Co-evolutionary Differential Evolution with Dynamic Population Size and Adaptive Migration Strategy

Zhi-hui Zhan and Jun Zhang (Corresponding Author)
Department of Computer Science, Sun Yat-sen University
Key Laboratory of Digital Life, Ministry of Education
Key Laboratory of Software Technology, Education Dept. of Guangdong Province, P.R. China
junzhang@ieee.org

ABSTRACT
As the performance of differential evolution (DE) is significantly affected by its mutation schemes and parameter settings when solving different problems, this paper proposes a simple yet efficient co-evolutionary DE (CEDE) to enhance the algorithm performance. The CEDE algorithm uses multiple populations to optimize the problem cooperatively, with each population using different operators and/or different parameters. Moreover, as different populations may show different performance on the same problem, we further design an efficient adaptive migration strategy (AMS) to dynamically control the population size of different populations. The CEDE algorithm is tested and compared on four benchmark functions. Experimental results demonstrate the good performance of CEDE when compared with conventional DEs using different operators and/or parameters.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods*; G.1.6 [**Numerical Analysis**]: Optimization – *Global optimization*.

General Terms
Algorithms, Performance, Reliability, Experimentation

Keywords
Differential evolution (DE); co-evolutionary algorithm; dynamic population size; adaptive migration strategy.

1. INTRODUCTION
Differential evolution (DE) was proposed by Price and Storn as a kind of evolutionary algorithm (EA) [1]. However, the performance of DE on different problems is significantly affected by its evolutionary operators, especially the mutation operator, and parameter settings, especially the '*crossover rate*' *CR*. Therefore, many researches have been conducted in the literature to enhance the DE performance by designing efficient mutation scheme or/and adaptively controlling the parameter settings [2]. However, using adaptive or self-adaptive strategies requires the users to understand the evolutionary process well so as to elaborately design good control strategies. In this paper, we propose to use a novel co-evolutionary DE (CEDE) with adaptive migration strategy (AMS) to address the operator choice and parameter settings problems in DE algorithm.

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

2. CEDE WITH AMS
There are M populations in CEDE, with each population using different mutation schemes (e.g., DE/best scheme or DE/rand scheme) and different parameter settings (e.g., CR=0.1 or CR=0.9). These M populations work cooperatively and each population optimizes the problem like a conventional DE. CEDE works as the following 6 steps.

Step 1: Initialization. For each population m, define its mutation scheme MS_m and two control parameters F_m and CR_m. For each individual i in the m^{th} population, initial its position $\boldsymbol{X}_{m,i}$ to a random value within the search space and then evaluate the individual's fitness $f(\boldsymbol{X}_{m,i})$. Find out the best individual among all the populations and its fitness is denoted as *Best*.

Step 2: Evolutionary process. In every generation, perform Step 3, Step 4, and Step 5 as follows:

Step 3: For each individual i in the m^{th} population, perform the following operations:

Step 3.1: Perform the mutation and crossover operations on $\boldsymbol{X}_{m,i}$ based on MS_m, F_m, and CR_m, so as to obtain a new position $\boldsymbol{U}_{m,i}$.

Step 3.2: Evaluate $f(\boldsymbol{U}_{m,i})$ and compare with $f(\boldsymbol{X}_{m,i})$. If $f(\boldsymbol{U}_{m,i})$ is better than $f(\boldsymbol{X}_{m,i})$, then $\boldsymbol{X}_{m,i}$ is set to $\boldsymbol{U}_{m,i}$. Otherwise, keeps $\boldsymbol{X}_{m,i}$.

Step 4: Find out the best individual among all the population and update the value of *Best*.

Step 5: Perform AMS among all the M populations.

AMS calculates the mean fitness value of each population and sorts the M populations according to the obtained mean fitness values from good to poor. All the poor populations will randomly migrate one individual to the good populations. For example, all the populations from rank 2 to rank M randomly migrate one individual to the 1st rank population, all the populations from rank 3 to rank M randomly migrate one individual to the 2nd rank population, and so on. However, AMS uses a parameter p_m as the '*migration rate*' to control the migration. Before the migration, a random real number r is first generated in the range [0, 1], if r is smaller than p_m, then the migration occurs. Otherwise, the migration does not occur. The p_m is non-linear time varying depended on evolutionary process as:

$$p_m = 0.01 + 0.99\frac{(\exp(\frac{10g}{G}) - 1)}{(\exp(10) - 1)} \quad (1)$$

where g and G are the current generation and the maximal generation respectively.

Step 6: Termination check. If the termination condition is met, then print the value of *Best* and CEDE terminates. Otherwise, CEDE goes to Step 2 for the next generation.

3. EXPERIMENTAL STUDIES

Four functions as listed in Table 1 are used for experiments. f_1 and f_2 are unimodal, f_3 and f_4 are multimodal. Moreover, f_1 and f_4 are with separate variables, f_2 and f_3 are with linkage variables [3][4].

Table 1. Test functions for comparisons

Test function	*Range*	f_{min}	*Name*
$f_1(x)=\sum_{i=1}^{D} x_i^2$	$[-100,100]^{30}$	0	Sphere
$f_2(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i} x_j)^2$	$[-100,100]^{30}$	0	Quadric
$f_3(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	$[-10,10]^{30}$	0	Rosenbrock
$f_4(x)=\sum_{i=1}^{D} -x_i \sin(\sqrt{\lvert x_i \rvert})+418.9829\times D$	$[-500,500]^{30}$	0	Schwefel

Four DE variants, denoted as DE/best/0.1, DE/best/0.9, DE/rand/0.1, and DE/rand/0.9 are adopted for comparisons. Here DE/best and DE/rand indicate using the best mutation scheme and the random mutation scheme, while 0.1 and 0.9 indicate that CR=0.1 and CR=0.9. For CEDE, we set M=4 and the above four DE variants are adopted by each population respectively. Therefore, CEDE uses both the greedy and random mutation schemes, and also uses both a small CR value and a large CR value. All the DEs are with F=0.5 and the population size is set to 50. In CEDE, each population is with size of 50 and is dynamically controlled by AMS during the evolutionary process.

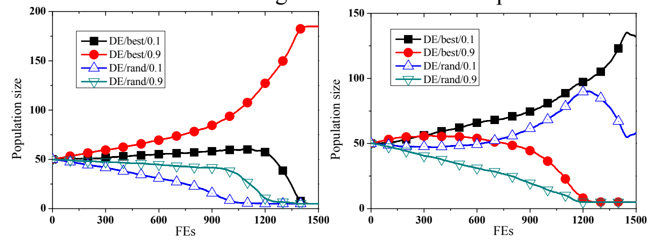
The maximal function evaluations (FEs) are 3.0×10^5 for each D=30 dimensions function. Each function is simulated 30 times independently and their mean results are used in the comparison. Moreover, Wilcoxon's rank sum tests with significant level α=0.05 are conducted to make the results statistically sound. The experimental results are presented in Table 2.

Table 2. Results comparisons with conventional DEs

Function		CEDE	DE/best/0.1	DE/best/0.9	DE/rand/0.1	DE/rand/0.9
f_1	mean	4.27E-139	3.24E-142–	**2.93E-292–**	3.26E-089+	9.04E-106+
	std	1.71E-138	8.94E-142	**0.00E+000**	3.25E-089	3.65E-105
f_2	mean	**8.03E-027**	1.15E+002+	1.67E+002+	2.33E+003+	9.28E-015+
	std	**2.00E-026**	4.98E+001	9.13E+002	4.79E+002	1.49E-014
f_3	mean	**5.96E-022**	3.04E+001+	1.20E+000+	2.36E+001+	1.72E+001+
	std	**3.25E-021**	1.97E+001	1.86E+000	1.59E+000	6.11E+000
f_4	mean	**3.82E-004**	5.50E+002+	4.40E+003+	**3.82E-004=**	2.50E+003+
	std	**0.00E+000**	2.28E+002	7.47E+002	**0.00E+000**	6.03E+002

'+', '=', and '–' mean that CEDE performs significantly better than, similar to, and significantly worse than the corresponding DE, respectively.

The comparisons show that the performance of conventional DEs on different problems is seriously affected by the operators and/or parameters. For example, DEs use greedy mutation schemes yields much better results than DE use random mutation schemes on unimodal functions. However, when solving multimodal functions, greedy mutation seems to be inefficient, e.g., DE/best/0.1 and DE/best/0.9 are totally trapped by the multimodal functions f_3 and f_4. Moreover, although using the same DE/best mutation scheme on f_1, DE/best/0.9 is remarkably better than DE/best/0.1. DE/rand/0.1 can obtain global optimum on f_4 but DE/rand/0.9 is totally trapped. Therefore, it seems no a single DE mutation scheme and/or a single parameter setting can perform well on different kinds of problems. As CEDE uses multiple populations with different operators and different parameters, it is expected that CEDE is suitable for different kinds of problems. The results show that CEDE can obtain the global optimum of all 4 functions in all the 30 independent runs, no matter on unimodal or multimodal functions, no matter on functions with variable linkage or with variable independent.



(a) f_1 (b) f_4

Figure 1. The dynamic population size during the search process when optimizing the function.

Fig. 1 plots the dynamic population size during the search process. Fig. 1(a) shows that the population size of DE/best/0.9 keeps increasing during the evolutionary process on f_1. This is consistent with the fact that DE/best mutation scheme and large CR value have fast convergence speed to optimize simple unimodal function. However, when optimizing multimodal function, e.g., f_4, as in Fig. 1(b), DE/best/0.9 performs very poor and therefore its population size keeps decreasing. Contrarily, the population sizes of DE/rand/0.1 and DE/best/0.1 increase during the evolutionary process. This is because that DE with small CR value can do well on separate multimodal function. Therefore, the figures show that AMS can catch the performance of different populations and let the individuals migrate to well-performed populations. This observation demonstrates the proposed AMS works well. Further study may test some other migration schemes as proposed by Zhong *et al*. [5] to evaluate the algorithm performance.

4. CONCLUSIONS

A CEDE with AMS is proposed. CEDE uses multiple populations to optimize the problem cooperatively, with each population using different operators and/or different parameters. When solving different kinds of problems, different populations may have different performance. Therefore, CEDE has promising performance on different problems. The experimental studies demonstrate this intuition and the good performance of CEDE.

5. ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (NSFC) No. 61070004, and by NSFC Joint Fund with Guangdong under Key Project U0835002. The authors are with the Key Lab of Digital Life, MoE, China, and the Key Lab. of Softw. Tech, Edu. Depart of Guangdong Province. The corresponding author is Jun Zhang, email: junzhang@ieee.org.

6. REFERENCES

[1] R. Storn and K. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[2] Z. H. Zhan and J. Zhang, "Self-adaptive differential evolution based on PSO learning strategy," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2010, pp. 39–46.

[3] Z. H. Zhan, J. Zhang, Y. Li, and H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. Syst., Man, and Cybern., B*, vol. 39, no. 6, pp. 1362-1381, Dec. 2009.

[4] Z. H. Zhan, J. Zhang, Y. Li, and Y. H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, (In press)

[5] J. H Zhong, J. Zhang, and Z. Fan, "MP-EDA: A robust estimation of distribution algorithm with multiple probabilistic models for global continuous optimization," *SEAL 2010, LNCS 6457*, pp. 85-94, 2010.