Empirical Measurements on the Convergence Nature of Differential Evolution Variants

G. Jeyakumar[1] and C. Shanmugavelayutham[2]

[1,2] Assistant Professor
Department of Computer Science and Engineering
Amrita School of Engineering
Amrita Vishwa VidyaPeetham
Coimbatore, Tamil Nadu, India
g_jeyakumar@cb.amrita.edu,
cs_velayutham@cb.amrita.edu

Abstract. In this paper, we present an empirical study on convergence nature of Differential Evolution (DE) variants to solve unconstrained global optimization problems. The aim is to identify the convergence behavior of DE variants and compare. We have chosen fourteen benchmark functions grouped by feature: unimodal separable, unimodal nonseparable, multimodal separable and multimodal nonseparable. Fourteen variants of DE were implemented and tested on these problems for dimensions of 30. The variants are well compared by their Convergence Speed, Quality Measure and Population Convergence Measure.

Keywords: Differential Evolution, Variants, Global Optimization, Convergence, Population Variance.

1 Introduction

Evolutionary algorithms (EA) have been widely used to solve optimization problems. Differential Evolution [1] is an EA proposed to solve optimization problems, mainly to continuous search spaces. The DE algorithm has been successfully applied to many global optimization problems [2]. As traditional EAs, several optimization problems have been successfully solved by using DE [3]. It shows superior performance in both widely used benchmark functions and real-world application [4, 5]. DE shares similarities with traditional EAs. As in other EAs, two main processes are the perturbation process (crossover and mutation) which ensures the exploration of the search space and the selection process which ensures the exploitation properties of the algorithm. Both perturbation and the selection process are simpler in DE, the perturbation of a population element is done by probabilistically replacing it with an offspring obtained by adding to a randomly selected element a perturbation proportional with the difference between other two randomly selected elements. The selection is done by one to one competition between the parent and its offspring. The structure of DE algorithm is presented in Figure. 1.

N. Meghanathan et al. (Eds.): CCSIT 2011, Part I, CCIS 131, pp. 472–480, 2011.

Based on different strategies followed for perturbation, there exists various DE variants, they differ in the way how the solution is generated. With seven commonly used differential mutation strategies, and two crossover schemes (binomial and exponential), we get fourteen possible variants of DE (as listed in Table 2). In this paper, an empirical convergence analysis of these 14 variants has been attempted.

The remainder of the paper is organized as follows. After a brief review of the related work in Section 2, Section 3 details the design of experiments and describes about the numerical tests in our study, Section 4 discusses the simulation results and finally Section 5 concludes the paper.

```
Population Initialization X(0) ← {x1(0),...,xNP(0)}
  g ←0
  Compute { f(x1(g)),...,f(xNP(g)) }
  while the stopping condition is false do
   for i = 1 to NP do
     MutantVector:yi ← generatemutant(X(g))
     TrialVector:zi←crossover(xi(g),yi)
     if f(zi) < f(xi(g)) then
        xi(g+1) ← zi
     else
        xi(g+1) ← xi(g)
     end if
   end for
  g ← g+1
  Compute{ f(x1(g)),...,f(xNP(g))}
  end while
```

Fig. 1. Description of DE algorithm

2 Related Works

Menzura-Montes et. al. [6] empirically compared the performance of eight DE variants. He used convergence measure to identify the competitiveness of the variants. The study concluded *rand/1/bin, best/1/bin, current-to-rand/1/bin* and *rand/2/dir* as the most competitive variants. However, the potential variants like *best/2/*, rand-to-best/1/** and *rand/2/** were not considered in their study.

Daniela Zaharie [7] provides theoretical insights on explorative power of Differential Evolutional algorithms, she describes an expression as a measure of the explorative power. In her results, the evolution of population variance for *rand/1/bin* is measured for two test functions. Control of diversity and associated parameter tuning are discussed in [8, 9].

Hans-Georg Beyer [10] analyzed how the ES/EP-like algorithms perform the evolutionary search in the real-valued N-dimensional spaces. He described the search behavior as the antagonism of exploitation and exploration.

3 Design of Experiments

In this paper, we investigate the convergence nature of DE variants and compare them by implementing on a set of benchmark. We have chosen fourteen test functions [6, 11], of dimensionality 30, grouped by the feature - unimodal separable, unimodal nonseparable, multimodal separable and multimodal nonseparable. The details of the benchmark problems are described in Table 1.

All the test functions have an optimum value at zero except *f08*. In order to show the similar results, the description of *f08* is adjusted to have its optimum value at zero by just adding the value (12569.486618164879) [6]. The parameters for all the DE variants were: population size NP = 60 and maximum number of generations = 3000. The variants will stop before the maximum number of generations is reached only if the tolerance error (which has been fixed as an error value of 1 x 10^{-12}) is obtained. Following [6, 12], we defined a range for the scaling factor, F ∈ [0.3, 0.9]. We use the same value for K as F.

Table 1. Details of the test functions used in the experiment

Functions and Ranges	Functions and Ranges
$f01: f_{Sp}(x) = \sum_{i=1}^{30} x_i^2$; $-100 \le x_i \le 100$	$f08: f_{Sch}(x) = \sum_{i=1}^{30}(x_i \sin(\sqrt{\lvert x_i \rvert}))$; $-500 \le x_i \le 500$
$f02: f_{Sch}(x) = \sum_{i=1}^{30}\lvert x_i\rvert + \prod_{i=1}^{30}\lvert x_i\rvert$; $-10 \le x_i \le 10$	$f09: f_{Ras}(x) = \sum_{i=1}^{30}[x_i^2 - 10\cos(2\pi x_i) + 10]$; $-5.12 \le x_i \le 5.12$
$f03: f_{schDS}(x) = \sum_{i=1}^{30}\left(\sum_{j=1}^{i} x_j\right)^2$; $-100 \le x_i \le 100$;	$f10: f_{Ack}(x) = 20 + e - 20exp\left(-0.2\sqrt{\frac{1}{p}\sum_{i=2}^{p} x_i^2}\right) - exp\left(\frac{1}{p}\sum_{i=1}^{p}\cos(2\pi x_i)\right)$; $-30 \le x_i \le 30$
$f04: f_{sch}(x) = max_i\{\lvert x_i\rvert, 1 \le i \le 30\}$; $-100 \le x_i \le 100$	$f11: f_{Gri}(x) = \frac{1}{4000}\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$; $-600 \le x_i \le 600$
$f05: f_{Ros}(x) = \sum_{i=1}^{29}\lvert 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\rvert$; $-30 \le x_i \le 30$	$f12: f_{GPF12} = \frac{\pi}{30}\{10sin^2(\pi y_1) + \sum_{i=1}^{30}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{30} u(x_i, 10,100,4)$; $-50 \le x_i \le 50$
$f06: f_{St}(x) = \sum_{i=1}^{30}(\lfloor x_i + 0.5\rfloor)^2$; $-1.28 \le x_i \le 1.28$	$f13: f_{GPF13} = 0.1\{sin^2(\pi 3x_1) + \sum_{i=1}^{29}(x_i - 1)^2[1 + sin^2(3\pi x y_{i+1})] + (x_n - 1)^2\}[1 + sin^2(2\pi x_{30})] + \sum_{i=1}^{30} u(x_i, 10,100,4)$; $-50 \le x_i \le 50$
$f07: f_{QF}(x) = \sum_{i=1}^{30} ix_i^4 + randon[0,1)$; $-1.28 \le x_i \le 1.28$	$f14: f_{Boh} = x_i^2 + 2x_{i+1}^2 - 0.3cos(3\pi x_i) - 0.4cos(4\pi x_{i+1}) + 0.7$; $-100 \le x_i \le 100$

The crossover rate, CR, was tuned for each variant-test function combination. Eleven different values for the CR viz. {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0} were used to conduct a bootstrap test, in order to determine the confidence interval for the mean objective function value. The CR value corresponding to the best confidence interval was chosen to be used in our experiment. The fourteen variants of DE along with the CR values for each test function are presented in Table 2.

As EA's are stochastic in nature, 100 independent runs were performed per variant per test function. For the sake of performance analysis among the variants, we present the mean objective function values (MOV), Convergence Speed [6], Quality Measure (Q-Measure) [13] and Probability of Convergence Measure (P-Measure) [13] for each variant-test function combination.

Convergence Speed is used to detect which variant is most competitive. To measure the convergence speed, we calculated the mean percentage out of the total 1,80,000 function evaluations required by each of the variant to reach its best objective function value, for all the 100 independent runs.

Table 2. CR Value Measured for Each Pair of Variant-Function

Sno	Variant	*f01/f08*	*f02/f09*	*f03/f10*	*f04/f11*	*f05/f12*	*f06/f13*	*f07/f14*
1	*rand/1/bin*	0.9/0.5	0.2/0.1	0.9/0.9	0.5/0.1	0.9/0.1	0.2/0.1	0.8/0.1
2	*rand/1/exp*	0.9/0.0	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9
3	*best/1/bin*	0.1/0.1	0.1/0.1	0.5/0.1	0.2/0.1	0.8/0.3	0.1/0.8	0.7/0.1
4	*best/1/exp*	0.9/0.7	0.8/0.9	0.9/0.8	0.9/0.8	0.8/0.9	0.8/0.8	0.9/0.8
5	*rand/2/bin*	0.3/0.2	0.1/0.1	0.9/0.1	0.2/0.1	0.9/0.1	0.2/0.1	0.9/0.1
6	*rand/2/exp*	0.9/0.3	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9
7	*best/2/bin*	0.1/0.7	0.3/0.1	0.7/0.4	0.2/0.1	0.6/0.1	0.1/0.1	0.5/0.1
8	*best/2/exp*	0.9/0.3	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9
9	*current-to-rand/1/bin*	0.5/0.4	0.1/0.1	0.9/0.1	0.2/0.1	0.1/0.2	0.1/0.3	0.2/0.1
10	*current-to-rand/1/exp*	0.9/0.3	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9
11	*current-to-best/1/bin*	0.2/0.8	0.1/0.1	0.9/0.1	0.2/0.2	0.1/0.2	0.3/0.1	0.2/0.1
12	*current-to-best/1/exp*	0.9/0.1	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9
13	*rand-to-best/1/bin*	0.1/0.8	0.1/0.1	0.9/0.9	0.4/0.1	0.8/0.1	0.4/0.2	0.8/0.1
14	*rand-to-best/1/exp*	0.9/0.4	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9

We used Q-Measure and P-Measure to study the convergence nature of DE Variants. Quality measure or simply Q-Measure is an empirical measure of the algorithm's convergence. In our experiment, it is used to study the behavior of our DE variants. The formula of Q-measure is $Q_m = C / P_c$ (where, P_c is Probability of convergence and C is Convergence Measure). The Convergence Measure(C) is calculated as C = $SumE_j / nc$ (Where, nc is number of successful runs and $SumE_j$ is total number of function evaluations taken for all the successful runs). The value for P_c is calculated as $P_c = nc/nt$ (where, nt is the total number of runs).

The convergence of a population is a measure that allows us to observe the convergence rate from the point of view of variables. The population convergence or simply P-measure (P_m) is a radius of a population. It can also be stated as the Euclidean distance between the center of a population and the individual farthest from it. The center of the population (O_p) is calculated as the average vector of all individuals, O_p = *Sum of Indi / NP*. The P-measure is calculated as $P_m = max \; || \; indi - Op \; || \; E$, for $i = 1,...,NP$.

4 Results and Discussion

The mean objective function values obtained for the unimodal separable functions: *f01, f02, f04, f06* and *f07,* and the unimodal nonseparable function *f03* are presented in Table 3. The results shows that best performance were provided by *rand-to-best/1/bin, rand/1/bin, best/2/bin* and *rand/2/bin* variants for the unimodal functions. *best/1/*, current-to-rand/1/exp* and *current-to-best/1/exp* were the poorly performing variants.

Table 3. MOV Obtained for Unimodal Functions

Variant	***f01***	***f02***	***f04***	***f06***	***f07***	***f03***
rand/1/bin	0.00	0.00	0.00	0.02	0.00	0.07
rand/1/exp	0.00	0.00	3.76	0.00	0.02	0.31
best/1/bin	457.25	0.14	1.96	437.25	0.09	13.27
best/1/exp	583.79	4.05	37.36	591.85	0.06	57.39
rand/2/bin	0.00	0.00	0.06	0.00	0.01	1.64
rand/2/exp	0.00	0.02	32.90	0.00	0.05	269.86
best/2/bin	0.00	0.00	0.00	0.07	0.00	0.00
best/2/exp	0.00	0.00	0.05	0.39	0.01	0.00
current-to-rand/1/bin	0.00	0.02	3.68	0.03	0.04	3210.36
current-to-rand/1/exp	24.29	44.22	57.52	43.07	0.27	3110.90
current-to-best/1/bin	0.00	0.02	3.71	0.00	0.04	3444.00
current-to-best/1/exp	24.37	45.04	56.67	41.95	0.26	2972.62
rand-to-best/1/bin	0.00	0.00	0.00	0.00	0.00	0.07
rand-to-best/1/exp	0.00	0.00	3.38	0.00	0.01	0.20

Table 4 displays the simulation results for the multimodal separable functions: *f08, f09* and *f14,* and for the multimodal non-separable functions *f05, f10, f11, f12* and *f13*. In case of multimodal separable problems, the best performance was shown by *rand/1/bin, rand-to-best/1/bin* and *rand/2/bin* variants once again as in the case of unimodal separable problems. Similarly, the variants *current-to-rand/1/exp* and *current-to-best/1/exp* consistently showed poor performance. In the case of multimodal nonseparable problems, function *f05* and *f10* were not solved by any variants. In this case of *f11, f12* and *f13*, the best performing variants were *rand-to-best/1/bin, rand/1/bin, rand/2/bin* and *best/2/bin*, once again along with *current-to-rand/1/bin* and *rand-to-best/1/bin*. *current-to-rand/1/exp* and *current-to-best/1/exp* were the poorly performing variants along with *best/1/** variants.

Table 4. MOV Obtained for Multimodal Functions

Variant	***f08***	***f09***	***f14***	***f05***	***f10***	***f11***	***f12***	***f13***
rand/1/bin	0.13	0.00	0.00	21.99	0.09	0.00	0.00	0.00
rand/1/exp	0.10	47.93	0.00	25.48	0.09	0.05	0.00	0.00
best/1/bin	0.00	4.33	12.93	585899.88	3.58	3.72	15.78	973097.03
best/1/exp	0.01	50.74	32.18	64543.84	6.09	5.91	131448.66	154434.94
rand/2/bin	0.22	0.00	0.00	19.01	0.09	0.00	0.00	0.00
rand/2/exp	0.27	101.38	0.01	2741.32	0.01	0.21	0.00	0.01
best/2/bin	0.17	0.69	0.12	2.32	0.09	0.00	0.00	0.00
best/2/exp	0.08	80.63	2.53	1.12	0.83	0.03	0.14	0.00
current-to-rand/1/bin	0.14	37.75	0.00	52.81	0.01	0.00	0.00	0.00
current-to-rand/1/exp	0.12	235.14	18.35	199243.32	13.83	1.21	10.89	24.11
current-to-best/1/bin	0.19	37.04	0.00	56.91	0.01	0.00	0.00	0.00
current-to-best/1/exp	0.10	232.80	18.21	119685.68	13.69	1.21	10.37	23.04
rand-to-best/1/bin	0.22	0.00	0.00	17.37	0.09	0.00	0.00	0.00
rand-to-best/1/exp	0.12	48.09	0.00	24.54	0.09	0.05	0.00	0.00

Based on the overall results in Table 3 and 4 the most competitive variants were *rand-to-best/1/bin, best/2/bin* and *rand/1/bin.* The variants *rand/2/bin* and *best/2/exp* also showed good performance consistently. On the other hand, the worst overall performance was consistently displayed by variants *current-to-best/1/exp* and *current-to-rand/1/exp*. It is worth noting that binomial recombination showed a better performance over the exponential recombination. Next in our experiment, we measured the Convergence Speed as the mean percentage of total number of function evaluations required by each of the variant, for 100 runs, to reach their best objective function value. Table 5 shows the Convergence Speed of the variants, ordered by function groups.

Table 5. Percentage of the total number of function evaluations taken by each variant for 100 runs (grouped by the function class). The lowest percentage is marked with "*".

Variant Function	f01	f02	f04	f06	f07	f03	f08	f09	f14	f05	f10	f11	f12	f13
1.DE/rand/1/bin	40.93*	56.45	100	10.89*	100	100	100	65.5	42.13	100	100	46.88	38.89*	41.38
2.DE/rand/1/exp	100	100	100	39.34	100	100	100	100	100	100	100	100	100	100
3.DE/best/1/bin	100	100	100	100	100	100	100	100	100	100	100	100	100	100
4.DE/best/1/exp	100	100	100	100	100	100	100	100	100	100	100	100	100	100
5.DE/rand/2/bin	70.05	90.59	100	17.78	100	100	100	100	59.84	100	100	70.02	56.07	58.62
6.DE/rand/2/exp	100	100	100	81.64	100	100	100	100	100	100	100	100	100	100
7.DE/best/2/bin	42.67	48.9*	100	12.47	100	100	100	87.29	48.6	99.97*	100	49.6	40.05	41.23
8.DE/best/2/exp	73.42	100	100	51.93	100	99.85*	100	100	98.35	100	100	88.94	80.43	81.42
9.DE/current-to-rand/1/bin	100	100	100	40.13	100	100	100	100	100	100	100	100	100	100
10.DE/current-to-rand/1/exp	100	100	100	100	100	100	100	100	100	100	100	100	100	100
11.DE/current-to-best/1/bin	100	100	100	50.84	100	100	100	100	100	100	100	100	100	100
12.DE/current-to-best/1/exp	100	100	100	100	100	100	100	100	100	100	100	100	100	100
13.DE/rand-to-best/1/bin	43.11	63.57	100	11.45	100	100	100	65.1*	42.09*	100	100	46.6*	38.94	38.34*
14.DE/rand-to-best/1/exp	100	100	100	39.11	100	100	100	100	100	100	100	100	100	100

The results show that for the unimodal functions the top four variants with the fastest convergence are *rand/1/bin, rand/2/bin, best/2/bin and rand-to-best/1/bin*. The variants *best/1/bin, best/1/exp, current-to-rand/1/exp* and *current-to-best/1/exp* are slow in convergence. For *f03, best/2/exp* is comparatively faster than other variants by 0.05%. For the multimodal functions variants with the fastest convergence are the same variants as in the case of unimodal functions. The results suggest that, the variants *rand-to-best/1/bin* and *rand/1/bin* are out performing others by its convergence speed and the variants *best/1/bin*, *best/1/exp*, *current-to-rand/1/exp* and *current-to-best/1/exp* are slow in convergence.

The values of Q_m are presented in the Table 6. For the unimodal functions, the top four variants with least Q_m values are *rand/1/bin*, *best/2/bin*, *rand-to-best/1/bin* and *rand/2/bin*. The variants *best/1/exp*, *current-to-rand/1/exp* and *current-to-best/1/exp* could not provide any successful run. This suggests the influence of binomial recombination in DE variants. In case of multimodal functions also the similar trend was observed.

The population convergence (P_m) is used as a measure to compare the convergence nature of the variants. We have done this experiment for all the Variant-Function combination. It is presented for *f01* and *f09* by the variants *rand/1/bin* and *best/1/bin*, because these are the two variants have different convergence behavior in our study. The results, Table 7, show that for the function *f01*, the variant *best/1/bin* gives both the best and worst run as unsuccessful one, due to its premature convergence. And the variant *rand/1/bin* gives both runs as successful run, because it could maintain the diversity in population. For the function *f03*, it is observed that the *rand/1/bin* reaches the global optimum, but *best/1/bin* could not converge to global optimum due to stagnation problem.

Table 6. Q-measure Measurement for the Variants

Variant Function	f01	f02	f03	f04	f05	f06	f07	f08	f09
1.DE/rand/1/bin	73686.6	10162080	13140000	18000000	-	1922580	10800000	720000	11790660
2.DE/rand/1/exp	18000000	18000000	720000	-	-	7081860	-	1260000	-
3.DE/best/1/bin	540000	7200000	15480000	14220000	-	-	3960000	15840000	540000
4.DE/best/1/exp	-	-	10440000	-	-	-	-	15300000	-
5.DE/rand/2/bin	12608200	16305480	-	-	-	3201060	360000	180000	18000000
6.DE/rand/2/exp	10980000	-	-	-	-	14695800	-	360000	-
7.DE/best/2/bin	7679760	8819160	18000000	18000000	6835320	2135280	13500000	180000	6171360
8.DE/best/2/exp	13214760	18000000	18000000	180000	5220000	3767640	360000	3060000	-
9.DE/current-to-rand/1/bin	18000000	-	-	-	-	7222800	-	360000	-
10.DE/current-to-rand/1/exp	-	-	-	-	-	-	-	540000	-
11.DE/current-to-best/1/bin	18000000	-	-	-	-	9150660	-	540000	-
12.DE/current-to-best/1/exp	-	-	-	-	-	-	-	900000	-
13.DE/rand-to-best/1/bin	7759500	11442000	14220000	18000000	-	2061060	10800000	-	11718600
14.DE/rand-to-best/1/exp	18000000	-	1800000	-	-	7039200	-	1080000	-

Variant Function	f10	f11	f12	f13	f14	SumEj	C	Qm=C/Pc
1.DE/rand/1/bin	-	8437680	7000500	7448880	7584180	97080246.6	93797.34	1268.76
2.DE/rand/1/exp	-	12240000	18000000	18000000	18000000	111301860	163920.27	3379.8
3.DE/best/1/bin	-	180000	-	-	-	57960000	180000	7826.09
4.DE/best/1/exp	-	-	-	-	-	25740000	180000	17622.38
5.DE/rand/2/bin	-	12603420	10092120	10551840	10772040	94674160	117900.57	2055.55
6.DE/rand/2/exp	11520000	540000	18000000	9000000	4680000	81295800	200235.96	6904.69
7.DE/best/2/bin	-	8927340	7028940	7422060	6767700	111466920	106769.08	1431.77
8.DE/best/2/exp	-	5929380	8896560	9076380	956880	86661600	142770.35	3292.89
9.DE/current-to-rand/1/bin	10080000	17280000	17280000	18000000	18000000	106222800	163419.69	3519.81
10.DE/current-to-rand/1/exp	-	-	-	-	-	540000	180000	840000
11.DE/current-to-best/1/bin	10080000	17280000	16380000	18000000	18000000	107430660	166301.33	3604.05
12.DE/current-to-best/1/exp	-	-	-	-	-	900000	180000	504000
13.DE/rand-to-best/1/bin	-	8388600	7008720	6901200	7576620	105876300	101902.12	1373.08
14.DE/rand-to-best/1/exp	-	12420000	18000000	18000000	18000000	94339200	161263.59	3859.3

Table 7. P_m Measurement for *f01*

DE/rand/1/bin							DE/best/1/bin						
Best Run – Successful Run							Best Run – Unsuccessful Run						
Run	G	ObjValue	Mean	Stddev	Variance	Pm	Run	G	ObjValue	Mean	Stddev	Variance	Pm
9	0	57466.93	98460.94	18822.72	3233.92	396.43	8	0	63579.26	97917.25	16186.32	3210.66	361.25
9	100	4319.44	6929.9	1255.32	194.56	100.58	8	100	733.05	1594.22	639.62	46.72	60.21
9	200	115.76	170.93	30.98	4.14	14.35	8	200	6.76	13.13	4.36	0.38	4.73
9	300	1.47	2.48	0.4	0.06	1.82	8	292	0.09	0.17	0.07	0	0.64
9	369	0.13	0.19	0.03	0	0.49	8	351	0.02	0.02	0	0	0.15
9	453	0	0.01	0	0	0.1	8	510	0.01	0.01	0	0	0
9	601	0	0	0	0	0	8	600	0.01	0.01	0	0	0
9	900	0	0	0	0	0	8	900	0.01	0.01	0	0	0
9	1084	0	0	0	0	0	8	1200	0.01	0.01	0	0	0
							8	1500	0.01	0.01	0	0	0
							8	1800	0.01	0.01	0	0	0
							8	2100	0.01	0.01	0	0	0
							8	2400	0.01	0.01	0	0	0
							8	2999	0.01	0.01	0	0	0
Worst Run – Successful Run							Worst Run – Unsuccessful Run						
Run	G	ObjValue	Mean	Stddev	Variance	Pm	Run	G	ObjValue	Mean	Stddev	Variance	Pm
34	0	63161.05	97475.45	18002.08	3184.79	378.27	39	0	56650.36	98206.15	17038.53	3233.84	367.82
34	100	2740.98	4292.98	538.96	105.59	70.93	39	100	3338.62	4554.26	1038.42	61.38	76.01
34	200	134.65	188.87	29.6	3.41	13.96	39	200	2523.86	2536.92	7.8	0.58	6.58
34	300	7.16	8.99	0.93	0.12	2.51	39	300	2517.97	2518.11	0.08	0.01	0.68
34	417	0.2	0.27	0.04	0	0.56	39	313	2517.9	2517.99	0.05	0	0.54
34	565	0	0.01	0	0	0.09	39	365	2517.84	2517.84	0	0	0.17
34	700	0	0	0	0	0.02	39	534	2517.83	2517.83	0	0	0
34	770	0	0	0	0	0	39	900	2517.83	2517.83	0	0	0
34	900	0	0	0	0	0	39	1200	2517.83	2517.83	0	0	0
34	1200	0	0	0	0	0	39	1500	2517.83	2517.83	0	0	0
34	1514	0	0	0	0	0	39	1800	2517.83	2517.83	0	0	0
							39	2100	2517.83	2517.83	0	0	0
							39	2400	2517.83	2517.83	0	0	0
							39	2700	2517.83	2517.83	0	0	0
							39	2999	2517.83	2517.83	0	0	0

For the function *f09*, Table 8, both the variants could reach the global optimum at their best run, but *rand/1/bin* is faster in convergence, it reaches with less number of function evaluations. And the variant *rand/1bin* could reach the global optimum in its worst run also. But the variant *best/1/bin* falls in premature convergence in its worst run. Similarly for the function *f05* the variant *best/1/bin* falls in premature convergence, even at its best run. But the variant "DE/rand/1/bin" could reach the global optimum in its best run.

The results suggest that the variant *rand/1/bin* is good in convergence. It is observed from various results that, even though selection decreases the population variance the variation operators of *rand/1/bin* could balance it by exploring the population in all the directions. This is due to the randomness property of the variant, which is used for selecting the candidates for perturbation. On the other hand, the variant *best/1/bin* falls in premature convergence (in most of the cases), the identified reason for this behavior is that the variation operators of this variant is not balancing the effect made by the selection operator in the population diversity.

In *best/1/bin,* the candidates for mutation operation is selected only in the direction from the best candidate of the current population, this makes the exploration operation to lose its power, and subsequently vanishes the population diversity soon. This effect leads to premature convergence.

Table 8. P_m Measurement for *f09*

DE/rand/1/bin							DE/best/1/bin						
Best Run – Successful Run							Best Run – Successful Run						
Run	G	ObjValue	Mean	Stddev	Variance	Pm	Run	G	ObjValue	Mean	Stddev	Variance	Pm
37	0	390.49	555.73	59.33	8.46	18.6	83	0	437.19	569.66	60.82	8.98	19.03
37	100	194.04	262.05	37.1	3.08	12.61	83	100	150.8	199.89	23.28	2.06	10.47
37	230	104.07	155.73	22.19	1.7	9.51	83	200	78.07	120.79	16.11	1.23	8.85
37	310	82.5	114.17	16.91	1.33	8.24	83	300	52.02	77.61	13.74	0.81	6.45
37	600	40.48	51.78	8.99	0.65	6.14	83	576	8.31	14.64	3.44	0.16	3.36
37	900	4.91	9.11	2.4	0.09	2.49	83	600	7.51	10.65	2.54	0.12	3.19
37	971	0.47	1.58	0.64	0	1.36	83	682	0.25	0.71	0.45	0	1.01
37	1159	0	0	0	0	0.01	83	769	0	0.01	0	0	0.01
37	1133	0	0.01	0	0	0.01	83	900	0	0	0	0	0
37	1330	0	0	0	0	0	83	1240	0	0	0	0	0
37	1500	0	0	0	0	0	83	1510	0	0	0	0	0
37	1800	0	0	0	0	0	83	1809	0	0	0	0	0
37	1848	0	0	0	0	0	83	2114	0	0	0	0	0
							83	2500	0	0	0	0	0
							83	2800	0	0	0	0	0
							83	2999	0	0	0	0	0
Worst Run – Successful Run							Worst Run – Unsuccessful Run						
Run	G	ObjValue	Mean	Stddev	Variance	Pm	Run	G	ObjValue	Mean	Stddev	Variance	Pm
95	0	433.86	551.58	55.59	8.64	19.29	48	0	394.12	557.07	61.43	8.54	19.83
95	100	189.82	248.56	29.82	2.72	11.67	48	100	160.48	209.33	23.53	1.77	10.1
95	230	116.94	159.46	22.07	1.61	8.66	48	200	83.03	132.93	20.39	1.15	7.22
95	310	104.86	134.92	19.56	1.38	8.24	48	300	60.94	89.86	13.03	0.7	6.11
95	600	47	67.92	11.5	0.83	6.56	48	576	15.77	16.51	0.57	0	1.04
95	900	20.82	32.76	7.02	0.45	4.74	48	600	15.53	15.67	0.19	0	1
95	971	14.81	24.89	5.1	0.35	4.37	48	682	15.37	15.37	0	0	0.01
95	1159	0.69	1.85	0.62	0	1.04	48	769	15.36	15.36	0	0	0
95	1133	2.55	4.12	1.21	0.03	1.71	48	900	15.36	15.36	0	0	0
95	1330	0	0.01	0	0	0.01	48	1240	15.36	15.36	0	0	0
95	1500	0	0	0	0	0	48	1510	15.36	15.36	0	0	0
95	1800	0	0	0	0	0	48	1809	15.36	15.36	0	0	0
95	2083	0	0	0	0	0	48	2114	15.36	15.36	0	0	0
							48	2500	15.36	15.36	0	0	0
							48	2800	15.36	15.36	0	0	0
							48	2999	15.36	15.36	0	0	0

Based on the overall results the most competitive variants were *rand/1/bin*, *rand/2/bin*, *best/2/bin* and *rand-to-best/1/bin*, they all could provide global optimum in all the function classes. On the other hand the worst results were provided by the variants *best/1/bin, best/1/exp, current-to-best/1/exp and current-to-rand/1/exp*. The variant *best/2/exp* could provide good result only for the function *f03*. Finally, the other five variants *rand/1/exp, rand/2/exp, rand-to-best/1/exp, current-to-best/1/bin* and *current-to-rand/1/bin* were continuously shown different performances.

5 Conclusion

In this paper, we presented various empirical measurements on convergence property of DE variants. Comparison of fourteen DE variants to solve fourteen global optimization problems is done. Regardless of the characteristics and dimension of the functions, relatively better results seem to have been provided by the variants with binomial crossover. The reasons for poor performance of the other variants are identified as due to either stagnation or premature convergence, which are lead by improper balance between exploration and exploitation processes, this is evident in our results. This work can be still further analyzed by focusing on improving the performance of variants in the light of bringing balance between exploration and exploitation during the generations.

References

1. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95-012, ICSI (March 1995)
2. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic Strategy for Global Optimization and Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)
3. Price, K.V.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. Mc Graw-Hill, UK (1999)
4. Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: A practical Approach to Global Optimzation. Springer, Heidelberg (2005)
5. Vesterstrom, J., Thomsen, R.: A Comparative Study of Differential Evolution Particle Swarm Optimization and Evolutionary Algorithm on Numerical Benchmark Problems. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2004), vol. 3, pp. 1980–1987 (2004)
6. Mezura-Montes, E., Velazquez-Reyes, J., Coello Coello, C.A.: A Comparative Study on Differential Evolution Variants for Global Optimization. In: GECCO 2006 (2006)
7. Zaharie, D.: On the Explorative Power of Differential Evolution Algorithms. In: 3rd Int. Workshop Symbolic and Numeric Algorithms of Scientific Computing-SYNASC 2001, Romania (2001)
8. Zaharie, D.: Critical values for the control parameters of Differential Evolution algorithms. In: Proc. of the 8th International Conference of Soft Computing, pp. 62–67 (2002)
9. Zaharie, D.: Control of Population Diversity and Adaptation in Differential Evolution Algorithms. In: Matouek, R., Omera, P. (eds.) Proceedings of Mendel, Ninth International Conference on Soft Computing, pp. 41–46 (2003)
10. Beyer, H.-G.: On the Explorative Power of ES/EP-like Algorithms. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 323–334. Springer, Heidelberg (1998)
11. Yao, H., Liu, Y., Lian, K.H., Lin, G.: Fast Evolutionary Algorithms. In: Rozenberg, G., Back, T., Eiben, A. (eds.) Advances in Evolutionary Computing Theory and Applications, pp. 45–94. Springer, New York (2003)
12. Personal Communication.: Unpublished
13. Feoktistov, V.: Differential Evolution In Search of Solutions. Springer, Heidelberg (2006)