Elitistic Evolution: An Efficient Heuristic for Global Optimization

Francisco Viveros Jiménez[1], Efrén Mezura-Montes[2], and Alexander Gelbukh[3]

[1] Universidad del Istmo Campus Ixtepec, Cd. Ixtepec, Oaxaca, México
pacovj@hotmail.com
[2] Laboratorio Nacional de Informática Avanzada, Xalapa, Veracruz, México
emezura@lania.mx
[3] Centro de Investigación en Computación del Instituto Politécnico Nacional, D.F. México
gelbukh@gelbukh.com

Abstract. A new evolutionary algorithm, Elitistic Evolution (termed EEv), is proposed in this paper. EEv is an evolutionary method for numerical optimization with adaptive behavior. EEv uses small populations (smaller than 10 individuals). It have an adaptive parameter to adjust the balance between global exploration and local exploitation. Elitism have great influence in EEv' proccess and that influence is also controlled by the adaptive parameter. EEv' crossover operator allows a recently generated offspring individual to be parent of other offspring individuals of its generation. It requires the configuration of two user parameters (many state-of-the-art approaches uses at least three). EEv is tested solving a set of 16 benchmark functions and then compared with Differential Evolution and also with some well-known Memetic Algorithms to show its efficiency. Finally, EEv is tested solving a set of 10 benchmark functions with very high dimensionality (50, 100 and 200 dimensions) to show its robustness.

1 Introduction

The global optimization problem is unsolved because of its high level of complexity. Many different scopes were generated in recent years. Evolutionary algorithms is one of the many different approaches that have been used in recent years. Evolutionary algorithms (**EAs**) are stochastic techniques based on the Darwinian principle of the *survival of the fittest*. Most EAs offer important advantages such robustness, reliability, global search capability and low information requirement. Some representative techniques of EAs are: Genetic Algorithms [5], Evolutive Strategies [1] and Differential Evolution [2].

EAs are population-based techniques. Each individual of the population represent a candidate solution (commonly conformed at least by 30 individuals). EAs generate offspring using a crossover operator. Normally, some population individuals are selected to be parents of the offspring individuals. Then, sligth alterations to some individuals are performed with a mutation operator. Finally,

M. Kolehmainen et al. (Eds.): ICANNGA 2009, LNCS 5495, pp. 171–182, 2009.

old solutions are replaced with some specific new solutions. Normally, EAs use the number of generations as termination condition. A generation represent the proccess of creating new solutions and replacing the old ones. Evaluations of the objetive function (**FEs**) are performed in each iteration. Most of state-of-the-art technique use at least three user parameters.

This paper describes a new EA called Elitistic Evolution (**EEv**). EEv has the following major differences with a traditional EA:

1. Uses small populations (smaller than 10 individuals).
2. Uses an adaptive parameter to adjust the balance between global exploration and local exploitation in all the processes.
3. Elitism have great influence in EEv' proccess and that influence is also controlled by the adaptive parameter.
4. Its mutation operator generate the offspring.
5. It uses a new crossover operator. EEv' crossover operator allows a recently generated offspring individual to be parent of other offspring individuals of its generation.
6. It uses two replacement politics balanced by the adaptive parameter.
7. It requires the configuration of two user parameters.

The adaptive nature of EEv allows the algorithm to have precision and high convergency speed, even in complex problems with very high dimensionality ($N > 50$). The performance and features of this technique are shown by testing it in a set of 16 well-known functions taken from the literature [4,8]. The paper also presents a performance comparison with DE/rand/1/bin and some well-known Memetic Algorithms. However, it is important to remember the No Free Lunch Theorems for Search [3]. No Free Lunch Theorems state that an algorithm which performs well on some test functions, will not necessarily be competitive in a different set of problems.

The sections are organized as follows. The following section contains a detailed description of EEv. Section 3 contains the experimental design and results. Section 4 concludes this paper.

2 Elitistic Evolution

EEv is a population-based stochastic optimizer. The two main foundations of EEv are:

1. **Elitism.** The best individual have special considerations in each EEv stage the elite individual have special considerations.
2. **Adaptive Behavior.** An adaptive parameter control the behavior of all stages of an EEv iterations. The two main behaviors affected by this parameter are: the effect of elitism and the balance of global vs local exploration. The step size used by the mutation operator on each dimension is also adaptive.

Algorithm 1 describes an EEv iteration. The following differences are observed between EEv and a common EA:

1. Each X^g individual became a parent.
2. Mutation is performed before crossover.
3. Crossover replaces offspring instead of generating new individuals.
4. Mutation, crossover and replacement are affected by an adaptive parameter called Enviromental Pressure (C).

Algorithm 1. Algorithm for any g iteration of EEv

```
1 Recalculation of adaptive parameters;
2 Copy each X^g individual in n^g;
3 Perform mutation_rs(n^g, C, b⃗) to each n^g individual;
4 Replace each n^g with a new n^g created by crossover(X^g, n^g, C);
5 Evaluate the new n^g individuals;
6 for the first C individuals of X^{g+1} do
7 |  Use the C best individuals of the union of X^g and n^g;
8 for the remaining P − C individuals of X^{g+1} do
9 |  Select P − C random individuals from n^g;
```

EEv needs the adjustment of two parameters which are:

- Population size (P) which tells how many individuals will form the population. $P > 2$.
- Base mutation (B) used as the initial step size for the mutation operator. $B \in [0.0, 1.0]$.

A large P or B value increase the diversity of possible solutions, promoting the global exploration of the search space, and, consequently, demoting the exploitation capacity. The balance between P and B is crucial for the efficiency of the algorithm.

EEv uses two adaptive parameters which are:

- $\overrightarrow{b}$, the N step sizes for random step mutation operator, where N is the number of decision variables in the problem. $\overrightarrow{b}$ have an initial value of B in each N dimension.
- C. Indicates the number of individuals to be affected by local exploration processes. It adjust the balance between global and local search. $C \in [1, P]$. Lower C values promote global exploration; higher C values promote local exploitation. C value depends on the success of the search of the better solution. C have an initial value of 1. The specific effects of C are explained in the description of each stage below.

2.1 Recalculation of Parameters

C and $\overrightarrow{b}$ values change depending on the search success of the last generation, determined by the comparison of the best fitness values of the current and previous generations. Algorithm 2 shows the recalculation for C and $\overrightarrow{b}$. We take $F(X^0_{best}) = F(X^1_{best})$, where X^g_{best} is the elite individual. If better results are found, then C is decreased, encouraging global exploration, and $\overrightarrow{b}$ values are adaptively increased to provide a more precise exploration. Otherwise, C is increased and $\overrightarrow{b}$ values are decreased to encourage local exploitation.

Algorithm 2. Recalculation of adaptive parameters

```
if F(X^{g-1}_best) > F(X^g_best) then
    if C > 1 then
        C = C − 1;
    b⃗ = |X^{g-1}_best − X^g_best/(upbound − lowbound)| × (1.0 + rnd(0.0, 1.0) × (G − g)/G);
    if a b⃗ value is equal to 0 then
        Replace it with B × (1.0 − rnd(0.0, 1.0) × g/G) ;
else
    if C < P then
        C = C + 1;
    b⃗ = b⃗ × (1.0 − rnd(0.0, 1.0) × g/G);
```

2.2 Mutation

Mutation operator is very similar to Hill-climbing method. This operator generate the offspring individuals by performing a random number of sligth alterations in some dimensions of each X^g individual. The steps stored in $\overrightarrow{b}$ represent the maximum percent of search space that the mutation can perform. Algorithm 3 describes the mutation operator.

EEv mutation operator is based on the mutation technique proposed in [6]. It generates a new individual close to the original point. Due to the adaptive behavior of EEv we have to consider the following facts:

1. The first C mutations represent local exploitation and the last $P - C$ mutationts represent global exploraitions. That is because the first C mutations are affected in an sligther way due to the calculus of the M ratio (see equation 1 below).
2. The operator tends to perform smaller changes at later stages of the process due to decreasing of the values $\overrightarrow{b}$.
3. The number of alterations depends on C value. .

$$M = \begin{cases} \overrightarrow{b_k} & \text{if } i \leq C \\ B \times (1.0 - rnd(0.0, 1.0) \times g/G) & \text{if } i > C \end{cases} \quad (1)$$

Algorithm 3. Mutation algorithm for an i individual

```
1 alterations = rnd(⌈nv × (C/P)⌉, N) ;
2 for all alterations do
3     k = rnd(1, N);
4     Calculate M with equation 1 ;
5     repeat
6         aux = n^g_{k,i} + (upbound_k − lowbound_k) × rnd(−M, M);
7     until (aux > lowerbound_k) and (aux < upperbound_k) ;
8     n^g_{k,i} = aux;
```

Algorithm 4. Crossover operator algorithm

```
1 for each n^g_i offspring individual do
2     k = rnd(0, P − C) + 1;
3     l = rnd(0, P − C) + 1;
4     m = rnd(1, P);
5     c1 = rnd(0.0, 1.0);
6     c2 = rnd(0.0, 1.0 − c1);
7     c3 = 1.0 − c2 − c1;
8     n^g_i = c1 × n^g_k + c2 × X^g_l + c3 × n^g_m;
```

2.3 Reproduction

The reproduction stage creates the final offspring individuals. Algorithm 4 illustrates the crossover operator. This crossover operator uses two offspring individuals, k and m, and one population individual l. We have to consider the following facts:

- The crossover operator allows each offspring individual to have a chance of being selected as a parent of the following alterations to offspring individuals.
- Each offspring individual will be always at a point between its three parents.
- Any X^g and n^g individual have the posibility of remain intact.
- The crossover can take into account only two individuals: two n^g individuals or one individual from X^g and n^g.

C value controls the selection of l and k: when $C = 1$, any individual can be selected; when $C = P$, elitism is ensured, promoting exploration around the best individual. Evaluation of the offspring individuals is performed at the end of this stage.

2.4 Replacement

EEv uses two replacement processes: non-generational replacement, borrowed from $(\mu + \lambda)$-ES [1], and random generational replacement. The first replacement method provides convergence capabilities, while the second one provides exploration capabilities.

As in the previous stages, the balance between these two processes is controlled by C: the first C individuals of the new generation population are selected by non-generational replacement and the remaining ones are select randomly from the offspring individuals. Elitism is ensured by non-generational replacement.

The first C individuals of the new generation are ordered by their fitness values, and EEv replacement ensures that the first population individual always be the elite; no search for the elite is required at any stage.

3 Experimental Setup

We followed the experimental setup similar to [7] with 6 tests:

1. Evaluation of performance;
2. Study of the adaptive behavior;
3. Evaluation of sensitivity to population size;
4. Evaluation of sensitivity to the B parameter;
5. Study of scalability;
6. Comparison with MAs.

The benchmark functions are specified in table 1. The functions f were taken from [4], and functions F, from the test suite for CEC 2005 Special Session on real-parameter optimization [8]. In each test, we conducted 50 trials with each function. All the experiments were performed using a Pentium 4 PC with 512 MB of RAM, in C Linux environment.

3.1 Evaluation of Performance

It was based on the test proposed in [8]. The *function error value* ε for a solution x is $f(x) - f(x^*)$, where x^* is the global optimum of f [7]. The maximum number **MAX** of evaluations of f was $10,000\,N$, where N is the dimension of the problem. The fitness evaluation criteria were as follows:

1. **Error** is a compound value formed by all trials as $AVG(\overrightarrow{\varepsilon}) \pm STDDEV(\overrightarrow{\varepsilon})$.
2. **Evaluation** is another compound value formed by all trials as $AVG(\overrightarrow{\alpha}) \pm STDDEV(\overrightarrow{\alpha})(\beta)$, where α is the number of function evaluations (**FEs**) required to reach certain error value before MAX value and β is the number of trials with an α value. For all the functions, this value was 10^{-6} except for F_8, F_9, and F_{10}, where the accuracy was 10^{-2}.

We compared the performance of EEv and DE/rand/1/bin. The tests were conducted on a set of sixteen functions with $N = 30$. Error and evaluation values are reported on Table 2. Best results are marked in boldface. We used $P = 5$ and $B = 0.6$ in this test. The *error* and *evaluation* values for DE/rand/1/bin were calculated using a $P = N$, $CR = 0.9$ and $F = 0.9$ as suggested in [7].

Table 1. Test functions

Unimodal functions	
Separable	
f_{sph}	Sphere model
F_1	Shifted sphere function
Non-separable	
F_3	Shifted rotated high conditioned elliptic function
F_4	Shifted Schwefel's problem 1.2 w/ noise in fitness
Multimodal functions	
Separable	
f_{sch}	Generalized Schefel's problem 2.26
f_{ras}	Generalized Rastrigin's function
F_9	Shifted Rastrigin's function
Non-separable	
f_{ros}	Generalized Rosenbrock's function
f_{ack}	Ackley's function
f_{grw}	Generalized Griewank's function
f_{sal}	Salomon's function
f_{whi}	Whitley's function
$f_{pen1,2}$	Generalized penalized functions
F_8	Shifted rotated Ackley's function with global optimum on bounds
F_{10}	Shifted Rotated Rastrigin's function

We observe that EEv:

- Has fast convergence: the reported evaluation values are small.
- Is consistent: the standard deviation is relatively small as compared with the mean value on all the test problems.
- Is competitive: it overperforms DE/rand/1/bin on nine out of sixteen functions.
- Is faster than DE/rand/1/bin. It reaches the accuracy value on ten out of sixteen functions using less FEs than DE/rand/1/bin.

EEv' success key features are:

1. Adaptive restart mechanism. EEv' restart mechanism allows partial population restart, accelerating convergence.
2. Mutation operator. The operator is very similar to a Hill-Climbing operator, giving EEv a similar behavior to a memetic algorithm. The adaptive step size vector allows a more efficient exploration for the current search situation.
3. Use of recently crossover' generated offspring individuals.

3.2 Study of the Adaptive Behavior

Experiments with $P = 5$ and $B = 0.6$ were conducted to evaluate the adaptive behavior of EEv. The tests were conducted on a set of sixteen functions with

Table 2. Comparison between EEv and DE/rand/1/bin error and evaluations values on problems with $N = 30$

Error	EEv	DE/Rand/1/Bin
f_{sph}	$\mathbf{4.35E-22 \pm 8.37E-22}$	$5.73E-17 \pm 2.03E-16$
F_1	$1.67E-12 \pm 4.37E-12$	$\mathbf{3.58E-81 \pm 1.36E-81}$
F_3	$\mathbf{9.21E+05 \pm 9.22E+04}$	$3.63E+06 \pm 9.22E+05$
F_4	$6.80E+04 \pm 1.62E+04$	$\mathbf{5.54E+01 \pm 6.37E+01}$
f_{ras}	$\mathbf{4.32E-12 \pm 1.34E-11}$	$2.55E+01 \pm 8.14E+00$
F_9	$\mathbf{1.59E-01 \pm 2.59E-01}$	$2.43E+01 \pm 6.22E+00$
f_{sch}	$1.67E+03 \pm 6.41E+01$	$\mathbf{4.90E+02 \pm 2.34E+02}$
f_{ros}	$\mathbf{1.42E+01 \pm 7.28E+00}$	$5.20E+01 \pm 8.56E+01$
f_{whit}	$\mathbf{2.33E+01 \pm 1.94E+01}$	$3.10E+02 \pm 1.07E+02$
f_{pen1}	$\mathbf{5.82E-25 \pm 4.80E-25}$	$4.56E-02 \pm 1.31E-01$
F_8	$\mathbf{2.02E+01 \pm 2.65E-02}$	$2.09E+01 \pm 1.33E-01$
f_{pen2}	$\mathbf{9.01E-24 \pm 1.12E-23}$	$1.44E-01 \pm 7.19E-01$
f_{grw}	$3.08E-02 \pm 1.57E-02$	$\mathbf{2.66E-03 \pm 5.73E-03}$
f_{ack}	$1.04E-07 \pm 1.50E-07$	$\mathbf{1.70E-09 \pm 1.32E-09}$
f_{sal}	$1.04E+00 \pm 6.58E-01$	$\mathbf{2.52E-01 \pm 8.14E+00}$
F_{10}	$3.34E+02 \pm 3.24E+01$	$\mathbf{7.33E+01 \pm 6.66E+01}$
Evaluations		
f_{sph}	$\mathbf{93827.6 \pm 1157.87(50)}$	$148650.8 \pm 6977.7(50)$
F_1	$\mathbf{97640.80 \pm 1356.82(50)}$	$153450.1 \pm 5780.4(50)$
f_{ras}	$\mathbf{111151.8 \pm 7367.1(50)}$	–
F_9	$\mathbf{115550.8 \pm 11845.2(43)}$	–
f_{pen1}	$\mathbf{67299.5 \pm 1279.93(50)}$	$160955.2 \pm 63176.3(43)$
f_{pen2}	$\mathbf{78541.6 \pm 1529.95(50)}$	$156016.9 \pm 31515.8(48)$
f_{ras}	$\mathbf{200743.1 \pm 31447.5(8)}$	–
f_{ros}	$\mathbf{127230(1)}$	–
f_{ack}	$\mathbf{164930.0 \pm 15637.1}(49)$	$215456.1 \pm 9721.4(\mathbf{50})$
f_{grw}	$\mathbf{104585.31 \pm 17771.17}(16)$	$190292.5 \pm 63478.8(\mathbf{38})$

$N = 30$. Frequencies and mean values of C parameters were taken from a random test and reported in Table 3. The following observations can be obtained from the experiments:

- Elitism is predominant in EEv's optimization process. A statistic mode of P for C parameter is present on all the problems. Additionally, the mean value of C on the test is equal to 4.94 (very close to P).
- C parameter is sensitive to the kind of search space EEv is working with. Each problem needs a different amount of global and local exploration.

3.3 Sensitivity to Population Size

Experiments with different P values were conducted to evaluate the sensitivity of EEv to variations of population size. The tests were conducted on a set of ten

Table 3. C frequencies for a random test run with $P = 5$, $B = 0.6$ and $N = 30$

	f_{sph}	F_1	F_3	F_4	F_9	f_{ras}	F_{sch}	f_{ros}	f_{ack}	f_{grw}
1	15	11	18	2	4	4	3	11	11	15
$P-3$	84	71	247	4	46	46	27	128	61	61
$P-2$	365	281	1366	4	289	315	161	598	306	301
$P-1$	2258	1457	7094	13	1760	1960	1399	3556	1506	1550
P	57278	58180	51275	59977	57901	57675	58410	55707	58116	58073
μ	4.94	4.96	4.82	4.99	4.95	4.95	4.96	4.91	4.96	4.96

functions with $N = 30$. The P values used were 3, 10, and 30. Table 4 shows the error values obtained in the test with 10 benchmark functions.

Performance of EEv is sensitive to population size in all the cases. EEv works best with very small populations. EEv works better with smaller P values in 7 out of 10 test cases. It is important to observe that:

- A smaller P value increments the probability of elitism.
- Smaller P values amplify the effect of the C parameter: major changes in the proportion of individuals involved in exploitation and exploration occur between generations.

3.4 Sensitivity to the B Parameter

Experiments with different B values were conducted to evaluate the sensitivity of EEv to variations of B parameter. The tests were conducted on a set of ten functions with $N = 30$. The B values used were 0.1, 0.45, and 0.8.

B controls the initial and maximum step size for $\overrightarrow{b}$ – the adaptive parameter that contains the step size to be used by the mutation operator. The mutation operator is very similar to a hill-climbing algorithm. It allows the generation of offspring individuals around a parent with a maximum step size specified by $\overrightarrow{b}$. Table 4 shows the error values obtained by using different B values in the test with sixteen benchmark functions. That table shows that:

- Performance in EEv is more sensitive to the B parameter.
- B value is problem-dependent. However, EEv tends to have similar behavior on problems with similar features.
- Greater B values encourage a more optimal exploration.
- EEv performs competitively without fine adjustment of B.

3.5 Scalability Test

Experiments with different N dimensions values were conducted to evaluate the robustness of EEv. The tests were conducted on a set of ten functions. The N values used were 100, and 200. The error values obtained are reported in Table 5.

Table 4. EEv Error values obtained for different P and B values on problems with $N = 30$

$B = 0.6$	$P = 3$	$P = 10$	$P = 30$
f_{sph}	$\mathbf{6.15E-27 \pm 8.24E-27}$	$4.50E-18 \pm 5.51E-18$	$1.68E-12 \pm 1.69E-12$
F_1	$\mathbf{1.90E-14 \pm 1.02E-14}$	$1.40E-10 \pm 2.56E-10$	$3.23E-07 \pm 1.69E-06$
F_3	$\mathbf{7.37E+05 \pm 8.16E+04}$	$1.30E+06 \pm 1.31E+05$	$3.08E+06 \pm 1.88E+05$
F_4	$\mathbf{3.96E+04 \pm 1.04E+04}$	$7.14E+04 \pm 1.46E+04$	$4.08E+04 \pm 2.69E+03$
f_{ras}	$6.63E-02 \pm 1.69E-01$	$\mathbf{2.91E-08 \pm 8.33E-08}$	$3.32E-02 \pm 1.75E-01$
f_{sch}	$\mathbf{1.61E+03 \pm 9.52E+01}$	$1.68E+03 \pm 1.24E+02$	$1.66E+03 \pm 1.02E+02$
f_{grw}	$4.03E-02 \pm 2.30E-02$	$2.69E-02 \pm 1.79E-02$	$\mathbf{1.00E-02 \pm 5.79E-03}$
f_{pen1}	$\mathbf{1.77E-29 \pm 1.24E-29}$	$2.11E-19 \pm 5.45E-19$	$1.33E-10 \pm 6.18E-10$
f_{ack}	$\mathbf{2.24E-10 \pm 1.60E-10}$	$2.41E-03 \pm 6.97E-03$	$5.13E-01 \pm 2.93E-01$
f_{ros}	$1.52E+01 \pm 9.89E+00$	$1.76E+01 \pm 9.64E+00$	$\mathbf{7.23E+00 \pm 1.10E+01}$
$P = 5$	$B = 0.1$	$B = 0.45$	$B = 0.8$
f_{sph}	$\mathbf{1.53E-23 \pm 7.16E-23}$	$1.11E-22 \pm 9.36E-23$	$2.46E-22 \pm 2.97E-22$
F_1	$\mathbf{1.03E-12 \pm 2.03E-12}$	$2.17E-12 \pm 8.33E-12$	$2.12E-12 \pm 8.33E-12$
F_3	$9.47E+05 \pm 1.52E+05$	$\mathbf{9.00E+05 \pm 7.33E+04}$	$9.27E+05 \pm 1.29E+05$
F_4	$1.57E+05 \pm 5.38E+04$	$8.16E+04 \pm 2.26E+04$	$\mathbf{5.94E+04 \pm 6.83E+03}$
f_{ras}	$3.67E+01 \pm 4.22E+00$	$3.31E-02 \pm 1.75E-01$	$\mathbf{1.70E-12 \pm 2.51E-12}$
f_{sch}	$5.67E+03 \pm 2.42E+02$	$1.83E+03 \pm 1.40E+02$	$\mathbf{1.61E+03 \pm 9.44E+01}$
f_{pen1}	$\mathbf{1.74E-25 \pm 7.85E-25}$	$1.31E-24 \pm 2.71E-24$	$9.81E-25 \pm 4.31E-25$
f_{ack}	$1.13E-07 \pm 2.43E-07$	$1.80E-07 \pm 4.28E-07$	$\mathbf{9.21E-08 \pm 1.13E-07}$
f_{ros}	$2.08E+01 \pm 5.19E+01$	$3.87E+01 \pm 2.29E+01$	$\mathbf{1.46E+01 \pm 2.02E+01}$

This table also show a comparison between EEv, DE/rand/1/bin, and DEahcSPX. DEahcSPX is a Memetic Algorithm which overcomes DE/rand/1/bin performance and was proposed in [7]. The error values from this technique were taken also from [7]. We obtain the following conclusions:

- EEv is a robust algorithm.
- EEv mantains its performance on problems with very high dimensionality.
- EEv is highly competitive on problems with $N \geq 100$. It overperforms the other two techniques on all test problems.

3.6 Comparison with Memetic Algorithms

A comparison with two well-knowm memetic algorithms was conducted. We select two techniques: minimal generation gap (**MGG**) [9] and generalized generation gap (**G3**) [10]. Two crossover operators were selected for this paper: unimodal normal distribution crossover (**UNDX**) and parent centric crossover (**PCX**). The selected combinations were MGG+UNDX and G3+PCX. The tests were conducted on a set of ten functions with $N = 30$. Error values are reported in Table 6. Best results are marked in boldface. The *error* values for both algorithms were taken from [7]. We reach the following conclusions:

Table 5. Comparison between EEv, DE/rand/1/bin and DEahcSPX error values in ten problems with $N = 100$ and $N = 200$

$N = 100$	EEv	DEahcSPX	DE/Rand/1/Bin
F_{sph}	$\mathbf{1.39E-21 \pm 5.66E-22}$	$5.01E+01 \pm 8.94E+01$	$4.28E+03 \pm 1.27E+03$
F_{ras}	$\mathbf{6.635E-02 \pm 1.69E-01}$	$4.75E+02 \pm 6.55E+01$	$8.30E+02 \pm 6.51E+01$
F_{sch}	$\mathbf{5.79E+03 \pm 2.36E+02}$	$2.48E+04 \pm 2.71E+03$	$2.54E+04 \pm 2.15E+03$
F_{ros}	$\mathbf{3.66E+01 \pm 1.77E+01}$	$1.45E+05 \pm 1.11E+05$	$3.33E+08 \pm 1.67E+08$
F_{ack}	$\mathbf{9.30E-09 \pm 1.12E-08}$	$1.91E+00 \pm 3.44E-01$	$8.81E+00 \pm 8.07E-01$
F_{sal}	$\mathbf{2.39E+00 \pm 2.39E-01}$	$3.11E+00 \pm 5.79E-01$	$1.02E+01 \pm 7.91E-01$
F_{whit}	$\mathbf{5.67E+02 \pm 1.95E+02}$	$4.06E+10 \pm 6.57E+10$	$5.44E+15 \pm 5.07E+15$
F_{pen1}	$\mathbf{1.13E-23 \pm 7.33E-24}$	$4.33E+00 \pm 1.75E+00$	$6.20E+06 \pm 7.38E+05$
$N = 200$	EEv	DEahcSPX	DE/Rand/1/Bin
F_{sph}	$\mathbf{6.58E-21 \pm 1.87E-21}$	$7.01E+03 \pm 1.07E+03$	$1.26E+05 \pm 1.06E+04$
F_{ras}	$\mathbf{3.64E-01 \pm 2.96E-01}$	$1.53E+03 \pm 8.31E+01$	$2.37E+03 \pm 7.24E+01$
F_{sch}	$\mathbf{1.18E+04 \pm 3.71E+02}$	$6.61E+04 \pm 1.44E+03$	$6.66E+04 \pm 1.32E+03$
F_{ros}	$\mathbf{3.61E+01 \pm 2.91E+01}$	$1.11E+08 \pm 2.63E+07$	$2.97E+10 \pm 3.81E+09$
F_{ack}	$\mathbf{3.02E-09 \pm 7.34E-10}$	$8.45E+00 \pm 4.13E-01$	$1.81E+01 \pm 2.26E-01$
F_{sal}	$\mathbf{4.54E+00 \pm 2.09E-01}$	$1.10E+01 \pm 4.38E-01$	$3.69E+01 \pm 1.80E+00$
F_{whit}	$\mathbf{4.09E+03 \pm 1.14E+03}$	$4.21E+13 \pm 1.74E+13$	$3.13E+18 \pm 9.48E+17$
F_{pen1}	$\mathbf{5.98E-23 \pm 1.26E-23}$	$2.27E+01 \pm 5.73E+00$	$3.49E+08 \pm 7.60E+07$

Table 6. Comparison between EEv error values and two MAs in problems with N=30

	EEv	MGG+UNDX	G3+PCX
F_{sph}	$4.35E-22 \pm 8.37E-22$	$1.37E-11 \pm 1.94E-11$	$\mathbf{3.58E-81 \pm 1.36E-81}$
F_{ras}	$\mathbf{4.32E-12 \pm 1.34E-11}$	$1.35E+00 \pm 1.03E+00$	$1.75E+02 \pm 3.37E+01$
F_{sch}	$\mathbf{1.67E+03 \pm 6.41E+01}$	$4.12E+03 \pm 1.72E+03$	$4.04E+03 \pm 1.09E+03$
F_{whit}	$\mathbf{2.33E+01 \pm 1.94E+01}$	$4.28E+02 \pm 3.82E+01$	$3.44E+02 \pm 2.97E+00$
F_{pen1}	$\mathbf{5.82E-25 \pm 4.80E-25}$	$4.93E-02 \pm 3.50E-02$	$4.35E+00 \pm 6.94E+00$
F_{ack}	$\mathbf{1.04E-07 \pm 1.50E-07}$	$8.23E-07 \pm 4.64E-07$	$1.48E+01 \pm 4.17E+00$
F_{pen2}	$\mathbf{9.01E-24 \pm 1.12E-23}$	$4.39E-04 \pm 2.20E-03$	$1.50E+01 \pm 1.58E+01$
F_{sal}	$1.04E+00 \pm 6.58E-01$	$\mathbf{1.50E-01 \pm 4.95E-02}$	$4.64E+00 \pm 4.74E+00$
F_{grw}	$3.08E-02 \pm 1.57E-02$	$\mathbf{2.96E-04 \pm 1.48E-03}$	$1.07E-02 \pm 1.30E-02$
F_{ros}	$1.42E+01 \pm 7.28E+00$	$2.81E+01 \pm 1.23E+01$	$\mathbf{4.18E+00 \pm 9.68E+01}$

1. EEv is a competitive technique. It overperforms the other two techniques on six out of ten functions.
2. Uses one less parameter than the other two techniques: MGG and G3 require the adjustment of P, μ, and λ.
3. The algorithm is simplier than the two memetic algorithms.

4 Conclusions and Future Work

The paper describes a new evolutionary method called EEv. EEv is a population-based technique that uses only two user-defined parameters (one less than the

majority of the state-of-the-art techniques). The adaptive parameter C allows EEv to reach an effective balance between local exploitation and global exploration.

The tests show that EEv is a competitive approach: it performs well in most test cases and have great efficiency on problems with high dimensionality. In addition to that, it is very fast and precise in some cases: it reach the target accuracy value faster than DE/rand/1/bin. However, it has difficulty solving unimodal non-separable functions. We conclude that EEv performs better with small populations EEV is sensitive to B parameter value. B value depends on the problem features.

More comparative work and further studies should be carried out to provide a more detailed analysis and refinement. Future work with constrained functions should be performed to observe the behavior of EEv. New mechanisms should be tested to improve EEv's performance on shifted functions.

References

1. Beyer, H.-G., Schwefel, H.-P.: Evolution strategies: a comprehensive introduction. Natural Computing 1(1) (2002)
2. Storn, R., Price, K.: Differential Evolution - a simple and efficient heuristic for global optimization. Journal of Global Optimization 11(4) (1997)
3. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. on Evolutionary Computation (1997)
4. Mezura-Montes, E., Coello, C.C.A., Velazquez, R.J.: A comparative study of differential evolution variants for global optimization. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation (2006)
5. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading (1989)
6. Viveros, J.F.: DSE: An Hybrid Evolutionary Algorithm with Mathematical Search Method. Special issue journal Research in Computing Science (2008)
7. Noman, N., Iba, H.: Accelerating Differential Evolution Using an Adaptive Local Search. IEEE Transactions on Evol. Comput. 12(1) (2008)
8. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.-P., Auger, A., Tiwari, S.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Nanyang Technol. Univ., Singaporem IIT Kanpur, India, KanGal Rep. 2005005 (2005)
9. Satoh, H., Yamamura, M., Kobayashi, S.: Minimal generation gap models for GAs considering both exploration and exploitation. In: Proc. IIZUKA 1996 (1996)
10. Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter optimization. Evol. Comput. 10(4) (2002)