A Study of Constraint-Handling Techniques in Brain Storm Optimization

Adriana Cervantes-Castillo & Efrén Mezura-Montes
Artificial Intelligence Research Center
University of Veracruz
Sebastián Camacho 5, Centro,
Xalapa, Veracruz, 91000, MÉXICO
Email: anairdacc@gmail.com, emezura@uv.mx

***Abstract*—A study on three BSO algorithm versions: Brain Storm Optimization Algorithm (BSO), Modified Brain Storm Optimization Algorithm (MBSO) and Simple Modified Brain Storm Optimization Algorithm (SMBSO), for constrained numerical optimization problems is presented in this paper. The aim of the study is to know the performance of this recent Swarm Intelligence (SI) algorithm on constrained search spaces. The feasibility rules, ε-constrained method, and stochastic ranking are used as constraint-handling techniques. The performance of each version is analysed by solving 24 well-known benchmark problems. The final results suggest MBSO and the ε-constrained method as a good option to deal with constrained problems.**

I. Introduction

Several complex real-world problems have associated constraints which make them hard to resolve. In the specialized literature a Constrained Numerical Optimization Problem (CNOP) is defined, without loss of generality, as to:
find

$$f(\vec{x}) \tag{1}$$

subject to:

$$g_i(\vec{x}) <= 0; i = 1, 2, \ldots, m$$

$$h_j(\vec{x}) = 0; j = 1, 2, \ldots, p$$

where $f(\vec{x})$ represents the objective function, $g_i(\vec{x})$, $i = 1, \ldots, m$ and $h_j(\vec{x})$, $j = 1, \ldots, p$ are the set of inequality and equality constraints, respectively, $\vec{x} = [x_1, x_2, \ldots, x_n]$ is the decision variable vector of the problem, where each x_k, $k = 1, \ldots, n$ is bounded by a lower L_k and upper limits U_k, $L_k \leq x_k \leq U_k$. The boundaries define the search space S and the feasible region $F \subseteq S$ is defined by those solutions that satisfy the set of equality and inequality constraints.

In the efforts to give solution to constrained problems, nature-inspired algorithms have been used [1], [2] and Swarm Intelligence (SI) has been particularly active in that regard. SI emerges through the analysis of complex collective behaviour among individuals in distinct social groups as humans or biological organisms, giving as a result different SI algorithms, such as, Particle Swarm Optimization (PSO), Ant Colony System (ACS), Bacterial Foraging Optimization Algorithm (BFOA), Artificial Bee Colony Algorithm (ABC) and others [3], [4], [5], [6].

A recent algorithm, which is based on the process of how humans generate ideas to solve complex problems, is the Brain Storm Optimization Algorithm (BSO). This one has shown encouraging results when used on unconstrained problems. In 2011, Shi [7] proposed the BSO algorithm, testing its behaviour on two benchmark functions and getting good final results. In 2012, Zhan et al. [8] proposed some modifications to the BSO algorithm and presented a new BSO version called Modified Brain Storm Optimization (MBSO). Two main modifications were done to BSO. The first of them was the usage of the simple grouping method instead of the k-means method to generate clusters. The second modification was discarding the Gaussian random strategy to generate new individuals. A different strategy was proposed instead. In 2013, a parameter study on BSO was presented in [9], in this work, the two earlier BSO versions were used to analyse three main parameters, which were: p-replace, p-one and p-center. The first parameter is a probability to give more exploration to the algorithm when it starts. The second one, is related to a probability to use one or two clusters to generate new ideas and the last parameter is a probability to generate a new idea based on the centroid or a random idea. Their best results were obtained if the p-replace parameter is omitted and only one cluster is used to get new individuals. In addition, a Gaussian distribution with a mean of 0.4 and a standard deviation of 0.1 was used to assign the value of the p-center parameter.

Some others works using the BSO algorithm have been carried out [10], [11], [12]. However, its performance on constrained numerical optimization has been, to the best of the authors' knowledge, widely ignored or few explored. Being BSO a recent algorithm and motivated by those earlier and encouraging results on unconstrained spaces, a study on three versions of BSO algorithms [7], [8], [9] for constrained numerical optimization problems using three different constraint-handling techniques is presented in this work.

The rest of this paper is organized as follows. In Section II the BSO algorithm is detailed. Section III explains the constraint-handling techniques used in this work. The experimental design is presented in Section IV and in Section V results and discussion are mentioned. Finally, the conclusions and future work are presented in Section VI.

II. BSO Algorithm

The BSO algorithm mimics the way that humans resolve complex problems. When difficult problems cannot be resolved by a single person, a group of people turn to generate a brainstorming process. This process lets them generate suitable ideas which could be combined for giving a successful solution to the problem of interest [7], [13]. Algorithm 1 shows the original version of BSO [7], which uses three main operators (grouping, replacing and creating) to generate new ideas at each generation. Those operators are reflected in steps 4, 7 and 11-28, respectively in Algorithm 1.

Algorithm 1 BSO Algorithm

```
1: Randomly generate N ideas and evaluate them;
2: while Not stop condition do
3:    // Grouping operator
4:    Cluster the N ideas into M clusters;
5:    Record the best idea in each cluster as cluster center;
6:    // Replacing operator
7:    if random(0,1) < p-replace then
8:       Select a random cluster and replace the cluster center
         with a new random idea;
9:    end if
10:   // Creating operator
11:   for i=1 to N do
12:      if random(0,1)< p-one then
13:         Randomly select a cluster
14:         if random(0,1) < p-one-center then
15:            Add random values to the selected cluster center
               to generate a new idea;
16:         else
17:            Generate a new idea adding random values to a
               random idea taken of the selected cluster;
18:         end if
19:      else
20:         Randomly select two clusters;
21:         if random (0,1) < p-two-center then
22:            Generate a new idea, combining the two selected
               clusters' center;
23:         else
24:            Generate a new idea, combining two random
               ideas from the two selected clusters;
25:         end if
26:      end if
27:      Evaluate the new idea. If the new idea is better than
         the current idea, replace the actual by the new idea;
28:   end for
29: end while
```

In steps 11-28, BSO generate new ideas using the Gaussian random strategy with Equations (2) and (3):

$$Y_i = X_i + \xi * n(\mu, \sigma) \tag{2}$$

$$\xi = logsig(\frac{(0.5 * T - t)}{k}) * rand() \tag{3}$$

where Y_i is the new solution, X_i is the selected solution of the actual cluster; $n(\mu, \sigma)$ is a Gaussian function with mean μ and variance σ; T is the maximum number of iterations, t is the actual iteration and k is used for changing logsig () function's slope.

Besides the BSO algorithm [7], the other variants that will be analysed in this research are the modified version of BSO (MBSO) [8] and the simple BSO (SMBSO) [9]. The differences between BSO and MBSO are the following: on step 4, MBSO uses a Simple Grouping Method (SGM) to cluster the ideas, which is described as follows:

1) M ideas are selected in the initial generation as the seed of the M groups.
2) Measure, for each idea in the current generation, its distance to each one of the M ideas selected in the early step.
3) Compare the distances to each one of the M ideas and add the idea to nearest group.

Instead of the Gaussian random strategy, MBSO uses the Idea Difference Strategy (IDS) as shown in Equation 4:

$$Y_i = \begin{cases} rand(L, H) & \text{if } rand(0,1) < pr; \\ X_i + rand(0,1) \times (X_a - X_b) & \text{otherwise.} \end{cases} \tag{4}$$

where X_a and X_b are two random ideas taken from the current set of ideas to represent the ideas difference, and pr is a parameter to control the open minded to create the new idea. This parameter allows the unusual ideas being welcome as in the human brainstorming process.

Finally, the differences between BSO and SMBSO are the following: in SMBSO the replacing operator is omitted, only one cluster is used to generate the new ideas, and the value of p-center is computed using Gaussian random values with a mean of 0.4 and a standard deviation of 0.1.

III. Constraint-Handling Techniques

In this work, three constraint-handling techniques are studied on the aforementioned BSO algorithms: the feasibility rules proposed by Deb [14], the ε-constrained method proposed by Takahama and Sakai [15] and the stochastic ranking proposed by Runarsson and Yao [16]. The three constraint-handling techniques are described in this section.

A. Feasibility Rules

The feasibility rules make a distinction between feasible and infeasible solutions which does not require any penalty parameter to be configured. To define the best, out of two solutions, the author proposed the next rules:

1) Between two feasible solutions, that with better objective function value is preferred.
2) Between two infeasible solutions, that with smaller constraint violation is preferred.
3) Between a feasible solution and an infeasible solution, the feasible one is preferred.

B. ε-Constrained method

This method transforms a constrained problem into an unconstrained one [15]. The constraint violation can be calculated as follows:

$$\phi = max(max(0, g_j(x)), max|h_j(x)|) \tag{5}$$

$$\phi = \sum_j ||max(0, g_j(x))||^p + \sum_j ||h_j(x)||^p \tag{6}$$

where p is a positive number, h and g represent the equality and inequality constraints, respectively. $x1$ and $x2$ are two solutions to compare and f_{x1}, f_{x2} are their objective function values and ϕ_{x1},ϕ_{x2} are their constraint violations, respectively. The comparison considers the objective function value and the constraint violation. According to that, the ϵ level comparison between the two solutions is computed as follows:

$$(f_{x1}, \phi_{x1}) <_\epsilon (f_{x2}, \phi_{x2}) \iff \begin{cases} f_{x1} < f_{x2}, \text{if } \phi_{x1}, \phi_{x2} \leq \epsilon; \\ f_{x1} < f_{x2}, \text{if } \phi_{x1} = \phi_{x2}; \\ \phi_{x1} < \phi_{x2}, \text{otherwise} \end{cases}$$

when $\epsilon = 0$, the infeasible solutions are compared based on their constraint violation. On the other hand, when $\epsilon = \infty$ the solutions are compared based only on their objective function value.

The ϵ level is controlled as in Equation (7).

$$\epsilon(0) = \phi(x_\theta)$$

$$\epsilon(t) = \begin{cases} \epsilon(0)(1 - \frac{t}{Tc})^{cp} & 0 < t < Tc; \\ 0 & t \leq Tc. \end{cases} \tag{7}$$

where t is the actual iteration; Tc = $Max-iter$ and X_ϕ is the top ϕ-th solution, θ = 0.2N and cp is the parameter to control the speed of constraint tolerance reduction.

C. Stochastic Ranking

Runarsson and Yao proposed in 2002 [16] an alternative constraint-handling technique named stochastic ranking. In this technique, a probability Pf is used to balance the dominance of two solutions according to its objective function or sum of constraint violation value. In this way, when two solutions are feasible the probability to compare them based on the objective value is 1, otherwise, it is Pf. Stochastic ranking uses a bubble-sort-procedure to rank individuals as shown in Algorithm 2. One advantage of stochastic ranking is that no penalty factors must be defined by the user.

In Algorithm (2) U(0,1) is a uniform random generator; N is the number of sweeps in the population; $f(x)$ refers to the objective function; $\phi(x)$ refers to the sum of constraint violation; λ are the individuals who will be ranked and Pf is the probability to compare solutions according to $f(x)$.

IV. Experimental design

Two experiments were carried out. The first of them consisted of analysing the three BSO versions [7], [8], [9] with each constraint-handling technique. The aim was to get evidence about positive effects of a constraint-handling technique with a particular BSO version. According to the results obtained in the first experiment, a second experiment was carried out using the three BSO versions with the constraint-handling method that reported the best results in the first experiment. The goal was to know if one combination of BSO/constraint-handling technique was able to outperform the others. The 95%-confidence Rank Sum Wilcoxon test was applied to the results so as to get statistical confidence about the findings on 30 independent runs per BSO variant with each constraint-handling technique.

A set of well-known test problems was adopted in this research [17]. A summary of their features is presented in Table I.

Considering the fact that stochastic ranking requires a ranking process, for the BSO variants under study, the new and current sets of ideas were combined in one set so as to apply the stochastic ranking and choose the first N ideas to keep the set size fixed at each iteration.

The parameter setting used for each algorithm is shown in Table II

Algorithm 2 Stochastic Ranking.

$x_j = j \forall j \in \{1, \ldots, \lambda\}$
for i=1 to N **do**
  for j=1 to λ -1 **do**
    random u $\in$ U(0,1).
    if ($\phi(x_j)$ =0 $\phi(x_{j+1})$ = 0) or (u $< Pf$) **then**
      if $f(x_j) > f(x_{j+1})$ **then**
        swap(x_j, x_{j+1})
      else if $\phi(x_j) > \phi(x_{j+1})$ **then**
        swap(x_j, x_{j+1})
      end if
    end if
  end for
  if no swap done, break
end for

TABLE II
PARAMETER SETTING. THE VALUES WERE TAKEN FROM THE ORIGINAL REFERENCES OF EACH BSO VERSION [7], [8], [9] AND ALSO FROM THE REFERENCES OF THE CONSTRAINT-HANDLING TECHNIQUES [15], [16].

Algorithm	Parameter	Value
BSO, MBSO, SMBSO	N	100
	M	5
	Max-iter	500,000
BSO	k	20
	μ	0
	σ	1
MBSO	pr	0.005
SMBSO	p-one-center	N(0.4,0.1)
BSO , MBSO	p-replace	0.2
	p-one	0.8
	p-one-center	0.4
	p-two-center	0.5
ε-constrained	cp	0.5
stochastic ranking	Pf	0.45
Test Problems	CEC2006 [19]	

TABLE I

TEST PROBLEMS USED IN THE EXPERIMENTS. "n" IS THE NUMBER OF VARIABLES OF THE PROBLEM, "LI" IS THE NUMBER OF LINEAR INEQUALITY CONSTRAINTS, "NI" IS THE NUMBER OF NONLINEAR INEQUALITY CONSTRAINTS, "LE" IS THE NUMBER OF LINEAR EQUALITY CONSTRAINTS, "NE" IS THE NUMBER OF NONLINEAR INEQUALITY CONSTRAINTS, "a" IS THE NUMBER OF ACTIVE CONSTRAINTS AND "ρ" IS THE ESTIMATED SIZE OF THE FEASIBLE REGION WITH RESPECT TO THE WHOLE SEARCH SPACE [18]

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%	2	0	0	0	1
g03	10	nonlinear	0.0026%	0	0	0	1	1
g04	5	quadratic	27.0079%	4	2	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%	6	0	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	2	3
g14	10	nonlinear	0.0000%	0	0	3	0	3
g15	3	quadratic	0.0000%	0	0	1	1	2
g16	5	nonlinear	0.0204%	4	34	0	0	4
g17	6	nonlinear	0.0000%	0	0	0	4	4
g18	9	quadratic	0.0000%	0	13	0	0	6
g19	15	nonlinear	33.4761%	0	5	0	0	0
g20	24	linear	0.0000%	0	6	2	12	16
g21	7	linear	0.0000%	0	1	0	5	6
g22	22	linear	0.0000%	0	1	8	11	19
g23	9	linear	0.0000%	0	2	3	1	6
g24	2	linear	79.6556%	0	2	0	0	2

V. RESULTS AND DISCUSSION

The results of experiment 1 are presented in Tables III, IV and V, while the results of experiment 2 are shown in Table VI. In all of those tables the first column is for the name of the test problem, and the next three include the results obtained by each BSO version with each constraint-handling technique, ε-constrained method (ε), Feasibility Rules (FR) and Stochastic Ranking (SR). The last column presents the result of the statistical test (ST), where (=) means no significant difference between the reference algorithm and the compared algorithm, (+) means that the reference algorithm outperforms the algorithm compared and (-) means that the compared algorithm outperforms the reference algorithm. An empty space in the tables means that no feasible solutions were found for that test problem (columns 2-4). In those cases, statistical tests were not reported. The best results from the samples of runs are marked in bold. Test problems omitted are those where no BSO version was able to find feasible solutions in any single run.

For experiment 1, the BSO versions with the ε-constrained method were taken as reference algorithms for the statistical tests (column 2 in Tables III, IV, and V) because those combinations provided better results in the samples of independent runs. For experiment 2, MBSO (column 2 in Table VI) was used as the reference algorithm for similar reasons.

A. Experiment 1

Regarding the results for BSO in Table III, the Rank Sum Wilcoxon test indicates almost a tie between ε and FR i.e., similar results were obtained in fifteen out of nineteen test problems, ε outperformed FR in one test problem, while FR outperformed ε in two problems. ε was clearly better than SR because better results were obtained in fifteen test problems, similar results in two and SR was better than ε in just one.

TABLE III

MEDIANS REPORTED BY BSO WITH EACH CONSTRAINT-HANDLING TECHNIQUE. ε IS TAKEN AS A REFERENCE OVER FR AND SR TO SHOW THE STATISTICAL TEST (ST) RESULTS, WHERE (=) MEANS NO SIGNIFICANT DIFFERENCE BETWEEN THE REFERENCE ALGORITHM AND THE COMPARED ALGORITHM, (+) MEANS THAT THE REFERENCE ALGORITHM OUTPERFORMS THE ALGORITHM COMPARED AND (-) MEANS THAT THE COMPARED ALGORITHM OUTPERFORMS THE REFERENCE ALGORITHM.

		BSO		
Functions	ε	FR	SR	ST
g01	**-13.7259834**	-13.4356792	-12.1210574	(=)(+)
g02	-0.78487125	**-0.78544357**	-0.75745431	(=)(+)
g03	-0.28321891	-0.21301742	**-0.28573573**	(=)(=)
g04	**-30665.2927**	-30665.0269	-30653.4706	(=)(+)
g05	5294.00651	**5178.08668**		
g06	-6961.46576	**-6961.7764**	-6939.50205	(-)(+)
g07	**24.6005362**	24.6343637	27.2056687	(=)(+)
g08	-0.09582504	-0.09582504	-0.09582504	(=)(=)
g09	680.666694	**680.663085**	681.246018	(=)(+)
g10	**13318.7426**	14411.6122	18149.3709	(-)(+)
g11	**0.74997285**	0.7499743	0.75171115	(=)(+)
g12	-1	-1	-1	(=)(+)
g13	**0.61567416**	0.85296517	0.99999849	(+)(+)
g14	-43.6473924	**-43.6896006**	-42.7105425	(=)(-)
g15	**961.729572**	961.910793	967.332323	(=)(+)
g16	**-1.90423112**	-1.90402015	-1.88560212	(=)(+)
g18	-0.86579247	**-0.86599032**	-0.820099	(=)(+)
g19	**62.1777691**	63.8393044	84.1350502	(=)(+)
g23		**-97.9634101**		
g24	-5.50801327	-5.50801327	-5.50656964	(=)(+)

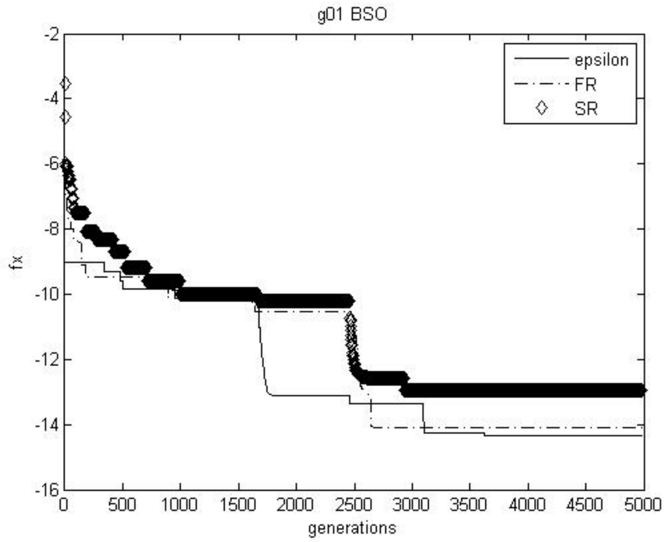

Fig. 1. Convergence graph for function g01 using BSO with the three constraint-handling methods.

In Figure 1 the convergence graphs for the three BSO versions in representative test problem g01 are plotted. As it can be seen, ε is able to avoid feasible local optimum and maintain improvement late in the search. This behaviour contrasts with the obtained by the other two constraint-handling techniques.

Regarding MBSO, the statistical test results in Table IV indicate that ε outperformed FR in eleven out of twenty two test problems, similar results were found in nine problems and FR was not better in any single test problem. With respect to SR, ε outperformed it in thirteen test problems, similar results were obtained in five and SR was better in just two test problems.

The convergence plots for the three MBSO versions are in Figure 2 for representative test problem g01. ε was able to converge faster to better solutions. On the other hand, FR and SR required significantly more time to get to such good result.

Finally, Table V presents a similar pattern of results for SMBSO with respect to MBSO, where ε outperformed FR in thirteen out of twenty one test problems, similar results were found in five and FR was better in just one test problem. ε outperformed SR fifteen test problems, similar results were obtained in four and SR could not outperform ε in any single test problem.

A similar behaviour as that observed for MBSO is presented now for SMBSO in Figure 3 for test problem g01. ε was able to converge faster than their two counterparts (the ε plot is close to the x-axis in Figure 3).

An important finding of experiment 1 was the fact that the ε-constrained method was the most competitive when used with the three BSO versions. Furthermore, it promoted a suitable and faster convergence to competitive results. It remains to be seen which BSO version is more competitive and this issue is tackled in the next experiment.

TABLE IV

MEDIANS REPORTED BY MBSO WITH EACH CONSTRAINT-HANDLING TECHNIQUE. ε IS TAKEN AS A REFERENCE OVER FR AND SR TO SHOW THE STATISTICAL TEST (ST) RESULTS, WHERE (=) MEANS NO SIGNIFICANT DIFFERENCE BETWEEN THE REFERENCE ALGORITHM AND THE COMPARED ALGORITHM, (+) MEANS THAT THE REFERENCE ALGORITHM OUTPERFORMS THE ALGORITHM COMPARED AND (-) MEANS THAT THE COMPARED ALGORITHM OUTPERFORMS THE REFERENCE ALGORITHM.

		MBSO		
Functions	ε	**FR**	**SR**	**ST**
g01	-15	-15	-15	(=)(=)
g02	**-0.72905855**	-0.72195232	-0.70338773	(=)(=)
g03	**-1.00029549**	-0.99932567	-0.99689488	(+)(+)
g04	-30665.5387	-30665.5387	-30665.5387	(=)(-)
g05	**5126.57055**	5182.01052	5220.21686	(+)(+)
g06	-6961.81388	-6961.81388	-6961.81387	(=)(+)
g07	**24.3420728**	24.4072619	24.5270199	(+)(+)
g08	-0.09582504	-0.09582504	-0.09582504	(=)(=)
g09	**680.633326**	680.639841	680.652859	(+)(+)
g10	**7148.88021**	7267.76964	7327.29121	(=)(+)
g11	**0.74990035**	0.7546776	0.76300509	(+)(+)
g12	-1	-1	-1	(=)(=)
g13	**0.05395068**	0.95770084	0.88762401	(+)(+)
g14	**-47.6011839**	-43.3648135	-42.9230471	(+)(+)
g15	**961.716101**	964.030798	964.857912	(+)(+)
g16	-1.90515526	-1.90515526	-1.90515526	(=)(+)
g17	**8853.57277**	8957.76168	8957.47279	(+)(+)
g18	**-0.8658459**	-0.8643457	-0.86453845	(+)(-)
g19	**33.1293739**	36.4198197	40.4967993	(+)(+)
g21		329.575325		
g23		78.705459	242.962969	
g24	-5.50801327	-5.50801327	-5.50801327	(=)(=)

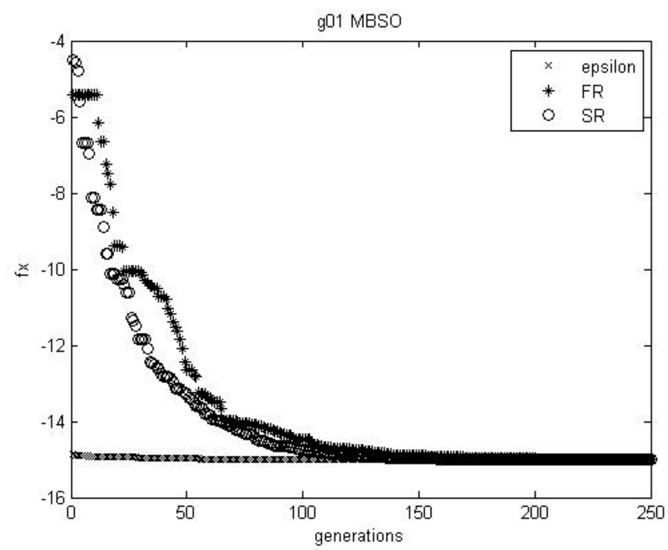

Fig. 2. Convergence graph for function g01 using MBSO and the three constraint-handling methods. Only the first 250 generations are plotted to identify the differences for each MBSO version.

B. Experiment 2

Table VI presents the comparison of the three BSO versions, all of them with the ε-constrained method as constraint-handler. MBSO, the reference algorithm for the Wilcoxon statistical test, outperformed SMBSO in seven test problems, reached similar results in eleven and was not outperformed by

TABLE V

MEDIANS REPORTED BY SMBSO WITH EACH CONSTRAINT-HANDLING TECHNIQUE. ε IS TAKEN AS A REFERENCE OVER FR AND SR TO SHOW THE STATISTICAL TEST (ST) RESULTS, WHERE (=) MEANS NO SIGNIFICANT DIFFERENCE BETWEEN THE REFERENCE ALGORITHM AND THE COMPARED ALGORITHM, (+) MEANS THAT THE REFERENCE ALGORITHM OUTPERFORMS THE ALGORITHM COMPARED AND (-) MEANS THAT THE COMPARED ALGORITHM OUTPERFORMS THE REFERENCE ALGORITHM.

		SMBSO		
Functions	ε	**FR**	**SR**	**ST**
g01	**-14.9999992**	-15	-14.9993215	(=)(+)
g02	-0.67133279	**-0.71850795**	-0.69465066	(-)(=)
g03	**-1.00027242**	-0.99144829	-0.67607676	(+)(+)
g04	**-30665.5387**	-30665.431	-30654.787	(+)(+)
g05	**5126.75765**	5205.27027	5346.15283	(+)(+)
g06		-6659.64967	-6619.74874	
g07	**24.3630671**	24.5280728	25.1630264	(+)(+)
g08	-0.09582504	-0.09582504	-0.09582504	(+)(+)
g09	**680.636058**	680.649006	680.71517	(+)(+)
g10	7322.8129	**7309.84216**	7525.8942	(=)(=)
g11	**0.7499004**	0.75636641	0.76383497	(+)(+)
g12	-1	-1	-1	(=)(=)
g13	**0.05394833**	0.86163753	0.92297506	(+)(+)
g14	**-47.6608268**	-41.9986648	-42.523399	(+)(+)
g15	967.519963	965.925127	**965.770638**	(=)(=)
g16	**-1.90514953**	-1.90454387	-1.83348496	(+)(+)
g17	**8855.91728**	8954.23605	8955.62533	(+)(+)
g18	**-0.8658068**	-0.86331758	-0.77768727	(=)(+)
g19	**33.7540853**	39.8877045	52.0551162	(+)(+)
g23		119.152469	145.598656	
g24	**-5.50801327**	-5.50798652	-5.5002577	(+)(+)

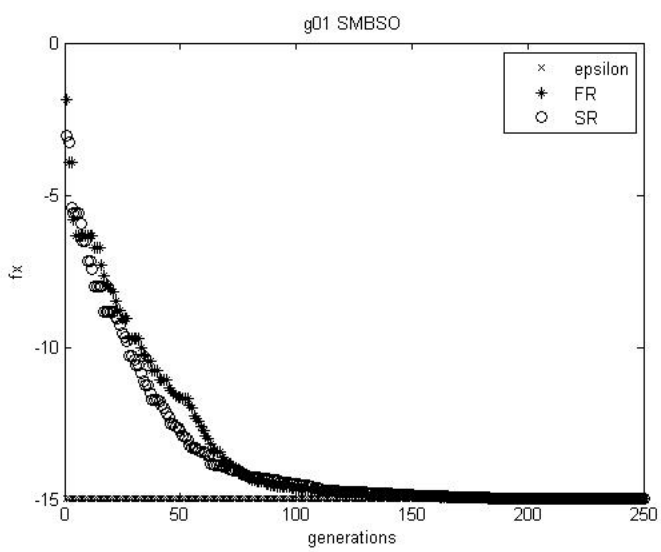

Fig. 3. Convergence graph for function g01 using SMBSO and the three constraints-handling methods. Only the first 250 generations are plotted to identify the differences for each SMBSO version.

SMBSO in any test problem. Moreover, MBSO outperformed BSO in twelve test problems, obtained similar results in five and was outperformed by the BSO in just one test problem.

Figure 4 presents the convergence plots for the three BSO versions with the ε-constrained method. A clear fast convergence was obtained by MBSO and SMBSO (both algorithms have their plot close to the x-axis). In contrast, BSO needed much more time to converge and it was indeed trapped in a local optimum solution.

The main finding of experiment 2 is that the combination of MBSO and the ε-constrained method provided the most competitive results in the set of constrained problems adopted in this work. Furthermore, an interesting observation is that SMBSO, being the most simple of the BSO versions, and with less parameters to calibrate by the user, presented a very competitive behavior. Therefore, it can be a BSO version which deserves further analysis.

TABLE VI

MEDIANS OBTAINED FOR MBSO, SMBSO AND BSO USING ε-CONSTRAINED AS A CONSTRAINT-HANDLING METHOD. MBSO IS TAKEN AS REFERENCE OVER SMBSO AND BSO TO SHOW THE STATISTICAL TEST (ST) RESULTS, WHERE (=) MEANS NO SIGNIFICANT DIFFERENCE BETWEEN THE REFERENCE ALGORITHM AND THE COMPARED ALGORITHM, (+) MEANS THAT THE REFERENCE ALGORITHM OUTPERFORMS THE ALGORITHM COMPARED AND (-) MEANS THAT THE COMPARED ALGORITHM OUTPERFORMS THE REFERENCE ALGORITHM.

		MBSO, SMBSO and BSO		
Functions	**MBSO**	**SMBSO**	**BSO**	**ST**
g01	**-15**	-14.99999925	-13.72598342	(+)(+)
g02	-0.729058554	-0.671332787	**-0.784871252**	(+)(-)
g03	**-1.000295492**	-1.000272424	-0.283218907	(=)(+)
g04	-30665.53867	**-30665.53867**	-30665.29271	(+)(+)
g05	**5126.570551**	5126.757652	5294.006507	(=)(+)
g06	**-6961.813876**		-6961.46576	
g07	**24.34207279**	24.36306708	24.60053615	(=)(+)
g08	-0.095825041	-0.095825041	-0.095825041	(+)(=)
g09	**680.6333264**	680.6360583	680.6666943	(=)(+)
g10	**7148.880214**	7322.812904	13318.74261	(=)(+)
g11	**0.749900347**	0.749900398	0.749972845	(=)(+)
g12	-1	-1	-1	(=)(=)
g13	0.053950678	**0.053948325**	0.615674157	(=)(+)
g14	-47.60118387	**-47.6608268**	-43.64739239	(=)(+)
g15	**961.7161006**	967.5199633	961.7295719	(+)(=)
g16	**-1.905155259**	-1.905149534	-1.904231122	(+)(+)
g17	**8853.572769**	8855.917276		
g18	**-0.865845899**	-0.8658068	-0.865792468	(=)(=)
g19	**33.12937392**	33.75408534	62.17776907	(=)(+)
g24	-5.508013272	-5.508013272	-5.508013272	(+)(=)

VI. CONCLUSION AND FUTURE WORK

Three constraint-handling techniques were used in this research with the aim to analyse their effects on three recently BSO versions when dealing with constrained search spaces. Two experiments were carried out. In the first, the effects of the three constraint-handling techniques over each BSO version were analysed. The ε-constrained method was the most competitive constraint-handler for the three BSO versions under study. Based on that finding, in the second experiment, the three BSO versions, all with the ε-constrained method, were compared. The results pointed MBSO as the most competitive version, while SMBSO, the most simple version compared, was the runner-up.

From the findings obtained in this research, MBSO and SMBSO will be analysed by using other measures related with their capabilities to reach and sample the feasible region. Furthermore, scalable test problems will be solved to analyse the effects of the dimensionality in their performance.

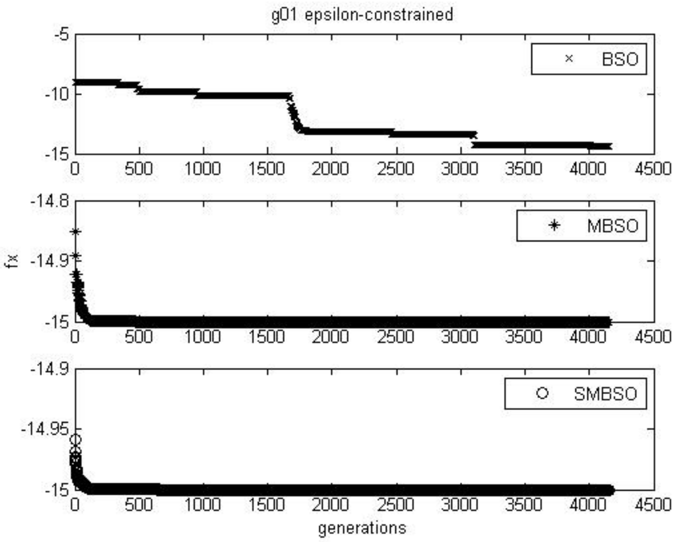

Fig. 4. Convergence graph for function g01 using ε-constrained as constraint-handling technique on the three BSO versions.

ACKNOWLEDGMENT

The first author acknowledges support from CONACyT and the University of Veracruz to pursue graduate studies at the Artificial Intelligence Research Center. The second author acknowledges support from CONACyT project No. 220522.

REFERENCES

[1] E. Mezura-Montes and C. A. Coello-Coello., "Evolutionary algorithms for constrained engineering problems," *Swarm and Evolutionary Computation*, vol. 1, pp. 173–194, 2011.

[2] T. Takahama and S. Sakai, "Efficient constrained optimization by the epsilon constrained differential evolution with rough approximation using kernel regression," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, June 2013, pp. 1334–1341.

[3] D. Swagatam, B. Arijit, D. Sambarta, and A. Ajith, "Bacterial foraging optimization algorithm: Theoretical foundations, analysis, and applications," in *Foundations of Computational Intelligence Volume 3*, ser. Studies in Computational Intelligence, A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht, Eds. Springer Berlin Heidelberg, 2009, vol. 203, pp. 23–55. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01085-9-2

[4] D. Karaboga and B. Basturk, "Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems," in *Foundations of Fuzzy Logic and Soft Computing*, ser. Lecture Notes in Computer Science, P. Melin, O. Castillo, L. Aguilar, J. Kacprzyk, and W. Pedrycz, Eds. Springer Berlin Heidelberg, 2007, vol. 4529, pp. 789–798. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72950-1-77

[5] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2006.

[6] C. Shu-Chuan, H. Hsiang-Cheh, R. JohnF., and P. Jeng-Shyang, "Overview of algorithms for swarm intelligence," in *Computational Collective Intelligence. Technologies and Applications*, ser. Lecture Notes in Computer Science, P. Jedrzejowicz, N. Nguyen, and K. Hoang, Eds. Springer Berlin Heidelberg, 2011, vol. 6922, pp. 28–41. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23935-9-3

[7] S. Yuhui, "Brain storm optimization algorithm," in *Advances in Swarm Intelligence*, ser. Lecture Notes in Computer Science, Y. Tan, Y. Shi, Y. Chai, and G. Wang, Eds. Springer Berlin Heidelberg, 2011, vol. 6728, pp. 303–309. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21515-5-36

[8] Z. h. Zhan, J. Zhang, Y. h. Shi, and H. lin Liu, "A modified brain storm optimization," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, June 2012, pp. 1–8.

[9] Z. H. Zhan, W. N. Chen, Y. Lin, Y. J. Gong, Y. L. Li, and J. Zhang, "Parameter investigation in brain storm optimization," in *Swarm Intelligence (SIS), 2013 IEEE Symposium on*, April 2013, pp. 103–110.

[10] A. R. Jordehi, "Brainstorm optimisation algorithm (bsoa): An efficient algorithm for finding optimal location and setting of {FACTS} devices in electric power systems," *International Journal of Electrical Power & Energy Systems*, vol. 69, pp. 48 – 57, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142061515000083

[11] X. Jingqian, W. Yali, S. Yuhui, and C. Shi, "Brain storm optimization algorithm for multi-objective optimization problems," in *Proceedings of the Third International Conference on Advances in Swarm Intelligence - Volume Part I*, ser. ICSI'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 513–519. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30976-2-62

[12] C. Junfeng, C. Shi, C. Yang, X. Yingjuan, and S. Yuhui, "Enhanced brain storm optimization algorithm for wireless sensor networks deployment," in *Advances in Swarm and Computational Intelligence*, ser. Lecture Notes in Computer Science, Y. Tan, Y. Shi, F. Buarque, A. Gelbukh, S. Das, and A. Engelbrecht, Eds. Springer International Publishing, 2015, vol. 9140, pp. 373–381. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-20466-6-40

[13] R. Smith, *The 7 Levels of Change*, 2nd Edition, Ed. Tapeslry Press, 2002.

[14] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311 – 338, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782599003898

[15] T.Takahama and S. Sakai, "Constrained optimization by the epsilon constrained differential evolution with an archive and gradient-based mutation." *WCCI 2010 IEEE World Congress on Computational Intelligence July, 18-23, 2010 - CCIB, Barcelona, Spain*, 2010.

[16] T. Runarsson and Y. X, "Stochastic ranking for constrained evolutionary optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 4, no. 3, pp. 284–294, Sep 2000.

[17] J. J. Liang, T. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. A. Coello Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," Nanyang Technological University, Singapore, Tech. Rep., December, 2005, available at: http://www.lania.mx/˜emezura.

[18] Z. Michalewicz and M. Schoenauer, "Evolutionary Algorithms for Constrained Parameter Optimization Problems," *Evolutionary Computation*, vol. 4, no. 1, pp. 1–32, 1996.

[19] J. J. Liang, T. P. Runarsson, E. Mezura-Montes., M. Clerc, P. N. Suganthan, and C. A. C.-C. K. Deb, "Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization," *Technical Report*, September 18, 2006.