μJADEε: Micro adaptive differential evolution to solve constrained optimization problems

Aldo Márquez-Grajales and Efrén Mezura-Montes,
Artificial Intelligence Research Center
University of Veracruz, Sebastián Camacho 5, Xalapa, Veracruz, 91000, México.
Email: li.aldomg@gmail.com, emezura@uv.mx

***Abstract*—A highly competitive micro evolutionary algorithm to solve unconstrained optimization problems called μJADE (micro adaptive differential evolution), is adapted to deal with constrained search spaces. Two constraint-handling techniques (the feasibility rules and the ε-constrained method) are tested in μJADE and their performance is analyzed. The most competitive version is then compared against two highly-competitive algorithms for constrained optimization when solving a well-known set of 36 test problems, and also against a small population algorithm tested on another well-known set of thirteen problems. The results show that μJADE provides a better performance when coupled with the ε-constrained method and also that its results are competitive against those provided by state-of-the-art approaches.**

I. INTRODUCTION

Several real-world optimization problems are hard to solve [1]. Normally, those problems have constraints that delimit the feasible region within the whole search space, and the search algorithm must find it, sample it, and find the best feasible solution. Evolutionary algorithms (EAs) have been very competitive to deal with such search problems, and differential evolution (DE) has been particularly successful [2].

There are research efforts on the usage of DE with small populations (*micro differential evolution, μDE*), but mainly focused on unconstrained optimization problems. Caraffini et al. in [3] applied a μDE to solve problems with high dimensionality using a simple local search, which consisted in moving the best vector along the axes. Such mechanism was applied at every generation. They called this algorithm μDEA. In their paper, the algorithm was compared against state-of-the-art DE variants like Self-Adaptive Differential Evolution (SADE), Adaptive Differential Evolution (JADE) and Modified Differential Evolution with p-Best Crossover (MDE-pBX), in four different benchmarks (CEC2005, BBOB2010, CEC2008, CEC2010). The result of the comparison indicated that μDEA outperformed all the algorithms aforementioned, concluding that small populations can work well with DE in unconstrained search spaces.

Another μDE was proposed by Salehinejad et al. in [4], where they changed the concept of the mutation factor by converting it from a real number into a vector, i.e., a different random value for the mutation factor was generated for each mutation applied to the population. The authors called their proposal *Micro Differential Evolution with Vectorized random Mutation factor (MDEVM)*. MDEVM was compared against the Standard Micro-Differential Evolution (SMDE) and the Micro-differential evolution with scalar random mutation factor (MDESM) in 28 functions of the CEC 2013 benchmark. The results showed that MDEVM had a higher performance than those of the compared algorithms. Such conclusion remarks the viability of using a small population in DE to solve high-dimensional problems.

In [5], Brown et al. proposed a small population version of the Self-adaptive differential evolution (μJADE) for unconstrained continuous optimization. μJADE uses an adaptive mechanism of the F and CR parameters, similar to the original JADE, i.e., adapting their values every $max(100, 10D)$ generations and proposing a new mutation operator denoted as current-by-rand-to-pbest/1. This approach was tested in the classical multimodal benchmark in 30 dimensions and it was compared against some variants that use conventional sized populations [6]. The results showed that μJADE is more reliable than state-of-the-art DE algorithms.

It is important to mention that, the main limitation of the μEAs is the difficulty maintaining the diversity of solutions. This limitation causes a premature convergence on possible local optima due to the small populations. This problem hardly occurs in conventionally population sized algorithms. Most of the μEAs have implemented mechanisms to deal with this situation, getting competitive results in the context where they were tested.

On the other hand, there are also works on μEAs for constrained spaces. In [7], Fuentes-Cabrera et al. implemented a local version of particle swarm optimization with a small population (μPSO) to solve the thirteen functions described in [8].

As it can be seen, μDE has been, to the best of the authors' knowledge, extensively tested on unconstrained search spaces, but overlooked or scarcely analyzed in constrained search spaces. For that reason, the main motivation of this work lies on studying the behavior of one of the μDE algorithms, μJADE in this case, when solving numerical constrained problems. Two constraint-handling mechanisms are added to μJADE and their performance is studied.

The rest of this document is divided as follows: Section II introduces the basics on μJADE. Section III describes the constraint-handling techniques added in μJADE. Section IV presents the experiments and results obtained. Finally, Section V includes the conclusions and future work.

II. μJADE: MICRO ADAPTIVE DIFFERENTIAL EVOLUTION

μJADE has four elements which are described in the next subsections.

A. Mutation operator

The mutation operator introduced in μJADE is shown in Equation 1 and it is denoted as current-by-rand-to-pbest/1.

$$v_i = x_i + F_i(x^p_{best} - x_a) + F_i(x_b - \tilde{x_c}) \quad (1)$$

where v_i is the mutation vector to be generated, x_i is the target vector, F_i is the current value for the mutation factor (detailed later), x^p_{best} is a randomly chosen vector from the the best $(3/NP)\%$ vectors of the population, x_a and x_b are vectors chosen randomly from the current population, and $\tilde{x_c}$ is a vector randomly chosen from the union of the current population and the auxiliary file where vectors are stored. The main idea of this operator was to improve the exploratory capacity of small populations maintaining a good convergence. When $x_i \not\approx x_a$ and $x_i \neq x_a$, current-by-rand-to-pbest/1 is exploratory. However, when $x_i \approx x_a$, current-by-rand-topbest/1 is similar to current-to-pbest/1. The objective was to accelerate convergence at the end of the optimization process and reducing the probability of finding local optimum in early stages of optimization. Moreover, two constraints used in the original JADE were removed: (1) $\tilde{x_c} \neq x_b$, and (2) $\tilde{x_c} \neq x_a$, both to avoid a greedy behavior of the operator.

B. F and CR adaptation

μJADE introduced a modification in the adaptation of the F and CR parameters to prevent a quick value decrease at early generations and avoiding the diversity loss on small population algorithms. This modification consists in the updating process of F and CR at every $max(100, 10D)$ generations rather than at each generation, where D is the dimensionality of the optimization problem.

C. Perturbation

A perturbation was introduced to provide μJADE with the ability to escape from local optima and increase diversity. This perturbation was proposed by Farfaj et al. in [9], and it is incorporated after the crossover operator as in Equations 2 and 3 . To indicate that the perturbation was applied to a particular variable of a given vector, it is marked with a 0 in a binary vector b as shown in Equation 4.

$$r_{i,j} = rand(0, 1) \quad (2)$$

$$u_{i,j} = \begin{Bmatrix} L_j + rand(0,1)(U_j - L_j) & \text{if} & r_{i,j} \leqslant 0.005 \\ u_{i,j} & & otherwise \end{Bmatrix} \quad (3)$$

$$b_{i,j} = \begin{Bmatrix} 0 & \text{if} & r_{i,j} \leqslant 0.005 \\ b_{i,j} & & otherwise \end{Bmatrix} \quad (4)$$

D. Restarts

One way to avoid stagnation on μEAs is restarting the population at every generation. Despite the aforementioned modifications, it is possible that the search stagnates on local optima. For that reason, the last modification for μJADE was to apply this mechanism at every $max(1000, 100D)$ generations if the best solution did not improve. It is worth noticing that the restart mechanism is applied by excluding the best vector in the current population.

The complete pseudocode of μJADE is shown in Algorithm 1.

III. CONSTRAINT-HANDLING TECHNIQUES

Different constraint-handling techniques are reported in the specialized literature of bio-inspired algorithms [2]. In this section, two competitive techniques are introduced and adopted in this research: the *feasibility rules* and the *ε-constrained method*. Those techniques were selected because their competitive performance and also because they modify only the selection criteria in μJADE, i.e., they do not significantly change the algorithm [2].

A. Feasibility Rules

The set of three feasibility rules is one of the most popular constraint-handlers [10]. This mechanism consists in three selection criteria as follows: (1) when two feasible individuals are compared, the individual with the best objective function value is chosen, (2) if one of the individuals is infeasible and the other one is feasible, then the feasible individual is chosen, and (3) when two infeasible individuals are compared, the individual with the lowest sum of constraint violation is chosen. The idea is to favor good feasibles individuals .

B. ε-constrained method

Takahama et al. in [11] introduced the ε-constrained method, which converts a constrained optimization problem into an unconstrained optimization problem. To evaluate if an individual is feasible or not, the ε-constrained method defines a *constraint violation* $\phi(x)$ whose value is the same as the sum of constraint violation used in the feasibility rules, see Equation 5:

$$\phi(x) = \sum_j \|max\{0, g_j(x)\}\|^p + \sum_j \|h_j(x)\|^p \quad (5)$$

where p is a positive number, 2 in this case. Furthermore, ε *level comparisons* are defined as an order relation on a pair of objective function constraint violation values $(f(x), \phi(x))$. Let $f_1(f_2)$ and $\phi_1(\phi_2)$ be the objective function and constraint violation values of vectors $x_1(x_2)$, respectively. The ε level comparisons are defined in Equation 6:

$$(f_1, \phi_1) \leqslant_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{Bmatrix} f_1 \leqslant f_2 & \text{if} & \phi_1, \phi_2 \leqslant \varepsilon \\ f_1 \leqslant f_2 & \text{if} & \phi_1 = \phi_2 \\ \phi_1 < \phi_2 & & otherwise \end{Bmatrix} \quad (6)$$

It is worth noticing that when $\varepsilon = \infty$ two infeasible vectors are compared based on their objective function values. In

Algorithm 1 μJADE

1: Initialise population (P)
2: μ_{CR} = 0.5
3: μ_F = 0.5
4: Initialize file (A) empty
5: **for** $g = 1$ to $numberofgeneration$ **do**
6: **for** $i = 1$ to NP **do**
7: $CR_i = randn_i(\mu_{CR}, 0.1)$
8: $F_i = randc_i(\mu_F, 0.1)$
9: Randomly select $x_a \neq x_i$
10: Randomly select $x_b \neq x_a \neq x_i$
11: Randomly select $x^p_{best} \neq x_a$ from pNP best population vectors
12: Randomly select x_c from $P \cup A$
13: Randomly select $j_{rand} \in \mathbb{N}^{+}_{\leqslant D}$
14: $v_i = x_i + F_i(x^p_{best} - x_a) + F_i(x_b - \tilde{x_c})$
15: $v_{i,j} = \begin{Bmatrix} \frac{L_j + x_{i,j}}{2} & \text{if} & v_{i,j} < L_j \\ \frac{U_j + x_{i,j}}{2} & \text{if} & v_{i,j} > U_j \\ v_{i,j} & & otherwise \end{Bmatrix}$
16: **for** $j = 1$ to D **do**
17: $d_{i,j} = rand(0, 1)$
18: $u_{i,j} = \begin{Bmatrix} v_{i,j} & \text{if} & OR(d_{i,j} < CR_i, j = j_{rand}) \\ x_{i,j} & & otherwise \end{Bmatrix}$
19: $b_{i,j} = \begin{Bmatrix} 1 & \text{if} & OR(d_{i,j} < CR_i, j = j_{rand}) \\ 0 & & otherwise \end{Bmatrix}$
20: **end for**
21: **for** $j = 1$ to D **do**
22: $r_{i,j} = rand(0, 1)$
23: $u_{i,j} = \begin{Bmatrix} L_j + rand(0,1)(U_j - L_j) & \text{if} & r_{i,j} \leqslant 0.005 \\ u_{i,j} & & otherwise \end{Bmatrix}$
24: $b_{i,j} = \begin{Bmatrix} 0 & \text{if} & r_{i,j} \leqslant 0.005 \\ b_{i,j} & & otherwise \end{Bmatrix}$
25: **end for**
26: $CR_i = \frac{\sum_{j=1}^{D} b_{i,j}}{D}$
27: **if** $f(u_i) < f(x_i)$ **then**
28: $x_i \rightarrow A$
29: $x_i = u_i$
30: $CR_i \rightarrow S_{CR}$
31: $F_i \rightarrow S_F$
32: **if** u_i is better than best population vector **then**
33: $BIR = BIR + 1$
34: **end if**
35: **end if**
36: **end for**
37: Randomly remove vectors from A so that $|A| \leqslant NP$
38: **if** $mod(g, max(100, 10D)) = 0$ **then**
39: $\mu_{CR} = (1 - c)\mu_{CR} + cL_1(S_{CR}), L_1(\varnothing) = 0$
40: $\mu_F = (1 - c)\mu_F + cL_2(S_F), L_2(\varnothing) = 0$
41: $S_{CR} = S_F = \varnothing$
42: **end if**
43: **if** $mod(g, max(1000, 100D)) = 0$ **then**
44: **if** BIR= 0 **then**
45: Reinitialize population excluding the best vector.
46: **end if**
47: **end if**
48: **end for**

contrast, when $\varepsilon = 0$, the two infeasible vectors are compared based on their constraint violations.

The ε level is controlled using Equation 7, where $\varepsilon(0)$ is initialized with the constraint violation of the θ-th vector in the population (x_θ). ε is updated as long as the number of iterations t does not exceed the value T_c, after that, ε is set to 0.

$$\begin{aligned} \varepsilon(0) &= \phi(x_\theta) \\ \varepsilon(t) &= \begin{Bmatrix} \varepsilon(0)(1 - \frac{t}{T_c})^{cp} & , & 0 < t < T_c, \\ 0 & & t \geqslant Tc \end{Bmatrix} \end{aligned} \tag{7}$$

C. *Adding the constraint-handling techniques to μJADE*

The addition of each one of the two constraint-handling techniques into μJADE is straightforward because both of them are just comparison criteria which replace the original μJADE criterion based only in the objective function value for unconstrained optimization, e.g., lines 27, 32, and 45 in Algorithm 1.

IV. Experiments and Results

Three experiments were carried out to evaluate the performance of μJADE with each constraint-handling technique. For the first two experiments, a well-known set of constrained optimization problems with 18 test problems with 10 and 30 dimensions was used [12] ; meanwhile for the third experiment, a well-known set of thirteen constrained problems was used [13]. In the first experiment, the two constraint-handling techniques aforementioned were compared to identify that with the best performance: μJADE (FR) is μJADE with the feasibility rules and μJADEε is μJADE with the ε-constrained method. In the second experiment, the most competitive algorithm of the first experiment was compared against two highly competitive EAs for constrained optimization. Those algorithms are the εDEag proposed by Takahama et al. in [11], and *DEwAPI* proposed by Elsayed et al. in [14]. Finally, in the third experiment, μJADEε was compared against μPSO to validate the performance in the world of small population size algorithms.

The 95%-confidence Wilcoxon Signed Rank Test was applied to the samples of results, where three symbols were used: "=", "+" and "-"; where "=" indicates that there is no significant difference between the two algorithms, "+" denotes that the first algorithm is significantly better than the second one, and "-" means that the first algorithm is significantly worse than the second algorithm.

A. *First experiment: μJADE (FR) vs. μJADEε*

The parameters used by μJADE using both constraint-handling techniques are summarized in Table I, in order to keep the same execution environment. The number of evaluations was 200000 for 10D test problems and 600000 for 30D test problems. Those parameters were obtained by preliminary experiments.

TABLE I
PARAMETER SETTING FOR μJADE (FR) AND μJADEε.

Parameter	Value
Population size	8
c	1.5
T_c	4000
cp	5

Tables II and III show the results found by each μJADE version for 10D and 30D. In both dimensionalities, μJADEε obtained better results than μJADE (FR) based on the Wilcoxon test. For 10D, μJADEε was significantly better in 11 test problems out of 18, in the remaining ones, there were no significant differences. In the same way, for 30D, μJADEε was significantly better in 10 test problems out of 18, while in the other seven, no significant differences were observed. C04 in 30D was omitted because both versions were unable to find feasible solutions.

Four measures were used to increase the empirical comparison to determine which constraint-handling technique performed better: Feasibility Probability (FP), Probability of convergence (P), Average number of Function Evaluations ($AFES$) and Successful Performance (SP) [15]. FP represents the number of feasible executions (i.e., those runs with at least one feasible solution found) divided by the total number of independent executions; FP goes from 0 to 1, where 1 indicates that each independent run reached the feasible region. P is calculated with the number of successful runs (i.e. those runs where the neighborhood of the best feasible solution was reached) divided by the total number of independent executions. To determine the closeness to the best feasible solution, a small tolerance, i.e., 1E-4, was used. P goes from 0 to 1, where 1 denotes that all independent executions were successful runs. $AFES$ is calculated by averaging the number of evaluations required by a successful run to reach the best known feasible value. For $AFES$, lower values are preferred. Finally, SP is a combination of $AFES$ and P in order to measure the speed and reliability of the algorithm It is calculated by dividing $AFES$ from P. As a result, lower values are preferred because they mean a good compromise between speed and consistency of the algorithm. It is important to mention that the best values used to calculate the successful runs in P, $AFES$, and SP were obtained from [11]. The results for those four measures are included in Tables II and III for 10D and 30D, respectively.

The results of those measures indicate that μJADEε outperformed μJADE (FR), because the first one presented better results in most test problems. For FP, both algorithms had similar results. However, only in two test problems μJADE (FR) reached higher values with respect to those of μJADEε, having a better chance of finding feasible solutions. Regarding other measures μJADEε provided superior results, being more capable to find a successful solution in less time, i.e., μJADEε provided better speed and reliability compared with μJADE (FR).

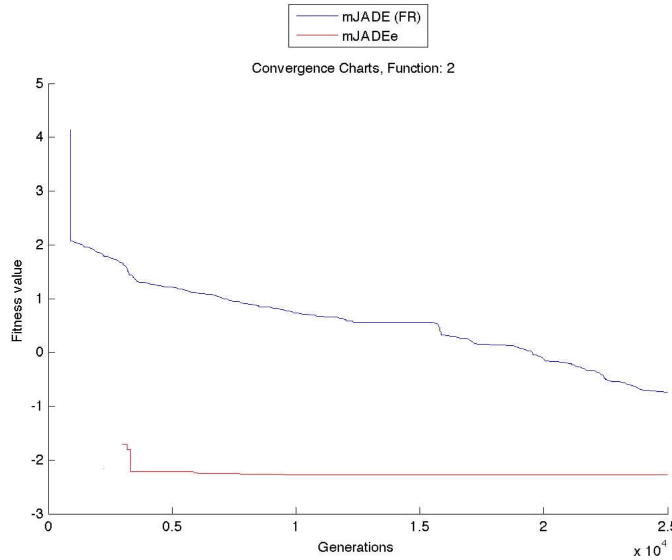

Fig. 1. Convergence plots of μJADE (FR) and μJADEε for 10D C02 test problem.

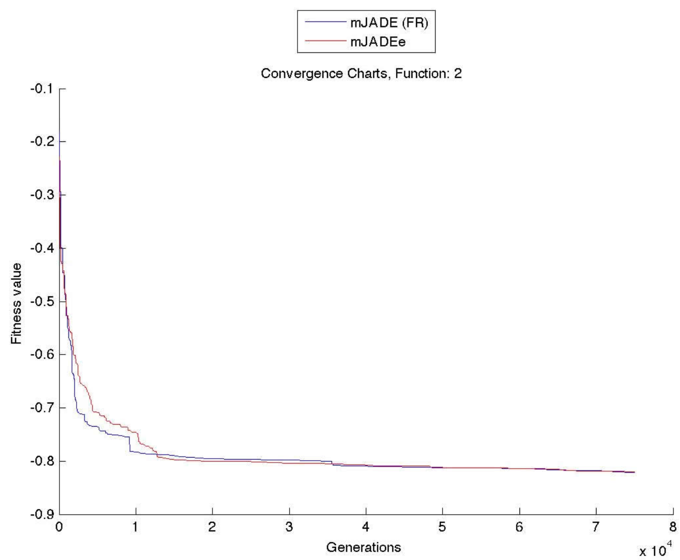

Fig. 2. Convergence plots of μJADE (FR) and μJADEε for 30D C01 test problem.

Finally, Figs. 1 and 2 include two representative convergence plots where μJADEε (red line) provided a faster approach to a good solution. Such behavior is easier to see in 10D than in 30D. For all the above mentioned results, μJADEε was chosen as the most competitive approach, which will be compared in the next experiment against highly competitive algorithms for constrained optimization.

B. Second experiment: μJADEε vs. conventional sized population algorithms

μJADEε was compared against two competitive conventional sized population algorithms: εDEag proposed by Takahama et al. in [11] and *DEwAPI* proposed by Elsayed et al.

TABLE II
10D RESULTS BY μJADE (FR) AND μJADEε. BOLD NUMBERS REPRESENT THE BEST VALUES FOUND.

Function	Algorithm	Best	Worst	Median	Mean	St. D.	FP	P	AFES	SP	Stats (μJADE (FR) vs μJADEε)
C01	μJADE (FR)	**-7.4731E-01**	-7.2588E-01	**-7.4731E-01**	-7.4331E-01	7.4244E-03	1	0.64	55390	86548	=
	μJADEε	**-7.4731E-01**	**-7.2921E-01**	**-7.4731E-01**	**-7.4397E-01**	**5.0826E-03**	1	0.64	54301	84845	
C02	μJADE (FR)	-7.4095E-01	3.5817E+00	1.8511E+00	1.6164E+00	1.2319E+00	1	0	-	-	-
	μJADEε	**-2.2768E+00**	**-1.2413E+00**	**-2.2071E+00**	**-2.1281E+00**	**2.2529E-01**	1	0	-	-	
C03	μJADE (FR)	1.3008E+07	1.2412E+15	1.2466E+14	2.4585E+14	3.1525E+14	1	0	-	-	-
	μJADEε	**0.0000E+00**	**8.8756E+00**	**4.8227E-24**	**3.5502E+00**	**4.4378E+00**	1	0.6	38359	63932	
C04	μJADE (FR)	**-1.0000E-05**	1.6189E+01	6.7844E-01	4.4765E+00	6.9197E+00	0.88	0.36	73600	2.04E+05	-
	μJADEε	**-1.0000E-05**	**9.8934E-01**	**-1.0000E-05**	**1.7528E-01**	**3.2354E-01**	1	0.72	93651	1.30E+05	
C05	μJADE (FR)	1.4418E+02	5.3614E+02	4.2672E+02	4.0444E+02	1.1559E+02	1	0	-	-	-
	μJADEε	**-4.8361E+02**	**-4.5982E+02**	**-4.8361E+02**	**-4.8060E+02**	**6.0478E+00**	1	0.64	85986	1.34E+05	
C06	μJADE (FR)	-5.5580E+01	5.6372E+02	4.0285E+02	3.8466E+02	1.5349E+02	1	0	-	-	-
	μJADEε	**-5.7866E+02**	**-5.7718E+02**	**-5.7811E+02**	**-5.7800E+02**	**6.2002E-01**	1	0.2	1.28E+05	6.39E+05	
C07	μJADE (FR)	**0.0000E+00**	3.9866E+00	**2.5180E-27**	1.5946E-01	7.9732E-01	1	0.96	54094	56348	=
	μJADEε	**0.0000E+00**	**4.9311E-26**	**2.5180E-27**	**7.6850E-27**	**1.1185E-26**	1	1	42510	42510	
C08	μJADE (FR)	**0.0000E+00**	**4.0876E+01**	3.9866E+00	8.2170E+00	**1.0210E+01**	1	0.48	50476	1.05E+05	=
	μJADEε	**0.0000E+00**	5.0845E+01	**1.5541E-26**	**6.0469E+00**	1.1916E+01	1	0.64	35539	55530	
C09	μJADE (FR)	1.1526E+12	3.1278E+13	8.1924E+12	9.0270E+12	6.1296E+12	1	0	-	-	-
	μJADEε	**0.0000E+00**	**9.0073E+07**	**7.7031E-26**	**3.6038E+06**	**1.8014E+07**	1	0.56	41056	73314	
C10	μJADE (FR)	1.6163E+12	1.8234E+13	6.7388E+12	7.8844E+12	4.6637E+12	1	0	-	-	-
	μJADEε	**0.0000E+00**	**4.8393E+03**	**4.1728E+01**	**2.6529E+02**	**9.6220E+02**	1	0.32	70161	2.19E+05	
C11	μJADE (FR)	**-1.5227E-03**	-1.3683E-04	**-1.5227E-03**	-1.4625E-03	2.8898E-04	0.92	0.96	38394	39993	=
	μJADEε	**-1.5227E-03**	**-1.5227E-03**	**-1.5227E-03**	**-1.5227E-03**	**3.6696E-09**	1	1	69860	69860	
C12	μJADE (FR)	-3.0549E+02	4.7980E+00	**-1.9925E-01**	-3.8502E+01	**7.6328E+01**	1	0.04	70016	1.75E+06	=
	μJADEε	**-4.2652E+02**	**-1.9924E-01**	**-1.9925E-01**	**-1.0413E+02**	1.6552E+02	0.96	0	-	-	
C13	μJADE (FR)	**-6.8429E+01**	-6.3517E+01	**-6.8429E+01**	**-6.7009E+01**	1.7203E+00	1	0.4	1.23E+05	3.07E+05	=
	μJADEε	**-6.8429E+01**	**-6.5578E+01**	-6.5578E+01	-6.6719E+01	**1.4253E+00**	1	0.32	1.27E+05	3.95E+05	
C14	μJADE (FR)	**0.0000E+00**	3.1303E+11	2.7719E+02	1.3609E+10	6.2554E+10	1	0.24	1.20E+05	4.99E+05	-
	μJADEε	**0.0000E+00**	**3.9866E+00**	**4.8878E-27**	**3.1893E-01**	**1.1038E+00**	1	0.92	46007	50008	
C15	μJADE (FR)	8.4408E+10	1.6020E+14	4.4344E+13	5.7217E+13	4.6972E+13	1	0	-	-	-
	μJADEε	**0.0000E+00**	**1.0607E+01**	**4.8878E-27**	**1.3706E+00**	**2.5952E+00**	1	0.68	43658	64203	
C16	μJADE (FR)	2.1900E-01	1.0394E+00	6.8271E-01	6.5823E-01	**2.5065E-01**	1	0	-	-	-
	μJADEε	**0.0000E+00**	**9.4810E-01**	**2.9752E-01**	**3.7532E-01**	3.3789E-01	1	0.16	55534	3.47E+05	
C17	μJADE (FR)	2.5030E+01	9.0098E+02	1.8648E+02	2.4508E+02	2.0868E+02	1	0	-	-	-
	μJADEε	**1.2326E-32**	**3.8959E+02**	**2.6384E-01**	**5.0345E+01**	**1.1189E+02**	1	0.2	1.19E+05	5.93E+05	
C18	μJADE (FR)	2.0309E+02	8.2869E+03	3.4694E+03	3.4103E+03	2.0938E+03	1	0	-	-	-
	μJADEε	**0.0000E+00**	**3.1554E-30**	**0.0000E+00**	**2.5244E-31**	**8.7370E-31**	1	1	26044	26044	

in [14]. The detailed results are shown in Table IV. Table V shows the statistical comparison using the Wilcoxon test, where μJADEε was particularly competitive against εDEag, mainly in 30D test problems. Regarding *DEwAPI*, μJADEε was competitive in some test problems, but was outperformed in others, mainly in 30D. As a conclusion of this second experiment, μJADEε was competitive, but still not clearly better, than conventional sized populations algorithms for constrained optimization.

C. Third experiment: μJADEε vs μPSO

Finally, μJADEε was compared against one μEA for constrained spaces: μPSO [7]. The detailed results are shown in Table VI, where μJADEε outperformed μPSO in ten out of thirteen functions, according to the statistical results. Moreover, the μJADEε results were close to the optimal known value. In conclusion, μJADEε showed to be highly competitive, compared with other μEA for constrained optimization.

TABLE V
STATISTICAL COMPARISON AMONG μJADEε, εDEG AND DEWAPI USING THE 95%-CONFIDENCE WILCOXON SIGNED RANK TEST. "NA" MEANS THAT THE PROPOSED ALGORITHM DID NOT FIND FEASIBLE SOLUTIONS.

Function	μJADEε vs.			
	10D		30D	
	εDEag	DEwAPI	εDEag	DEwAPI
C01	-	-	-	=
C02	-	-	+	-
C03	-	-	=	-
C04	+	-	NA	NA
C05	-	-	-	-
C06	-	-	-	-
C07	-	-	=	=
C08	=	=	=	=
C09	=	=	=	=
C10	=	=	=	=
C11	-	-	=	=
C12	-	+	-	-
C13	-	-	+	=
C14	=	=	=	=
C15	-	-	=	-
C16	=	-	=	=
C17	+	-	+	-
C18	+	=	+	-
Summary				
+	3	1	4	0
-	10	12	4	8
=	5	5	9	9

V. CONCLUSIONS

A highly competitive μEA for unconstrained optimization called μJADE was adapted to solve constrained optimization problems. Two constraint-handling techniques, the feasibility rules and the ε-constrained method, were added to μJADE and compared. The latter provided a better performance based on final statistical results and also based on four performance measures and convergence behavior. μJADEε was further compared against two conventionally sized population al-

TABLE III
30D RESULTS BY μJADE (FR) AND μJADEε. BOLD NUMBERS REPRESENT THE BEST VALUES FOUND. "NA" MEANS THAT THE PROPOSED ALGORITHM DID NOT FIND FEASIBLE SOLUTIONS.

Function	Algorithm	Best	Worst	Median	Mean	St. D.	FP	P	AFES	SP	Stats (μJADE (FR) vs μJADEε)
C01	**μJADE (FR)**	**-8.2122E-01**	**-7.9373E-01**	**-8.1427E-01**	**-8.1299E-01**	**6.7506E-03**	1	0	-	-	=
	μJADEε	-8.2083E-01	-7.8861E-01	-8.0987E-01	-8.0980E-01	8.6780E-03	1	0	-	-	
C02	μJADE (FR)	7.3664E-01	3.9577E+00	2.9239E+00	2.9973E+00	7.7112E-01	1	0	-	-	-
	μJADEε	**-2.1954E+00**	**-1.2243E+00**	**-1.8217E+00**	**-1.8123E+00**	**2.8390E-01**	1	0.04	4.29E+05	1.07E+07	
C03	μJADE (FR)	2.1771E+10	7.9054E+13	3.1189E+13	2.8109E+13	2.5969E+13	0.52	0	-	-	-
	μJADEε	**1.8437E-22**	**1.8321E+02**	**2.8673E+01**	**2.7987E+01**	**3.4643E+01**	1	0.64	2.35E+05	3.68E+05	
C04	μJADE (FR)	-	-	-	-	-	-	0	-	-	NA
	μJADEε	-	-	-	-	-	-	0	-	-	
C05	μJADE (FR)	2.4224E+02	5.5826E+02	4.8759E+02	4.7822E+02	**6.5806E+01**	1	0	-	-	-
	μJADEε	**-4.1540E+02**	**5.3688E+02**	**-2.3891E+02**	**-7.4365E+01**	3.5260E+02	1	0	-	-	
C06	μJADE (FR)	2.2947E+02	5.5473E+02	4.8014E+02	4.6503E+02	**7.9274E+01**	1	0	-	-	-
	μJADEε	**-5.2248E+02**	**-9.4134E-01**	**-4.7936E+02**	**-4.0917E+02**	1.4767E+02	1	0	-	-	
C07	μJADE (FR)	8.7536E-24	2.0070E+01	**1.0319E-19**	1.6777E+00	4.8985E+00	1	0.84	2.45E+05	2.91E+05	=
	μJADEε	**3.9874E-24**	**4.2653E+00**	8.3422E-18	**2.4312E-01**	**8.7094E-01**	1	0.8	2.68E+05	3.35E+05	
C08	μJADE (FR)	1.6567E-23	**8.1094E+02**	1.3873E-02	**5.3416E+01**	**1.6469E+02**	1	0.56	2.80E+05	4.99E+05	=
	μJADEε	**1.6493E-23**	8.1589E+02	**2.7339E-15**	6.2209E+01	1.7294E+02	1	0.6	3.02E+05	5.04E+05	
C09	μJADE (FR)	1.3525E+13	8.0384E+13	3.3298E+13	3.5741E+13	1.4720E+13	1	0	-	-	-
	μJADEε	**1.1528E+00**	**5.5671E+10**	**9.3421E+01**	**2.2268E+09**	**1.1134E+10**	1	0	-	-	
C10	μJADE (FR)	1.0306E+13	5.5780E+13	3.3853E+13	3.3608E+13	1.1527E+13	1	0	-	-	-
	μJADEε	**6.6681E+00**	**1.9900E+04**	**4.6657E+01**	**1.6146E+03**	**4.6110E+03**	1	0	-	-	
C11	μJADE (FR)	**-3.9226E-04**	**-2.0060E-04**	**-3.9048E-04**	**-3.7968E-04**	**4.3503E-05**	0.76	0.88	3.14E+05	3.57E+05	+
	μJADEε	-3.9117E-04	1.8907E-04	-3.8097E-04	-2.5862E-04	2.3879E-04	0.36	0.28	4.02E+05	1.43E+06	
C12	μJADE (FR)	**-1.9926E-01**	3.1282E+02	**-8.2157E-02**	2.1560E+01	6.9655E+01	0.96	0.4	2.10E+05	5.24E+05	+
	μJADEε	-1.9291E-01	**1.6772E+01**	5.5424E-01	**3.5009E+00**	**4.7389E+00**	0.72	0	-	-	
C13	μJADE (FR)	-6.6706E+01	**-6.2795E+01**	**-6.4908E+01**	**-6.4877E+01**	**1.1535E+00**	1	0	-	-	=
	μJADEε	**-6.6975E+01**	-6.1539E+01	-6.4693E+01	-6.4531E+01	1.6259E+00	1	0	-	-	
C14	μJADE (FR)	6.4662E-21	6.7450E+02	**4.5105E-11**	5.2739E+01	1.4456E+02	1	0.68	3.91E+05	5.74E+05	=
	μJADEε	**8.7215E-25**	**1.9508E+01**	9.1448E-07	**2.0635E+00**	**5.3917E+00**	1	0.56	2.57E+05	4.59E+05	
C15	μJADE (FR)	1.8480E+13	3.0384E+14	1.0210E+14	1.3167E+14	8.3267E+13	1	0	-	-	-
	μJADEε	**7.3754E-24**	**7.6821E+01**	**2.1604E+01**	**2.2673E+01**	**1.6213E+01**	1	0.04	5.24E+05	1.31E+07	
C16	μJADE (FR)	7.8035E-01	1.1456E+00	1.0504E+00	1.0497E+00	**8.6980E-02**	1	0	-	-	-
	μJADEε	**0.0000E+00**	**8.3683E-01**	**0.0000E+00**	**3.5339E-02**	1.6713E-01	1	0.84	16817	20020	
C17	μJADE (FR)	3.8417E+02	1.7673E+03	1.0668E+03	1.0196E+03	3.7742E+02	1	0	-	-	-
	μJADEε	**3.5857E-02**	**8.8086E+02**	**1.6151E+02**	**1.9562E+02**	**2.1804E+02**	1	0.12	4.37E+05	3.65E+06	
C18	μJADE (FR)	5.6738E+03	5.9899E+04	1.3873E+04	1.5886E+04	1.0906E+04	1	0	-	-	-
	μJADEε	**2.2070E-01**	**1.1974E+04**	**1.3645E+01**	**1.9528E+03**	**3.5428E+03**	1	0	-	-	

gorithms, where the results were competitive, but still not clearly better. Moreover, μJADEε provided better results when compared against one μEA for constrained optimization. The overall observed performance provides encouraging insights about the viability of successfully using small populations in DE to deal with constrained search spaces.

As future work, it is necessary to revisit μJADEε to analyze its behavior in those problems where local optima were found. The design of mechanisms to increase the exploration abilities with a small population will be studied.

ACKNOWLEDGMENTS

The first author acknowledges support from the Mexican National Council of Science and Technology (CONACyT) through a scholarship to pursue graduate studies at the University of Veracruz. The second author acknowledges support from CONACyT through project No. 220522.

REFERENCES

[1] P. Siarry and Z. Michalewicz, Eds., *Advances in Metaheuristics for Hard Optimization*, ser. Natural Computing. Springer, 2008.

[2] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.

[3] F. Caraffini, F. Neri, and I. Poikolainen, "Micro-differential evolution with extra moves along the axes," in *Differential Evolution (SDE), 2013 IEEE Symposium on*, April 2013, pp. 46–53.

[4] H. Salehinejad, S. Rahnamayan, H. R. Tizhoosh, and S. Y. Chen, "Micro-differential evolution with vectorized random mutation factor," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*, 2014, pp. 2055–2062. [Online]. Available: http://dx.doi.org/10.1109/CEC.2014.6900606

[5] C. Brown, Y. Jin, M. Leach, and M. Hodgson, "μ jade: adaptive differential evolution with a small population," *Soft Computing*, pp. 1–10, 2015.

[6] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 2, pp. 82–102, Jul 1999.

[7] J. C. F. Cabrera and C. A. C. Coello, "Handling constraints in particle swarm optimization using a small population size." in *MICAI*, 2007, pp. 41–51.

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

[9] I. Fajfar, T. Tuma, J. Puhan, J. Olenšek, and Á. Bűrmen, "Towards smaller populations in differential evolution," *Electronic Components and Materials*, vol. 42, no. 3, pp. 152–163, 2012.

[10] G. T. Pulido and C. A. C. Coello, "A constraint-handling mechanism for particle swarm optimization," in *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 2. Ieee, 2004, pp. 1396–1403.

[11] T. Takahama and S. Sakai, "Constrained optimization by the epsilon constrained differential evolution with an archive and gradient-based mutation." in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–9.

[12] R. Mallipeddi and P. Suganthan, "Problem definitions and evaluation criteria for the cec 2010 competition and special session on single objective constrained real-parameter optimization," *Nangyang Technological University, Singapore*, 2010.

[13] J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. C. Coello, and K. Deb, "Problem definitions and evaluation criteria

TABLE IV
COMPARISON AMONG μJADEε, εDEG AND DEWAPI. BOLD NUMBERS REPRESENT THE BEST VALUES FOUND.

Function	Alg	10D			30D		
		Best	Mean	St. D.	Best	Mean	St. D.
C01	μJADEε	**-7.4731E-01**	-7.4397E-01	5.0826E-03	-8.2083E-01	-8.0980E-01	8.6780E-03
	εDEag	**-7.4731E-01**	-7.4704E-01	1.3233E-03	-8.2183E-01	**-8.2087E-01**	**7.1039E-04**
	DEwAPI	**-7.4731E-01**	**-7.4731E-01**	**2.2662E-16**	**-8.2188E-01**	-8.1062E-01	1.1242E-02
C02	μJADEε	-2.2768E+00	-2.1281E+00	2.2529E-01	-2.1954E+00	-1.8123E+00	2.8390E-01
	εDEag	**-2.2777E+00**	-2.2589E+00	2.3898E-02	-2.1692E+00	-2.1514E+00	1.1976E-02
	DEwAPI	**-2.2777E+00**	**-2.2776E+00**	**6.9912E-04**	**-2.2810E+00**	**-2.2751E+00**	**3.9470E-03**
C03	μJADEε	**0.0000E+00**	3.5502E+00	4.4378E+00	1.8437E-22	2.7987E+01	3.4643E+01
	εDEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.8673E+01	2.8838E+01	8.0472E-01
	DEwAPI	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**1.2770E-24**	**1.2199E-23**	**1.1757E-23**
C04	μJADEε	**-1.0000E-05**	1.7528E-01	3.2354E-01	-	-	-
	εDEag	-9.9923E-06	-9.9185E-06	1.5467E-07	4.6981E-03	8.1630E-03	3.0678E-03
	DEwAPI	**-1.0000E-05**	**-1.0000E-05**	**0.0000E+00**	**-3.3247E-06**	**-3.2497E-06**	**8.6837E-08**
C05	μJADEε	**-4.8361E+02**	-4.8060E+02	6.0478E+00	-4.1540E+02	-7.4365E+01	3.5260E+02
	εDEag	**-4.8361E+02**	**-4.8361E+02**	3.8904E-13	-4.5313E+02	-4.4955E+02	2.8991E+00
	DEwAPI	**-4.8361E+02**	**-4.8361E+02**	**1.1603E-13**	**-4.8361E+02**	**-4.8361E+02**	**2.6559E-03**
C06	μJADEε	**-5.7866E+02**	-5.7800E+02	6.2002E-01	-5.2248E+02	-4.0917E+02	1.4767E+02
	εDEag	**-5.7866E+02**	-5.7865E+02	3.6272E-03	-5.2858E+02	-5.2791E+02	**4.7484E-01**
	DEwAPI	**-5.7866E+02**	**-5.7866E+02**	**1.8415E-08**	**-5.3064E+02**	**-5.3033E+02**	1.1786E+00
C07	μJADEε	**0.0000E+00**	7.6850E-27	1.1185E-26	3.9874E-24	2.4312E-01	8.7094E-01
	εDEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	1.1471E-15	2.6036E-15	1.2334E-15
	DEwAPI	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**1.6140E-26**	**1.1662E-25**	**7.1627E-26**
C08	μJADEε	**0.0000E+00**	6.0469E+00	1.1916E+01	1.6493E-23	6.2209E+01	1.7294E+02
	εDEag	**0.0000E+00**	6.7275E+00	5.5606E+00	2.5187E-14	**7.8315E-14**	**4.8552E-14**
	DEwAPI	**0.0000E+00**	**1.8075E+00**	**3.1361E+00**	**8.9686E-26**	5.3824E+00	2.4456E+01
C09	μJADEε	**0.0000E+00**	3.6038E+06	1.8014E+07	1.1528E+00	2.2268E+09	1.1134E+10
	εDEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	2.7707E-16	1.0721E+01	2.8219E+01
	DEwAPI	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**1.6343E-23**	**7.8733E-23**
C10	μJADEε	**0.0000E+00**	2.6529E+02	9.6220E+02	6.6681E+00	1.6146E+03	4.6110E+03
	εDEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	3.2520E+01	3.3262E+01	4.5456E-01
	DEwAPI	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**5.7761E-26**	**2.0463E-24**	**5.3676E-24**
C11	μJADEε	**-1.5227E-03**	**-1.5227E-03**	3.6696E-09	-3.9117E-04	-2.5862E-04	2.3879E-04
	εDEag	**-1.5227E-03**	**-1.5227E-03**	**6.3410E-11**	-3.2685E-04	-2.8639E-04	2.7076E-05
	DEwAPI	**-1.5227E-03**	**-1.5227E-03**	5.9927E-10	**-3.9234E-04**	**-3.9234E-04**	**2.7736E-09**
C12	μJADEε	-4.2652E+02	-1.0413E+02	1.6552E+02	-1.9291E-01	3.5009E+00	4.7389E+00
	εDEag	**-5.7009E+02**	**-3.3673E+02**	1.7822E+02	-1.9915E-01	3.5623E+02	2.8893E+02
	DEwAPI	-3.0549E+02	-2.4554E+01	**8.4295E+01**	**-1.9926E-01**	**-1.9926E-01**	**1.7433E-06**
C13	μJADEε	**-6.8429E+01**	-6.6719E+01	1.4253E+00	**-6.6975E+01**	-6.4531E+01	1.6259E+00
	εDEag	**-6.8429E+01**	**-6.8429E+01**	1.0260E-06	-6.6425E+01	**-6.5353E+01**	**5.7330E-01**
	DEwAPI	**-6.8429E+01**	**-6.8429E+01**	**5.5258E-09**	-6.6362E+01	-6.4343E+01	1.1404E+00
C14	μJADEε	**0.0000E+00**	3.1893E-01	1.1038E+00	8.7215E-25	2.0635E+00	5.3917E+00
	εDEag	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	5.0159E-14	**3.0894E-13**	**5.6084E-13**
	DEwAPI	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**1.5033E-26**	1.5947E-01	7.9732E-01
C15	μJADEε	**0.0000E+00**	1.3706E+00	2.5952E+00	7.3754E-24	2.2673E+01	1.6213E+01
	εDEag	**0.0000E+00**	1.7990E-01	8.8132E-01	2.1603E+01	2.1604E+01	**1.1048E-04**
	DEwAPI	**0.0000E+00**	**1.6315E-15**	**8.1577E-15**	**7.4342E-27**	**6.7441E-01**	1.5771E+00
C16	μJADEε	**0.0000E+00**	3.7532E-01	3.3789E-01	**0.0000E+00**	3.5339E-02	1.6713E-01
	εDEag	**0.0000E+00**	3.7021E-01	3.7105E-01	**0.0000E+00**	2.1684E-21	1.0623E-20
	DEwAPI	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
C17	μJADEε	1.2326E-32	5.0345E+01	1.1189E+02	3.5857E-02	1.9562E+02	2.1804E+02
	εDEag	1.4632E-17	1.2496E-01	1.9372E-01	2.1657E-01	6.3265E+00	4.9867E+00
	DEwAPI	0.0000E+00	**3.4513E-33**	**3.1223E-33**	**1.4385E-13**	**1.4861E-02**	**1.3969E-02**
C18	μJADEε	**0.0000E+00**	2.5244E-31	8.7370E-31	2.2070E-01	1.9528E+03	3.5428E+03
	εDEag	3.7314E-20	9.6788E-19	1.8112E-18	1.2261E+00	8.7546E+01	1.6648E+02
	DEwAPI	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**9.4958E-12**	**3.6518E-01**	**1.7345E+00**

for the cec 2006 special session on constrained real-parameter optimization," *Journal of Applied Mechanics*, vol. 41, p. 8, 2006.

[14] S. M. Elsayed, R. Sarker *et al.*, "Differential evolution with automatic population injection scheme for constrained problems," in *Differential Evolution (SDE), 2013 IEEE Symposium on*. IEEE, 2013, pp. 112–118.

[15] E. Mezura-Montes, M. E. Miranda-Varela, and R. del Carmen Gómez-Ramón, "Differential evolution in constrained numerical optimization: an empirical study," *Information Sciences*, vol. 180, no. 22, pp. 4223–4262, 2010.

TABLE VI
STATISTICAL COMPARISON BETWEEN μJADEε, AND μPSO USING THE 95%-CONFIDENCE WILCOXON SIGNED RANK TEST ON 13 TEST PROBLEMS.

Function	**Optimal**	**Algorithm**	**Best**	**Worst**	**Median**	**Mean**	**St. D.**	**Stats (μJADEε vs μPSO)**
g01	**-15**	μJADEε	**-15**	**-15**	**-15**	**-15**	**2.54E-16**	+
		μPSO	**-15**	-13.273	-13	-9.7012	1.41	
g02	**-0.803619104**	μJADEε	-0.8036	-0.7535	**-0.7926**	**-0.79298**	**0.011143**	-
		μPSO	**-0.80362**	**-0.77714**	-0.77848	-0.7116	0.0191	
g03	**-1.0005001**	μJADEε	**-1.0005**	**-1.0003**	**-1.0005**	**-1.0005**	**3.61E-05**	+
		μPSO	-1.0004	-0.9936	-1.0004	-0.6674	0.0471	
g04	**-30665.53867**	μJADEε	**-30666**	**-30666**	**-30666**	**-30666**	**1.23E-11**	+
		μPSO	**-30666**	**-30666**	**-30666**	**-30666**	0.000683	
g05	**5126.496714**	μJADEε	**5126.5**	**5126.5**	**5126.5**	**5126.5**	**4.20E-12**	+
		μPSO	5126.6	5495.2	5261.8	6272.7	405	
g06	**-6961.813876**	μJADEε	**-6961.8**	**-6961.8**	**-6961.8**	**-6961.8**	**4.59E-12**	+
		μPSO	**-6961.8**	**-6961.8**	**-6961.8**	**-6961.8**	0.000261	
g07	**24.30620907**	μJADEε	**24.307**	**24.557**	**24.328**	**24.354**	**0.05834**	+
		μPSO	24.328	24.7	24.645	25.296	0.252	
g08	**-0.095825042**	μJADEε	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	8.41E-17	-
		μPSO	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**0**	
g09	**680.6300574**	μJADEε	**680.63**	**680.63**	**680.63**	**680.63**	**0.00057886**	+
		μPSO	**680.63**	680.64	680.64	680.67	0.00668	
g10	**7049.248021**	μJADEε	**7049.3**	**7251**	**7060.7**	**7100.9**	**70.704**	+
		μPSO	7090.5	7747.6	7557.4	10534	552	
g11	**0.7499**	μJADEε	**0.7499**	**0.7499**	**0.7499**	**0.7499**	**3.36E-16**	+
		μPSO	**0.7499**	0.7673	**0.7499**	0.9925	0.06	
g12	**-1**	μJADEε	**-1**	**-1**	**-1**	**-1**	**0**	=
		μPSO	**-1**	**-1**	**-1**	**-1**	**0**	
g13	**0.053941514**	μJADEε	**0.053942**	1	**0.053942**	**0.10365**	**0.16684**	+
		μPSO	0.05941	**0.81335**	0.90953	2.4442	0.381	