TADE: Tight Adaptive Differential Evolution

Weijie Zheng[1,2(✉)], Haohuan Fu[1(✉)], and Guangwen Yang[1,2(✉)]

[1] Ministry of Education Key Laboratory for Earth System Modeling, Center for Earth System Science, Tsinghua University, Beijing, China
haohuan@mail.tsinghua.edu.cn

[2] Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Computer Science and Technology, Tsinghua University, Beijing, China
zhengwj13@mails.tsinghua.edu.cn, ygw@mail.tsinghua.edu.cn

Abstract. Differential Evolution (DE) is a simple and effective evolutionary algorithm to solve optimization problems. The existing DE variants always maintain or increase the randomness of the differential vector when considering the trade-off of randomness and certainty among three components of the mutation operator. This paper considers the possibility to achieve a better trade-off and more accurate result by reducing the randomness of the differential vector, and designs a tight adaptive DE variant called TADE. In TADE, the population is divided into a major subpopulation adopting the general "current-to-*p*best" strategy and a minor subpopulation utilizing our proposed strategy of sharing the same base vector but reducing the randomness in differential vector. Based on success-history parameter adaptation, TADE designs a simple information exchange scheme to avoid the homogeneity of parameters. The extensive experiments on CEC2014 suite show that TADE achieves better or equivalent performance on at least 76.7 % functions comparing with five state-of-the-art DE variants. Additional experiments are conducted to verify the rationality of this tight design.

Keywords: Differential evolution · Differential vector · Adaptive

1 Introduction

Differential Evolution (DE), proposed in [1], is a simple and effective evolutionary algorithm to solve complex optimization problems. It has been shown to outperform some nature-inspired metaheuristics, such as genetic algorithm and particle swarm optimization over several benchmark functions [2], and has been adopted to various applications according to [3,4]. However, due to its stochastic nature, it suffers from long computing period and has the potential to improve the accuracy further. Since the mutation operator is the main engine that drives the population toward improvement, to achieve a more accurate and efficient DE algorithm, plenty of researches have been done based on its three components, the base vector, scaling factor and differential vector(s).

J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 113–122, 2016.
DOI: 10.1007/978-3-319-45823-6_11

The basic DE variants make the performance improvement mainly by reducing the randomness of base vector, such as DE/best/1, increasing the randomness of differential vector(s), such as DE/rand/2, or performing on both aspects, such as DE/best/2. The control parameters remain unchanged throughout the process. However, this certainty of the unchanged parameters makes it impractical due to high time cost of the required parameter tuning step, and this certainty somehow does harm to the performance since different parameter settings are fit for different stages. Therefore, many existing methods turned to introducing the randomness into the parameters and making them alterable and adaptable to different stages, such as jDE [6], NSDE [7], JADE [8] and SHADE [9] on a single mutation strategy, and CoDE [11], SaDE [10] and EPSDE [12] further combining multiple strategies. These adaptive (or self-adaptive) variants achieved more accurate result via increasing the randomness of search length, but still did not reduce the randomness of the differential vector.

These existing variants maintained the wide randomness on differential vector, which seems sensible since the randomness can ensure the possibility of reaching global optimum. However, it may waste time in the computation on the unpromising area, which results in its slow convergence. Introducing the certainty may result in a more accurate result in limited function evaluation times. Recently, our previous work [5] took the first step to reduce the diversity of the differential vector, and achieved a better result against DE/rand/1 and DE/best/1 on several benchmark functions. This shows the possibility of reducing the randomness of differential vector to achieve a more accurate result. However, although that work obtained a trade-off among the base vector and the differential vector, it maintained the certain scaling factor. As far as we know, there has not been any work considering the trade-off among all the three aspects while reducing the randomness of differential vector to get a competitive result.

In this paper, we firstly propose a novel DE mutation strategy. It takes current-to-*p*best as the base vector, maintains the random choice of the starting point of differential vector and adopts the current (target) vector as the ending point. Due to the reduced randomness in the search direction, this greedy mutation may lead to premature and is unfit to drive the whole population. Therefore, it is then utilized as the engine of a minor subpopulation. The major subpopulation is evolved via current-to-*p*best [8] which has the same base vector and can share the exploration information in time with the minor subpopulation. Both subpopulations adopt the success history based parameter adaption [9], and the exchange of the two subpopulations is designed to further enhance the diversity of the scaling factor. Extensive experiments are conducted to compare this Tight Adaptive DE (TADE) with five state-of-the-art DE variants (SHADE, JADE, CoDE, EPSDE and SaDE) on the CEC2014 benchmark suite.

The contributions of this paper can be summarized as follows:

- This paper designs a tight adaptive DE scheme, which includes a proposed mutation reducing the randomness in differential vector, the information exchange on mutation strategies to get out of the local optima, and the information exchange on control parameters to enhance the randomness

– TADE achieves better or equivalent performance on at least 76.7 % functions comparing with five state-of-the-art DE variants
– Verification for the rationality of our tight design is conducted by the experiments varying partition and parameter exchange rate

Organization of the rest paper is as follows. The brief introduction of Differential Evolution and an insight view are shown in Sect. 2. Section 3 discusses the motivation and detail of the proposed method. Extensive experiments and analysis are conducted on Sect. 4. Finally, Sect. 5 concludes the paper and discusses the future work.

2 Differential Evolution

This section briefly describes the framework and an insight view of DE algorithm. DE undergoes mutation, crossover and selection operators iteratively until satisfying the accuracy condition or reaching a predefined function evaluation times FES_{max}. The random change happens in mutation so that every candidate has the opportunity to enter the next generation and get itself inherited. Crossover operator generates the trial vector u_i^g that exchanges the information of the mutant vector v_i^g with the target vector x_i^g and further widen the diversity. Then the selection operator is employed to preserve the most promising vector entering the next generation, ensuring the non-degeneration evolution process. Obviously, mutation is the main engine to pull the population to improvement.

All mutation operators are composed of three parts: the base vector, differential vector(s) and the search length (scaling factor). Taking "current-to-$p$best" in JADE [8] as an example:

$$v_i^g = x_i^g + F_i(x_{pbest}^g - x_i^g) + F_i(x_{r_1}^g - \tilde{x}_{r_2}^g) \tag{1}$$

The mutation happens around the neighborhood of the base vector $x_i^g + F_i(x_{pbest}^g - x_i^g)$, giving a rough guess on where the promising search area is. Then the differential vector $x_{r_1}^g - \tilde{x}_{r_2}^g$ determines the search direction, and the scaling factor F_i controls how far it will search along the direction.

Throughout the DE developing history, many existing methods can be regarded as looking for a more accurate result via achieving a better tradeoff between the randomness and the certainty. Randomness represents the huge diversity of the offspring candidates that can ensure the possibility of reaching the global minimum, while certainty carries the information which leads to a strategy or belief on where the promising area is. Therefore, although randomness maintains the possibility, the huge area to search may cause inefficiency, and although certainty provides relatively smaller area and processes faster, it may cause premature due to its myopia or greediness.

Specifically, as for the base vector, differential vector(s) and the search length (scaling factor) in mutation, the classical operators discuss the first two compositions but fix the scaling factor. DE/rand/1 maximizes the randomness of both

Algorithm 1. General framework of TADE

– *Initialization*

$g = 0, partition = 8/10, exR = 0.3$, Archive $A = \emptyset$;
Index counter $k_1 = k_2 = 1$;, $N_1 = partition * N$, $N_2 = N - N_1$;
$M_1 = \{(0.5, 0.5)_i | i = 1, ..., N_1\}$, $M_2 = \{(0.5, 0.5)_i | i = 1, ..., N_2\}$
$E_1 = \{N_1 - N_2 * exR + 1, ..., N_1\}, E_2 = \{N_1 + 1, ..., N_1 + N_1 * exR)\}$;
Initialize population $P^0 = \{x_1^0, ..., x_N^0\}$, evaluating P^0, $FES = N$
where major subpop $P_1^0 = \{x_1^0, ..., x_{N_1}^0\}$; minor subpop $P_2^0 = \{x_{N_1+1}^0, ..., x_N\}$;

– *Evolution*

while $FES < FES_{max}$ **do**
  $S_1 = S_2 = \emptyset$;
  for i = 1,2 **do**
    for j in P_i^g **do**
      generate (F_j^g, CR_j^g) from M_i;
      generate *p*best from P^g;
      generate mutator v_j^g by (eq.i);
      generate trial vector u_j^g
      if $f(x_j^g) > f(u_j^g)$ **then**
        $x_j^{g+1} = u_j^g$; $x_i^g \rightarrow A$;
        $(F_j^g, CR_j^g) \rightarrow S_i$;
      else
        $x_j^{g+1} = x_j^g$;
      end if
    end for
  end for
  for $i = 1, 2$ **do**
    if $S_i \neq \emptyset$ **then**
      $AS = \{(F_j^g, CR_j^g) \in S_{3-i} | j \in E_{3-i}\}$
      Update M_i based on $S_i \cup AS$
      $k_i = (k_i + 1) mod N_i$
    end if
  end for
  $g = g + 1$, $FES = FES + N$
end while

parts, making it a most robust one, and DE/best/1 believes that the promising offsprings may be more likely to surround the best vector, and utilizes this certainty to design the best vector as the base vector and achieves a greedy but more rapid process that helps to reach a more accurate result in a limited time. Recently, our previous work further reduced the uncertainty of the differential vector, and get a competitive performance over DE/rand/1 and DE/best/1 on several functions. The basic operators hold the certainty that the control parameter is constant in the whole period. However, some randomness adding into this certainty indeed improves the performance, like jDE, JADE, SHADE and other adaptive methods.

3 TADE

Special attentions are given to JADE and SHADE. JADE increased the randomness to overcome the premature caused by greedy base vector. Firstly, JADE utilized a random vector from top $p\%$ best individuals to replace the greedy best vector in "current-to-best", which introduced the uncertainty of base vector. Moreover, JADE added an archive A to the population when generating the differential vector, which increased the randomness of candidate search directions. Besides, JADE used the current succeed parameters to partly influence the new ones, and added more uncertainty like Gaussian or Cauchy distribution. These strategies well alleviated the greediness of the base vector and enlarged the diversity. SHADE, the work based on JADE, further increased the diversity of the parameter. Instead of the current succeed parameters, SHADE maintained a historical succeed parameters, which can bring a wider randomness.

Since our previous work [5] found the possibility of reducing the randomness of the differential vector, this paper discussed whether it can be combined with the trade-off of other two aspects to achieve a more accurate result. We firstly propose a novel mutation strategy reducing the randomness of the differential vector. Different from [5] that selected the target vector as the starting point of the differential vector and maintained the randomness of the ending point, this mutation maintains the uncertainty of the starting point and takes the target vector as the ending. For the base vector, we adopt "current-to-*p*best" vector to increase the diversity and avoid the myopia of best vector. This mutation strategy is noted as "current-to-*p*best(half-rand)", and can be written as

$$v_i^g = x_i^g + F_i(x_{pbest}^g - x_i^g) + F_i(x_i^g - \tilde{x}_{r_2}^g) = x_i^g + F_i(x_{pbest}^g - \tilde{x}_{r_2}^g) \tag{2}$$

The proposed mutation strategy largely reduces the number of possible candidate directions from $(N-1)(N+|A|-2)$ to $N+|A|-1$ for the current population. It is unfit to drive all or even the majority of the population to evolve since too much searching area has been cut. Therefore, this mutation strategy could act on a small part of the population as a pioneer soldier to rapidly explore on a narrow area. The majority of the population is controlled by a relatively farsighted "current-to-pbest(rand)" (that is the "current-to-pbest" in JADE and (rand) represents both randomness in differential vector). The reason of this choice is that "current-to-pbest(rand)" and "current-to-pbest(half-rand)" have the same base vector, the explored information from the pioneer soldier can easily feedback to the farsighted commanding officer in the next generation, and the officer can give a real-time and global-view new command with no extra cost to change the local search area the pioneer will explore in the next generation.

As for the parameter adaption, both parts adopt the way in SHADE due to its larger diversity. The basic thought of SHADE maintained the historical succeed parameters of several generations. Simply, the major and minor subpopulation update the success history merely by their own success parameters. However, with the iterative evolution, each element in the memory may become homoplastic, which hurts the diversity of search length and becomes harder to jump out. If the homoplasty can be postponed, larger area may be explored and a more accurate result may be achieved. Therefore, an exchange on the subpopulation to update the success memory is designed due to the inherently different mechanism of two mutation strategies. Specifically, an exchange rate exR is designed to determine the proportion of one subpopulation that will be used for the other subpopulation when updating the success memory.

The framework of this tight adaptive DE, called TADE, is shown in Algorithm 1. In *Evolution* phase, Lines 2–16 undergo the general process of DE for major and minor subpopulations. In line 6, the mutation information exchange happens since $p$best for each subpopultion is generated from the whole population P^g. Lines 17–23 update the success memory for both subpopulations. The success parameter exchange happens in Lines 19–20. Taking updating M_1 for major subpopulation as an example, the set AS is from minor success parameters and there are at most $|E_2| = N_1 * exR$ success parameters in the minor subpopulation that will join in the success memory update for the major subpopulation.

The Tight in TADE reflects on two aspects: the real-time information exchange on generating base vector, and the influence of success parameters in one subpopulation to the other. Next section shows the performance and rationality of this tight trade-off among base vector, scaling factor and differential vector.

4 Experiments

4.1 Settings

CEC2014 [13] benchmark suite is employed to demonstrate the performance comparison. Functions in this suite are all single objective optimization problems, containing both unimodal (F1-F3) and multimodal functions (F4-F30). More specifically, F4-F16 are simple multimodal functions, F17-F22 are hybrid functions and F23-F30 are composition functions. For each problem, 51 independent runs are conducted to obtain a reliable result. The dimension D is all set to be 30. The evolution process ends when the function evaluation time reaches $FES_{max} = 10000 * D$, and the error value smaller than 10^{-8} is taken as 0 [13].

4.2 Comparison and Analysis

Five state-of-the-art DEs, SHADE [9], JADE [8], CoDE [11], EPSDE [12] and SaDE [10], are utilized to compare with TADE. The settings of comparing methods are configured the same as the related papers. In TADE, the total population size N=100, the population partition is set to 8:2 for the major and minor subpopulation, and the success memory updating exchange rate $exR = 0.3$.

Wilcoxon's rank sum test at 5 % significance level is conducted between TADE and the comparing methods to measure whether TADE can obtain a significantly superior result or not. When TADE performs significantly better, "−" is marked, and "+" when TADE performs significantly poorer. "=" is marked when there is no significant difference. Due to the limited space, the detailed mean error value and the standard deviation of 51 independent runs are provided in http://thuhpgc.org/images/8/8e/Sup.jpg, but the number of cases on different function categories that TADE achieves better "−", equivalent "=" and worse "+" results against the comparing method are summarized in Table 1.

(a) Comparison with SHADE and JADE. From Table 1, TADE shows a quite competitive performance over SHADE, and it achieves better results on 12 functions and loses 7 functions. The attention should be paid to the unimodal as well

Table 1. Experimental results over 51 independent runs on unimodal functions of 30 variables with 300 000 FES

Function type	SHADE	JADE	CoDE	EPSDE	SaDE
Unimodal (F1-3)	$1^-2^=0^+$	$2^-1^=0^+$	$1^-2^=0^+$	$1^-2^=0^+$	$2^-1^=0^+$
Simple multimodal (F4-16)	$3^-6^=4^+$	$9^-3^=1^+$	$5^-5^=3^+$	$11^-2^=0^+$	$11^-1^=1^+$
Hybrid (F17-22)	$3^-2^=1^+$	$5^-1^=0^+$	$3^-1^=2^+$	$6^-0^=0^+$	$5^-1^=0^+$
Composition (F23-30)	$5^-1^=2^+$	$3^-4^=1^+$	$5^-2^=1^+$	$3^-0^=5^+$	$6^-1^=1^+$
Total	$12^-11^=7^+$	$19^-9^=2^+$	$14^-10^=6^+$	$21^-4^=5^+$	$24^-4^=2^+$

as the hybrid and composition functions. Indeed on the unimodal function F1, the mean error of SHADE is around $E{+}02$ and that value is $E{-}07$ for TADE, which demonstrates TADE's superior performance. For the hybrid and composition functions, the better cases TADE achieves are quite more than the better cases SHADE achieves. The difference between TADE and SHADE lies on the proposed mutation strategy for subpopulation and the mechanism preventing from homogeneity of the success memory. The real-time information exchange of two different mutations can prevent the whole population from being stuck in a local area, and the parameter information exchange prolongs the coming time of homogeneity, ensuring more time for exploration. These are the essential reasons why TADE performs better than SHADE. Also from Table 1, TADE shows an overwhelming superiority over JADE on every category. This advantage comes partly from the success history inherited from SHADE that widens the diversity, and partly from the tight major-minor scheme that makes a clear division of labour and improves the whole generation more effectively.

(b) Comparison with CoDE, EPSDE and SaDE. This comparison shows the performance against some state-of-the-art variants that combined the strengths of several mutation strategies. These methods employed a wide diversity of mutation strategies but with no tight information exchange. From Table 1, we see an overwhelming superiority of TADE over these methods, especially EPSDE and SaDE. A major contribution comes from the real-time information exchange in TADE, and the clear division and role of different methods help to achieve a win-win situation that pulls the generation towards improvement.

(c) Overall Comparison. From Table 1, TADE performs better or equivalently on at least 23(76.7 %) functions when compared with these five methods, and that number can reach 28(93.3 %) when compared with JADE and SaDE. The strength of the tight and cooperative scheme can be seen especially in the unimodal and the hybrid functions. The unimodal functions are relatively simple and the greedy "current-to-pbest(half-rand)" can rapidly explore the promising simple area. For hybrid functions that are usually the sum of several functions, two different mutations work for different components and search areas, and are tightly cooperated as well, thus resulting in the fast searching ability and enabling the possibility of reaching more promising area.

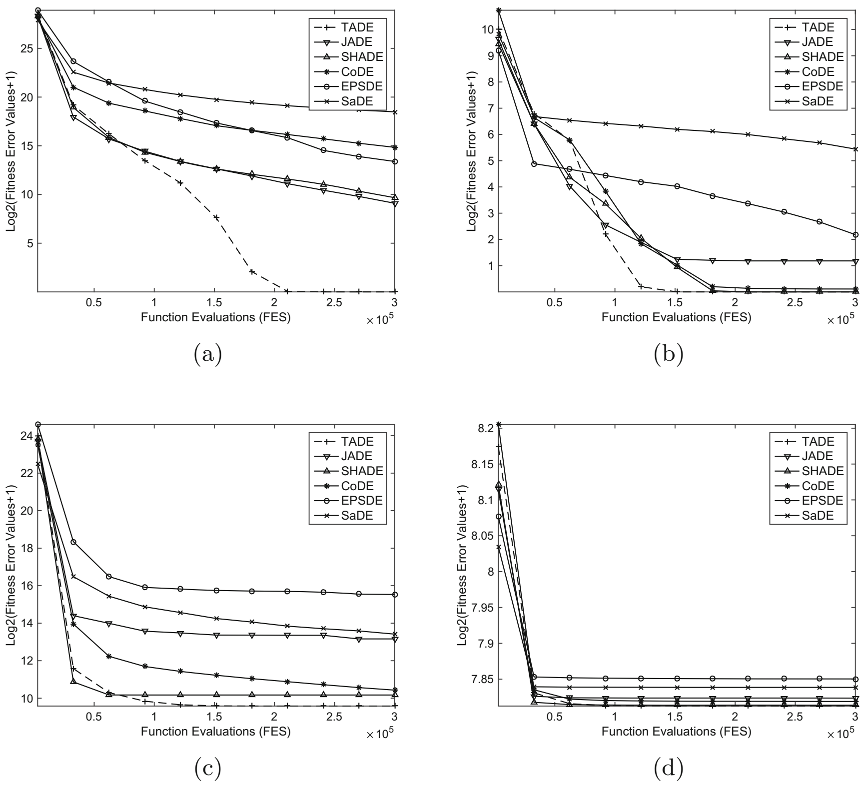

Fig. 1. Convergence curves of six DE variants on (a) F1, (b) F4, (c) F17, (d) F24.

More visually, Fig. 1 plots the convergence curves of these six methods on four functions selected from four categories. We can see the overwhelming superiority of TADE on Fig. 1(a), where TADE reaches several-magnitude better result with much rapid convergence speed than other methods. From Fig. 1(b), TADE and SHADE outperform the others, but TADE has a rapid speed and achieves the minimum ahead of SHADE. In Fig. 1(c), all six methods have almost the same speed at the early phase of the process, but TADE can reach a further accurate result due to the tight cooperative scheme that prevents the whole population from being all trapped into the local minima. TADE reaches competitive results in Fig. 1(d) against SHADE and CoDE. In a word, the strength of TADE is further shown from these curves.

4.3 Rationality of Our Tight Design

The experiments on different subpopulation partitions as well as different exchange rates are conducted to demonstrate the rationality of the tight designed scheme. Table 2 shows the results of 51 independent runs on different partitions

Table 2. Different partitions

Part	1:9	3:7	5:5	7:3	8:2	9:1
	21	18	13	4	−	7
	6	6	12	25	=	21
	3	6	5	1	+	2

Table 3. Different exchange rates

exR	0.0	0.1	0.2	0.3	0.4
	8	3	0	−	0
	20	27	30	=	30
	2	0	0	+	0

from 1:9 to 9:1 with fixed $exR = 0.3$, and Table 3 shows the comparison results on different exchange rates from 0.0 to 0.4 with 8:2 population partition.

From Table 2, when the proportion of the proposed "current-to-*p*best (half rand)" becomes larger (*Part* from 5:5 to 1:9), the performance becomes worse, showing the greediness of this mutation. Therefore, the design is reasonable that this greedy mutation is assigned with a smaller subpopulation. The effect of this greedy but fast pioneer is to get a fast feedback of its exploring area to the dominant role, "current-to-*p*best (rand)". This design can prevent the whole population from being trapped in the local place, and can improve the efficiency as well. The comparison result against 10:0 partition, that is the comparison with SHADE in Table 1, $12^{-}11^{=}$ and 7^{+}, shows the actual effect of the pioneer discussed before and shows a win-win performance of both mutation strategies.

From Table 3, when $exR = 0.0$, which means no exchange between two subpopulations, this setting achieves 8 worse cases and only 2 better cases, which verifies TADE's parameter exchange can delay the homogeneity to some degree and results in the exploration of larger promising area. When the rate is from 0.2 to 0.4, there is no significant difference because almost every success setting of the minor subpopulation will join in the success history updating of the major one since there are only 20 individuals of minor subpopulation in this 8:2 partition.

5 Conclusion and Future Work

Generally, DEs maintain both randomness of the vectors that generate the differential vector to ensure the wide range of the candidate search directions. The competitive performance of our previous work on reducing the randomness of differential vector leads to a new thought on existing variants. The engine mutation operator of DE has three aspects, base vector, scaling factor and differential vector(s). The competitive performance and the behavior of existing variants can be explained by the trade-off of the randomness and certainty among these three components. However, existing methods only considered the trade-off without reducing the randomness of differential vector. This paper designed tight adaptive DE (TADE) that took the randomness-reduced differential vector into account. This proposed half-rand mutation was used to guide a minor subpopulation while the majority was leaded by the general "current-to-*p*best". The same base vector in both mutation helped to share the exploration information timely, and difference prevented from all being trapped in the local area. Moreover, based on success-memory parameter adaption, this paper designed a parameter information exchange scheme to delay the homogeneity and premature.

The extensive experiments in this paper were conducted on 30 CEC2014 benchmark functions. Firstly, TADE was compared with five state-of-the-art DE variants, SHADE, JADE, CoDE, EPSDE and SaDE. TADE showed a competitive performance against SHADE, and a superior performance against other four methods. Secondly, for the two tightness in this design, the different partitions and exchange rates were conducted. The results demonstrated the rationality of this design and reflected the characteristics of the proposed mutation strategy.

In the future, other basic designs and adaptive schemes on the randomness-reduced differential vector are encouraged to achieve a better performance.

Acknowledgments. This work is supported in part by National Natural Science Foundation of China (Grant Nos. 61303003, 41374113), by Tsinghua University Initiative Scientific Research Program (Grant No. 20131089356).

References

1. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
2. Vesterstrom, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: IEEE CEC, vol. 2, pp. 1980–1987 (2004)
3. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. **15**(1), 4–31 (2011)
4. Das, S., Mullick, S.S., Suganthan, P.N.: Recent advances in differential evolution – an updated survey. Swarm Evol. Comput. **27**, 1–30 (2016)
5. Zheng, W., Fu, H., Yang, G.: Targeted mutation: a novel mutation strategy for differential evolution. In: IEEE ICTAI, pp. 286–293 (2015)
6. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. **10**(6), 646–657 (2006)
7. Yang, Z., Yao, X., He, J.: Making a difference to differential evolution. In: Siarry, P., Michalewicz, Z. (eds.) Advances in Metaheuristics for Hard Optimization. NCS, pp. 397–414. Springer, Berlin (2007)
8. Zhang, J., Sanderson, A.C.: JADE: adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. **13**(5), 945–958 (2009)
9. Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for differential evolution. In: IEEE CEC, pp. 71–78 (2013)
10. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comput. **13**(2), 398–417 (2009)
11. Wang, Y., Cai, Z., Zhang, Q.: Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans. Evol. Comput. **15**(1), 55–66 (2011)
12. Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl. Soft. Comput. **11**(2), 1679–1696 (2011)
13. Liang, J.J., Qu, B.Y., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Technical report, Zhengzhou University and Nanyang Technological University (2013)