

# Differential evolution algorithm of soft island model based on *K*-means clustering

Xujie Tan<sup>1</sup>, Seong-Yoon Shin<sup>2</sup>

<sup>1</sup>School of Information Science and Technology, Jiujiang University, China

<sup>2</sup>School of Computer Information & Communication Eng., Kunsan National University, South Korea

---

## Article Info

### Article history:

Received Feb 10, 2020

Revised Mar 9, 2020

Accepted Apr 5, 2020

---

### Keywords:

Differential evolution

Evolutionary algorithm

K-means clustering

KSDE algorithm

Soft island model

---

## ABSTRACT

Differential evolution (DE) is a highly effective evolutionary algorithm. However, the performance of DE depends on strategies and control parameters. The combination of many strategies helps balance the exploitation and exploration of DE. In this study, a multi-population based on *k*-means clustering is proposed to realize an ensemble of multiple strategies, thereby resulting in a new DE variant, namely KSDE, where similar individuals in the population implement the same mutation strategies, and dissimilar subpopulations migrate information through the soft island model (SIM). Firstly, the population is virtually divided into *k* subpopulations by the *k*-means clustering algorithm. Secondly, the individual specific mutation scheme is selected from a strategy pool by random method. Finally, the migration of subpopulation information is done using the soft island model. The performance of the KSDE algorithm is evaluated on 13 benchmark problems. The experiments show that KSDE algorithm improves the performance of the DE algorithm.

Copyright © 2020 Institute of Advanced Engineering and Science.  
All rights reserved.

---

## Corresponding Author:

Seong-Yoon Shin,  
School of Computer Information & Communication Eng.  
Kunsan National University,  
Kunsan, 54150, South Korea.  
Email: s3397220@kunsan.ac.kr

---

## 1. INTRODUCTION

Differential evolution (DE) is a simple and efficient evolutionary algorithm (EA) for optimization problems proposed by Price and Storn [1]. Recently, DE has been widely used in diverse fields, such as pattern recognition [2], artificial neural networks [3], image processing [4], and electronics and communication engineering [5]. The DE performance mainly depends on its trial vector generation strategy (i.e., mutation and crossover operators) and its control parameters (i.e., population size *NP*, scaling factor *F*, and crossover control parameter *CR*). Appropriate trial vector generation strategies and control parameter settings are helpful for improving the performance of DE. Recently, the multi-island model is to maintain population diversity to improve the performance of the DE algorithm. The information exchange among islands can maintain the diversity of the entire population and balance the exploitation and exploration capabilities. The soft island model approach was applied to evolutionary algorithms to improve the performance of the algorithm [6].

Based on these considerations, a novel DE algorithm, namely the differential evolution algorithm of soft island model based on *k*-means clustering (KSDE), was proposed. In KSDE, the population is classified into a number of clusters by the *k*-means cluster algorithm. Consequently, a more suitable mutation strategy may be selected randomly to match different clusters. Finally, to improve the diversity of the population, the KSDE used the soft island model to migrate individuals. To evaluate the effectiveness of KSDE, the proposed KSDE framework was conducted on 13 benchmark functions with 30, 50, and 100 variables. Comprehensive experimental results indicate that KSDE is an effective and efficient DE variant.

The remainder of this paper is organized as follows. Section 2 introduces the classic DE algorithmic. Section 3 reviews the related works in literature. The proposed DE algorithm, namely KSDE, is presented in detail in Section 4. Section 5 gives the experimental results. The last section, Section 6, is devoted to conclusions and future work.

**2. DIFFERENTIAL EVOLUTION**

Differential evolution is used to solve the real number optimization problem. In this paper, the object function can be expressed as  $f(x)$ ,  $x = (x_1, x_2, \dots, x_D)$ , with  $D$  denoting the dimension of space. At first, the  $NP$  population  $x$  is randomly generated. Hence, each vector of the  $x_i$  in  $G$  generation can be generated by

$$x_{i,j} = L_i + rnd \cdot (U_i - L_i) \tag{1}$$

where  $rnd \in [0, 1]$  is a random number,  $x_{i,j} \in [L_i, U_i]$ .

**2.1. Mutation**

After initialization, a donor vector  $v_i$  is produced with respect to  $x_i$ . At the  $G$  generation,  $v_i$  can be generated through the following mutation strategies of (2)-(6).

a) DE/rand/1:

$$V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) \tag{2}$$

b) DE/best/1:

$$V_i = X_{best} + F_i \cdot (X_{r_2} - X_{r_3}) \tag{3}$$

c) DE/target-to-best/1:

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r_1,G} - X_{r_2,G}) \tag{4}$$

d) DE/best/2

$$V_{i,G} = X_{best,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) + F \cdot (X_{r_3,G} - X_{r_4,G}) \tag{5}$$

e) DE/rand/2

$$V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) + F \cdot (X_{r_4,G} - X_{r_5,G}) \tag{6}$$

Where  $i=1, 2, \dots, NP$ , random integer  $r_1, r_2, r_3, r_4,$  and  $r_5 \in [1, NP]$  are mutually different, which are different from the index  $i$ . The scaling factor  $F_i \in [0, 1]$  is a positive control parameter for scaling the difference vector. The  $x_{best}$  is the best vector with the minimum fitness at generation  $G$ .

**2.2. Crossover**

After the noise vector was generated through mutation, DE performs a binomial crossover on the target vector  $x_i$  and the noise vector  $v_i$  to generate a trial vector  $u_i = (u_{i,1}, u_{i,2}, \dots, u_{i,D})$ . The binomial crossover is defined as follows:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } (rnd \leq CR_i \text{ or } j = j_{rnd}) \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{7}$$

where  $j = 1, 2, \dots, D$ ,  $rnd \in [0,1]$  is a uniformly distributed random number and  $j_{rnd} \in [1, 2, \dots, D]$  is a randomly chosen index, which ensures that  $u_{i,j}$  gets at least one variable from the donor vector  $CR \in [0,1]$ .

### 2.3. Selection

Finally, a greedy selection scheme is used to choose the best vector that would survive to the next generation. A greedy selection scheme is described as

$$x_i = \begin{cases} u_i, & \text{if } (f(u_i) \leq f(x_i)) \\ x_i, & \text{otherwise} \end{cases} \quad (8)$$

where  $f(\cdot)$  is a function with minimizing feature.

The DE includes three steps, namely, mutation, crossover, and selection, which are repeated generation after generation until a termination criterion is satisfied.

## 3. THE PREVIOUS WORK

Differential evolution (DE) algorithm is an efficient evolutionary algorithm over continuous spaces. However, the performance of the DE algorithm depends on mutation, crossover strategies, and control parameters ( $NP$ ,  $F$ , and  $CR$ ). Generally, the appropriate selection of strategies and parameters may improve the performance of the DE algorithm. Because the best strategies and parameters may vary for optimization problems during the evolutionary process, it is difficult to find the appropriate strategies and parameters. In recent years, many researchers have proposed various empirical guidelines for choosing strategies and parameters depending on the problem.

Some research work focuses on mutation vector generation strategies. The standard DE algorithm employs DE/rand/1 strategy which focuses on exploration. To improve the exploitation of DE, the best individual in the current population is selected in the mutation strategies, such as DE/best/1 and DE/rand-to-best/1. The mutation strategies relying on the best individual are faster for easier optimization problems, but become unreliable when solving highly multi-modal problems.

Extensive researchers have been done on the appropriate parameter setting of DE.  $CR$  controls the mutation individual number in the current population. A large  $CR$  speeds up convergence. Extensive experimental results show that the right initial choice was  $CR = 0.1$ ; nevertheless,  $CR = 0.9$  or  $1.0$  could be tried to increase the convergence speed. In [7], a good value of  $CR$  was said to lie in  $[0.3, 0.9]$ .  $F$  is generally selected from  $[0.5, 1]$  [8]. A larger value of  $F$  increases the probability of getting away from a local optimum. Therefore, to avoid the tuning of trial-and-error, various techniques have been developed. Some parameter adaptation strategies were proposed, such as linear reduction [9] and random sampling [10]. A self-adaptation scheme (SDE) [11] is introduced, which uses the normal distribution  $N(0.5, 0.15)$  to generate  $CR$  randomly for each individual, where  $F$  is similar to the adaptation of  $CR$ . Brest et al. [12] proposed a self-adaptation scheme (jDE), in which parameters  $F$  and  $CR$  were encoded into the individuals and were adjusted in the process of DE. Biswas et al. [13] proposed a teaching- and learning-based self-adaptive DE (TLBSaDE) in which  $F$  was sampled from  $N(0.5, 0.3)$  and  $CR$  was picked from  $N(CR_m, 0.1)$ . JADE [14] uses a control parameter adaptation strategy based on updating the parameters of the probability distributions from which the values of  $F$  and  $Cr$  are sampled.

To improve the performance of DE algorithm, the parameter control and adaptive strategy has been explored in DE. A self-adaptive DE algorithm (SaDE) [15] is proposed, in which the strategies and the control parameter is self-adapted based on their potential to generate valuable solutions. CoDE [16] improves the performance of DE by combining several effective trial vector generation strategies with some suitable control parameter settings.

The concept of “island models” has been introduced to improve the performance of the evolutionary algorithm in several studies. MPCCA [17] is a multi-population based cooperative co-evolutionary algorithm to solve the multi-objective problem, in which a population is divided into multiple subpopulations with respect to its different direction vectors. Wu et al. [18] proposed a multipopulation-based approach (MPeDE) to realize an ensemble of multiple strategies that simultaneously consist of three mutation strategies, i.e., “current-to-pbest/1,” “current-to-rand/1,” and “rand/1”. Also, the DE algorithm was used optimization in [19-23]. DE algorithm was used as an optimization tool in many fields, i.e., tasks scheduling problem and data stream clustering, to improve its accuracy, security, and reliability.

## 4. THE PROPOSED ALGORITHM

In this section, a new DE algorithm, KSDE, is proposed, which applies the  $k$ -means clustering algorithm to divide the population into  $k$  subpopulations and which uses SIM to transfer information between subpopulations. After splitting the population into  $k$  subpopulations, multiple strategies are implemented to subpopulations. In this paper, three mutation strategies are chosen. Firstly, the “DE/rand/1” and “DE/rand/2” are selected. According to the characteristics of a subpopulation, a new mutation strategy is proposed to generate a mutation vector, such as (9).

$$v_i = a + F_i \cdot (x_{r_2} - b) + F_i \cdot (x_{r_4} - x_{r_5}) \tag{9}$$

Where the real  $a \in x_{r_1}$  and  $b \in x_{r_3}$  chosen randomly

To improve search diversity, the SIM is used to transfer information between populations. Firstly, the individual  $ind_i$  was found to belong to the island  $p_i$ . Subsequently, the vector  $r$  can be selected from either the current island or any island with probability  $P$ . The number  $m$  of vector  $r$  is determined by the algorithm. In this paper,  $m=5$  is set. The SIM flow chart is shown in Figure 1.

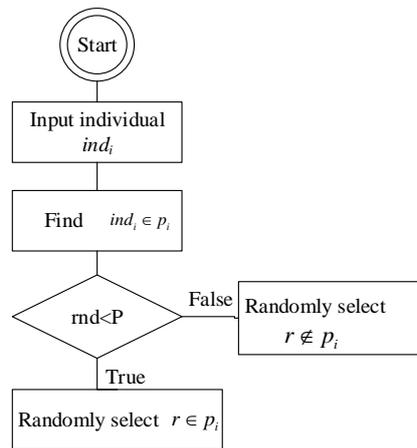


Figure 1. The SIM flow chart

Based on the above analysis, a novel DE variant, namely KSDE, is proposed, the primary key of which is to implement the same mutation strategy to the island with similar individuals, and SIM to the information exchange between islands. The pseudocode of the KSDE algorithm is presented in Algorithm 1.

KSDE algorithm has five input parameters, namely population size ( $NP$ ), dimension ( $D$ ), benchmark function ( $f$ ), and integers  $k$  and  $m$ . The global optimal  $best$  is the algorithm return. At lines 1 and 2 of algorithm 1, the population  $p$  and individual values  $fit$  are initialized. In line 3, a mutation strategy pool is built. In this paper, three strategies are chosen, namely as shown in (2), (5), and (9). In lines 4 and 5, the whole population evolution is controlled by the function to evaluate  $FES$ . At line 6, the population is split into  $k$  subpopulations according to the individual location. In this paper, parameters  $F$  and  $CR$  are sampled from the Gaussian distribution. The scaling factor  $F$  is sampled from the distribution  $N(0.5, 0.3)$  and  $N(0.9, 0.1)$  in line 7. To improve the search diversity,  $m$  individuals are selected by SIM in line 8. At lines 9 and 10, the subpopulation  $p_i$  is randomly assigned the mutation  $Sp_j$  to generate the mutation vector. Finally, the KSDE algorithm performs the crossover and selection operation and returns the global best  $fit_{best}$ . In the KSDE algorithm, multi-strategy is beneficial to improve population exploration ability, with the SIM enhancing population search diversity. Therefore, KSDE improves the exploration and exploitation of the population.

```

Algorithm 1: KSDE algorithm
Input: NP, D, f, k, m
Output: The population's best solution: fitbest
1. Generate the population p by the equations (1)
2. Calculate the individual function values fit
3. Strategy pool Sp={Sp1, Sp2, ..., Spn}
4. FES=NP
5. while FES<=NP*1000
6. The population p is divided into k subpopulation by k-means, p={p1,
   p2, ..., pk}
7. F=randn(0.5,0.3), Cr=randn(0.9,0.1)
8. Pick r1, r2, ..., rm using SIM
9. Randomly combine Spj and pi to Si,j, i ∈ [1, k], j ∈ [1, n]
10. pi implements the strategy Spj and generates noise vector
11. Apply Equation 7 to generate trial vector
12. Apply Equation 8 to select the best individual for the next generation
13. end while
14. Return fitbest
    
```

## 5. EXPERIMENT AND RESULT ANALYSIS

### 5.1. Benchmark functions and experimental setting

To evaluate the performance of the proposed SKDE, a set of 13 benchmark functions [24] are chosen for the experiment. Among the 13 functions,  $f_1$ - $f_5$  are unimodal.  $f_6$  is the step function while  $f_7$  is the noisy function.  $f_8$ - $f_{13}$  are multimodal. During evolution, excellent individuals can be preserved by evaluating function values. For all experiments, the parameters of all algorithms are set as follows unless a change is mentioned:

- Dimension:  $D = 30, 50, 100$
- Population size:  $NP = 100$
- Scaling factor:  $F = N(0.5, 0.3)$
- Crossover:  $CR = N(0.9, 0.1)$
- The termination criterion of function evaluations  
( $MaxFES$ ):  $MaxFES=10E+4$

Moreover, each function in each algorithm runs 25 independently. All the experiments are carried on a computer with 3.4GHz quad-core Processor and 16GB RAM under Windows 10. For the results of each algorithm, the average and standard values of the function are recorded separately. To further verify the result, Wilcoxon's statistical tests are conducted [25].

### 5.2. Result and analysis of comparison with basic DE

In this section, the superiority of the proposed algorithm is shown by comparing the KSDE algorithm with four other state-of-the-art DE variants, namely, CoDE, jDE, JADE, and MPEDE. In the experiments, the control parameters  $F$  and  $CR$  of these four DE variants consist of the original literature, with  $MaxFES$  in all algorithms set to  $NP*1000$ . Table 2 shows the experiment comparison results of 25 runs on 13 functions.

Table 2 shows the results of the function mean and standard deviation obtained by the KSDE and other advanced DE variants for 25 independent runs. As for the functions  $f_1$ - $f_{13}$ , apart from  $f_6$  and  $f_8$ , KSDE exhibits the best performance among the four algorithms. This is because KSDE could improve the population diversity by migrating individual information among different groups.

In addition to the above analysis, the Friedman test [26] was conducted on the experimental results for all dimensions. The average ranking of the five DE algorithms is presented in Table 1. The smaller average ranking value indicates better performance. Table 1 shows that KSDE is the best one, improving the performance of the DE algorithm.

The mean ranking comparing five algorithms is obtained to visualize their differences. Figure 2 shows the results considering all functions. One can observe that KSDE is the instance that achieves the best results in all functions.

Table 1. Average ranking based on the Friedman test

Algorithm	$D$	CoDE	jDE	JADE	MPEDE	KSDE
Ranking	30	3.35	3.73	2.38	4.38	1.15
	50	3.58	3.62	2.58	4.00	1.23
	100	3.65	3.12	3.65	3.54	1.04

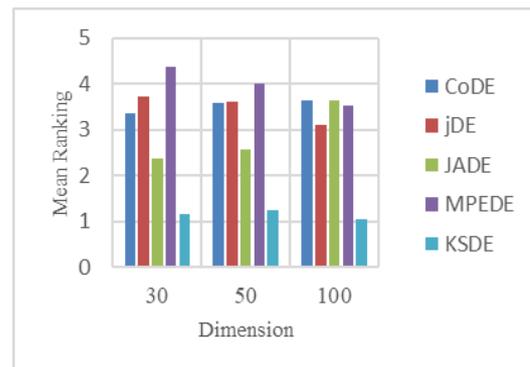


Figure 2. Average ranking based on the Friedman test

The statistical results of function values show that KSDE is the best option among 13 test functions, as shown in Table 2. The possible reason for the good performance of KSDE includes two aspects. First, the clustering method can improve one's exploration ability. Second, the SIM method enhances the population's diversity by migrating individual information among different groups.

Table 2. Comparison with reference algorithm for each dimension

F		CoDE		jDE		JADE		MPEDE		KSDE	
		Mean	Std								
f <sub>1</sub>	D=30	1.16E-19	2.05E-19	6.90E-18	4.38E-18	3.59E-39	1.24E-38	4.42E-13	2.99E-13	9.34E-105	3.04E-104
	D=50	8.71E-13	7.36E-13	3.01E-11	2.41E-11	4.79E-28	2.11E-27	2.37E-09	1.42E-09	1.97E-98	9.59E-98
	D=100	2.34E-05	1.16E-05	1.93E-05	1.07E-05	9.06E-12	6.94E-12	2.71E-04	1.66E-04	1.92E-97	4.47E-97
f <sub>2</sub>	D=30	7.89E-11	4.88E-11	3.37E-11	1.53E-11	1.29E-16	3.96E-16	1.25E-06	5.89E-07	5.53E-52	1.50E-51
	D=50	2.90E-07	1.03E-07	2.30E-07	1.00E-07	2.65E-15	3.59E-15	6.73E-05	2.13E-05	5.91E-49	2.63E-48
	D=100	1.24E-03	4.34E-04	8.37E-04	2.49E-04	4.45E-07	7.83E-07	2.29E-02	9.93E-03	2.70E-46	6.65E-46
f <sub>3</sub>	D=30	2.25E-03	2.93E-03	3.07E+00	1.64E+00	2.43E-09	3.26E-09	4.07E-06	9.49E-06	1.34E-24	3.69E-24
	D=50	1.62E+01	7.16E+00	4.82E+02	2.30E+02	8.61E-02	5.43E-02	1.56E+00	1.23E+00	8.83E-17	3.52E-16
	D=100	2.02E+03	7.09E+02	1.14E+04	1.62E+04	5.22E+02	1.78E+02	4.05E+02	1.43E+02	4.85E-03	2.42E-02
f <sub>4</sub>	D=30	1.26E-04	8.06E-05	1.06E+00	1.05E+00	1.29E-05	1.38E-05	3.17E-04	1.11E-04	6.28E-46	1.66E-45
	D=50	1.58E-01	1.34E-01	9.50E+00	3.45E+00	1.99E+00	9.22E-01	5.21E-02	2.10E-02	8.82E-43	1.59E-42
	D=100	1.42E+01	2.73E+00	2.58E+01	3.43E+00	1.27E+01	1.66E+00	3.38E+00	5.08E-01	1.33E-39	2.15E-39
f <sub>5</sub>	D=30	1.91E+01	1.60E+01	2.74E+01	1.68E+01	2.68E+00	1.21E+00	2.02E+01	1.16E+00	1.07E+00	5.37E+00
	D=50	4.87E+01	2.08E+01	6.54E+01	3.24E+01	3.96E+01	1.69E+01	4.91E+01	1.52E+01	1.88E+00	9.41E+00
	D=100	2.26E+02	6.04E+01	2.41E+02	7.18E+01	2.41E+02	5.65E+01	1.74E+02	6.48E+01	1.17E+01	3.22E+01
f <sub>6</sub>	D=30	0.00E+00	0.00E+00								
	D=50	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.40E-01	5.97E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	D=100	0.00E+00	0.00E+00	4.00E-02	2.00E-01	7.20E+00	4.48E+00	2.60E+00	2.92E+00	0.00E+00	0.00E+00
f <sub>7</sub>	D=30	8.12E-03	2.13E-03	1.02E-02	2.45E-03	2.07E-03	5.94E-04	3.14E-03	8.26E-04	3.11E-04	1.70E-04
	D=50	2.28E-02	7.04E-03	2.34E-02	5.12E-03	5.15E-03	1.49E-03	9.53E-03	3.22E-03	4.07E-04	1.86E-04
	D=100	1.36E-01	3.05E-02	6.58E-02	1.12E-02	5.62E-02	1.19E-02	4.72E-02	1.12E-02	3.36E-04	1.45E-04
f <sub>8</sub>	D=30	-1.26E+04	6.33E-12	-1.26E+04	2.07E-12	-1.26E+04	3.28E+01	-1.23E+04	1.39E+02	-1.23E+04	1.86E-12
	D=50	-2.09E+04	1.70E+02	-2.09E+04	3.34E-06	-2.09E+04	1.61E+01	-1.79E+04	5.13E+02	-2.09E+04	1.11E-11
	D=100	-2.85E+04	1.32E+03	-4.17E+04	1.50E+02	-3.32E+04	6.03E+02	-2.81E+04	1.18E+03	-4.19E+04	1.49E-11
f <sub>9</sub>	D=30	8.35E+00	3.72E+00	4.00E-05	5.55E-05	1.26E-04	6.75E-05	1.57E+01	3.30E+00	0.00E+00	0.00E+00
	D=50	8.84E+01	1.18E+01	2.04E+01	4.12E+00	1.05E+01	1.54E+00	6.13E+01	7.19E+00	0.00E+00	0.00E+00
	D=100	4.23E+02	3.70E+01	1.85E+02	1.94E+01	1.58E+02	1.09E+01	2.45E+02	2.33E+01	0.00E+00	0.00E+00
f <sub>10</sub>	D=30	6.22E-11	3.32E-11	5.67E-10	2.53E-10	4.87E-15	1.18E-15	2.01E-07	1.06E-07	8.88E-16	0.00E+00
	D=50	1.65E-07	7.24E-08	8.67E-07	3.90E-07	1.27E-14	2.85E-15	1.33E-05	5.00E-06	8.88E-16	0.00E+00
	D=100	7.41E-02	2.54E-01	5.76E-04	1.47E-04	1.61E+00	2.54E-01	3.36E-01	4.90E-01	8.88E-16	0.00E+00
f <sub>11</sub>	D=30	1.00E-13	5.01E-13	1.47E-16	5.14E-16	5.17E-15	2.46E-14	2.23E-08	1.12E-07	0.00E+00	0.00E+00
	D=50	5.92E-04	2.05E-03	4.26E-11	4.03E-11	1.18E-03	3.35E-03	2.36E-03	5.20E-03	0.00E+00	0.00E+00
	D=100	3.56E-03	8.09E-03	1.21E-05	8.00E-06	5.70E-03	1.06E-02	3.18E-03	1.04E-02	0.00E+00	0.00E+00
f <sub>12</sub>	D=30	1.48E-21	1.05E-21	6.64E-19	9.10E-19	1.78E-32	1.03E-32	1.13E-14	1.19E-14	1.57E-32	5.59E-48
	D=50	2.49E-03	1.24E-02	2.34E-12	3.61E-12	3.45E-25	1.71E-24	2.49E-03	1.24E-02	9.42E-33	0.00E+00
	D=100	1.55E-03	6.11E-03	8.09E-06	6.89E-06	6.73E-02	1.22E-01	4.37E-02	6.46E-02	4.71E-33	0.00E+00
f <sub>13</sub>	D=30	1.67E-20	2.46E-20	5.18E-18	5.42E-18	1.43E-32	1.77E-33	2.25E-13	2.13E-13	1.35E-32	5.59E-48
	D=50	4.39E-04	2.20E-03	3.45E-11	2.60E-11	1.87E-27	6.83E-27	9.15E-10	1.26E-09	1.35E-32	5.59E-48
	D=100	1.06E-03	3.01E-03	6.88E-04	2.14E-03	5.67E-01	1.12E+00	2.35E-03	5.34E-03	1.35E-32	5.59E-48

For the convenience of illustration, evolution graphs are plotted on seven functions:  $f_1, f_3, f_5, f_7, f_9, f_{10}$ , and  $f_{13}$ . The evolution process of the mean best values has a dimension of  $n=30$ . The results are averaged over 25 runs. Figure 2 shows the convergence graphs respectively. From Figure 3, one can conclude that KSDE improves the convergence of the DE algorithm.

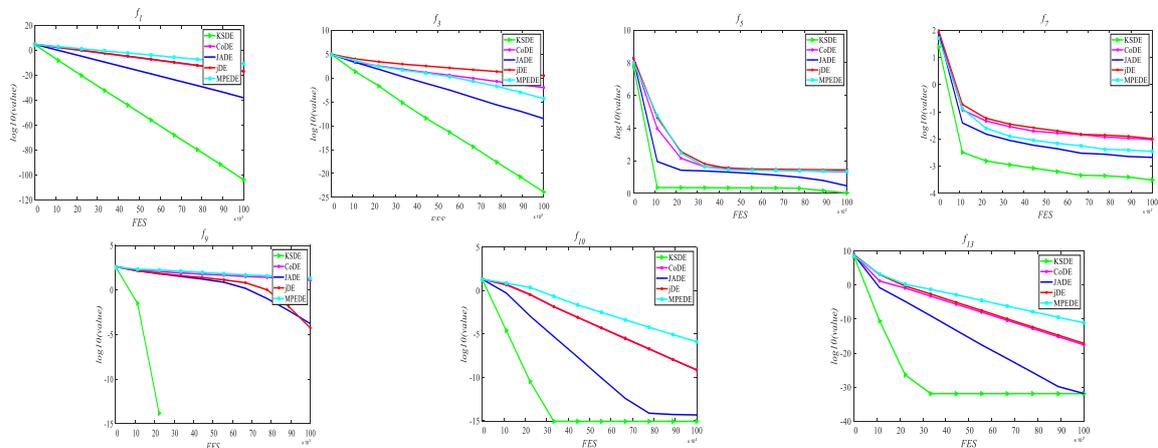


Figure 3. The evolution process of the average best values for  $f_1, f_3, f_5, f_7, f_9, f_{10}$ , and  $f_{13}$  with a dimension of  $D=30$  over 25 runs

### 5.3. Parameter analysis

KSDE introduces parameters  $k$  and  $m$  which determine the diversity of the population. The  $m$  is the number of individuals selected in the mutation strategy, and  $k$  is the number of islands. In this paper,  $m$  is 5. To discover the influence of  $k$  on the algorithm efficiency, a number of experiments have been performed using 13 problems with a dimension of 30. To select the appropriate parameter  $k$ , 25 runs for 13 benchmark functions have been completed. Experimental results are shown in Tables 3 and 4. These illustrations show that  $k=2$  improves the efficiency of the KSDE algorithm.

Table 3. Experimental results with KSDE for different numbers of islands

F	k=1		k=2		k=3		k=4		k=5	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
$f_1$	2.99E-18	2.71E-18	9.58E-124	3.95E-123	3.38E-104	1.03E-103	2.84E-79	7.37E-79	2.09E-63	5.92E-63
$f_2$	9.50E-11	4.46E-11	1.32E-61	4.88E-61	1.57E-51	4.49E-51	4.90E-40	6.81E-40	1.78E-31	5.81E-31
$f_3$	3.25E-01	3.68E-01	6.62E-28	2.24E-27	2.79E-25	6.91E-25	4.91E-19	6.86E-19	1.12E-13	3.88E-13
$f_4$	1.28E+01	5.54E+00	1.57E-53	3.00E-53	8.82E-46	2.04E-45	3.11E-35	7.11E-35	9.90E-29	1.22E-28
$f_5$	2.91E+01	2.13E+01	0.00E+00	0.00E+00	1.08E+00	5.39E+00	1.10E+00	5.48E+00	0.00E+00	0.00E+00
$f_6$	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
$f_7$	9.99E-03	3.20E-03	2.84E-04	1.70E-04	3.10E-04	1.91E-04	5.56E-04	3.79E-04	5.30E-04	3.43E-04
$f_8$	-1.15E+04	3.49E+02	-1.26E+04	1.86E-12	-1.26E+04	1.86E-12	-1.26E+04	1.86E-12	-1.26E+04	1.86E-12
$f_9$	1.38E+01	4.01E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
$f_{10}$	3.52E-10	1.67E-10	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
$f_{11}$	3.35E-03	7.39E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
$f_{12}$	1.24E-02	4.56E-02	1.57E-32	5.59E-48	1.57E-32	5.59E-48	1.57E-32	5.59E-48	1.57E-32	5.59E-48
$f_{13}$	4.39E-04	2.20E-03	1.35E-32	5.59E-48	1.35E-32	5.59E-48	1.35E-32	5.59E-48	1.35E-32	5.59E-48

Table 4. Average ranking based on the Friedman test

Algorithm	k=1	k=2	k=3	k=4	k=5
Ranking	4.85	1.88	2.38	2.92	2.96

## 6. CONCLUSION AND FUTURE WORK

In this paper, a novel differential evolution algorithm, the soft island model based on  $k$ -means clustering (KSDE), which maintains population diversity through the soft island model, was proposed. In the process of population evolution, the population was divided into many sub-populations by the  $k$ -means clustering algorithm, and each sub-population performed different mutation strategies. To improve the diversity of DE, the population data were divided into different groups by the  $k$ -means clustering algorithm, followed by a novel information exchange mechanism. The superior performance of KSDE was evaluated on the basis of a set of benchmark functions compared with other state-of-the-art DE variants. The experimental results showed that KSDE was effective and efficient. In future studies, the effects will be focused on large-scale optimization problems with high dimension. Another direction is to apply various tensor operations in DE to optimize real-world problems further.

## REFERENCES

- [1] Storn R, Price K., "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol 11, no. 4, pp. 341-359, 1997.
- [2] Melin P, Sanchez D., "Multi-objective optimization for modular granular neural networks applied to pattern recognition," *Information Sciences*, vol. 460, pp. 594-610, 2018.
- [3] C. Dong, et al, "An improved differential evolution and its application to determining feature weights in similarity-based clustering," *Neurocomputing*, vol. 146, pp. 95–103, 2014.
- [4] G. Y. Wiederschain, "Data mining techniques for the life sciences," *Biochemistry (Moscow)*, vol. 76, no. 4, pp. 494–494, 2011.
- [5] Gundry S, Zou J, Uyar M U, et al., "Differential evolution-based autonomous and disruption tolerant vehicular self-organization in MANETs," *Ad Hoc Networks*, vol. 25, pp. 454-471, 2015.
- [6] Akhmedova S, Stanovov V, Semenkin E., "Soft island model for population-based optimization algorithms," *International Conference on Swarm Intelligence, Springer, Cham*, pp. 68-77, 2018.
- [7] Gämperle R, Müller S D, Koumoutsakos P., "A parameter study for differential evolution," *Advances in intelligent systems, fuzzy systems, evolutionary computation*, vol. 10, no. 10, pp. 293-298, 2002.
- [8] Storn R., "On the usage of differential evolution for function optimization," *Proceedings of North American Fuzzy Information Processing. IEEE*, 1996, pp. 519-523.
- [9] Das S, Konar A, Chakraborty U K., "Two improved differential evolution schemes for faster global search," *Proceedings of the 7th annual conference on Genetic and evolutionary computation, ACM*, pp. 991-998, 2005.

- [10] Chakraborty U K, Das S, Konar A., "Differential evolution with local neighborhood," *2006 IEEE International Conference on Evolutionary Computation, IEEE*, pp. 2042-2049, 2006.
- [11] Omran M G H, Salman A, Engelbrecht A P., "Self-adaptive differential evolution," *International Conference on Computational and Information Science, Springer, Berlin, Heidelberg*, pp. 192-199, 2005.
- [12] Brest J, et al., "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE transactions on evolutionary computation*, vol. 10, no. 6, pp. 646-657, 2006.
- [13] Biswas S, Kundu S, Das S, et al., "Teaching and learning best differential evolution with self adaptation for real parameter optimization," *IEEE Congress on Evolutionary Computation, IEEE*, pp. 1115-1122, 2013.
- [14] Zhang J, Sanderson A C., "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on evolutionary computation*, vol. 3, no. 5, pp. 945-958, 2009.
- [15] Qin A K, Huang V L, Suganthan P N., "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398-417, 2008.
- [16] Wang Y, Cai Z, Zhang Q., "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55-66, 2011.
- [17] Shang R, Wang Y, Wang J, et al., "A multi-population cooperative coevolutionary algorithm for multi-objective capacitated arc routing problem," *Information Sciences*, vol. 277, pp. 609-642, 2014.
- [18] Wu G, Mallipeddi R, Suganthan P N, et al., "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 329, pp. 329-345, 2016.
- [19] Qiao F, Zhang G, "A Fuzzy Differential Evolution Scheduling Algorithm Based on Grid," *Bulletin of Electrical Engineering and Informatics*, vol. 1, no. 4, pp. 279-284, 2012.
- [20] Rao D S, Kumar M S, Raju M R, "Design of Robust Controller for Higher Order Interval System Using Differential Evolutionary Algorithm," *International Journal of Robotics and Automation (IJRA)*, vol. 7, no. 4, pp. 232-250, 2018.
- [21] Alhamrouni I, Salem M, Khairuddin A B, et al. "AC-Based Differential Evolution Algorithm for Dynamic Transmission Expansion Planning," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 5, pp. 2316-2330, 2018.
- [22] Adepu B, Gyani J, Narsimha G, "An improved differential evolution algorithm for data stream clustering," *International Journal of Electrical & Computer Engineering*, pp. 2088-8708, 2019.
- [23] Son N N, Nguyen D K, Tran M C, "Black-box modeling of nonlinear system using evolutionary neural NARX model," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 1861, 2019.
- [24] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary computation*, vol. 3, no. 2, pp. 82-102, 1999.
- [25] Tan X, Shin S Y, "Differential Evolution with Multi-strategies based Soft Island Model," *Journal of information and communication convergence engineering*, vol. 17, no. 4, pp. 261-266, 2019.
- [26] López-Vázquez C, Hochstain E, "Extended and updated tables for the Friedman rank test," *Communications in Statistics-Theory and Methods*, vol. 48, no. 2, pp. 268-281, 2019.

## BIOGRAPHIES OF AUTHORS



**Xujie Tan** is currently studying for a Ph.D degree at Kunsan National University, Korea. He is a lecturer in the School of Information Science and Technology at Jiujiang University, JiuJiang, China. His research interests include nature-inspired computations and applications, Pervasive Computing, Big Data and Cloud Computing. Contact him at [tanxj@kunsan.ac.kr](mailto:tanxj@kunsan.ac.kr).



**Seong-Yoon Shin** received his M.S. and Ph.D. degrees from the Dept. of Computer Information Engineering at Kunsan National University, Gun-san, Korea, in 1997 and 2003, respectively. From 2006 to the present, he has been a professor in the same department. His research interests include image processing, computer vision, and virtual reality.