Bacterial Foraging for Engineering Design Problems: Preliminary Results

Efrén Mezura-Montes* **and Betania Hernández-Ocaña**‡

* Laboratorio Nacional de Informática Avanzada (LANIA A.C.)
Rébsamen 80, Centro, Xalapa, Veracruz, 91000 México
email: emezura@lania.mx

‡Universidad Juárez Autónoma de Tabasco
Div. Acad. de Informática y Sistemas Km. 1 Carr. Cunduacán-Jalpa de Méndez
042H4005@dais.ujat.mx

***Abstract*— In this paper we present an approach based on 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 bacterial foraging optimization algorithm as to adapt it to solve constrained problems in numerical search spaces. The modifications aim to decrease the number of parameters used in the algorithm, to add a constraint-handling mechanism and to improve the communication capabilities among bacteria. The approach is tested on three well-known engineering design problems and its performance is compared against some state-of-the-art approaches. Based in these preliminary results, some conclusions are established and the future work is defined.**

I. Introduction

Nowadays, the use of alternative approaches to solve complex optimization problems is very common [15]. Evolutionary Algorithms have received many interest from researchers and practitioners as they have provided very competitive results when solving engineering design problems [13]. Furthermore, Swarm Intelligence approaches have been also used to solve this kind of problems [19], [5]. However, most of the work is centered on some algorithms such as Particle Swarm Optimization [7] Ant Colony Optimization [9] and Artificial Bee Colony [6].

In this paper we explore another swarm-intelligence-based model: Bacterial Foraging Optimization Algorithm (BFOA), inspired in the behavior of bacteria E. Coli in its search for food. Three behaviors were modeled by Passino in his original proposal [18]: (1) Chemotaxis, (2) reproduction and (3) elimination-dispersal. BFOA has been successfully applied to solve different type of problems like forecasting [11], transmission loss reduction [17] and Identification of nonlinear dynamic systems [10]. Furthermore, BFOA has been combined with other algorithms as to solve multimodal optimization problems [2]. However, research focused on using BFOA for numerical optimization in presence of constraints is 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 four modifications to the original BFOA: (1) A single generation loop to include the chemotactic, reproduction and elimination-dispersal steps, (2) a stepsize definition based on the features of the problem to be solved, (3) a constraint-handling mechanism and (4) a simple communication model among bacteria as to allow them to move towards promising regions of the search space.

Some engineering design problems can be stated as nonlinear optimization problems (NOPs)in which the goal is to:

$$\text{Find} \quad \vec{x} \text{ which optimizes } f(\vec{x}) \tag{1}$$

subject to:

$$g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, m \tag{2}$$

$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, p \tag{3}$$

where $\vec{x}$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_n]^T$, where each $x_i, \;\; i = 1, \ldots, n$ 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 (in both cases, constraints could be linear or nonlinear). If we denote with $\mathcal{F}$ to the feasible region and with $\mathcal{S}$ to the whole search space, then it should be clear that $\mathcal{F} \subseteq \mathcal{S}$.

This paper is organized as follows: In Section II we describe the original BFOA, Section III presents the four modifications made to the original BFOA and presents our modified BFOA called MBFOA. Section IV presents the experimental design, the problems used to test MBFOA and the obtained results compared against state-of-the-art approaches. Finally, we conclude in Section V with some interesting findings and our future work.

II. Bacterial Foraging Optimization Algorithm

As other swarm intelligence algorithms, BFOA is based on social and cooperative behaviors found in nature. In fact, the way Bacteria look for regions of high levels of nutrients can be seen as an optimization process. This idea was explored by Bremermann [3] and extended later by Passino [18]. Each bacteria tries to maximize its obtained energy per each unit of time expended on the foraging process and avoiding noxious substances. Besides, swarm search assumes communication among individuals. The swarm of bacteria S behavies as follows [18]:

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 reproduce (split).
3) Bacteria are located in promising regions of the map of nutrients as they try to attract other bacteria by generating chemical attractants.
4) Bacteria are now located in the highest-nutrient region.
5) Bacteria now disperse as to look for new nutrient regions in the map.

Three main steps comprise bacterial foraging behavior: (1) Chemotaxis (tumble and swimming), (2) reproduction and (3) elimination-dispersal.

Based on these steps, Passino [18] proposed the Bacterial Foraging Optimization Algorithm which is summarized in Figure 1.

Begin
  Initialize input parameters (see caption of this figure)
  Create a random initial swarm of bacteria $\theta^i(j,k,l)$ $\forall i,\ i=1,\ldots,S_b$
  Evaluate $f(\theta^i(j,k,l))$ $\forall i,\ i=1,\ldots,S_b$
  For l=1 to N_{ed} **Do**
    For k=1 to N_{re} **Do**
      For j=1 to N_c **Do**
        For i=1 to S_b **Do**
          Perform the chemotaxis step
          (tumble-swim or tumble-tumble)
          for bacteria $\theta^i(j,k,l)$
        End For
      End For
      Perform the reproduction step by eliminating the S_r (half) worst bacteria and duplicating the other half
    End For
    Perform the elimination-dispersal step for all bacteria $\theta^i(j,k,l)$ $\forall i,\ i=1,\ldots,N_b$ with probability $0 \leq P_{ed} \leq 1$
  End For
End

Fig. 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 (they depend of the dimensionality of the problem) and probability of elimination dispersal p_{ed}.

The chemotactic step was modeled by Passino with the generation of a random direction search (Equation 4)

$$\phi(i) = \frac{\Delta(i)}{\sqrt{\Delta(i)^T \Delta(i)}} \quad (4)$$

where $\Delta_(i)^n$ is a randomly generated vector with elements within the following interval: $[-1,1]$. After that, each bacteria $\theta^i(j,k,l)$ modifies its positions as indicated in Equation 5.

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(i) \quad (5)$$

Equation 4 represents a tumble (search direction) and Equation 5 represents a swim. The swim will be repeated N_s times if the new position is better than the previous one: $f(\theta^i(j+1,k,l)) < f(\theta^i(j,k,l))$.

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

The elimination-dispersal step consists on eliminate each bacteria $\theta^i(j,k,l), \forall i,\ i=1,\ldots,N_b$ with a probability $0 \leq P_{ed} \leq 1$.

Passino [18] also modeled a swarming step, which is not considered in this paper by sake of simplicity in this work. Instead, we propose a simpler way to simulate swarming in bacteria.

III. MBFOA FOR ENGINEERING DESIGN

As noted in Section II, BFOA requires 7 parameters and the n stepsizes dependeing of the number of variables of the problem to be fine-tuned by the user. Furthermore, BFOA, as other nature-inspired heuristics, lack a mechanism to deal with the constraints of a problem [16]. Therefore, as to make BFOA more suitable to solve engineering design problems modeled as NOPs, we propose the following simplifications to the original approach.

- Instead of having four nested loops controlled by the number of chemotactic, reproduction, elimination-dispersal and population size loops, a single generation loop is proposed where each bacteria will perform its own chemotactic loop. After that, a single reproduction step and a single elimination-dispersal loop are performed within 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 bacteria. Furthermore, the elimination-dispersal step is simplified as to only eliminate the worst bacteria in the population. Then, the N_{re}, N_{ed}, and p_{ed} parameters are eliminated and just the $GMAX$, (number of generations) parameter is added.
- The stepsize $C(i)$ for each decision variable x_i is defined by considering the lower and upper limits L_i and U_i by using the following formula proposed in [12]:

$$C_{new}(i) = R * \left(\Delta\vec{x}_i / \sqrt{n}\right) \quad (6)$$

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 small initial stepsizes are more convenient in constrained optimization [12].

- A parameterless constraint-handling technique was added to our BFOA. It is based on three feasibility criteria [4] used in the selection mechanism (swimming and reproduction steps).
  1) Between two feasible solutions, the one with the best objective function value is selected.
  2) Between a feasible and an infeasible solution, the feasible one is selected.
  3) Between two infeasible solutions, the one with the lowest sum of constraint violation is selected. The sum of constraint violation is calculated as follows: $\sum_{i=1}^{m+p} \max(0, g_i(\vec{x}))$
- A simple swarming mechanism was added to the redefined chemotactic step for each bacteria in the population. At the half and at the end of the chemotactic loop, instead of each bacteria to determine its search direction as pointed out in Equations 4 and 5, a communication step is modeled as to allow this bacteria to bias its direction search to the neighborhood of the best bacteria so far in the current population. This search direction is defined in Equation 7:

$$\theta^i(j+1,G) = \theta^i(j,G) + \beta(\theta^B(G) - \theta^i(j,G)) \quad (7)$$

where $\theta^i(j+1,G)$ is the new position of bacteria i, $\theta^B(G)$ is the current position of the best bacteria in generation G so far and $0 \leq \beta \leq 1$ is a scaling factor which regulates how close will be the bacteria i from the best one B. The remaining steps in the chemotactic loop will be performed as in Equation 8

$$\theta^i(j+1,G) = \theta^i(j,G) + C_{new}(i)\phi(i) \quad (8)$$

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

As it can be noted in pseudocodes of BFOA in Figure 1 and MBFOA in Figure 2, the number of parameters was decreased and the definition of the stepsize values are not defined by the user, the number of nested loops was also decreased, which represent a lower computational complexity. Finally, a simple collaboration mechanism was added to MBFOA as to allow a balance between exploration (Equation 8) and exploitation (Equation 7) of the search space.

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 $f(\theta^i(j,G))$ $\forall i,\ i=1,\ldots,S_b$
    For G=1 to $GMAX$ **Do**
        For i=1 to S_b **Do**
            For j=1 to N_c **Do**
                Perform the chemotaxis step
                (tumble-swim or tumble-tumble)
                for bacteria $\theta^i(j,G)$
                by using Eq. 7 and 8
                and the set of feasibility criteria
            End For
        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 bacteria $\theta^w(j,G)$ in the current population, based on the feasibility criteria
    End For
End

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 $GMAX$.

IV. Experiments and Results

We selected three well-known engineering design problems to be solved by MBFOA and also a set of approaches representative of the state-of-the-art in nature-inspired optimization as to perform a comparison of final results. Graphical information of the problems is in Figure 3 and formal statements are in the Appendix at the end of the paper.

30 independent runs per each test problem with the same parameter values (see Table I were conducted and statistical results (best, mean, standard deviation) were calculated. As to maintain new values for the design variables within valid values the following adjustment was used [8]: if $x_i > U_i$ then $x_i = 2U_i - x_i$ or if $x_i < L_i$ then $x_i = 2L_i - x_i$. The obtained results are summarized in Table II and the details of the best solution found are presented in Tables III, IV and V for the welded beam, the pressure vessel and the tension/compression spring problems respectively.

TABLE I
Parameter values for MBFOA in the experiments

Parameter	Value
S_b	50
N_c	12
$GMAX$	80
S_r	25
R	2.1E-3
F	0.44

The results of BFOA are compared against six approaches found in the specialized literature: The society and civilization simulation proposed by Ray & Liew [19] and by Akthar et al. [1], the fly-back mechanism added to a PSO proposed by He et al. [5] and the four multiobjective-based constraint-handling genetic algorithms proposed by Mezura & Coello [13].

The results are discussed based on three measures: (1)Quality i.e. the best solution found so far, (2) Consistency i.e. the mean value closer to the BKS and the lowest standard deviation value and (3) computational cost i.e. the number of evaluations required by an approach.

For the welded beam problem, He's approach [5] obtained the highest quality and consistency results. However,

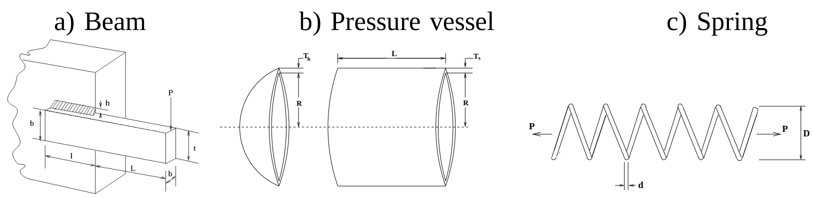

Fig. 3. Figures of the engineering design problems.

TABLE II

STATISTICAL SUMMARY. A RESULT IN **BOLDFACE** MEANS THE BEST RESULT. "BKS" MEANS BEST KNOWN SOLUTION. EVALS IS THE NUMBER OF EVALUATIONS REQUIRED BY EACH APPROACH. "NA" MEANS RESULT NOT AVAILABLE

Problem	Approach	BKS	Best	Mean	St. Dev	Evals
Welded beam	Ray & Liew	2.380	2.385	3.255	9.6E-1	33000
	He et al.		**2.381**	**2.381**	5.2E-3	30000
	COMOGA		2.471	2.726	1.20E-1	80000
	CHVEGA		2.386	2.393	**3.8E-3**	80000
	CHNPGA		2.382	2.420	2.56E-2	80000
	CHMOGA		2.386	2.504	9.9E-2	80000
	MBFOA		2.386	2.404	1.6E-2	48000
Pressure vessel	Akhtar et al.	6059.701	6171.00	6335.05	NA	**20000**
	He et al.		**6059.7143**	6289.928	3.1E+2	30000
	COMOGA		6369.428	7795.412	7.01E+2	80000
	CHVEGA		6064.724	6259.964	1.7E+2	80000
	CHNPGA		6059.926	6172.527	1.24E+2	80000
	CHMOGA		6066.967	6629.064	3.85E+2	80000
	MBFOA		6060.460	**6074.625**	**1.56E+1**	48000
Spring	Ray & Liew	0.012665	0.012669	0.012923	5.96E-4	25167
	He et al.		**0.012665**	**0.012702**	**4.1E-5**	**15000**
	COMOGA		0.012929	0.014362	8.64E-4	80000
	CHVEGA		0.012688	0.012886	2.09E-4	80000
	CHNPGA		0.012683	0.012752	6.20E-5	80000
	CHMOGA		0.12680	0.012960	3.63E-4	80000
	MBFOA		0.012671	0.012759	1.36E-4	48000

TABLE III

DETAILS OF THE BEST SOLUTION FOUND FOR THE WELDED BEAM PROBLEM

Welded Beam	Ray & Liew [19]	He et al. [5]	MBFOA
x_1	0.244438	0.244369	0.244540
x_2	6.237967	6.217520	6.183924
x_3	8.288576	8.291471	8.326537
x_4	0.244566	0.244369	0.244677
$g_1(\vec{x})$	-5760.110471	-5741.176933	-9.021085
$g_2(\vec{x})$	-3.245428	-0.000001	-289.614794
$g_3(\vec{x})$	-0.000128	-0.000000	-0.000137
$g_4(\vec{x})$	-3.020055	-3.022955	-3.015408
$g_5(\vec{x})$	-0.119438	-0.119369	-0.119540
$g_6(\vec{x})$	-0.234237	-0.234241	-0.234459
$g_7(\vec{x})$	-13.079305	-0.000309	-40.379616
$f(\vec{x})$	2.385434	2.380957	2.386845

MBFOA was the third best apprach regarding consistency. Furthermore, MBFOA requires $32,000$ less evaluations to reach the reported results, compared to the $80,000$ required by COMOGA, CHVEGA, CHNPGA and CHMOGA [13].

TABLE IV
DETAILS OF THE BEST SOLUTION FOUND FOR THE PRESSURE VESSEL PROBLEM

Pressure vessel	**Akhtar et al. [1]**	**He et al. [5]**	**MBFOA**
x_1	0.8125	0.8125	0.8125
x_2	0.4375	0.4375	0.4375
x_3	41.9768	42.098446	42.096394
x_4	182.2845	176.636052	176.683231
$g_1(\vec{x})$	-0.0023	-0.000000	-0.000040
$g_2(\vec{x})$	-0.037	-0.035881	-0.035900
$g_3(\vec{x})$	-23420.5966	-0.000000	-121.085825
$g_4(\vec{x})$	-57.7155	-63.36340	-63.316769
$f(\vec{x})$	6171.0	6059.7143	6060.46

TABLE V
DETAILS OF THE BEST SOLUTION FOUND FOR THE TENSION/COMPRESSION SPRING PROBLEM

Tension/Compression Spring	**Ray & Liew [19]**	**He et al. [5]**	**MBFOA**
x_1	0.0521602	0.051690	0.051825
x_2	0.368159	0.356750	0.359935
x_3	10.648442	11.287126	11.107103
$g_1(\vec{x})$	-0.000000	-0.000004	-0.000176
$g_2(\vec{x})$	-0.000000	0.000000	-0.000147
$g_3(\vec{x})$	-4.075805	-4.053827	-4.058410
$g_4(\vec{x})$	-0.719787	-0.727706	-0.725493
$f(\vec{x})$	0.012669	0.012665	0.012671

Finally, MBFOA provided better consistency results with respect to Ray & Liew [19] by using $15,000$ more evaluations.

For the pressure vessel problem, He's approach [5] obtained the best quality result. However, MBFOA was the most consistent, requiring $32,000$ less evalautions than the four multiobjective-based approaches [13] but using $28,000$ more evaluations than Akhtar et al. [1].

In the tension/compression spring, Hes approach obtained the best quality and consistency results and required the lowest number of evaluations to reach those results. MBFOA was the third best approach based on quality results, the third best based on consistency and required $32,000$ less evaluations than the number required by the four multiobjective-based approaches [13].

Based on the discussion of results it is clear that He's approach seems to be the most competitive. However, this algorithm requires a completely feasible initial population. because its operators are designed to maintain only feasible solutions. This is a serious drawback with respect to the other compared proposals, including MBFOA, which start from a completely random-generated initial population (with several, if not all, infeasible solutions).

As a general finding, the overall results suggest that MBFOA is a promising heuristic to solve engineering design problems. This first attempt of adapting BFOA to solve this kind of problems still have some drawbacks such as a tendency to get trapped in local optima and also that its number of evaluations is still a little bit high with respect to some state-of-the-art approaches.

V. CONCLUSIONS AND FUTURE WORK

In this paper we presented a first attempt to simplify the original Bacterial Foraging Optimization Algorithm (BFOA) as to solve engineering design problems. Four modifications were implemented: (1) A single generation loop where the chemotactic step (now per each bacteria) the reproduction and the elimination-dispersal steps were included as to decrease the computational complexity of the heuristic, (2) A definition of the stepsize values based on the limits of the decision variables as to keep the user to define them, (3) a parameterless but effective constraint-hadling technique as to deal with the feasibility of solutions and (4) a simple swarming operator which allow bacteria to direct its swim to the best bacteria in the current population in order to improve the exploitation capabilities of the original BFOA. The modified BFOA (MBFOA) was tested on 3 engineering design problems and its performance was compared against six state-of-the-art approaches found in the specialized literature. MBFOA's behavior was competitive mostly in the number of evaluations required and in the consistency of results, regardless of starting with a randomly-generated initial population (with feasible and infeasible solutions). As a main drawback, MBFOA still tends to get trapped in local optima, affecting the quality of the obtained results. As future paths of research, we will analyze more in-depth the balance between exploration (Equation 8) and exploitation (Equation

7) BFOA has, and to improve it. Finally, we will test MBFOA in other constrained optimization problems [14].

ACKNOWLEDGMENTS

The first authors acknowledges support from CONACyT through project number 79809. The second author acknowledges support from the Juárez Autónoma de Tabasco University through a scholarship for a summer stay at LANIA.

REFERENCES

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

[2] A. Biswas, S. Dasgupta, S. Das, and A. Abraham. Synergy of pso and bacterial foraging optimization - a comparative study on numerical benchmarks. In E. C. et al., editor, *Innovations in Hybrid Intelligent Systems 2007*, pages 255–263. Spronger-Verlag, 2007.

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

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

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

[6] D. Karaboga and B. Basturk. On the performance of artificial bee colony (abc) algorithm. *Applied Soft Computing*, 8(1):687–697, 2008.

[7] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann, UK, 2001.

[8] S. Kukkonen and J. Lampinen. Constrained Real-Parameter Optimization with Generalized Differential Evolution. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 911–918, Vancouver, BC, Canada, July 2006. IEEE.

[9] G. Leguizamón and C. Coello-Coello. A Boundary Search based ACO Algorithm Coupled with Stochastic Ranking. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 165–172, Singapore, September 2007. IEEE Press.

[10] B. Majhi and G. Panda. Bacteria foraging based identificacion of nonlinear dynamic system. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 1636–1641. Singapore, IEEE Service Center, September 2007.

[11] R. Majhi, G. Panda, G. Sahoo, P. Dash, and D. Das. Stock market prediction of s&p 500 and djia using bacterial foraging optimization technique. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 2569–2575. Singapore, IEEE Service Center, September 2007.

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

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

[14] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello. Comparing Differential Evolution Models for Global Optimization. In *2006 Genetic and Evolutionary Computation Conference (GECCO'2006)*, volume 1, pages 485–492, July 2006.

[15] Z. Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer, Germany, 2nd edition, 2004.

[16] Z. Michalewicz and M. Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, 1996.

[17] S. Mishra, G. D. Reddy, P. E. Rao, and K. Santosh. Implementation of new evolutionary techniques for transmission loss reduction. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 2331–2336. Singapore, IEEE Service Center, September 2007.

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

[19] T. Ray and K. Liew. Society and Civilization: An Optimization Algorithm Based on the Simulation of Social Behavior. *IEEE Transactions on Evolutionary Computation*, 7(4):386–396, August 2003.

APPENDIX

Full description of the four problems used in the experiments:

Problem 1: (Design of a Welded Beam) A welded beam is designed for minimum cost subject to constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c), end deflection of the beam (δ), and side constraints. There are four design variables as shown in Figure 3a: h (x_1), l (x_2), t (x_3) and b (x_4). The problem can be stated as follows:

Minimize: $f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$

Subject to:
$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leq 0$
$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leq 0$
$g_3(\vec{x}) = x_1 - x_4 \leq 0$
$g_4(\vec{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$
$g_5(\vec{x}) = 0.125 - x_1 \leq 0$
$g_6(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leq 0$
$g_7(\vec{x}) = P - P_c(\vec{x}) \leq 0$

where $\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$
$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right)$
$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}$
$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$ $\quad \sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\mathbf{X}) = \frac{4PL^3}{Ex_3^3x_4}$
$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$ $P = 6000\ lb,\quad L = 14\ in,$
$E = 30 \times 10^6\ psi,\quad G = 12 \times 10^6\ psi\ \tau_{max} = 13,600\ psi,$
$sigma_{max} = 30,000\ psi,\quad \delta_{max} = 0.25\ in$
where $0.1 \leq x_1 \leq 2.0$, $0.1 \leq x_2 \leq 10.0$, $0.1 \leq x_3 \leq 10.0$ y $0.1 \leq x_4 \leq 2.0$.

Problem 2: (Design of a Pressure Vessel) A cylindrical vessel is capped at both ends by hemispherical heads as shown in Figure 3b. The objective is to minimize the total cost, including the cost of the material, forming and welding. There are four design variables: T_s (thickness of the shell), T_h (thickness of the head), R (inner radius) and L (length of the cylindrical section of the vessel, not including the head). T_s and T_h are integer multiples of 0.0625 inch, which are the available thicknesses of rolled steel plates, and R and L are continuous. The problem can be stated as follows:

Minimize : $f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$

Subject to :
$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0$
$g_2(\vec{x}) = -x_2 + 0.00954x_3 \leq 0$
$g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0$
$g_4(\vec{x}) = x_4 - 240 \leq 0$

where $1 \leq x_1 \leq 99$, $1 \leq x_2 \leq 99$, $10 \leq x_3 \leq 200$ y $10 \leq x_4 \leq 200$.

Problem 3: (Minimization of the Weight of a Tension/Compression String) This problem consists of minimizing the weight of a tension/compression spring (see Figure 3c) subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are the mean coil diameter D (x_2), the wire diameter d (x_1) and the number of active coils N (x_3). Formally, the problem can be expressed as:

Minimize: $(N + 2)Dd^2$

Subject to:
$g_1(\vec{x}) = 1 - \frac{D^3N}{71785d^4} \leq 0$
$g_2(\vec{x}) = \frac{4D^2 - dD}{12566(Dd^3 - d^4)} + \frac{1}{5108d^2} - 1 \leq 0$
$g_3(\vec{x}) = 1 - \frac{140.45d}{D^2N} \leq 0$
$g_4(\vec{x}) = \frac{D+d}{1.5} - 1 \leq 0$

where $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$ y $2 \leq x_3 \leq 15$.