2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Enhanced Differential Evolution with Adaptive Direction Information

Yiqiao Cai and Jixiang Du

Abstract— **Most recently, a DE framework with neighborhood and direction information (NDi-DE) was proposed to exploit the information of population and was demonstrated to be effective for most of the DE variants. However, the performance of NDi-DE heavily depends on the selection of direction information. In order to alleviate this problem, two adaptive operator selection (AOS) mechanisms are introduced to adaptively select the most suitable type of direction information for the specific mutation strategy during the evolutionary process. The new method is named as adaptive direction information based NDi-DE (aNDi-DE). In this way, the good balance between exploration and exploitation can be dynamically achieved. To evaluate the effectiveness of aNDi-DE, the proposed method is applied to the well-known DE/rand/1 algorithm. Through the experimental study, we show that aNDi-DE can effectively improve the efficiency and robustness of NDi-DE.**

I. Introduction

DIFFERENTIAL EVOLUTION (DE), proposed by Storn and Price [1], is a simple and powerful evolutionary algorithm for global optimization over continuous space. It has many attractive characteristics, such as compact structure, ease to use, speedness and robustness. Recently, DE has been extended for handling multiobjective, constrained, large scale, dynamic and uncertain optimization problems [2]. The rapidly growing popularity of DE has made it be successfully used in various scientific and engineering fields [2][3], such as chemical engineering, engineering design, pattern recognition and so on.

When DE is applied to a given optimization problem, there are two main factors which significantly affect the behavior of DE. One is the control parameters (i.e., population size NP, mutation scaling factor F and crossover rate C_r), and the other is the evolutionary operators (i.e., mutation, crossover and selection). During the last decade, there are many researchers working on the improvement of DE. According to [4], these modern DE variants can be divided into two categories, DE integrating an extra component and modified structures of DE. Modifications on DE in these DE variants mainly focus on introducing the self-adaptive strategies for the control parameters [5][6][7], devising the new mutation operators [8][9][10], developing the ensemble strategies [5], proposing the hybrid DE with other optimization algorithms [11] and population topology (multi or parallel population)[12], and so on.

Yiqiao Cai and Jixiang Du are with the College of Computer Science and Technology, Huaqiao University, Xiamen, China (email: caiyq@hqu.edu.cn).
This work was supported in part by the National Natural Science Foundation of China (61305085, 61175121), the grant of the National Science Foundation of Fujian Province (No.2013J06014) and the Fundamental Research Funds for the Central Universities (12BS216)

Distinct from other evolutionary algorithms (EAs), the salient feature of DE is its mutation mechanism. However, in most of the DE algorithms, the neighborhood and direction information of population are not fully and simultaneously exploited. In order to alleviate this drawback and enhance the performance of DE, a DE framework with neighborhood and direction information (NDi-DE) [9] was proposed, with two novel operators: the neighbor guided selection scheme (NGS) and direction induced mutation strategy (DIM). Although NDi-DE is demonstrated to be effective for most of the DE variants, the performance of NDi-DE heavily depends on the selection of direction information for the specific mutation strategy. Furthermore, although the guideline for selecting the type of direction information is developed in [9], the application of NDi-DE to the new DE algorithm is still a difficult problem. In view of this limitation, a novel framework, adaptive direction information based NDi-DE (aNDi-DE), is developed to enhance the performance of DE and is expected to be more robustness than NDi-DE. Specifically, aNDi-DE employs the adaptive operator selection (AOS) techniques, named Probability Matching (PM) [13] and Adaptive Pursuit (AP) [14], to automatically select the most suitable type of direction information for the specific mutation strategy during the evolutionary process. In this way, the good balance between exploration and exploitation of aNDi-DE can be dynamically achieved.

To evaluate the effectiveness of the proposed method, aNDi-DE is applied to a famous and widely used DE algorithm, DE/rand/1. Through the extensive experimental study, we show that aNDi-DE is able to adaptively select the most suitable type of direction information for DE/rand/1 during the evolutionary process. The high performance of aNDi-DE is also confirmed by comparing with the DE and NDi-DE algorithms.

The rest of this paper is organized as follows. In Section II, the related works are briefly reviewed. In Section III, the proposed aNDi-DE is presented in detail. Next, the results of an experimental analysis are reported in Section IV. Finally, the conclusions are drawn in Section V.

II. Related Work

In this section, the original DE algorithm is introduced firstly. Then, the related work to NDi-DE is reviewed.

A. DE

DE is for solving the numerical optimization problem [1]. Without loss of generality, we consider the optimization problem to be minimized is $f(X)$, $X \in S$, where $S \subseteq R^D$

and D is the dimension of the decision variables. DE evolves a population of NP vectors representing the candidate solutions by the mutation, crossover and selection operators. Each vector is donated as $X_{i,G} = [x_{1,i,G}, x_{2,i,G}, ..., x_{D,i,G}]$, where $i = 1, 2, ..., NP$, NP is the size of the population and G is the number of current iteration.

Initialization: In DE, the initial population is generated by uniformly randomizing within the search space constrained by the prescribed minimum and maximum bounds. The jth parameter of $X_{i,G}$ is initialized as follows:

$$x_{j,i,G} = L_j + rand(0,1) \cdot (U_j - L_j) \tag{1}$$

wherer $rand(0,1)$ represents a uniformly distributed random number within the range [0, 1] and L_j and U_j represents the lower and upper bounds of the jth variable respectively.

Mutation: After initialization, DE generates a mutant vector $V_{i,G}$ with respect to each individual $X_{i,G}$ (called target vector) in the current population by the mutation strategy. The general notation for the mutation strategies is "DE/x/y", where DE stands for the differential evolution algorithm, x represents the vector to be perturbed and y represents the number of difference vectors considered for perturbation of x. There are various mutation strategies well-known and widely used in the literature, which are listed as follows:

- DE/rand/1

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) \tag{2}$$

- DE/best/1

$$V_{i,G} = X_{best,G} + F \cdot (X_{r1,G} - X_{r2,G}) \tag{3}$$

- DE/rand/2

$$V_{i,G} = X_{r1,G}+F\cdot(X_{r2,G}-X_{r3,G})+F\cdot(X_{r4,G}-X_{r5,G}) \tag{4}$$

- DE/current-to-best/1

$$V_{i,G} = X_{i,G}+F\cdot(X_{best,G}-X_{i,G})+F\cdot(X_{r1,G}-X_{r2,G}) \tag{5}$$

More details of them can be found in [1] and [2]. In order to have a better insight into these strategies, a general formula is introduced as follows [15]:

$$V_{i,G} = \beta_{i,G} + F \cdot \delta_{i,G} \tag{6}$$

where F is called the mutation scaling factor, $\beta_{i,G}$ is the base vector and $\delta_{i,G}$ is the difference vector. In Eq. (6), $\beta_{i,G}$ can be randomly or locally selected, and $\delta_{i,G}$ can be constructed in a random or directed manner [15].

Crossover: The crossover operator is applied to each pair of the target vector $X_{i,G}$ and the corresponding mutant vector $V_{i,G}$ to generate a trial vector $U_{i,G}$. There are two types of crossover scheme: binomial or exponential. The binomial crossover is widely used in the implementation of DE, which is outlined as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if} rand(0,1) \leq C_r \text{or} j = j_{rand}; \\ x_{j,i,G} & \text{otherwise.} \end{cases} \tag{7}$$

Selection: Following crossover, DE uses a one-to-one selection operator to select the better one between $X_{i,G}$ and $U_{i,G}$ to survive into the the next generation. The selection operator is described as follows:

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{if} f(U_{i,G}) \leq f(X_{i,G}); \\ X_{i,G} & \text{otherwise.} \end{cases} \tag{8}$$

B. NDi-DE

In order to fully and simultaneously exploit the neighborhood and direction information of population, NDi-DE synergizes two operators, NGS for selecting parents during mutation and DIM for guiding the mutation[9].

NGS: To utilize the neighborhood information of population to select the parents for mutation, NGS uses a probability selection operator based on the Euclidean distance. Specifically, the probability of each candidate vector is inversely proportional to its Euclidean distance from the target vector, which is calculated as follows:

$$Pro_{i,j,G} = 1 - \frac{\parallel X_{i,G}, X_{j,G} \parallel}{DIS} \tag{9}$$

where $||a,b||$ is the Euclidean distance between a and b, and DIS is the sum distance of all the individuals from $X_{i,G}$. In addition, to enhance the ability of exploring the search space around the promising vectors, the tournament best process is employed to decide the base vector from the selected vectors.

DIM: Based on the direction information with two different sources, DIM introduces three types of direction information, i.e., Directional Attraction (DA), Directional Repulsion (DR) and Directional Convergence (DC), for different mutation strategies to achieve a good tradeoff between exploration and exploitation (TEE). These two different sources are the best near-neighbor vector (denoted as $X_{att\_i,G}$), and the worst near-neighbor vector (denoted as $X_{rep\_i,G}$), which are determined in Eq. (10) and Eq. (11), respectively.

$$X_{att\_i,G} = \underset{X_{j,G}, j=1,...,NP \wedge j \neq i}{\arg\max} \frac{f(X_{i,G}) - f(X_{j,G})}{\parallel X_{i,G}, X_{j,G} \parallel} \tag{10}$$

$$X_{rep\_i,G} = \underset{X_{j,G}, j=1,...,NP \wedge j \neq i}{\arg\max} \frac{f(X_{j,G}) - f(X_{i,G})}{\parallel X_{i,G}, X_{j,G} \parallel} \tag{11}$$

With the different sources of near-neighbor, the three types of direction information play different roles to guide the search through the mutation strategy. That is, DIM adopts a new direction-inducing scheme to incorporate the defined direction information into the DE mutation strategy, as follows:

$$V_{i,G} = \beta_{i,G} + F \times \delta_{i,G} + DT_{i,G}, \tag{12}$$

where $DT_{i,G}$ means the type of direction information for $X_{i,G}$. Furthermore, in order to select the suitable type of direction information for the specific strategy, a guideline with three cases is developed based on TEE [9]. With the developed guideline, different mutation strategy equips with the direction information selected based on the case which it belongs to.

Issues Existed in NDi-DE: Although NDi-DE is demonstrated to be effective for most of the DE variants, the performance of NDi-DE heavily depends on the selection of direction information. As shown in [9], different mutation strategy equips with the different type of direction information based on its search characteristics. However, when applying NDi-DE to a new mutation strategy, how to select the suitable type of direction information is still a difficult problem. Therefore, AOS techniques are introduced in this paper to automatically select the direction information to enhance the performance and robustness of DE, specifically NDi-DE.

III. aNDi-DE

In order to automatically select the most suitable type of direction information for the specific DE mutation strategy, two AOS techniques, PM [13] and AP [14], are introduced into NDi-DE. By applying the AOS method to NDi-DE, aNDi-DE is developed. In this section, the details of the two AOS techniques and the direction information pool are firstly given. Then, the aNDi-DE framework by integrating AOS with NDi-DE is presented.

A. AOS

AOS is an adaptive control paradigm, which deals with the on-line selection among the available operators during the evolutionary process according to their rewards on the search up to now [16][17]. In AOS, there are two major phases: the *credit assignment* which defines how to turn the impact of the application of an operator into reward, and the *operator selection* that selects the operator to be applied based on their rewards. In this work, we focus on the two most promising selection mechanisms: PM and AP, which are the probability-based methods. In PM and AP, there are four steps, i.e., credit assignment, quality update and probability update and operator selection.

1) Credit Assignment: In the credit assignment, we adopt the fitness improvement rates (FIR) proposed in [18] to measure the quality for each type of direction information. Specifically, the FIR obtained by the ith type of direction information at time t is defined as follows:

$$FIR_{i,t} = \frac{pf_{i,t} - cf_{i,t}}{pf_{i,t}} \tag{13}$$

where $pf_{i,t}$ and $cf_{i,t}$ represent the fitness of the target vector and its offspring, respectively.

Then, at the end of each generation, the average value of all the FIR values achieved by applying the ith type of direction information is calculated, which is denoted as S_i. After that, the credit value (or reward) to the ith type of direction information is assigned as follows:

$$CV_{i,t} = \frac{S_i}{\sum_{j=1}^{K} S_j} \tag{14}$$

where K is the number of types of direction information.

2) Quality Update: Once the credit values of all the type of direction information are assigned, the quality (or estimated quality) of a direction information i, denoted as $q_{i,t+1}$, is updated as follows [14]:

$$q_{i,t+1} = (1 - \alpha) \times q_{i,t} + \alpha \times CV_{i,t} \tag{15}$$

where $\alpha \in [0, 1]$ is the adaptation rate.

3) Probability Update: Based on the updated quality of each type of direction information, PM and AP employ different approaches to update the selection probability.

- PM [13]
  In PM, the probability $p_{i,t+1}$ of the direction information i is updated as follows:
  $$p_{i,t+1} = p_{min} + (1 - K \times p_{min}) \times \frac{q_{i,t+1}}{\sum_{j=1}^{K} q_{j,t+1}} \tag{16}$$
  where $p_{min} \in [0, 1]$ is the minimal selection probability value of each direction information, which ensures that each direction information has a chance to become useful during the evolutionary process. The inefficient direction information will converge its probability towards p_{min} if it obtains no rewards for a long time, and the probability of the overwhelming direction information will converges to $p_{max} = p_{min} + (1 - K \times p_{min})$.
- AP [14]
  Originally proposed for learning automata, AP adopts a winner-take-all strategy to updated the selection probability, which works as follows:
  $$p_{i,t+1} = \begin{cases} p_{i,t} + \beta \times (p_{max} - p_{i,t}) & \text{if} \quad i = i^*_{t+1}; \\ p_{i,t} + \beta \times (p_{min} - p_{i,t}) & \text{otherwise.} \end{cases} \tag{17}$$
  where $i^*_{t+1} = \arg\max_{i \in \{1,...,K\}} \{q_{i,t+1}\}$ and $\beta \in [0, 1]$ is the learning rate, which is used to control the greediness of the winner-take-all strategy.

4) Operator Selection: With the updated probability of each direction information, both PM and AP use a roulette wheel-like method to select a type of direction information for the specific mutation strategy [13][14]. That is, the larger $p_{i,t}$ of the direction information, the higher of the possibility to be selected.

B. Direction Information Pool

In order to integrate AOS with NDi-DE for automatically selecting the type of direction information, we consider the three types of direction information proposed in [9], as well as no direction information option (i.e., direction without, DW). Therefore, the direction information pool is set as follows:

- DA:
  $$DA_{i,G} = I_{att} \times (X_{att\_i,G} - X_{i,G}) \tag{18}$$
- DR:
  $$DR_{i,G} = I_{rep} \times (X_{i,G} - X_{rep\_i,G}) \tag{19}$$
- DC:
  $$\begin{aligned} DC_{i,G} = I_{con1} \times (X_{att\_i,G} - X_{i,G}) - \\ I_{con2} \times (X_{rep\_i,G} - X_{i,G}) \end{aligned} \tag{20}$$

- DW:

$$DW_{i,G} = null \tag{21}$$

where I_{att}, I_{rep}, and $I_{con1}(I_{con2})$ are called the attraction, repulsion, and convergence scaling factors, respectively.

In these four types of direction information, DA, DR, DC and DW can play different roles to guide the search. That is, DA enhances the ability of vectors to exploit the promising regions of near-neighbor, DR prevents the vectors from entering the local optimum, and DC accelerates the convergence speed of vectors. For DW, it can maintain the search bias of the original mutation strategy when the direction information of near-neighbor cannot bring benefit to improving the performance. By introducing DW into the direction information pool, aNDi-DE can keep the advantages of the original mutation strategy and utilize the guiding information from DA, DR and DC.

C. The framework of aNDi-DE

Combining AOS with NDi-DE, the complete framework of aNDi-DE is summarized in **Algorithm 1** where the differences with respect to NDi-DE are highlighted with "$*$". From **Algorithm 1**, it is clear that aNDi-DE differs from the NDi-DE only in the selection of the direction information for the specific mutation strategy by PM or AP.

Algorithm 1 aNDi-DE

```
Generate the initial population P and set t = 1;
Evaluate the fitness for each individual in P;
* Set K, α, β and p_min;
While the terminated condition is not satisfied do
  For each individual X_{i,G} do
    * Choose a type of direction information based on
    the selection probability;
    Use NGS [9] and DIM [9] with the selected direc-
    tion information to generate a mutant vector;
    Use Eq. (7) to generate a trial vector;
    Use Eq. (8) to determine
    the survived vector;
  End For
  Set t = t + 1
  * Calculate the credit value CV_{i,t} for each type of
  direction information using Eq. (14);
  * Update the quality q_{i,t} using Eq. (15) and the prob-
ability value p_{i,t} for each type of direction information
using Eq. (16) (PM) or Eq. (17) (AP);
End while
```

IV. Simulation Results

In order to evaluate the performance of aNDi-DE, 14 benchmark functions are chosen from the CEC2005 test suite [19]. In this section, the benchmark functions are presented firstly. Secondly, the experimental setup is shown. Thirdly, the simulation results obtained from different experimental studies are analyzed and discussed.

A. Benchmark Functions

In this section, 14 benchmark functions are used, which are denoted as F1–F14. These functions are selected from the special session on real-parameter optimization of the 2005 IEEE Congress on evolutionary computation (CEC2005) [19]. They can be categorized into three groups: unimodal functions (F1–F5), basic multimodal functions (F6 – F12) and expanded multimodal functions (F13– F14). The short descriptions of these 14 functions are shown in Table I, and more details of them can be found in [19].

B. Experimental Setup

For a fair comparison, the same random initial population is used to evaluate different algorithms, and the parameters for all the experiments are set as follows unless a change is mentioned.

- Dimension of each function: $D = 30$;
- Population size: $NP = 100$;
- $F = 0.5$, $Cr = 0.9$;
- $\alpha = 0.8$, $\beta = 0.8$, $p_{min} = 0.1$;
- $I_{att} = I_{rep} = I_{con1} = I_{con2} = F/2$;
- Maximum number of function evaluations: $MNFEs = 10^4 \times D$;
- Number of runs: $NumR = 30$.

The control parameters in PM and AP are recommended by [14] and the scaling factors for different types of direction information are recommended by [9]. The influence of these parameters in aNDi-DE will be discussed in the future work. Furthermore, in order to show the significant differences among the algorithms, several nonparametric statistical tests [20] are also carried out by the KEEL software [21].

TABLE III

RANKS COMPUTED BY THE WILCOXON TEST FOR THE ANDI-DE WITH DIFFERENT AOS METHODS. •= THE METHOD IN THE ROW IMPROVES THE METHOD OF THE COLUMN. ○= THE METHOD IN THE COLUMN IMPROVES THE METHOD OF THE ROW. UPPER DIAGONAL OF LEVEL SIGNIFICANCE $\alpha = 0.1$, LOWER DIAGONAL LEVEL OF SIGNIFICANCE $\alpha = 0.05$.

	(1)	(2)	(3)	(4)
Ori (1)	-	13.5 ○	15.0 ○	15.0 ○
PM (2)	77.5 •	-	45.0	46.0
AP (3)	76.0 •	46.0	-	57.5
RN (4)	76.0 •	45.0	33.5	-

C. Comparison on different AOS methods

In order to compare the performance of aNDi-DE with different AOS methods, three aNDi-DE variants are considered as follows:

- aNDi-DE/AP: aNDi-DE uses AP mechanism for selecting the direction information.
- aNDi-DE/PM: aNDi-DE uses PM mechanism for selecting the direction information.
- aNDi-DE/RN: aNDI-DE uses the uniform strategy as baseline. In this variant, the probability of each direction information is equal and unchanged during the evolutionary process.

TABLE I

BENCHMARK FUNCTIONS

Test Function	Characteristics
F_1: Shifted Sphere Function	Shifted, separable, scalable
F_2: Shifted Schwefel's Problem 1.2	Shifted, nonseparable, scalable
F_3: Shifted Rotated High Conditioned Elliptic Function	Shifted, rotated, nonseparable, scalable
F_4: Shifted Schwefel's Problem 1.2 with Noise in Fitness	Shifted, nonseparable, scalable, noise in fitness
F_5: Schwefel's Problem 2.6 with Global Optimum on Bounds	Nonseparable, scalable
F_6: Shifted Rosenbrock's Function	Shifted, nonseparable, scalable, narrow valley from local to global optimum
F_7: Shifted Rotated Griewank's Function without Bounds	Rotated, shifted, nonseparable, scalable
F_8: Shifted Rotated Ackley's Function with Global Optimum on Bounds	Rotated, shifted, nonseparable, scalable
F_9: Shifted Rastrigin's Function	Shifted, separable, scalable, numerous local optima
F_{10}: Shifted Rotated Rastrigin's Function	Shifted, rotated, noseparable, scalable, numerous local optima
F_{11}: Shifted Rotated Weierstrass Function	Shifted, rotated, noseparable, scalable
F_{12}: Schwefels Problem 2.13	Shifted, nonseparable, scalable
F_{13}: Shifted Expanded Griewanks + Rosenbrocks Function	Shifted, nonseparable, scalable
F_{14}: Shifted Rotated Expanded Scaffers F6	Shifted, nonseparable, scalable

TABLE II

COMPARISON ON THE ERROR VALUES OF ANDI-DE WITH DIFFERENT AOS METHODS FOR ALL FUNCTIONS AT $D = 30$. THE NFES REQUIRED TO ACHIEVE THE ACCURACY LEVEL ARE ALSO SHOWN IN SQUARE BRACKETS WHEN THE COMPETITOR OBTAINS THE OPTIMUM VALUE WITHIN MNFES.

Func.	Ori	PM	AP	RN
F1	[106386±2145.683]	[35946±564.302]	[**32226±773.230**]	[37143±869.278]
F2	6.726e-005±4.427e-005	6.821e-014±2.313e-014	**6.253e-014±1.734e-014**	**6.253e-014±1.734e-014**
F3	4.187e+005±2.346e+005	**5.766e+004±3.162e+004**	5.922e+004±3.473e+004	7.021e+004±4.192e+004
F4	2.435e-002±2.385e-002	**6.689e-013±1.234e-012**	4.824e-012±2.301e-011	1.174e-011±1.809e-011
F5	4.411e+001±2.691e+001	8.471e+000±1.213e+001	7.325e+000±1.064e+001	**3.571e+000±5.009e+000**
F6	3.013e+000±1.312e+000	9.303e-001±1.715e+000	8.351e-001±1.516e+000	**7.973e-001±1.622e+000**
F7	4.696e+003±1.589e-002	4.696e+003±0.000e+000	4.696e+003±0.000e+000	4.696e+003±9.067e-005
F8	2.093e+001±6.812e-002	**2.093e+001±6.024e-002**	2.096e+001±3.429e-002	2.097e+001±4.701e-002
F9	1.295e+002±2.110e+001	**2.600e+001±7.603e+000**	2.793e+001±8.074e+000	2.723e+001±7.642e+000
F10	1.824e+002±1.152e+001	**3.206e+001±1.081e+001**	3.288e+001±9.474e+000	3.632e+001±1.374e+001
F11	3.967e+001±9.936e-001	1.404e+001±5.631e+000	**1.283e+001±5.414e+000**	1.322e+001±5.730e+000
F12	1.459e+003±3.308e+003	2.007e+003±2.694e+003	**1.610e+003±1.777e+003**	1.894e+003±3.148e+003
F13	1.521e+001±1.058e+000	**2.886e+000±6.069e-001**	3.114e+000±7.575e-001	2.899e+000±7.084e-001
F14	1.331e+001±1.443e-001	**1.268e+001±4.021e-001**	1.270e+001±4.971e-001	1.289e+001±3.679e-001
Avg_Rank	3.4643	2.1071	2.1429	2.2857

The comparisons among the aNDi-DE variants are carried out on the 14 benchmark functions at $30D$. The results are shown in Tables II where "Ori", "PM", "AP" and "RN" means the original algorithm, aNDi-DE/PM, aNDi-DE/AP and aNDi-DE/RN with DE/rand/1, respectively, and the convergence graphs for some selected functions are plotted in Fig. 1. The best performance for each functions is highlighted in **boldface** in the tables. From Table II, all the algorithms can obtain the equal optimum value within MNFEs for F1, and aNDi-DE/AP requires the leat number of function evaluations to achieve the accuracy level. For the unimodal functions (F1-F5), both aNDi-DE/PM and aNDi-DE/AP can obtain the best result on 2 functions. For the multimodal functions (F6-F14), aNDi-DE/PM can achieve the best result on 5 functions, while aNDi-DE/AP can obtain the best result on 2 functions.

In order to compare the performance of different algorithms overall, Table II summarizes the average rank values (avg_rank) in the table which are evaluated based on the descending order of the error values for all the functions. From Table II, aNDi-DE/PM obtains the best average rank among all the algorithms. To address the issue that an intelligent AOS method is better than a random one, aNDi-DE/PM and aNDi-DE/AP are also compared with aNDi-

TABLE IV

COMPARISON ON THE ERROR VALUES BETWEEN ANDI-DE AND NDI-DE WITH SINGLE DIRECTION INFORMATION FOR ALL FUNCTIONS AT $D = 30$. THE NFES REQUIRED TO ACHIEVE THE ACCURACY LEVEL ARE ALSO SHOWN IN SQUARE BRACKETS WHEN THE COMPETITOR OBTAINS THE OPTIMUM VALUE WITHIN MNFES.

func.	Ori	DA	DR	DC	PM
F1	[106386±2145.683]	5.053e-002±2.260e-001	2.084e-014±2.786e-014	[43566±1089.237]	**[35946±564.302]**
F2	6.726e-005±4.427e-005	7.439e-001±3.243e+000	1.307e-006±1.100e-006	**5.874e-014±1.818e-014**	6.821e-014±2.313e-014
F3	4.187e+005±2.346e+005	5.833e+004±1.143e+005	2.970e+005±1.500e+005	7.024e+004±3.443e+004	**5.766e+004±3.162e+004**
F4	2.435e-002±2.385e-002	1.365e-008±6.559e-008	1.454e-002±2.352e-002	1.078e-010±1.731e-010	**6.689e-013±1.234e-012**
F5	4.411e+001±2.691e+001	1.182e+003±4.018e+002	1.523e+002±1.314e+002	**5.461e+000±1.104e+001**	8.471e+000±1.213e+001
F6	3.013e+000±1.312e+000	2.583e+004±1.209e+005	5.728e+000±1.528e+001	**1.329e-001±7.279e-001**	9.303e-001±1.715e+000
F7	4.696e+003±1.589e-002	**4.696e+003±0.000e+000**	4.745e+003±1.532e+001	4.707e+003±8.143e+000	**4.696e+003±0.000e+000**
F8	2.093e+001±6.812e-002	2.096e+001±3.092e-002	2.094e+001±5.642e-002	2.094e+001±5.308e-002	**2.093e+001±6.024e-002**
F9	1.295e+002±2.110e+001	3.161e+001±9.678e+000	2.497e+001±8.025e+000	**2.325e+001±7.416e+000**	2.600e+001±7.603e+000
F10	1.824e+002±1.152e+001	3.807e+001±1.218e+001	3.452e+001±9.035e+000	9.398e+001±6.913e+001	**3.206e+001±1.081e+001**
F11	3.967e+001±9.936e-001	**1.197e+001±4.581e+000**	3.294e+001±1.196e+001	3.596e+001±9.900e+000	1.404e+001±5.631e+000
F12	1.459e+003±3.308e+003	1.045e+004±6.531e+003	1.304e+003±2.786e+003	**1.212e+003±1.801e+003**	2.007e+003±2.694e+003
F13	1.521e+001±1.058e+000	2.991e+000±6.763e-001	3.902e+000±3.324e+000	9.736e+000±5.635e+000	**2.886e+000±6.069e-001**
F14	1.331e+001±1.443e-001	**1.209e+001±4.208e-001**	1.338e+001±1.642e-001	1.330e+001±1.462e-001	1.268e+001±4.021e-001
Avg_Rank	3.8214	3.4286	3.2857	2.5000	**1.9643**

DE/RN. From Table II, we can find both aNDi-DE/PM and aNDi-DE/AP perform better than aNDi-DE/RN in terms of the average rank value. (i.e., 2.1071 *vs.* 2.2857, and 2.1429 *vs.* 2.2857 respectively). Furthermore, in order to show the significant differences among the algorithms, the results of the multiple-problem Wilcoxon signed-rank test [20] are also presented in Table III. As the results shown in Table III, all the aNDi-DE variants obtain higher $R+$ values than $R-$ values. Furthermore, all the aNDi-DE variants can improve DE/rand/1 both at level of significance $\alpha = 0.1$ and $\alpha = 0.05$.

Generally, all of the aNDi-DE variants are able to improve the performance of DE/rand/1, and aNDi-DE/PM is the best one in this comparison. The reason may lie in that the four types of direction information, i.e., DA, DR, DC and DW, are effective to guide the search through playing different roles.

TABLE V

RANKS COMPUTED BY THE WILCOXON TEST BETWEEN ANDI-DE AND NDI-DE WITH SINGLE DIRECTION INFORMATION. •= THE METHOD IN THE ROW IMPROVES THE METHOD OF THE COLUMN. ○= THE METHOD IN THE COLUMN IMPROVES THE METHOD OF THE ROW. UPPER DIAGONAL OF LEVEL SIGNIFICANCE $\alpha = 0.1$, LOWER DIAGONAL LEVEL OF SIGNIFICANCE $\alpha = 0.05$.

	(1)	(2)	(3)	(4)	(5)
Ori (1)	-	42.0	28.0	9.5 ○	13.5 ○
DA (2)	49.0	-	45.0	51.0	12.0 ○
DR (3)	63.0	60.0	-	24.0	17.0 ○
DC (4)	81.5 •	54.0	67.0	-	38.5
PM (5)	77.5 •	79.0 •	74.0 •	66.5	-

D. Comparison with NDi-DE with single direction information

In this section, to investigate the benefits obtained by using a pool of direction information in aNDi-DE, the comparisons between aNDi-DE/PM and NDi-DE with single direction information are carried out. For this purpose, three NDi-DE variants are used in this experiment. They are NDi-DE/DA, NDi-DE/DR and NDi-DE/DC which represents the NDi-DE with DA, DR, and DC, respectively. The results are presented in Table IV where "DA", "DR" and "DC" means NDi-DE/DA, NDi-DE/DR and NDi-DE/DC, respectively, and the convergence graphs for some selected functions are plotted in Fig. 1.

From Table IV and Fig. 1, it is clear that aNDi-DE/PM is the best one among all the algorithms. Specifically, aNDi-DE/PM obtains the best performance on 7 out of 14 functions, while NDi-DE/DA, NDi-DE/DR and NDi-DE/DC achieves the best results on 3, 0 and 5 functions respectively. In addition, to show the significant differences among different algorithms, the results of the multiple-problem Wilcoxon signed-rank test [20] are also presented in Table V. We can find that aNDi-DE/PM is significantly better than NDi-DE/DA and NDi-DE/DR. For NDi-DE/DC, aNDi-DE/PM does not significantly outperform it. It is worth noting that DC is the most suitable type of direction information for DE/rand/1 in the NDi-DE framework [9]. Therefore, these results indicate that aNDi-DE is effective to manage a pool of direction information by using AOS and is more robust than NDi-DE with using a single direction information.

E. Analysis of strategy adaptation

In order to study the adaptation characteristics of PM and AP, we analyze how the probability value of each direction information changes during the evolutionary process. In this experiment, the whole search process is divided into 50 phases and the probability value of each direction information in each phase is recorded and plotted in Fig. 2.

For F1, the adaptation trajectory of PM is similar to AP. That is, the probability values of the four direction information are oscillatory during the first 13 phases. After that, the trajectories of them are tending towards stability. In both aNDi-DE/PM and aNDi-DE/AP, DA dominates on phase 14 to phase 50. However, the difference between them during the evolutionary process is that PM converges rapidly and select the current best direction information in a much higher probability.

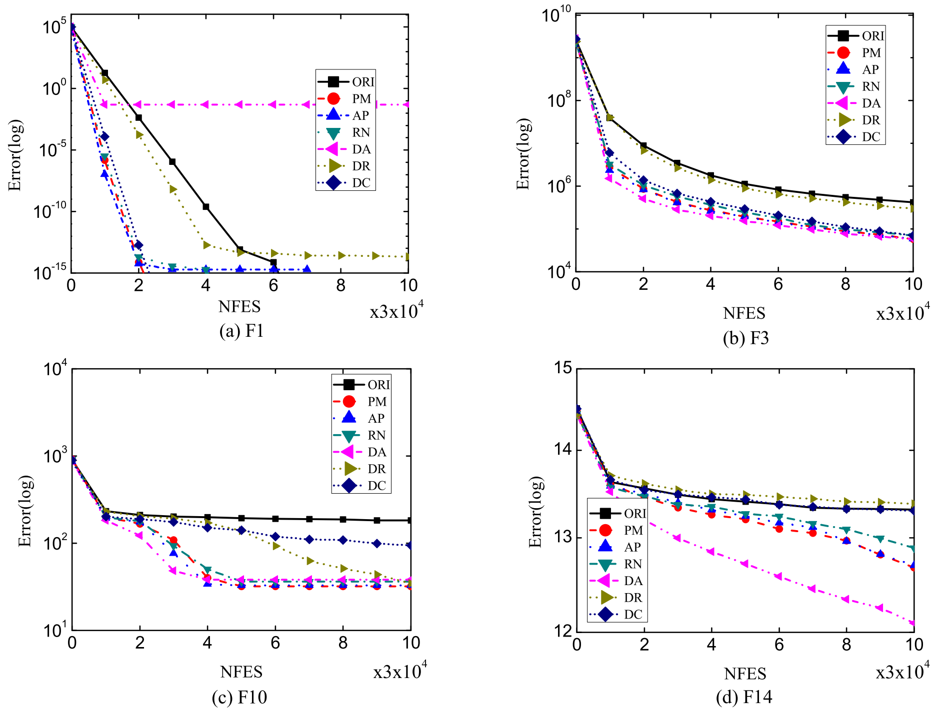

Fig. 1. Convergence graphs of different aNDi-DE variants for the selected functions at $D = 30$

For F10, when using AP as the adaptive mechanism, the probability values of the four direction information are oscillatory on stage 0 to stage 24, and the trajectories of them are tending towards stability on stage 25 to stage 50 on which DA dominates. When using PM, the probability values of them are oscillatory on stage 0 to stage 31, and DA dominates on stage 32 to stage 50.

According to the results shown Fig. 2, it can be observed that:

- As stated in [14] and [17], the AP technique converges more rapidly to a probability distribution than the PM method, which is confirmed in this experiment. However, aNDi-DE/PM is better than aNDi-DE/AP on the test functions. The reason may lie in that aNDi-DE/AP tend to select the most exploitive type of direction information in a much higher probability at the beginning of evolution. It will make the population become easier to trap in the local optimum.
- It is interesting to find that when the convergence graphs of aNDi-DE/PM and aNDi-DE/AP are unchanged, the adaptation trajectories of the direction information are also tending towards stability. It suggests that no type of direction information can dominate for a long time before the population converges.
- Fig. 2 clearly indicates that each type of direction information is useful for enhancing the search ability of NDi-DE, which is also confirmed in Section IV-D. In addition, Fig. 2 also demonstrates that NDi-DE at different evolution stages may need different search strategies.

V. Conclusions

In NDi-DE, the performance heavily depends on the selection of direction information, and how to select the suitable type of direction information is still a difficult problem. In order to address the exited issue in NDi-DE and enhance the performance and robustness of it, we propose a new method, adaptive direction information based NDi-DE (aNDi-DE) in this paper. In aNDi-DE, two adaptive operator selection (AOS) mechanisms, i.e., Probability Matching (PM) and Adaptive Pursuit (AP), are introduced to adaptively select the most suitable type of direction information for the specific mutation strategy. In this way, the good balance between exploration and exploitation can be dynamically achieved.

The experiments have been carried out on a suite of 14 benchmark functions to evaluate the effectiveness of aNDi-DE.Through the experimental study, we show that aNDi-DE with PM and AP is able to improve the performance of NDi-DE. In addition, the experimental results also demonstrate the effectiveness of the four types of direction information at different evolution stages.

In the future, aNDi-DE will be applied to other DE algorithms to test the effectiveness, and other AOS mechanisms (e.g., multi-armed bandit, MAB [22]) will also be studied in aNDi-DE.

References

[1] R. Storn and K. Price, " Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim*, vol. 11, no. 4, pp. 341-359, 1997.

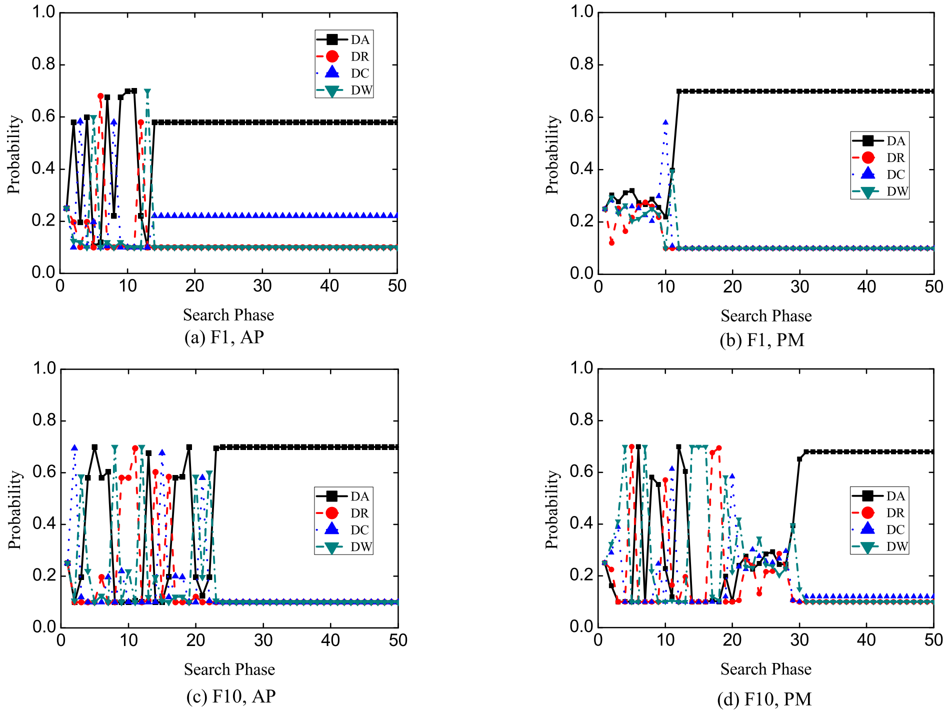

Fig. 2. Adaptation trajectory of the probability values for each direction information in aNDi-DE with AP and PM respectively.

[2] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[3] V. Plagianakos, D. Tasoulis and M. Vrahatis M, "A review of major application areas of differential evolution," *In: Advances in Differential Evolution*, U. Chakraborty, Ed. Berlin, Germany: Springer-Verlag, 2008, pp. 197-238.

[4] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1, pp. 61–106, 2010.

[5] A. Qin, V. Huang and P.N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans Evol Comput*, vol. 13, no. 2, pp. 398-417, 2009.

[6] R. Sarker, S. Elsayed and T. Ray, "Differential evolution with dynamic parameters selection for optimization problems," *IEEE Transactions on Evolutionary Computation*, doi: 10.1109/TEVC.2013.2281528.

[7] W. Yu, M. Shen, W. Chen, et al., "Differential evolution with two-level parameter adaptation," *IEEE Transactions on Cybernetics*, doi: 10.1109/TCYB.2013.2279211.

[8] J. Zhang and A. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[9] Y. Cai and J. Wang, "Differential evolution with neighborhood and direction information for numerical optimization," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp.2202 – 2215, Dec. 2013.

[10] Y. Cai, J. Wang and J. Yin, "Learning-enhanced differential evolution for numerical optimization," *Soft Computing*, vol. 16, no. 2, pp. 303–330, 2012.

[11] B. Xin, J. Chen, J. Zhang, et al., "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol.42, no.5, pp.744 –767, Sept. 2012.

[12] B. Dorronsoro and P. Bouvry, "Improving classical and decentralized differential dvolution with new mutation operator and population topologies," *IEEE Transactions on Evolutionary Computation*, vol.15, no.1, pp.67– 98, Feb. 2011.

[13] D. E. Goldberg, "Probability matching, the magnitude of reinforcement, and classifier system bidding," *Machine Learning*, vol. 5, pp. 407– 425, 1990.

[14] D. Thierens, "An adaptive pursuit strategy for allocating operator probabilities," *in GECCO05: Proc. 2005 Annual Conference on Genetic and Evolutionary Computation*, H. G. Beyer, Ed. ACM press, 2005, pp. 1539 – 1546.

[15] V. Feoktistov and S. Janaqi, "Generalization of the strategies in differential evolution," *in Parallel and Distributed Processing Symposium*, 2004, pp. 165–170.

[16] A.E. Eiben, Z. Michalewicz, M. Schoenauer and J.E. Smith, "Parameter control in evolutionary algorithms," *in: F.G. Lobo et al. (Eds.), Parameter Setting in Evolutionary Algorithms*, Springer, 2007, pp. 19 – 46

[17] W. Gong, A. Fialho, Z. Cai, and H. Li, "Adaptive strategy selection in differential evolution for numerical optimization: An empirical study," *Information Sciences*, vol. 181, no. 24, pp. 5364 - 5386, Dec. 2011 .

[18] K. Li, A. Fialho, S. Kwong and Q. Zhang, "Adaptive operator selection with bandits for multiobjective evolutionary algorithm based decomposition," *IEEE Transactions on Evolutionary Computation*, doi: 10.1109/TEVC.2013.2239648.

[19] J. Ronkkonen, S. Kukkonen and V. Price, " Real parameter optimization with differential evolution," *In Proc. IEEE Congr. Evol. Comput. (CEC-2005)* , vol. 1. Piscataway, NJ: IEEE Press, 2005, pp. 506-513.

[20] S. García, A. Fernández, J. Luengo and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability," *Soft Comput*, vol. 13, no. 10, pp. 959 – 977, 2009.

[21] J. Alcalá-Fdez, L. Sánchez and S. García, "KEEL: A software tool to assess evolutionary algorithms to data mining problems," [Online]. Available: http://www.keel.es/

[22] P. Auer, N. Cesa-Bianchi and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47 no. 2-3, pp. 235-256, 2002.