Artificial Bee Colony Metaheuristic to Optimize Goal Programming Engineering Design Problems

Saïma DHOUIB, Souhail DHOUIB* and Habib CHABCHOUB

*Research unit of L.O.G.I.Q.
University of Sfax, TUNISIA.
Souhail.dhouib@gmail.com

Abstract—**In this paper, an Artificial Bee Colony (ABC) metaheuristic is adapted to optimize Goal Programming (GP) Engineering Design Problems. The ABC is personalized to support the GP by means of minimizing deviations from fixed goals. This proposed metaheuristic is named Goal Programming Artificial Bee Colony (GP-ABC). By computer simulations, it is shown that the performance of the GP-ABC is better than the previous metaheuristics using very few user-defined parameters.**

Keywords-: Metaheuristic, artificial bee colony, Continuous Optimization, Engineering Design Problems.

I. INTRODUCTION

The Goal Programming (GP) model was first introduced by [6]. This model is one of the most popular and known of multiple-objective programming technique. The main idea of the GP model is to establish for every objective a level of aspiration (the goal) and then to find a solution which satisfies all the given constraints, by minimizing the sum of the deviations from targets.

Over the last few years some approaches based on metaheuristics have been proposed for the general GP model. These methods are more flexible, efficient and can be modified and/or adapted to specific problems requirements. Among research developed in this field we mention the work of [2], [3], [8], [15] and [9]. In [2], the engineering design problems are modeled as multi-objective optimization problems and then solved using the multiple objective taboo search algorithms. These types of problems are also solved as multi-objective optimization problems by [8] using the non-dominated sorting genetic algorithm (NSGA). Furthermore, [9] solve these Multi-Objective Optimization (MOO) problems by means of two local search algorithms the Record to Record Travel and the Variable Neighborhood Search. But in [3] and [15], they are solved as single-objective optimization problems using the SA algorithm and the record to record travel algorithm respectively.

As far as we know, in literature no one adapted the Artificial Bees Colony (ABC) concept originally introduced by [19] to the GP problems. This paper presents the first one that brings together research and applications of ABC metaheuristic within GP problems. Thus, the mean idea of this paper is to adapt the ABC metaheuristic to solve the GP problems. This proposed metaheuristic is named Goal Programming Artificial Bee Colony (GP-ABC).

The remainder of the paper is organized as follows. In section 2, we recall the description of the ABC metaheuristic. Section 3 details our GP-ABC implementation for the GP Engineering Design Problems. Computational results are reported and analyzed in section 4. Section 5 concludes the paper and suggests future research directions for the application of ABC to GP problems.

II. THE ARTIFICIAL BEES COLONY

A The Artificial Bees Colony (ABC) metaheuristic is an optimization algorithm proposed by [19] based on the collective intelligence of foragers bees to find food in nature. In the ABC the bees are subdivided into employed or unemployed foragers; the main difference between them is the maintain of knowledge about food source. The employed foragers know the distance, direction and profitability of a specific food source; they have three possibilities: 1) if the nectar in the food source is lower, the employed foragers abandons the food source and become unemployed bees 2) they exploit the food source 3) they share them knowledge with unemployed bees using waggle dance. After watching this dance the unemployed bees can be either onlooker or scout bees. The onlooker bees use the knowledge given by the waggle dance and try to find a food source, whereas the scout bees do not use any knowledge and attempt to found randomly a new food source.

The artificial bee systems has become one of the most successful optimization algorithms due to its successful implementation in various applications for optimization problems such as Job Shop scheduling [21], travelling salesman [17], biological simulation [1], assignment problem [18], Ride-matching [20], water resource [11] and Data Mining [5].

In general, the usage of 10% of employed bees as the mean number of scouts averaged has been shown to be a very successful strategy [12]; also for the maximum number of

978-1-4673-5549-0/13/$31.00 ©2013 IEEE

unemployed forgers, it is preferred to be equal to the number of employed foragers.

III. THE GOAL PROGRAMMING ARTIFICIAL BEE COLONY (GP-ABC)

The main components of the GP-ABC metaheuristic are summarized in Figure 1. This figure explains that this method is composed of two phases: diversification and intensification. The diversification phase is carried out by the unemployed bees (scouts and onlookers); the scouts exploit the work space randomly whereas the onlookers use knowledge given by the waggle dance to exploit the work space. The intensification phase is ensured by the employed bees which exploit the food source and exchange knowledge with waiting bees until the nectar in the food source attempts a low level.

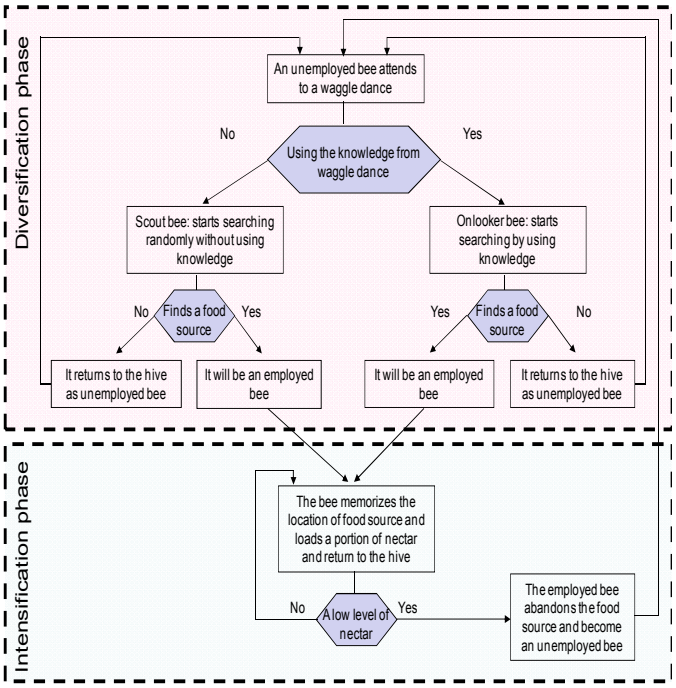

Figure 1. The main steps of GP-ABC metaheuristic

The developed GP-ABC algorithm is firstly adapted to solve any type of variables, i.e. integer, discrete and continuous variables, by using the neighbourhood moves developed by [2]. These moves are formulated as follows:

Integer variables: a move is a change of a variable from one integer value to another with a random step size: $x_i^* = x_i + integer\left[\left(2\times random() - 1\right)\times step\ i_i\right]$;

Discrete variables: a move is a change of a variable from a pre-specified discrete value to another: $x_i^* = d_{\left(l + integer\left[\left(2*random()-1\right)\times step\ d_i\right]\right)}$ if $x_i = d_l$

Continuous variables: a move is a random change of a variable: $x_i^* = x_i + \left(2\times random() - 1\right)\times step\ c_i$

Where:

x_i : Value of the ith variable prior to the neighbourhood move.

x_i^* : Value of the ith variable after the neighbourhood move.

$random()$: Random number generator, $random() \in (0,1)$.

$step\ i_i, step\ d_i, step\ c_i$: The step size for integer, discrete and continuous variables.

d_l : The lth element of the discrete variable subset X^d.

$integer[\ ]$: Function to convert a real value to integer value.

There are two parameters for which good values are difficult to determine a priori in the GP-ABC metaheuristic: the number of employed bees (*N*) and the number of maximum iterations (*Max-itera*) which define the steps size for the weight vectors (see figure 2). Typically, the size of the mean number of scouts Sc, is suggest to be 10 % of employed bees (see [13]) and the maximum number of onlooker (*Onlo*) is preferred to be equal to the number of employed foragers.

1. Initialize parameters
   a. *N*=number of employed bee
   b. *Sc*=number of *scouts*=0.1 * *N*
   c. *Max-itera*= maximum number of iterations
   d. *Step*=1/*Max_itera* (Step is used for changing the weight vectors)
2. Generate an initial employed population randomly
3. Evaluate the fitness value for the population
4. *Cycle* = 1
5. Repeat
6. *i*:=1
7. Repeat
8. Produce a new solution x_i for the employed bee by and evaluate it
9. if the fitness is higher than that of the previous one, the bee memorizes the new position and forgets the old one
10. Calculate the probability values P_i for the solution x_i by (6) and determine the number of onlooker bees which will be sent to food sources (*Onloi*=P_i* *N*)
11. Produce the new solutions for the onlookers from the solutions x_i and evaluate them
12. Select the best fitness value of onlooker bees and replace this onlooker solution with the employed bee solution only if the fitness value of this onlooker bee is better than the fitness value of employed bee
13. *i*=*i*+1
14. Until (*i*>*N*)
15. scouts bees:
    a. Generate randomly Sc solutions
    b. Archive the non dominate solutions and remove the dominated ones
    c. The *Sc* worst employed bees are respectively replaced with the scout solutions
16. *Cycle*=*Cycle*+1
17. Until (*Cycle*>*Max-itera*)

Figure 2. Detailed pseudo-code of the proposed GP-ABC algorithm

In order to perform empirical tests on parameters settings, a nonlinear test function is used. We first tested different value of *N* and *Max-itera* while setting *Sc*=10% and *Onlo*<=*N* on thirty runs of 1 second each. The numbers of employed bees were varied in size, with *N* from fifty to one hundred and *Max-*

itera from teen to sixty, both with a step length of ten. It is clear from the tests that the size of the number of employed bees should not be too small in our basic GP-ABC implementation. Good results were often found when using *N* equal either to eighty or to ninety, and with *Max-itera* close to forty. Similar results were found on the other test example2, based on which a choice was made to continue with *N*=80 and *Max-itera*=40.

IV. COMPUTATIONAL EXPERIMENTS

The proposed algorithm GP-ABC was applied in two engineering design problems in order to demonstrate the effectiveness of our method. Computational experiments show that our GP-ABC approach obtained very good results compared to the other algorithms. For the two engineering design tests, it obtains better result than the other algorithm presented in the literature.

A. Test problem P1

This example is a design optimization problem for a machine tool spindle. The problem was originally modelled by [10]. This problem is remodeled in [2] as a multiple objective optimization problem and solved it in [8] using a genetic algorithm known as MOSES. They also compared their results with four other multiple objective optimization techniques with respect to the best results obtained for each objective function. This model is also presented as a preemptive goal program and solved by the SA algorithm by [3]. The model is shown as follows:

$$lex\min\left\{z_1=\left(\delta_1^+\right)\right\}$$

Subject to:

$$\left(\frac{\pi}{4}\left[a\left(d_a^2-d_0^2\right)+l\left(d_b^2-d_0^2\right)\right]\right)-\delta_1^+=450000,$$

$$\left(\frac{Fa^3}{3EI_a}\left(1+\frac{l}{a}\frac{I_a}{I_b}\right)+\frac{F}{c_a}\left[\left(1+\frac{a}{l}\right)^2+\frac{c_a a^2}{c_b l^2}\right]\right)-\delta_2^+=0.011,$$

$$l-l_g\le 0,\ l_g-l\le 0,\ d_{a_1}-d_a\le 0,\ d_a-d_{a_2}\le 0,\ d_{b_1}-d_b\le 0,$$

$$d_b-d_{b_2}\le 0,\ d_{0m}-d_0\le 0,\ p_1d_0-d_b\le 0,\ p_2d_b-d_a\le 0,$$

$$\left|\Delta_a+\left(\Delta_a-\Delta_b\right)\frac{a}{l}\right|-\Delta\le 0,$$

$$I_a=0.049\left(d_a^4-d_0^4\right),$$

$$I_b=0.049\left(d_b^4-d_0^4\right), c_a=35400\left|\delta_{ra}\right|^{1/9}d_a^{10/9}, c_b=35400\left|\delta_{rb}\right|^{1/9}d_b^{10/9},$$

$d_0, l\ge 0,$ is continuous and d_a, d_b are discrete.

In the earlier model, d_a and d_b are discrete variables; d_a should be selected from the following set {80, 85, 90, 95}, and d_b from the set {75, 80, 85, 90}. For the spindle design problem, the GP-ABC algorithm converged to the following solution in 1s of computational time (fig. 5): $d_0=60$, $d_b=80$, $d_b=75$, $l=191.827$ with $f_1=481\ 016.849$. The following parameters are used: $step\ c_{d_0,l}=(0.15,3)$, $step\ d_{a,b}=(0.15,0.75)$.

Comparison of this result with those found by other techniques reported in [2] and [8] shows that the solution obtained from the GP-ABC algorithm is the best. These comparisons are also shown in table 1.

TABLE I. COMPARISON OF THE BEST RESULTS FOR TEST PROBLEM P1

Techniques	$f_1(x)$	PC
Monte Carlo-1	606 765.47	-
GA (binary)	494 015.44	-
GA (floating point)	1 124 409.37	-
SA	499 182.12	11 s
GP-ABC	481 016.84	2 s

Table 1 show that the obtained solution from the proposed method has a better value $f_1(x)$ than the other obtained solutions from the literature. Moreover, it dominates the solutions found by [3] using the SA algorithm and by the GA (binary) algorithm. This solution is found in less computational time (less than 2s).

B. Test problem P2

This is an engineering design problem where a beam needs to be welded on another beam and must carry a certain load. In this problem, four design parameters should be found (thickness of the beam *b*, width of the beam *t*, length of weld *l* and weld thickness *h*) for which the cost of the beam is minimum.

This welded beam design problem is taken from [8] where it is solved using NSGA. The model is presented as follow:

$$lex\min\left\{z_1=\left(\delta_1^+\right)\right\}$$

Subject to

$$1.10471h^2l+0.04811tb\left(14.0+l\right)-\delta_1^+=5$$

$$\frac{2.1952}{t^3b}-\delta_2^+=0.001$$

$$13600-\tau(x)\ge 0,\quad 30000-\sigma(x)\ge 0$$

$$b-h\ge 0,\quad P_c(x)-6000\ge 0$$

$$0.125\le h,b\le 5.0,$$

$$0.125\le l,t\le 10.0$$

978-1-4673-5549-0/13/$31.00 ©2013 IEEE

$$\tau(x) = \sqrt{\left(\tau'^2 + \tau''^2\right) + l\tau'\tau'' \Big/ \sqrt{0.25\left(l^2 + (h+t)^2\right)}}$$

$$\tau' = \frac{6000}{\sqrt{2}hl}, \quad \tau'' = \frac{6000(14+0.5l)\sqrt{0.25\left(l^2+(h+t)^2\right)}}{2\left\{0.707hl\left(l^2/12 + 0.25(h+t)^2\right)\right\}}$$

$$\sigma(x) = \frac{504000}{t^2 b},$$

$$P_c(x) = 64746.022(1 - 0.0282346t)tb^3$$

The GP-ABC algorithm found this solution: $h = 0.521$, $b = 0.797$, $t = 5.16109$, $l = 5.13$, with $f_1 = 0.000909$. This solution is found in less computational time (4 s).

Table 2 shows that this solution is better than the solution of [8]. The obtained value f_1 is nearly equal to the fixed goal and it is 99.78 % better than [8] solution.

TABLE II. COMPARISON OF THE BEST RESULTS FOR TEST PROBLEM P2

Techniques	$f_1(x)$	PC
NSGA	0. 7	-
GP-ABC	0.000909	4 s

V. CONCLUSION

In this paper, we have proposed an ABC metaheuristic, namely GP-ABC, to optimize Goal Programming Engineering Design Problems. As far as we knew, this is the first time that ABC is applied to this kind of problems.

The efficiency of the proposed approach is demonstrated by solving two engineering design optimization problems. The results are encouraging to apply the GP-ABC approach to find Pareto non dominated solutions set for multi-criteria engineering design problems.

REFERENCES

[1] Andrew M. Gregson A., Hart A., Holcombe M. and Francis Ratnieks L. (2003): 'Partial nectar loads as a cause of multiple nectar transfer in the honeybee (Apis mellifera): a simulation model', in Journal of Theoretical Biology, vol. 222, no. 1, pp 1-8.

[2] Baykasoğlu A. (2001): 'Goal Programming using Multiple Objective Tabu Search' in Journal of Operational Research Society, vol 52, pp 1359-1369.

[3] Baykasoğlu A. (2005): 'Preemptive goal programming using simulated annealing', in Engineering Optimization, Vol. 37 (1), pp. 49-63.

[4] Baykasoğlu A., Owen S., and Gindy N. (1999): 'Solution of Goal Programming Models using a Basic Taboo Search Algorithm' Journal of Operational Research Society, vol 50, pp 960-973.

[5] Benatchba K., Admane L. and Koudil M. (2005): 'Using Bees to Solve a Data-Mining Problem Expressed as a Max-Sat One', in Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approch, Lecture Notes in Computer Science, vol. 3562, pp 75-86.

[6] Charnes A., Cooper W. and Ferguson R. (1955): 'Optimal estimation of executive compensation by linear programming', in Management Science, vol 1, pp 138 – 151.

[7] Coello C. and Christiansen A. (1999): 'MOSES: a multiple objective optimization tool for engineering design', in Engineering Optimization, vol 31, pp 337–368.

[8] Deb K. (2001): 'Non-linear Goal Programming using Multi- objective Genetic Algorithms', in Journal of Operational Research Society, vol. 52, pp 291-302.

[9] Dhouib S., Kharrat A. et Chabchoub H., (2010): 'A Multi-start threshold accepting algorithm for multiple objective continuous optimization problems', in International Journal for Numerical Methods in Engineering, vol. 83, no. 11, pp. 1498-1517.

[10] Eschenauer H., Koski J. and Osyczka A. (1990): 'Multicriteria Design Optimization', in Springer-Verlag: Berlin.

[11] Haddad O., Afshar A. and Mariño M.(2006):'Honey-Bees Mating Optimization (HBMO) Algorithm: A New Heuristic Approach for Water Resources Optimization', in Water Ressources Management, vol. 20, no. 5, pp 661-680.

[12] Karaboga D. (2005): 'An Idea Based On Honey Bee Swarm For Numerical Optimization', in Technical Report-TR06, Erciyes University, in Engineering Faculty, Computer Engineering Department.

[13] Karaboga D. and Akay B. (2009): 'A comparative study of Artificial Bee Colony algorithm', in Applied Mathematics and Computation, vol. 214, pp 108–132.

[14] Karaboga D. and Basturk B. (2008): 'On the performance of artificial bee colony (ABC) algorithm', in Applied Soft Computing, vol 8, pp 687–697.

[15] Kharrat A., Dhouib S. et Chabchoub H. (2010): 'Adaptive record-to-record travel method to solve lexicographic goal programming models', International Journal of Information and Decision Sciences, vol. 2, no. 2, pp. 147 - 169.

[16] Koudil M., Benatchba K., Tarabet A. and Sahraoui B. (2007): 'Using artificial bees to solve partitioning and scheduling problems in codesign', in Applied Mathematics and Computation, vol. 186, pp 1710–1722.

[17] Marinakis Y., Marinaki M. and Dounias G. (2011): 'Honey bees mating optimization algorithm for the Euclidean traveling salesman problem', in Information Sciences, vol. 181, no. 20, pp 4684-4698.

[18] Özbakir L., Baykasoğlu A. and Tapkan P. (2010): 'Bees algorithm for generalized assignment problem", in Applied Mathematics and Computation, vol. 215, no. 11, pp 3782-3795.

[19] Pham D., Ghanbarzadeh A., Koç E., Otri S., Rahim S. and Zaidi M.(2006): ' The Bees Algorithm — A Novel Tool for Complex Optimisation Problems', in Intelligent Production Machines and Systems, pp 454-459.

[20] Teodorović D. and Orco M. (2008): 'Mitigating Traffic Congestion: Solving the Ride-Matching Problem by Bee Colony Optimization', in Transportation Planning and Technology, vol. 31, no. 2, pp 135-152.

[21] Zhang R., Song S. and Wu C. (2012): 'A hybrid artificial bee colony algorithm for the job shop scheduling problem', in International Journal of Production Economics, In Press.

[22]