Inteligencia Artificial 16(52) (2013), 3-15

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Hybridization of Differential Evolution using Hill Climbing to solve Constrained Optimization Problems

Sebastián Hernández[1], Guillermo Leguizamón[2] and Efrén Mezura-Montes[3].

[1]Dpto. de Ciencias Exactas y Naturales, Unidad Académica Río Gallegos, Universidad Nacional de la Patagonia Austral, Argentina.
sebastian.unpa@gmail.com

[2]Lab. de Investigación y Desarrollo en Inteligencia Computacional, Universidad Nacional de San Luis, Argentina.
legui@unsl.edu.ar

[3]Dpto. de Inteligencia Artificial, Universidad Veracruzana, Xalapa, Veracruz, México.
emezura@uv.mx

Abstract Differential Evolution (DE) is a powerful metaheuristic method used to numerically optimize functions or multidimensional problems not solved by traditional methods of global optimization. In turn, if boundary conditions are added, constraint handling techniques should be need. To improve DE's performance, algorithms of local search are a good alternative. Hybridization of Differential Evolution and Hill Climbing are presented in this paper. The obtained results show similar or superior quality to those achieved by methods already tested.

Keywords: Differential evolution, Hill Climbing, Hybridization, Constrained Optimization.

1 Introduction

Optimization is an area that decades ago began to grow in parallel with the rise of computer processing. Constantly under development, new algorithms and techniques are proposed, according to the progress of disciplines such as biology, chemistry or physics, as they require optimizing problems with increasingly realistic models and therefore more complex.

Differential Evolution (DE) [5] is one of the most well-known metaheuristic techniques due its ease implementation and the quality of its results. This global optimization technique explores the space, according to internal parameters and regardless of the size of the region. Similarly, local search algorithms —Hill Climbing (HC) [4] is one of them— seek optimal points on small regions of randomly selected points through simple operations.

Hybridization of DE and HC is done here using the best DE individuals to apply HC. A non classical HC implementation that operates on more than one dimension at a time, by a random vector, is given. The obtained results are compared with the winner algorithm of the CEC 2010 special session: εDEag [6].

Organization of paper is as follows: (2) Differential Evolution, (3) Hill Climbing, (4) the proposed hybrid algorithm DE+HC, (5) Experiments, (6) Conclusions and no numbered section, Acknowledgements.

ISSN: 1988-3064(on-line)

2 Differential Evolution

DE is a population-based metaheuristic technique of numerical vectors. DE process is characterized by an iterating population vector evolving to candidate solutions comparing with an aptitude function called fitness. Four main steps can be identified in this process: Initialization, Mutation, Crossover, and Selection.

- **Initialization**. Before population vectors are initialized, upper and lower limits for each of the variables involved are selected, b_U and b_L respectively. After that, and by using random number generator, one numeric value between b_U and b_L is assigned for each component of the vector. For example, the initial value, $g = 0$, of j^{th} component in the i^{th} vector is

$$x_{j,i,0} = \text{rand}_j\,[0,1) \cdot (b_{j,U} - b_{j,L}) + b_{j,L}. \tag{1}$$

  The random number generator is executed once for each value of the component to ensure a completely random population. Each component is a real number since DE uses by definition a floating point representation. Each vector is identified by a number from 0 to N_{p-1}.

- **Mutation**. Like other population-based methods, new points in DE are generated as perturbations of the existing points. One by one, every point of the population is perturbed by adding a scaled difference between two different points randomly selected of the same population, $\mathbf{x}_{r1}$ and $\mathbf{x}_{r2}$. For each vector $\mathbf{x}_i$, a new mutation or noise vector, $\mathbf{v}_i$, is generated through the following equation:

$$\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}), \tag{2}$$

  where F is the scale value used, normally selected between 0 and 1. This process is called *differential mutation.*

- **Crossover**. In order to add genetic diversity to the population, another evolutionary component is used, the so-called *uniform crossover* or just crossover. Test vectors are generated through a crossover process, by using information that is copied in two different vectors. In particular, in DE each vector (called them, *trial*) is crossed with a mutated vector depending on the crossing probability of the population (Cr). Cr is a user-defined value, used to control the selection of the values that will be copied in the child vector, $\mathbf{u}_i$, from the respective mutation vector:

$$\mathbf{u}_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} \text{ if } \text{rand}_j\,(0,1) \leq C_r \\ x_{j,i,g} \text{ otherwise.} \end{cases} \tag{3}$$

- **Selection**. The child vector $\mathbf{u}_i$, generated after crossover, will replace the original vector in the population only if its fitness is better than the previous one. Otherwise, the parent vector or target $\mathbf{x}$ will remain in population by at least one generation. In this case, to obtain a zero (or a minimum in absolute value), the selection function is:

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} \text{ if } f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} \text{ otherwise.} \end{cases} \tag{4}$$

The above scheme corresponds to the classic DE version, called *DE/rand/1/bin* because the mutation base vector is randomly (*rand*) chosen, and only one (1) subtraction vector is done which is then scaled to add it to the base vector. A closely binomial distribution (*bin*) is exhibited by the number of donated parameters by the mutant vector.

2.1 Constraints

DE algorithm was designed to optimize numerical problems and its main feature is that the quality for any potential solution, called fitness, can be computed. However, most of DE applications are based on real models and certain variables restrictions exist that are not expressed in the fitness function. This means that, in addition to achieve optimal fitness, variables must be in the feasible region of the problem.

Direct application of DE is prevented by feasibility conditions because it was not originally designed to operate so. However, constraint handling techniques are used in DE to manage them in such complex problems [3].

The aim of optimization is to find the optimum $\mathbf{x}$ which minimizes the objective function satisfying simultaneously certain constraints. Two types of functions, g and h represent the constraints on the decision variables:

- **Regions**. They are the constraints of type $g_i(x) \leq 0$, where is important to measure how far an individual is from the feasible region. If a positive value is obtained, this can be used as a violation measure of the respective constraint.

- **Borders**. These are equality constraints, i.e., of the type $h_i(x) = 0$. They can be transformed in region constraints, by choosing a small ϵ value and rewriting them as $|h_i(x)| - \epsilon \leq 0$. It can also be used as boundary restriction according to management method used.

Static Penalty is one of the most simple and direct technique for constraint handling [1], [3]. A feasible minimum must satisfy all requested restrictions, as well as minimize the objective value. The static penalty function is external to the fitness, making violation of the constraints on a value that is added to that obtained in fitness. Thus, a pseudo-objective function, $\phi(\mathbf{x})$, is created:

$$\phi(\mathbf{x}) = f(\mathbf{x}) + P(\mathbf{x}). \tag{5}$$

This paper employs this constraint handling technique where the adopted penalty function $P(\mathbf{x})$ is as follows:

$$P(\mathbf{x}) = \sum_{i=1}^{n} C_p \times \max\{0, g_i(\mathbf{x})\}, \tag{6}$$

where C_p is a constant chosen by the user. Keep in mind that if C_p is small, the weight of $P(\mathbf{x})$ will be little in $\phi(\mathbf{x})$ but allow a broad exploration of the solution space and slow convergence. Otherwise, if a large C_p value is considered, premature convergence can be achieved by punishing excessively to infeasible solutions.

3 Hill Climbing

Hill Climbing (HC) method is a loop that continually moves in the direction of increasing value of a function [4]. The process begins by randomly selecting an individual from the population. This individual, called *father* is mutated, generating a *child*. Between these two individuals, the one better fits is chosen, *i.e.*, the highest value of fitness remains in the population. After each selection, the age of the individual selected is checked. If a parent is selected twice, his age is increased. Each time a new individual is selected, its age is inherited from his father. Mutation is done by adding a scalar to a component of the individual. If restrictions in the search space are involved, care must be taken to not to exceed the limits. In an n-dimensional space $2n$ mutations are applied, one for each possible direction in space. The magnitude of mutations decreases with age. This ensures the adaptation of the grain to the local structure of search space [7].

HC is an algorithm of simple implementation and very useful in one-dimensional space, but in higher dimensional spaces some implementation problems occur. Problems, like in the real case of climbing mountains, are shown:

- **Local maxima**. Like is a local search algorithm, not exhaustive, does not ensure that converge to the best value possible. Only if the search is started in a region near the global maximum, it can occur.

- **Plateaus**. The algorithm can explore a flat surface locally (in the same sense of fitness values). Improvements with respect to the starting point will not happen. Both, father and son, have the same fitness value.

- **Peaks**. Just only one direction of all, improves fitness. Smaller values in whatever route and any direction, are obtained. Only exploring in one particular address, better fitness is obtained.

4 Hybrid algorithm DE+HC

To achieve the hybrid version of HC with DE, some variants were included to the previously described HC loop. The changes are the following:

- **Parent**. The individual selected to mutate (parent) is not chosen randomly, but is the best individual in the population of ED. If the fitness of the child is lower than his father[1], the child will remain in the population, replacing the father.
- **Age**. Information about the age of the selected individual is not preserved.
- **Variables to mutate**. Exploration by mutation over all the variables of the problem is not done. A random vector named $\mathbf{V}$, with values from 1 to D (dimension of the problem to be optimized) is generated. The first component of $\mathbf{V}$, $v1$, will be the quantity of variables to mutate. These mutant variables will be selected taking care of $v1$-elements of $\mathbf{V}$. For example, if trying to optimize a function of dimension 10, a possible random vector is:

$$\mathbf{V} = [5, 4, 10, 1, 8, 6, 9, 3, 2, 7],$$

  from here, $v1 = 5$, so five variables will be mutated. Indexes of the mutant variables are the first 5 elements of $\mathbf{V}$. Therefore, x_5, x_4, x_{10}, x_1, and x_8 will mutate. But x_6, x_9, x_3, x_2, and x_7 will not be mutated. If vector $\mathbf{V}$ were:

$$\mathbf{V} = [6, 3, 7, 8, 5, 1, 2, 4, 9, 10],$$

  as $v1 = 6$, variables corresponding to the first six positions are mutated x_6, x_3, x_7, x_8, x_5 and x_1. Other variables (x_2, x_4, x_9 and x_{10}) are not involved in local exploration.
- **Mutation**. Selected variables are mutated according to the following formula:

$$H_j = x_j + (-1)^{v_j} \cdot k \cdot inc, \tag{7}$$

  where H_j represents the son, x_j is the father, k is an integer scaling factor ($k_0 = 1$) for the mutation and inc is the real value (rand/100) with which the variable will be mutated. This process is executed a specified number of times (parameter called hc). If after the mutation, fitness of H has no more difference than 0.0001 in absolute value (modifiable parameter of the algorithm) than the $\mathbf{x}$ fitness, the scale factor value k is doubled. This process is repeated as many times as necessary. If the selected individual is in a supposed plateau, *i.e.*, its fitness values do not change significantly, a maximum of $k = 128$ is established.

5 Experiments

The test suite of functions, defined in the *Single Objective Constrained Real - Parameter Optimization* of CEC2010 [2], was taken to evaluate the performance of DE+HC respect to DE. The solutions obtained by the winners of contest, Takahama and Sakai [6] were used to compare the obtained results.

Runtime parameters of DE and DE+HC are:

$$DE + HC(f, popsize, t_{max}, Cr, F, dim, hc),$$

where:

- f is the number of function to be optimized, varying between 1 and 18;
- *popsize* is the size of the population;
- t_{max} is the evolution time. The number of function evaluations (*FEs*) does not exceed 6×10^5 for 30D or 2×10^5 for 10D (no DE, nor DE+HC) to allow comparison with the results of Takahama and Sakai;

[1] minimization problem

Table 1: Parameters for 10D and 30D

	popsize	t_{max}	*hc*
ED - 10D	44	4540	0
ED+HC - 10D	40	4540	4
ED - 30D	85	7000	0
ED+HC - 30D	55	7000	30

- C_r is the probability of crossover and F is the scale factor of DE. 0.6 was used on both to ensure balance between exploration and exploitation;
- dim indicates the dimension of the used functions; and
- hc indicates presence or absence of Hill Climbing.

Boundary restrictions were transformed into region restrictions, to use a single coefficient penalty, $C_p = 150$. ϵ was 0.0001. Other parameters are specified in Table 1.

Tables 2 to 7 show results obtained in experiments. For each function, with algorithms, DE, DE+HD and ϵDEag, there were 25 executions. Row *Best* shows the fitness of the best individual possible; *Median*, *Worst*, *Average*, and *Deviation* are the statistics of the final population; *Violations* shows the amount of violations with regard to restrictions on the individual whose fitness is the median and $\overline{v}$ is the average violation committed on the solution median and is calculated as follows:

$$\overline{v} = \frac{\sum_{i=1}^{p} G_i(X) + \sum_{j=p+1}^{m} H_j(X)}{m},$$

where

$$G_i(X) = \begin{cases} g_i(X), & \text{if } g_i(X) > 0 \\ 0, & \text{if } g_i(X) \leqslant 0 \end{cases}$$

and

$$H_j(X) = \begin{cases} |h_j(X)|, & \text{if } |h_j(X)| - \varepsilon > 0 \\ 0, & \text{if } |h_j(X)| - \varepsilon \leqslant 0. \end{cases}$$

The results obtained by this proposal are promising. DE+HC achieved superior values compared to those obtained by Takahama and Sakai, showing the potency of the proposed hybrid algorithm. Tables 2 to 7 show in boldface the minimum values obtained on the metrics *Best* and *Median*. In this case, which obtains the best performance is DE+HC, followed closely by DE and ϵDEag, suggesting that the three algorithms implemented obtain in general good results. In a direct comparison of DE and DE+HC, it can be noticed a clear improvement achieved by the hybrid approach because, despite good results are achieved using DE, with DE+HC, the reached fitness values are even lower than those obtained with the original version of the algorithm of Storn and Price [5].

Figures 1 and 2 show the evolution of best individual of DE and DE+HC through 7000 iterations of functions $F01$, $F02$, $F03$, and $F04$ for 30D. Of the rest of the functions the respective plots are omitted because are similar to any of those already shown. The scales are logarithmic in both axes. From these plots we conclude that the convergence is similar to the shown algorithms. No premature convergence observed but evolution curve continues a steady decline, except near t_{max} where values tend to stabilize.

6 Conclusions

We presented a hybrid version of DE and HC (DE+HC) through an alternative implementation of the classic HC. Hybridization with HC was achieved by applying the best individuals obtained by ED. This algorithm was implemented to optimize the suite chosen for the contest of the special session of optimization of CEC2010 and results were compared with those obtained by winners of the contest: Takahama and Sakai. DE+HC shows results competitive with those reported by Takahama and Sakai for the same test suite.

For future work will consider to increase the problem dimensions in order to study scalability while implementing other local search techniques.

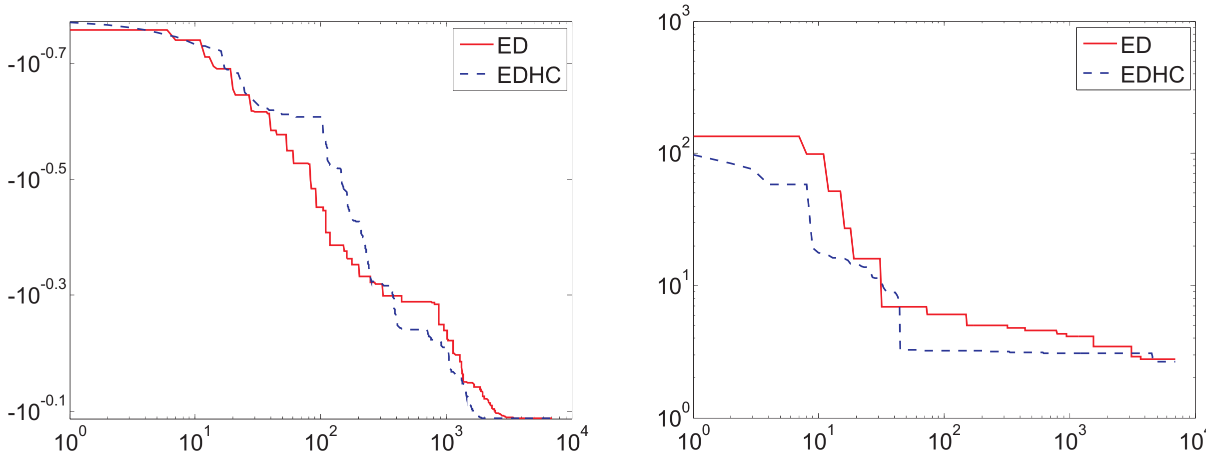

Figure 1: Evolution of the best individual in the population for F01 and F02 (30D) respectively.

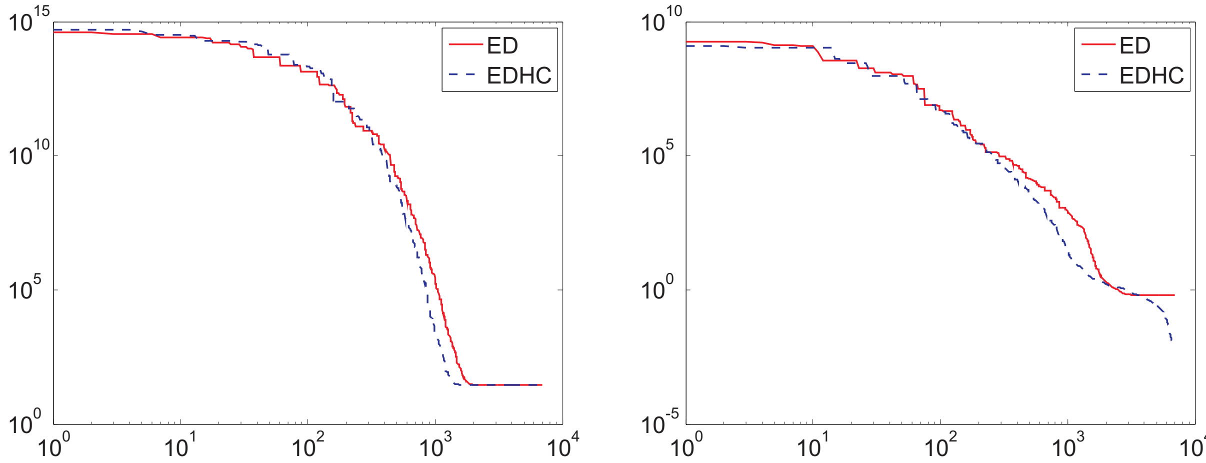

Figure 2: Evolution of the best individual in the population for F03 and F04 (30D) respectively.

Acknowledgements

The second author acknowledges support MinCyT through the bilateral project Mexico-Argentina MX1103 and the UNSL project P38403. The third author acknowledges support from CONACyT through the Mexico-Argentina bilateral project 164626 and project No. 79809.

References

[1] Carlos A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art, 2002.

[2] Rammohan Mallipeddi and Ponnuthurai Nagaratnam Suganthan. Problem definitions and evaluation criteria for the cec 2010 competition on constrained real-parameter optimization. Technical report, Nanyang Technological University, Singapore, 2010.

[3] Efrén Mezura-Montes and Carlos A. Coello Coello. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation*, 1(4):173–194, 2011.

[4] Stuart Rusell and Peter Norvig. *Inteligencia Artificial (2da Edición) - Un Enfoque Moderno*. Pearson - Prentice Hall, 2004.

[5] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[6] Tetsuyuki Takahama and Setsuko Sakai. Constrained optimization by the ϵconstrained differential evolution with an archive and gradient-based mutation. *IEEE World Congress on Computational Intelligence*, I, 2010.

[7] Deniz Yuret and Michael de la Maza. Dynamic hill climbing: Overcoming the limitations of optimization techniques. Technical report, Numinous Noetics Group, Artificial Intelligence Laboratory, MIT, 1993.

Table 2: Results for 10D: F01 - F06 CEC2010

F01 - 10D			
	DE	DE+HC	εDEag
Best	**-7,473104E-01**	**-7,473104E-01**	**-7,473104E-01**
Median	**-7,473104E-01**	-7,405572E-01	**-7,473104E-01**
Worst	-7,258790E-01	-7,130017E-01	-7,405572E-01
Mean	-7,441479E-01	-7,383533E-01	-7,470402E-01
Desv	6,029407E-03	9,295751E-03	1,323339E-03
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F02 - 10D			
	DE	DE+HC	εDEag
Best	-2,271573E+00	**-2,277707E+00**	-2,277702E+00
Median	**-2,271573E+00**	**-2,271573E+00**	-2,269502E+00
Worst	-2,259051E+00	-2,271573E+00	-2,174499E+00
Mean	-2,270571E+00	-2,271818E+00	-2,258870E+00
Desv	3,467045E-03	1,226804E-03	2,389779E-02
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F03 - 10D			
	DE	DE+HC	εDEag
Best	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Median	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Worst	8,748466E+00	0,000000E+00	0,000000E+00
Mean	3,499386E-01	0,000000E+00	0,000000E+00
Desv	1,749693E+00	0,000000E+00	0,000000E+00
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F04 - 10D			
	DE	DE+HC	εDEag
Best	**-9,999999E-06**	**-9,999999E-06**	-9,992345E-06
Median	**-9,999999E-06**	**-9,999999E-06**	-9,977276E-06
Worst	9,853501E-01	-9,999999E-06	-9,282295E-06
Mean	3,951567E-02	-9,999999E-06	-9,918452E-06
Desv	1,970496E-01	0,000000E+00	1,546730E-07
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F05 - 10D			
	DE	DE+HC	εDEag
Best	**-4,836106E+02**	**-4,836106E+02**	**-4,836106E+02**
Median	-4,830219E+02	**-4,836106E+02**	**-4,836106E+02**
Worst	-4,662568E+02	-4,836016E+02	-4,836106E+02
Mean	-4,805688E+02	-4,836103E+02	-4,836106E+02
Desv	5,282227E+00	1,805706E-03	3,890350E-13
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F06 - 10D			
	DE	DE+HC	εDEag
Best	**-5,786625E+02**	**-5,786625E+02**	-5,786581E+02
Median	-5,786618E+02	**-5,786625E+02**	-5,786533E+02
Worst	-5,785854E+02	-5,786624E+02	-5,786448E+02
Mean	-5,786526E+02	-5,786625E+02	-5,786528E+02
Desv	2,055751E-02	2,065582E-05	3,627169E-03
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

Table 3: Results for 10D: F07 - F12 CEC2010

F07 - 10D			
	DE	DE+HC	εDEag
Best	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Median	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Worst	0,000000E+00	0,000000E+00	0,000000E+00
Mean	0,000000E+00	0,000000E+00	0,000000E+00
Desv	0,000000E+00	0,000000E+00	0,000000E+00
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F08 - 10D			
	DE	DE+HC	εDEag
Best	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Median	**1,094154E+01**	**1,094154E+01**	**1,094154E+01**
Worst	1,094155E+01	1,094154E+01	1,537535E+01
Mean	6,782825E+00	8,809662E+00	6,727528E+00
Desv	5,142431E+00	4,181430E+00	5,560648E+00
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F09 - 10D			
	DE	DE+HC	εDEag
Best	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Median	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Worst	1,464080E+02	0,000000E+00	0,000000E+00
Mean	9,089229E+00	0,000000E+00	0,000000E+00
Desv	3,014485E+01	0,000000E+00	0,000000E+00
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F10 - 10D			
	DE	DE+HC	εDEag
Best	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Median	2,807825E+01	2,807825E+01	**0,000000E+00**
Worst	1,398130E+02	2,807825E+01	0,000000E+00
Mean	3,845815E+01	2,172444E+01	0,000000E+00
Desv	3,364877E+01	1,159544E+01	0,000000E+00
Viol	1	1	0
v	1,667730E+01	1,667730E+01	0,000000E+00

F11 - 10D			
	DE	DE+HC	εDEag
Best	**-1,522713E-03**	**-1,522713E-03**	**-1,522713E-03**
Median	**-1,522713E-03**	**-1,522713E-03**	**-1,522713E-03**
Worst	4,873684E+01	-1,522713E-03	-1,522713E-03
Mean	1,948012E+00	-1,522713E-03	-1,522713E-03
Desv	9,747672E+00	2,764045E-14	6,341035E-11
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F12 - 10D			
	DE	DE+HC	εDEag
Best	**-5,701311E+02**	**-5,701311E+02**	-5,700899E+02
Median	-2,099679E-01	-2,099679E-01	**-4,231332E+02**
Worst	-2,099679E-01	-2,099679E-01	-1,989129E-01
Mean	-9,124999E+01	-9,300149E+01	-3,367349E+02
Desv	1,604507E+02	1,496788E+02	1,782166E+02
Viol	1	1	0
v	1,776750E-02	1,776750E-02	0,000000E+00

Table 4: Results for 10D: F13 - F18 CEC2010

F13 - 10D			
	DE	DE+HC	εDEag
Best	**-6,842937E+01**	**-6,842937E+01**	**-6,842937E+01**
Median	-6,557847E+01	**-6,842937E+01**	-6,842936E+01
Worst	-6,351751E+01	-6,557847E+01	-6,842936E+01
Mean	-6,661714E+01	-6,761578E+01	-6,842936E+01
Desv	1,909954E+00	1,298916E+00	1,025960E-06
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F14 - 10D			
	DE	DE+HC	εDEag
Best	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Median	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Worst	5,074661E+05	1,304958E+05	0,000000E+00
Mean	2,029864E+04	8,337588E+03	0,000000E+00
Desv	1,014932E+05	2,646768E+04	0,000000E+00
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F15 - 10D			
	DE	DE+HC	εDEag
Best	**0,000000E+00**	**0,000000E+00**	**0,000000E+00**
Median	3,673239E+00	3,673239E+00	**0,000000E+00**
Worst	3,673239E+00	3,673239E+00	4,497445E+00
Mean	3,085521E+00	2,497803E+00	1,798978E-01
Desv	1,374400E+00	1,748812E+00	8,813156E-01
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F16 - 10D			
	DE	DE+HC	εDEag
Best	6,663501E-01	5,520899E-01	**0,000000E+00**
Median	9,857054E-01	1,001524E+00	**2,819841E-01**
Worst	1,055176E+00	1,051114E+00	1,018265E+00
Mean	9,637395E-01	9,431711E-01	3,702054E-01
Desv	9,372062E-02	1,191967E-01	3,710479E-01
Viol	1	0	0
v	3,936864E-03	0,000000E+00	0,000000E+00

F17 - 10D			
	DE	DE+HC	εDEag
Best	1,677206E-13	**2,422344E-26**	1,463180E-17
Median	1,088417E+00	**1,416121E-13**	5,653326E-03
Worst	1,088417E+00	1,088417E+00	7,301765E-01
Mean	5,659767E-01	5,224401E-01	1,249561E-01
Desv	5,549859E-01	5,549858E-01	1,937197E-01
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

F18 - 10D			
	DE	DE+HC	εDEag
Best	2,164168E-10	1,040859E-14	**3,731439E-20**
Median	2,127288E-05	2,333675E-08	**4,097909E-19**
Worst	1,261033E-03	5,835989E-06	9,227027E-18
Mean	1,668915E-04	6,984057E-07	9,678765E-19
Desv	3,328166E-04	1,432176E-06	1,811234E-18
Viol	0	0	0
v	0,000000E+00	0,000000E+00	0,000000E+00

Table 5: Results for 30D: F01 - F06 CEC2010

F01 - 30D			
	DE	DE+HC	εDEag
Best	-8.199328E-01	-8.199345E-01	**-8.218255E-01**
Median	-8.127212E-01	-8.199343E-01	**-8.206172E-01**
Worst	-7.725882E-01	-8.127290E-01	-8.195466E-01
Mean	-8.108757E-01	-8.194010E-01	-8.208687E-01
Desv	1.139819E-02	1.663286E-03	7.103893E-04
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00

F02 - 30D			
	DE	DE+HC	εDEag
Best	-1.729535E+00	-1.961395E+00	**-2.169248E+00**
Median	-1.414513E+00	-1.752225E+00	**-2.152145E+00**
Worst	-1.139107E+00	-1.080479E+00	-2.117096E+00
Mean	-1.397607E+00	-1.683329E+00	-2.151424E+00
Desv	1.528931E-01	2.786315E-01	1.197582E-02
Viol	1	0	0
v	2.368519E-03	0.000000E+00	0.000000E+00

F03 - 30D			
	DE	DE+HC	εDEag
Best	2.867347E+01	2.867347E+01	2.867347E+01
Median	2.867347E+01	2.867347E+01	2.867347E+01
Worst	2.867347E+01	2.867347E+01	3.278014E+01
Mean	2.867347E+01	2.867347E+01	2.883785E+01
Desv	4.683177E-09	1.347989E-06	8.047159E-01
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00

F04 - 30D			
	DE	DE+HC	εDEag
Best	1.505793E-03	**3.528574E-04**	4.698111E-03
Median	3.196393E-03	**1.641278E-03**	6.947614E-03
Worst	8.019282E-03	3.836657E-03	1.777889E-02
Mean	4.106104E-03	1.529283E-03	8.162973E-03
Desv	2.002642E-03	8.342196E-04	3.067785E-03
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00

F05 - 30D			
	DE	DE+HC	εDEag
Best	-4.771455E+02	**-4.833088E+02**	-4.531307E+02
Median	-4.746565E+02	**-4.815287E+02**	-4.500404E+02
Worst	-4.665849E+02	-4.798812E+02	-4.421590E+02
Mean	-4.744888E+02	-4.815579E+02	-4.495460E+02
Desv	3.132289E+00	1.022818E+00	2.899105E+00
Viol	0	2	0
v	0.000000E+00	3.962481E-03	0.000000E+00

F06 - 30D			
	DE	DE+HC	εDEag
Best	-5.275624E+02	**-5.304233E+02**	-5.285750E+02
Median	-5.253639E+02	**-5.296760E+02**	-5.280407E+02
Worst	-5.171673E+02	-5.275613E+02	-5.264539E+02
Mean	-5.242637E+02	-5.294831E+02	-5.279068E+02
Desv	3.095089E+00	6.397257E-01	4.748378E-01
Viol	0	2	0
v	0.000000E+00	6.271008E-04	0.000000E+00

Table 6: Results for 30D: F07 - F12 CEC2010

F07 - 30D			
	DE	DE+HC	εDEag
Best	2.665074E-23	**2.470121E-29**	1.147112E-15
Median	4.872507E-22	**8.212328E-26**	2.114429E-15
Worst	5.596772E-21	1.261938E-24	5.481915E-15
Mean	9.119600E-22	1.853336E-25	2.603632E-15
Desv	1.249715E-21	2.904451E-25	1.233430E-15
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00

F08 - 30D			
	DE	DE+HC	εDEag
Best	1.404690E-23	**2.470121E-29**	2.518693E-14
Median	8.151972E-22	**6.092315E-26**	6.511508E-14
Worst	9.180257E+01	9.180261E+01	2.578112E-13
Mean	7.069181E+00	3.672104E+00	7.831464E-14
Desv	2.448721E+01	1.836052E+01	4.855177E-14
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00

F09 - 30D			
	DE	DE+HC	εDEag
Best	4.662842E-15	**2.370624E-20**	2.770665E-16
Median	6.947647E+01	6.920619E+01	**1.124608E-08**
Worst	1.164175E+02	1.617975E+02	1.052759E+02
Mean	5.783478E+01	6.193226E+01	1.072140E+01
Desv	3.090653E+01	4.094189E+01	2.821923E+01
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00

F10 - 30D			
	DE	DE+HC	εDEag
Best	3.130917E+01	**3.130908E+01**	3.252002E+01
Median	3.131222E+01	**3.130909E+01**	3.328903E+01
Worst	2.255249E+02	2.650643E+02	3.463243E+01
Mean	3.908073E+01	4.802656E+01	3.326175E+01
Desv	3.884254E+01	5.830148E+01	4.545577E-01
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00

F11 - 30D			
	DE	DE+HC	εDEag
Best	-3.923431E-04	**-3.923384E-04**	-3.268462E-04
Median	**-3.923406E-04**	-3.922639E-04	-2.843296E-04
Worst	-3.923335E-04	-3.917380E-04	-2.236338E-04
Mean	-3.923403E-04	-3.922238E-04	-2.863882E-04
Desv	2.357245E-09	1.408836E-07	2.707605E-05
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00

F12 - 30D			
	DE	DE+HC	εDEag
Best	**-1.992635E-01**	**-1.992635E-01**	-1.991453E-01
Median	-1.992635E-01	**-1.992634E-01**	5.337125E+02
Worst	1.181452E+02	1.181352E+02	5.461723E+02
Mean	-1.621667E+01	-1.460144E+01	3.562330E+02
Desv	1.096712E+02	1.010505E+02	2.889253E+02
Viol	0	0	1
v	0.000000E+00	0.000000E+00	3.240709E-01

Table 7: Results for 30D: F13 - F18 CEC2010

F13 - 30D			
	DE	DE+HC	εDEag
Best	**-6.741604E+01**	-6.561286E+01	-6.642473E+01
Median	-4.599168E+01	-6.110990E+01	**-6.531507E+01**
Worst	-4.121510E+01	-4.535138E+01	-6.429690E+01
Mean	-4.905102E+01	-5.770437E+01	-6.535310E+01
Desv	7.655549E+00	8.005156E+00	5.733005E-01
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00
F14 - 30D			
	DE	DE+HC	εDEag
Best	1.372617E-23	**0.000000E+00**	5.015863E-14
Median	3.934449E-20	**9.064256E-25**	1.359306E-13
Worst	2.504195E-18	5.824681E-23	2.923513E-12
Mean	2.084984E-19	4.914055E-24	3.089407E-13
Desv	5.021597E-19	1.179499E-23	5.608409E-13
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00
F15 - 30D			
	DE	DE+HC	εDEag
Best	**2.160323E+01**	2.160324E+01	2.160345E+01
Median	**2.160323E+01**	2.160329E+01	2.160375E+01
Worst	2.160325E+01	2.160353E+01	2.160403E+01
Mean	2.160324E+01	2.160331E+01	2.160376E+01
Desv	2.713109E-06	7.244528E-05	1.104834E-04
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00
F16 - 30D			
	DE	DE+HC	εDEag
Best	1.105227E+00	1.096205E+00	**0.000000E+00**
Median	1.148119E+00	1.133664E+00	**0.000000E+00**
Worst	1.184353E+00	1.179000E+00	5.421011E-20
Mean	1.148630E+00	1.136243E+00	2.168404E-21
Desv	1.979490E-02	1.909056E-02	1.062297E-20
Viol	1	0	0
v	5.068577E-03	0.000000E+00	0.000000E+00
F17 - 30D			
	DE	DE+HC	εDEag
Best	1.964317E-21	**4.124379E-31**	2.165719E-01
Median	**3.513397E-01**	3.513484E-01	5.315949E+00
Worst	3.513403E-01	3.514742E-01	1.889064E+01
Mean	3.232326E-01	2.810930E-01	6.326487E+00
Desv	9.728158E-02	1.434447E-01	4.986691E+00
Viol	0	0	0
v	0.000000E+00	0.000000E+00	0.000000E+00
F18 - 30D			
	DE	DE+HC	εDEag
Best	3.788228E-02	**2.538331E-04**	1.226054E+00
Median	8.855682E-01	**4.070273E-03**	2.679497E+01
Worst	3.886298E+00	1.231137E-02	7.375363E+02
Mean	1.236158E+00	4.627798E-03	8.754569E+01
Desv	1.072778E+00	3.173508E-03	1.664753E+02
Viol	1	0	0
v	2.115780E-05	0.000000E+00	0.000000E+00