Varying Variants for AncDE with MDV Between Target and Trial Vector (MDV) Measurement

Siti Khadijah Mohd Salleh[1], Diarmuid O'Donoghue[2], Abd Samad Shibghatullah[3] and Zuraida Abal Abas[3]
[1]*JTMK, Politeknik Ungku Omar, Malaysia*
[2]*Dept. Computer Science, Maynooth University, Ireland*
[3]*FTMK, Universiti Teknikal Malaysia Melaka, Malaysia.*
sitikhadijah@puo.edu.my

***Abstract*—This paper compares standard Differential Evolution algorithm with AncDE, which adds a separate cache of recent ancestors that serve as an additional source of high-quality genetic information. We compare the solutions produced by both DE and AncDE algorithms using benchmarks of 15 different numeric optimisation problems. Two distinct explorations are presented. The first test is distinct algorithmic variants of AncDE. The second part of this paper defines an MDV attribute and results are presented indicating some interesting differences in MDV between the DE and AncDE algorithms. Our findings indicate that ancestors can help to overcome some of the local variations in solutions quality and improve solution quality by improving population diversity.**

***Index Terms*—Different Vector; Ancestor Archive; Ancestor Usage Probability; Ancestor Replacement Probability; Trial Vector; Donor Vector.**

I. INTRODUCTION

Evolutionary algorithms are population-based optimisation systems that gradually converge towards some optimum solution. However, the search landscape may vary significantly in its structure and complexity across its surface, and this may cause the algorithm to be unable to reach a global optimum. Instinctively one way to overcome this drawback is by increasing the diversity [1] within the population. Evolutionary algorithms have adopted and explored the four (4) causes of improvement in natural evolution: selection, mutation, migration and genetic drift. However, Lolle *et al.* [2] and Hopkins et al. [3] controversially proposed the 5th cause of genetic improvements, that of ancestor-based genetic repair. Lolle's work has been successfully adapted for use in combinatorial optimisation in [4], while this paper builds on previous work [5] adapting the ancestral cache idea for numeric optimisation based on DE in the form of an ancestor driven extension to the DE algorithm. Thus, we used the ancestor vector as an archive population to extend the diversity contained in the main population.

This paper use enhances Differential Evolution DE (Storn & Price, 1997 [6]; Babu et al., 2016 [7]) called AncDE proposed by [8] with variant *AncDE/best/1/bin*. AncDE suggests using ancestor from the archive that stores the previous history of current population and the ancestor is selected randomly using two parameters to control both the age and the frequency of ancestral archive. Since AncDE has been introduced recently and merely been used therefore we would like to test AncDE with other variants over *CEC2015 Bound Constrained Single-Objective Computationally Expensive Numerical Optimization Problems* and record its performance over for further investigation. We also run another test to compute the distance between the target vector and donor vector for each generation in our propose calculation called magnitude difference between target and donor vector (MDV). We would like to scrutinise the AncDE landscape for each different variant and compare to MDV landscape of DE as a benchmark as well as to verify the impact of ancestor vectors upon the evolutionary process.

The paper is organised as follows: In Section 2 briefly describes the review of the related work as well as the different parameters and variants used in this work. The result and discussion obtained in Section 3. We draw conclusions and future work in Section 4.

II. RELATED WORKS

Mendel's Law of Inheritance stated three laws for an inheritance to occur; segregation, independent assortment and dominance [9]. There is four sources of genetic variation that is generally accepted in natural evolution: *natural selection, mutation, genetic drift* and *migration* and these have been widely adopted and emulated within evolutionary algorithms. Lolle et al. (2005) [2] Hopkins et al. (2013) [3] controversially proposed the 5th source of genetic variation, in the form of *ancestor-based genetic repair*, where she discovered *Arabidopsis* plant stores the genetic information and reappearing in the subsequent generation. Additionally, the genetic information can fix the current generation.

In this work, we explore different variants of an ancestrally driven algorithm. In regard to the archive algorithms performance in multi-objective optimisation as well to favour the diversity, there are four (4) causes of improvement in natural evolution: selection, mutation, migration and genetic drift. Building on previous work [4] and [5], this paper attempts to improve both quality and reliability of the results produced by an ancestor driven extension to the DE algorithm, by comparing variants of AncDE and verify which variants would perform better.

III. STANDARD DIFFERENTIAL EVOLUTION

Differential Evolution (DE) [6] [7] has become extremely popular because of its efficiency and simple implementation and has been shown to be one of the most reliable algorithms in dealing with optimisation problems [10]. In standard DE with *best/1/bin*, the best vector is selected from initial population G = {*x1, x2,...,x*|*G*|}. Then the other two distinct vectors ($x_{r1,G}$) and ($x_{r2,G}$) are selected from the population G and calculate the difference vector between them. This different vector then multiplies with *F*; a mutation factor in [0, 1] that controls the extension of differential variation, then added with the best vector to produce *donor vector* ($v_{i,G}$) in Equation (1). Crossover phase produces a trial vector using binomial crossover in Equation (2).

$$v_{i,G} = x_{best,G} + F\left(x_{r1,G} - x_{r2,G}\right) \tag{1}$$

$$u_{j,i,G} = \begin{cases} u_{j,i,G}, & \text{if } \left(rand_j[0,1] \le CR\right) or\ (j = j_{rand}) \\ x_{j,i,G}, & otherwise \end{cases} \tag{2}$$

CR is crossover constant in the range [0,1] while j_{rand} in randomly chosen integer in the range [1, D], where D is the size of dimension to ensure the trial vector $U_{i,G}$ differs from related target vector $X_{i,G}$ at least in one dimension. The last stage is selection phase in Equation (3), and all stage repeated until it reaches the stopping criteria [6]:

$$X_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } f\left(U_{i,G}\right) \le f\left(X_{i,G}\right) \\ x_{i,G}, & otherwise \end{cases} \tag{3}$$

One of the benefits of difference vectors is that the magnitude of moves decreases as the population converges to an optimum, allowing a more fine-grained search to occur [11].

IV. ANCESTRAL DIFFERENTIAL EVOLUTION - ANCDE

In this section, we describe previous work on AncDE, using a version of the algorithm called AncDE_trial. We shall describe this version in detail in the first section, and then we shall introduce three new variants on the AncDE algorithm.

AncDE proposed the following extension to the standard Differential Evolution: first, a second "shadow population" (cache) of recently discarded solution vectors is created and updated stochastically along with the new trial vectors to enter the population. In this paper, we explore a cache that is equal in size to the main population and where each cached solution is the ancestor of the corresponding solution in the main population. The second modification concerns the use of the ancestral cache when generating donor vectors. AncDE allows the use of both current population and the ancestral cache to generate difference vectors. The main attractive quality of the cache is that it offers (reasonably) high-quality solutions to broaden the search space, overcoming the limitation of alternate techniques such as the use of random mutations. Two parameters have been introduced by AncDE to control the ancestral cache. The first parameter is the ancestor replacement probability (arp) to control the age of ancestral archive, and the value is between [0, 1]. If the arp value is too low, this will cause the ancestor archive unrelated to the current population. However, the higher value of arp will make the ancestor archive too similar to the current population. The second parameter is ancestor usage probability (aup) to moderate the frequency of an ancestor, and the value is between [0,1]. If the aup = 0, the ancestral difference vector will not be applied and reduce the possibility of having an ancestral template in the next population. However, aup value that closes to 1 will increase the impact of archive vector on the current population.

A. *AncDE_trial*

The first version of AncDE is based on *DE/best/1/bin*, but the techniques can be easily applied to most variants of DE. At mutation stage, AncDE will select a trial vector $\vec{V_i}$ from the current population, and using ancestor usage probability (aup) to select ancestral vector $A(\vec{X_i})$. AncDE has proposed a new ancestral difference vector, which differ from DE to calculate the difference vector when random value *j* is smaller than aup. AncDE will retain normal difference vector when the random value *j* is bigger than aup, as shown in Equation (4). We call this version as AncDE_trial [8]:

$$\vec{V}_{,i,G} = \begin{cases} \vec{V_i} + \mathrm{F}(A(\vec{X_i}) - \vec{V_i}), & \text{if } \left(rand_j[0,1] < aup\right) \\ \vec{X}_{best,j} + \mathrm{F}(\vec{X}_{r1,j} - \vec{X}_{r2,j}), & otherwise \end{cases} \tag{4}$$

B. *AncDE_best*

In this section, we introduced AncDE with a new variant called AncDE_best where we will select best vector ($\vec{X}_{best,j}$) from the current population. We calculated the difference vector by having ancestor vector ($A(\vec{X_i})$) to subtract with best vector ($\vec{X}_{best,j}$) and multiply it with F [0,1] if $rand_j[0,1]$ is less than aup value. Otherwise, we will apply the standard DE mutation in Equation (5). We believe by applying best vector from the current population may give positive impact over overall performance for AncDE.

$$\vec{V}_{,i,G} = \begin{cases} \vec{X}_{best,j} + \mathrm{F}(A(\vec{X_i}) - \vec{X}_{best,j}), & \text{if } \left(rand_j[0,1] < aup\right) \\ \vec{X}_{best,j} + \mathrm{F}(\vec{X}_{r1,j} - \vec{X}_{r2,j}), & otherwise \end{cases} \tag{5}$$

C. *AncDE_CTB1*

The second algorithmic variant that we introduce is AncDE_CTB1. This strategy is based upon the DE variant called current-to-best/2/bin, which has been shown as good convergence property [12] as shown in Equation (6).

$$\vec{V}_{,i,G} = \begin{cases} V_{i,G} + F(X_{best,G} - V_{i,G}) + F(A(X_i) - X_{best,G}), & \text{if } \left(rand_j[0,1] < aup\right) \\ V_{i,G} + F(X_{best,G} - X_{i,G}) + F(X_{r1,G} - X_{r2,G}), & otherwise \end{cases} \tag{6}$$

In this variant $V_{i,G}$ is the trial vector from current population, $X_{best,G}$ is the best vector from the current population, $AX_{i,G}$ is random ancestor vector, $X_{r1,G}$ and $X_{r2,G}$ are any two random vectors from current population. If the value of $rand_j[0,1]$ is bigger than aup value, we will apply the standard DE mutation.

D. *AncDE_CTB2*

The third variant that we introduce is AncDE_CTB2. We use current-to-best with ancestral vector if random value *j* is less than aup, else we will use standard *DE/best/1/bin* as shown in Equation (7):

$$\vec{V}_{,i,G} = \begin{cases} V_{i,G} + F(X_{best,G} - X_{r1,G}) + F(A(X_i) - V_{i,G}), & \text{if } \left(rand_j[0,1] < aup\right) \\ X_{best,G} + F(X_{r1,G} - X_{r2,G}), & otherwise \end{cases} \tag{7}$$

All the variants (*A*, *B*, *C*, and *D*) will follow that standard AncDE crossover and selection stage as stated in [8]. In crossover stage, for binomial crossover, we use the following formula with probability *p* = Cr, such that for each vector *i* in the population and each *j* is an element of a vector $\vec{X_i}$:

$$\vec{U}_{i,j} = \begin{cases} \vec{V}_{i,j} & if\ rand(0,1) \le Cr\ or\ \le i = j \\ \vec{X}_{i,j} & otherwise \end{cases} \tag{8}$$

AncDE uses the same selection and replacement process as stated in standard DE where the best solution will be added to the new population. However, AncDE had added additional step; if the new trial vector is better than the current trial vector and the random value *j* is lower than ancestor replacement probability (arp) then the ancestral archive will be updated, but if random value *j* is bigger than ancestor replacement probability (arp) then the ancestral archive will remain the same. All stages repeated until it reaches the stopping criteria.

V. MAGNITUDE DIFFERENCE VECTOR (MDV)

In the following results section, we shall compare the performance of DE with the variants of AncDE as described above. While our results focus on solution quality, we also introduce an additional quality that we use to monitor the progress of each algorithm. Watson in [13] gives a wide explanation on the fitness landscape. However, he did mention about researchers keep modifying and enhancing current algorithm without giving any explanation on why those algorithms work so well and under what conditions. Therefore, we introduce *magnitude of the difference vectors* (MDV) between the target vector and the trial vector for each generation. This is an effective measure of the steps eyes employed while the solutions traverse across the problem space towards the optimal values. Where *n* is a number of population, X^{t+1} is the target vector; X^t is a trial vector. This experiment focuses on the value of jump size between target vector and trial vector thus it represents the distance between current solution and new solution as in Equation (9).

$$\vec{U}_{i,j} = \begin{cases} \vec{V}_{i,j} & if\ rand(0,1) \leq Cr\ or\ \leq i = j \\ \vec{X}_{i,j} & otherwise \end{cases} \tag{9}$$

VI. RESULT AND DISCUSSION

This section compares the performances of different variants of AncDE: AncDE_trial. AncDE_best, AncDE_CTB1 and AncDE_CTB2 with standard DE in the first section. We also test over our proposed magnitude computation called MDV for all variants and DE in the second section.

For variants comparison, we use the following parameters setup where NP = 25, F = 0.6, CR = 0.6, Range = 0.75 with *aup* = 0.5 and *arp* value = 0.15 for each AncDE variant to compare with standard DE (NP = 55, F = 0.55, CR = 0.95, Range = 0.75) with *d* = 30. Parameters for AncDE and DE (NP, F, and CR) as suggested in [8]. We run the test using the benchmark of 15 CEC2015 *Bound Constrained Single-Objective Computationally Expensive Numerical Optimization Problems* [14] consist of unimodal for problem 1 and 2, simple multimodal for problem 3, 4 and 5, hybrid function for problem 6, 7, and 8, problem 9 to 15 are composition functions.

A. Different Variants Result

The overall performance for different variants in AncDE, AncDE_best has produced the best result compared to other variants as shown in Figure 1. Although AncDE_CTB1 surprisingly good for Problem 5, however as for overall result it only did a good performance for four problems (Problem 1, 5, 6 and 7). Meanwhile, AncDE_CTB2 satisfied the expectation by did perform well on 8 out of 15 problems (Problem 1, 3, 4, 5, 6, 7, 13 and 15). AncDE_CTB2 even beat AncDE_trial for problem 1, 4, 5, 6, 13 and 14.

We used Wilcoxon Signed-Rank Test to compare the results produced by each algorithm with a significant level of 0.05 and one-tailed test to support our result. The following analysis is based on a comparison between AncDE_best and the other AncDE variants, as we shall show that this is the best variant of these AncDE algorithms. From the statistic result, we found that AncDE_best significantly outperformed AncDE_CTB2 on 11 out of 15 problems (Problem 2, 3, 4, 7, 8, 9, 10, 11, 12, 14, 15). AncDE_best outperformed AncDE_CTB1 on 14 out of 15 problems exclude Problem 5. AncDE_best also outperformed AncDE_trial with 13 out 15 problems (Problem 1, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15).

For overall performance between AncDE_and DE, AncDE_best has produced a better result than DE on 13 out of 15 problems (Problem 1, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14 and 15). DE has produced good results for problem 2 and 10. Based on the overall result AncDE_best produced the best results on this collection of problems outperform the other four variants. In the next section, we re-visit DE and AncDE_best to analyse their behaviour further.

B. Magnitude Difference Vector (MDV) Results

The ancestral archive of AncDE is initialised with an exact copy of the initial population, and for low values of arp, it may take many generations before these values are surpassed by new fitter solutions. Thus, we may expect AncDE to be "slow" during early convergence – hampered by these low-quality solutions in the cache.

Analysis of the MDV results showed in Figure 2 highlights a common pattern observed in the DE algorithm in particular. Looking at the MDV results for problem 1, we see that the MDV value for the DE algorithm is close to zero for the first 50 (approximate) generations. Between generations 50 to 100, we found that this MDV value reached its maximum value (reaching a larger value than all AncDE variants), before gradually reducing. In contrast, the AncDE variants generally generate high jump as early as 30 generations and reduce gradually after that. This pattern of early quiescence in the MDV value for DE is repeated across 12 distinct problems, numbered: 1, 2, 3, 6, 7, 8, 10, 11, 12, 13, 14, and 15. Of these, AncDE_best produced the best solutions on solution quality of these problems, numbered: 1, 3, 4, 6, 7, 8, 10, 11, 12, 14, and 15.

Thus, the hypothesis that we propose is an algorithm that generates high MDV value at the very early generation has high possibility to produce a good result. This hypothesis is supported by the fact that the MDV pattern and the best results for this problem set agree on problems: 1, 3, 4, 6, 7, 8, 10, 11, 12, 14, and 15. Based on the result of MDV, AncDE has fast convergence during the early generations, and we believe ancestral vector has caused the effect. It is obvious that by having the *arp* to control the age of ancestral vector did not slow the convergence process.

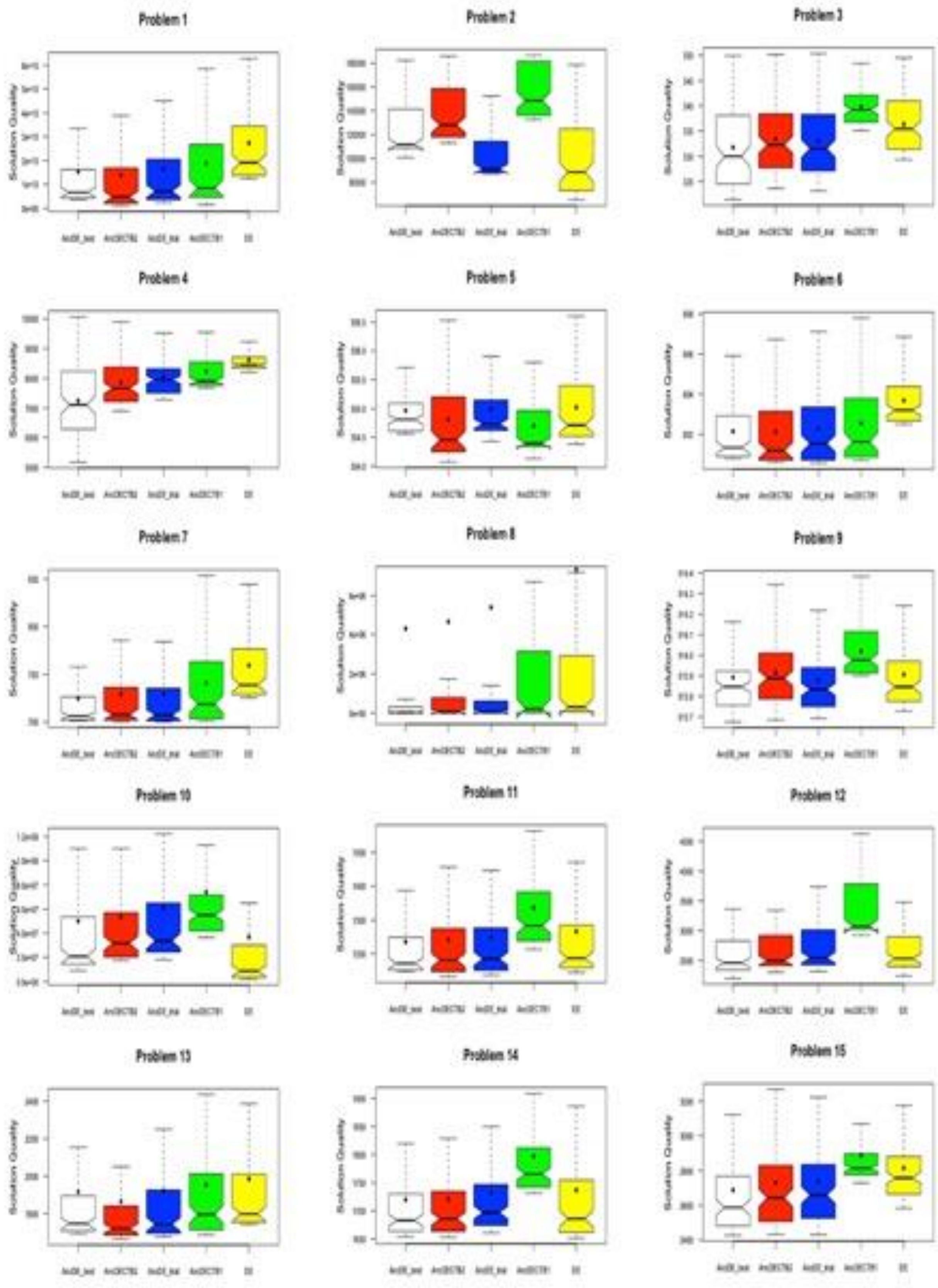

Figure 1: Solution Quality for AncDE_trial, AncDE_best, $AncDE_{CTB1}$, $AncDE_{CTB2}$ and DE

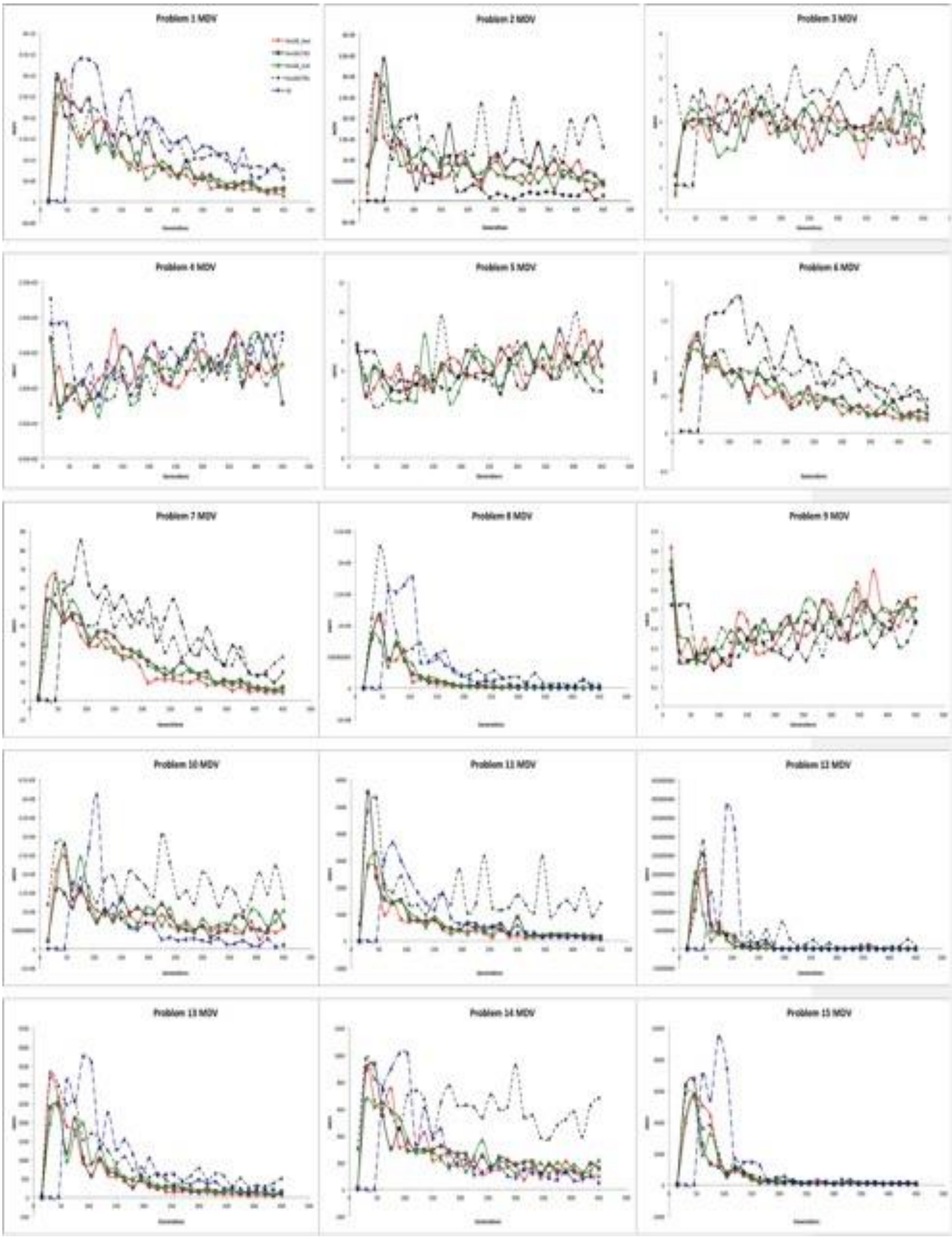

Figure 2: MDV for each AncDE variants and DE for first 450 generation

VII. Conclusion

This paper presents the evaluation of four versions of an ancestral extension to the standard DE algorithm: AncDE_trial, AncDE_best, AncDE_CTB1 and AncDE_CTB2. AncDE_trial is using the trial vector from current population, and AncDE_best is using the best vector from the current population. Both AncDE_CTB1 and AncDE_CTB2 are using the current-to-best method with a different approach. We also introduce magnitude computation called *Magnitude Difference Vector (MDV)* to calculate the distance between current solution and a new solution for every four variants and DE.

From the result, AncDE with the *best* vector has outperformed other variants for this particular problem with same parameters for each variant. However, since parameters value may affect the algorithm performance, therefore we would like to have a further investigation on AncDE_best (and the other algorithms) with dynamic *aup* and *arp* controller on different problems in the future.

For MDV result, we can conclude that the ancestral extension to DE, AncDE converged faster during the early generations than DE. We believe the ancestral vector may influence this convergence and then become an advantage to AncDE to produce a better result than DE in this problem. We would like to implement MDV in other algorithms to support our hypothesis.

Arising from our results, we propose the hypothesis algorithm that is able to generate high magnitude difference at the very early generation may have high possibility to perform better. This is because big magnitude size between current and new solution will allow the algorithm to converge faster and satisfy the exploration of search space.

Acknowledgement

We thank Mr. Donagh Hatton, Mr. Sawant Rushikesh and Computer Science Department, Maynooth University for help with AncDE procedures and resources.

References

[1] T. T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 6, pp. 1–24, 2012.

[2] S. J. Lolle, J. L. Victor, J. M. Young, and R. E. Pruitt, "inheritance of extra-genomic information in Arabidopsis," vol. 434, no. MARCH, pp. 505–509, 2005.

[3] M. T. Hopkins *et al.*, "De novo genetic variation revealed in somatic sectors of single Arabidopsis plants.," *F1000Research*, vol. 2, no. 0, p. 5, 2013.

[4] A. FitzGerald and D. P. O'Donoghue, "Genetic repair for optimization under constraints inspired by Arabidopsis thaliana," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5199 LNCS, pp. 399–408, 2008.

[5] D. Hatton and D. P. O. Donoghue, "Arabidopsis thaliana Inspired Genetic Restoration Strategies," *Int. J. Biometrics Bioinforma.*, vol. 7, no. 1, pp. 35–48, 2013.

[6] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, pp. 341–359, 1997.

[7] M. A. Babu *et al.*, "Application of Self Adaptive Differential Evolution Algorithm for Optimal Placement and Sizing of Renewable Dg Sources in Distribution Network Including Different Load."

[8] R. Sawant, D. Hatton, and D. P. O'Donoghue, "An ancestor based extension to Differential Evolution (AncDE) for Single-Objective Computationally Expensive Numerical Optimization," in *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings*, 2015, pp. 3228–3234.

[9] G. Mendel, "Experiments in Plant Hybridization," *J. R. Hortic. Soc.*, vol. IV, no. 1865, pp. 3–47, 1865.

[10] S. Das and P. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, 2011.

[11] T. Guide and T. S, *Natural Computing Series*. 2003.

[12] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," *IEEE Congr. Evol. Comput.*, pp. 1785–1791, 2005.

[13] J.-P. Watson, "An Introduction to Fitness Landscape Analysis and Cost Models for Local Search," *Handb. Metaheuristics*, vol. 146, no. International Series in Operations Research & Management Science, pp. 599–623, 2010.

[14] J. J. Liang, B. Y. Qu, P. N. Suganthan, and Q. Chen, "Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on Learning-based Real-Parameter Single Objective Optimization," *Tech. Report201411A, Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China Tech. Report, Nanyang Technol. Univ. Singapore*, no. November 2014, 2014.