Empirical Analysis of a Modified Artificial Bee Colony for Constrained Numerical Optimization

Efrén Mezura-Montes* and Omar Cetina-Domínguez

Laboratorio Nacional de Informática Avanzada (LANIA A.C.)
Rébsamen 80, Centro, Xalapa, Veracruz, 91000, MÉXICO
Tel.: +52-228-8416100
Fax: +52-228-8416101
emezura@lania.mx, omarcetina@hotmail.com
** Corresponding author*

Abstract

A modified artificial bee colony algorithm to solve constrained numerical optimization problems is presented in this paper. Four modifications related with the selection mechanism, the scout bee operator, and the equality and boundary constraints are made to the algorithm with the aim to modify its behavior in a constrained search space. Six performance measures found in the specialized literature are employed to analyze different capabilities in the proposed algorithm such as the ability and cost to generate feasible solutions, the capacity and cost to locate the feasible global optimum solution and the competency to improve feasible solutions. Three experiments, including a comparison with state-of-the-art algorithms, are considered in the test design where twenty four well-known benchmark problems with different features are utilized. The overall results show that the proposed algorithm differs in its behavior with respect to the original artificial bee colony algorithm but its performance is improved, mostly in problems with small feasible regions due to the presence of equality constraints.

Keywords: Swarm Intelligence, Evolutionary Algorithms, Constraint-Handling, Global Optimization

1. Introduction

Swarm Intelligence algorithms (SIAs) [1], besides evolutionary algorithms (EAs) [2] have become a popular option to solve optimization problems.

Among those problems, the one tackled in this paper is the constrained nonlinear optimization problem (CNOP), which, without loss of generality, can be defined as to:

Find x which minimizes

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

subject to

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

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

where $x \in \mathbb{R}^n$ is the vector of solutions $x = [x_1, x_2, \dots, x_n]^T$ and each $x_i, \ i = 1, ..., n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$ which define the search space $\mathcal{S}$, $\mathcal{F}$comprises the set of all solutions which satisfy the constraints of the problems and it is called the feasible region; m is the number of inequality constraints and p is the number of equality constraints. Both, the objective function and the constraints can be linear or nonlinear. To handle equality constraints they are transformed into inequalities constraints as follows: $|h_j(x)| - \epsilon \leq 0$, where ϵ is the tolerance allowed (a very small value).

In a similar way to EAs, SIAs in their original versions lack a mechanism to deal with the constraints (Equations 2 and 3) in a CNOP [3]. Therefore, there are different mechanisms which can be added to those algorithms in order to bias the search to the feasible region $\mathcal{F}$ of the search space $\mathcal{S}$.

Most of those adapted versions to solve CNOPs are based on EAs [4, 5]. Regarding SIAs, most of the work has been focused in particle swarm optimization (PSO) [6, 7, 8, 9, 10]. In this way, the usage and analysis of other SIAs to solve CNOPs are still scarce and this is the case of those based on honey bee behaviors.

Several approaches based on those insects have been designed to deal with combinatorial optimization problems [11, 12, 13]. However, there are some proposals focused on unconstrained continuous search spaces.

The most popular behaviors taken as sources of inspiration for bee-based algorithms to deal with numerical optimization problems are the foraging behavior (the most popular) and the mating behavior.

Among those algorithms based on the foraging behavior, the Bees Algorithm (BA) [14] considers bees located in random solutions. The search is biased by fitness into the “the elite sites” i.e. the best set of solutions in the population. In this way, more bees exploit those promising regions. The remaining sites are visited with a lower frequency. To promote diversity, scout bees perform a random-search-like process. The use of a shrinking mechanism for the sites (called patches) was added to BA in [15] and that new version was called Improved Bees Algorithm (IBA).

The Virtual Bee Algorithm (VBA) proposed in [16] considers a communication process based on waggle dances based on fitness from bees located in food sources to other bees located in the neighborhood.

One of the most popular algorithms based on the foraging behavior is the Artificial Bee Colony (ABC) [17] which was proposed to deal with unconstrained nonlinear optimization problems. ABC's selection criterion was changed as to tackle CNOPs in [18]. Moreover, some possible mechanisms to further improve ABC to solve CNOPs were initially sketched in [19]. Those preliminary ideas are the base of this work.

ABC [20] has been also updated to improve its performance in unconstrained search spaces as in the Interactive Artificial Bee Colony (IABC) [21], where a modification inspired in the Newtonian law of universal gravitation aimed to improve the exploration ability of the ABC.

Regarding the mating behavior, the Honey Bee Mating Optimization (HBMO) algorithm [22] utilizes a set of queen bees i.e., the best solutions in the population, which are recombined with randomly generated solutions i.e., drones, by using a fitness-based probabilistic function which considers the velocity and energy of that queen bee during its mating flight. From this recombination process, a set of broods is obtained, and they are improved by worker bees (different heuristic operators). The HBMO algorithm has been sparingly employed to solve CNOPs [22]. However, results are reported in just one test problem with only two variables ($n = 2$) by using a penalty function. HBMO required more than one million evaluations to reach competitive results in this single problem.

As noted in the review of the specialized literature previously detailed, the design of bee-based algorithms to solve CNOPs is scarce. Furthermore, to the best of the authors' knowledge, there are not empirical studies on the behavior of those algorithms when solving CNOPs (just final results are usually reported). Motivated for those reasons, this paper presents a modified ABC algorithm to improve its capabilities to solve CNOPs. Furthermore, an in-depth analysis of this proposed algorithm with respect to the original ABC for constrained optimization [18] is presented. The comparison considers performance measures proposed to analyze the behavior of meta-heuristic algorithms in constrained numerical search spaces. The aim is to distinguish the way both algorithms solve this type of problems. Finally, a comparison of the final results obtained by the modified ABC is compared against state-of-the-art approaches.

The paper is organized as follows: Section 2 describes the original ABC algorithm, while Section 3 details each modification made to the original ABC to deal with CNOPs and a pseudocode of this version is included as well. After that, in Section 4 the experimental design and the performance measures employed are presented and the obtained results are shown and discussed. Finally, some conclusions are drawn in Section 5, where the future work is also included.

2. Artificial Bee Colony

ABC is based in two natural processes: The recruitment of bees into a food source and the abandonment of a source [20].

An important difference between ABC and other swarm intelligence algorithms is that in the ABC, the solutions of the problem are represented by the food sources, not by the bees. In contrast, bees act as variation operators able to discover (generate) new food sources based on the existing ones. Three types of bees are considered in the ABC: employed, onlooker and scout bees. The number of employed bees is equal to the number of food sources and each employed bee is assigned to one of the sources. Upon reaching the source, the bee will calculate a new solution (fly to another nearby food source) from it and retain the best solution (in a greedy selection). The number of onlooker bees is also the same that the 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. When a source does not improve after a certain number of iterations, it is abandoned and replaced by those found by a scout bee, which involves calculating a new solution at random. A graphical representation of the model can be found in Figure 1

Three user-defined parameters are required by the ABC: the number of solutions (food sources) SN, the total number of cycles (iterations) of the algorithm MCN, and the number of cycles that a non improved solution will be kept before being replaced by a new solution generated by the scout bee mechanism *limit*. ABC uses real-encoding for the solutions of the optimization problem. The selection mechanism used in ABC occurs when the employed bees share the information of their food sources in the hive by waggle dances, emulated as a fitness proportional selection, very similar to a roulette wheel technique used in EAs [23]. The replacement process in ABC takes place in two phases: (1) in the greedy selection between the current food source and the new one generated by either an employed or an onlooker bee and (2) when an employed bee leaves a food source which could not be improved within *limit* cycles and a scout bee generates a new one by using a random process. The variation operator used by both, employed and onlooker bees, aims to generate a new candidate solution $v_{i,j}$ by using the formula given in 4:

$$v^g_{i,j} = x^g_{i,j} + \phi_j \cdot \left(x^g_{i,j} - x^g_{k,j}\right) \tag{4}$$

where x^g_i represents the solution in which the bee is located at that moment in cycle g, x^g_k is a randomly chosen food source (and different from x^g_i), $i \in \{1, 2, \ldots, SN\}$, $j \in \{1, 2, \ldots, n\}$ and ϕ_j is a random real number within [-1, 1] generated at random for every $j \in \{1, 2, \ldots, n\}$.

The detailed pseudocode of the ABC algorithm is presented in Figure 2. The process begins with the initial set of solutions generated at random and their evaluations (steps $1-3$ in Figure 2). After that, a cycle that repeats MCN times starts. Within each cycle there are two inner loops. The first one considers the generation of new solutions by the employed bees by using their assigned food sources (steps $6-14$ in Figure 2). After that, the second nested loop includes the selection of those better food sources by the onlooker bees and the generation of new solutions (steps $15-22$ in Figure 2). Finally, the scout bees replace those abandoned solutions by generating new ones (step 23 in Figure 2).

The ABC described above has been tested in different types of unconstrained optimization problems [24, 15] and there is an adaptation to solve CNOPs in [18]. The approach kept the same structure ABC has. However, the greedy selection mechanism was modified and three rules based on feasibility, proposed by Deb [25] to be used in an EA, were added to ABC. Furthermore, a recombination mechanism between the original food source and that generated by the bee in the ABC was included. This operator worked as indicated in Equation 5 where a new parameter $0 \leq MR \leq 1$ was considered.

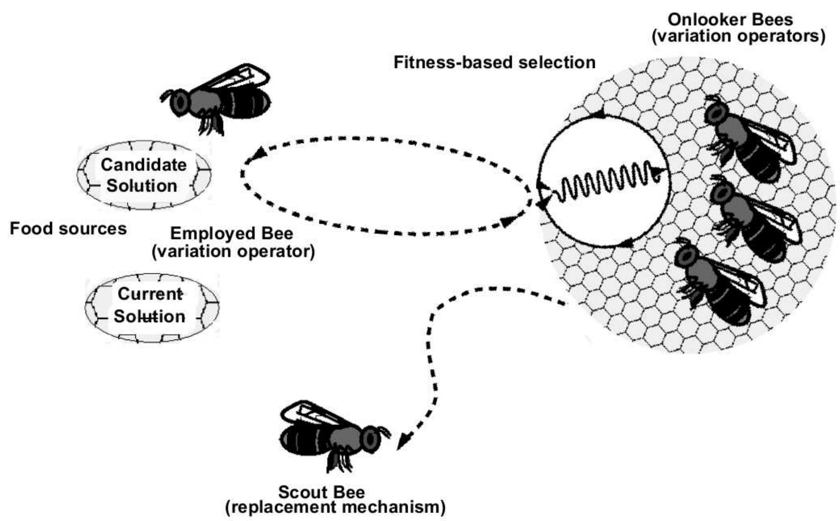

Figure 1: Graphical representation of the elements in the ABC algorithm.

$$v^g_{i,j} \begin{cases} x^g_{i,j} + \phi_j \cdot \left(x^g_{i,j} - x^g_{k,j}\right) & \text{, if rand(0,1)} < MR \\ x^g_{i,j} & \text{, otherwise} \end{cases} \tag{5}$$

Finally, the scout bee was activated at certain periods of time, regardless the presence of abandoned food sources because the aim was to avoid premature convergence.

3. Modified ABC

As it was pointed out in the last part of Section 2, the modification proposed for the ABC in [18] to consider a constrained search space was focused in the greedy selection between pairs of food sources. On the other hand, in [19] a new way of working for the scout bee was proposed. By considering those previous ideas, a modified ABC (M-ABC) is detailed in this section. The four mechanisms added to the original ABC are explained below:

BEGIN
Initialize the set of food sources $x_i^0,\ i = 1, \ldots, SN$
Evaluate each $x_i^0,\ i = 1, \ldots, SN$
$g = 1$
REPEAT
FOR $i = 1$ **TO** SN
Generate v_i^g with x_i^{g-1} by using Eq. (4)
Evaluate v_i^g
IF v_i^g is better than x_i^{g-1}
$x_i^g = v_i^g$
ELSE
$x_i^g = x_i^{g-1}$
END IF
END FOR
FOR $i = 1$ **TO** SN
Select, based on fitness proportional selection food source x_l^g
Generate v_l^g with x_l^g by using Eq. (4)
Evaluate v_l^g
IF v_l^g is better than x_l^g
$x_l^g = v_l^g$
END IF
END FOR
Generate new food sources at random for those whose limit to be improved has been reached
Keep the best solution so far
$g = g + 1$
UNTIL $g = MCN$
END

Figure 2: Artificial Bee Colony algorithm.

3.1. Tournament selection

As indicated in Section 2, the information sharing process (waggle dances) from employed to onlooker bees in the hive is emulated in the original ABC with a fitness proportional selection. In contrast, in this modified version, tournament selection was employed. The motivation was two-fold: (1) from a biological point of view the communication via a waggle dance of an employed bee can be seen only by onlooker bees in its neighborhood (i.e., only a subset of the onlooker bees participate in each selection step) and (2) it is important to maintain a low selection pressure and this type of selection helps to keep it that way. The set of three feasibility rules defined by Deb [25] were used as selection criteria in the tournament:

1. Between two feasible food sources, the one with the best objective function value is preferred.
2. Between a feasible food source and an infeasible food source, the feasible one is preferred.
3. Between two infeasible food sources, the one with the lowest value of the sum of constraint violation is preferred.

In this way, two food sources compete for the chance to be modified by an onlooker bee based on those aforementioned criteria. As a result, no probabilities or modified fitness values must be computed as in the original ABC [17].

3.2. Dynamic tolerance for equality constraints

Equality constraints are difficult to satisfy because they define a very small feasible region [26]. Therefore it is a common practice to rewrite the equality constraints as inequality constraints as recalled from Section 1, where a tolerance ϵ is defined. This (usually) small value slightly extends the region of the search space where the corresponding constraint is satisfied, as detailed in Figure 3.

One alternative to deal with the ϵ value is to fix it during the search [27]. On the other hand, different control mechanisms [28] can be employed to conveniently vary its value within the optimization process [26]. In this modified ABC algorithm the second option was chosen to explore its performance in the modified ABC. Moreover, a fixed value was employed by the original ABC [18].

A dynamic parameter control mechanism, inspired in [29], is proposed as defined in Equation 6:

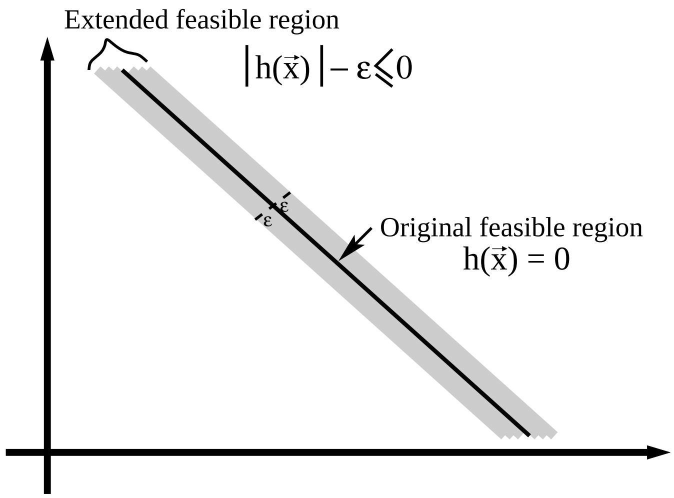

Figure 3: Effect of the ϵ tolerance value in an equality constraint.

$$\epsilon(g+1) = \frac{\epsilon(g)}{dec} \tag{6}$$

where g is the current cycle and dec is the decreasing rate value of each cycle ($dec > 1$). The goal is to start with a larger feasible region than the original one, i.e, equality constraints will be easier to satisfy at the beginning of the process. As the algorithm progresses through the cycles, this tolerance is reduced so that the constraint violation of the obtained solutions is lower than those of the solutions calculated in the first cycles.

3.3. Smart flight operator

Recalling from Section 2, in the original ABC algorithm the scout bee replaces solutions which were not improved in a predefined number of cycles by newly ones generated at random. However, based on the fact that in a CNOP the feasible region is usually small with respect to the whole search space [30], it is hard to randomly generate either a feasible solution or an infeasible solution close to the feasible region. Therefore, with the goal to generate more convenient attractors by the scout bees, a mechanism originally proposed in [31] to be used in particle swarm optimization, was adapted in [19] as follows: A new solution v_i^g is generated by the scout bee with the help of the food source to be replaced x_i^g, which is utilized as a base to create a search direction defined by the best solution in the current population x_B^g and a randomly chosen solution x_k^g. As a result, this new solution v_i^g is generated as indicated in Equation 7

$$v_{i,j}^g = x_{i,j}^g + \phi \cdot (x_{k,j}^g - x_{i,j}^g) + (1 - \phi) \cdot (x_{B,j}^g - x_{i,j}^g) \tag{7}$$

The mechanism in Equation 7, called smart-flight, combines three key elements: (1) the information of the solution to be replaced x_i^g, before its elimination, is used as a reference point to generate a new one (v_i^g), (2) the location of this new solution will be biased by the best solution x_B^g, which may lead to find a feasible solution or, at least, an infeasible solution closer to the feasible region, and (3) the presence of a randomly chosen solution x_k^g aims to keep the new solution to be fully attracted by the best solution so far.

3.4. Boundary constraint-handling

In a similar way to other real-encoding nature-inspired meta-heuristic algorithms for global optimization such as evolution strategies [32], differential evolution [33], and particle swarm optimization [34], the application of the variation operators may generate values outside the search space defined by the lower and upper limits defined per each variable of the problem (see Section 1). Therefore, a mechanism to fix those invalid values [35] is added to the employed, onlooker, and scout bees when they apply the operators defined in Equations 5 and 7 as indicated in Equation 8:

$$v_{i,j}^g = \begin{cases} 2 * L_j - v_{i,j}^g & , \text{if } v_{i,j}^g < L_j \\ 2 * U_j - v_{i,j}^g & , \text{if } v_{i,j}^g > U_j \\ v_{i,j} & , \text{otherwise} \end{cases} \tag{8}$$

where $v_{i,j}^g$ is the variable j of the candidate solution i at cycle g, L_j is the lower limit of the variable j and U_j is the upper limit of variable j. This proposal differs with that used in the original ABC, where the limit value, either L_j or U_j, is assigned when the newly generated value $v_{i,j}^g$ violates it.

If the boundary constraint-handling is used very often as in the original ABC, the solutions will be focused in the extreme values of the search space and the diversity will be affected. In contrast, the proposal added to the modified ABC aims to generate a diverse set of values each time it is used.

3.5. Pseudocode of the approach

The complete pseudocode of the M-ABC is presented in Figure 4. It is important to note that both, the employed and the onlooker bees utilize the operator defined in Equation 5, where a recombination-like mechanism between the current solution and the combination of the current solution with a randomly chosen one is computed.

4. Experiments and Results

To evaluate the performance and behavior of M-ABC and to also carry out a comparison with state-of-the-art algorithms, we used a set of well-known constrained optimization problems proposed in [27]. The Appendix of this paper includes the details of each test problem an a summary of their main features is included in Table 1.

In Table 1 "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 a metric to find the approximate size of the feasible area calculated with: $\rho = \frac{|\mathcal{F}|}{|\mathcal{S}|}$, where $|\mathcal{F}|$ is the number of feasible solutions and $|\mathcal{S}|$ is the total number of solutions randomly generated. Michalewicz and Schoenauer [4] suggested a total number of 1,000,000 solutions for $|\mathcal{S}|$.

4.1. Performance measures

Before describing the performance measures employed in this work, some definitions are provided:

- **Evaluation:** It is the calculation of the objective function value and the values of the constraints for a given solution. The total number of evaluations of an algorithm is considered in this work as a measure of computational cost.

- **Run:** It is an independent execution of an algorithm.

- **Feasible solution:** It is a solution that satisfies all the constraints of the problem.

- **Feasible run:** It is a run where at least one feasible solution was found.

1 **BEGIN**
2 Initialize the set of food sources $x_i^0,\ i = 1, \ldots, SN$
3 Evaluate each $x_i^0,\ i = 1, \ldots, SN$
4 $g = 1$
5 **IF** There are equality constraints
6 Initialize $\epsilon(g)$
7 **END IF**
8 **REPEAT**
9 **IF** There are equality constraints
10 Evaluate each $x_i^0,\ i = 1, \ldots, SN$ with $\epsilon(g)$
11 **END IF**
12 **FOR** $i = 1$ **TO** SN
13 Generate v_i^g with x_i^{g-1} by using Eq. (5)
14 Evaluate v_i^g
15 **IF** v_i^g is better than x_i^{g-1} (based on feasibility criteria in Section 3.1)
16 $x_i^g = v_i^g$
17 **ELSE**
18 $x_i^g = x_i^{g-1}$
19 **END FOR**
20 **FOR** $i = 1$ **TO** SN
21 Select food source x_l^g based on binary tournament selection (Section 3.1)
22 Generate v_l^g with x_l^g by using Eq. (5)
23 Evaluate v_l^g
24 **IF** v_l^g is better than x_l^g (based on feasibility criteria in Section 3.1)
25 $x_l^g = v_l^g$
26 **END IF**
27 **END FOR**
28 Apply the smart flight by the scout bees (Eq. 7) for those
solutions whose limit to be improved has been reached
29 Keep the best solution so far
30 $g = g + 1$
31 **IF** There are equality constraints
32 Update $\epsilon(g)$ by using Eq. (6)
33 **END IF**
34 **UNTIL** $g = MCN$
35 **END**

Figure 4: Modified Artificial Bee Colony algorithm (M-ABC). Steps marked with **boldface** were modified with respect to the original ABC.

Table 1: Summary of main characteristics of the test problems.

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

- **Successful solution:** It is a feasible solution with an objective function value $f(\vec{x})$ which is close to that of the best known or global optimal feasible solution $f(\vec{x}^*)$. This closeness is measured by a small tolerance on the difference between these two solutions $f(\vec{x}^*) - f(\vec{x}) \leq \delta$. A value of $\delta = 1E - 04$ was used in this work.

- **Successful run:** It is a run where at least one successful solution was found.

The following performance measures were utilized in the experiments:

1. **FP:** The feasibility probability consists on the number of feasible runs (fr) divided by the total number of runs performed (tr), as indicated in Equation 9.

$$FP = \frac{fr}{tr} \tag{9}$$

The range of values for FP goes from 0 to 1, where 1 means that all runs were feasible runs. In this way, a higher value is preferred.

2. **P:** The probability of convergence [36] is calculated by the ratio of the number of successful runs (sr) to the total number of runs performed (tr), as indicated in Equation 10.

$$P = \frac{sr}{tr} \tag{10}$$

Similar to FP, the range of values for P goes from 0 to 1, where 1 means that all runs were successful runs. Therefore, a higher value is preferred.

3. **AFES:** Average number of function evaluations [36]. It is calculated by averaging the number of evaluations required on each successful run to find the first successful solution, as indicated in Equation 11.

$$AFES = \frac{1}{sr} \sum_{i=1}^{sr} EVAL_i \tag{11}$$

where $EVAL_i$ is the number of evaluations required to find the first successful solution in successful run i. For $EVALS$, a lower value is preferred because it means that the average computational cost is lower for an algorithm to reach the vicinity of the feasible optimum solution.

4. **SP:** The two previous performance measures (P and $AFES$) are combined to measure the speed and reliability of a variant through a successful performance [36], as indicated in Equation 12.

$$SP = \frac{AFES}{P} \tag{12}$$

For this measure, a lower value is preferred because it means a better ratio between speed and consistency of the algorithm.
5. **EVALS:** Proposed in [37], the objective is to count the number of evaluations needed to found the first feasible solution in every run of the algorithm. For this performance measure, some statistical values (best, worst, mean and standard deviation) on a sample of runs are provided. Low values are preferred for this measure i.e. a low computational cost to find a feasible solution is preferred.
6. **PROGRESS RATIO:** Proposed in [30], the objective is to measure the improvement capability of the algorithm within the feasible region of the search space. For this measure high values are preferred because they indicate a higher improvement of the first feasible solution found. It is calculated as shown in Equation 13.

$$PR = \begin{cases} \left| ln\sqrt{\frac{f_{min}(G_{ff})}{f_{min}(MCN)}} \right| & \text{, if} \quad f_{min}(MCN) > 0 \\ \\ \left| ln\sqrt{\frac{f_{min}(G_{ff})+1}{f_{min}(MCN)+1}} \right| & \text{, if} \quad f_{min}(MCN) = 0 \\ \\ \left| ln\sqrt{\frac{f_{min}(G_{ff})+2*\mid f_{min}(MCN)\mid}{f_{min}(MCN)+2*\mid f_{min}(MCN)\mid}} \right| & \text{, if} \quad f_{min}(MCN) < 0 \end{cases} \tag{13}$$

Where $f_{min}(G_{ff})$ is the value of the objective function of the first feasible solution found and $f_{min}(MCN)$ is the value of the objective function of the best solution found. For this measure, statistical values are also provided.

4.2. Experimental Design

Two types of comparisons were considered in the experiments: (1) a direct comparison, where the algorithms were implemented and their corresponding performances evaluated as well, and (2) an indirect comparison, where the results of other algorithms were taken from the specialized literature and compared with those obtained by the proposed M-ABC.

The three experiments performed are detailed below:

1. An indirect comparison between the final results reported by the original ABC algorithm to solve CNOPs [18] and those obtained by M-ABC. Statistical values (best, mean, worst results and the standard deviation were considered). Based on the fact that in [18] only the first 13 benchmark functions (g01-g13) were solved, the comparison will only be made in such test problems.
2. A direct comparison between M-ABC and our implementation of the original ABC algorithm. Some changes were introduced to this algorithm to promote a fair comparison, such as the dynamic tolerance in equality constrained problems, binary tournament selection of food sources by the onlooker bees and $MCN = 5800$. The rest of the parameters of the ABC algorithm were not modified and neither the scout behavior nor the value of *limit*. In other words, the difference between M-ABC and ABC is the smart flight operator. Statistical values of the final results were compared. Furthermore, the six performance measures detailed in Section 4.1 were utilized for comparison. We compared the results on the 24 benchmark functions (g01-g24).
3. An indirect comparison with some state-of-the-art algorithms: (1) the Ensemble of Constraint-Handling Techniques with Evolutionary Programming (ECHT-EP2) [38], (2) the Hybrid Constrained Optimization Evolutionary Algorithm (HCOEA) [39], and (3) the Adaptative Trade-off Model Evolutionary Strategy (ATMES) [40].

The fixed parameter values utilized by M-ABC in all the experiments are the following: number of solutions $SN = 20$, maximum cycle number $MCN = 5800$, $Limit = MCN/(2 * SN) = 145$ and the modification rate $MR = 0.8$. The total number of evaluations performed by M-ABC is $240,000$, even in the problems with equality constraints that require to calculate the best solution found, in every cycle, when the tolerance is adjusted. 30 independent runs were performed in each experiment. The initial value of ϵ is 1.0, the final value of the tolerance is 0.0001 and the decreasing rate (dec) is 1.002.

The ABC version utilized in the second experiment used the same parameter values as those reported for M-ABC ($SN = 20$, $MCN = 5800$, *Limit* = 145, *MR* = 0.8, and 240,000 evaluations, initial $\epsilon = 1.0$, final $\epsilon = 0.0001$ and $dec = 1.002$).

The discussion of results will be based on quality (best result found so far) and consistency (better mean and standard deviation values).

4.3. General performance of M-ABC

Before discussing the results of the three experiments, we give a glimpse of the overall final results obtained by M-ABC when solving the 24 test problems: it reached the best known or optimum feasible solution in 14 problems: g01, g02, g03, g04, g06, g08, g11, g12, g13, g15, g16, g17, g18, and g24. Within these problems, in g01, g03, g04, g06, g08, g11, g12, g16, and g24, the algorithm obtained the best known or optimum feasible solution in all 30 runs. M-ABC generated feasible solutions in 22 of 24 benchmark problems (only in problems g20 and g22 no feasible solutions were found). Finally, with the exception of problems g21 and g23, the algorithm found the feasible region in the 30 runs for each test problem.

4.4. Results of experiment 1

The results for this experiment are summarized in Table 2. Based on [41], two-sample t-tests were applied to the results reported by the samples of runs where the feasible global best known or optimum solution was not consistently reached. As observed in Table 2, in six test problems: g01, g03, g04, g08, g11, and g12 both, ABC and M-ABC obtained similar results. Moreover, in problems g05, g07, and g10 no significant differences were found based on the results of the two-sample t-tests. On the other hand, significant differences were observed in problems g02, g06, g09 and g13. Therefore the discussion will be centered on these test problems.

M-ABC provided a “better” best solution with respect to ABC in test problems g02, g09 and g13. In the remaining problem g06 both algorithms obtained a “similar” best result.

M-ABC provided a “better” mean and standard deviation values in problems g02 and g06, while a “better” mean value was obtained in problem g13. ABC found a “better” mean and standard deviation results in problem g09 and a better standard deviation value in problem g13.

The overall results suggest that M-ABC enhances ABC’s ability to provide high-quality results in problems with high-dimensionality (g02 with 20 variables, the highest in that set of problems) and in presence of only equality constraints (g13 with two nonlinear and one linear equality constraints).

Table 2: Statistical results for experiment 1: Indirect comparison between ABC and by M-ABC. A result in boldface indicates a better result or that the best known solution was reached. A test problem remarked in gray indicates that a significant difference was observed based on 95%-confidence two-sample t-test.

Function/ Optimal		Algorithms	
		ABC	M-ABC
g01 **-15.000**	Best	**-15**	**-15**
	Mean	**-15**	**-15**
	Worst	**-15**	**-15**
	Std. Dev.	**0.00E+00**	**0.00E+00**
g02 **-0.803619**	Best	-0.803598	**-0.803615**
	Mean	-0.792412	**-0.799336**
	Worst	-0.749797	**-0.777438**
	Std. Dev.	1.20E-02	**6.84E-03**
g03 **-1.000**	Best	**-1.000**	**-1.000**
	Mean	**-1.000**	**-1.000**
	Worst	**-1.000**	**-1.000**
	Std. Dev.	**0.00E+00**	**4.68E-05**
g04 **-30665.539**	Best	**-30665.539**	**-30665.539**
	Mean	**-30665.539**	**-30665.539**
	Worst	**-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**
	Worst	5438.387	**5317.197**
	Std. Dev.	7.54E+01	**5.61E+01**
g06 **-6961.814**	Best	**-6961.814**	**-6961.814**
	Mean	-6961.813	**-6961.814**
	Worst	-6961.805	**-6961.814**
	Std. Dev.	2.00E-03	**0.00E+00**
g07 **24.306**	Best	24.330	**24.315**
	Mean	24.473	**24.415**
	Worst	25.19	**24.854**
	Std. Dev.	1.86E-01	**1.24E-01**
g08 **-0.095825**	Best	**-0.095825**	**-0.095825**
	Mean	**-0.095825**	**-0.095825**
	Worst	**-0.095825**	**-0.095825**
	Std. Dev.	**0.00E+00**	**4.23E-17**
g09 **680.630**	Best	680.634	**680.632**
	Mean	**680.640**	680.647
	Worst	**680.653**	680.691
	Std. Dev.	**4.00E-03**	1.55E-02
g10 **7049.248**	Best	7053.904	**7051.706**
	Mean	**7224.407**	7233.882
	Worst	7604.132	**7473.109**
	Std. Dev.	1.34E+02	**1.10E+02**
g11 **0.75**	Best	**0.75**	**0.75**
	Mean	**0.75**	**0.75**
	Worst	**0.75**	**0.75**
	Std. Dev.	**0.00E+00**	**2.30E-05**
g12 **-1.000**	Best	**-1.000**	**-1.000**
	Mean	**-1.000**	**-1.000**
	Worst	**-1.000**	**-1.000**
	Std. Dev.	**0.00E+00**	**0.00E+00**
g13 **0.05395**	Best	0.760000	**0.053985**
	Mean	0.968000	**0.158552**
	Worst	1.000000	**0.442905**
	Std. Dev.	**5.50E-02**	1.73E-01

4.5. Results of experiment 2

The results of experiment 2 are divided in four parts: (1) Final results, (2) FP, P, AFES, and SP, (3) EVALS and (4) Progress Ratio.

4.5.1. Comparison of final results

The statistical results of the direct comparison between ABC and M-ABC are reported in Table 3 for the first twelve test problems (g01 to g12) and in Table 4 for the last twelve problems (g13 to g24). As in experiment 1, a two-sample 95%-confidence t-test was applied to each pair of samples to verify statistical significance of the differences found.

Problems g20 and g22 are discarded of the discussion because both algorithms were unable to generate feasible solutions in any given run. In the same way, in test problems g01, g04, g08, g12, g16, and g24 both, ABC and M-ABC obtained the same statistical values. Finally, in problems g02, g05, and g17 no significant differences, based on the two-sample t-tests, were found. The discussion will then focuses in the remaining thirteen test problems (g03, g06, g07, g09, g10, g11, g13, g14, g15, g18, g19, g21, and g23).

M-ABC obtained a "better" best result in eleven test problems: g03, g07, g09, g10, g13, g14, g15, g18, g19, g21, and g23. In problems g06 and g11 ABC and M-ABC obtained a "similar" best result. Regarding the mean value, M-ABC provided a "better" mean value in test problems g03, g06, g07, g09, g10, g11, g13, g14, g15, g19, g21, and g23. ABC in contrast had a better mean value in problem g18. Finally, a "better" standard deviation value was found by M-ABC in problems g03, g06, g09, g10, g11, g14, g15, g19, and g21. ABC had a "better" standard deviation value in problems g07, g13, g18, and g23. However, except for problem g18 the results obtained by ABC are very poor with respect to those reached by M-ABC. Those results show that M-ABC is able to improve both, the quality and consistency of the final results with respect to the original ABC in most of the test problems considered in this work.

4.5.2. FP, P, AFES, and SP

The results obtained for the first four performance measures (FP, P, AFES and P) are summarized in Table 5 for the first twelve test problems and in Table 6 for the last twelve problems. Once again problems g20 and g22 are discarded from discussion because no feasible solution was found by any algorithm.

Table 3: Statistical results of the first part of experiment 2: Direct comparison between ABC and by M-ABC in the first 12 test problems. A result in boldface indicates a better result or that the best known solution was reached. A test problem remarked in gray indicates that a significant difference was observed based on 95%-confidence two-sample t-test.

Function/ Optimal		Algorithms	
		ABC	M-ABC
g01 **-15.000**	Best	**-15**	**-15**
	Mean	**-15**	**-15**
	Worst	**-15**	**-15**
	Std. Dev.	**0.00E+00**	**0.00E+00**
g02 **-0.803619**	Best	-0.803498	**-0.803615**
	Mean	**-0.8010467**	-0.799336
	Worst	**-0.792528**	-0.777438
	Std. Dev.	**3.66E-03**	6.84E-03
g03 **-1.000**	Best	-0.984	**-1.000**
	Mean	-0.185	**-1.000**
	Worst	0.000	**-1.000**
	Std. Dev.	2.50E-01	**4.68E-05**
g04 **-30665.539**	Best	**-30665.539**	**-30665.539**
	Mean	**-30665.539**	**-30665.539**
	Worst	**-30665.539**	**-30665.539**
	Std. Dev.	**2.22E-11**	**2.22E-11**
g05 **5126.498**	Best	5127.302	**5126.736**
	Mean	**5158.673**	5178.139
	Worst	**5295.980**	5317.196
	Std. Dev.	**3.82E+01**	5.60E+01
g06 **-6961.814**	Best	**-6961.814**	**-6961.814**
	Mean	-6961.814	**-6961.814**
	Worst	-6961.810	**-6961.814**
	Std. Dev.	8.25E-04	**0.00E+00**
g07 **24.306**	Best	24.455	**24.315**
	Mean	24.611	**24.415**
	Worst	24.873	**24.854**
	Std. Dev.	**9.83E-02**	1.24E-01
g08 **-0.095825**	Best	**-0.095825**	**-0.095825**
	Mean	**-0.095825**	**-0.095825**
	Worst	**-0.095825**	**-0.095825**
	Std. Dev.	**4.23E-17**	**4.23E-17**
g09 **680.630**	Best	680.645	**680.632**
	Mean	680.682	**680.647**
	Worst	680.715	**680.691**
	Std. Dev.	1.63E-02	**1.55E-02**
g10 **7049.248**	Best	7084.129	**7051.706**
	Mean	7321.184	**7233.882**
	Worst	7498.392	**7473.109**
	Std. Dev.	1.32E+02	**1.10E+02**
g11 **0.75**	Best	**0.75**	**0.75**
	Mean	0.81	**0.75**
	Worst	1.00	**0.75**
	Std. Dev.	7.74E-02	**2.30E-05**
g12 **-1.000**	Best	**-1.000**	**-1.000**
	Mean	**-1.000**	**-1.000**
	Worst	**-1.000**	**-1.000**
	Std. Dev.	**0.00E+00**	**0.00E+00**

Table 4: Statistical results of the first part of experiment 2: Direct comparison between ABC and by M-ABC in the last 12 test problems. A result in boldface indicates a better result or that the best known solution was reached. A test problem remarked in gray indicates that a significant difference was observed based on 95%-confidence two-sample t-test."-" means no feasible solution found.

Function/ Optimal		Algorithms	
		ABC	M-ABC
g13 **0.053942**	Best	0.999901	**0.053985**
	Mean	0.999985	**0.158552**
	Worst	1.000000	**0.442905**
	Std. Dev.	**2.30E-05**	1.72E-01
g14 **-47.765**	Best	-	**-47.641**
	Mean	-	**-47.271**
	Worst	-	**-46.537**
	Std. Dev.	-	**2.46E-01**
g15 **961.715**	Best	967.518	**961.715**
	Mean	968.639	**961.719**
	Worst	972.317	**961.793**
	Std. Dev.	2.06E+00	**1.42E-02**
g16 **-1.905**	Best	**-1.905**	**-1.905**
	Mean	**-1.905**	**-1.905**
	Worst	**-1.905**	**-1.905**
	Std. Dev.	**4.52E-16**	**4.52E-16**
g17 **8853.540**	Best	8959.546	**8866.618**
	Mean	9002.701	**8987.459**
	Worst	**9025.873**	9165.219
	Std. Dev.	**3.74E+01**	9.57E+01
g18 **-0.866025**	Best	-0.865677	**-0.866006**
	Mean	**-0.8598209**	-0.7950187
	Worst	**-0.771239**	-0.672216
	Std. Dev.	**1.69E-02**	9.39E-02
g19 **32.656**	Best	35.213	**33.285**
	Mean	36.482	**34.267**
	Worst	37.771	**35.746**
	Std. Dev.	6.64E-01	**6.31E-01**
g20 **0.096700**	Best	-	-
	Mean	-	-
	Worst	-	-
	Std. Dev.	-	-
g21 **193.725**	Best	277.157	**266.500**
	Mean	454.137	**306.609**
	Worst	1000.000	**329.960**
	Std. Dev.	2.55E+02	**1.98E+01**
g22 **236.431**	Best	-	-
	Mean	-	-
	Worst	-	-
	Std. Dev.	-	-
g23 **-400.055**	Best	-0.008	**-159.739**
	Mean	-0.001	**-35.272**
	Worst	**0.000**	109.010
	Std. Dev.	**1.97E-03**	8.28E+01
g24 **-5.508**	Best	**-5.508**	**-5.508**
	Mean	**-5.508**	**-5.508**
	Worst	**-5.508**	**-5.508**
	Std. Dev.	**2.71E-15**	**2.71E-15**

Table 5: Results of the second part of experiment 2: Performance measure values for FP, P, AFES and SP obtained by ABC and M-ABC in the first 12 test problems. A result in boldface indicates a better result. "-" means that the measure value could not be calculated

PERFORMANCE MEASURES PART I			
Function		**ABC**	**M-ABC**
g01	FP	**1**	**1**
	P	**1**	**1**
	AFES	**1.24E+04**	2.05E+04
	SP	**1.24E+04**	2.05E+04
g02	FP	**1**	**1**
	P	0	**0.67**
	AFES	-	**8.35E+04**
	SP	-	**1.25E+05**
g03	FP	**1**	**1**
	P	0	**1**
	AFES	-	**1.89E+05**
	SP	-	**1.89E+05**
g04	FP	**1**	**1**
	P	**1**	**1**
	AFES	**4.39E+04**	7.64E+04
	SP	**4.39E+04**	7.64E+04
g05	FP	0.7	**1**
	P	0	0
	AFES	-	-
	SP	-	-
g06	FP	**1**	**1**
	P	0.63	**1**
	AFES	1.43E+05	**1.07E+05**
	SP	2.26E+05	**1.07E+05**
g07	FP	**1**	**1**
	P	0	0
	AFES	-	-
	SP	-	-
g08	FP	**1**	**1**
	P	**1**	**1**
	AFES	**1.27E+03**	1.55E+03
	SP	**1.27E+03**	1.55E+03
g09	FP	**1**	**1**
	P	0	0
	AFES	-	-
	SP	-	-
g10	FP	**1**	**1**
	P	0	0
	AFES	-	-
	SP	-	-
g11	FP	**1**	**1**
	P	0.03	**1**
	AFES	1.89E+05	**1.89E+05**
	SP	5.69E+06	**1.89E+05**
g12	FP	**1**	**1**
	P	**1**	**1**
	AFES	**1.25E+03**	1.35E+03
	SP	**1.25E+03**	1.35E+03

Table 6: Results of the second part of experiment 2: Performance measure values for FP, P, AFES and SP obtained by ABC and M-ABC in the last 12 test problems. A result in boldface indicates a better result. "-" means that the measure value could not be calculated

PERFORMANCE MEASURES PART II			
Function		**ABC**	**M-ABC**
g13	FP	0.97	**1**
	P	0	**0.03**
	AFES	-	**1.89E+05**
	SP	-	**5.69E+06**
g14	FP	0	**1**
	P	0	0
	AFES	-	-
	SP	-	-
g15	FP	**1**	**1**
	P	0	**0.3**
	AFES	-	**1.89E+05**
	SP	-	**6.32E+05**
g16	FP	**1**	**1**
	P	**1**	**1**
	AFES	2.75E+04	**2.33E+04**
	SP	2.75E+04	**2.33E+04**
g17	FP	0.1	**1**
	P	0	**0.03**
	AFES	-	**1.89E+05**
	SP	-	**5.68E+06**
g18	FP	**1**	**1**
	P	0.03	**0.37**
	AFES	2.19E+05	**7.06E+04**
	SP	6.60E+06	**1.92E+05**
g19	FP	**1**	**1**
	P	0	0
	AFES	-	-
	SP	-	-
g20	FP	0	0
	P	0	0
	AFES	-	-
	SP	-	-
g21	FP	0.47	**0.73**
	P	0	0
	AFES	-	-
	SP	-	-
g22	FP	0	0
	P	0	0
	AFES	-	-
	SP	-	-
g23	FP	**0.87**	0.83
	P	0	0
	AFES	-	-
	SP	-	-
g24	FP	**1**	**1**
	P	**1**	**1**
	AFES	6.95E+03	**6.58E+03**
	SP	6.95E+03	**6.58E+03**

- **FP:** M-ABC was able to provide better results in problems g05, g13, g14, g17 and g21. ABC was better in problem g23. In the remaining problems (g01, g02, g03, g04, g06, g07, g08, g09, g10, g11, g12, g15, g16, g18, g19, and g24) both algorithms had similar results.

- **P:** M-ABC obtained a better P value in problems: g02, g03, g06, g11, g13, g15, g17, and g18. Moreover, in problems g01, g04, g08, g12, g16 and g24 both algorithms reached the same value of P. On the other hand, both M-ABC and ABC were unable to provide successful runs in problems g05, g07, g09, g10, g14, g19, g21, and g23.

- **AFES:** A better AFES value was computed by M-ABC with respect to ABC in problems g02, g03, g06, g11, g13, g15, g16, g17, g18, and g24. In contrast, ABC reached a better AFES value in problems g01, g04, g08, and g12. In the remaining test problems (g05, g07, g09, g10, g14, g19, g21, and g23) neither M-ABC nor ABC ended in successful runs.

- **SP:** In a similar way as the results presented for the AFES measure, a better SP value was shown by M-ABC in problems g02, g03, g06, g11, g13, g15, g16, g17, g18, and g24. ABC found better results for this measure in problems g01, g04, g08 and g12. No SP value could be calculated in the rest of the problems (g05, g07, g09, g10, g14, g19, g21, and g23).

The overall results on the four performance measures show that both algorithms were very capable to reach the feasible region of the search space (FP value). However, M-ABC improved this capability in problems with equality constraints (g05, g13, g14, g17 and g21). In the same way, M-ABC outperformed ABC in its ability to reach the vicinity of the feasible global optimum (P value). Nevertheless, both algorithms showed difficulties to reach positive P values in eight test problems. Regarding the computational cost measured by the number of evaluations required in a successful run, M-ABC required less evaluations, in average, with respect to ABC in ten test problems with respect to only four problems where ABC was more competitive. This behavior was also observed for the SP measure values, where M-ABC showed a better ratio between successful runs and evaluations computed.

4.5.3. EVALS

The summary of the statistical values of the EVALS measure can be found in Table 7 for the first twelve test problems and in Table 8 for the last twelve test problems. Based on those tables, some test problems are excluded from the discussion: Problems g20 and g22 are not considered because no feasible solutions were found in any single run. Furthermore, problems g02, g04, g08, g09, g12, g19, and g24 are also not discussed because the first feasible solution appears either in the initial population or in the first cycle of the algorithm. Finally, based on the two-sample t-test results no significant differences were observed in test problems g03, g06, g07, g10 and g16, i.e., the performance is similar in both algorithms. Therefore the problems to be discussed are g01, g05, g11, g13, g14, g15, g17, g18, g21 and g23.

M-ABC obtained a "better" best value in problems g11, g14, g15, and g17 while ABC provided better values in problems g01, g05, g13, g18, g21 and g23.

Based on the values observed in Tables 7 and 8, M-ABC had a "better" mean and standard deviation values in problems g05, g11, g13, g14, g15, g17, and g21 and it also obtained a "better" standard deviation value in problems g01 and g23. ABC obtained a "better" mean and standard deviation values in problem g18 and only a better standard deviation value in problems g01 and g23.

The performances by both algorithms point out that M-ABC is more consistent on reaching faster the feasible region of the search space than ABC. However, ABC requires less evaluations to generate the first feasible solution in some runs for some test problems, but it showed a lack of robustness in such behavior.

4.5.4. Progress Ratio

The obtained results for the progress ratio measure are summarized in Tables 9 for the first twelve problems and in Table 10 for the last twelve problems. Problems g20 and g22 were excluded from discussion because no feasible solutions were found in any single run.

It is clear from the results obtained by the two-sample t-test reported in Tables 9 and 10 that in most of the test problems no significant difference was obtained, i.e., both algorithms mostly performed a quite similar improvement inside the feasible region of the search space. Nonetheless, in the following problems there were differences on this regard: g01, g03, g10, g11, g13, g15, g21, and g23.

Table 7: Results of the third part of experiment 2: Statistical results for the EVALS measure by ABC and M-ABC in the first 12 test problems. A result in boldface indicates a better result. A test problem remarked in gray indicates that a significant difference was observed based on 95%-confidence two-sample t-test.

EVALS PART I			
Function		**ABC**	**M-ABC**
g01	Best	**76**	433
	Mean	**353**	709
	Worst	**656**	1030
	Std. Dev.	1.63E+02	**1.35E+02**
g02	Best	**21**	**21**
	Mean	24	**21**
	Worst	34	**21**
	Std. Dev.	3.08E+00	**0.00E+00**
g03	Best	**189270**	189533
	Mean	189955	**189554**
	Worst	203659	**189594**
	Std. Dev.	2.66E+03	**1.23E+01**
g04	Best	**21**	**21**
	Mean	**23**	23
	Worst	**35**	**35**
	Std. Dev.	**2.90E+00**	3.04E+00
g05	Best	**189317**	189368
	Mean	199964	**189445**
	Worst	234047	**189498**
	Std. Dev.	1.25E+04	**2.63E+01**
g06	Best	**148**	**148**
	Mean	540	**469**
	Worst	1121	**812**
	Std. Dev.	1.92E+02	**1.57E+02**
g07	Best	**354**	361
	Mean	**707**	742
	Worst	1127	**1079**
	Std. Dev.	2.17E+02	**1.91E+02**
g08	Best	**28**	**28**
	Mean	101	**98**
	Worst	**197**	222
	Std. Dev.	**4.01E+01**	5.24E+01
g09	Best	**22**	**22**
	Mean	117	**93**
	Worst	294	**240**
	Std. Dev.	5.98E+01	**4.91E+01**
g10	Best	568	**396**
	Mean	**1300**	1378
	Worst	**2167**	2981
	Std. Dev.	**5.02E+02**	6.13E+02
g11	Best	189621	**189459**
	Mean	189681	**189539**
	Worst	189747	**189576**
	Std. Dev.	2.95E+01	**2.37E+01**
g12	Best	**21**	**21**
	Mean	41	**34**
	Worst	87	**80**
	Std. Dev.	1.82E+01	**1.43E+01**

Table 8: Results of the third part of experiment 2: Statistical results for the EVALS measure by ABC and M-ABC in the last 12 test problems. A result in boldface indicates a better result. "-" means that the measure value could not be calculated. A test problem remarked in gray indicates that a significant difference was observed based on 95%-confidence two-sample t-test.

EVALS PART II			
Function		**ABC**	**M-ABC**
g13	Best	**189309**	189427
	Mean	195210	**189472**
	Worst	223538	**189533**
	Std. Dev.	8.05E+03	**2.66E+01**
g14	Best	-	**189481**
	Mean	-	**189508**
	Worst	-	**189575**
	Std. Dev.	-	**2.65E+01**
g15	Best	189444	**189404**
	Mean	190585	**189486**
	Worst	200426	**189546**
	Std. Dev.	2.75E+03	**3.69E+01**
g16	Best	**109**	142
	Mean	722	**605**
	Worst	**1303**	1882
	Std. Dev.	**3.13E+02**	3.81E+02
g17	Best	193876	**189284**
	Mean	207002	**189358**
	Worst	231505	**189450**
	Std. Dev.	2.12E+04	**4.25E+01**
g18	Best	**932**	1310
	Mean	**1624**	1839
	Worst	**1970**	2370
	Std. Dev.	**2.05E+02**	3.03E+02
g19	Best	**21**	**21**
	Mean	25	**23**
	Worst	35	**34**
	Std. Dev.	3.84E+00	**2.75E+00**
g20	Best	-	-
	Mean	-	-
	Worst	-	-
	Std. Dev.	-	-
g21	Best	**189213**	189462
	Mean	200868	**194528**
	Worst	226314	**207310**
	Std. Dev.	1.25E+04	**5.01E+03**
g22	Best	-	-
	Mean	-	-
	Worst	-	-
	Std. Dev.	-	-
g23	Best	**189173**	189392
	Mean	**190946**	193557
	Worst	215252	**202538**
	Std. Dev.	6.10E+03	**3.09E+03**
g24	Best	**21**	**21**
	Mean	**23**	**23**
	Worst	**29**	**29**
	Std. Dev.	2.08E+00	**2.01E+00**

M-ABC obtained a "better" progress ratio value in problem g13, while ABC outperformed M-ABC in problems g01, g03, g10, g11, g15, g21, and g23. M-ABC presented a "better" mean and standard deviation values in problem g21 and it reached a "better" mean value in problem g13. M-ABC found a better standard deviation values in problems g03, g10, g11, g15, and g23 but the best and mean values were very poor. In contrast, ABC obtained a "better" mean and standard deviation values in problem g01 and a "better" mean value in problems g03, g10, g11, g15, and g23. Moreover, ABC obtained a better standard deviation value in problem g13 but the best and mean values were outperformed by M-ABC.

The isolated results of the progress ratio measure suggest a better improvement by ABC inside the feasible region with respect to M-ABC. However, it is interesting to note that ABC could not improve its first feasible solution in a single run in problems g03, g11, g13, g15, g21, and g23, while M-ABC was able to slightly improve it for such test problems.

4.5.5. Discussion

Interesting findings on the behavior of M-ABC and ABC were observed from the results in the four parts of experiment 2:

- M-ABC clearly improved ABC's capabilities to reach better final results, based on both, quality and consistency.
- M-ABC was able to generate feasible solutions with more consistency in problems with equality constraints.
- M-ABC was able to reach the vicinity of the best known or global optimum solution more frequently and with a lower average number of evaluations in more test problems with respect to ABC.
- Despite the fact that ABC required, in the best case, less evaluations to generate the first feasible solution in more test problems than M-ABC, the latter was better, in average, mostly in problems with equality constraints.
- ABC was more capable to improve the first feasible solution found in more test problems with respect to M-ABC, even in those with presence of equality constraints.

Table 9: Results of the fourth part of experiment 2: Statistical results for the Progress Ratio measure by ABC and M-ABC in the first 12 test problems. A result in boldface indicates a better result. A test problem remarked in gray indicates that a significant difference was observed based on 95%-confidence two-sample t-test.

PROGRESS RATIO PART I			
Function		**ABC**	**M-ABC**
g01	Best	**0.370**	0.345
	Mean	**0.328**	0.295
	Worst	**0.274**	0.233
	Std. Dev.	**2.69E-02**	2.96E-02
g02	Best	**0.331**	0.329
	Mean	**0.315**	0.314
	Worst	0.299	**0.299**
	Std. Dev.	6.85E-03	**6.42E-03**
g03	Best	**0.348**	0.001
	Mean	**0.184**	3.58E-04
	Worst	0	**8.49E-05**
	Std. Dev.	1.59E-01	**1.45E-04**
g04	Best	**0.104**	0.101
	Mean	0.067	**0.068**
	Worst	0.026	**0.029**
	Std. Dev.	2.06E-02	**1.91E-02**
g05	Best	**1.64E-06**	1.64E-07
	Mean	**2.29E-07**	7.94E-08
	Worst	0	**3.03E-08**
	Std. Dev.	4.09E-07	**3.60E-08**
g06	Best	**0.292**	0.280
	Mean	0.215	**0.219**
	Worst	0.107	**0.123**
	Std. Dev.	5.34E-02	**3.90E-02**
g07	Best	**2.524**	2.517
	Mean	**1.982**	1.882
	Worst	**1.112**	0.663
	Std. Dev.	**3.88E-01**	4.19E-01
g08	Best	**0.517**	0.483
	Mean	**0.338**	0.338
	Worst	**0.047**	**0.047**
	Std. Dev.	**7.14E-02**	7.51E-02
g09	Best	**4.798**	4.794
	Mean	**2.855**	2.268
	Worst	**0.800**	0.267
	Std. Dev.	**1.41E+00**	1.50E+00
g10	Best	**0.700**	0.633
	Mean	**0.526**	0.482
	Worst	**0.356**	0.305
	Std. Dev.	8.47E-02	**7.41E-02**
g11	Best	**0.144**	2.59E-04
	Mean	**0.018**	7.36E-05
	Worst	0	**2.00E-05**
	Std. Dev.	3.81E-02	**5.23E-05**
g12	Best	0.145	**0.164**
	Mean	0.058	**0.066**
	Worst	**0.010**	1.44E-04
	Std. Dev.	**3.79E-02**	4.03E-02

Table 10: Results of the fourth part of experiment 2: Statistical results for the Progress Ratio measure by ABC and M-ABC in the last 12 test problems. A result in boldface indicates a better result. "-" means that the measure value could not be calculated. A test problem remarked in gray indicates that a significant difference was observed based on 95%-confidence two-sample t-test.

PROGRESS RATIO PART II			
Function		**ABC**	**M-ABC**
g13	Best	4.95E-05	**6.48E-04**
	Mean	6.45E-06	**1.05E-04**
	Worst	0	**1.94E-05**
	Std. Dev.	**1.16E-05**	1.23E-04
g14	Best	-	**1.00E-03**
	Mean	-	**8.49E-05**
	Worst	-	**1.66E-05**
	Std. Dev.	-	**1.80E-04**
g15	Best	**2.47E-03**	1.52E-07
	Mean	**4.11E-04**	7.29E-08
	Worst	0	**2.24E-08**
	Std. Dev.	9.35E-04	**3.84E-08**
g16	Best	**2.17E-01**	2.15E-01
	Mean	**1.70E-01**	1.59E-01
	Worst	**8.08E-02**	6.54E-02
	Std. Dev.	**3.27E-02**	3.78E-02
g17	Best	2.14E-07	**2.40E-07**
	Mean	9.04E-08	**1.02E-07**
	Worst	2.02E-08	**4.03E-08**
	Std. Dev.	1.07E-07	**6.33E-08**
g18	Best	**3.42E-01**	3.27E-01
	Mean	**2.92E-01**	2.75E-01
	Worst	**2.18E-01**	1.81E-01
	Std. Dev.	**3.25E-02**	4.20E-02
g19	Best	**3.575**	3.480
	Mean	**3.130**	3.123
	Worst	2.460	**2.500**
	Std. Dev.	3.04E-01	**2.63E-01**
g20	Best	-	-
	Mean	-	-
	Worst	-	-
	Std. Dev.	-	-
g21	Best	**0.562**	0.495
	Mean	0.194	**0.336**
	Worst	0	**0.018**
	Std. Dev.	1.97E-01	**1.29E-01**
g22	Best	-	-
	Mean	-	-
	Worst	-	-
	Std. Dev.	-	-
g23	Best	**6.909**	0.004
	Mean	**0.924**	0.001
	Worst	0	**2.56E-05**
	Std. Dev.	1.95E+00	**9.17E-04**
g24	Best	**0.313**	0.303
	Mean	**0.211**	0.203
	Worst	**0.036**	**0.036**
	Std. Dev.	7.87E-02	**7.65E-02**

Recalling from Section 4.2, the difference between ABC and M-ABC in experiment 2 is the smart flight operator. The join between this difference and the findings summarized in the previous list suggests that this operator favors the approach to the feasible region from more promising regions of the search space (i.e., better final results and higher P values by M-ABC combined with lower values for the AFES and Progress Ratio measures). However, this convenient approach may consume more evaluations to reach the feasible region in some test problems (i.e., higher EVALS values in some runs for some test problems). Finally, even both algorithms had the improved equality constraint handling mechanism, M-ABC could generate feasible solutions in more test problems in presence of equality constraints (FP measure).

It is well known from the No Free Lunch Theorems for optimization [42] that using such a limited number of test problems can not guarantee, in any way, that an algorithm (M-ABC in this case) which performs well on them, will be equally competitive in another set of problems. However, this empirical study aimed to provide some evidence on the behavior of ABC and its improved version in constrained continuous search spaces. This information may be useful for practitioners and researchers interested on using this nature-inspired algorithm on specific situations e.g., limited evaluations due to high computational cost, special interest in feasible solutions more than in optimal solutions, etc. For example, ABC seems to be more suitable to find feasible solutions faster than M-ABC. In contrast, M-ABC can provide a better feasible solution than ABC, but the evaluations required could be more than those required by ABC. Another example could be a problem with equality constraints, where M-ABC seems to be a better choice with respect to ABC.

4.6. Results of experiment 3

The comparison of the final results obtained by M-ABC and some state-of-the-art nature-inspired algorithms to solve CNOPs is presented in Table 11. The results are restricted to the first thirteen test problems (g01 to g13) because no results were found for the rest of them.

The algorithms used for comparison were: (1) the ensemble of constraint-handling techniques ECHT-EP2 [38] where four constraint-handling techniques cooperate among them by using Evolutionary Programming as a search algorithm, (2) HCOEA [39] where several mechanisms such as multi-objective optimization, global search based on a niching genetic algorithm and a parallel local search operator with multi-parent crossover are combined and (3) ATMES [40], where different trade-off models based on feasibility are adopted The four compared algorithms (ECHT-EP2, HCOEA, ATMES and M-ABC) performed 240,000 evaluations in their reported results.

The statistical values in Table 11 indicate that the four algorithms presented a similar performance in problems g01, g03, g04, g06, g08, g11 and g12. Based on t-test results, in problem g02 the differences observed between each compared algorithm and M-ABC were not significant. In problems g05, g07, g09 and g13 M-ABC was outperformed by the compared approaches. Nevertheless, in problems g07 and g09 the differences in statistical results were minimal. Finally, in problem g10 the differences were not significant with respect to ATMES, i.e., the performance was similar with respect to M-ABC, but ECHT-EP2 and HCOEA were better than M-ABC.

The comparison with state-of-the-art algorithms shows that, even M-ABC is a novel meta-heuristic nature-inspired approach, it has comparable results in most of the test problems.

5. Conclusions and future work

A modified version of the artificial bee colony algorithm to solve constrained numerical optimization problems was proposed in this paper. The mechanisms to handle equality and boundary constraints were improved with the aim to favor a more convenient approach to the feasible region of the search space. Furthermore, the fitness proportional selection of food sources utilized in the original ABC was supplanted by a binary tournament selection based on feasibility. Finally, to facilitate the location of food sources as convenient attractors in a constrained search space, the so called smart flight operator was proposed to be used by the scout bee instead of the purely random generation of solutions employed by the original ABC.

Different aspects were compared in M-ABC and the original ABC such as their capabilities to reach the feasible region and the feasible global optimum, besides their associated computational cost measured by the number of evaluations required. Moreover, the evaluations required to generate the first feasible solution and the capability to improve such solution were also measured.

The results obtained in the first experiment showed that M-ABC outperformed the original ABC version when solving thirteen test problems, mostly in those with a high dimensionality and in presence of equality constraints.

Table 11: Results of experiment 3. Final results comparison of M-ABC with respect to state-of-the-art algorithms (HCOEA [39], ATMES [40] and ECHT-EP2 [38]) on 13 benchmark functions (g01-g13). A result in boldface indicates either a better result or that the global optimum (or best known solutions) was reached.

Function/ Optimal		**Algorithms**			
		ECHT-EP2[38]	**HCOEA[39]**	**ATMES[40]**	**M-ABC**
	Best	**-15**	**-15**	**-15**	**-15**
g01	Mean	**-15**	**-15**	**-15**	**-15**
-15	Worst	**-15**	-14.999	**-15**	**-15**
	Std. Dev.	**0.00E+00**	4.30E-07	1.60E-14	**0.00E+00**
	Best	**-0.803619**	-0.803241	-0.803388	-0.803615
g02	Mean	-0.799822	**-0.801258**	-0.790148	-0.799336
-0.803619	Worst	-0.785182	**-0.792363**	-0.756986	-0.777438
	Std. Dev.	6.29E-03	**3.83E-03**	1.30E-02	6.84E-03
	Best	**-1**	**-1**	**-1**	**-1**
g03	Mean	**-1**	**-1**	**-1**	**-1**
-1	Worst	**-1**	**-1**	**-1**	**-1**
	Std. Dev.	0.00E+00	**1.30E-12**	5.90E-05	4.68E-05
	Best	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**
g04	Mean	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**
-30665.539	Worst	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**
	Std. Dev.	**0.00E+00**	5.40E-07	7.40E-12	2.22E-11
	Best	**5126.497**	5126.498	5126.498	5126.736
g05	Mean	**5126.497**	5126.498	5127.648	5178.139
5126.498	Worst	**5126.497**	5126.498	5135.256	5317.196
	Std. Dev.	**0.00E+00**	1.73E-07	1.80E+00	5.61E+01
	Best	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**
g06	Mean	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**
-6961.814	Worst	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**
	Std. Dev.	**0.00E+00**	8.51E-12	4.60E-12	**0.00E+00**
	Best	**24.306**	**24.306**	**24.306**	24.315
g07	Mean	**24.306**	24.307	24.316	24.415
24.306	Worst	**24.306**	24.309	24.359	24.854
	Std. Dev.	**3.19E-05**	7.12E-04	1.10E-02	1.24E-01
	Best	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
g08	Mean	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
-0.095825	Worst	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
	Std. Dev.	**0.00E+00**	2.42E-17	2.80E-17	4.23E-17
	Best	**680.630**	**680.630**	**680.630**	680.632
g09	Mean	**680.630**	**680.630**	680.639	680.647
680.63	Worst	**680.630**	**680.630**	680.673	680.691
	Std. Dev.	**2.61E-08**	9.41E-08	1.00E-02	1.55E-02
	Best	**7049.248**	7049.287	7052.253	7051.706
g10	Mean	**7049.249**	7049.525	7250.437	7233.882
7049.248	Worst	**7049.250**	7049.984	7560.224	7473.109
	Std. Dev.	**6.60E-04**	1.50E-01	1.20E+02	1.10E+02
	Best	**0.75**	**0.75**	**0.75**	**0.75**
g11	Mean	**0.75**	**0.75**	**0.75**	**0.75**
0.75	Worst	**0.75**	**0.75**	**0.75**	**0.75**
	Std. Dev.	**0.00E+00**	1.55E-12	3.40E-04	2.30E-05
	Best	**-1**	**-1**	**-1**	**-1**
g12	Mean	**-1**	**-1**	**-1**	**-1**
-1	Worst	**-1**	**-1**	**-1**	**-1**
	Std. Dev.	**0.0E+00**	**0.00E+00**	1.00E-03	**0.00E+00**
	Best	**0.053942**	0.053949	0.053950	0.053985
g13	Mean	**0.053942**	0.053949	0.053959	0.158552
0.053942	Worst	**0.053942**	0.053949	0.053999	0.442905
	Std. Dev.	**1.00E-12**	8.68E-08	1.30E-05	1.73E-01

The findings of the second experiment, which covered a direct comparison between M-ABC and a version of ABC with all the elements of M-ABC but without the smart flight operator, suggested that M-ABC is able to reach the feasible region, even in problems with equality constraints, from a more promising region with respect to ABC (tough this behavior may cost more evaluations). Therefore, M-ABC is capable of reaching a better feasible solution despite a smaller capability to improve within the feasible region with respect to ABC.

The comparison made in the third experiment showed that M-ABC provided comparable results with respect to three state-of-the-art nature-inspired algorithms to solve CNOPs. Nevertheless, its performance still requires improvement as to cover more test problems with competitive results.

The future work includes testing other constraint-handling mechanisms suitable for a binary tournament selection such as the ϵ-constrained method [43], using local search operators to improve the quality of solutions [44] and solving multi-objective constrained optimization problems. Finally, some tests will be carried out to analyze the sensitivity of M-ABC to its parameters such as SN, MR, and $Limit$.

Acknowledgments

The first author acknowledges support from the Consejo Nacional de Ciencia y Tecnología (CONACyT) through project number 79809. The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at LANIA.

References

[1] A. P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, Wiley, 2005.

[2] A. Eiben, J. E. Smith, Introduction to Evolutionary Computing, Natural Computing Series, Springer Verlag, 2003.

[3] E. Mezura-Montes (Ed.), Constraint-Handling in Evolutionary Optimization, Vol. 198 of Studies in Computational Intelligence, Springer-Verlag, 2009.

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

[5] C. A. C. Coello, Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art, Computer Methods in Applied Mechanics and Engineering 191 (11-12) (2002) 1245–1287.

[6] C. K. Monson, K. D. Seppi, Linear Equality Constraints and Homomorphous Mappings in PSO, in: 2005 IEEE Congress on Evolutionary Computation (CEC'2005), Vol. 1, IEEE Service Center, Edinburgh, Scotland, 2005, pp. 73–80.

[7] L. Cagnina, S. Esquivel, C. Coello-Coello, A Bi-population PSO with a Shake-Mechanism for Solving Constrained Numerical Optimization, in: 2007 IEEE Congress on Evolutionary Computation (CEC'2007), IEEE Press, Singapore, 2007, pp. 670–676.

[8] E. Mezura-Montes, J. I. Flores-Mendoza, Multiobjective problems solving from nature: From concepts to applications, in: R. Chiong (Ed.), Nature-Inspired Algorithms for Optimization, Vol. 193, Springer-Verlag, Studies in Computational Intelligence Series, 2009, ISBN: 978-3-540-72963-1., 2009, Ch. Improved Particle Swarm Optimization in Constrained Numerical Search Spaces, pp. 299–332.

[9] Angel E. Muñoz-Zavala, A. Hernández-Aguirre, E. R. Villa-Diharce, S. Botello-Rionda, PESO+ for Constrained Optimization, in: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), IEEE, Vancouver, BC, Canada, 2006, pp. 935–942.

[10] A. E. Muñoz Zavala, A. Hernández Aguirre, E. R. Villa Diharce, S. Botello Rionda, Constrained optimization with an improved particle swarm optimization algorithm, International Journal of Intelligent Computing and Cybernetics 1 (3) (2008) 425–453.

[11] H. A. Abbass, Marriage in honey bees optimization (mbo) : A haplometrosis polygynous swarming approach, in: Congress on Evolutionary Computation, CEC 2001, IEEE Press, 2001, pp. 207–214.

[12] O. B. Haddad, M. A. Marino, Dynamic penalty function as a strategy in solving water resources combinatorial optimization problems with honey-bee mating operation (hbmo) algorithm, Journal of Hydroinformatics 9 (3) (2007) 233–250.

[13] A. Baykasoglu, L. Ozbakir, P. Tapkan, Artificial bee colony algorithm and its application to generalized assignment problem, in: F. T. Chan, M. K. Tiwari (Eds.), Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Itech Education and Pub., Vienna, Austria, 2007, pp. 113–144, iSBN 978-3-902613-09-7.

[14] D. Pham, A. Ghanbarzadeh, S. O. E. Ko, S. Rahim, M. Zaidi, The bees algorithm: A novel tool for complex optimisation problems, in: 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006), Oxford:Elsevier, 2006, pp. 454–461.

[15] D. Karaboga, B. Akay, Artificial bee colony (abc), harmony search and bees algorithms on numerical optimization, in: Innovative Production Machines and Systems Virtual Conference (IPROMS 2009), 2009, available at: http://conference.iproms.org/sites/conference.iproms.org/files/IPROMSABCv2.pdf.

[16] X.-S. Yang, Engineering optimizations via nature-inspired virtual bee algorithms, in: Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Vol. 3562/2005, Springer Berlin/Heidelberg, 2005, pp. 317–323.

[17] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, Journal of Global Optimization 39 (3) (2007) 459–471.

[18] D. Karaboga, B. Basturk, Artificial bee colony(abc) optimization algorithm for solving constrained optimization problems, in: P. Melin, O. Castillo, L. T. Aguilar, J. Kacptrzyk, W. Pedrycz (Eds.), Foundations of Fuzzy Logic and Soft Computing, 12th International Fuzzy Systems Association, World Congress, IFSA 2007, Springer, Lecture Notes in Artificial Intelligence Vol. 4529, Cancun, Mexico, 2007, pp. 789–798.

[19] E. Mezura-Montes, O. Cetina-Domínguez, Exploring promising regions of the search space with the scout bee in the artificial bee colony for

constrained optimization, in: C. H. Dagli, et al. (Eds.), Intelligent Engineering Systems through Artificial Neural Networks (ANNIE'2001), Vol. 19, ASME Press, St. Louis Missouri, USA, 2009, pp. 253–260.

[20] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Tech. rep., Erciyes University, Engineering Faculty, Turkey (2005).

[21] P.-W. TSai, J.-S. Pan, B.-Y. Liao, S.-C. Chu, Enhanced artificial bee colony optimization, International Journal of Innovative Computing, Information and Control 5 (12).

[22] O. B. Haddad, A. Afshar, M. A. Mario, Honey-bees mating optimization (hbmo) algorithm: A new heuristic approach for water resources optimization, Water Resources Management 20 (2006) 661–680.

[23] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.

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

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

[26] K. Zielinski, S. P. Vudathu, R. Laur, Influence of Different Deviations Allowed for Equality Constraints on Particle Swarm Optimization and Differential Evolution, in: N. Krasnogor, G. Nicosia, M. Pavone, D. Pelta (Eds.), Nature Inspired Cooperative Strategies for Optimization, Springer, Berlin, 2008, pp. 249–259, iSBN 978-3-540-78986-4.

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

[28] G. Eiben, M. Schut, New Ways to Calibrate Evolutionary Algorithms, in: P. Siarry, Z. Michalewicz (Eds.), Advances in Metaheuristics for

Hard Optimization, Natural Computing, Springer, Heidelberg, Germany, 2008, pp. 153–177.

[29] S. B. Hamida, M. Schoenauer, ASCHEA: New Results Using Adaptive Segregational Constraint Handling, in: Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002), Vol. 1, IEEE Service Center, Piscataway, New Jersey, 2002, pp. 884–889.

[30] E. Mezura-Montes, C. A. C. Coello, Identifying On-line Behavior and Some Sources of Difficulty in Two Competitive Approaches for Constrained Optimization, in: 2005 IEEE Congress on Evolutionary Computation (CEC'2005), Vol. 2, IEEE Press, Edinburgh, Scotland, 2005, pp. 1477–1484.

[31] H. Lu, W. Chen, Self-adaptive velocity particle swarm optimization for solving constrained optimization problems, Journal of Global Optimization 41 (3) (2008) 427–445.

[32] H.-P. Schwefel (Ed.), Evolution and Optimization Seeking, Wiley, New York, 1995.

[33] K. Price, R. Storn, J. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Natural Computing Series, Springer-Verlag, 2005.

[34] J. Kennedy, R. C. Eberhart, Swarm Intelligence, Morgan Kaufmann, UK, 2001.

[35] S. Kukkonen, J. Lampinen, Constrained Real-Parameter Optimization with Generalized Differential Evolution, in: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), IEEE, Vancouver, BC, Canada, 2006, pp. 911–918.

[36] K. V. Price, J. I. Rönkkönen, Comparing the uni-modal scaling performance of global and local selection in a mutation-only differential evolution algorithm, in: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), IEEE Press, Vancouver, Canada, 2006, pp. 7387–7394.

[37] J. Lampinen, A Constraint Handling Approach for the Differential Evolution Algorithm, in: Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002), Vol. 2, IEEE Service Center, Piscataway, New Jersey, 2002, pp. 1468–1473.

[38] R. Mallipeddi, P. Suganthan, Ensemble of constraint handling techniques, IEEE Transactions on Evolutionary Computation 14 (4) (2010) 561–579.

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

[40] Y. Wang, Z. Cai, Y. Zhou, W. Zeng, An Adaptive Tradeoff Model for Constrained Evolutionary Optimization, IEEE Transactions on Evolutionary Computation 12 (1) (2008) 80–92.

[41] M. Zhang, W. Luo, X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, Information Sciences 178 (15) (2008) 3043–3074.

[42] D. H. Wolpert, W. G. Macready, No Free Lunch Theorems for Optimization, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 67–82.

[43] T. Takahama, S. Sakai, Constrained Optimization by the ϵ Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites, in: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), IEEE, Vancouver, BC, Canada, 2006, pp. 308–315.

[44] V. L. Huang, A. K. Qin, P. N. Suganthan, Self-adaptative Differential Evolution Algorithm for Constrained Real-Parameter Optimization, in: 2006 IEEE Congress on Evolutionary Computation (CEC'2006), IEEE, Vancouver, BC, Canada, 2006, pp. 324–331.

Appendix A.

The details of the 24 test problems utilized in this work are the following:

g01

Minimize:

$$f(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \tag{A.1}$$

Subject to:

$$\begin{array}{l} g_1(\vec{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0 \\ g_2(\vec{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0 \\ g_3(\vec{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0 \\ g_4(\vec{x}) = -8x_1 + x_{10} \leq 0 \\ g_5(\vec{x}) = -8x_2 + x_{11} \leq 0 \\ g_6(\vec{x}) = -8x_3 + x_{12} \leq 0 \\ g_7(\vec{x}) = -2x_4 - x_5 + x_{10} \leq 0 \\ g_8(\vec{x}) = -2x_6 - x_7 + x_{11} \leq 0 \\ g_9(\vec{x}) = -2x_8 - x_9 + x_{12} \leq 0 \end{array}$$

where $0 \leq x_i \leq 1$ $(i = 1, \ldots, 9)$, $0 \leq x_i \leq 100$ $(i = 10, 11, 12)$, and $0 \leq x_{13} \leq 1$. The feasible global optimum is located at $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ where $f(x^*) =$ -15. where g_1, g_2, g_3, g_7, g_8 g_9 are active constraints.

g02

Minimize:

$$f(\vec{x}) = -\left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right| \tag{A.2}$$

Subject to:

$$\begin{array}{l} g_1(\vec{x}) = 0.75 - \prod_{i=1}^{n} x_i \leq 0 \\ g_2(\vec{x}) = \sum_{i=1}^{n} x_i - 7.5n \leq 0 \end{array}$$

where $n = 20$ and $0 \leq x_i \leq 10$ $(i = 1, \ldots, n)$. The best known solution is located at $x =$ (3.16246061572185, 3.12833142812967, 3.09479212988791, 3.06145059523469, 3.02792915885555, 2.99382606701730, 2.95866871765285, 2.92184227312450, 0.49482511456933, 0.48835711005490, 0.48231642711865, 0.47664475092742, 0.47129550835493, 0.46623099264167, 0.46142004984199, 0.45683664767217, 0.45245876903267, 0.44826762241853, 0.44424700958760, 0.44038285956317),
with $f(x^*) = 0.80361910412559$. g_1 is close to be active.

g03

Minimize:

$$f(\vec{x}) = -\left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i \tag{A.3}$$

Subject to:

$$h(\vec{x}) = \sum_{i=1}^{n} x_i^2 - 1 = 0$$

where $n = 10$ and $0 \leq x_i \leq 1$ $(i = 1, \ldots, n)$. The feasible global minimum is located at $x_i^* = 1/\sqrt{n}$ $(i = 1, \ldots, n)$ where $f(x^*) =$ -1.00050010001000.

g04

Minimize:

$$f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \quad \text{(A.4)}$$

Subject to:

$$\begin{aligned}
g_1(\vec{x}) &= 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 &\leq 0\\
g_2(\vec{x}) &= -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 &\leq 0\\
g_3(\vec{x}) &= 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 &\leq 0\\
g_4(\vec{x}) &= -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 &\leq 0\\
g_5(\vec{x}) &= 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 &\leq 0\\
g_6(\vec{x}) &= -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 &\leq 0
\end{aligned}$$

where: $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, $27 \leq x_i \leq 45$ $(i = 3, 4, 5)$. The feasible global optimum is located at $x^* = (78, 33, 29.9952560256815985, 45, 36.7758129057882073)$ where $f(x^*)$=-30665.539. g_1 and g_6 are active constraints.

g05

Minimize:

$$f(\vec{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3 \quad \text{(A.5)}$$

Subject to:

$$\begin{aligned}
g_1(\vec{x}) &= -x_4 + x_3 - 0.55 &\leq 0\\
g_2(\vec{x}) &= -x_3 + x_4 - 0.55 &\leq 0
\end{aligned}$$

$$\begin{aligned}
h_3(\vec{x}) &= 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 &= 0\\
h_4(\vec{x}) &= 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 &= 0\\
h_5(\vec{x}) &= 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 &= 0
\end{aligned}$$

where $0 \leq x_1 \leq 1200$, $0 \leq x_2 \leq 1200$, $-0.55 \leq x_3 \leq 0.55$, and $-0.55 \leq x_4 \leq 0.55$. The best known solution is located at: x^*=(679.945148297028709, 1026.06697600004691, 0.118876369094410433, $-$0.396233485215178266) where $f(x^*) = 5126.4967140071$.

g06

Minimize:

$$f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3 \quad \text{(A.6)}$$

Subject to:

$$\begin{aligned}
g_1(\vec{x}) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 &\leq 0\\
g_2(\vec{x}) &= (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 &\leq 0
\end{aligned}$$

where $13 \leq x_1 \leq 100$ and $0 \leq x_2 \leq 100$. The feasible global optimum is located at: $x^* = (14.09500000000000064, 0.8429607892154795668)$ where $f(x^*) = -6961.81387558015$. Both constraints are active.

g07

Minimize:

$$\begin{aligned}
f(\vec{x}) \quad = \quad & x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 +\\
& 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45
\end{aligned} \quad \text{(A.7)}$$

Subject to:

$$
\begin{array}{lll}
g_1(\vec{x}) & = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 & \leq 0 \\
g_2(\vec{x}) & = 10x_1 - 8x_2 - 17x_7 + 2x_8 & \leq 0 \\
g_3(\vec{x}) & = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 & \leq 0 \\
g_4(\vec{x}) & = 3(x_1-2)^2 + 4(x_2-3)^2 + 2x_3^2 - 7x_4 - 120 & \leq 0 \\
g_5(\vec{x}) & = 5x_1^2 + 8x_2 + (x_3-6)^2 - 2x_4 - 40 & \leq 0 \\
g_6(\vec{x}) & = x_1^2 + 2(x_2-2)^2 - 2x_1x_2 + 14x_5 - 6x_6 & \leq 0 \\
g_7(\vec{x}) & = 0.5(x_1-8)^2 + 2(x_2-4)^2 + 3x_5^2 - x_6 - 30 & \leq 0 \\
g_8(\vec{x}) & = -3x_1 + 6x_2 + 12(x_9-8)^2 - 7x_{10} & \leq 0
\end{array}
$$

where $-10 \leq x_i \leq 10$ ($i = 1, \ldots, 10$). The feasible global optimum is located at $x^* = (2.17199634142692, 2.3636830416034, 8.77392573913157, 5.09598443745173, 0.990654756560493, 1.43057392853463, 1.32164415364306, 9.82872576524495, 8.2800915887356, 8.3759266477347)$ where $f(x^*) = 24.30620906818$. g_1, g_2, g_3, g_4, g_5, and g_6 are active constraints.

g08

Minimize:

$$f(\vec{x}) = -\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)} \tag{A.8}$$

Subject to:

$$
\begin{array}{lll}
g_1(\vec{x}) & = x_1^2 - x_2 + 1 & \leq 0 \\
g_2(\vec{x}) & = 1 - x_1 + (x_2-4)^2 & \leq 0
\end{array}
$$

where $0 \leq x_1 \leq 10$ and $0 \leq x_2 \leq 10$. The feasible global optimum is located at: $x^* = (1.22797135260752599, 4.24537336612274885)$ with $f(x^*) = -0.0958250414180359$.

g09

Minimize:

$$f(\vec{x}) = (x_1-10)^2 + 5(x_2-12)^2 + x_3^4 + 3(x_4-11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \tag{A.9}$$

Subject to:

$$
\begin{array}{lll}
g_1(\vec{x}) & = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 & \leq 0 \\
g_2(\vec{x}) & = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 & \leq 0 \\
g_3(\vec{x}) & = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 & \leq 0 \\
g_4(\vec{x}) & = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 & \leq 0
\end{array}
$$

where $-10 \leq x_i \leq 10$ ($i = 1, \ldots, 7$). The feasible global optimum is located at:x^*=(2.33049935147405174, 1.95137236847114592, −0.477541399510615805, 4.36572624923625874, −0.624486959100388983, 1.03813099410962173, 1.5942266780671519) with $f(x^*) = 680.630057374402$. g_1 and g_4 are active constraints.

g10

Minimize:

$$f(\vec{x}) = x_1 + x_2 + x_3 \tag{A.10}$$

Subject to:

$$
\begin{array}{lll}
g_1(\vec{x}) & = -1 + 0.0025(x_4 + x_6) & \leq 0 \\
g_2(\vec{x}) & = -1 + 0.0025(x_5 + x_7 - x_4) & \leq 0 \\
g_3(\vec{x}) & = -1 + 0.01(x_8 - x_5) & \leq 0 \\
g_4(\vec{x}) & = -x_1 x_6 + 833.33252 x_4 + 100 x_1 - 83333.333 & \leq 0 \\
g_5(\vec{x}) & = -x_2 x_7 + 1250 x_5 + x_2 x_4 - 1250 x_4 & \leq 0 \\
g_6(\vec{x}) & = -x_3 x_8 + 1250000 + x_3 x_5 - 2500 x_5 & \leq 0
\end{array}
$$

where $100 \leq x_1 \leq 10000$, $1000 \leq x_i \leq 10000$, $(i = 2, 3)$, $10 \leq x_i \leq 1000$, $(i = 4, \ldots, 8)$. The feasible global optimum is located at $x^* = (579.306685017979589, 1359.97067807935605, 5109.97065743133317, 182.01769963061534, 295.601173702746792, 217.982300369384632, 286.41652592786852, 395.601173702746735)$ with $f(x^*) = 7049.24802052867$. g_1, g_2, and g_3 are active constraints.

g11

Minimize:

$$f(\vec{x}) = x_1^2 + (x_2 - 1)^2 \tag{A.11}$$

Subject to:

$$h(\vec{x}) = x_2 - x_1^2 = 0$$

where: $-1 \leq x_1 \leq 1$, $-1 \leq x_2 \leq 1$. The feasible global optimum is located at:$x^* = (\pm 1/\sqrt{2}, 1/2)$ with $f(x^*) = 0.7499$.

g12

Minimize:

$$f(\vec{x}) = -\frac{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2}{100} \tag{A.12}$$

Subject to:

$$g_1(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0$$

where $0 \leq x_i \leq 10$ $(i = 1, 2, 3)$ and $p, q, r = 1, 2, \ldots, 9$. The feasible region consists on 9^3 disjoint spheres. A point (x_1, x_2, x_3) is feasible if and only if there exist p, q, r such that the above inequality holds. The feasible global optimum is located at $x^* = (5, 5, 5)$ with $f(x^*) = -1$.

g13

Minimize:

$$f(\vec{x}) = e^{x_1 x_2 x_3 x_4} \tag{A.13}$$

Subject to:

$$\begin{aligned} h_1(\vec{x}) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 &= 0 \\ h_2(\vec{x}) &= x_2 x_3 - 5 x_4 x_5 &= 0 \\ h_3(\vec{x}) &= x_1^3 + x_2^3 + 1 &= 0 \end{aligned}$$

where $-2.3 \leq x_i \leq 2.3$ $(i = 1, 2)$ and $-3.2 \leq x_i \leq 3.2$ $(1 = 3, 4, 5)$. The feasible global optimum is at $\vec{x^*} = (-1.71714224003, 1.59572124049468, 1.8272502406271, -0.763659881912867, -0.76365986736498)$ with $f(\vec{x^*}) = 0.053941514041898$.

g14

Minimize:

$$f(\vec{x}) = \sum_{i=1}^{10} x_i \left(c_i + \ln \frac{x_i}{\sum_{j=1}^{10} x_j} \right) \tag{A.14}$$

Subject to:

$$\begin{aligned} h_1(\vec{x}) &= x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 &= 0 \\ h_2(\vec{x}) &= x_4 + 2x_5 + x_6 + x_7 - 1 &= 0 \\ h_3(\vec{x}) &= x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 &= 0 \end{aligned}$$

where $0 < x_i \leq 10$ $(i = 1, ..., 10)$, and c_1 = -6.089, c_2 = -17.164, c_3 = -34.054, c_4 = -5.914, c_5 = -24.721, c_6 = -14.986, c_9 = -26.662, c_{10} = -22.179. The best known solution is at $x^* = (0.0406684113216282, 0.147721240492452, 0.783205732104114, 0.00141433931889084, 0.485293636780388, 0.000693183051556082, 0.0274052040687766, 0.0179509660214818, 0.0373268186859717, 0.0968844604336845)$ with $f(x^*) = -47.7648884594915$.

g15

Minimize:

$$f(\vec{x}) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1 x_2 - x_1 x_3 \tag{A.15}$$

Subject to:

$$\begin{aligned} h_1(\vec{x}) &= x_1^2 + x_2^2 + x_3^2 - 25 &= 0 \\ h_2(\vec{x}) &= 8x_1 + 14x_2 + 7x_3 - 56 &= 0 \end{aligned}$$

where $0 \leq x_i \leq 10$ $(i = 1, 2, 3)$. The best known solution is at $x^* = (3.51212812611795133, 0.216987510429556135, 3.55217854929179921)$ with $f(x^*) = 961.715022289961$.

g16

Minimize:

$$\begin{aligned} f(\vec{x}) \quad = \quad & 0.000117 y_{14} + 0.1365 + 0.00002358 y_{13} + 0.000001502 y_{16} \\ & + 0.0321 y_{12} + 0.004324 y_5 + 0.0001 \tfrac{c_{15}}{c_{16}} \\ & + 37.48 \tfrac{y_2}{c_{12}} - 0.0000005843 y_{17} \end{aligned} \tag{A.16}$$

Subject to:

$$
\begin{array}{lll}
g_1(\vec{x}) & = \frac{0.28}{0.72}y_5 - y_4 & \leq 0 \\
g_2(\vec{x}) & = x_3 - 1.5x_2 & \leq 0 \\
g_3(\vec{x}) & = 3496\frac{y_2}{c_{12}} - 21 & \leq 0 \\
g_4(\vec{x}) & = 110.6 + y_1 - \frac{62212}{c_{17}} & \leq 0 \\
g_5(\vec{x}) & = 213.1 - y_1 & \leq 0 \\
g_6(\vec{x}) & = y_1 - 405.23 & \leq 0 \\
g_7(\vec{x}) & = 17.505 - y_2 & \leq 0 \\
g_8(\vec{x}) & = y_2 - 1053.6667 & \leq 0 \\
g_9(\vec{x}) & = 11.275 - y_3 & \leq 0 \\
g_{10}(\vec{x}) & = y_3 - 35.03 & \leq 0 \\
g_{11}(\vec{x}) & = 214.228 - y_4 & \leq 0 \\
g_{12}(\vec{x}) & = y_4 - 665.585 & \leq 0 \\
g_{13}(\vec{x}) & = 7.458 - y_5 & \leq 0 \\
g_{14}(\vec{x}) & = y_5 - 584.463 & \leq 0 \\
g_{15}(\vec{x}) & = 0.961 - y_6 & \leq 0 \\
g_{16}(\vec{x}) & = y_6 - 265.916 & \leq 0 \\
g_{17}(\vec{x}) & = 1.612 - y_7 & \leq 0 \\
g_{18}(\vec{x}) & = y_7 - 7.046 & \leq 0 \\
g_{19}(\vec{x}) & = 0.146 - y_8 & \leq 0 \\
g_{20}(\vec{x}) & = y_8 - 0.222 & \leq 0 \\
g_{21}(\vec{x}) & = 107.99 - y_9 & \leq 0 \\
g_{22}(\vec{x}) & = y_9 - 273.366 & \leq 0 \\
g_{23}(\vec{x}) & = 922.693 - y_{10} & \leq 0 \\
g_{24}(\vec{x}) & = y_{10} - 1286.105 & \leq 0 \\
g_{25}(\vec{x}) & = 926.832 - y_{11} & \leq 0 \\
g_{26}(\vec{x}) & = y_{11} - 1444.046 & \leq 0
\end{array}
$$

$$
\begin{array}{lll}
g_{27}(\vec{x}) & = 18.766 - y_{12} & \leq 0 \\
g_{28}(\vec{x}) & = y_{12} - 537.141 & \leq 0 \\
g_{29}(\vec{x}) & = 1072.163 - y_{13} & \leq 0 \\
g_{30}(\vec{x}) & = y_{13} - 3247.039 & \leq 0 \\
g_{31}(\vec{x}) & = 8961.448 - y_{14} & \leq 0 \\
g_{32}(\vec{x}) & = y_{14} - 26844.086 & \leq 0 \\
g_{33}(\vec{x}) & = 0.063 - y_{15} & \leq 0 \\
g_{34}(\vec{x}) & = y_{15} - 0.386 & \leq 0 \\
g_{35}(\vec{x}) & = 71084.33 - y_{16} & \leq 0 \\
g_{36}(\vec{x}) & = -140000 + y_{16} & \leq 0 \\
g_{37}(\vec{x}) & = 2802713 - y_{17} & \leq 0 \\
g_{38}(\vec{x}) & = y_{17} - 12146108 & \leq 0
\end{array}
$$

where

$$
\begin{aligned}
y_1 &= x_2 + x_3 + 41.6\\
c_1 &= 0.024x_4 - 4.62\\
y_2 &= \tfrac{12.5}{c_1} + 12\\
c_2 &= 0.0003535x_1^2 + .5311x_1 + 0.08705y_2x_1\\
c_3 &= 0.052x_1 + 78 + 0.002377y_2x_1\\
y_3 &= \tfrac{c_2}{c_3}\\
y_4 &= 19y_3\\
c_4 &= 0.04782(x_1 - y_3) + \tfrac{0.1956(x_1-y_3)^2}{x_2} + 0.6376y_4 + 1.594y_3\\
c_5 &= 100x_2\\
c_6 &= x_1 - y_3 - y_4\\
c_7 &= 0.950 - \tfrac{c_4}{c_5}\\
y_5 &= c_6c_7\\
y_6 &= x_1 - y_5 - y_4 - y_3\\
c_8 &= (y_5 + y_4)0.995\\
y_7 &= \tfrac{c_8}{y_1}\\
y_8 &= \tfrac{c_8}{3798}\\
c_9 &= y_7 - \tfrac{0.0663y_7}{y_8} - 0.3153\\
y_9 &= \tfrac{96.82}{c_9} + 0.321y_1\\
y_{10} &= 1.29y_5 + 1.258y_4 + 2.29y_3 + 1.71y_6\\
y_{11} &= 1.71x_1 - 0.452y_4 + 0.580y_3\\
c_{10} &= \tfrac{12.3}{752.3}\\
c_{11} &= (1.75y_2)(0.995x_1)\\
c_{12} &= 0.995y_{10} + 1998\\
y_{12} &= c_{10}x_1 + \tfrac{c_{11}}{c_{12}}\\
y_{13} &= c_{12} + 1.75y_2\\
y_{14} &= 3623 + 64.4x_2 + 58.4x_3 + \tfrac{146312}{y_9+x_5}\\
c_{13} &= 0.995y_{10} + 60.8x_2 + 48x_4 - 0.1121y_{14} - 5095\\
y_{15} &= \tfrac{y_{13}}{c_{13}}\\
y_{16} &= 148000 - 331000y_{15} + 40y_{13} - 61y_{15}y_{13}\\
c_{14} &= 2324y_{10} - 28740000y_2\\
y_{17} &= 14130000 - 1328y_{10} - 531y_{11} + \tfrac{c_{14}}{c_{12}}\\
c_{15} &= \tfrac{y_{13}}{y_{15}} - \tfrac{y_{13}}{0.52}\\
c_{16} &= 1.104 - 0.72y_{15}\\
c_{17} &= y_9 + x_5
\end{aligned}
$$

and where $704.4148 \leq x_1 \leq 906.3855$, $68.6 \leq x_2 \leq 288.88$, $0 \leq x_3 \leq 134.75$, $193 \leq x_4 \leq 287.0966$, and $25 \leq x_5 \leq 84.1988$.

The best known solution is at: $x^* = (705.174537070090537, 68.5999999999999943, 102.899999999999991, 282.324931593660324, 37.5841164258054832)$ with $f(x^*) = -1.90515525853479$.

g17

Minimize:

$$f(\vec{x}) = f(x_1) + f(x_2) \tag{A.17}$$

where

$$f_1(x_1) = \begin{cases} 30x_1 & 0 \le x_1 < 300 \\ 31x_1 & 300 \le x_1 < 400 \end{cases}$$

$$f_2(x_2) = \begin{cases} 28x_2 & 0 \le x_2 < 100 \\ 29x_2 & 100 \le x_2 < 200 \\ 30x_2 & 200 \le x_2 < 1000 \end{cases}$$

Subject to:

$$h_1(\vec{x}) = -x_1 + 300 - \frac{x_3 x_4}{131.078}\cos(1.48477 - x_6) + \frac{0.90798 x_3^2}{131.078}\cos(1.47588)$$

$$h_2(\vec{x}) = -x_2 - \frac{x_3 x_4}{131.078}\cos(1.48477 + x_6) + \frac{0.90798 x_4^2}{131.078}\cos(1.47588)$$

$$h_3(\vec{x}) = -x_5 - \frac{x_3 x_4}{131.078}\sin(1.48477 + x_6) + \frac{0.90798 x_4^2}{131.078}\sin(1.47588)$$

$$h_4(\vec{x}) = 200 - \frac{x_3 x_4}{131.078}\sin(1.48477 - x_6) + \frac{0.90798 x_4^2}{131.078}\sin(1.47588)$$

and where $0 \le x_1 \le 400$, $0 \le x_2 \le 1000$, $340 \le x_3 \le 420$, $340 \le x_4 \le 420$, $-1000 \le x_5 \le 1000$, and $0 \le x_6 \le 0.5236$. The best known solution is at $x^* = (201.784467214523659, 99.9999999999999005, 383.071034852773266, 420, -10.9076584514292652, 0.0731482312084287128)$ with $f(x^*) = 8853.53967480648$.

g18

Minimize:

$$f(\vec{x}) = -0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7) \tag{A.18}$$

Subject to:

$$\begin{aligned}
g_1(\vec{x}) &= x_3^2 + x_4^2 - 1 && \le 0 \\
g_2(\vec{x}) &= x_9^2 - 1 && \le 0 \\
g_3(\vec{x}) &= x_5^2 + x_6^2 - 1 && \le 0 \\
g_4(\vec{x}) &= x_1^2 + (x_2 - x_9)^2 - 1 && \le 0 \\
g_5(\vec{x}) &= (x_1 - x_5)^2 + (x_2 - x_6)^2 - 1 && \le 0 \\
g_6(\vec{x}) &= (x_1 - x_7)^2 + (x_2 - x_8)^2 - 1 && \le 0
\end{aligned}$$

$$\begin{aligned}
g_7(\vec{x}) &= (x_3 - x_5)^2 + (x_4 - x_6)^2 - 1 && \le 0 \\
g_8(\vec{x}) &= (x_3 - x_7)^2 + (x_4 - x_8)^2 - 1 && \le 0 \\
g_9(\vec{x}) &= x_7^2 + (x_8 - x_9)^2 - 1 && \le 0 \\
g_{10}(\vec{x}) &= x_2x_3 - x_1x_4 && \le 0 \\
g_{11}(\vec{x}) &= -x_3x_9 && \le 0 \\
g_{12}(\vec{x}) &= x_5x_9 && \le 0 \\
g_{13}(\vec{x}) &= x_6x_7 - x_5x_8 && \le 0
\end{aligned}$$

where $-10 \le x_i \le 10$ $(i = 1, ..., 8)$ and $0 \le x_9 \le 20$. The best known solution is at: $x^* = (-0.657776192427943163, -0.153418773482438542, 0.323413871675240938, -0.946257611651304398, -0.657776194376798906, -0.753213434632691414, 0.323413874123576972, -0.346462947962331735, 0.59979466285217542)$ with $f(x^*) = -0.866025403784439$.

g19

Minimize:

$$f(\vec{x}) = \sum_{j=1}^{5}\sum_{i=1}^{5} c_{ij}x_{(10+j)}x_{(10+j)} + 2\sum_{j=1}^{5} d_j x^3_{(10+j)} - \sum_{i=1}^{10} b_i x_i \tag{A.19}$$

Subject to:

$$g_j(\vec{x}) = -2\sum_{i=1}^{5} c_{ij}x_{(10+i)} - e_j + \sum_{i=1}^{10} a_{ij}x_i \leq 0 \quad j = 1, \ldots, 5$$

where $\vec{b}$ = [-40, -2, -0.25, -4, -4, -1, -40, -60, 5, 1] and the remaining values are taken from Table A.1, $0 \leq x_i \leq 10$ $(i = 1, \ldots, 15)$. The best known solution is at $x^* =$ (1.66991341326291344e^{-17}, 3.95378229282456509e^{-16}, 3.94599045143233784, 1.06036597479721211e^{-16}, 3.2831773458454161, 9.99999999999999822, 1.12829414671605333e^{-17}, 1.2026194599794709e^{-}17, 2.50706276000769697e^{-15}, 2.24624122987970677e^{-15}, 0.370764847417013987, 0.278456024942955571, 0.523838487672241171, 0.388620152510322781, 0.298156764974678579) with $f(x^*) = 32.6555929502463$.

j	1	2	3	4	5
e_j	-15	-27	-36	-18	-12
c_{1j}	30	-20	-10	32	-10
c_{2j}	-20	39	-6	-31	32
c_{3j}	-10	-6	10	-6	-10
c_{4j}	32	-31	-6	39	-20
c_{5j}	-10	32	-10	-20	30
d_j	4	8	10	6	2
a_{1j}	-16	2	0	1	0
a_{2j}	0	-2	0	0.4	2
a_{3j}	-3.5	0	2	0	0
a_{4j}	0	-2	0	-4	-1
a_{5j}	0	-9	-2	1	-2.8
a_{6j}	2	0	-4	0	0
a_{7j}	-1	-1	-1	-1	-1
a_{9j}	1	2	3	4	5
a_{10j}	1	1	1	1	1

Table A.1: Data set for test problem g19

g20

Minimize:

$$f(\vec{x}) = \sum_{i=1}^{24} a_i x_i \tag{A.20}$$

Subject to:

$g_i(\vec{x}) = \frac{(x_i + x_{(i+12)})}{\sum_{j=1}^{24} x_j + e_i} \leq 0$ $\quad i = 1, 2, 3$

$g_i(\vec{x}) = \frac{(x_{i+3} + x_{(i+15)})}{\sum_{j=1}^{24} x_j + e_i} \leq 0$ $\quad i = 4, 5, 6$

$h_1(\vec{x}) = \frac{x_{(i+12)}}{b_{i+12} \sum_{j=13}^{24} \frac{x_j}{b_j}} - \frac{c_i x_i}{40 b_i \sum_{j=1}^{12} \frac{x_j}{b_j}} = 0$ $\quad i = 1, \ldots, 12$

$h_{13}(\vec{x}) = \sum_{i=1}^{24} x_i - 1 = 0$

$h_{14}(\vec{x}) = \sum_{i=1}^{12} \frac{x-i}{d_i} + k \sum_{i=13}^{24} \frac{x_i}{b_i} - 1.671 = 0$

i	a_i	b_i4	c_i	$4d_i$	e_i
1	0.0693	44.094	123.7	31.244	0.1
2	0.0577	58.12	31.7	36.12	0.3
3	0.05	58.12	45.7	34.784	0.4
4	0.2	137.4	14.7	92.7	0.3
5	0.26	120.9	84.7	82.7	0.6
6	0.55	170.9	27.7	91.6	0.3
7	0.06	62.501	49.7	56.708	
8	0.1	84.94	7.1	82.7	
9	0.12	133.425	2.1	80.8	
10	0.18	82.507	17.7	64.517	
11	0.1	46.07	0.85	49.4	
12	0.09	60.097	0.64	49.1	
13	0.0693	44.094			
14	0.5777	6	58.12		
15	0.05	58.12			
16	0.2	137.4			
17	0.26	120.9			
18	0.55	170.9			
19	0.06	62.501			
20	0.1	84.94			
21	0.12	133.425			
22	0.18	82.507			
23	0.1	46.07			
24	0.09	60.097			

Table A.2: Data set for test problem g20

where $k = (0.7302)(530)(14.740)$ and the data set is detailed Table A.2. $0 \leq x_i \leq 10$ ($i = 1, \ldots, 24$). The best known solution is at $x^* = (1.28582343498528086e^{-18}$, $4.83460302526130664e^{-34}$, 0, 0, $6.30459929660781851e^{-18}$, $7.57192526201145068e^{-34}$, $5.03350698372840437e^{-34}$, $9.28268079616618064e^{-34}$, 0, $1.76723384525547359e^{-}17$, $3.55686101822965701e^{-34}$, $2.99413850083471346e^{-34}$, 0.158143376337580827, $2.29601774161699833e^{-19}$, $1.06106938611042947e^{-18}$, $1.31968344319506391e^{-18}$, 0.530902525044209539, 0, $2.89148310257773535e^{-18}$, $3.34892126180666159e^{-18}$, 0, 0.310999974151577319, $5.41244666317833561e^{-05}$, $4.84993165246959553e^{-16}$). This solution is slightly infeasible and no feasible solution has been reported so far.

g21

Minimize:

$$f(\vec{x}) = x_1 \tag{A.21}$$

Subject to:

$$g_1(\vec{x}) = -x_1 + 35x_2^{0.6} + 35x_3^{0.6} \leq 0$$

$$\begin{aligned} h_1(\vec{x}) &= -300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 &= 0 \\ h_2(\vec{x}) &= 100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 24x_4x_7 - 15536.5 &= 0 \end{aligned}$$

$$\begin{aligned} h_3(\vec{x}) &= -x_5 + \ln(-x_4 + 900) &= 0 \\ h_4(\vec{x}) &= -x_6 + \ln(x_4 + 300) &= 0 \\ h_5(\vec{x}) &= -x_7 + \ln(-2x_4 + 700) &= 0 \end{aligned}$$

where $0 \leq x_1 \leq 1000$, $0 \leq x_2, x_3 \leq 40$, $100 \leq x_4 \leq 300$, $6.3 \leq x_5 \leq 6.7$, $5.9 \leq x_6 \leq 6.4$, and $4.5 \leq x_7 \leq 6.25$. The best known solution is at: $x^* = (193.724510070034967$, $5.56944131553368433e^{-27}$, 17.3191887294084914, 100.047897801386839, 6.68445185362377892, 5.99168428444264833, 6.21451648886070451) with $f(x^*) = 193.724510070035$.

g22

Minimize:

$$f(\vec{x}) = x_1 \tag{A.22}$$

Subject to:

$$\begin{aligned} g_1(\vec{x}) &= -x_1 + x_2^{0.6} + x_3^{0.6}x_4^{0.6} &\leq 0 \\ h_1(\vec{x}) &= x_5 - 100000x_8 + 1 * 10^7 &= 0 \\ h_2(\vec{x}) &= x_6 - 100000x_8 - 100000x_9 &= 0 \\ h_3(\vec{x}) &= x_7 - 100000x_9 - 5 * 10^7 &= 0 \\ h_4(\vec{x}) &= x_5 - 100000x_{10} - 3.3 * 10^7 &= 0 \\ h_5(\vec{x}) &= x_6 - 100000x_{11} - 4.4 * 10^7 &= 0 \\ h_6(\vec{x}) &= x_7 - 100000x_{12} - 6.6 * 10^7 &= 0 \\ h_7(\vec{x}) &= x_5 - 120x_2x_{13} &= 0 \\ h_8(\vec{x}) &= x_6 - 80x_3x_{14} &= 0 \\ h_9(\vec{x}) &= x_7 - 40x_4x_{15} &= 0 \\ h_{10}(\vec{x}) &= x_8 - x_{11} + x_{16} &= 0 \\ h_{11}(\vec{x}) &= x_9 - x_{12} + x_{17} &= 0 \end{aligned}$$

$$\begin{aligned} h_{12}(\vec{x}) &= -x_{18} + \ln(x_{10} - 100) &= 0 \\ h_{13}(\vec{x}) &= -x_{19} + \ln(-x_8 + 300) &= 0 \\ h_{14}(\vec{x}) &= -x_{20} + \ln(x_{16}) &= 0 \\ h_{15}(\vec{x}) &= -x_{21} + \ln(-x_9 + 400) &= 0 \\ h_{16}(\vec{x}) &= -x_{22} + \ln(x_{17}) &= 0 \\ h_{17}(\vec{x}) &= -x_8 - x_{10} + x_{13}x_{18} - x_{13}x_{19} + 400 &= 0 \\ h_{18}(\vec{x}) &= x_8 - x_9 - x_{11} + x_{14}x_{20} - x_{14}x_{21} + 400 &= 0 \\ h_{19}(\vec{x}) &= x_9 - x_{12} - 4.60517x_{15} + x_{15}x_{22} + 100 &= 0 \end{aligned}$$

where $0 \leq x_1 \leq 20000$, $0 \leq x_2, x_3, x_4 \leq 1 * 10^6$, $0 \leq x_5, x_6, x_7 \leq 4 * 10^7$, $100 \leq x_8 \leq 299.99$, $100 \leq x_9 \leq 399.99$, $100.01 \leq x_{10} \leq 300$, $100 \leq x_{11} \leq 400$, $100 \leq x_{12} \leq 600$, $0 \leq x_{13}, x_{14}, x_{15} \leq 500$, $0.01 \leq x_{16} \leq 300$, $0.01 \leq x_{17} \leq 400$, $-4.7 \leq x_{18}, x_{19}, x_{20}, x_{21}, x_{22} \leq 6.25$. The best known solution is at: $x^* = (236.430975504001054$, 135.82847151732463, 204.818152544824585, 6446.54654059436416, 3007540.83940215595, 4074188.65771341929, 32918270.5028952882, 130.075408394314167, 170.817294970528621, 299.924591605478554, 399.258113423595205, 330.817294971142758, 184.51831230897065, 248.64670239647424, 127.658546694545862, 269.182627528746707, 160.000016724090955, 5.29788288102680571, 5.13529735903945728, 5.59531526444068827, 5.43444479314453499, 5.07517453535834395) with $f(x^{*}) = 236.430975504001$.

g23

Minimize:

$$f(\vec{x}) = -9x_5 - 15x_8 + 6x_1 + 16x_2 + 10(x_6 + x_7) \tag{A.23}$$

Subject to:

$$\begin{aligned}
g_1(\vec{x}) &= x_9x_3 + 0.02x_6 - 0.025x_5 && \leq 0 \\
g_2(\vec{x}) &= x_9x_4 + 0.02x_7 - 0.015x_8 && \leq 0 \\
h_1(\vec{x}) &= x_1 + x_2 - x_3 - x_4 && = 0 \\
h_2(\vec{x}) &= 0.03x_1 + 0.01x_2 - x_9(x_3 + x_4) && = 0 \\
h_3(\vec{x}) &= x_3 + x_6 - x_5 && = 0 \\
h_4(\vec{x}) &= x_4 + x_7 - x_8 && = 0
\end{aligned}$$

where $0 \leq x_1, x_2, x_6 \leq 300$, $0 \leq x_3, x_5, x_7 \leq 100$, $0 \leq x_4, x_8 \leq 200$, and $0.01 \leq x_9 \leq 0.03$. The best known solution is at:x^{*} = (0.00510000000000259465, 99.9947000000000514, 9.01920162996045897e^{-18}, 99.9999000000000535, 0.000100000000027086086, 2.75700683389584542e^{-14}, 99.9999999999999574, 2000.0100000100000100008) with $f(x^{*}) = -400.055099999999584$.

g24

Minimize:

$$f(\vec{x}) = -x_1 - x_2 \tag{A.24}$$

Subject to:

$$\begin{aligned}
g_1(\vec{x}) &= -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 && \leq 0 \\
g_2(\vec{x}) &= -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x - 2 - 36 && \leq 0
\end{aligned}$$

where $0 \leq x_1 \leq 3$ and $0 \leq x_2 \leq 4$. The feasible global minimum is at:x^{*} = (2.329520197477623, 17849307411774) with $f(x^{*}) = -5.50801327159536$. This problem has a feasible region consisting on two disconnected sub-regions.