IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 1, March 2013
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org

Differential Evolution with Tournament-based Mutation Operators

Chen Qiu[1], Miaomiao Liu[2] and Wenyin Gong[3*]

[1] **School of Computer, China university of Geosciences,
Wuhan, 430074, P. R. China**

[2] **School of Computer, China university of Geosciences,
Wuhan, 430074, P. R. China**

[3]**School of Computer, China university of Geosciences,
Wuhan, 430074, P. R. China**
*Corresponding author

Abstract

Differential Evolution(DE) has emerged as a powerful and efficient evolutionary algorithm for solving global optimization problems. It adopts the stochastic searching method to make selection of the parents in the mutation operator, which benefits the search of global optimization value. However, the selection method reveals the convergence in low speed. So for the sake of better convergence performance, in this paper, we propose the Tournament-based mutation operators to accelerate the differential evolution. The proposed algorithm employs the tournament selection for mutation. The process of tournament-based mutation operators is that the base and differential vectors are replaced by the tournament best vector but other vectors are randomly selected. It is helpful to improve the convergence besides maintain the diversity of DE algorithms. We also integrate the algorithm into jDE to verify the effect on it. Experimental results indicate that our proposed tournament-based mutation operators are highly competitive to the original DE algorithm and are able to enhance the performance of jDE.

Keywords: *Differential evolution, tournament-based, mutation operator,the tournament selection.*

1. Introduction

DIFFERENTIAL EVOLUTION(DE) was proposed by Price and Storn in 1995[1][2].It is a simple yet efficient evolutionary algorithm which only has a few control parameters. As a emerging technology, it has been successfully applied to diverse domains of science and engineering, such as engineering optimal design, digital filter design, learn-able evolution model, image processing[3], signal processing[4], machine intelligence, and pattern recognition[5],[6]etc.

According to frequently reported studies, DE[7] has superiority in the diversity and robustness over benchmark and real-word problems than many other algorithms. But there is still a performance drawback. As we know, the parents in the mutation operator are chosen randomly from the current population. It helps in exploring the search space and locating the region of global minimum but be slow at convergence speed. There are some hybridization techniques proposed by other researchers to improve the mutation operator, such as Trigonometric mutation operator proposed by Fan and Lampinen[8], neighborhood search proposed by Yang *et al.*[9]. and one-step-K-means proposed by Cai *et al*[10] *etc.*

The Tournament Selection is a popular method of selecting an individual from a population of individuals in a genetic algorithm(GAs)[11]. Kaelo and Ali [12] proposed some modified differential algorithms referred from the tournament selection. They incorporated the tournament selection into the classical DE algorithm. The main idea behind modified algorithm is the winner of three random ones is selected as base vector and the remaining two are contributed to the differential vectors. Motivated by these considerations, standard tournament selection is introduced to differential evolution. Instead of selecting the three vectors randomly from the whole population, we make the base and difference vectors being selected. The process of selecting is repeatedly obtaining the best within three random vectors until the both vectors are assigned. Different from selecting the best of three random ones as base vector in [12], the tournament-based mutation operators make the base and difference vectors being selected and obtain these vectors by a repeating process. Our objective in this search is to observe the combined effect between the tournament selection and DE. Further, In order to investigate the effect of fusion, in this paper we combine the tournament selection with jDE[13] , which is the latest modifications of DE. We have discussed the improvements when they are fused together in tournament selection. Meanwhile, we perform series of contrast experiments to compare the tournament-based mutation operator with the modified algorithm proposed by Kaelo and Ali[12].

The remainder of the paper is structured as follows. Section 2 describes the basic Differential Evolution and some related work including the concept of tournament selection. In Section 3 we introduce the proposed tournament-based mutation operators. Performance metrics and experimental settings are given in Section 4. Sample graphs (mid-value) for performance comparison between the tournament-based mutation operators and the corresponding original algorithms are also been represented in Section 4. Finally, in Section 5. we will provide our conclusion.

2. Related Work

For the completeness of this paper a brief describe of DE is given firstly. Then we introduce some related work including some mutation operators and the concept of tournament selection.

Algorithm 1 The DE algorithm with "DE/rand/1/bin" strategy

Generate the initial population randomly
Evaluate the fitness for each individual in the population
while the stop criterion is not satisfied **do**
  for i=1 to NP **do**
    Select uniform randomly $r_1 \neq r_2 \neq r_3 \neq i$
    j_{rand} =rndint$(1, D)$
    for j=1 to D **do**
      if $rndreal_j[0,1) < CR$ or j is equal to j_{rand} then
        $u_{i,j} = x_{r_1,j} + F \cdot (x_{r_2,j} - x_{r_3,j})$
      else
        $u_{i,j} = x_{i,j}$
      end if
    end for
  end for
  for i=1 to NP **do**
    Evaluate the offspring u_i
    if $f(u_i)$ is better than **or** equal to $f(x_i)$ **then**
      Replace x_i with u_i
    end if
  end for
end while

2.1 Differential Evolution

DE algorithm is a population-based one in global optimization, with the merits of being easy to use and fast convergence. The mechanism of population-based algorithm is generating new points by combining the parent individuals and several other individuals in the same population and attempting to replace the original ones if the new point is superior to the original one. Owing to the minimize functions that we are researching, The definition of better individual is having an equal or lower fitness value. The pseudo-code of original DE is shown in Algorithm 1. The meaning of those signs in Algorithm 1 are as follows: 1) D is the number of decision variables.2)NP is the population size. 3) F is the mutation scaling factor.4)CR is the crossover rate.5)$x_{i,j}$ is the j-th variable of the solution xi;6)u_i is the offspring.

In the mutation phase, DE randomly selects three distinct points from current population called x_{p1}, x_{p2} and x_{p3}. Meanwhile, none of these points should be the same as the current target point x_i. Many mutation strategies to generate new points are available. In Algorithm 1, we employ the classical "DE/rand/1" mutation strategy.

2.2 Mutation Operators in DE

In our comprehensive experiments, we employ several mutation strategies to test the proposed algorithm performance of generality. In this section, we introduce these mutation strategies which have been employed[14],[3]. In order to distinguish among DE's mutation operators, the notation "DE/a/b" is used, where "DE" stands for the Differential Evolution; "a" denotes the vector to be mutated; and "b" is the number of difference vectors being used. The four employed typical mutation strategies are as follow:

1) "DE/rand/1":

$$v_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}) \quad (1)$$

2) "DE/rand/2":

$$v_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}) + F \cdot (x_{r_4} - x_{r_5}) \quad (2)$$

3) "DE/current-to-best/2":

$$v_i = x_i + F \cdot (x_{best} - x_i) + F \cdot (x_{r_2} - x_{r_3}) + F \cdot (x_{r_4} - x_{r_5}) \quad (3)$$

4) "DE/rand-to-best/2":

$$v_i = x_{r_1} + F \cdot (x_{best} - x_{r_1}) + F \cdot (x_{r_2} - x_{r_3}) + F \cdot (x_{r_4} - x_{r_5}) \quad (4)$$

x_{best} represents the best individual in the population. $r1, r2, r3, r4$ and $r5$ $\in \{1, \cdots$,NP$\}$, and $r1 \neq r2 \neq r3 \neq r4 \neq r5 \neq i$ .In the formula of Eq.(3), x_i is referred to as the *target* vector; u_i is the *trial* vector; v_i is the *mutant* vector; x_{r_1} is the *base* vector; and $x_{r_2} - x_{r_3}$ are the *differential* vectors.

2.3 The Tournament Selection

In this paper, we propose the tournament-based mutation operators to accelerate the differential evolution. The tournament algorithm[11] is a popular method of selection which is commonly used with genetic algorithm, which involves a parameter to determine the number of exploring individuals at random. It can be called Tour and we fix the size of Tour as three. The process of the tournament algorithm is getting tour individuals uniformly at random. Then the best among these individuals is chosen for vector. The process is repeated the number of times necessary to reach the desired size of the vectors, which are meaning of the base vector and difference vectors in this paper.

3. The Tournament Selection in DE

In the original DE all the vectors are chosen at random for mutation. This has a exploratory effect but it slows down in convergence of process.In order to obtain better convergence performance, in this section, we propose the tournament selection to enhance the exploitation ability. The base vector and difference vectors are replaced by the tournament best vector, which is selected from three random individuals. The remaining vectors are generated by selecting at random. The key points of our approach are described in detail as follows.

Algorithm 2 Tournament-based mutation operator for "DE/rand/1"

Input: The target vector index i
Output: The offspring u_i
Generate three candidate individuals and set the best one as $w[0]$
$r_1 = w[0]$
do $\{$
Generate three candidate individuals and set the best one as $w[0]$
$r_2 = w[0]$
$\}$**while** $r_2 == r_1$
do
r_3=rndint$(1, NP)$
while $r_3 == r_2$ or $r_3 = r_1$
j_{rand}=rndint$(1, D)$
for $j = 1$ to D **do**
if $rndreal_j[0,1) < CR$ or j is equal to j_{rand} then
$u_{i,j} = x_{r_1,j} + F \cdot (x_{r_2,j} - x_{r_3,j})$
else
$u_{i,j} = x_{i,j}$
end if
end for

3.1 Our Approach

3.1.1 Generating Three Candidate Individuals At Random

In order to utilize the tournament selection, firstly we discuss how to generate three candidate individuals. The tournament best vector is selected among these candidate individuals. It is worth noticing that the process of generating three candidate individuals is repeated because both base vectors and difference vectors are regarded as the tournament best vectors. The main aim of utilizing the tournament selection is to obtain better base and difference vectors that are different from target vector x_i, which is ensured by the rule of $w[0] \neq w[1] \neq w[2] \neq i$.

To make the tournament vectors much easier to be selected among the three candidate individuals, we store the three candidate ones into an array that could be selected from the whole algorithm. Then the three candidate ones stored in the array are sored in ascent order (*i.e.*, from the best to the worst) based on the fitness of each candidate vector. Finally the first vector is more suitable to be the base vector or different vector.

3.1.2 Generating The Tournament Best Vectors

After achieving the process of forming three candidate individuals, the base vector and difference vectors could be formed by utilizing the process repeatedly. In addition, those vectors are different from one to another, which is ensured by the rule of $r_1 \neq r_2 \neq r_3 \neq i$

The pseudo-code of mutation operator introduced the tournament selection is presented in Algorithm 2.

3.2 DE With The Tournament Selection Mutation Operators

Apart from the mutation operator mentioned above, the tournament selection is suitable to apply in other mutation operators in DE, such as DE/current-to-best/2 and DE/rand-to-best/2. Comparing with Algorithm 1 and Algorithm 2, we could know that DE with the Tournament Selection still maintain the advantages of original DE including simple structure, realization process and so on. Meanwhile, instead of selecting vectors at random, the base vector and difference vectors are selected among three candidate individuals. It has a better chance of getting excellent vectors. it can explore the convergence. candidate individuals. It has a better chance of getting excellent vectors. So, it can explore the convergence.

4. Experimental Results and Analysis

In order to verify the viability of proposed the tournament selection with DE algorithm, we perform comprehensive experiments to test its efficiency, robustness and reliability. We adopt 13 benchmark functions[15] uniformly. These functions can be categorized into three groups:1) unimodal functions (F01 - F05); 2) basic multimodal functions (F06 - F12); 3) expanded multimodal functions (F13).The specific functional concepts are provided in Table 1. In addition, we have recorded the performance of all algorithms in terms of average value and standard deviation. In this paper, we utilize the Wilcoxon signed-rank test at α = 0.05 to analyze all the control experiments.

TABLE I
COMPARISON ON THE ERROR VALUES BETWEEN TOURNAMENT-BASED AND ITS CORRESPONDING DE WITH DIFFERENT MUTATION OPERATORS FOR FUNCTIONS F01 - F13 AT D = 30.

Prob	DE/rand/1/bin					DE/rand/2/bin				
	Tournament-DE		DE			Tournament-DE		DE		
	AVEDEV	STDEV		AVEDEV	STDEV	AVEDEV	STDEV		AVEDEV	STDEV
F01	0.0E+00	0.0E+00	+	5.0E-14	3.2E-14	4.6E-01	2.0E-01	+	1.2E+02	2.8E+01
F02	1.8E-14	4.5E-15	+	5.0E-07	2.2E-07	3.7E+00	1.7E+00	+	3.4E+01	7.6E+00
F03	7.5E-06	6.8E-06	+	2.0E-01	1.2E-01	2.6E+03	8.1E+02	+	1.8E+04	8.8E+03
F04	1.1E-02	1.2E-02	+	1.3E+00	2.6E+00	1.0E+01	1.3E+00	+	2.2E+01	1.8E+00
F05	6.6E+00	1.7E+00	+	1.9E+01	1.2E+00	2.3E+02	1.6E+02	+	2.4E+04	8.2E+03
F06	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00	1.8E+00	1.1E+00	+	1.2E+02	3.4E+01
F07	6.3E-03	1.7E-03	+	1.3E-02	3.5E-03	1.2E+02	9.8E+01	+	4.8E+05	2.2E+05
F08	2.1E+03	1.5E+03	+	4.2E+03	4.8E+02	4.4E+03	6.0E+02	=	4.4E+03	5.0E+02
F09	1.0E+02	3.5E+01	+	1.7E+02	8.5E+00	2.2E+02	1.1E+01	+	2.3E+02	1.0E+01
F10	6.6E-16	5.0E-16	+	9.3E-08	3.6E-08	7.7E-01	2.9E-01	+	4.5E+00	3.0E-01
F11	6.4E-04	2.2E-03	+	1.5E-04	1.1E-03	8.6E-01	9.1E-02	+	2.2E+00	3.0E-01
F12	2.5E-30	3.6E-30	+	7.6E-15	1.1E-14	1.9E+00	6.4E-01	+	1.8E+01	4.1E+00
F13	1.6E-26	1.1E-25	+	8.8E-13	7.2E-13	1.6E+01	3.8E+00	+	5.1E+03	6.7E+03
w/t/l	12/1/0			--		12/1/0			--	

Prob	DE/current-to-best/2/bin					DE/rand-to-best/2/bin				
	Tournament-DE		DE			Tournament-DE		DE		
	AVEDEV	STDEV		AVEDEV	STDEV	AVEDEV	STDEV		AVEDEV	STDEV
F01	5.1E-30	2.6E-29	+	1.5E-21	8.5E-22	0.0E+00	0.0E+00	+	2.0E-25	1.5E-25
F02	1.3E-12	6.5E-13	+	6.6E-09	5.1E-09	1.7E-14	8.3E-16	+	1.3E-11	6.8E-12
F03	2.8E-06	1.6E-06	+	1.7E-03	8.5E-04	2.8E-08	2.8E-08	+	5.1E-04	2.6E-04
F04	8.4E-06	5.7E-06	+	9.9E-04	4.1E-04	1.1E-08	5.7E-09	+	3.1E-05	1.7E-05
F05	3.8E-08	8.2E-08	+	5.7E-02	2.3E-01	4.7E-13	7.7E-13	+	2.2E-03	4.1E-03
F06	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00
F07	7.1E-03	1.7E-03	+	9.8E-03	2.8E-03	4.7E-03	1.4E-03	+	7.5E-03	2.1E-03
F08	4.4E+03	4.6E+02	+	4.5E+03	3.5E+02	4.3E+03	4.6E+02	+	4.3E+03	5.1E+02
F09	1.8E+02	1.0E+01	+	1.8E+02	1.1E+01	1.8E+02	1.0E+01	=	1.8E+02	1.1E+01
F10	8.0E-15	4.5E-15	+	2.0E-11	6.1E-12	5.9E-16	0.0E+00	+	1.8E-13	5.5E-14
F11	2.7E-03	5.1E-03	=	1.7E-03	3.9E-03	1.7E-03	4.9E-03	=	3.5E-04	1.7E-03
F12	1.2E-28	3.9E-28	+	1.2E-20	1.4E-20	1.6E-32	1.4E-47	+	1.0E-25	1.2E-25
F13	2.7E-25	8.1E-25	+	8.2E-17	2.2E-16	1.5E-32	0.0E+00	+	7.2E-23	8.0E-23
w/t/l	11/2/0			--		10/3/0			--	

* "+", "-", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$

4.1 Parameter Settings

To be fair, we compare the performance of the tournament-based DE with its corresponding original DE and jDE in terms of the uniform parameter standard. All the original algorithms including DE and jDE are kept the same as used in their original literature. The population size is set to 100. For the tournament-based DE and its corresponding DE, the whole process sets $F = 0.5$ and $CR = 0.9$ respectively. For the tournament-based jDE and its corresponding jDE, they regenerate with probabilities $\tau_1 = \tau_2 = 0.1$ at each generation.

The maximal number of fitness function evaluations(Max NFFEs) are set to D × 5, 000 . For the fair comparison for different algorism, the results are recorded from 50 dependent runs. In this paper, the only difference between those algorithms is the mutation operator. We use the same set of initial random to evaluate different algorithms and there is no other improvements except the application of the tournament selection.

4.2 Influence On DE With Different Mutation Operators

To validate the performance of proposed modified algorithm referred to the tournament selection, we applied the modified algorithm in DE. Four typical mutation operators are used in our experiments. In general, those mutation operators with two difference vectors are superior to other mutation operators with only one difference vectors in the respect of convergence performance. As the Table II shown, our results conform the law mentioned above.

All the results for 13 benchmark functions at D=30 are displayed in Table II, including the results of average value and standard deviation of 50 independence runs. To compare the significance between the modified algorithm and original algorithm, the Wilcoxon signed-rank test is employed. In Table II, according to the test, the results are summarized as "$w/t/l$", which denotes that our proposed algorithm wins in *w* functions,ties in *t* functions, and loses in *l* functions, compared with its corresponding DE method.

In fact, when dealing with all the 13 benchmark functions at D=30, our proposed modified algorithm obtain better errors values compared with the corresponding algorithm . In the strategy of DE/rand/1, except the function 06, our proposed modified algorithm wins the corresponding DE method. And in the 6th function both of the two algorithm convergence to zero. With "DE/rand/2" strategy, modified algorithm wins 12 functions, only ties in one function.With "DE/current-to-best/2" strategy, modified algorithm wins 11 functions, only ties in two functions. With "DE/rand-to-best/2" strategy, the tournament-based mutation operators wins 10 functions, while ties 3 functions.

According to the results shown in Table II and above analysis, we can verify that using the tournament selection to obtain vectors can significantly accelerate the corresponding original algorithm. Then we perform the comparison between modified jDE and original jDE in section 4.3.

TABLE II
COMPARISON ON THE ERROR VALUES BETWEEN TOURNAMENT-BASED AND ITS CORRESPONDING JDE FOR FUNCTIONS F01 - F13 AT D = 30.

Prob	jDE/rand1/bin				
	Tournament-jDE		jDE		
	AVEDEV	STDEV		AVEDEV	STDEV
F01	0.0E+00	0.0E+00	+	1.4E-28	3.3E-28
F02	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00
F03	2.9E-05	6.2E-05	+	2.6E-02	2.2E-02
F04	1.5E-03	4.0E-03	+	8.9E-04	3.4E-04
F05	1.6E+01	9.1E-01	+	2.1E+01	7.7E-01
F06	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00
F07	5.4E-03	1.8E-03	+	8.1E-03	1.8E-03
F08	-1.2E+04	5.5E+01	+	-1.2E+04	8.1E-03
F09	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00
F10	6.6E-16	5.0E-16	+	9.9E-15	2.3E-15
F11	3.9E-04	2.0E-03	-	0.0E+00	0.0E+00
F12	1.6E-32	0.0E+00	+	1.1E-29	1.2E-29
F13	1.4E-32	0.0E+00	+	1.6E-27	1.9E-27
w/t/l	9/3/1		--		

* "+", "-", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$

4.3 Influence On jDE

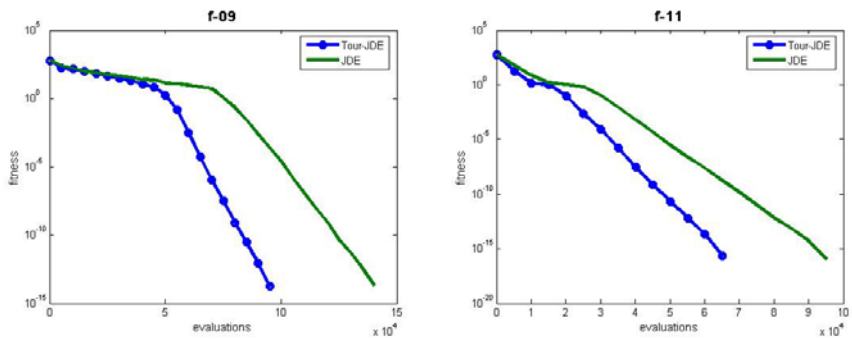

Fig. 1: the comparison of tournament-jDE and the original jDE with "jDE/rand1/bin" (D=30)

In the experiments of jDE, we employ the mutation strategy of "DE/rand/1. At the last part of Table II, we can find that modified algorithm is also superior to the original jDE. With the strategy at D=30, modified algorithm wins 9 functions, ties 3 functions, but lost in the 11th function. All of the average values of equal functions can converge to zero. The average value of 11th function should be zero, but the magnitudes of modified algorithm is E^{-04}. In fact, to the 11th function, there are only one or two results to be local convergence and the remainder of results would converge to zero in the process of 50 dependence runs. The reason might be that although modified algorithm significantly accelerate the convergence speed, it is lost in population diversity, which results in the premature convergence. As seen, for f-09 and f-11, tournament-jDE still outperforms jDE on the rate of convergence.

Table III

COMPARISON ON THE ERROR VALUES OF TOURNAMENT-BASED AND DERL FOR FUNCTIONS F01 - 13 AT D = 30.

Prob	*DE/rand/1bin*					*jDE/rand/1bin*				
	Tournament-DE		*DERL*			*Tournament-jDE*		*DERL*		
	AVEDEV	STDEV		AVEDEV	STDEV	AVEDEV	STDEV		AVEDEV	STDEV
F01	0.0E+00	0.0E+00	+	6.0E-25	4.6E-25	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00
F02	1.8E-14	4.5E-15	+	2.1E-12	1.3E-12	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00
F03	7.5E-06	6.8E-06	+	2.4E-04	2.5E-04	2.9E-05	6.2E-05	+	9.2E-05	1.1E-04
F04	1.1E-02	1.2E-02	+	3.6E-01	5.4E-01	1.5E-03	4.0E-03	+	2.6E-03	7.4E-03
F05	6.6E+00	1.7E+00	+	1.2E+01	1.7E+00	1.4E+01	9.1E-01	+	1.6E+01	8.5E-01
F06	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00
F07	6.3E-03	1.7E-03	+	7.1E-03	1.9E-03	5.4E-03	1.8E-03	-	4.9E-03	1.3E-03
F08	2.1E+03	1.5E+03	+	3.4E+03	1.1E+03	-1.2E+04	5.5E+01	+	-1.2E+04	9.0E+01
F09	1.0E+02	3.5E+01	+	1.4E+02	2.3E+01	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00
F10	6.6E-16	5.0E-16	+	3.4E-13	2.0E-13	6.6E-16	5.0E-16	=	5.9E-16	0.0E+00
F11	6.4E-04	2.2E-03	=	2.0E-04	1.4E-03	3.9E-04	2.0E-03	=	3.0E-04	1.5E-03
F12	2.5E-30	3.6E-30	+	1.2E-25	2.2E-25	1.6E-32	0.0E+00	=	1.6E-32	1.4E-47
F13	1.6E-26	1.1E-25	+	2.0E-22	1.3E-21	1.4E-32	0.0E+00	=	1.7E-32	2.7E-32
w/t/l	11/2/0		--			4/8/1		--		

* "+", "-", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$

4.4 Comparison with Other Modified Algorithm Applied the Tournament Selection

In the section of introduce, we mentioned a modified algorithm applied the tournament selection called DERL in [12]. To compare the performance of these two strategies, we perform a controlled experiment between the two strategies, which involves in the algorithm of DE and jDE. In Table III we provide the results of controlled experiment at D=30.

From this table we can see that in the case of DE with "DE/rand/1"strategy, for 11 functions our modified algorithm performed better than DERL, while for 2 cases both the algorithms performed similarity. In the case of jDE, for 4 functions our modified algorithm performed better than DERL, and for 8 functions both the algorithms performed equivalently, while for only 1 case DERL outperformed our modified algorithm.

4.5 Influence of the Population Size

In order to observe the performance of varying population size on proposed tournament-based mutation operators, we set two different population size $NP = 100$ and $NP = 400$ and record the results of comparison experiments about DE and jDE algorithm. In the nature, the ability of exploring would be declining with the increasing population size. The corresponding results are giving in Table IV and Table V. The gaps between Table I and Table V have been proved. However, for the larger population size(NP=400) the tournament-based mutation operators still perform reasonably better than the corresponding algorithm in terms of the error values.

TABLE IV

COMPARISON ON THE ERROR VALUES FOR FUNCTIONS F01 - F13 AT D = 100.

Prob	*jDE/rand1/bin*				
	Tournament-jDE		*jDE*		
	AVEDEV	STDEV		AVEDEV	STDEV
F01	2.1E-13	7.8E-14	+	1.1E-07	3.4E-08
F02	3.3E-08	7.1E-09	+	5.0E-05	9.1E-06
F03	7.4E+02	4.0E+02	+	1.1E+04	2.4E+03
F04	5.5E+00	1.4E+00	+	1.5E+01	6.4E-01
F05	9.3E+01	4.1E-01	+	9.7E+01	3.1E-01
F06	0.0E+00	0.0E+00	=	0.0E+00	0.0E+00
F07	3.6E-02	5.0E-03	+	9.7E-02	1.1E-02
F08	-2.4E+04	7.0E+02	+	-2.2E+04	9.7E+02
F09	6.9E+01	6.7E+00	+	9.3E+01	6.4E+00
F10	1.1E-07	2.5E-08	+	7.6E-05	1.0E-05
F11	1.1E-13	4.9E-14	+	6.2E-08	1.9E-08
F12	4.8E-14	2.1E-14	+	4.6E-08	2.1E-08
F13	1.6E-11	1.0E-11	+	7.9E-05	5.2E-05
w/t/l	12/1/0		--		

* "+", "-", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor

5.Conclusions and Future Work

The tournament selection is helpful to improve the optimization performance of an evolutionary algorithm. It is natural to incorporate the tournament selection and original mutation operators to accelerate the rate of convergence while maintaining the diversity of DE. Inspired by the speculation, we propose the tournament-

TABLE V
COMPARISON ON THE ERROR VALUES BETWEEN TOURNAMENT-BASED AND ITS CORRESPONDING DE WITH DIFFERENT MUTATION OPERATORS FOR FUNCTIONS F01 - F13 AT D = 100.

Prob	*DE/rand/1/bin*					*DE/rand/2/bin*				
	Tournament-DE		***DE***			***Tournament-DE***		***DE***		
	AVEDEV	**STDEV**	**AVEDEV**		**STDEV**	**AVEDEV**	**STDEV**	**AVEDEV**		**STDEV**
F01	6.2E-01	2.5E-01	+	5.3E+02	9.2E+01	1.1E+05	5.9E+03	+	1.2E+05	5.5E+03
F02	8.6E+00	3.0E+00	+	1.1E+03	6.1E+03	2.0E+24	4.7E+24	+	4.5E+25	2.3E+26
F03	2.1E+07	7.1E+06	+	1.3E+08	1.9E+07	3.9E+08	2.1E+07	+	4.4E+08	3.0E+07
F04	3.8E+01	4.2E+00	+	7.6E+01	4.2E+00	1.2E+02	3.1E+00	+	1.2E+02	2.7E+00
F05	1.8E+02	3.1E+01	+	3.9E+04	1.7E+04	2.5E+08	3.1E+07	+	3.0E+08	2.4E+07
F06	0.0E+00	0.0E+00	+	5.0E+02	9.9E+01	1.0E+05	5.6E+03	+	1.2E+05	4.9E+03
F07	3.1E+02	2.0E+02	+	1.3E+07	3.7E+06	4.9E+10	6.2E+09	+	6.4E+10	7.6E+09
F08	2.7E+04	7.1E+02	+	2.7E+04	7.7E+02	2.7E+04	8.3E+02	=	2.7E+04	8.5E+02
F09	9.5E+02	2.4E+01	+	1.0E+03	3.4E+01	1.5E+03	3.7E+01	+	1.6E+03	3.8E+01
F10	5.0E-01	1.1E-01	+	7.4E+00	5.0E-01	2.1E+01	8.6E-02	+	2.1E+01	7.2E-02
F11	7.7E-01	1.2E-01	+	2.5E+01	5.1E+00	1.8E+03	1.3E+02	+	2.1E+03	8.1E+01
F12	6.3E-01	3.8E-01	+	4.5E+03	7.1E+03	1.9E+09	2.5E+08	+	2.4E+09	3.2E+08
F13	1.2E+01	6.0E+00	+	1.3E+03	1.2E+03	9.9E+08	1.2E+08	+	1.2E+09	1.2E+08
w/t/l	13/0/0		--			12/1/0		--		
Prob	*DE/current-to-best/2/bin*					*DE/rand-to-best/2/bin*				
	Tournament-DE		***DE***			***Tournament-DE***		***DE***		
	AVEDEV	**STDEV**	**AVEDEV**		**STDEV**	**AVEDEV**	**STDEV**	**AVEDEV**		**STDEV**
F01	5.6E+01	8.7E+00	+	7.1E+02	8.2E+01	2.5E+00	4.0E-01	+	2.1E+02	3.1E+01
F02	1.2E+12	8.1E+12	+	1.6E+15	9.0E+15	2.3E+02	2.5E+02	+	8.7E+06	2.7E+07
F03	9.5E+07	1.0E+07	+	1.3E+08	1.3E+07	5.2E+07	7.4E+06	+	9.0E+07	1.2E+07
F04	6.9E+01	5.0E+00	+	8.5E+01	4.6E+00	5.0E+01	4.2E+00	+	7.1E+01	4.5E+00
F05	1.0E+04	2.2E+03	+	2.5E+05	4.5E+04	6.7E+02	1.0E+02	+	4.2E+04	9.5E+03
F06	7.9E+01	9.4E+00	+	7.3E+02	8.9E+01	1.1E+01	2.5E+00	+	2.4E+02	2.9E+01
F07	6.5E+05	1.7E+05	+	2.7E+07	5.4E+06	4.5E+03	2.1E+03	+	3.7E+06	1.0E+06
F08	2.7E+04	8.6E+02	+	2.7E+04	7.6E+02	2.7E+04	9.6E+02	+	2.7E+04	6.3E+02
F09	1.1E+03	3.6E+01	+	1.2E+03	2.7E+01	1.1E+03	3.2E+01	+	1.1E+03	3.0E+01
F10	3.6E+00	1.5E-01	+	6.9E+00	3.8E-01	1.3E+00	1.7E-01	+	4.7E+00	2.0E-01
F11	2.3E+00	1.8E-01	+	1.6E+01	1.7E+00	1.0E+00	2.5E-02	+	5.5E+00	6.0E-01
F12	8.1E+02	1.1E+03	+	1.9E+05	1.3E+05	1.2E+01	2.9E+00	+	5.2E+03	6.0E+03
F13	3.8E+03	4.2E+03	+	4.4E+05	1.4E+05	1.9E+02	4.4E+01	+	3.7E+06	1.3E+04
w/t/l	13/0/0		--			13/0/0		--		

* "+", "-", and "=" indicate our approach is respectively better than, worse than, or similar to its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$

based mutation operators for DE algorithms. The process is that the base and difference vectors are replaced by the tournament best vector, which is selected from three random individuals,and the remaining vectors are generated by selecting at random. In addition, None of our modified algorithms add any other parameters or heavy work, which ensure that the tournament-based mutation operators are efficient and easy to achieve.

The comprehensive experiments have been performed to verify the performance of the tournament-based mutation operators. We have discussed it with four typical mutations in DE, one mutation strategy of "rand/1" in jDE ,comparison experiments with DERL and diverse population size. From the comparison in the previous sections it is quite clear that the tournament-based mutation operators are significantly superior to the corresponding original algorithm, including DE, jDE and DERL.

A disturbing fact of DE is its totally nature evolution with less heuristic search. Further research is under way in applying intelligent computing to DE and in developing in an efficient DE model for large dimensional problems.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China under Grant No. 61203307, the Fundamental Research Funds for the Central Universities at China University of Geosciences (Wuhan) under Grant No. CUG130413, and the Research Fund for the Doctoral Program of Higher Education under Grant No. 20110145120009.

References

[1] R. Storn and K. Price, "Differential evolution–A simple and efficient adaptive scheme for global optimization over continuous spaces," Berkeley, CA, Tech. Rep., tech. Rep. TR-95- 012, 1995.

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

[3] K. Price, R. Storn, and J. Lampinen, "Differential Evolution: A Practical Approach to Global Optimization,"Springer-Verlag, Berlin,2005.

[4] S. Das and A. Konar, "Design of two dimensional IIR filters with modern search heuristics: A comparative study," *Int. J. Comput. Intell. Applicat.* Vol. 6, no. 3,2006,pp. 329–355.

[5] M. Omran, A. P. Engelbrecht, and A. Salma, "Differential evolution methods for unsupervised image classification,"in *Proc. 7th Congr. Evol. Comput.(CEC-2005)*, 2. Piscataway, NJ: IEEE Press, 966–973.

[6] S. Das, A.Abraham and A.Konar, "Adaptive clustering using improved differential evolution algorithm," *IEEE Trans Syst., Man, Cybern.A*,38,1, 2008,218-237.

[7] B. Dorronsoro, and P. Bouvry, "Improving classical and decentralized differential evolution with new mutation operator and population topologies," *IEEE* ,vol.15,2011,no. 1.

[8] H.-Y. Fan and J.Lampinen, "A trigonometric mutation operation to differential evolution," *J.of Global Optimization*, vol.27,no. 1, 2003,pp.105-129.

[9] Z. Yang, K. Tang, and X. Yao, "Self-adaptive differential evolution with neighborhood search," in *Proc. IEEE Congress on Evol. Comput.*, 2008,pp. 1110–1116.

[10] Z. Cai, W. Gong, C. X. Ling, and H. Zhang, "A clustering-based differential evolution for global optimization," *Applied Soft Computing*, vol. 11, no. 1, 2011,pp. 1363 – 1379.

[11] D. Thierens and David E.Goldberg, "Convergence models of genetic algorithm selection schemes," *PPSN,*Springer Berlin Heidelberg, 1994,pp.119-129.

[12] P. Kaelo and M. Ali, "A numerical study of some modified differential evolution algorithms," *European Journal of Operational Research*, vol. 169, no. 3, 2006,pp. 1176–1184.

[13] J. Brest, S. Greiner, B. Boˇskovi´c, M. Mernik, and V. ˇZumer, "elfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems,"*IEEE Trans. on Evol. Comput.*, vol. 10, no. 6, 2006,pp. 646–657.

[14] R. Storn and K. Price, "Home page of differential evolution," 2010. [Online]. Available: http://www.ICSI.Berkeley.edu/˜storn/code.html

[15] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen,A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC2005 special session on real-parameter optimization,"2005.[Online].Available: http://www.ntu.edu.sg/home/EPNSugan

Chen Qiu is still a student in China University of Geosciences. She started the study of DE in 2011, and current research interest is differential evolution.

Miaomiao Liu is still a student in China University of Geosciences. She started the study of DE in 2011, and current research interest is differential evolution.

Dr. Wenyin Gong
Address: No. 388, Lumo Road,
School of Computer Science,
China University of Geosciences,
Wuhan 430074, P.R. China
Tel: 86-27-67883716 Fax: 86-27-67883716
Homepage: http://cs.cug.cn/teacherweb/gwy/