generate a new solution by adding the global experience information ($x_{best,j}$ - *the best global food source*) to Eq. (5). A new solution will be generated by using information about the food source that is abandoned, the best global food source and a randomly chosen food source as stated in Eq. (11):

$$\upsilon_{i,j} = x_{i,j} + \phi * (x_{i,j} - x_{k,j}) + (1-\phi) * (x_{i,j} - x_{best,j}) \quad (11)$$



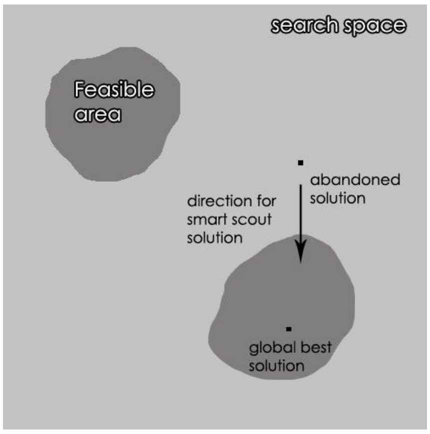


Fig. 3: Possible directions for random scout solution

The proposed modification will increase the capabilities of the ABC algorithm to produce new solutions located near the boundaries of the feasible region or if the best solution is feasible in the promising region by choosing direction based on the best global food source (Fig. 3).

5 Engineering optimization problems

In order to study the performance of solving the real-world engineering design problems, the proposed method is applied to 4 well-known constrained engineering problems: Pressure vessel, tension/compression spring, speed reducer and welded beam. The number of linear and nonlinear inequality constraints of the problems is given in Table 1. On constrained optimization problems, no single parameter (number of linear, nonlinear, active constraints, the ratio $\rho = |F| / |S|$, type of the function, number of variables) is proved to be significant as a major measure of difficulty of the problem [13].

Problem	LI	NI
Pressure vessel	3	1
Tension/comp. spring	1	3
Speed reducer	4	7
Welded beam	2	5

Table 1: Number of linear and nonlinear inequality constraints

The pressure vessel problem is to design a compressed air storage tank with a working pressure of 3000 psi and a minimum volume of 750 ft^3. A cylindrical vessel (Fig. 4) is capped at both ends by hemispherical heads. Using rolled steel plate, the shell is made in two halves that are joined by the longitudinal welds to form a cylinder. The objective is to minimize the total cost of material, forming and welding of a cylindrical vessel. The four design variables are x_1 (thickness of the shell), x_2 (thickness of the head), x_3 (inner radius R) and x_4 (length of the cylindrical section of the vessel, not including the head). x_1 and x_2 are to be in integral multiples of 0.0625 inch which are the available thicknesses of rolled steel plates. The radius x_3 and the length x_4 are continuous variables.

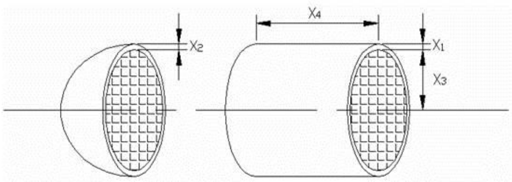


Fig. 4: Pressure vessel design

The tension/compression problem deals with minimizing of the weight of the tension/ compression spring subject to constraints on the minimum deflection, shear stress, surge frequency, diameter and design variables. This problem has a nonlinear objective function, a linear and three nonlinear inequality constraints. There are three continuous variables: the wire diameter x_1 , the mean coil diameter x_2 , and the number of active coils x_3 .

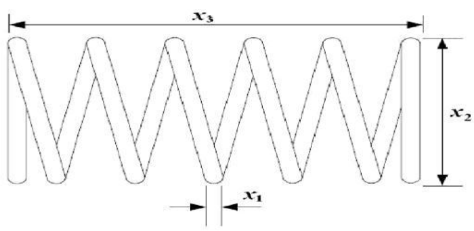


Fig. 5: Tension/compression spring

The aim of the speed reducer design is to minimize the weights of the speed reducer subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts

ISBN: 978-960-474-297-4

and stresses in the shafts. The design of the speed reducer, is considered with the face width x_1, module of teeth x_2, number of teeth on pinion x_3, length of the first shaft between bearings x_4 , length of the second shaft between bearings x_5 , diameter of the first shaft x_6 , and diameter of the first shaft x_7. All variables are continuous except x_3 that is integer. Speed reducer problem has seven nonlinear and four linear constraints.

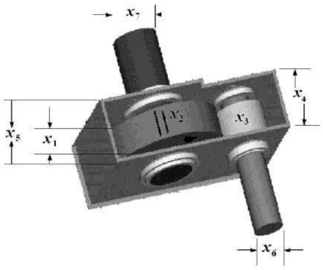


Fig.6: Speed reducer

Welded beam design problem is a standard test problem for constrained design optimization. The problem aims to minimize the cost of beam subject to constraints on shear stress, τ, bending stress in the beam, σ, buckling load on the bar, Pc, end deflection of the beam, δ, and side constraints. Welded beam design is illustrated in Fig. 7. This problem consists of a nonlinear objective function, five nonlinear and two linear inequality constraints. The solution is located on the boundaries of the feasible region.

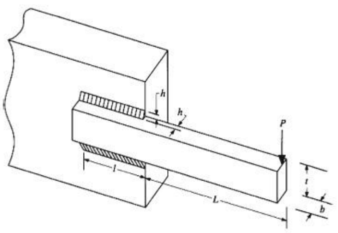


Fig. 7: Welded beam problem

6 Parameter settings and results

The performance of the GABC algorithm was compared with original ABC algorithm [14], particle swarm optimization (PSO) [15] and the evolution strategy [16]. We performed 30 independent runs per problem. Our algorithm used the same parameters' values as original ABC algorithm: Swarm Size = 30, Maximum cycle number = 1000, Modification rate = 0.9. Our results were compared with respect to the best results reported in the specialized literature. Comparisons show that GABC outperforms or performs similarly to three state-of-the-art approaches in terms of the quality of the resulting solutions – Table 2. From the results, it can be concluded that GABC algorithm is a promising ABC modification for optimizing constrained engineering problems.

7 Conclusions

A new method named GABC is introduced in this paper, which improves the performance of the ABC algorithm by incorporating self-adaptive guidance method. The approach obtains competitive results on 4 well-known constrained engineering problems. From the comparative study our modified algorithm GABC has shown its potential to handle various constrained problems and its performance is much better than original ABC algorithm, so we can conclude that this mechanism does improve the robustness of the ABC. Thus, we consider our approach to be a viable choice for solving constrained engineering optimization problems, due to its simplicity, speed and reliability.

As part of our future work, we are interested in exploring other constrained problems and in performing a more detailed statistical analysis of the performance of our proposed approach.

Problem	Stats.	PSO	($\mu + \lambda$)-ES	ABC	GABC
Pressure vessel	Best	6059.714	6059.714	6059.714	6059.714
	Mean	6289.928	6379.938	6245.308	6218.515
	St. Dev	3.1E+02	2.1E+02	2.05.E+02	1.9 E+02
Ten/comp. spring	Best	0.012	0.012	0.0126	0.0126
	Mean	0.012	0.013	0.0127	0.0127
	St. Dev	4.1E-05	3.9E-04	1.28E-4	2.8E-4
Speed reducer	Best	NA	2996.348	2997.058	2996.783
	Mean	NA	2996.348	2997.058	2996.783
	St. Dev	NA	0.000	0.000	0.000
Welded beam	Best	NA	1.724	1.724	1.724
	Mean	NA	1.777	1.741	1.763
	St. Dev	NA	0.088	0.031	0.033

Table 2. Statistical results of the PSO, ($\mu + \lambda$)-ES, ABC and GABC algorithms

ISBN: 978-960-474-297-4

References:

[1] Karaboga D, Basturk B, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, *Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, Vol. 4529/2007, No. 0302-9743, 2007, pp.789–798.

[2] N. Gomez, L. F. Mingo, J. Bobadilla, F. Serradilla, J. A. C. Manzano: Particle Swarm Optimization models applied to Neural Networks using the R language, *WSEAS Transactions on Systems*, Vol. 9, Issue 2, 2010, pp. 192-202

[3] R. M. Idris, A. Khairuddin and M.W. Mustafa, Optimal Allocation of FACTS Devices in Deregulated Electricity Market Using Bees Algorithm, *WSEAS Transactions on Power Systems*, Vol. 5, Issue 2, Apr 2010, pp. 108-119.

[4] A.V. Baterina, C. Oppus, Image Edge Detection Using Ant Colony Optimization, *WSEAS Transactions On Signal Processing*, Vol. 6, No. 2, 2010, pp.58-67.

[5] Michalewicz Z, Fogel D B, *How to Solve It: Modern Heuristics*, Springer, 2004.

[6] A. R. Hedar, M. Fukushima, Derivative-Free Filter Simulated Annealing Method for Constrained Continuous Global Optimization, *Journal of Global Optimization*, Vol. 35, ISSN: 0925-5001, 2004, pp. 521 – 549.

[7] T. Jayabarathi, S. Chalasani, Z. A. Shaik, Hybrid Differential Evolution and Particle Swarm Optimization Based Solutions to Short Term Hydro Thermal Scheduling, *WSEAS Transactions On Power Systems*, Vol. 2, No. 11, 2007, pp. 245-254

[8] Haiyan Lu, Weiqi Chen, Self-adaptive velocity particle swarm optimization for solving constrained optimization problems, *Journal of global optimization*, Vol. 41, No. 3, 2008, pp. 427-445.

[9] Karaboga D, Basturk B, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. of Global Optimization*, Vol. 39, No. 0925-5001, 2007, pp. 459-471.

[10] Deb K, An Efficient Constraint-handling Method for Genetic Algorithms, *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 0045-7825, 2000, pp. 311-338.

[11] Deb K. Optimization for Engineering Design, Algorithms and Examples 4th. Edition, chapter 1 Introduction, pages 1–30, Prentice-Hall India, 2000.

[12] R. M. Gamot, A. Mesa: Particle swarm optimization: Tabu search approach to constrained engineering optimization problems, *WSEAS Transactions on Mathematics*, Vol.7, 2008, pp. 666-675.

[13] Michalewicz, Z., Deb, K., Schmidt, M., & Stidsen, T. *Evolutionary algorithms for engineering applications, Springer*, 1997.

[14] Akay B, Karaboga D, Artificial bee colony algorithm for large-scale problems and engineering design optimization, *Journal of Intelligent Manufacturing*, DOI: 10.1007/s10845-010-0393-4, 2010, pp. 1-14

[15] Parsopoulos, K., Vrahatis, M., Unified particle swarm optimization for solving constrained engineering optimization problems, *Advances in natural computation*, Vol. 3612, 2005, pp. 582–59.

[16] Mezura-Montes, E., Coello C. A. Useful infeasible solutions in engineering optimization with evolutionary algorithms, *In Proceedings of the 4th Mexican International Conference on Artificial Intelligence*, Vol. 3789, 2005, pp. 652–662.