Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 17(4): 261-266, Dec. 2019 Regular paper

# Differential Evolution with Multi-strategies based Soft Island Model

**Xujie Tan**[1] **and Seong-Yoon Shin**[2,*], ***Member, KIICE***

[1]School of Information Science and Technology, Jiujiang University, Jiujiang 332005, China
[2]School of Computer Information & Communication Engineering, Kunsan National University, Kunsan 54150, South Korea

## Abstract

Differential evolution (DE) is an uncomplicated and serviceable developmental algorithm. Nevertheless, its execution depends on strategies and regulating structures. The combination of several strategies between subpopulations helps to stabilize the probing on DE. In this paper, we propose a unique *k*-mean soft island model DE(KSDE) algorithm which maintains population diversity through soft island model (SIM). A combination of various approaches, called KSDE, intended for migrating the subpopulation information through SIM is developed in this study. First, the population is divided into k subpopulations using the k-means clustering algorithm. Second, the mutation pattern is singled randomly from a strategy pool. Third, the subpopulation information is migrated using SIM. The performance of KSDE was analyzed using 13 benchmark indices and compared with those of high-technology DE variants. The results demonstrate the efficiency and suitability of the KSDE system, and confirm that KSDE is a cost-effective algorithm compared with four other DE algorithms.

**Index Terms**: Differential evolution, Evolutionary algorithm, Soft island model, K-means clustering

## I. INTRODUCTION

Differential evolution (DE), proposed by Price and Storn [1], is an unsophisticated and beneficial evolutionary algorithm (EA) for solving optimization problems. Recently, DE has been widely applied in diverse fields, such as pattern recognition [2], artificial neural networks [3], image processing [4], and electronics and communication engineering [5].

DE's functioning primarily depends on its trial vector generation strategy (i.e., mutation and crossover operators) and regulatory boundaries (i.e., population scope *NP*, scaling component *F*, and crossover structures *CR*). The relevant plans and regulatory scales are invaluable in advancing DE implementation. Recently, the multi-island model has been adopted to advance DE execution to obtain the apt sequence of strategy and control metrics. The information exchange among islands can maintain diversity and balance the exploitation and exploration capabilities.

Based on these considerations, a novel multi-island DE, called *k*-mean soft island model DE(KSDE), is proposed in this study. In KSDE, the population is classified into clusters using *k*-means cluster algorithm. Consequently, a more suitable mutation strategy may be randomly selected to match different clusters. KSDE uses the soft island model (SIM) [6] to migrate individuals and enhance population assortment. To evaluate the effectiveness of KSDE, a KSDE analysis was conducted on 13 benchmark functions with 30 variables.

The remainder of this study is structured as follows. In Section II, the classic DE is established. In Section III, related literature is reviewed. The proposed DE algorithm, named KSDE, is presented in detail in Section IV. The experimental results are presented in Section V. Finally, Sec-

**Received** 17 November 2019, **Revised** 24 December 2019, **Accepted** 24 December 2019
***Corresponding Author** Seong-Yoon Shin (E-mail: s3397220@kunsan.ac.kr, Tel: +82-63-469-4860, Fax. +82-63-469-4560)
School of Computer Information & Communication Engineering, Kunsan National University, Kunsan 54150, Korea.

Open Access https://doi.org/10.6109/jicce.2019.17.4.261 print ISSN: 2234-8255 online ISSN: 2234-8883

tion VI concludes the study and provides recommendations for future studies.

## II. DIFFERENTIAL EVOLUTION

Differential evolution is used to solve real number optimization problems. The object function is given as $f(x)$, $x = (x_1, x_2, \ldots, x_D)$, where $D$ denotes the space dimension.

First, the $NP$ population $x$ is randomly generated; hence, each vector of the $x_i$ in the $G$ generation can be generated using (1):

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

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

### A. Mutation

After initialization, the donor vector $v_i$ is produced with respect to $x_i$ using (2). During the $G$ generation, $v_i$ can be generated through the DE/rand/1 mutation strategies.

$$v_i = x_{r_1} + F_i \cdot (x_{r_2} - x_{r_3}). \quad (2)$$

Here, $i = 1, 2, \ldots, NP$, and random integers $r_1$, $r_2$, and $r_3 \in [1, NP]$ are reciprocally dissimilar and distinct from the key $i$. The scaling component $F_i \in [0, 1]$ is a positive control criterion for gauging the trajectory difference.

### B. Crossover

After the noise vector is generated through modification, DE actuates a binomial shift on the pursued vector $x_i$ and noise vector $v_i$ to engender an experimental vector $u_i = (u_{i,1}, u_{i,2}, \cdots, u_{i,D})$. The binomial crossover is defined as expressed in (3).

$$u_{i,j} = \begin{cases} v_{i,j}, if(rnd \le CR_i \, or \, j = j_{rnd}) \\ x_{i,j}, otherwise \end{cases}. \quad (3)$$

Here, $j = 1, 2, \ldots, D$, $rnd \in [0,1]$ is an unvaryingly dispersed random digit, and $j_{rnd} \in [1, 2, \ldots, D]$ is an arbitrarily selected index, which guarantees that $u_{i,j}$ secures a minimum variable, originating from the donor vector $CR \in [0,1]$.

### C. Selection

Finally, an acquisitive selection arrangement is employed to ensure that the best vector survives to the succeeding generation. An insatiable selection order is described in (4).

$$x_i = \begin{cases} u_i, if(f(u_i) \le f(x_i)) \\ x_i, otherwise \end{cases}. \quad (4)$$

In (4), $f(.)$ is a function with a minimized feature.

The DE includes three rungs, namely, mutation, crossover, and election. The DE is consecutively reiterated while waiting for a dissolution benchmark to be satisfied.

## III. PREVIOUS STUDIES

The DE system is an efficient and developmental algorithm over continuous spaces. However, the implementation of the DE algorithm depends on mutation, crossover schemes, and control structures ($NP$, $F$, and $CR$). Generally, the appropriate combination of strategies and restrictions could improve the performance of the DE algorithm. Recently, several scholars have proposed various empirical guidelines for choosing relevant strategies and parameters depending on the problem.

Some studies focused on mutation vector generation strategies. The conventional DE system employs the DE/rand/1 strategy, which focuses on exploration. To improve the utilization of DE, the most agreeable entity in the existing population is selected in the mutation strategy, such as DE/best/1 and DE/rand-to-best/1.

Extensive research has been conducted on appropriate parameter settings of DE. Therefore, various tactics have been developed to circumvent maladjustment through trial and error. Several parameter reworking approaches have been proposed, including linear decrease [7] and arbitrary sampling [8]. Brest et al. [9] recommends a self-adaptation arrangement (jDE), where $F$ and $CR$ were preset into individuals and fine-tuned during the DE process. JADE, presented by Zhang and Sanderson [10], operates a control parameter acclimatization scheme in keeping the parameters up-to-date in terms of dispersion, from which the merits of F and Cr are assessed.

To build up the operation of the DE algorithm, the parameter settings and adaptive strategy have been explored in DE. A self-adaptive DE algorithm (SaDE) [11] has been developed, where the procedures and control metrics are self-acclimatized in line with their familiarities in enhancing clarifications. CoDE [12] improves the execution of DE by combining effective trial vector generation strategies with appropriate control boundary contexts.

The concept of "island models" has been initiated in several studies to enhance the performance of the EA. Wu et al. [13] proposed a multi-population grounded methodology (MPEDE) to accomplish multiple collective strategies, which concurrently entails three modification schemes: "current-to-pbest/1," "current-to-rand/1," and "rand/1."

## IV. PROPOSED ALGORITHM

This section describes a novel DE algorithm, i.e., KSDE.

KSDE applies the $k$-means assembling algorithm in dividing the population into $k$ subpopulations and uses SIM to transfer information between subpopulations.

After splitting the population into $k$ subpopulations, multiple strategies were implemented to the subpopulations. In this study, three mutation strategies were selected. Firstly, the strategy “DE/rand/1” and the strategy “DE/rand/2” are selected. According to the characteristics of subpopulation, a new mutation strategy to generate a mutation vector is proposed in (5)

$$v_i = a + F_i \cdot (x_{r_2} - b) + F_i \cdot (x_{r_4} - x_{r_5}). \quad (5)$$

In (5), the real $a \in x_n$ and $b \in x_r$ are chosen randomly.

To improve the search diversity, SIM was utilized for transferring information between the subpopulations. The individual $ind_i$ was found to belong to the island $p_i$. Subsequently, the vector $r$ can be selected from either the contemporary island or any island possessing the probability $P$. The number $m$ of vector $r$ is determined from the algorithm. In this study, $m$ = 5 is set. Based on the analysis above, the pseudocode of the KSDE algorithm is presented in Algorithm 1.

**Algorithm 1**. KSDE algorithm

**Input**: $NP$, $D$, $f$, $k$, $m$
**Output**: Population’s best solution: $fit_{best}$

1. Generate the population $p$ using equation (1);
2. Calculate the individual function values $fit$;
3. Strategy pool $Sp = \{Sp_1, Sp_2, \ldots, Sp_n\}$;
4. $FES$=$NP$;
5. while $FES$ <= $NP$*1000
6. The population $p$ is divided into $k$ subpopulation using $k$-means, $p = \{p_1, p_2, \ldots, p_k\}$;
7. $F$ = randn(0.5,0.3), $Cr$ = randn(0.9,0.1);
8. Pick $r_1, r_2, \ldots, r_m$ using SIM;
9. Randomly combine $Sp_j$ and $p_i$ to $S_{i,j}$, $i \in [1, k]$, $j \in [1, n]$;
10. $p_i$ implements the strategy $Sp_j$ and generates the noise vector;
11. Apply equation (3) to generate the trial vector;
12. Apply equation (4) to select the best individual for the next generation;
13. end while
14. Return $fit_{best}$.

The KSDE algorithm had five input parameters, namely population size $NP$, dimension $D$, benchmark function $f$, and integers $k$ and $m$. In lines 1 and 2 of Algorithm 1, the population $p$ and individual value fit are initialized. In line 3, a mutation strategy pool is built. The whole population evolution is controlled by the function evaluates ($FES$) in lines 4 and 5. In line 6, the population is split into $k$ subpopulations based on the individual location. In this study, the quantitative metrics $F$ and $CR$ are samples from the Gaussian distribution. The parameters $F$ and $CR$ are assembled from the dispersion $N$(0.5, 0.3) and $N$(0.9, 0.1), respectively, in line 7. To improve the search diversity, $m$ individuals are selected by SIM in line 8. In lines 9 and 10, the subpopulation $p_i$ is randomly assigned to mutation $Sp_j$ to generate the mutation vector. Finally, the KSDE algorithm performs the crossover and selection operation and returns the global best solution $fit_{best}$.

In KSDE Algorithm 1, the multi-strategy improves the population exploration ability, while SIM enhances the population search diversity. Therefore, KSDE improves the exploration of the population.

## V. EXPERIMENTAL STUDY

### A. Benchmark Functions and Experimental Setting

To evaluate the performance of KSDE, 13 benchmark functions [14] were selected in carrying out the experiment. Among the 13 tasks, $f_1$-$f_5$ are unimodal, $f_6$ is the step function, $f_7$ is the noisy function, and $f_8$-$f_{13}$ are multimodal.

In all the experiments, the parameters of all algorithms were set unless a change is required: $D$ = 30, $NP$ = 100, $F$ = $N$(0.5, 0.3), and $CR$ = $N$(0.9, 0.1). The termination criterion of function evaluations was 10E+4. Moreover, every function in each algorithm independently has 25 runs on a Windows 10 computer with a 3.4 GHz quad-core processor and 16 GB RAM. Wilcoxon’s statistical tests were conducted for the results of each algorithm [15].

### B. Experimental results and comparisons with other DE variants

This section checks the superiority of the proposed algorithm by comparing the results of the KSDE algorithm with those of four other state-of-the-art DE modifications: CoDE [12], jDE [9], JADE [10], and MPEDE [13]. The parameters of the four DE variants used in this study are consistent with those in the literature. Table 1 displays the statistical results of 13 function values achieved using CoDE, jDE, JADE, MPEDE, and KSDE. The Wilcoxon’s rank-sum test results are summarized at the lowermost portion of Table 2, where “–,” “+,” and “≈” indicate that the performance of the associated algorithm is inferior, superior, or similar to that of KSDE, respectively. For functions $f_1$-$f_{13}$, KSDE exhibits the best performance among the five algorithms, except $f_6$. The main reason is that KSDE can improve the population diversity by migrating individual information between subpopulations.

In addition to the analysis above, the Friedman test was performed on the experimental results for all dimensions. The median standing of the five DE sets is presented in Table 2. The average ranking with a smaller value represents

**Table 1.** Comparison with reference algorithm for each dimension

| F | CoDE | | jDE | | JADE | | MPEDE | | KSDE | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Mean | Standard | Mean | Standard | Mean | Standard | Mean | Standard | Mean | Standard |
| $f_1$ | 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 |
| $f_2$ | 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 |
| $f_3$ | 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 |
| $f_4$ | 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 |
| $f_5$ | 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 |
| $f_6$ | 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 |
| $f_7$ | 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 |
| $f_8$ | -1.26E+04 - | 6.33E-12 | -1.26E+04 - | 2.07E-12 | -1.26E+04 - | 3.28E+01 | -1.23E+04 - | 1.39E+02 | -1.26E+04 | 1.86E-12 |
| $f_9$ | 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 |
| $f_{10}$ | 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 |
| $f_{11}$ | 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 |
| $f_{12}$ | 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 |
| $f_{13}$ | 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 |
| -/+/≈ | 12/0/1 | | 12/0/1 | | 12/0/1 | | 12/0/1 | | | |

**Table 2.** Average ranking based on the Friedman test

| Algorithm | CoDE | jDE | JADE | MPEDE | KSDE |
|---|---|---|---|---|---|
| Ranking | 3.35 | 3.73 | 2.38 | 4.38 | 1.15 |

a better performance. KSDE has the best performance among the five schemes used in the test.

The statistical results of the function values demonstrate that KSDE is the best algorithm for the 13 test functions. There are two possible reasons for the excellent performance of KSDE. First, the clustering method can enhance exploration ability. Second, the SIM method improves the population diversity by migrating individual information among different groups.

For the convenience of illustration, the evolution graphs of five functions, $f_1$, $f_3$, $f_5$, $f_{10}$, and $f_{13}$, are provided. Fig. 1 depicts the convergence graphs of functions $f_1$, $f_3$, $f_5$, $f_{10}$, and $f_{13}$, respectively, in which the curves of each graph uses an average value of 25 runs. From Fig. 1, it can be observed that KSDE improves the convergence of the DE algorithm.

## VI. CONCLUSION

In this study, a unique KSDE algorithm, which maintains population diversity using a soft island model, was proposed. During the process of population evolution, the population was divided into several subpopulations through $k$-means clustering algorithm, where individual subpopulations performed distinct mutation functions. To advance the DE range, the population data were divided into different groups using the $k$-means clustering algorithm and then the new information exchange mechanism by SIM. The experiments

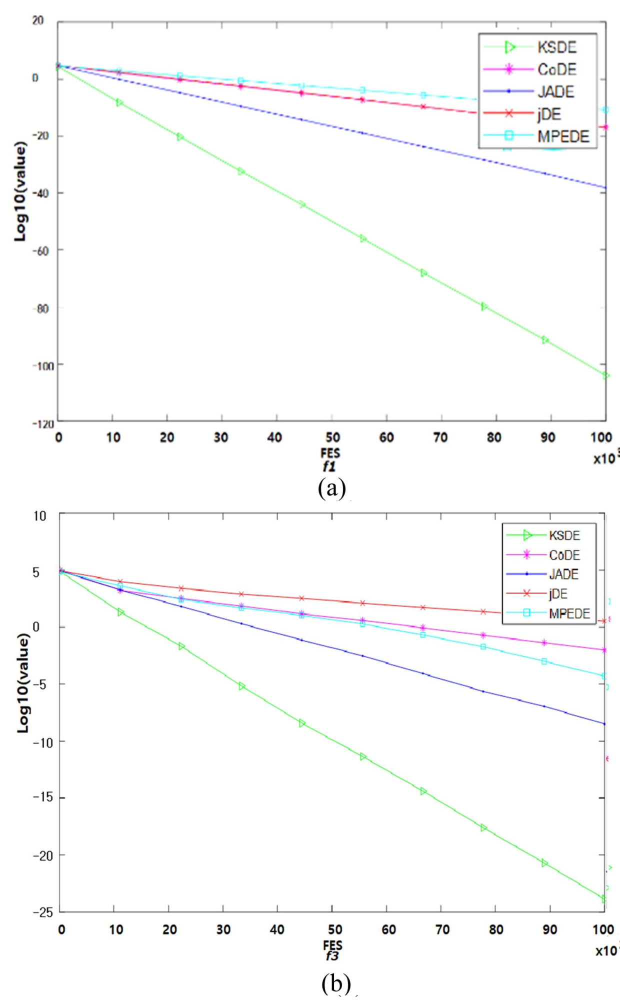
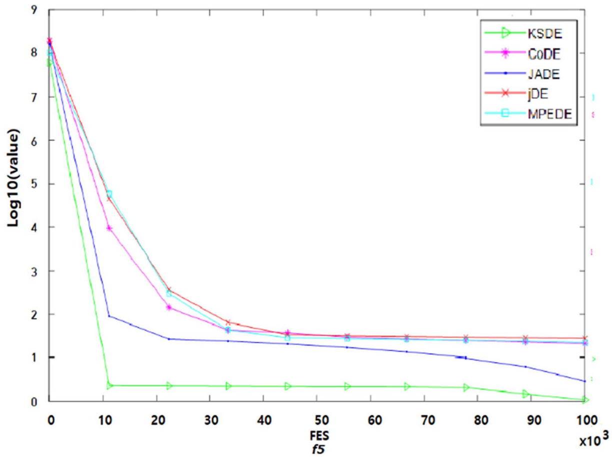
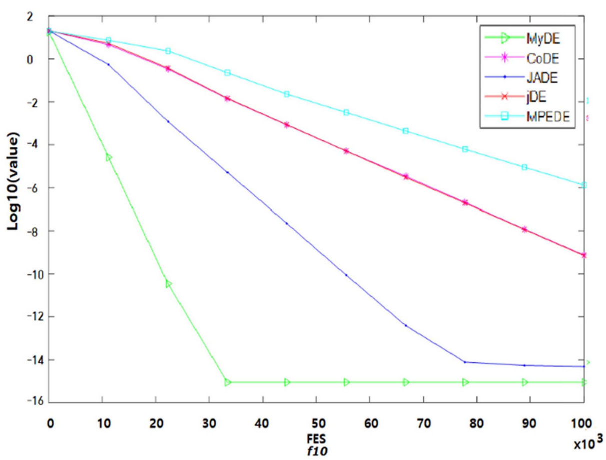


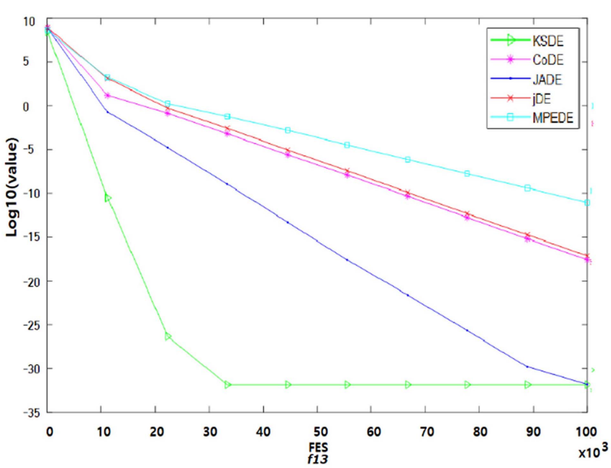

**Fig. 1.** Evolution process of the average best values for (a) $f_1$, (b) $f_3$, (c) $f_5$, (d) $f_{10}$, and (e) $f_{13}$ with dimension D = 30 over 25 runs.




(c)




(d)




(e)

**Fig. 1.** Evolution process of the average best values for (a) $f_1$, (b) $f_3$, (c) $f_5$, (d) $f_{10}$, and (e) $f_{13}$ with dimension D = 30 over 25 runs (continued).

were performed using 13 benchmark indices. The performance of KSDE was compared with those of similar high-technology DE variants. The results demonstrate the efficiency and suitability of the KSDE system.

In future studies, the effects on large-scale optimization problems with high dimension should be investigated. Another research direction involves applying various tensor operations in DE to optimize real-world problems further.

## REFERENCES

[ 1 ] R. Storn and K. Price, “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. DOI: 10. 1023/A:100820282.

[ 2 ] P. Melin and D. Sanchez, “Multi-objective optimization for modular granular neural networks applied to pattern recognition,” *Information Sciences*, vol. 460, pp. 594-610, 2018. DOI: 10.1016/j.ins.2017.09.031.

[ 3 ] J. Kumar and A. K. Singh, “Workload prediction in cloud using artificial neural network and adaptive differential evolution,” *Future Generation Computer Systems*, vol. 81, pp. 41-52, 2018. DOI: 10.1016/j.future.2017.10.047.

[ 4 ] S. Sarkar, S. Das, and S. S. Chaudhuri, “Hyper-spectral image segmentation using Rényi entropy based multi-level thresholding aided with differential evolution,” *Expert Systems with Applications*, vol. 50, pp. 120-129, 2016. DOI: 10.1016/j.eswa.2015.11.016.

[ 5 ] Z. Li, X. Zhou, Y. Liu, H. Xu, and L. Miao, “A non-cooperative differential game-based security model in fog computing,” *China Communications*, vol. 14, no. 1, pp. 180-189, 2017. DOI: 10.1109/ CC.2017.7839768.

[ 6 ] S. Akhmedova, V. Stanovov, and E. Semenkin, “Soft island model for population-based optimization algorithms,” *International Conference on Swarm Intelligence*, Springer, Cham, pp. 68-77, 2018. DOI: 10.1007/978-3-319-93815-8_8.

[ 7 ] M. Z. Ali, N. H. Awad, P. N. Suganthan, and R. G. Reynolds, “An adaptive multipopulation differential evolution with dynamic population reduction,” *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2768-2779, 2016. DOI: 10.1109/TCYB.2016.2617301.

[ 8 ] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, Vienna, Austria, pp. 308-318, 2016. DOI: 10.1145/2976749.2978318.

[ 9 ] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, “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. DOI: 10.1109/TEVC.2006.872133.

[10] J. Zhang and A. C. Sanderson, “JADE: Adaptive differential evolution with optional external archive,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945-958, 2009. DOI: 10.1109/TEVC.2009.2014613.

[11] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398-417, 2008. DOI: 10.1109/TEVC.2008.927706.

[12] Y. Wang, Z. Cai, and Q. Zhang, “Differential evolution with composite trial vector generation strategies and control parameters,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55-66, 2011. DOI: 10.1109/TEVC.2010.2087271.

[13] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang, and H. Chen, “Differential evolution with multi-population based ensemble of

mutation strategies," *Information Sciences*, vol. 329, pp. 329-345, 2016. DOI: 10.1016/j.ins.2015.09.009.
[14] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82-102, 1999. DOI: 10.1109/4235.771163.
[15] H. Peng, X. Tan, C. Deng, and S. Peng, "SparkCUDE: A spark-based differential evolution for large-scale global optimization," *International Journal of High Performance Systems Architecture*, vol. 7, no. 4, pp. 211-222, 2017. DOI: 10.1504/IJHPSA.2017.092390.

**Xujie Tan**
Xujie Tan is currently studying for a Ph.D. degree in the School of Computer Information & Communication Engineering at Kunsan National University, Korea, and he is a lecturer in the School of Information Science and Technology at Jiujiang University, JiuJiang, China. His research interests include nature-inspired computation and application, pervasive computing, big data, and cloud computing. He can be reached through tanxj@kunsan.ac.kr.

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