対比較ベースの Memetic 探索による局所 fitness 景観情報を用いた対話型差分進化

Pei, Yan
Computer Science Division, Universally of Aizu

Takagi, Hideyuki
Faculty of Design, Kyushu University

出版情報：進化計算学会研究会. 7, pp.1-, 2014-08-28. 進化計算学会
バージョン：
権利関係：

対比較ベースのMemetic探索による局所fitness景観情報を用いた対話型差分進化

裴岩[†], 高木英行[††],

会津大学　コンピュータ理工部[†]

九州大学 大学院芸術工学研究院[††],

1 Introduction

Evolutionary computation (EC) is a meta-heuristic technique that is used to optimize complex problem, which is hard solved by conventional optimization methods [12]. Interactive EC (IEC) is a niche research field in EC community that embeds feeling, knowledge and experience of a real human into EC optimization, so as to make IEC algorithm convergence to a real human's preference rather than fitness function(s) of an optimized problem. Extending IEC application scale and enhancing IEC algorithm performance (including IEC interface) are two primary research subjects in IEC field. The one is attempting to apply IEC optimization principle and technique to a variety of industrial and business applications that need assistance from a real human in the optimization process. The other pursues to discover more effective and efficient IEC algorithms or interfaces to obtain a better optimization result, meanwhile to relieve human fatigue due to human psychological and physiological limitations when interaction with an IEC algorithm.

There are three research perspectives in IEC for obtaining more effective and efficient IEC algorithm and interface [7]. First is to approximate fitness landscape of subjective evaluation space of a real human, which try to build structures of optimized problem to assist IEC search. Several methods were proposed in this aspect, such as dimensionality technique [9, 1], Fourier analysis [8, 2], etc. Second is to develop a new search mechanism in conventional IEC algorithm to enhance its performance or to design a better interface for communication between human and computer. A triple and quadruple comparison based interactive differential evolution (IDE) follows this research direction [10, 3]. Third is to create new EC/IEC algorithm to achieve better EC/IEC optimization performance [6].

Differential evolution (DE) has characteristic of paired comparison scheme when competition between target vector and trail vector. It benefits to IEC application that allows a real human to give fitness based on paired comparison of two subjects rather than to give multiple fitness values of subjects at the same time. Because human has his/her limitation when supporting fitness to IEC algorithms from one generation to the next, this paired comparison can relieve IEC user fatigue significantly [13]. A triple and quadruple comparison based IDE proposed to use opposite point(s) of target vector and/or trail vector from opposition based learning for implementing triple and quadruple comparison schemes in conventional IDE. It can enhance optimization performance of conventional IDE significantly [10, 3].

By inspiring from multiple comparison implementation in IDE algorithm, fitness landscape can support information on search condition and problem structure to develop a new multiple comparison IDE algorithm. There are a variety of representations of fitness landscape, specifically, fitness of target vector and trail vector is one of representation form in IDE algorithm. When fitness of target vector is better than that of trail vector, it indicates that searching around the target vector is potential to obtain the global optimum with higher probability, and vice versa. In this paper, we propose a new triple comparison based IDE due to this hypothesis, which conducts a memetic search around whichever the better one of target vector and trail vector. We implement the memetic search by perturbation with a normal distribution or a uniform dis-

---

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

[†] Yan PEI (peiyan@u-aizu.ac.jp)

[††] Hideyuki TAKAGI (hideyuki.takagi.457@m.kyushu-u.ac.jp)

Computer Science Division, the University of Aizu, Japan (†)

Faculty of Design, Kyushu University, Japan (††)

tribution. We use a Gaussian mixture model (GMM) as a pseudo IDE user to evaluate our proposal. From evaluation of our proposal, this new triple comparison based IDE algorithm can obtain extraordinary better optimization performance.

Following this introductory section, we briefly review conventional paired comparison based IDE and triple and quadruple comparison based IDE in section 2. Section 3 presents our new triple comparison based IDE based on 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 or a uniform distribution. In section 4, we evaluate our proposal by using GMM. We analyze and discuss some open topics and issues from the evaluation results in section 5. Finally, we conclude whole works, and present some future opportunities in section 6.

2 Interactive Differential Evolution and Multiple Comparison Mechanism

2.1 Paired Comparison Based Interactive Evolutionary Computation

When individuals of an IEC optimization are voice, image or video, i.e., time series object, IEC users have to compare an individual with others in their memory. So IEC users' mental stress and fatigue become heavy. It was pointed that human has a memory limitation and cannot process more than five to nine different information simultaneously [5]. Population sizes of many IEC systems frequently exceed this memory limitation. 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. It is expected to reduce IEC user fatigue. One of its concrete implementations of paired comparison is a tournament interactive genetic algorithm (IGA) [4]. The obtained fitness of tournament IGA has noise because the tournament is not a round robin competition against the original IGA algorithm. The noise influences an IGA selection operation and results to reduce IGA search performance.

2.2 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 (a third random vector from population) to implement mutation operation and obtain a mutant vector. It conducts a crossover operation between the mutant vector and proceeded target vector to create a trail vector. And then, it compares the fitness of target vector and trail vector to survive the better one 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. 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.

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

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

2.3 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 good enough, how about the 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).

$$\begin{aligned} OP(X) &= \{OP(x_1), OP(x_2), ..., OP(x_n)\} \qquad (2) \\ OP(x_i) &= a_i + b_i - x_i \qquad (3) \end{aligned}$$

Algorithm 1 Paired comparison based interactive differential evolution algorithm. 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

2.4 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 [10, 3]. There are three implementations. Two triple comparison based IDEs are implemented by comparing a target vector, a trail vector and either opposite point of a target vector or opposite point of a 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 not a significant difference between the two [10, 3]. So we use the latter one, whole searching ranges, in our experimental evaluation.

3 Memetic Search in Interactive Differential Evolution

3.1 Local Fitness Landscape from Paired Comparison

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

In the IDE, the fitness of target vector and trail vector supports a local fitness landscape when comparing their value of fitness of paired comparison. The individual (either target vector or trail vector) with related great fitness indicates the potential search region, where there is 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 target vector is greater than that of trail vector, it means searching around target vector has a great probability to find the global optimum, and vice versa. The promising direction is from the vector with lower fitness to the vector with higher fitness, and promising searching range is around the vector with higher fitness.

3.2 Memetic Search in Interactive Differential Evolution for Implementing Triple Comparison

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 the 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 to

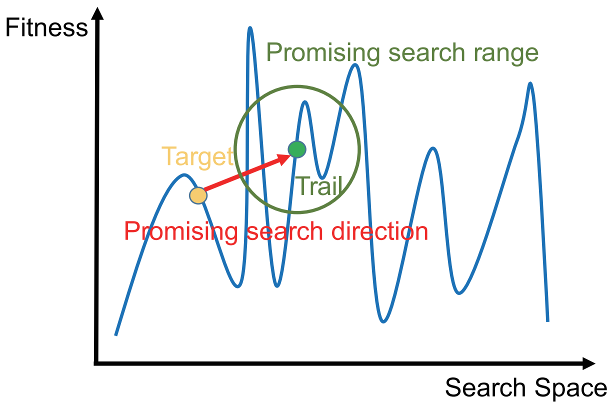

Fig. 1 Local fitness landscape from paired comparison of target vector and trail vector in IDE. The fitness of trail vector is higher than that of target in this figure, that means the promising search range is around the trail vector, promising search direction is from target vector to trail vector,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

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

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

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

Besides a target vector and a trail vector in the canonical DE framework, the third vector is from perturbation on whichever 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 and a uniform distribution, 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.

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

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

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 problem, such as music, or movies that we cannot compare spatially and simultaneously, 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, the user fatigue drastically increases. We pay our attention to this fact, 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 of motivation of multiple comparison mechanism in our proposed IEC algorithm.

4 Optimization Evaluation

4.1 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 that IEC user fatigue becomes low thanks to easy evaluation even if total evaluation time until to the goal is long; there are opposite cases that 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, and after then we need to conduct a human subjective evaluation to confirm the simulation results. This paper handles the IEC simulation of the former stage. We use Gaussian mixture models (GMM) as pseudo IDE user for evaluation in this section. Concretely, 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.

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

where

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

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

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

4.2 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 1. Fig. 2 shows the average convergence curves of best fitness values from 30 trial runs for all 4 benchmark functions. Table 3 shows their means. Abbreviations used in Figures 2 and Table 3 are given in Table 2.

We apply Wilcoxon sign ranked test on our proposed algorithms and their competitive algorithms to evaluate significant differences of two competitors. As well as, 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 of these competitors. Note that we take our proposed algorithm as a control algorithm in Bonferroni-Dunn test.

Table 1 DE and CE experiment parameters setting.

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

Table 2 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 [10] .
DE-best-trail	triple comparison based DE with opposite point of trail vector [10] .
DE-best-normal	triple comparison based DE by memetic search with normal distribution.
DE-best-rand	triple comparison based DE by memetic search with uniform distribution.
DE-rand	standard DE/rand/1/bin [11] .
DE-rand-target	triple comparison based DE with opposite point of target vector [10] .
DE-rand-trail	triple comparison based DE with opposite point of trail vector [10] .
DE-rand-normal	triple comparison based DE by memetic search with normal distribution.
DE-rand-rand	triple comparison based DE by memetic search with uniform distribution.

5 Discussions

5.1 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 of the original one. This is one of motivation in our proposed memetic search in the IDE.

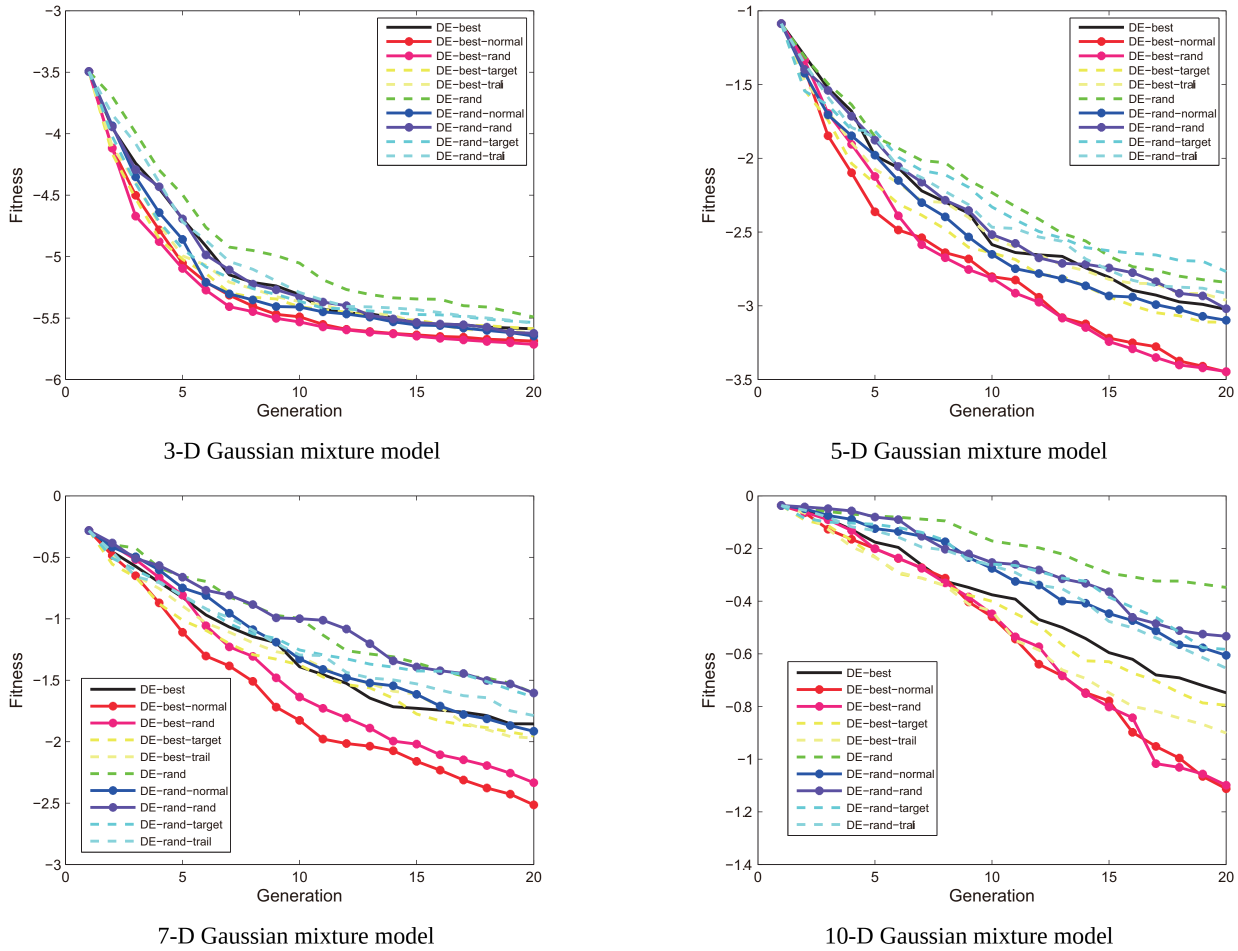

Fig. 2 Average convergence curves for 30 trial runs with 3-D, 5-D, 7-D and 10-D Gaussian mixture models.

Table 3 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†

Figure 2 demonstrates that convergence speed of the proposed algorithms is faster than the corresponding IDE algorithm and triple comparison based IDE algorithms in [10]. We observe that all of our proposed algorithms significantly outperform canonical IDE and triple comparison based IDE algorithms of [10] for the IDE with best vector as base vector from Table 3 by Wilcoxon sign ranked test. However, for the DE algorithms with random vector as base vector, our proposed algorithm performance is not obvious, and proposed algorithm applied in lower dimensional problem is better than that applied to higher dimensional problem. This may be because the memetic search in our proposal can enhance the exploitation capability of IDE and its search range influence performance of our proposal. In our experimental evaluation, we only investigate the performance of our proposal with normal distribution ($N(\mu, \sigma), \mu = 0, \sigma = 1$) and uniform distribution ($I(a, b), a = 0, b = 1$). We will further investi-

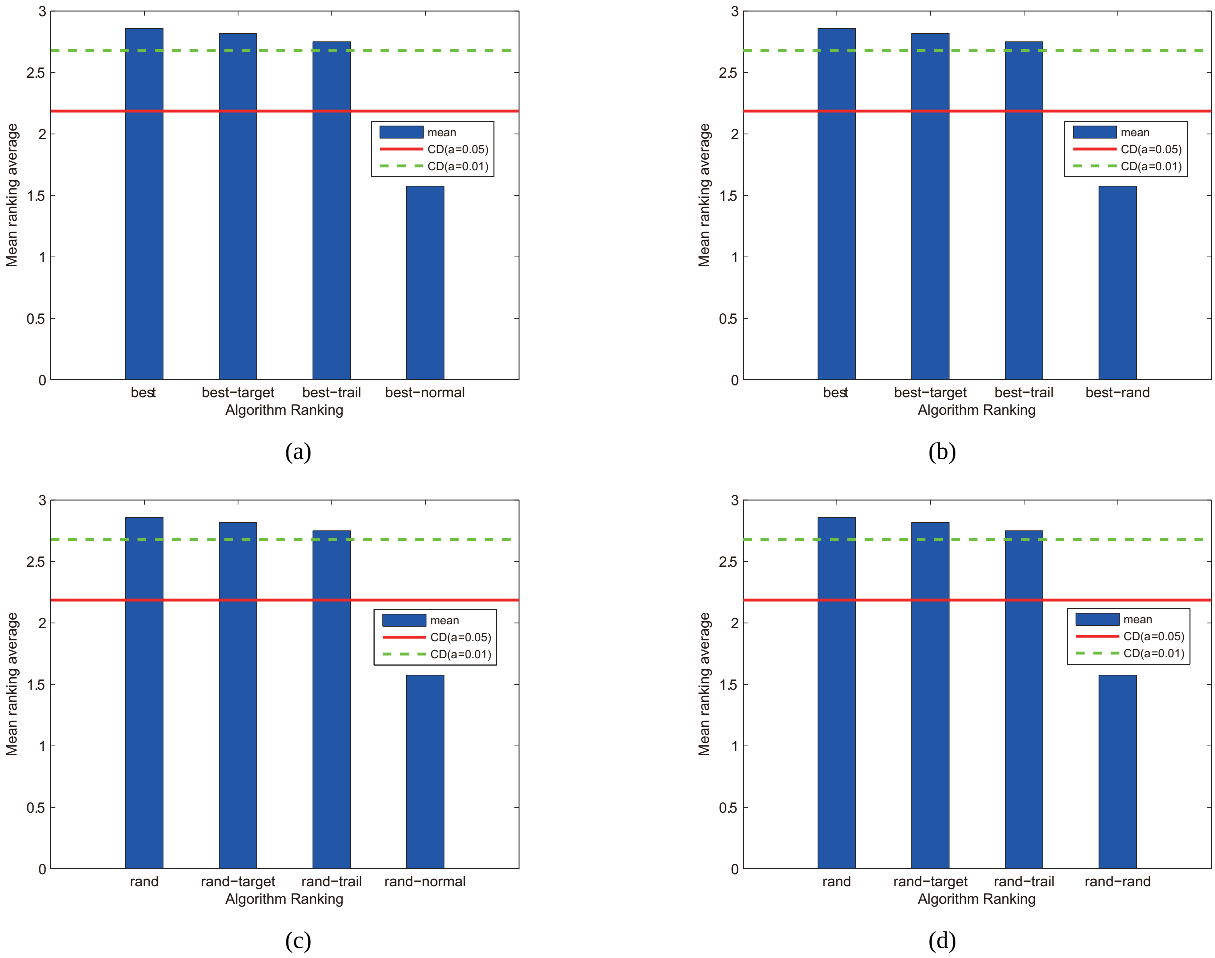

Fig. 3 Bonferroni-Dunn test with taking our proposed algorithm as a control algorithm in each competitive group. From this evaluation, we can conclude that our proposed algorithm are significantly better than their competitors in each group.

gate parameter setting issue in our future work.

We conduct Wilcoxon sign ranked test on our proposed algorithm between with normal distribution perturbation and uniform distribution perturbation. Except rand-normal and rand-rand algorithms applied on 3D and 7D problem, there is not any significant difference between these two category algorithms. We can make a hypothesis that memetic search with different distribution can obtain the same evaluation result.

5.2 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, it indicates that our proposed algorithms can obtain significantly better performance than canonical IDE and triple comparison based IDE by OBL. It does as well as present that 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.

6 Conclusion and Future Works

We proposed a triple comparison based IDE algorithm by memetic search from a fitness landscape obtained by comparison of target and trail vectors. The local fitness landscape obtained from the original DE algorithm supports information that promising search region. We implement the memetic search by perturbation the vector with better fitness from a normal distribution and a uniform distribution. The motivation and originality of this work are implementing a new triple comparison based IDE for relieving IDE user fatigue. From the evaluation result, we initially confirm the performance of our proposed algorithm. We will investigate some mentioned issues and problems in the future.

参考文献

1) 裴岩, 高木英行. 次元削減によって得られたエリートを用いた進化計算の高速化. In **第*1*回進化計算研究会・第*7*回進化計算フロンティア研究会合同研究会**, pages 25–31, 2011.

2) 裴岩, 高木英行. 適応度景観のフーリエ解析による進化的探索高速化の試み. In **進化計算シンポジウム**, pages 167–173, 2011.

3) 裴岩, 高木英行. 3点および4点比較ベースの対話型差分進化と差分進化. **進化計算学会論文誌**, 3(3):98–108, 2012.

4) 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.

5) 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.

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

7) Y. Pei. *Study on Efficient Search in Evolutionary Computation*. PhD thesis, Kysuhu Univeristy, 2014.

8) 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.

9) 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.

10) 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.

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.