Local Information of Fitness Landscape Obtained by Paired Comparison-Based Memetic Search for Interactive Differential Evolution

Pei, Yan
Computer Science Division, Universally of Aizu

Takagi, Hideyuki
Faculty of Design, Kyushu University

出版情報：IEEE Congress on Evolutionary Computation. 2015, pp.2215-2221, 2015-05-25. IEEE
バージョン：

Local Information of Fitness Landscape Obtained by Paired Comparison-Based Memetic Search for Interactive Differential Evolution

Yan PEI
Tsuruga, Ikki-machi, Aizuwakamatsu, 965-8580 Japan
Computer Science Division, the University of Aizu, Japan
Email: peiyan@u-aizu.ac.jp
http://web-ext.u-aizu.ac.jp/~peiyan/

Hideyuki TAKAGI
4-9-1, Shiobaru, Minami-ku, Fukuoka, 815-8540 Japan
Faculty of Design, Kyushu University, Japan
Email: h.takagi.457@m.kyushu-u.ac.jp
http://www.design.kyushu-u.ac.jp/~takagi/

***Abstract*—We propose a triple comparison-based interactive differential evolution (IDE) algorithm. The comparison of target vector and trail vector supports a local fitness landscape for IDE algorithm to conduct a memetic search. Besides target vector and trail vector in canonical IDE algorithm framework, we conduct a memetic search around whichever is the vector with better fitness. We use a random number from a normal distribution generator or a uniform distribution generator to perturb the vector for generating a third vector. By comparing the target vector, the trail vector and the third vector, we implement a triple comparison mechanism in IDE algorithm. A Gaussian mixture model is applied as a pseudo IDE user in our evaluation. We compare our proposal with canonical IDE and triple comparison-based IDE implemented by opposite-based learning, and apply several statistical tests to investigate the significance of our proposed algorithm. From the evaluation results, our proposed triple comparison-based IDE algorithm shows significantly better performance optimization. We also investigate potential issues arising from our proposal, and discuss some open topics and future opportunities.**

I. Introduction

Evolutionary computation (EC) is a meta-heuristic technique that is used to optimize complex problems, which are hard to solve using conventional optimization methods [12]. Interactive EC (IEC) is a niche research field in the EC community that embeds the feeling, knowledge and experience of a real human into EC optimization, so as to make IEC algorithm converge to a real human's preference rather than to the fitness function(s) of an optimized problem. Extending IEC application scale and enhancing IEC algorithm performance (including improving IEC interface) are two primary research subjects within the IEC field. One of these attempts is to apply IEC optimization principles and techniques to a variety of industrial and commercial applications that require the assistance of a real human in the optimization process. The other pursues the discovery of more effective and efficient IEC algorithms or interfaces to obtain a better optimization result, while at the same time relieving human fatigue due to psychological and physiological limitations of real humans when they interact with an IEC algorithm.

There are three research perspectives in IEC for obtaining a more effective and efficient IEC algorithm and interface [5]. The first is to approximate the fitness landscape of a real human's subjective evaluation space, attempting to build optimized problem structures to assist the IEC search. Several methods were proposed which deal with this aspect, such as dimensionality reduction technique [7], Fourier analysis [6], and support vector regression [8]. The second perspective is to develop a new search mechanism within a conventional IEC algorithm to enhance its performance or to design a better interface, which improves human computer communication. The triple and quadruple comparison-based interactive differential evolution (IDE) follows this research direction [9] and a kernel method-based human model is studied to assist the IEC user in reducing his/her fatigue [10]. The third research approach is to create a new EC/IEC algorithm in order to achieve better performance of EC/IEC optimization [3], [4].

Differential evolution (DE) has the characteristics of a paired comparison scheme where there is competition between target vector and trail vector. Its benefit to IEC application is that it allows a real human to give fitness based on paired comparison of two objects rather than to give multiple fitness values of objects at the same time. Because each human has his/her limitation when supporting fitness to IEC algorithms from one generation to the next, the paired comparison can significantly relieve IEC user fatigue [13]. Reference [9] uses the opposite point(s) of a target vector and/or a trail vector from opposition-based learning for implementing triple and quadruple comparison schemes in conventional IDE. This can significantly enhance optimization performance of conventional IDE .

By drawing inspiration from multiple comparison implementation in IDE algorithm, fitness landscapes can support information on search conditions and problem structures to develop a new multiple comparison-based IDE algorithm. There are a variety of representations of fitness landscape. One specific representation from IDE algorithm is that of the fitness of target vector and trail vector. When the fitness of target vector is better than that of the trail vector, it indicates that searching around the target vector has the potential to obtain the global optimum with higher probability, and vice versa. Based on the hypothesis, in this paper, we propose a new triple comparison-based IDE, which conducts a memetic search around whichever is the better vector, be it the target vector or the trail vector. We implement the memetic search by perturbation with a normal distribution generator or a uniform distribution generator. We use a Gaussian mixture model (GMM) as a pseudo IDE user to evaluate our proposal. From the evaluation of our proposal, this new triple comparison-based IDE algorithm can obtain a better optimization performance than a canonical IDE algorithm.

Following this introductory section, in section II, we briefly review conventional paired comparison-based IDE, and triple and quadruple comparison-based IDE. Section III presents our new triple comparison-based IDE by memetic search with the local fitness landscape obtained from a paired comparison of target vector and trail vector. The memetic search is implemented by adding a perturbation from a normal distribution generator or a uniform distribution generator. In section IV, we evaluate our proposal by using GMM. We analyze and discuss some open topics and issues arising from the evaluation results in section V. Finally, in section VI, we conclude the current work, and present some future opportunities, which invite investigation.

II. Interactive Differential Evolution and Multiple Comparison Mechanism for Human Evaluation

A. Paired Comparison-Based Interactive Evolutionary Computation

When the individuals of an IEC optimization are voice, image or video, i.e., time series objects, IEC users have to compare an individual with others in their memory. As a result, IEC users' mental stress and fatigue become heavy. It was pointed out that human beings have a memory limitation and cannot process more than five to nine different pieces of information simultaneously [2]. Population sizes of many IEC systems frequently exceed this memory limitation. Consequently, displaying 10 – 20 voices, images or videos to an IEC user is not practical.

Paired comparison-based IEC solves this problem by replacing comparison of all individuals with paired comparisons. This is expected to reduce IEC user fatigue. One of the concrete implementations of paired comparison is a tournament interactive genetic algorithm (IGA) [1]. The obtained fitness of tournament IGA has noise because the tournament is not a round robin competition against the canonical IGA algorithm. The noise influences an IGA selection operation and results in reducing IGA search performance. One promising subject for future research is the way in which an efficient paired comparison-based IEC algorithm should be implemented.

B. Paired Comparison-Based Interactive Differential Evolution

Differential evolution (DE) is a population-based optimization algorithm [11]. DE uses a differential vector from two random individuals to perturb a base vector (the vector with best fitness value or a third random vector from a population) to implement a mutation operation and obtain a mutant vector. It conducts a crossover operation between the mutant vector and the target vector to create a trail vector. Following this, it compares the fitness of target vector and trail vector to enable the better one to survive into the next generation. The formal expression of this search mechanism is shown in Eq. (1), where $mutant_{i,j}$ is a mutant vector, $base_{i,j}$ is a base vector, $x1_{i,j}$ and $x2_{i,j}$ are two random vectors (i and j are the indexes of individual and dimension, respectively.). F is a scale factor that needs to be set whose range is usually within $(0, 2]$ from the discussion of [11]. Note that the target vector, base vector and two random vectors are four different vectors, so the minimum size of population is four in DE.

$$mutant_{i,j} = base_{i,j} + F * (x1_{i,j} - x2_{i,j}) \tag{1}$$

DE algorithm includes paired comparison naturally in the algorithm 1. Paired comparison-based IDE does not revise any parts of its canonical DE algorithm [13]. Since it displays paired comparisons of individuals to an IDE user with canonical DE algorithm, IDE algorithm is expected to be a promising paired comparison IEC method.

Algorithm 1 Paired comparison-based interactive differential evolution algorithm, taking a minimum optimization problem as an example. PS: population size; Dim: dimension; G: generation; $maxIter$: maximum generation; i: index of individual; j: index of dimension. f(*) is a fitness function

Generate an initial population.
Evaluate the fitness for each individual.
for $G = 1$ to $maxIter$ **do**
  for $i = 1$ to PS **do**
    k=rand(1,Dim)
    for $j = 1$ to Dim **do**
      if $rand[0, 1) < C_r$ **or** $j == k$ **then**
        $mutant_{i,j} = base_{i,j} + F * (x1_{i,j} - x2_{i,j})$
        $trail_{i,j} = mutant_{i,j}$
      else
        $trail_{i,j} = target_{i,j}$
      end if
    end for
    /*Paired Comparison Mechanism*/
    for $i = 1$ to PS **do**
      if $f(trail_i) < f(target_i)$ **then**
        replace $target_i$ with $trail_i$
      end if
    end for
  end for
end for
return the optimum

C. Opposition-based Learning

Opposition-based learning (OBL) [14] is used for machine learning [15] and acceleration of optimization search (OBL optimization). The philosophy of OBL indicates if original hypothesis is not adequate, with respect to its opposite hypothesis. Suppose that $x \in [a, b]$ is a real number, the opposition point of x is given as $OP(x) = a + b - x$. By extending this principle into a multi-dimensional space, opposition point, $OP(X)$, of one point on a n-dimensional real space, X =$(x_1, x_2, ..., x_n)$ ($x_i \in [a_i, b_i], i = 1, 2, ..., n$; $a_i, b_i \in R$), is given by Eq.s (2) and (3).

$$OP(x_i) = a_i + b_i - x_i \tag{2}$$

$$OP(X) = \{OP(x_1), OP(x_2), ..., OP(x_n)\} \tag{3}$$

D. Triple and Quadruple Comparison-Based Interactive Differential Evolution

The triple and quadruple comparison-based IDE uses not only a target vector and a trial vector, but also their opposition vector(s) at every comparison in DE search [9]. There are three implementations. Two triple comparison-based IDEs are implemented by comparing a target vector, a trail vector and either opposite point of the target vector or opposite point of the trail vector. A quadruple comparison-based IDE is implemented by comparing a target vector, a trail vector and opposite points of the target vector and the trail vector.

Two different mirror points for calculating opposition points can be used. One is the center gravity point of an individual distribution, the other is the whole searching space because a big shift of individuals may accelerate DE convergence especially in the early generations. From the empirical study of these two methods, there is no significant difference between them [9]. Accordingly, we use the latter one, whole searching ranges, in our experimental evaluation.

III. Memetic Search in Interactive Differential Evolution for Implementing a New Triple Comparison Mechanism

A. Local Fitness Landscape from Paired Comparison

Fitness landscape is originally a biological concept that is used to visualize the relationship between a biological entity and its evolutionary process. In the evolutionary optimization field, it presents the solution of optimized problems and the extent to which these problems are capable of being solved. Most such problems can be represented by fitness function(s). In IEC, fitness landscape can act as a tool to analyze human models of physiology or psychology, which presents a human's preference according to the optimized objective of an IEC application.

In the IDE, the fitness of target vector and trail vector supports a local fitness landscape when comparing their paired comparison fitness value. The individual (either target vector or trail vector) with better relative fitness indicates the potential search region where there may be a global optimum. This information provides the search condition and landscape to improve IDE/DE optimization performance. Figure 1 demonstrates the local fitness landscape from a paired comparison of target vector and trail vector in the IDE. If the fitness of trail vector is better than that of the target vector, it means that by searching around trail vector, there is a great probability of finding the global optimum, and vice versa. The promising direction is from the vector with lower fitness to the vector with higher fitness, and promising search range is around the vector with higher fitness.

B. Memetic Search in Interactive Differential Evolution for Implementing a New Triple Comparison Mechanism

With the local fitness landscape obtained from paired comparison in IDE, we propose a memetic search method in IDE algorithm to implement a new triple comparison-based IDE. When we find a vector with higher fitness from comparison of target and trail vectors, IDE algorithm can conduct a memetic search by perturbing the vector to generate a third vector in

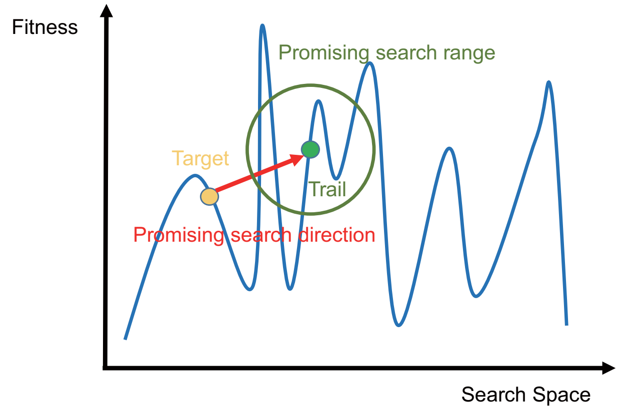


Fig. 1. Local fitness landscape from paired comparison of target vector and trail vector in IDE, taking a maximum optimization problem as an example. The fitness of the trail vector is higher than that of target vector in this figure, meaning that the promising search range is around the trail vector, promising search direction is from target vector to trail vector, and versa vice. We apply memetic search with perturbation from a certain distribution generator (we use a uniform distribution generator and a normal distribution generator in this paper) to implement a triple comparison-based IDE algorithm.

order to implement a triple comparison mechanism. Here the originality of our proposal can be seen. The promising search direction is from the one with lower fitness to the other, and perturbation can be implemented by adding a random number from a generator. In this paper, we attempt to use a normal distribution (Eq. (4), $\mu = 0, \sigma = 1$) or a uniform distribution (Eq. (5), $a = 0, b = 1$) as generator in our experimental evaluation.

$$N(x, \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} exp(\frac{-|x-u|^2}{2\sigma^2}) \tag{4}$$

$$U(x, a, b) = \frac{1}{b-a} \tag{5}$$

Besides a target vector and a trail vector in the canonical DE framework, the third vector is from perturbation on whichever is the better one from target and trail vectors. Eq.s (6) and (7) show the two implementations of the third vector ($third_{i,j}$) from a normal distribution generator and a uniform distribution generator, respectively. Abbreviations, $better_{i,j}$ and $worse_{i,j}$, are whichever the vector with better and worse fitness from target vector and trail vector in Eq.s (6) and (7). It is a way to implement a memetic search in the IDE / DE algorithm. After we obtain a third vector from Eq. (6) or Eq. (7), the IDE algorithm compares target vector, trail vector and the third vector to implement a triple comparison mechanism in IDE.

$$third_{i,j} = better_{i,j} + (better_{i,j} - worse_{i,j}) * N(0,1) \tag{6}$$

$$third_{i,j} = better_{i,j} + (better_{i,j} - worse_{i,j}) * U(0,1) \tag{7}$$

The evaluation metric for IDE is user fatigue extent rather than the number of fitness calculation, which is related with user fatigue but is not in proportion to it. Suppose to compare the user fatigues of choosing the best IDE object between two objects and that among three or four objects. The mental load from less comparison must be less than that from more comparisons, but it does not mean mental load from triple comparison is not 1.5 times of that from paired comparison. Even when IDE tasks are time series optimization problems, such as music, or movies that we cannot compare spatially and simultaneously, an IDE user's mental load must increase, but its ratio may not become 1.5 times as well. Generally speaking, when the number of individual comparisons is within the number that an IDE user can memorize, IEC user fatigue is lower; when it exceeds the maximum memory capacity, user fatigue drastically increases. Paying attention to this fact, we develop our proposed methods requiring triple comparisons, and aim to reduce the total user fatigue by accelerating IDE search even if the user fatigue of each comparison increases. This is the philosophy motivating our use of multiple comparison mechanism in our proposed IEC algorithm.

IV. Optimization Evaluation

A. Benchmark Functions and Experimental Conditions

User fatigue is an important evaluation factor for IEC. When mental loads for evaluating individuals are the same, we may say that the IEC user fatigue is in proportion to the total time until the IEC user finds the satisfactory individual. However, when mental loads for evaluating one individual are different due to different IEC interfaces, this relation is not always true. There are cases where IEC user fatigue becomes low thanks to easy evaluation even if total evaluation time until to the goal is long; there are also opposite cases where IEC user fatigue becomes low thanks to short total evaluation time, even if the mental load for one evaluation is high.

We need to evaluate acceleration methods by analyzing load of one evaluation and convergence characteristics through IEC simulation. Following this we need to conduct a human subjective evaluation to confirm the simulation results. This paper deals with the IEC simulation of the former stage. We use Gaussian mixture models (GMM) as pseudo IDE users for evaluation in this section. More specifically, we combine four Gaussian functions ($k = 4$) and implement the characteristics expressed by Eq. (8) in 3 dimensions (3-D), 5-D, 7-D, and 10-D.

$$GMM(\mathbf{x}) = \sum_{i=0}^{k} a_i \exp\left(-\sum_{j=0}^{n} \frac{(x_{ij} - \mu_{ij})^2}{2\sigma_{ij}^2}\right) \qquad (8)$$

where

$$\sigma = \begin{pmatrix} 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{pmatrix}$$

$$\mu = \begin{pmatrix} -1 & 1.5 & -2 & 2.5 & -1 & 1.5 & -2 & 2.5 & -1 & 1.5 \\ 0 & -2 & 3 & 1 & 0 & -2 & 3 & 1 & 0 & -2 \\ -2.5 & -2 & 1.5 & 3.5 & -2.5 & -2 & 1.5 & 3.5 & -2.5 & -2 \\ -2 & 1 & -1 & 3 & -2 & 1 & -1 & 3 & -2 & 1 \end{pmatrix}$$

$$a_i = \begin{pmatrix} 3.1, & 3.4, & 4.1, & 3.0 \end{pmatrix}^T$$

B. Algorithm Parameter Setting and Evaluation Metrics

We test each benchmark function for 20 generations with 30 trial runs. The parameter setting of canonical IDE, triple IDE and our proposed new triple IDE by memetic search are listed in Table I. Figure 2 shows the average convergence curves of best fitness values from 30 trial runs for all 4 benchmark functions (simulated users). Table III shows their means. Abbreviations used in Figures 2 and Table III are given in Table II.

TABLE I. IDE EXPERIMENT PARAMETERS SETTING.

Item	Value
population size	20
max. search generation	20
dimensions of benchmark functions, D	3, 5, 7, 10
# of trial runs	30
scale factor F	1
crossover rate	1

We apply the Wilcoxon signed rank test on our proposed algorithms and their competitive algorithms to evaluate significant differences of two competitors. In addition, we apply the Friedman test and Bonferroni-Dunn test on one of our proposed algorithms and their competitive algorithms to rank these algorithms and evaluate significant difference between these competitors. Note that we take our proposed algorithm as a control algorithm in the Bonferroni-Dunn test.

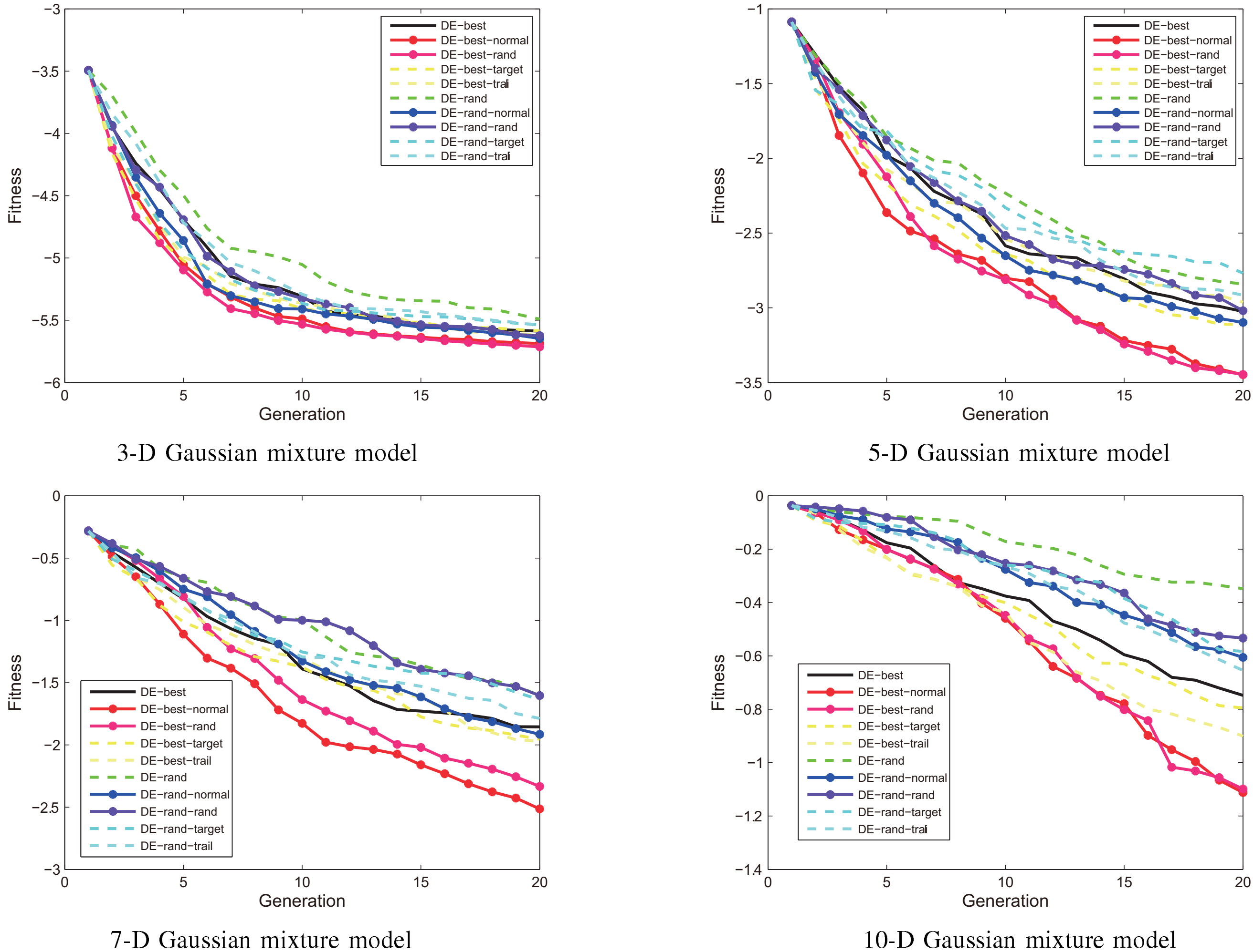
TABLE II. ABBREVIATIONS USED IN THE EXPERIMENTAL EVALUATIONS.

Abbreviations	Meaning
DE-best	standard DE/best/1/bin [11].
DE-best-target	triple comparison-based DE with opposite point of target vector [9].
DE-best-trail	triple comparison-based DE with opposite point of trail vector [9].
DE-best-normal	triple comparison-based DE by memetic search with normal distribution generator.
DE-best-rand	triple comparison-based DE by memetic search with uniform distribution generator.
DE-rand	standard DE/rand/1/bin [11].
DE-rand-target	triple comparison-based DE with opposite point of target vector [9].
DE-rand-trail	triple comparison-based DE with opposite point of trail vector [9].
DE-rand-normal	triple comparison-based DE by memetic search with normal distribution generator.
DE-rand-rand	triple comparison-based DE by memetic search with uniform distribution generator.

V. Discussion

A. Discussion on Optimization Performance of Our Proposal

When we find the promising search region from a comparison of target and trail vector, we conduct a memetic search in the region to implement a triple comparison-based IDE so as to relieve IDE user fatigue. As an IDE user compares three objects against two objects in original IDE, it does not mean user fatigue will become 1.5 times that of the original one. This is one factor motivating our proposed memetic search in the triple comparison-based IDE.



3-D Gaussian mixture model

5-D Gaussian mixture model

7-D Gaussian mixture model

10-D Gaussian mixture model

Fig. 2. Average convergence curves for 30 trial runs with 3-D, 5-D, 7-D and 10-D Gaussian mixture models.

Figure 2 demonstrates that convergence speed of the proposed algorithms is faster than the corresponding IDE algorithm and triple comparison-based IDE algorithms in [9]. We observe that all of our proposed algorithms significantly outperform canonical IDE and triple comparison-based IDE algorithms of [9] for the IDE with best vector as base vector from Table III according to Wilcoxon signed rank test. However, for the DE algorithms with random vector as base vector, our proposed algorithm acceleration performance is not obvious, and proposed algorithm applied in a lower dimensional problem is better than that applied to a higher dimensional problem. This may be because the memetic search in our proposal can enhance the exploitation capability of IDE and its search range influences our proposal's performances. In our experimental evaluation, we only investigate the performance of our proposal with normal distribution generator ($N(\mu,\sigma), \mu = 0, \sigma = 1$) and uniform distribution generator ($U(a,b), a = 0, b = 1$). We will further investigate the issue of parameter setting in our future work.

We conduct Wilcoxon signed rank tests on our proposed algorithms between the one with normal distribution perturbation and the one with uniform distribution perturbation. With the exception of rand-normal and rand-rand algorithms applied on 3D and 7D problem, there is not any significant difference between these two algorithms. It can be concluded that a memetic search with different distribution would obtain the same evaluation result.

Our proposed algorithm needs fitness evaluation times more than that of canonical DE and as the same as that of DE with opposition-based learning. For DE/best algo-

TABLE III. MEAN VALUE OF ALL THE COMPETITIVE ALGORITHMS AT 20TH GENERATION. †, ‡, § PRESENT OUR PROPOSED ALGORITHMS ARE SIGNIFICANTLY BETTER THAN CANONICAL DE, TRIPLE COMPARISON-BASED DE WITH OPPOSITE POINT OF TARGET VECTOR AND TRIPLE COMPARISON-BASED DE WITH OPPOSITE POINT OF TRAIL VECTOR, RESPECTIVELY.

Algorithm	3D	5D	7D	10
DE-best	-5.58818	-3.02461	-1.85508	-0.74791
DE-best-target	-5.58665	-3.11135	-1.95467	-0.79504
DE-best-trail	-5.62214	-2.96281	-1.97154	-0.90041
DE-best-normal	-5.68788†‡§	-3.44824†‡§	-2.51374†‡§	-1.11205†‡§
DE-best-rand	-5.71373†‡§	-3.44642†‡§	-2.33407†‡§	-1.09868†‡§
DE-rand	-5.49153	-2.84135	-1.60453	-0.34802
DE-rand-target	-5.5353	-2.76868	-1.637	-0.583
DE-rand-trail	-5.54368	-2.91589	-1.78787	-0.65426
DE-rand-normal	-5.64681†‡§	-3.09815†‡	-1.91501†‡	-0.6054†
DE-rand-rand	-5.62434†‡§	-3.01962‡	-1.60471	-0.53337†

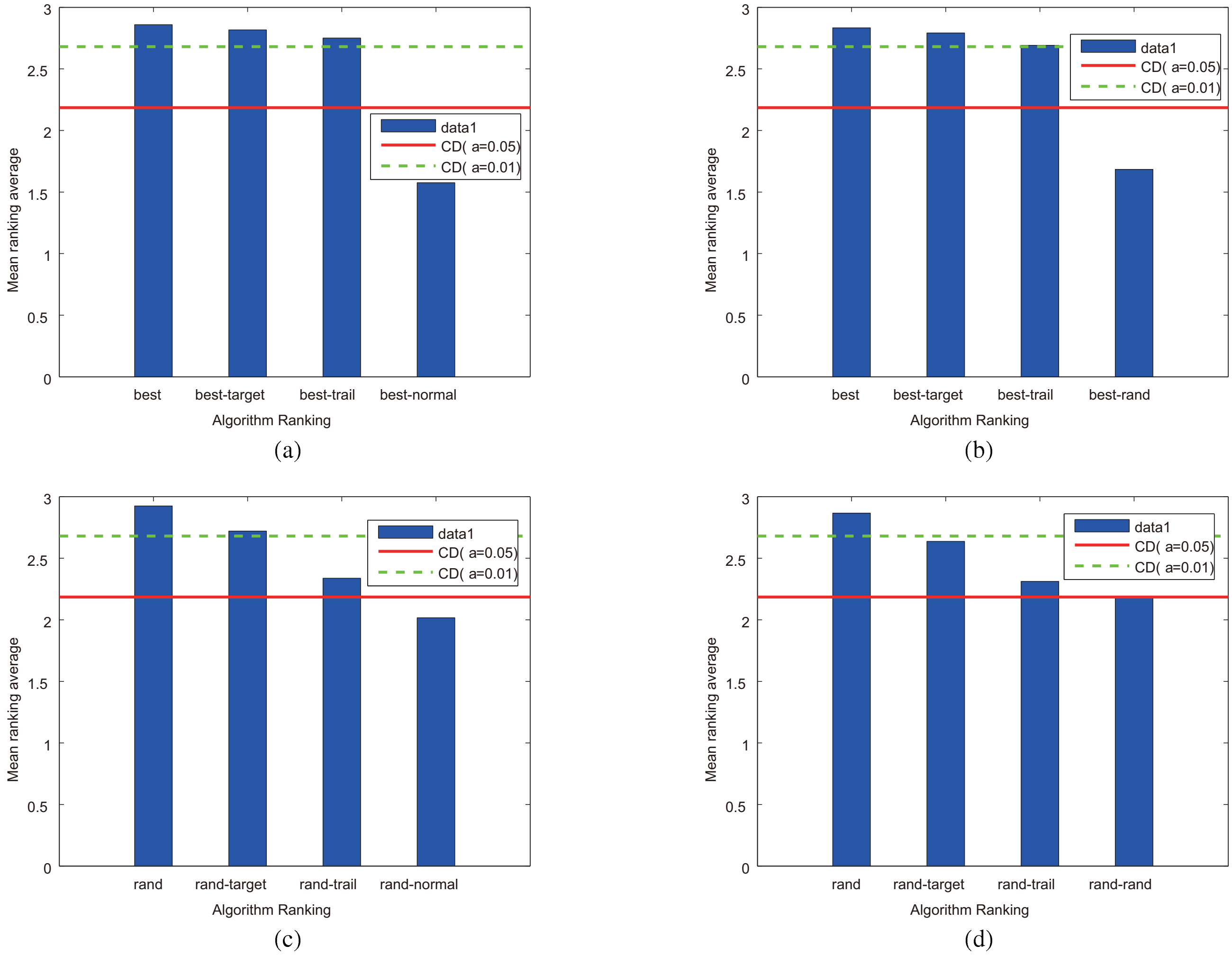


Fig. 3. Bonferroni-Dunn test with taking our proposed algorithm as a control algorithm in each competitive group. From this evaluation, we can conclude that our proposed algorithm are significantly better than their competitors in each group.

rithm, in every generation, it needs $(2 \times populationsize + populationsize) \times generation$ times evaluations, $2 \times populationsize$ times evaluations are for paired comparison and $populationsize$ times evaluations are for choosing the best vector as base vector. For DE/rand algorithm, in every generation, it needs $(2 \times populationsize) \times generation$ times evaluations, $2 \times populationsize$ times evaluations are for paired comparison. In our evaluation experiments, DE/best needs 1200 times fitness evaluation at the 20th generation, and DE/best + our proposal achieves to this times fitness evaluation at the 15th generation. For the same reason, DE/rand needs 600 times fitness evaluation at the 15th generation, and DE/rand + our proposal achieves to this times fitness evaluation at the 10th generation. Table IV presents the Wilcoxon signed rank test in this condition, the result indicates that our proposal DE/best + our proposal is significantly better than some of its competitors, and DE/rand + our proposal seems as the same as the canonical DE and proposals of [9].

TABLE IV. MEAN VALUE OF ALL THE COMPETITIVE ALGORITHMS AT THE SAME FITNESS EVALUATION TIMES (DE-BEST ALGORITHM GROUP IS UP TO 1200 EVALUATION TIMES, AND DE-RAND ALGORITHM GROUP IS UP TO 600 EVALUATION TIMES.). MARKS †, ‡, § PRESENT OUR PROPOSED ALGORITHMS ARE SIGNIFICANTLY BETTER THAN CANONICAL DE, TRIPLE COMPARISON-BASED DE WITH OPPOSITE POINT OF TARGET VECTOR AND TRIPLE COMPARISON-BASED DE WITH OPPOSITE POINT OF TRAIL VECTOR, RESPECTIVELY.

Algorithm	3D	5D	7D	10
DE-best	-5.58818	-3.02461	-1.85508	-0.74791
DE-best-target	-5.51597	-2.94033	-1.77527	-0.62989
DE-best-trail	-5.53919	-2.82137	-1.61431	-0.74783
DE-best-normal	-5.64594‡§	-3.14348‡§	-2.15022†‡§	-0.9563‡
DE-best-rand	-5.64783†‡§	-3.24266‡§	-2.02005§	-0.80181
DE-rand	-5.34486	-2.66371	-1.35938	-0.29417
DE-rand-target	-5.36312	-2.33027	-1.25474	-0.2598
DE-rand-trail	-5.29119	-2.46786	-1.29446	-0.26509
DE-rand-normal	-5.35965	-2.55776	-1.15553	-0.27792
DE-rand-rand	-5.32643	-2.51741	-0.9993	-0.25344

B. Discussion on Algorithms Ranking

We apply the Friedman test and Bonferroni-Dunn test on our proposed algorithm and their competitive algorithms. The metric evaluation of critical difference is calculated by Eq. (9). Figure 3 demonstrates the visual presentations of the critical difference between these algorithm ranks. And $k = 4$ for each comparison group (one of our proposed algorithm and

3 competitive algorithms, note that our proposed algorithm is a control method.), and $N = 4$ (4 benchmark problems), q is equal to $q_\alpha(0.01) = 2.936$, and $q_\alpha(0.05) = 2.394$ from Appendix Table B.16 of [16].

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6 * N}} \tag{9}$$

From Figure 3, all of our proposed algorithms are considered as a control algorithm in each sub figures. It indicates that our proposed algorithms can obtain significantly better performance than canonical IDE and triple comparison-based IDE by OBL in significant level of $\alpha < 0.05$. It also demonstrates that the memetic search method to implement a triple comparison mechanism in IDE is better than that implemented by OBL. We will investigate these two implementations of the triple comparison method theoretically in the future.

VI. Conclusion and Future Work

We proposed a triple comparison-based IDE algorithm by memetic search from a fitness landscape obtained by a comparison of target and trail vectors. The local fitness landscape obtained from the original DE algorithm supports information that indicates a promising search region. We implement the memetic search by perturbating the vector with better fitness from a normal distribution and a uniform distribution. The originality of implementing a new triple comparison-based IDE for relieving IDE user fatigue motivates this work. From the evaluation results, we can initially confirm the performance of our proposed algorithm.

In the future, we will investigate parameter setting issue of random number generator, and implementations of our proposal by using other distribution generators. The question as to why a third vector that comes from memetic search is better than that from OBL in theory needs further research. It will be necessary to establish related mathematical models to explain these differences. This and the other issues and problems arising from the current study will be the subject of ongoing work in the future.

References

[1] I. S. Lim and D. Thalmann. Tournament selection for browsing temporal signals. In *Proceedings of the 2000 ACM symposium on Applied computing-Volume 2*, pages 570–573. ACM, 2000.

[2] G. A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[3] Y. Pei. Chaotic evolution: fusion of chaotic ergodicity and evolutionary iteration for optimization. *Natural Computing*, 13(1):79–96, 2014.

[4] Y. Pei. From determinism and probability to chaos: Chaotic evolution towards philosophy and methodology of chaotic optimization. *The Scientific World Journal*, page Article ID 704587, 2014.

[5] Y. Pei. *Study on Effecient Search in Evolutionary Computation.* PhD thesis, Kyushu University, Japan, 2014.

[6] Y. Pei and H. Takagi. Fourier analysis of the fitness landscape for evolutionary search acceleration. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–7. IEEE, 2012.

[7] Y. Pei and H. Takagi. Accelerating iec and ec searches with elite obtained by dimensionality reduction in regression spaces. *Evolutionary Intelligence*, 6(1):27–40, 2013.

[8] Y. Pei and H. Takagi. Fitness landscape approximation by adaptive support vector regression with opposition-based learning. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 1329–1334. IEEE, 2013.

[9] Y. Pei and H. Takagi. Triple and quadruple comparison-based interactive differential evolution and differential evolution. In *Proceedings of the twelfth workshop on Foundations of genetic algorithms XII*, pages 173–182. ACM, 2013.

[10] Y. Pei, Q. Zhao, and Y. Liu. Kernel method based human model for enhancing interactive evolutionary optimization. *The Scientific World Journal*, page Article ID 185860, 2014.

[11] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[12] H. Takagi. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, 2001.

[13] H. Takagi and D. Pallez. Paired comparison-based interactive differential evolution. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 475–480. IEEE, 2009.

[14] H. R. Tizhoosh. Opposition-based learning: A new scheme for machine intelligence. In *CIMCA/IAWTIC*, pages 695–701, 2005.

[15] H. R. Tizhoosh. Opposition-based reinforcement learning. *JACIII*, 10(4):578–585, 2006.

[16] J. H. Zar. *Biostatistical analysis*. Pearson Education India, 1999.