MODIFIED BACTERIAL FORAGING OPTIMIZATION FOR ENGINEERING DESIGN

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

BETANIA HERNÁNDEZ- OCAÑA
Univ. Juárez Autónoma de Tabasco
División Académica
de Informática y Sistemas
Km. 1Carretera
Cunduacán-Jalpa de Méndez
betania_h_o@hotmail.com

ABSTRACT

We present an adaptation of the bacterial foraging optimization algorithm (inspired on bacteria moving in their environment looking for high-nutrient areas) to solve engineering design problems. This proposal simplifies the original algorithm, proposed for unconstrained optimization, as to adapt it to solve constrained problems in numerical search spaces. The modifications look to decrease the number of parameters used in the algorithm, adding a constraint-handling mechanism and improving the communication capabilities among bacteria. The approach is tested on some well-known engineering design problems and its performance is compared against state-of-the-art algorithms. Based on the obtained results, some conclusions are established and the future work is defined.

1. INTRODUCTION

During the last years, the number of nature-inspired heuristics to sample complex search spaces, such as optimization problems, has significantly increased. Besides evolutionary algorithms, EAs (Eiben & Smith, 2003), the addition of swarm intelligence, SI (Kennedy et al., 2001) has enriched the area. Within SI, where the initial paradigms were particle swarm optimization, PSO (Kennedy et al., 2001) and the ant colony optimization, ACO (Dorigo & Stützle, 2004), there are novel models which are gaining popularity among researchers and practitioners. This is the case of the bacterial foraging optimization algorithm (BFOA), inspired in the behavior of bacterium E. Coli in its search for food. This approach, proposed by Passino (2002) considers three steps: (1) Chemotaxis, (2) reproduction and (3) elimination-dispersal of bacteria. BFOA has been successfully applied to solve different type of problems such as the identification of nonlinear dynamic systems (Majhi, 2007). Furthermore, BFOA has been combined with other algorithms to solve multimodal optimization problems (Biswas, 2007). Based on the literature review (brief in the paper by space restrictions), the research efforts on BFOA to solve constrained numerical optimization problems are scarce. This is the main motivation of this work. We aim to explore the capabilities of BFOA when solving numerical constrained optimization problems. To achieve this objective, we propose the following modifications to the original BFOA: (1) A single loop to include the chemotactic, reproduction and elimination-dispersal steps, (2) a definition of the stepsize values based on the features of the problem, (3) a constraint-handling mechanism and (4) a simple communication mechanism among bacteria to allow them to move towards promising regions of the search space.

The paper is organized as follows: The problem of interest is defined in Section 2, while the original BFOA is described in Section 3. In Section 4 we include the explanation of each one of the four modifications proposed in order to define the modified BFOA (MBFOA). The experimental design and the obtained results are detailed in Section 5. Finally, in Section 6 we summarize our findings and future work.

2. STATEMENT OF THE PROBLEM

Some engineering design problems can be stated as nonlinear optimization problems (NOPs) in which the goal is to, without loss of generality, find $\vec{x}$ which minimizes $f(\vec{x})$, subject to: $g_i(\vec{x}) \leq 0, i = 1, \dots, m$, where $\vec{x}$ is the vector of solutions $\vec{x} = [x_{1,}x_2, \dots, x_n]^T$, where 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 which could be, like the objective function, linear or nonlinear. If we denote with $\mathcal{F}$ to the feasible region and with S to the whole search space, then it should be clear that $\mathcal{F} \subseteq S$.

3. BACTERIAL FORAGING OPTIMIZATION ALGORITHM

BFOA, as other SI algorithms, is based on some social and cooperative behaviors found in nature i.e. the way bacteria look for regions of high levels of nutrients. This task has been seen as an optimization process. The first attempt to model this idea was presented by Bremermann (1974) and extended later by Passino (2002). Each bacterium tries (1) to maximize its obtained energy per unit of time expended on the foraging process and (2) to avoid noxious substances. Besides, SI assumes communication among individuals. The biological swarm of bacteria, during a certain period of time, behaves as follows (Passino, 2002):

1) Bacteria are randomly distributed in the map of nutrients.
2) Bacteria move towards high-nutrient regions in the map. Those located in regions with noxious substances or low-nutrient regions will die and disperse, respectively. Bacteria in convenient regions will be able to reproduce.
3) Bacteria are located in promising regions within the map of nutrients and are able to communicate via attractant and repellant substances segregated by each bacterium
4) Bacteria are now located in the highest-nutrient region.
5) Bacteria now disperse as to look for new nutrient regions in the map.

Based on these steps, Passino (2002) proposed the bacterial foraging optimization algorithm which is summarized in Figure 1. The chemotactic step was modeled by Passino with the generation of a random search direction (Eq. 1)

$$\emptyset(i) = \frac{\Delta(i)}{\sqrt{\Delta(i)^T\Delta(i)}} \qquad (1)$$

where $\Delta(i)^n$ is a n-dimensional randomly generated vector with elements within the following interval: [-1,1]. After that, each bacterium $\theta^i(j,k,l)$ (where j, k and l are the chemotactic, reproduction and elimination-dispersal counters, respectively) modifies its position as indicated in Eq. 2, where $C(i)$ is the stepsize for search direction $\emptyset(i)$.

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\emptyset(i) \qquad (2)$$

Equation 1 represents a tumble (generation of a search direction) and Eq. 2 represents a swim (movement of a bacterium). The swim will be repeated N_s times if the new position is better than the previous one i.e. $f\left(\theta^i(j+1,k,l)\right) < f\left(\theta^i(j,k,l)\right)$.

```
BEGIN
  Initialize input parameters (see the caption of this figure)
  Create a random initial swarm of bacteria $\theta^i(j,k,l)\ \forall i, i = 1, \ldots, S_b$
  Evaluate $f\left(\theta^i(j,k,l)\right) \forall_i, i = 1, \ldots, S_b$
  $\boldsymbol{FOR}\ l = 1\ to\ N_{ed}\ \boldsymbol{DO}$
      $\boldsymbol{FOR}\ k = 1\ to\ N_{re}\ \boldsymbol{DO}$
          $\boldsymbol{FOR}\ j = 1\ to\ N_c\ \boldsymbol{DO}$
              $\boldsymbol{FOR}\ i = 1\ to\ S_b\ \boldsymbol{DO}$
                  Perform the chemotactic step (tumble-swim or tumble-tumble)
                  for bacterium $\theta^i(j,k,l)$ controlled by $\boldsymbol{N_s}$
              $\boldsymbol{END\ FOR}$
          $\boldsymbol{END\ FOR}$
          Perform the reproduction step by eliminating the $S_r$ (half) worst
          bacteria and duplicating the other half
      $\boldsymbol{END\ FOR}$
      Perform the elimination-dispersal step for all bacteria
      $\theta^i(j,k,l), \forall i, i = 1, \ldots, S_b$ with probability $0 \le P_{ed} \le 1$
  $\boldsymbol{END\ FOR}$
END
```

Figure 1. Original BFOA. Input parameters are: number of bacteria S_b, chemotactic loop limit N_c, swim loop limit N_s, reproduction loop limit N_{re}, number of bateria for reproduction S_r, elimination-dispersal loop limit N_{ed}, stepsizes C_i and probability of elimination dispersal P_{ed}.

The reproduction step consists on sorting the bacteria in the population $\theta^i(j,k,l), \forall i, i = 1, \ldots, S_b$ based on their objective function value $f\left(\theta^i(j,k,l)\right)$ and eliminating half of them with the worst values. The remaining half will be duplicated as to maintain a fixed population size.

The elimination-dispersal step consists on eliminating each bacterium $\theta^i(j,k,l)\ \forall i, i = 1, \ldots, S_b$ with a probability $0 \le P_{ed} \le 1$. Passino (2002) also modeled a swarming step, which is not considered in this paper for sake of simplicity in this current work. Instead, we propose a simpler way to simulate swarming with bacteria.

4. MBFOA FOR ENGINEERING DESIGN

Recalling from Section 3, BFOA requires seven parameters, besides the n stepsizes (where n is the number of variables of the problem), to be fine-tuned by the user. Moreover, BFOA, as EAs and other SI algorithms such as PSO, lacks a mechanism to deal with the constraints of the problem (Deb, 2000). Therefore, as to make BFOA suitable to solve engineering design problems modeled as NOPs, we propose the following modifications to the original approach.

1. A single generation loop is proposed to eliminate the four nested loops controlled by the number of chemotactic, reproduction and elimination-dispersal steps combined with the population size. In this generation loop, each bacterium will perform its own chemotactic loop. A single reproduction step and a single elimination-dispersal step are performed at the end of this generation loop. In this way, the N_s parameter is eliminated as the tumble-tumble or tumble-swim step will be only limited by N_c for each bacterium. Furthermore, the elimination-dispersal step is simplified because only the worst bacterium in the population is eliminated. As a result, N_{re}, N_{ed}, and P_{ed} parameters are also eliminated and just the GMAX, (number of generations) parameter is added due to this proposed modification.
2. The value of the stepsize $C(i)$ is not defined by the user. Instead, for each decision variable i, $C(i)$ is now computed by considering its lower and upper limits, L_i and U_i by using the following formula, utilized by Mezura-Montes & Coello-Coello (2005):
$$C_{new}(i) = R * \left(\frac{\Delta\vec{x}_i}{\sqrt{n}}\right) \qquad (3)$$
where $C_{new}(i)$ is the stepsize now not defined by the user, $\Delta\vec{x}_i$ is computed as $U_i - L_i$, n is the number of decision variables in the optimization problem and
R is the percentage of the total stepsize to be used, as low stepsize values are more convenient in constrained optimization (Mezura-Montes & Coello-Coello, 2005).
3. A parameter-less constraint-handling technique, originally proposed for genetic algorithms (Deb, 2000), was added to our BFOA. It is based on three feasibility criteria utilized in the selection mechanism (swimming and reproduction steps):
   a) Between two feasible bacteria, the one with the best objective function value is selected.
   b) Between a feasible and an infeasible bacterium, the feasible one is selected.
   c) Between two infeasible bacteria, the one with the lowest sum of constraint violation is selected. The sum of constraint violation is calculated as follows: $\sum_{i=1}^{m} \max(0, g_i(\vec{x}))$
4. A simple swarming mechanism was added to the redefined chemotactic step. Half way to the end of its chemotactic loop, each bacterium, instead of determining its search direction as pointed out in Eq. 1 and 2, uses a communication mechanism to bias its search direction to the neighborhood of the best bacterium so far in the current population. This search direction is defined in Eq. 4:
$$\theta^i(j+1,G) = \theta^i(j,G) + \beta\left(\theta^{\mathrm{B}}(G) - \theta^i(j,G)\right) \qquad (4)$$
where $\theta^i(j+1,G)$ and $\theta^i(j,G)$ are the new and current positions of bacterium i, respectively, $\theta^{\mathrm{B}}(G)$ is the current position of the best bacterium so far in generation G and $\beta > 0$ is a scaling factor which regulates how close will be the bacterium *i*

from the best one B. The remaining steps in the chemotactic loop will be performed as in Eq. 5 (tumble-swim, tumble-tumble).

$$\theta^i(j+1,G) = \theta^i(j,G) + C_{new}(i)\phi(i) \tag{5}$$

The modified BFOA, called MBFOA is detailed in Figure 2.

```
Begin
Initialize input parameters (see caption of this figure)
Create a random initial swarm of bacteria $\theta^i(j,G)\ \forall i, i = 1,\ldots,S_b$
Evaluate $f\left(\theta^i(j,G)\right) \forall_i, i = 1,\ldots,S_b$
  $\boldsymbol{FOR}\ G = 1\ to\ GMAX\ \boldsymbol{DO}$
      $\boldsymbol{FOR}\ i = 1\ to\ S_b\ \boldsymbol{DO}$
          $\boldsymbol{FOR}\ j = 1\ to\ N_c\ \boldsymbol{DO}$
              Perform the chemotactic step (tumble-swim, tumble-tumble or
              swarming) for bacterium $\theta^i(j,G)$ by using Eq. 5 and 4 and the set
              of feasibility criteria
          $\boldsymbol{End\ FOR}$
      $\boldsymbol{End\ FOR}$
      Perform the reproduction step by eliminating the $S_r$ (half) worst bacteria and
      duplicating the other half, based on the feasibility criteria
      Eliminate the worst bacterium $\theta^w(j,G)$ in the current population, based on the
      feasibility criteria
  $\boldsymbol{End\ FOR}$
$\boldsymbol{End\ FOR}$
```

Fig. 2. Modified BFOA. Input parameters are number of bacteria S_b, chemotactic loop limit N_c, number of bateria for reproduction S_r, scaling factor β, percentage of initial stepsize R, and number of generations GMAX.

5. EXPERIMENTS AND RESULTS

The experimental design to test MBFOA consisted on two phases. The first experiment aimed to analyze MBFOA's quality of solutions i.e. how close was the best solution found so far with respect to the best known solution. This experiment consisted on solving three minimization nonlinear programming problems. The details of these three problems can be found in (Ryoo & Sahinidis, 1995). A second experiment was designed to analyze, besides the quality of solutions, the consistency of good results obtained by MBFOA and its computational cost, measured by the number of evaluations of solutions computed in four minimization engineering design problems. Details of these problems can be found in (He et al., 2004). Furthermore, the results are compared with other nature-inspired approaches. 30 independent runs per each test problem with the same parameter values were conducted by BFOA in both experiments. In order to maintain newly generated values for the design variables within valid values, the following adjustment was used (Kukkonen et al., 2006): if $\theta_k^i(j+1,G) > U_k$ then $\theta_k^i(j+1,G) = 2 \cdot U_k - \theta_k^i(j+1,G)$ or if $\theta_k^i(j+1,G) < L_k$ then $\theta_k^i(j+1,G) = 2 \cdot L_k - \theta_k^i(j+1,G)$. The parameters used were the following: $S_b = 50$, $N_c = 12$, GMAX=50, S_r=25, R=1.62E-2, β =1.76, 30,000 evaluations were performed by MBFOA in each independent run for each test problem.

The obtained results for the first experiment are summarized in Table 1 and the statistical results for the second experiment (best, mean, standard deviation) are presented in Tables 2, 3, 4 and 5 for the design of a welded beam, a pressure vessel, a tension/compression spring and a speed reducer, respectively.

Table 1. Results obtained by MBFOA in the three nonlinear programming problems.

Process synthesis MINLP, quadratic objective function, 7 variables, 10 constraints $f(x^*)=4.579582$				
MBFOA	Best	Mean	St. Dev	Evaluations
	4.580396	4.781482	2.81E-1	30000
Design of a reinforced concrete beam, linear objective function, 2 variables, 2 constraints $f(x^*)=376.2919$				
MBFOA	Best	Mean	St. Dev.	Evaluations
	376.2977	376.3596	5.18E-2	30000
Quadratically constrained quadratic program, nonlinear objective function, 2 variables, 2 constraints $f(x^*)=-118.7048$				
MBFOA	Best	Mean	St. Dev.	Evaluations
	-118.7046	-118.7008	2.97E-3	30000

The results of the first experiment showed that MBFOA was able to consistently reach the vicinity of the best known feasible solution in the three different test problems. These preliminary results were encouraging. The next experiment provided more information regarding the behavior and performance of MBFOA.

From Tables 2, 3, 4 and 5, it was found that MBFOA provided a very competitive (e.g. feasible solutions were found in every independent run), but not better, performance with respect to different state-of-the-art approaches. The approach by He et al. (He et al., 2004) was the most competitive in the welded beam problem (quadratic objective function, 4 variables and 7 constraints). However, this approach requires an entire feasible population to work. As noted in Section 4 of the paper, MBFOA works with an initial population regardless the feasibility of solutions. Moreover, our algorithm required a similar number of evaluations (30,000), which is indeed a lower value with respect to Ray & Liew algorithm (2003). Riza's approach (2008) was the most competitive approach in the pressure vessel and spring design problems (quadratic objective function, 4 variables with 4 constraints and quadratic objective function, 3 variables with 4 constraints, respectively). With a similar number of evaluations, MBFOA provided competitive results, mostly in the design of the pressure vessel. Finally, Mezura et al. Differential Evolution (2007) provided the most competitive results in the speed reducer design problem (nonlinear objective function, 7 variables and 11 constraints). MBFOA exhibited a competitive performance by requiring a lower number of evaluations with respect to Ray & Liew algorithm (2003).

6. CONCLUSIONS AND FUTURE WORK

A novel adaptation of the bacterial foraging optimization algorithm to solve engineering design (constrained numerical optimization) problems was proposed. Four modifications were made to the original algorithm: (1) a simplification in the design of the algorithm, (2) a definition of the stepsizes used by the algorithm in order to keep the user from the fine-tuning of these parameters, (3) an effective but simple constraint-handling mechanism and (4) a swarming mechanism to promote collaboration among bacteria.

Table 2. Statistical results obtained for the welded beam design problem.

Approaches	Best	Mean	Std Dev	Evaluations
(Ray & Liew, 2003)	2.385000	3.255000	9.60E-1	33000
(He et al., 2004)	**2.381000**	**2.381000**	**5.20E-3**	**30000**
(Mezura & Coello, 2008)	2.386333	2.504377	9.90E-2	80000
MBFOA	2.385053	2.396375	9.93E-3	**30000**

Table 3. Statistical results obtained for the pressure vessel design problem.

Approaches	Best	Mean	Std Dev	Evaluations
(Akhtar et al., 2002)	6171.000	6335.05	NA	**20000**
(He et al., 2004)	**6059.714**	6289.928	3.10E+2	30000
(Riza, 2008)	**6059.714**	**6097.446**	**3.57E+1**	30000
(Coello & Cortes, 2004)	6061.122	6734.084	4.57E+2	150000
(He & Wang, 2007)	6061.077	6147.133	8.64E+1	200000
MBFOA	6059.945	6107.340	8.20E+1	30000

Table 4. Statistical results obtained for the tension-compression spring.

Approaches	Best	Mean	Std Dev	Evaluations
(Ray & Liew, 2003)	0.012669	0.012923	5.96E-4	25167
(He et al., 2004)	**0.012665**	0.012702	4.1E-5	**15000**
(Riza, 2008)	**0.012665**	**0.012673**	**6.24E-6**	30000
(He & Wang, 2007)	0.012874	0.012730	5.19E-5	200000
MBFOA	0.012671	0.012734	5.48E-5	30000

Table 5. Statistical results obtained for the speed reducer design problem.

Approaches	Best	Mean	Std Dev	Evaluations
(Ray & Liew, 2003)	**2994.744**	3001.758	4.0E+0	54456
(Mezura et al. 2007)	2996.357	**2996.367**	**8.2E-3**	**24000**
(Mezura et al. 2007)	3064.211	3244.569	2.0E+2	**24000**
MBFOA	2999.264	3014.759	1.10E+1	30000

The results provided in three nonlinear programming problems exhibited that MBFOA is capable of consistently reaching the best known feasible solution. Furthermore, a comparison in four engineering design problems suggested that MBFOA is able to provide, with an equal or slightly higher computational cost, a competitive performance with respect to state-of-the-art approaches. However, premature convergence was still observed in MBFOA. This shortcoming is the start of the future work, where a mechanism to avoid local optima will be designed. Finally, the parameters of MBFOA will be analyzed more in-depth.

Acknowledgements – The first author acknowledges support from CONACyT through project 79809-Y. The second author acknowledges support from CONACyT through project 102299.

REFERENCES

Akhtar, S., Tai, K. and Ray, T., 2002 A Socio-Behavioural Simulation Model for Engineering Design Optimization. Engineering Optimization, 34(4):341-354.

Biswas, A., Dasgupta, S., Das, S., and Abraham, A., 2007, Synergy of PSO and bacterial foraging optimization - a comparative study on numerical benchmarks, in Innovations in Hybrid Intelligent Systems, Springer-Verlag, pp. 255-263.

Bremermann, H., 1974, Chemotaxis and optimization. J. Franklin Inst., 297:397–404.

Coello-Coello C.A., Cruz-Cortés N., 2004, Hybridizing a genetic algorithm with an artificial immune system for global optimization. Engineering Optimization, 36(5):607–634

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.

He, S., Prempain, E., and Wu, Q.H., 2004, An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems, Engineering Optimization, 36(5):585-605.

He, Q., and Wang, L., 2007, An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence, 20(1):89-99.

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.

Majhi, B., and Panda, G., 2007, Bacteria foraging based identificacion of nonlinear dynamic system, In Proceedings *of the IEEE Congress on Evolutionary Computation*, IEEE, pp. 1636-1641.

Mezura-Montes, E., and Coello-Coello, C.A., 2005, A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems, IEEE Transactions on Evolutionary Computation, 9(1):1-17.

Mezura-Montes, E., and Coello-Coello, C.A., 2007, Velázquez-Reyes, J., and Muñoz-Dávila, L., Multiple Trial Vectors in Differential Evolution for Engineering Design, Engineering Optimization, 39(5):567-589.

Mezura-Montes, E. and Coello-Coello C.A., 2008, Constrained Optimization via Multiobjective Evolutionary Algorithms. In J. Knowles, D. Corne, and K. Deb, editors, Multiobjective Problem Solving from Nature, Springer, pp. 53-75.

Passino, K., 2002, Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine, 22(3):52–67.

Ray, T. and Liew, K., 2003, Society and civilization: An optimization algorithm based on the simulation of social behavior, IEEE Transactions on Evolutionary Computation, 4(7):386-396.

Riza, A.Y., 2008, A novel particle swarm optimization approach for product design and manufacturing. Int J Adv Manuf Technol, Springer-Verlag, 40(5-6):617-628.

Ryoo H.S., and Sahinidis N.V., 1995, Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design, *Computers Chem Eng*, 19(5):551-566.