Adaptive Differential evolution with candidate mutant vector

line 1: 1st Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 2nd Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

line 1: 3rd Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

***Abstract*—This electronic document is a "live" template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. **CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)**

Keywords—component, formatting, style, styling, insert (*key words*)

I. Introduction

The differential evolution (DE) is one of the most popular optimization algorithms proposed by Storn and Price[xx] in 1995. The algorithm has mutation, crossover, and selection operations. In DE process, the population consists of several individuals which represents a potential solution to an optimization problem. [xx] During the process DE has three operations as above to develop the potential in the population and this algorithm is expected to be closer to the optimal solution. When generation increases, the diversity of the population is becoming worse because the individual is similar, and it makes premature convergence. To solve this problem many researchers are focused on control parameters and mutation strategies. *Literature review*

II. Standard Differential Evolution Algorithm

Differential evolution algorithm has divided into four processes such as initialization, mutation, crossover, and selection operation, in the evolutionary phase these four processes are used to evaluate fitness function $f(x)$, and the best individual is recorded.

A. Initialization

At the beginning iteration, an initial population must be generated through the search space range in each dimension *j*th (j = 1,2, 3…,D) of individual *i*th (i = 1,2,3,…,NP) the population can be generated as follows.

$$x^{0}_{i,j} = x^{L}_{i,j} + rand\left(x^{U}_{i,j} - x^{L}_{i,j}\right), \quad (1)$$

Where rand function return uniformly distributed random number. U and L represent upper bound and lower bound of solution space.

B. Mutation operation

The mutation strategy of the DE algorithm can be expressed by "DE/x/y" "x" representing the vector in mutation operation and "y" representing the number of differential vectors. In the original DE it used the mutation strategy "DE/rand/1" is a common mutation strategy. The DE chooses a random vector from the population with one differential vector from the random vector, to generate mutant vector V_i as follows.

$$V^{G+1}_{i} = X^{G}_{r1} + F \cdot \left(X^{G}_{r2} - X^{G}_{r3}\right), \quad (2)$$

Where *r1, r2,* and *r3* are permutation index random vectors and $r1 \neq r2 \neq r3$. F denotes the scaling factor in the range [0,1]

C. Crossover operation

After mutation operation, mutant vector V_i brings to crossover operation with target vector X_i to generate trial vector U_i. By crossover probability (Cr) in the range [0,1], and in original DE we use Cr is 0.8. The crossover operation can be expressed as follow:

$$U^{G+1}_{i,j} = \begin{cases} V^{G+1}_{i,j}, & rand < Cr \text{ or } j = j_{rand}, \\ X^{G}_{i,j}, & otherwise, \end{cases} \quad (3)$$

Where $X^{G}_{i,j}$, $V^{G+1}_{i,j}$ denotes *j*th component of the *i*th individual and mutant vector, with the uniform distribution we select $V^{G+1}_{i,j}$ when rand value is small or *j*th component is equal index random index *j*th and otherwise, we select $X^{G}_{i,j}$.

D. Selection operation

In the original DE we use a greedy selection strategy is utilized compare between trial vector U_i and target vector X_i, which one is better fitness we will select this vector as X^{G+1}_{i}, the selection operation can be expressed as follow:

$$X^{G+1}_{i,j} = \begin{cases} U^{G+1}_{i}, & f\left(U^{G+1}_{i}\right) < f\left(X^{G}_{i}\right), \\ X^{G}_{i}, & otherwise, \end{cases} \quad (4)$$

Where $f(\cdot)$ stands for the fitness value.

III. Differential Evolution with Candidate Mutant Vector

In DECM, the crossover and selection operation is the same as the basic DE, as shown in equations 3 and 4. In mutation operation, we use random vectors to generate a candidate mutant vector and select one of the best candidate mutant vectors for crossover operation.

A. Candidate mutant vector

From equation 2 and inspired by mutation operation in DSIDE we create a set of candidate mutant vector $V^{G+1}_{candidate}$ using the three random vectors as follow:

$$V_{cadidate}^{G+1} = \begin{cases} \alpha_i^G \cdot X_{r1}^G + F_i \cdot \left(X_{r2}^G + X_{r3}^G \right) \\ \alpha_i^G \cdot X_{r1}^G + F_i \cdot \left(X_{r3}^G + X_{r2}^G \right) \\ \alpha_i^G \cdot X_{r2}^G + F_i \cdot \left(X_{r1}^G + X_{r3}^G \right) \\ \alpha_i^G \cdot X_{r2}^G + F_i \cdot \left(X_{r3}^G + X_{r1}^G \right) \\ \alpha_i^G \cdot X_{r3}^G + F_i \cdot \left(X_{r1}^G + X_{r2}^G \right) \\ \alpha_i^G \cdot X_{r3}^G + F_i \cdot \left(X_{r2}^G + X_{r1}^G \right) \end{cases}, (5)$$

$$\alpha_i^G = 1 - r^{\left(1 - \frac{G}{G_{max}}\right)^2}, (6)$$

In equation 5 α_i^G, F_i, Cr_i are the reference factor, scaling factor, and crossover probability for each target individual, $V_{candidate}^{G+1}$ are candidate mutant vector set and we select the best mutant vector (select by fitness value) in the candidate vector set to mutant vector V_i^{G+1}. G represents the current generation and G_{max} represents the maximum generation of the algorithm. From equation 6 r denotes a random number on interval $r \in [0,1]$. At beginning of the evolution stage, the value of α_i^G is large it makes a wide range of searches, as generation increases, the α_i^G value decrease and the search range is shrinking.

B. *Adaptive Scaling factor and Crossover probability strategy*

Inspired by DSIDE this paper is use Scaling factor and Crossover probability strategy as follows:

$$F_i^G = \frac{f_{max}^G - f_i^G}{f_{mean}^G}, (7)$$

$$Cr_i^G = \frac{f_i^G - f_{min}^G}{f_{mean}^G}, (8)$$

Where f_i^G is individual X_i^G fitness value, f_{max}^G and f_{min}^G are maximum and minimum fitness value of the current generation, f_{mean}^G is the average fitness value of the population in the current generation. In equation 7, the f_i^G is as same as f_{max}^G it makes a smaller F_i^G and it helps X_i^G convergence rapidly and f_i^G is extremely smaller than f_{max}^G it makes a large F_i^G to contain the diversity of this individual but reduce the search efficiency. Furthermore from equation 8 the f_i^G as the same as f_{max}^G it makes Cr_i^G is large and increases the opportunity for crossover to find a new solution, when f_i^G as the same as f_{min}^G it makes a smaller Cr_i^G to contain this solution of individual X_i^G. The process of DECM is shown in Algorithm I.

Initialize the original population *pop* and calculate their fitness value, NP = 100, G = 1, G_{max} = 5000
while ($G \leq G_{max}$) **do**
  for each individual X_i in *pop* **do**
    Calculate α_i in equation (6):
    Calculate F_i in equation (7):
    Calculate Cr_i in equation (8):
    Implement mutation in equation (5):
    Implement crossover in equation (3):
    Implement selection in equation (4):
  end for
  G = G+1
end while

Algorithm I: DECM

IV. EXPERIMENTAL AND COMPARISON

To test the performance of the proposed algorithm, therefore benchmark functions are utilized to evaluate the performance of the algorithm. In this section, the performance of DECM is tested on 9 benchmark functions listed in Table I, where *D* is the dimension of the problem. $f_1 - f_5$ are unimodal functions and $f_6 - f_9$ are multimodal functions. *f (*)* denotes the global minimum value.

Experiment environment: Windows 11 home x64 Operating System of a PC with intel core i5-11300H CPU (4.40 GHz), and algorithm are implemented in Python 3.10.5 Windows version.

A. *Comparison with 4 Improved DE Algorithms.*

To verify the performance of the DECM, it is compared with four classic DE-improvement algorithms: JADE [xx], jDE [xx], CoDE [xx], DSIDE, and the proposed DECM algorithm. Where *D* = 30, population size = 100, maximum generation = 5000. The parameters of other algorithms are the same as in the original literature. The experimental result of all algorithms is shown in Table II, mean/std (mean value and standard deviation) of fitness value over 30 independent runs. Symbols “+/-/=” mean better than, worse than, and similar to DECM.

Based on the result in Table1 and Figure1 we can see…. *Experimental Result*

Name	Function	Range	$f(*)$
Sphere (f_1)	$\sum_{i=1}^{D} x_i^2$	$[-100,100]^D$	0
Elliptic (f_2)	$\sum_{i=1}^{D} (10^6)^{t-\frac{1}{D}-1} x_i^2$	$[-100,100]^D$	0
Schwefel1.2 (f_3)	$\sum_{i=1}^{D} (\sum_{j=1}^{i} x_i)^2$	$[-100,100]^D$	0
Schwefel2.22 (f_4)	$\sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} x_i^2$	$[-100,100]^D$	0
Zakharov (f_5)	$\sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5x_i)^2 + (\sum_{i=1}^{D} 0.5x_i)^4$	$[-100,100]^D$	0
HGBat (f_6)	$\left\lvert \left(\sum_{i=1}^{D} x_i^2\right)^2 - \left(\sum_{i=1}^{D} x_i\right)^2 \right\rvert^{1/2} + \frac{0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i}{\mathrm{D}} + 0.5$	$[-100,100]^D$	0
Scaffer2 (f_7)	$\sum_{i=1}^{D} \left(x_i^2 + x_{i+1}^2\right)^{0.25} (\sin(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1)$	$[-100,100]^D$	0
HappyCat (f_8)	$\left\lvert \sum_{i=1}^{D} x_i^2 - D \right\rvert^{1/4} + \frac{0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i}{\mathrm{D}} + 0.5$	$[-100,100]^D$	0
ScafferF6 (f_9)	$\sum_{i=1}^{D} (\left(0.5 + \sqrt{x_i^2 + x_{i+1}^2}\right)^2 - 0.5)/(1 + 0.001(x_i^2 + x_{i+1}^2))^2$	$[-0.5,0.5]^D$	0

TABLE I. BENCHMARK FUNCTIONS

F	JADE mean±std	jDE mean±std	CoDE mean±std	DSIDE mean±std	DECM mean±std
f_1					
f_2					
f_3					
F_4					
f_5					
f_6					
f_7					
f_8					
f_9					
+/=/-					

TABLE II. EXPERIMENTALRESULT

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.