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)

# 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\*

# 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 ƒ(*x*), and the best individual is recorded.

## X subscript i comma j end subscript superscript G plus 1 end superscript space equals space open curly brackets table row cell U subscript i superscript G plus 1 end superscript comma space f open parentheses U subscript i superscript G plus 1 end superscript close parentheses space less than space f open parentheses X subscript i superscript G close parentheses comma end cell row cell X subscript i superscript G comma space o t h e r w i s e comma end cell end table close space open parentheses 4 close parenthesesInitialization

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 subscript i comma j end subscript superscript 0 space equals space x subscript i comma j end subscript superscript L plus r a n d open parentheses x subscript i comma j end subscript superscript U minus x subscript i comma j end subscript superscript L close parentheses comma space space open parentheses 1 close parentheses

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

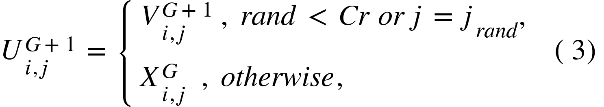
## Mutation operation

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 subscript i superscript G plus 1 end superscript space equals space X subscript r 1 end subscript superscript G plus F times open parentheses X subscript r 2 end subscript superscript G minus X subscript r 3 end subscript superscript G close parentheses comma space open parentheses 2 close parentheses

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

## Crossover operation

 After mutation operation, mutant vector Vi brings to crossover operation with target vector Xi to generate trial vector Ui. 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:

Where , denotes *j*th component of the *i*th individual and mutant vector, with the uniform distribution we select when rand value is small or *j*th component is equal index random index *j*th and otherwise, we select .

## Selection operation

In the original DE we use a greedy selection strategy is utilized compare between trial vector Ui and target vector Xi, which one is better fitness we will select this vector as , the selection operation can be expressed as follow:

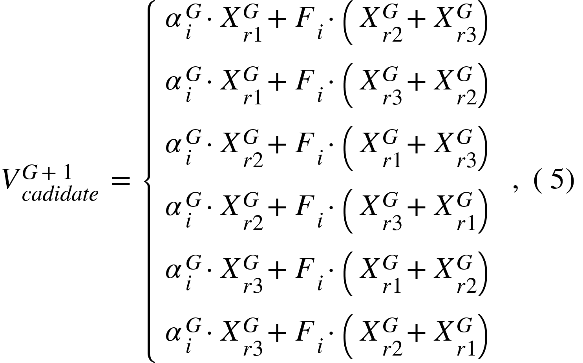
Where stands for the fitness value.

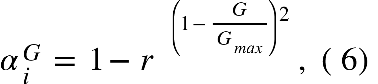
# Differential evolution with candidate mutant vector

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.

## Candidate mutant vector

From equation 2 and inspired by mutation operation in DSIDE we create a set of candidate mutant vector using the three random vectors as follow:





In equation 5 are the reference factor, scaling factor, and crossover probability for each target individual, are candidate mutant vector set and we select the best mutant vector (select by fitness value) in the candidate vector set to mutant vector . G represents the current generation and Gmax represents the maximum generation of the algorithm. From equation 6 *r* denotes a random number on interval . At beginning of the evolution stage, the value of is large it makes a wide range of searches, as generation increases, the value decrease and the search range is shrinking.

## Adaptive Scaling factor and Crossover probability strategy

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

F subscript i superscript G space equals space fraction numerator f subscript m a x end subscript superscript G minus f subscript i superscript G over denominator f subscript m e a n end subscript superscript G end fraction space comma open parentheses 7 close parentheses

C r subscript i superscript G space equals space fraction numerator f subscript i superscript G minus f subscript m i n end subscript superscript G over denominator f subscript m e a n end subscript superscript G end fraction space space comma space open parentheses 8 close parentheses

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

Initialize the original population *pop* and calculate their fitness value, NP = 100, G = 1, Gmax = 5000

**while** (G≤ Gmax) **do**

**for** each individual Xi in *pop* **do**

Calculate αi in equation (6):

Calculate *Fi* in equation (7):

Calculate *Cri* 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

# 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 – *f5* are unimodal functions and *f*6 – *f9* 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.

## 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 (*f1*) |  | [-100,100]*D* | 0 |
| Elliptic *(f2*) |  | [-100,100]*D* | 0 |
| Schwefel1.2 (*f3*) |  | [-100,100]*D* | 0 |
| Schwefel2.22 (*f4*) |  | [-100,100]*D* | 0 |
| Zakharov (*f5*) |  | [-100,100]*D* | 0 |
| HGBat (*f6*) |  | [-100,100]*D* | 0 |
| Scaffer2 (*f7*) |  | [-100,100]*D* | 0 |
| HappyCat (*f8*) |  | [-100,100]*D* | 0 |
| ScafferF6 (*f9*) |  | [-0.5,0.5]*D* | 0 |

1. Benchmark functions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *F* | JADE  mean±std | jDE  mean±std | CoDE  mean±std | DSIDE  mean±std | DECM  mean±std |
| *f1* |  |  |  |  |  |
| *f2* |  |  |  |  |  |
| *f3* |  |  |  |  |  |
| *F4* |  |  |  |  |  |
| *f5* |  |  |  |  |  |
| *f6* |  |  |  |  |  |
| *f7* |  |  |  |  |  |
| *f8* |  |  |  |  |  |
| *f9* |  |  |  |  |  |
| +/=/- |  |  |  |  |  |

1. Experimentalresult
2. 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)*
3. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
4. 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.
5. K. Elissa, “Title of paper if known,” unpublished.
6. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
7. 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].
8. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.