

Pseudo Multi-Population Differential Evolution for Multimodal Optimization

Hao-Feng Li, Yue-Jiao Gong (Corresponding Author), Zhi-Hui Zhan, Wei-Neng Chen, and Jun Zhang

Department of Computer Science, Sun Yat-sen University

Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education

Engineering Research Center of Supercomputing Engineering Software, Ministry of Education

Guangzhou 510006, China

gongyuejiao@gmail.com

Abstract—Multimodal optimization aims at locating multiple optima in a run, which has two main advantages over traditional single objective global optimization. First, it would be useful to provide multiple solutions since some solutions may be hard to realize physically. Second, a multimodal algorithm is not so easy to get stuck in a local optimum. In recent years, multi-population evolutionary algorithms have been used for multimodal optimization. However, their ability to locate multiple peaks is limited by the number of populations used. It is difficult to find out all the peaks if the populations are fewer than the peaks. When algorithms increase the number of populations, they have to maintain huge population sizes and hence encounter lower search efficiency. This paper overcomes such deficiencies by proposing a pseudo multi-population differential evolution (p-MPDE). The p-MPDE employs a small exemplar population to conduct normal DE operation. Each other individual uses the differential of two randomly chosen members in the exemplar population to mutate themselves and evolve. Each such individual represents a pseudo population and promises to find a global or local optimum. In the experiment, p-MPDE was compared to other state-of-the-art multimodal algorithms and the result shows that p-MPDE outperforms R3PSO, LIPS and CDE on CEC2013 niching benchmark.

Keywords- Evolution Algorithm; Multimodal Optimization; Differential Evolution.

I. INTRODUCTION

LOCATING multiple global and local peaks of an objective function in a run is advantageous to tackle real-world optimization problems. Due to physical limits, it may be hard to realize the optimal solution in many applications. If more than one acceptable result can be found, there will be more choices for the implementation. In addition, searching for multiple peaks contributes to reduced possibility of getting stuck in local peaks and hence increases the chance of finding out the global optimum [1].

Employing a conventional method is usually ineffective for such problems, since the algorithm needs to run repeatedly to detect a distinct optimal result each time and a

specific deterministic algorithm often inclines to locate some fixed peaks. Evolutionary algorithms (EAs) [2], which maintain a population of candidate solutions, show an inherent advantage over traditional optimization approaches if multiple favorable solutions can be maintained during the evolution.

However, EAs tend to converge to a single optimal or suboptimal peak, which makes recognition and preservation of multiple good results difficult when applying traditional EAs to multimodal optimization.

To tackle multimodal task, some methods maintaining multiple subpopulations were proposed [3]. Such methods need to reserve many subpopulations to locate as many as possible peaks. If the exploitation ability of each subpopulation needs to be guaranteed, the population size may be too large to tolerate, which limits the multi-population method considerably. In addition, a multi-population method may need to configure the subpopulation settings before generation. It is always difficult to preset subpopulation sizes and the number of subpopulation.

This paper proposes a pseudo multi-population differential evolution (p-MPDE) which not only satisfies the criteria that multimodal algorithm should locate as many as possible global peaks but also avoids the above-mentioned limitation of traditional multi-population method. The algorithm uses a small exemplar population to conduct a normal differential evolution. Meanwhile, each individual outside this exemplar population pretends to be a population evolving itself, which is termed a pseudo population. These individuals use differential vectors generated from the exemplar population to mutate themselves. A pseudo population is never allowed to cover another while it updates. Thus it is highly unlikely that each pseudo population converges to a single peak. Moreover, instead of mutating randomly, a pseudo population learns the perturbation pattern of the exemplar population to mutate more wisely. In short, each pseudo population promises to locate a distinct peak separately.

The advantage of p-MPDE is that the algorithm can achieve the effect of using multiple population without really using multi-population method. The number of pseudo populations is abundant because the pseudo population is actually an individual. Thus large number of pseudo population will not lead to huge population size. The p-MPDE not only avoids presetting multiple populations but

This work was supported in part by the National High-Technology Research and Development Program (863 Program) of China No.2013AA01A212, in part by the NSFC for Distinguished Young Scholars 61125205, in part by the NSFC No. 61332002 and No.61300044.

also avoids large population resulted from multiple populations. In this paper, the experiment displays that p-MPDE outperforms some existing multimodal algorithms on CEC2013 niching benchmark across five levels of accuracy.

The rest of this paper is organized as follows. Section II introduces related work for solving multimodal task. Section III describes the derivation and details of the proposed p-MPDE. Section IV is the experimental setting and the experimental result. Section V gives some concluding comments.

II. RELATED WORK

A. Differential Evolution

Differential Evolution (DE) [4][5] is a metaheuristic approach for minimizing nonlinear and non-differentiable continuous problems.

DE maintains a population and each individual in the population is a vector representing a candidate solution. At the start of the algorithm, the population is initialized with NP random vectors where NP stands for the population size. Similar to genetic algorithms, DE evolves the population in each generation consisting of mutation, crossover and selection.

In the mutation operation, for each individual $x_{i,G}$, $i=1,2,\dots, NP$, a mutation vector is generated using

$$v_{i,G+1} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}) \quad (1.)$$

with random and distinct indexes $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$. F is the scalar factor that amplifies the differential variation. In the crossover stage, a trial vector $u_{i,G+1}$ is generated using the following equation.

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } (randb(j) \leq CR) \text{ or } j = rnbr(i) \\ x_{ji,G} & \text{otherwise} \end{cases} \quad (2.)$$

$j = 1, 2, \dots, D$

where CR is the crossover rate; D is the dimensionality of the solution; $u_{ji,G+1}$ is the value of the individual $u_{i,G+1}$ in j th dimension; $randb(j)$ is a random number generator and generates a random number between 0 and 1 for each dimension; $rnbr(i)$ is a randomly chosen index which is proposed to ensure that $x_{i,G+1}$ exchanges at least one parameter with $v_{i,G+1}$. In the selection stage, if $u_{i,G+1}$ is fitter than $x_{i,G+1}$, $x_{i,G+1}$ is replaced by $u_{i,G+1}$. Table I is a pseudo code of DE. DE is a promising single objective global optimizer. However, some features of DE are not proper for maintaining the population diversity in some scenarios, which will be revealed in the following section.

B. Multimodal Differential Evolution

Crowding Differential Evolution: Crowding differential evolution (CDE) was first proposed by Thomsen [6]. In CDE, when an offspring is generated, it competes with the individual most similar to it (measured by Euclidean distance) in the current population. The offspring will replace this individual if it has a better fitness value.

C. Other Algorithms for Multimodal Optimization

Particle swarm optimization (PSO) [7]-[10], which was introduced by Kennedy and Eberhart [8] in 1995, is a powerful optimization approach. The population-based nature interests people to apply niching techniques to PSO for solving multimodal optimization problems.

1) *Ring Topology PSO:* In 2010, a ring topology PSO [10] for multimodal optimization was proposed by Li to conquer the niching parameter specification problem as the ring topology PSO does not require any niching parameters. Each member only interacts with its immediate left (and/or) right neighbors in the ring topology, which is simple yet useful.

2) *Distance-Based Locally Informed Particle Swarm:* To solve multimodal optimization tasks, recently B.Y. Qu *et al* [11] introduced an impressive distance-based locally informed particle swarm which exceeds nine other state-of-the-art multimodal optimizers over a test suite comprised of 15 basic and 15 composite multimodal benchmarks. The velocity update of LIPS uses the formula given below.

$$V_i^d = \omega \times (V_i^d + \phi (P_i^d - X_i^d)) \quad (3.)$$

$$P_i = \frac{\sum_{j=1}^{nsize} (\phi_j \cdot nbest_j)}{\phi} \quad (4.)$$

ϕ_j is a uniformly distributed random number in the range of $[0, 4.1/nsize]$, ϕ is equal to the summation of ϕ_j , $nsize$ (the neighborhood size) is dynamically increased from 2 to 5 over the function evaluations, $nbest_j$ is the j^{th} nearest neighborhood to the i^{th} particle's $pbest$, while the position update equation keeps unchanged [12].

III. PSEUDO MULTI-POPULATION DIFFERENTIAL EVOLUTION

A. Large Population and Multi-Population

Multimodal optimization aims at locating multiple peaks and maintains them throughout the algorithm. Firstly, multimodal algorithm needs to track not only a single solution. Secondly, multimodal algorithm runs without knowing how many global peaks the multimodal problem has. Apparently, it is very important to enhance the diversity of population in a multimodal algorithm.

A simple method to enhance diversity of population is to increase the population size. However, enlarging the population may lead to a great waste of function evaluation times and lowers the efficiency of exploration because of the usually fixed maximum function evaluations. Many individuals could crowd around a not promising area and it is no use evaluating all of them. Moreover, individuals of population tend to converge to a single peak if algorithm has no mechanism to make some adjacent members repel each other.

Another simple but useful measure is to separate the individuals into distinct subpopulations and each subpopulation is expected to locate a different optimum. However, normal multi-population algorithm is difficult to find out all of the peaks because the subpopulations may be

fewer than the peaks. Even though in some cases subpopulations are ample, locating all the optima is still hard since the algorithm has no appropriate mechanism to make subpopulations repel each other. The algorithm should maintain more subpopulations in multimodal optimization than in single objective global optimization. To keep the exploitation ability of each subpopulation at an acceptable level, multimodal algorithms require an immense population to track all peaks. Limited by the available computing resources, increased population size leads to decreased maximum of evolutionary generations, which worsen the performance of algorithm.

B. Mutation Operator in Differential Evolution

Differential Evolution is a simple and fast algorithm which performs well in single objective global optimization. When DE is applied to solve multimodal task, however, its standard mutation has a negative impact on the diversity of the population. The standard mutation in DE is depicted in equation (5.). *Trial* refers to a trial individual generated by mutation. X_{r1} , X_{r2} and X_{r3} are three different randomly chosen vectors which are distinct from the current individual X_i . In early search stage, trial vector could appear in an area near X_i or far from X_i because X_{r1} is chosen randomly. Thus equation (5.) ensures sufficient global search. During the late search stage, the peak X_{r1} exploits may be far away from the peak X_i exploits. The Trial vector derived from X_{r1} could be similar to X_{r1} . If *Trial* is fitter than X_i , X_i will be replaced by *Trial*. However, such replacement may be unreasonable because X_i and *Trial* could be discovering two distinct peaks which are far apart. It is not proper to replace an individual around a peak with an individual near another peak for the population diversity.

$$Trial = X_{r1} + F * (X_{r2} - X_{r3}) \quad (5.)$$

$$Trial = X_i + F * (X_{r1} - X_{r2}) \quad (6.)$$

This paper proposes equation (6.) as differential evolution mutation. Compared to equation (5.), equation (6.) replaces X_{r1} with X_i . In exploration stage, trial vector can reach many places in the whole landscape because the differential of two randomly chosen vector X_{r1} and X_{r2} can be large or small. Thus algorithms using equation (6.) have an acceptable ability to search the landscape globally in early stage. During exploitation stage, in (6.), if X_{r1} and X_{r2} are from the same niche, their differential will be not so large. Thus trial vector could appear around X_i and it is likely that *Trial* and X_i discover the same peak. Even if X_i is replaced by *Trial*, it does not affect diversity of population. Such replacement could be regarded as evolving X_i around a peak.

C. Pseudo Multi-Population Method

As already mentioned, normal multi-population evolutionary algorithms require immense population size. To solve such problem, this paper proposes a pseudo multi-population differential evolution (p-MPDE) which adopts equation (6.) as its mutation operator. The p-MPDE consists of a small exemplar population and multiple pseudo populations. A pseudo population actually refers to one individual which is not a member of the exemplar

population. Such a mechanism helps to avoid using large population size. Each pseudo subpopulation employs the differential between two members in the exemplar population to perform mutation using equation (6.). Note that X_{r1} and X_{r2} are two randomly chosen members of the exemplar population. After mutation, crossover and selection in p-MPDE remain the same as that of traditional DE.

The following is an explanation of the reasonability and effectiveness of p-MPDE. Suppose that two populations with the same population size evolve independently in the same landscape. Both of populations take equation (6.) as their mutation operator. Namely, Population 1 uses equation (7.) and Population 2 uses equation (8.). X_{1i} denotes the i^{th} individual of Population 1. X_{1r1} and X_{1r2} denote two randomly chosen members of Population 1. Similarly, X_{2i} is the i^{th} individual of Population 2. X_{2r1} and X_{2r2} are two randomly chosen individuals in Population 2.

$$Trial = X_{1i} + F * (X_{1r1} - X_{1r2}) \quad (7.)$$

$$Trial = X_{2i} + F * (X_{2r1} - X_{2r2}) \quad (8.)$$

$$Trial = X_{2i} + F * (X_{1r1} - X_{1r2}) \quad (9.)$$

Take notice of the distinction between equation (8.) and (9). (8.) and (9) have the same base X_{2i} . Their differential part are generated from different populations. Equation (9) means that an individual from Population 2 uses the differential generated by two randomly chosen members from Population 1. Actually equation (9) depicts what the individual in a pseudo subpopulation do in its mutation step. If replacing equation (8.) with equation (9) is reasonable and acceptable, using the individual representing a pseudo subpopulation and the differential generated from the exemplar population to generate a trial vector is also reasonable and acceptable.

At the very beginning, as individuals of Population 1 and Population 2 are all distributed randomly in the landscape so that the differentials generated from these two populations have no essential differences. In late search stage, Population 1 and Population 2 both almost converge to their respective peak. The magnitudes of $X_{1r1} - X_{1r2}$ and $X_{2r1} - X_{2r2}$ both are very small. Considering the direction of the differential vectors, as two member indices are selected in a complete random manner, the probability of each possible directions being selected is the same. Thus $X_{1r1} - X_{1r2}$ and $X_{2r1} - X_{2r2}$ are similar to a certain degree. To some extent, it is reasonable that an individual denoting a pseudo subpopulation uses the differential vector generated from the exemplar population to mutate the individual itself.

D. Pseudo Multi-Population Differential Evolution

Algorithm I is a detailed procedure in p-MPDE. Firstly, initialize the population and evaluate members of the population. Secondly, determine the individuals to constitute an exemplar population. Thirdly, in every generation, renew each member in the population. In update step, take equation (10) as mutation operator. After mutation, crossover and selection in p-MPDE are as the same as in DE.

$$Trial = X_i + F * (X_{er1} - X_{er2}) \quad (10)$$

ALGORITHM I
PSEUDO CODE OF P-MPDE

Step	Detail
Step 1	Initialize a population randomly. Evaluate the population.
Step 2	Determine an exemplar group of individuals.
Step 3	For $i:=1$ to NP do Choose 2 members (X_{er1} , X_{er2}) in the exemplar group randomly. $Trial = X_i + F * (X_{er1} - X_{er2})$; $Trial = \text{crossover}(Trial, X_i)$; If ($Trial$ is fitter than X_i) replace X_i with $Trial$;
Step 4	Go to step 3 if termination condition is not satisfied. Otherwise stop.

IV. EXPERIMENTAL STUDIES

A. Experimental Setting

All the algorithms are implemented using C and executed on the computer with an Intel(R) Core™ i5-2410M CPU and 4 GB of memory. Four different multimodal algorithms are tested through the experiments.

1) p-MPDE: the proposed pseudo multi-population differential evolution;

2) CDE [6]: the original crowding DE;

3) R3PSO [10]: an lbest PSO with a ring topology that each member interacts with its immediate member on its left and right;

4) LIPS [11]: the locally informed distance-based PSO.

The PSO parameters used in this paper are adopted from [12]. The DE parameters are $F=0.9$ and $CR=0.1$.

B. Test Functions

In the experiment, the following 20 multimodal test functions in CEC2013 niching benchmark [13] are used:

- F_1 : Five-Uneven-Peak Trap (1D)
- F_2 : Equal Maxima (1D)
- F_3 : Uneven Decreasing Maxima (1D)
- F_4 : Himmelblau (2D)
- F_5 : Six-Hump Camel Back (2D)
- F_6 : Shubert (2D, 3D)
- F_7 : Vincent (2D, 3D)
- F_8 : Modified Rastrigin – All Global Optima (2D)
- F_9 : Composition Function 1 (2D)
- F_{10} : Composition Function 2 (2D)
- F_{11} : Composition Function 3 (2D, 3D, 5D, 10D)
- F_{12} : Composition Function 4 (3D, 5D, 10D, 20D)

All test functions are formulated as maximization problems. F_1 , F_2 and F_3 are simple 1D multimodal functions. F_4 and F_5 are simple 2D multimodal functions. These functions are not scalable. F_6 to F_8 are scalable multimodal functions. The number of global optima for F_6 and F_7 are determined by the dimension D . F_9 to F_{12} are scalable multimodal functions constructed by several basic functions with different properties. [13]

C. Population Size, Accuracy Level and Maximal Number of Evaluations

TABLE I
MAXFES USED FOR 3 RANGES OF TEST FUNCTIONS

Range of functions	MaxFES
F_1 to F_5 (1D or 2D)	5.0E+04
F_6 to F_{11} (2D)	2.0E+05
F_6 to F_{12} (3D or higher)	4.0E+05

TABLE II
PARAMETERS USED FOR PERFORMANCE MEASUREMENT

Function	R	Peak height	No.global optima	Population size
F_1 (1D)	0.01	200.0	2	50
F_2 (1D)	0.01	1.0	5	50
F_3 (1D)	0.01	1.0	1	50
F_4 (2D)	0.01	200.0	4	50
F_5 (2D)	0.5	1.03163	2	50
F_6 (2D)	0.5	186.731	18	100
F_7 (2D)	0.2	1.0	36	200
F_6 (3D)	0.5	2709.0935	81	400
F_7 (3D)	0.2	1.0	216	600
F_8 (2D)	0.01	-2.0	12	100
F_9 (2D)	0.01	0	6	100
F_{10} (2D)	0.01	0	8	100
F_{11} (2D)	0.01	0	6	100
F_{11} (3D)	0.01	0	6	100
F_{12} (3D)	0.01	0	8	100
F_{11} (5D)	0.01	0	6	100
F_{12} (5D)	0.01	0	8	100
F_{11} (10D)	0.01	0	6	100
F_{12} (10D)	0.01	0	8	100
F_{12} (20D)	0.01	0	8	100

In this experiment, five levels of accuracy indicate how close to the solutions the exact global peaks are. If the distance between a computed solution and a known global optimum is below the level of accuracy, then the peak is considered to have been found. The maximal numbers of function evaluations are listed in Table I. The population size and niching radius(R) are listed in Table II. Level of accuracy (\mathcal{E}): {1.0E-01, 1.0E-02... 1.0E-05}.

D. Performance Measure

Peak ratio values are used as key criterion to rank algorithms. Peak ratio is the percentage of successfully located peaks. To compare the performance of multimodal algorithms, 50 runs of each algorithm are taken on each test function in each level of accuracy.

E. Experimental Result

The experiment results are presented on peak ratio across 20 test functions and five levels of accuracy. Table III compares p-MPDE with R3PSO on peak ratio. For each test function and each level of accuracy, if p-MPDE outperforms R3PSO, the peak ratio value of p-MPDE uses bold. Otherwise the peak ratio value of R3PSO uses bold. Win Times represents the number of test functions on which an algorithm outperforms the other in the same level of

accuracy. Table III showed that p-MPDE performs better than R3PSO in every level of accuracy. In low level of accuracy, the difference between p-MPDE and R3PSO is not so large. In higher level of accuracy, p-MPDE outperforms R3PSO in 16 or 17 functions while R3PSO performs better than p-MPDE in only one function.

Table IV presents the result of comparing p-MPDE with LIPS, which is a recently proposed multimodal algorithm in [11]. As Table IV shows, in low level of accuracy, p-MPDE has obvious advantages over LIPS. With the level of accuracy increasing, the difference between p-MPDE and LIPS becomes smaller. In the highest level of accuracy, LIPS even surpasses p-MPDE. From a general view, both p-MPDE and LIPS are promising for multimodal optimization, and p-MPDE performs slightly better than LIPS.

Table V is the result of comparing p-MPDE with CDE, which is a classical multimodal differential evolution. In all levels of accuracy, p-MPDE outperforms CDE. In lower level of accuracy, p-MPDE has a more obvious advantage over CDE. In higher level of accuracy, the difference between p-MPDE and CDE becomes smaller. In total, the performance of p-MPDE is better than that of R3PSO, LIPS and CDE.

V. CONCLUSION

In this paper, a novel multimodal extensions (p-MPDE) to the conventional differential evolution was proposed. The algorithm uses a small exemplar population to run basic DE. Then, each individual out of the exemplar represents one pseudo population. Each pseudo population learns the pattern of evolution of the exemplar population to evolve

itself. The evolution of each pseudo population is independent so that each pseudo population is supposed to locate a different optimum. Thus p-MPDE performs well on multimodal optimization.

The algorithm was tested on CEC2013 niching benchmark. As the experimental result shows, p-MPDE outperforms R3PSO, LIPS and CDE. Further work could be done to extend this paper. Pseudo multi-population method could be applied to other evolutionary algorithms.

References

- [1] X. Li, "A multimodal particle swarm optimizer based on fitness Euclidean-distance ration," in *Proc. Genet. Evol. Computat. Conf.*, 2007, pp. 78–85.
- [2] E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. New York: Springer, 2003.
- [3] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," in *Proc. Conf. Genet. Evol. Computat.*, 2005, pp. 873–880.
- [4] Rainer Storn and Kenneth Price, "Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces" *Journal of Global Optimization* 11: 341–359, 1997.
- [5] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Computat.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [6] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proc. IEEE Congr. Evol. Computat.*, Jun. 2004, pp. 1382–1389.
- [7] A. Petrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proc. 3rd IEEE Congr. Evol. Computat.*, May 1996, pp. 798–803.
- [8] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Nov.–Dec. 1995, pp. 1942–1948.
- [9] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann, 2001.

TABLE III
PEAK RATIO: P-MPDE VS R3PSO

Level of accuracy	1.0E-01		1.0E-02		1.0E-03		1.0E-04		1.0E-05	
	p-MPDE	R3PSO	p-MPDE	R3PSO	p-MPDE	R3PSO	p-MPDE	R3PSO	p-MPDE	R3PSO
F_1 (1D)	1	0.62	1	0.19	1	0.11	1	0.11	1	0.16
F_2 (1D)	1	0.996	1	1	1	0.988	1	0.976	1	0.96
F_3 (1D)	1	1	1	1	1	1	1	1	1	1
F_4 (2D)	1	1	1	0.98	1	0.89	1	0.885	1	0.795
F_5 (2D)	1	1	1	1	1	1	1	1	1	1
F_6 (2D)	0.958889	0.813333	0.962222	0.768889	0.914444	0.737778	0.855556	0.708889	0	0
F_7 (2D)	0.638889	0.994444	0.635	0.286667	0.612778	0.275	0.592222	0.243889	0.588333	0.227222
F_6 (3D)	0.005679	0.165926	0.000247	0.148148	0.000741	0.137037	0	0.121728	0	0.117284
F_7 (3D)	0.50213	0.078611	0.400926	0.078704	0.312963	0.067315	0.182407	0.057407	0.082037	0.050926
F_8 (2D)	0.998333	0.92	0.996667	0.836667	0.996667	0.783333	0.99	0.763333	0.995	0.716667
F_9 (2D)	0.78	0.9	0.686667	0.646667	0.67	0.63	0.673333	0.61	0.676667	0.606667
F_{10} (2D)	0.89	0.4025	0.8	0.3525	0.7625	0.36	0.715	0.34	0.69	0.3275
F_{11} (2D)	0.7	0.573333	0.676667	0.47	0.666667	0.446667	0.666667	0.466667	0.666667	0.413333
F_{11} (3D)	0.836667	0.83	0.666667	0.553333	0.666667	0.513333	0.666667	0.493333	0.666667	0.51
F_{12} (3D)	0.8175	0.57	0.545	0.2025	0.535	0.155	0.445	0.1325	0.445	0.145
F_{11} (5D)	0.976667	0.45	0.663333	0.226667	0.653333	0.216667	0.653333	0.18	0.613333	0.166667
F_{12} (5D)	0.7075	0.0925	0.2875	0.08	0.27	0.0675	0.2225	0.05	0.1875	0.0775
F_{11} (10D)	0.593333	0.716667	0.243333	0.11	0.21	0.1	0.18	0.123333	0.166667	0.133333
F_{12} (10D)	0.0675	0	0.02	0	0.0075	0	0.01	0	0.005	0
F_{12} (20D)	0.0925	0	0.0525	0	0.0425	0	0.04	0	0.025	0
Win Times	13	4	16	1	17	1	17	1	16	1

- [10] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Computat.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.
- [11] B. Y. Qu, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das, "A Distance-Based Locally Informed Particle Swarm Model for Multimodal Optimization" in *IEEE Trans Evol. Computat.*, vol. 17, no. 3, pp.387-402, JUNE 2013.
- [12] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Trans. Evol. Computat.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.
- [13] X. LI, Engelbrecht, A., & Eitropakis, M. (2013). Benchmark functions for CEC 2013 special session and competition on niching methods for multimodal function optimization. *Royal Melbourne Institute of Technology, Tech. Rep.*

TABLE IV
PEAK RATIO: P-MPDE VS LIPS

Level of accuracy function	1.0E-01		1.0E-02		1.0E-03		1.0E-04		1.0E-05	
	p-MPDE	LIPS	p-MPDE	LIPS	p-MPDE	LIPS	p-MPDE	LIPS	p-MPDE	LIPS
F_1 (1D)	1	1	1	1	1	1	1	1	1	1
F_2 (1D)	1	0.996	1	0.996	1	0.996	1	1	1	1
F_3 (1D)	1	1	1	1	1	1	1	1	1	1
F_4 (2D)	1	1	1	1	1	1	1	1	1	1
F_5 (2D)	1	1	1	1	1	1	1	1	1	1
F_6 (2D)	0.958889	0.773333	0.962222	0.755556	0.914444	0.758889	0.855556	0.733333	0	0
F_7 (2D)	0.638889	0.670556	0.635	0.671667	0.612778	0.665	0.592222	0.652222	0.588333	0.656667
F_6 (3D)	0.005679	0.443457	0.000247	0.437778	0.000741	0.431358	0	0.42	0	0.395802
F_7 (3D)	0.50213	0.375926	0.400926	0.375185	0.312963	0.36	0.182407	0.34	0.082037	0.323981
F_8 (2D)	0.998333	0.995	0.996667	0.995	0.996667	0.993333	0.99	0.995	0.995	0.995
F_9 (2D)	0.78	0.856667	0.686667	0.836667	0.67	0.853333	0.673333	0.853333	0.676667	0.853333
F_{10} (2D)	0.89	0.825	0.8	0.86	0.7625	0.7975	0.715	0.83	0.69	0.8275
F_{11} (2D)	0.7	0.7	0.676667	0.683333	0.666667	0.69	0.666667	0.69	0.666667	0.683333
F_{11} (3D)	0.836667	0.643333	0.666667	0.643333	0.666667	0.63	0.666667	0.61	0.666667	0.653333
F_{12} (3D)	0.8175	0.415	0.545	0.425	0.535	0.395	0.445	0.395	0.445	0.39
F_{11} (5D)	0.976667	0.176667	0.663333	0.17	0.653333	0.156667	0.653333	0.156667	0.613333	0.173333
F_{12} (5D)	0.7075	0.205	0.2875	0.2225	0.27	0.22	0.2225	0.2125	0.1875	0.195
F_{11} (10D)	0.593333	0.036667	0.243333	0.026667	0.21	0.05	0.18	0.026667	0.166667	0.043333
F_{12} (10D)	0.0675	0.0125	0.02	0.0125	0.0075	0.0075	0.01	0.01	0.005	0.0025
F_{12} (20D)	0.0925	0	0.0525	0	0.0425	0	0.04	0	0.025	0
Win Times	12	3	11	5	9	6	7	7	6	7

TABLE V
PEAK RATIO: P-MPDE VS CDE

Level of accuracy function	1.0E-01		1.0E-02		1.0E-03		1.0E-04		1.0E-05	
	p-MPDE	CDE	p-MPDE	CDE	p-MPDE	CDE	p-MPDE	CDE	p-MPDE	CDE
F_1 (1D)	1	1	1	1	1	1	1	1	1	0.76
F_2 (1D)	1	1	1	1	1	1	1	1	1	1
F_3 (1D)	1	1	1	1	1	1	1	1	1	1
F_4 (2D)	1	1	1	1	1	1	1	1	1	1
F_5 (2D)	1	1	1	1	1	1	1	1	1	1
F_6 (2D)	0.958889	1	0.962222	1	0.914444	0.911111	0.855556	0.022222	0	0
F_7 (2D)	0.638889	0.845556	0.635	0.834444	0.612778	0.827778	0.592222	0.816111	0.588333	0.715556
F_6 (3D)	0.005679	0.002222	0.000247	0	0.000741	0	0	0	0	0
F_7 (3D)	0.50213	0.401759	0.400926	0.38963	0.312963	0.393148	0.182407	0.358889	0.082037	0.160093
F_8 (2D)	0.998333	1	0.996667	1	0.996667	1	0.99	1	0.995	1
F_9 (2D)	0.78	0.96	0.686667	0.693333	0.67	0.67	0.673333	0.666667	0.676667	0.666667
F_{10} (2D)	0.89	0.37	0.8	0.09	0.7625	0.0125	0.715	0	0.69	0
F_{11} (2D)	0.7	0.813333	0.676667	0.673333	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667
F_{11} (3D)	0.836667	0.696667	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667	0.666667
F_{12} (3D)	0.8175	0.7225	0.545	0.6925	0.535	0.665	0.445	0.5525	0.445	0.3925
F_{11} (5D)	0.976667	0.686667	0.663333	0.666667	0.653333	0.666667	0.653333	0.666667	0.613333	0.666667
F_{12} (5D)	0.7075	0.555	0.2875	0.4425	0.27	0.27	0.2225	0.115	0.1875	0.0575
F_{11} (10D)	0.593333	0.54	0.243333	0.24	0.21	0.186667	0.18	0.19	0.166667	0.173333
F_{12} (10D)	0.0675	0	0.02	0	0.0075	0	0.01	0	0.005	0
F_{12} (20D)	0.0925	0.3	0.0525	0.015	0.0425	0	0.04	0	0.025	0
Win Times	9	6	7	7	6	5	6	5	6	5