EXPLORING PROMISING REGIONS OF THE SEARCH SPACE WITH THE SCOUT BEE IN THE ARTIFICIAL BEE COLONY FOR CONSTRAINED OPTIMIZATION

EFRÉN MEZURA-MONTES
Laboratorio Nacional de Informática Avanzada (LANIA A.C.)
Rébsamen 80, Centro, Xalapa
Veracruz, 91000 México
emezura@lania.mx

OMAR CETINA-DOMÍNGUEZ
Laboratorio Nacional de Informática Avanzada (LANIA A.C.)
Rébsamen 80, Centro, Xalapa
Veracruz, 91000 México
omarcetina@hotmail.com

ABSTRACT
In this paper we introduce a novel swarm intelligence approach, based in the artificial bee colony optimization algorithm (ABC), now designed specifically to solve constrained numerical optimization problems: The scout-behavior modified artificial bee colony (SM-ABC) algorithm. In SM-ABC, the behavior of the scout bee is modified as to get the capability to exploit the vicinity of the current best solution (food source). Also, the way to control the tolerance for equality constraints is altered. SM-ABC looks to improve the capabilities of ABC to find good solutions in problems with a high dimensionality and active constraints. The performance of SM-ABC is tested in 13 well-known benchmark problems found in the literature. A comparison is performed between the published results of the original ABC algorithm and SM-ABC. Finally, a performance comparison against algorithms from the state-of-the-art in bio-inspired constrained optimization is shown. The results suggest that SM-ABC is a promising heuristic to solve numerical constrained optimization problems.

1. INTRODUCTION

In the mid 1960's, the main source of inspiration taken from nature to generate heuristic-based approaches was natural evolution and its associated processes. From those research efforts, the field of evolutionary computing was defined (Eiben, 2003). In the mid 1990's, another natural process was taken as a basis to generate competitive heuristics to solve complex search problems: the social behavior among simple living organisms (insects, birds, fish). This area is called Swarm Intelligence (Kennedy, 2001) and its two original paradigms were the ant colony optimization, ACO (Dorigo and Stutzle, 2004) and particle swarm optimization, PSO (Kennedy et al., 2001). Nonetheless, other approaches have been recently proposed, such as the Artificial Bee Colony, ABC (Karaboga & Basturk, 2007a).

ABC was originally proposed to solve numerical optimization problems. However, to the best of the authors' knowledge, there is only one version adapted to deal with constrained search spaces i.e. the nonlinear programming problem (Karaboga & Basturk, 2007b). This was the main motivation to propose an improved version with mechanisms designed to locate the feasible global optimum mostly in problems where there is a high dimensionality and where the best solution lies in the boundaries of the feasible region.

The paper is organized as follows: Section 2 presents the problem of interest. Section 3 describes the original ABC algorithm while Section 4 shows the improved ABC. The experimental design and the obtained results are included in Section 5. Finally, the conclusions and future work are summarized in Section 6.

2. STATEMENT OF THE PROBLEM

The problem of interest in this paper is the general nonlinear programming problem, in which the objective, without loss of generality, is to find $\vec{x}$ which minimizes $f(\vec{x})$, subject to: $g_i(\vec{x}) \leq 0, i = 1, \dots, m$, and $h_j(\vec{x}) = 0, j = 1, \dots, p$ where $\vec{x}$ is the vector of solutions $\vec{x} = [x_1, x_2, \dots, x_n]^T$, and each $x_i, i = 1, \dots, n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$. These limits define the search space of the problem; m is the number of inequality constraints and p is the number of equality constraints which could be, like the objective function, linear or nonlinear. An inequality constraint is active in the optimum if it has a value of zero in this point ($g_i(\vec{x}) = 0$), i.e. the solution is located in the boundaries of the feasible region for the constraint i. Therefore, all equality constraints are active in the optimum. If we denote with $\mathcal{F}$ to the feasible region and with S to the whole search space, then it should be clear that $\mathcal{F} \subseteq S$. In order to solve problems with equality constraints, they are transformed into inequality constraints by using a small tolerance ε as follows: $|h_j(\vec{x})| - \varepsilon \leq 0$.

3. ARTIFICIAL BEE COLONY

The process of searching for nectar in flowers by honeybees has been seen as an optimization process. The way these social insects are able to focus efforts on areas with considerable amounts of high-quality food sources, has been modeled as an optimization heuristic. The biological model of gathering food in honeybees consists of the following minimum elements detailed below:

1. **Food sources**: The value of a food source depends on many factors, such as the proximity to the hive, the concentration of food and how easy is to extract it. For simplicity it is possible to represent the profitability of a source in a single numerical value, its fitness.
2. **Employed bees**: These bees are associated with a particular food source which is exploited by them. Each employed bee shares information of its food source, such as its location and profitability, to other collector bees.
3. **Unemployed bees**: They are constantly looking for a food source to exploit. There are two types: the scout, which searches in the vicinity of the hive for new food sources, and the onlooker, which waits in the hive and chooses a food source based on the information shared by an employed bee.

The social element in this model is given when the information on food sources is shared within the hive by an employed bee in a waggle dance, whose length indicates the profitability of the source, the angle with respect to the sun indicates the food source location and the number of *zigzag* movements during the dance shows the distance to the source (Karaboga & Basturk, 2007a). As the dances of the most profitable sources have a longer duration, they are more likely to be observed by onlooker bees, increasing the probability of those food sources to be chosen by these unemployed bees.

Based on the aforementioned biological model, the ABC algorithm is also composed of three groups of bees: employed bees, onlooker bees and scout bees. The number of employed bees is equal to the number of food sources and an employed bee is assigned to one of the sources. The food sources are the solutions of the optimization problem and the bees are the variation operators. Upon reaching the source, the bee will calculate a new solution (fly to another nearby food source) from it and retain the best one (in a greedy selection). The number of onlooker bees is also equal to the number of employed bees and they are allocated to a food source based on their profitability. Like the employed bees, they calculate a new solution from its food source. The operator used by both, the employed and the onlooker bee is detailed in Eq. 1:

$$v_{ij} = x_{ij} + \Phi_{\mathrm{ij}}\left(x_{i,j} - x_{k,j}\right), \qquad i,k = 1,\dots,SN \text{ and } j = 1,\dots,n \tag{1}$$

where v_i is the new food source generated by using both, the current food source x_i and a randomly chosen food source x_k from the population and $-1 \le \Phi_{\mathrm{ij}} \le 1$ (generated at random every time it is used) determines the stepsize of the movement. Both, i and j are generated at random but $k \neq i$. When a source does not improve after a certain number of iterations, it is abandoned and replaced by one found by a scout bee, which involves the generation of a new solution at random. Three input parameters are required by the ABC algorithm: The number of food sources (SN), the maximum number of cycles (MCN) and the Limit, which is the number of cycles a food source which has not been improved will be kept before being replaced by a new one generated by the scout bee. The pseudocode of the ABC algorithm (Karaboga & Basturk, 2007a) is detailed in Figure 1.

```
Begin
  Initialize the population of food sources x_i, i = 1,…,SN
  Evaluate each food source x_i, i = 1,…,SN
  cycle = 1
  Repeat
    For each food source x_i in the population
        Generate a new food source v_i by its corresponding employed bee (Eq. 1)
        Evaluate v_i
        Keep the best solution between x_i and v_i
    End
    Select, based on fitness proportional selection, the food sources to be visited by onlooker bees
    For each food source x_i chosen by an onlooker bee
        Generate a new food source v_i by its corresponding onlooker bee (Eq. 1)
        Evaluate v_i
        Keep the best solution between x_i and v_i
    End
    Use the scout bee to replace those abandoned food sources
    Save in memory the best food source so far
    cycle = cycle + 1
  Until cycle = MCN
End
```

Figure 1. Pseudocode of the ABC algorithm.

4. SCOUT MODIFIED ABC

Recalling from Section 1, an adapted ABC was proposed to deal with constrained search spaces by adding a recombination –like operator based on a parameter MR (see Eq. 2)

$$v_{ij} = \begin{cases} x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}), & \text{if } R_j < MR \\ x_{ij} \quad , & \text{otherwise} \end{cases} \tag{2}$$

(where $0<R_j\leq 1$ is a random number) and a modification in the selection criterion (Karaboga & Basturk, 2007b), which originally considered only the objective function value, by now using the three criteria proposed by Deb (2000): (1) The food source with the better objective function value is preferred between two feasible ones, (2) A feasible food source is preferred over an infeasible food source and (3) the food source with the lowest sum of constraint violation is preferred between two infeasible ones. Nonetheless, no further changes were considered as to modify the algorithm in order to improve the performance in this type of problems. Therefore, two new mechanisms are proposed in this paper:

1. It was observed that the scout bee ability to help the algorithm to escape from local optimum solutions is not useful at all in constrained spaces because a randomly generated food source (most of the time an infeasible one) has a very low probability of being an attractor to a promising region in a continuous space. Therefore, a modified scout mechanism, based on an original proposal utilized in PSO (Lu & Chen, 2008) is proposed now for ABC. Instead of generating a random food source, the scout bee will use this food source, subject to be replaced, as a base to generate a new search direction biased by the best food source so far and a randomly chosen food source, as indicated in Eq. 3:

   $$x'_{ij} = x_{ij} + \Phi_{ij}(x_{kj} - x_{ij}) + (1 - \Phi_{ij})(x_{bj} - x_{ij}) \quad j = 1, \dots, n \tag{3}$$

   Where x'_i is the new food source generated by the scout bee, x_i is the food source to be replaced, x_k is a randomly chosen food source, $x_b$is the best source food source in the population and $\Phi_{ij} \in [-1,1]$. This mechanism allows the ABC to replace a solution that has not been improved in several cycles with one that is in the vicinity of the current best solution. If the best solution is in the feasible region of the search space, the solutions generated by this mechanism will not very far from that area (i.e. they may be located in the boundaries of the feasible region).
2. The tolerance for equality constraints is handled by a dynamic process originally proposed by Hamida and Schoenauer (2002). In this way, early in the process the equality constraints will be easier to satisfy while later in the search, the tolerance will be smaller, forcing the algorithm to find solutions within the real feasible region. The expression used is detailed in Eq. 4:

   $$\varepsilon(t+1) = \frac{\varepsilon(t)}{dec} \tag{4}$$

   where $\varepsilon(t+1)$ and $\varepsilon(t)$ are the new and the current tolerance values, respectively and *dec* is a decreasing factor.

The complete pseudocode of the Scout Modified ABC (SM-ABC) is shown in Figure 2.

```
Begin
    Initialize the population of food sources x_i, i = 1,…,SN
    Evaluate each food source x_i, i = 1,…,SN
    If there is an equality constraint Then ϵ = 1.0
    cycle = 1
    Repeat
      For each food source x_i in the population
          Generate a new food source v_i by its corresponding employed bee (Eq. 2)
          Evaluate v_i
          Keep the best solution between x_i and v_i
      End
      Select, based on binary tournaments, the food sources to be visited by the onlooker bees
      For each food source x_i chosen by an onlooker bee
          Generate a new food source v_i by its corresponding onlooker bee (Eq. 2)
          Evaluate v_i
          Keep the best solution between x_i and v_i
      End
      Use the scout bee to replace those abandoned food sources (Eq. 3)
      Save in memory the best food source so far
      If there is an equality constraint and ϵ > 0.0001 Then reduce ϵ (Eq. 4)
     The constraints of food sources are recalculated using the new tolerance.
      cycle = cycle + 1
    Until cycle = MCN
End
```

Figure 2. Pseudocode of the SM-ABC algorithm.

5. EXPERIMENTAL STUDY

Two experiments were performed in order to (1) determine the performance of the SM-ABC with respect to the original ABC version for constrained optimization and (2) a comparison with respect to state-of-the-art approaches. 13 well-known test problems were used in the experiments. The details of each problem can be found in (Liang et al., 2006) and a summary of their features is included in Table 1.

Table 1. Main features of the test problems (n is the dimensionality of the problem, ρ is the estimated size of the feasible region with respect to the search space, LI and NI are linear and nonlinear inequality constraints, respectively and LE and NE are linear and nonlinear equality constraints, respectively and a is the number of active constraints).

Function	*n*	Type of function	ρ	LI	NI	LE	NE	*a*
g01	13	quadratic	0.0003%	9	0	0	0	6
g02	20	nonlinear	99.9973%	0	2	0	0	1
g03	10	nonlinear	0.0026%	0	0	0	1	1
g04	5	quadratic	27.0079%	0	6	0	0	2
g05	4	nonlinear	0.0000%	2	0	0	3	3
g06	2	nonlinear	0.0057%	0	2	0	0	2
g07	10	quadratic	0.0000%	3	5	0	0	6
g08	2	nonlinear	0.8581%	0	2	0	0	0
g09	7	nonlinear	0.5199%	0	4	0	0	2
g10	8	linear	0.0020%	3	3	0	0	6
g11	2	quadratic	0.0973%	0	0	0	1	1
g12	3	quadratic	4.7697%	0	1	0	0	0
g13	5	nonlinear	0.0000%	0	0	1	3	3

30 independent runs were performed by SM-ABC in each test problem by using the same set of parameters, defined as follows: SN=20, MCN=5800, Limit=MCN/(2*SN)=145 and

MR=0.8. The initial value for the tolerance of equality constraints was set to 1 and the final tolerance was set to $1E-4$ and *dec*=1.002. To deal with invalid variable values outside the boundaries of each one of them, the mechanism proposed by Kukkonen and Lampinen (2006) was utilized.

Table 2. Comparison of results of SM-ABC and ABC (Karaboga & Basturk, 2007b) on 13 benchmark functions. A result in boldface indicates either a better result or that the global optimum (or best known solutions) was reached.

Funcion/ Optimal		Methods	
		ABC	**SM-ABC**
g01 -15	Best	**-15**	**-15**
	Mean	**-15**	**-15**
	Std. Dev	**0.00E+00**	**0.00E+00**
g02 -0.803619	Best	-0.803598	**-0.803615**
	Mean	-0.792412	**-0.799336**
	Std. Dev	1.20E-02	**6.84E-03**
g03 -1	Best	**-1**	**-1**
	Mean	**-1**	**-1**
	Std. Dev	**0.00E+00**	4.68E-05
g04 -30665.539	Best	**-30665.539**	**-30665.539**
	Mean	**-30665.539**	**-30665.539**
	Std. Dev	**0.00E+00**	2.22E-11
g05 5126.498	Best	**5126.484**	5126.736
	Mean	5185.714	**5178.139**
	Std. Dev	7.54E+01	**5.61E+01**
g06 -6961.814	Best	**-6961.814**	**-6961.814**
	Mean	-6961.813	**-6961.814**
	Std. Dev	2.00E-03	**0.00E+00**
g07 24.306	Best	24.33	**24.315**
	Mean	24.473	**24.415**
	Std. Dev	1.86E-01	**1.24E-01**
g08 -0.095825	Best	**-0.095825**	**-0.095825**
	Mean	**-0.095825**	**-0.095825**
	Std. Dev	**0.00E+00**	4.23E-17
g09 680.63	Best	680.634	**680.631**
	Mean	**680.64**	680.647
	Std. Dev	**4.00E-03**	1.55E-02
g10 7049.248	Best	7053.904	**7051.706**
	Mean	**7224.407**	7233.882
	Std. Dev	1.34E+02	**1.10E+02**
g11 0.75	Best	**0.75**	**0.75**
	Mean	**0.75**	**0.75**
	Std. Dev	**0.00E+00**	2.30E-05
g12 -1	Best	**-1**	**-1**
	Mean	**-1**	**-1**
	Std. Dev	**0.00E+00**	**0.00E+00**
g13 0.05395	Best	0.76	**0.053985**
	Mean	0.968	**0.158552**
	Std. Dev	**5.50E-02**	1.73E-01

The comparison of those statistical results reported in (Karaboga & Basturk, 2007b) by the original ABC and the corresponding results obtained by SM-ABC is presented in Table 2. The results suggest that SM-ABC is able to keep or even improve the performance of the original ABC, mostly in problem g02 (twenty variables and large feasible region), problem g05 and g13 (four and five variables, respectively, very small feasible region and equality constraints), problem g07 (10 variables, six active constraints and a small feasible region) and problem g10 (eight variables and six active constraints). As a result, the modified scout mechanism provides a competitive performance in those problems with sources of difficulty (high dimensionality and the presence of equality and active inequality constraints). Regarding the comparison with three state-of-the-art nature-inspired algorithms: The self-adaptive parameter-free penalty function (SAPF-GA) by Tessema & Yen (2006), the multi-objective concepts added to an evolutionary algorithm (HCOEA) by Wang et al. (2007) and the trade-off models based on feasibility (ATMES) by Wang et al., (2008), a summary of results is presented in Table 3, where those obtained by SM-ABC are included. It is clear that the proposed approach is highly competitive with respect to these three methods.

Table 3: Comparison of results of SM-ABC with respect to state of the art algorithms: SAPF-GA), HCOEA and ATMES. A result in boldface indicates a better result or the global optimum (or best known solutions) was reached.

Function/ Optimal		Methods			
		SAPF-GA	HCOEA	ATMES	SM-ABC
g01 -15	B	**-15**	**-15**	**-15**	**-15**
	M	-14.552	**-15**	**-15**	**-15**
	SD	7.00E-01	4.30E-07	1.60E-14	**0.00E-00**
g02 -0.803619	B	-0.803202	-0.803241	-0.803388	**-0.803615**
	M	-0.755798	**-0.801258**	-0.790148	-0.799336
	SD	1.33E-01	**3.83E-03**	1.30E-02	6.84E-03
g03 -1	B	**-1**	**-1**	**-1**	**-1**
	M	-0.964	**-1**	**-1**	**-1**
	SD	3.01E-01	**1.30E-12**	5.90E-05	4.68E-05
g04 -30665.539	B	-30665.401	**-30665.539**	**-30665.539**	**-30665.539**
	M	306659.221	**-30665.539**	**-30665.539**	**-30665.539**
	SD	2.04E+00	5.40E-07	**7.40E-12**	2.22E-11
g05 5126.498	B	5126.907	5216.4981	**5126.498**	5126.736
	M	5214.232	**5216.4981**	5127.648	5178.139
	SD	2.47E+02	**1.73E-07**	1.80E+00	5.61E+01
g06 -6961.814	B	-6961.046	**-6961.81388**	**-6961.814**	**-6961.814**
	M	-6953.061	**-6961.81388**	**-6961.814**	**-6961.814**
	SD	5.88E+00	8.51E-12	4.60E-12	**0.00E+00**
g07 24.306	B	24.838	24.3064582	**24.306**	24.315
	M	27.328	**24.3073989**	24.316	24.415
	SD	2.17E+00	**7.12E-04**	1.10E-02	1.24E-01
g08 -0.095825	B	**-0.095825**	**-0.09285**	**-0.095825**	**-0.095825**
	M	-0.095635	**-0.09285**	**-0.095825**	**-0.095825**
	SD	1.06E-03	**2.42E-17**	2.80E-17	4.23E-17
g09 680.63	B	680.773	680.630057	**680.63**	680.631
	M	681.246	**680.630057**	680.639	680.647
	SD	3.22E-01	**9.41E-08**	1.00E-02	1.55E-02
g10 7049.248	B	7069.981	**7049.2866**	7052.253	7051.706
	M	7238.964	**7049.52544**	7250.437	7233.882
	SD	1.38E+02	**1.500E-01**	1.20E+02	1.10E+02
g11 0.75	B	**0.749**	**0.75**	**0.75**	**0.75**
	M	0.751	**0.75**	**0.75**	**0.75**
	SD	2.00E-03	**1.55E-12**	3.40E-04	2.30E-05
g12 -1	B	**-1**	**-1**	**-1**	**-1**
	M	-0.99994	**-1**	**-1**	**-1**
	SD	1.41E-01	**0.00E+00**	1.00E-03	**0.00E+00**
g13 0.05395	B	**0.053941**	0.0539498	0.05395	0.053985
	M	0.28627	**0.0539498**	0.053959	0.158552
	SD	2.75E-01	**8.68E-08**	1.30E-05	1.73E-01

6. CONCLUSIONS AND FUTURE WORK

A novel adaptation of the artificial bee colony algorithm (ABC) to solve constrained numerical optimization problems was proposed in this paper. Instead of generating a food source at random, the scout bee was designed to generate a food source influenced by the best food source so far and also by a randomly chosen food source. In this way, the scout bee will help the search to locate feasible solutions close to those already included in the population. Furthermore, the tolerance for equality constraints was handled by a dynamic mechanism in order to help the search to gradually locate the feasible region of the search space. The statistical results obtained in two experiments suggested that (1) SM-ABC improved the performance of the original ABC, mostly in those test problems where different sources of difficulty were found (high dimensionality and active constraints) and (2) the overall performance of SM-ABC is highly competitive with respect to those reported by three state-of-the-art algorithms. Part of the future work includes additional tests on the comparison of ABC and SM-ABC in order to know the different behaviors more in-depth and to test SM-ABC in other problems with equality constraints.

Acknowledgements – Both authors acknowledge support from CONACyT through project 79809-Y. The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at LANIA.

REFERENCES

Deb, K., 2000, An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering, 186(2/4):311-338.

Dorigo, M. and Stützle, T., 2004, Ant Colony Optimization, Bradford Books, MIT Press, 2004.

Eiben, A.E. and J.E. Smith, 2003, Introduction to Evolutionary Computing. Springer-Verlag.

Hamida, S.B., and Schoenauer, M., 2002, ASCHEA: New results using adaptive segregational constraint handling, Proceedings of the CEC, Vol. 1, pages: 884-889, IEEE Service Center.

Karaboga, D. and Basturk, B., 2007a, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3):459–471.

Karaboga, D. and Basturk, B., 2007b, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. *LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, 4529:789–798.

Kennedy, J, Eberhart, R. and Shi, Y., 2001, Swarm Intelligence, Morgan-Kaufmann Publishers.

Kukkonen, S., and Lampinen, J., 2006, Constrained real-parameter optimization with generalized differential evolution., In Proceedings *of the IEEE Congress on Evolutionary Computation*, IEEE, pp. 911-918.

Liang, J.J., Runarsson, T.P., Mezura-Montes E., Clerc, M., Suganthan, P.N., Coello Coello, C.A. and Deb, K., 2006, Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization, Technical Report, September, available at: http://www.lania.mx/~emezura/documentos/tr_cec06.pdf.

Lu, H. and Chen W., 2008, Self-adaptative velocity particle swarm optimization for solving constrained optimization problems. *Journal of Global Optimization*, 41:427–445.

Tessema, B. and Yen, G.G, 2006, A self adaptive penalty function based algorithm for constrained optimization. In *Proceedings of IEEE Congress on Evolutionary Computation 2006*, pages 950-957, Vancouver, BC, Canada, IEEE Service Center.

Wang, Y., Cai, Z., Guo, G. and Zhou, Y., 2007, Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(3):560 – 575.

Wang, Y., Cai, Z., Zhou, Y., and Zeng, W., 2008, An adaptive tradeoff model for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 12(1):80–92.