Competitive Strategies for Differential Evolution

Jun YU
Graduate School of Design
Kyushu University
4-9-1, Shiobaru, Minami-ku,
Fukuoka, 815-8540 Japan
yujun @kyudai.jp

Yan PEI
Computer Science Division
University of Aizu
Tsuruga, Ikki-machi,
Aizu-Wakamatsu, 965-8580 Japan
peiyan @u-aizu.ac.jp

Hideyuki TAKAGI
Faculty of Design
Kyushu University
4-9-1, Shiobaru, Minami-ku,
Fukuoka, 815-8540 Japan
takagi @design.kyushu-u.ac.jp

***Abstract*—We introduce two competitive strategies into conventional differential evolution (DE) to speed up its convergence by increasing competitive pressures among individuals and evaluate the proposals. The first strategy gives individuals with better fitness a higher opportunity for generating more offsprings, while conventional DE allows each parent individual to generate only one offspring individual fairly. This strategy compares each of poor individuals with a randomly selected individual from the current population. If the latter becomes a winner, the latter can generate one more offspring individual, but the former loses an opportunity for generating its offspring. If the former becomes a winner, no one loses this opportunity, and each of them generates one offspring individual. The second strategy does not compare a generated offspring individual with its parent but the worst individual in the current population, which can accelerate the elimination of poor individuals and keep better individuals. We design a set of controlled experiments to evaluate these two strategies using CEC2013 benchmark functions with three different dimensions. The experimental results indicate that properly enhancing competition among individuals in DE can speed up its convergence and improve optimization performance.**

***Index Terms*—Differential evolution, Evolutionary computation, Optimization, Acceleration convergence, Competition mechanism**

I. Introduction

Evolutionary computation (EC) is one of meta-heuristic optimization techniques and can solve complex optimization problems which are hard for conventional optimization methods. There are three study perspectives in the EC community for obtaining an efficient EC optimization capability. The first one is to approximate the fitness landscapes of target problems and attempt to build their structures to assist EC search. Several methods for dealing with this aspect have been proposed, such as a framework for managing approximate models [1], polynomial models, kriging models and neural networks [2] and Fourier analysis [3]. The second one is to develop new search strategies or mechanisms for enhancing EC optimization performance [4], [5]. The third one is to develop new biological or nature inspired EC algorithms with better EC optimization performance, such as particle swarm optimization [6], differential evolution (DE) [7], bacterial foraging optimization algorithm [8], artificial bee colony algorithm [9], cuckoo search [10], fireworks algorithm [11], bat algorithm [12], krill herd [13], chaotic evolution [14] and many others. Our proposals in this paper are categorized in the above second perspective, and we introduce new competitive mechanisms into conventional DE to enhance its optimization performance.

DE is a type of population-based optimization algorithm and has been widely studied in the EC community. It tries to find out the global optimum by applying a differential-based simple mutation operation for generating offspring candidates and a one-to-one competitive survival strategy for deciding survived individuals, iteratively. Due to its simplicity but high efficiency, many practitioners have dedicated to further improve DE performance and achieved gratifying results [15]. Although many novel and efficient mechanisms have been integrated into DE, few people are concerned about competition mechanisms to enhance its performance.

The main objective of this paper is to propose two competition strategies for increasing the competition among DE individuals, evaluate their effectiveness as well as applicability, and introduce some topics which are open to discussion. Basic strategy of DE is that one parent generates one individual surviving in the next generation. This fair strategy is helpful to keep diversity, avoid premature convergence, and converge to the global optimum gradually. However, this advantageous characteristic sometimes may hinder to escape from hopeless search areas to potential areas quickly. Our proposals aim this quick shift of population.

Following this introductory section, we briefly review DE features and the DE improvement point in the Section II. We explain our two proposals that consider the whole search information of individuals in the Section III and evaluate their optimization performance using CEC 2013 benchmark test suite in the Section IV. Finally, we analyze and discuss some open topics and issues arising from the evaluation results, and conclude the current work and present some future research in the Sections V and VI, respectively.

II. Differential Evolution

A. Feature and Advantage of DE

The feature of DE is to use a differential vector from two random individuals. A differential vector $(\boldsymbol{x1}_i - \boldsymbol{x2}_i)$ using randomly selected two individuals is quite simple, but this directional vector includes the distribution information of population. Suppose to collect all ${}_m\mathrm{C}_2$ differential vectors obtained from m population size, and put their initial points

$\boldsymbol{x2}_i$ ($i = 1, 2, ..., m$) on one point. Then, the distribution of their terminal points $\boldsymbol{x1}_i$ becomes similar to that of population. The lengths of differential vectors become shorter according to the convergence of population, but the distribution information of population is still kept.

B. Improvable Issue of DE

Our research is based on the following hypothesis. In general, better individuals are expected to locate closer to the global optimum than poorer ones and have higher possibility of generating better offspring. Worse individuals may search in hopeless areas or far away from the global optimum and need more computational resources to approach to the global optimum than better individuals. However, canonical DE and several DE variants have a fair policy, and provide an equal opportunity to all parent individuals. This make each of them generate only one offspring individual. We thought that this fair policy may be the point where we can improve DE.

DE compares a parent with its offspring generated in the same local area, and a winner is put in the next generation. This hill climb selection strategy can ensure outstanding offspring replaces its parent so that the whole population evolves steadily towards better areas. However, there must be the case that the replaced parents are better than other individuals in the current population. If poor offspring is generated by a poor parent and is not replaced, computational resources is used for exploring in the same local area until better offspring is generated. This reduces convergence speed and hinders population distribution to converge. This is not a good for especially large-scale optimizations or high computational cost problems. We thought that developing methods for handling poor individuals should be a point where we can improve DE.

III. Competitive Generation and Selection Strategies for DE

Our proposal for the mentioned improvements is to strengthen survival competition. We propose two competitive strategies, *competitive generation strategy* and *competitive selection strategy*, to increase the competitions among individuals and aim to enhance speed of eliminating inferior individuals and increase the possibility of generating more outstanding offsprings. The first strategy, *competitive generation strategy*, makes parent individuals to compete each other to obtain opportunity to generate more offsprings. The second one, *competitive selection strategy*, eliminates the worst individual when population of the next generation is produced.

A. Competitive Generation Strategy

As the first competitive strategy, we propose a competitive mechanism that gives better parent individuals higher opportunities to generate more offspring rather than only one offspring. We firstly mark the worst $a\%$ individuals and compare each of them with a randomly selected individual from the current population. If the latter is better than former, the randomly selected parent can have one more opportunity to generate an offspring and the marked poor parent lose the opportunity to do that. Otherwise, we do nothing and follow the canonical DE processing. This strategy guarantees not to change the population size but allow better individuals to generate more offspring and suppress poor individuals to generate their offspring. The Fig. 1 demonstrates our proposed strategy.

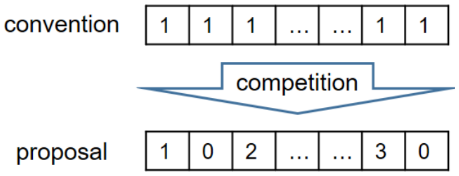


Fig. 1. Canonical DE makes each parent, p_i, generate only one offspring, o_i. Proposed competitive generation strategy makes parents compete each other; some parents can generate multiple offspring, but others may generate no offspring.

B. Competitive Selection Strategy

This second competitive strategy handles the worst individual in a potential hopeless searching area. The competitive selection strategy is a proposal to compare a newly generated offspring, i.e. a trial vector, with the worst individual in the current population and a winner is put in the next generation. The canonical DE compares it with its parent, i.e. a target vector. This strategy ensures that better individuals can survive and poorer individuals are eliminated, which is expected to improve search performance. Algorithm 1 shows the flow of DE with our two proposed strategies.

IV. Experimental Evaluations

We implement two proposed competitive strategies into conventional DE and evaluate their effectiveness using 20 benchmark functions from the CEC2013 benchmark test suite [16] with three dimensional settings, 2-D, 10-D, and 30-D. Their types, characteristics, variable ranges, and optimal fitness values are listed in the reference [16], and their landscape characteristics include shifted, rotated, global on bounds, unimodal and multi-modal. The DE parameter settings in our experiments are shown in the Table I.

TABLE I
DE algorithm parameter setting.

population size for 2-D, 10-D, and 30-D search	100
scale factor F	0.7
crossover rate	0.9
DE operations	DE/rand/1/bin
stop criterion; max. # of fitness evaluations for 2-D, 10-D, and 30-D search	1,000, 10,000, 40,000
dimension of benchmark functions, D	2, 10, and 30
# of trial runs	30

We compare four DE variants to evaluate two proposed strategies: canonical DE, (canonical DE + competitive generation strategy), (canonical DE + competitive selection strategy), and (canonical DE + both strategies). We run these four DE variants using 20 benchmark functions × 3 different

Algorithm 1 DE framework with our two proposed strategies. Step 4 describes the first strategy, and Steps 10-14 describes the second strategy.

```
1:  Initialize population randomly.
2:  Evaluate the population.
3:  while a termination condition is not satisfied do
4:    Determine the number of generated offspring for each parent individual using the first strategy (the total generated offspring is unchanged).
5:    for $i = 1 \ldots$ population size $m$ do
6:      while an individual has the opportunity to generate offspring do
7:        Choose two individuals, $\boldsymbol{x1}_i$ and $\boldsymbol{x2}_i$, randomly and one base vector, $\boldsymbol{base}_i$ from the best individual (DE/best) or randomly (DE/rand).
8:        Calculate a mutant vector as $\boldsymbol{mutant}_i = \boldsymbol{base}_i + F * (\boldsymbol{x1}_i - \boldsymbol{x2}_i)$, where $F$ is a scale factor.
9:        Cross the mutant vector and a target vector and generate a trial vector.
10:       if the generated offspring is better than the worst individual then
11:         Replace the worst one with the offspring.
12:       else
13:         Keep the population unchanged.
14:       end if
15:     end while
16:   end for
17: end while
18: end of program.
```

dimensions × 30 trial run and compare their fitness values at the stop condition, i.e., the maximum number of fitness evaluations. We apply the Friedman test and Holm's multiple comparison test to these fitness values for each benchmark function to check for significant difference among the averages of four DE variants.

Table II shows the result of these statistical tests. Fig. 2 shows the average convergence curves of four methods with 30 trial runs on 2-D benchmark functions. Fig. 3 indicates the average ratio of replacing parent individuals in each generation with 30 trial runs on 2-D benchmark functions.

V. Discussions

A. Analysis of Additional Calculations Cost

Both of two proposed strategies do not increase any extra fitness calculations. The first strategy introduces a new parameter, $a\%$, to control the proportion of individuals participating in the competition. When the a is set to 0, it becomes the conventional DE method. On the contrary, when a is set to 100, all individuals participate in competition. The second strategy changes only the comparison target from a parent to the worst individual in the current population and does not increase computational cost.

We discuss a parameter $a\%$ in the competitive generation strategy. If the fitness of individuals is ranked in the worst $a\%$, they may lose a chance of generating offspring with $1 - a\%$ probability. The average probability that all participating individuals generate no offspring is $\sum_{i=0}^{a\%*PS}(PS - a\%*PS - i)/PS$, where PS means the population size. The poorer the fitness of individuals is, the easier it is to lose an opportunity of generating offspring.

B. Discussion on Competitive Generation Strategy

The first competitive generation strategy gives multiple chances of generating offspring to better individuals, which means that some poor individuals may become unable to generate offspring due to a fixed population size. This feature causes a new problem, i.e., worse individuals that lose the chance are not replaced because of no generated offspring. Along with generations, potential individuals become better and better, but it becomes more and more difficult for these worse individuals to evolve and they remain as they are.

The Fig.s 2 and 3 verify this presumption, i.e., DE with only first strategy is not desirable. Although it ensures the increase of better individuals, it also ensures the increase of worse individuals remaining without evolution. This problem is solved by combining the second competitive selection strategy in the next subsection. The bigger value of the parameter, $a\%$, increases the number of the mentioned poor individuals, but its smaller value reduces the speed of generating better individuals. How to balance this confliction is a crucial issue. We can include that applying the first competitive strategy at every k-th generation instead of every generation to keep the chance that poor individuals evolve; introducing a dynamic parameter, $a\%$, to adjust the proportion of individuals participating in the competition according to generations.

C. Discussion on Competitive Selection Strategy

The second competitive selection strategy is to accelerate eliminating worse individuals by comparing generated offspring with the worst individual. Regardless the superiority of offspring to its parent, the offspring has an opportunity to survive in the next generation when it is better than the worst individual. If it replaces the worst one and the worst one has not yet generated its own offspring, it inherits the opportunity of the worst individual and generates a new offspring. This strategy can accelerate individual evolution towards to the global optimum. However, it also increases the risk of premature convergence, because both parent individual and its offspring in a local area remain, when the offspring is better than the worst individual. This may reduce population diversity. One of ideas to alleviate competitive overpressure and premature convergence is to compare offspring with a randomly selected individual from the worst $p\%$ individuals rather than only the worst individual.

From the experimental results, we can conclude that the proposed strategies, especially the second strategy, significantly improve optimization performance of DE . The proposed strategies have not shown any performance deterioration. We can observe that these strategies make DE convergence faster in early generations. Fig. 3 also supports this conclusion that accelerating individual elimination can speed up DE convergence.

We apply the Friedman test and the Holm multiple comparison test among canonical DE and the combinations of proposed strategies with DE. Although DE with proposed competitive strategies works well for majority of benchmark functions, they do not show significant difference for f_8 (Rotated Ackley's function), f_9 (Rotated Weierstrass function), f_{14} (Schwefel's function), and f_{15} (Rotated Schwefel's function). It may be the reason that these functions have many local optima. We need further analysis of these results to investigate the exact reasons well, and develop more suitable competition strategy for DE optimization.

TABLE II

STATISTICAL TEST RESULTS OF THE FRIEDMAN TEST AND HOLM'S MULTIPLE COMPARISON FOR AVERAGE FITNESS VALUES OF 30 TRIAL RUNS OF 4 METHODS. $A \gg B$ AND $A > B$ MEAN THAT A IS SIGNIFICANT BETTER THAN B WITH SIGNIFICANT LEVELS OF 1% AND 5%, RESPECTIVELY. $A \approx B$ MEANS THAT THERE IS NO SIGNIFICANT DIFFERENCE BETWEEN THEM. NUMBERS IN THE TABLE REPRESENT THAT 1: CANONICAL DE, 2: DE + PROPOSED STRATEGY 1, 3: DE + PROPOSED STRATEGY 2, AND 4: DE + PROPOSED STRATEGIES 1 AND 2.

	2-D	10-D	30-D
f_1	$3 \approx 4 \gg 1 \approx 2$	$4 \approx 3 \gg 1 \approx 2$	$4 \approx 3 \gg 1 \approx 2$
f_2	$3 \approx 4 \gg 1 > 2$	$3 \approx 4 \gg 1 \approx 2$	$4 \approx 3 \gg 2 \approx 1$
f_3	$3 \approx 4 \gg 1 \approx 2$	$3 \approx 4 \gg 1 \approx 2$	$3 \approx 4 \gg 1 \approx 2$
f_4	$3 \approx 4 \gg 1 > 2$	$3 \approx 4 \gg 1 \approx 2$	$3 \approx 2 \approx 4 \approx 1$
f_5	$4 \approx 3 \gg 1 > 2$	$4 \approx 3 \gg 1 > 2$	$4 \approx 3 \gg 1 > 2$
f_6	$3 \approx 4 \approx 1 \approx 2$	$4 \approx 3 \gg 1 \approx 2$	$4 \approx 3 \gg 1 \approx 2$
f_7	$4 \approx 3 \gg 1 \approx 2$	$3 \approx 4 \gg 1 \approx 2$	$4 \approx 3 \gg 1 \approx 2$
f_8	$1 \approx 4 \approx 2 \approx 3$	$2 \approx 1 \approx 3 \approx 4$	$3 \approx 4 \approx 1 \approx 2$
f_9	$4 \approx 3 \approx 2 \approx 1$	$4 \approx 2 \approx 1 \approx 3$	$3 \approx 4 \approx 1 \approx 2$
f_{10}	$2 \gg 4 \approx 3 \gg 1$	$3 \approx 4 \gg 1 \approx 2$	$4 \approx 3 \gg 1 \approx 2$
f_{11}	$4 \approx 3 \approx 1 \approx 2$	$4 \approx 3 \approx 1 \approx 2$	$3 \approx 4 \approx 1 \approx 2$
f_{12}	$3 \approx 4 \approx 1 \approx 2$	$4 \approx 3 > 1 \approx 2$	$3 \approx 4 \approx 2 \approx 1$
f_{13}	$4 \approx 3 \approx 1 \approx 2$	$4 \approx 3 \approx 1 \approx 2$	$4 \approx 3 \gg 2 \approx 1$
f_{14}	$4 \approx 3 \approx 2 \approx 1$	$4 \approx 3 \approx 1 \approx 2$	$4 \approx 1 \approx 2 \approx 3$
f_{15}	$4 \approx 1 \approx 3 \approx 2$	$3 \approx 1 \approx 4 \approx 2$	$3 \approx 1 \approx 2 \approx 4$
f_{16}	$2 \approx 3 \approx 4 \approx 1$	$3 \approx 2 \approx 4 > 1$	$3 \approx 2 \approx 1 \approx 4$
f_{17}	$4 \approx 2 \approx 3 \approx 1$	$4 \approx 3 \gg 1 > 2$	$3 \approx 4 \gg 1 \approx 2$
f_{18}	$3 \approx 4 \approx 2 \approx 1$	$4 \approx 3 \gg 1 \approx 2$	$3 \approx 4 \gg 1 \approx 2$
f_{19}	$4 \approx 3 \approx 1 \approx 2$	$4 \approx 3 \gg 1 \approx 2$	$4 \approx 3 \gg 1 \approx 2$
f_{20}	$4 \approx 3 \approx 2 \approx 1$	$3 \approx 4 > 1 \approx 2$	$3 \approx 4 \approx 2 \approx 1$

VI. CONCLUSION AND FUTURE WORKS

We proposed two strategies, the competitive generation strategy and the selection strategy, for the DE algorithm to increase competition among individuals and accelerate its convergence. The controlled experiments confirmed that proper introduction of the competition mechanism can improve the optimization performance of canonical DE. Concretely speaking, the competitive selection strategy is effective, which eliminates the worst individual and generates its alternative near a better individual. However, the acceleration performance of the competitive generation strategy that gives potential individuals more opportunities to generate offspring is not obvious.

In future work, we will further study the proposed competitive strategies and try to obtain a balance between competition and cooperation. Consequently, how to rationally eliminate poor individuals and maintain the diversity of population is also an open topic for further investigation. We will discuss and analyse these topics.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grant Number 18K11470 and 17H06197.

REFERENCES

[1] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.

[2] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005.

[3] Y. Pei and H. Takagi, "Fourier analysis of the fitness landscape for evolutionary search acceleration," in *2012 IEEE Congress on Evolutionary Computation, CEC 2012*, 2012, pp. 1–7.

[4] T. Back, U. Hammel, and H. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.

[5] R. J. Mullen, D. Monekosso, S. Barman, and P. Remagnino, "A review of ant algorithms," *Expert Systems with Applications*, vol. 36, no. 6, pp. 9608–9617, 2009.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks - Conference Proceedings*, vol. 4, 1995, pp. 1942–1948.

[7] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341– 359, 1997.

[8] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems*, vol. 22, no. 3, pp. 52–67, 2002.

[9] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[10] X. Yang and S. Deb, "Cuckoo search via lvy flights," in *2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings*, 2009, pp. 210–214.

[11] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6145 LNCS, no. PART 1, 2010, pp. 355–364.

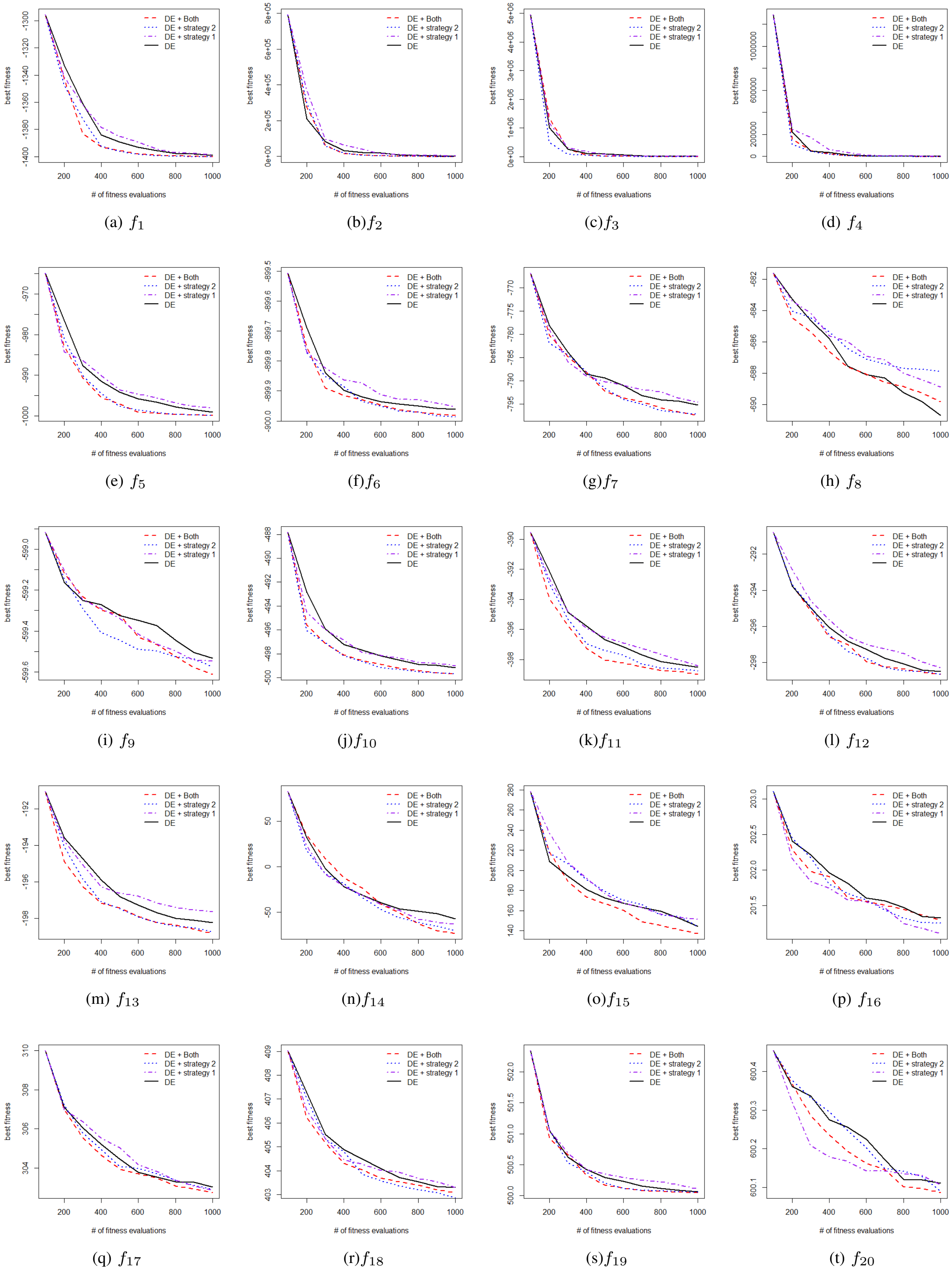
[12] X. Yang and A. H. Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Engineering Computations (Swansea, Wales)*, vol. 29, no. 5, pp. 464–483, 2012.

[13] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.

[14] Y. Pei, "Chaotic evolution: Fusion of chaotic ergodicity and evolutionary iteration for optimization," *Natural Computing*, vol. 13, no. 1, pp. 79–96, 2014.

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

[16] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization," 2013. [Online]. Available: http://al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimization/CEC2013_RealParameterOptimization_TechnicalReport.pdf



(a) f_1 (b) f_2 (c) f_3 (d) f_4

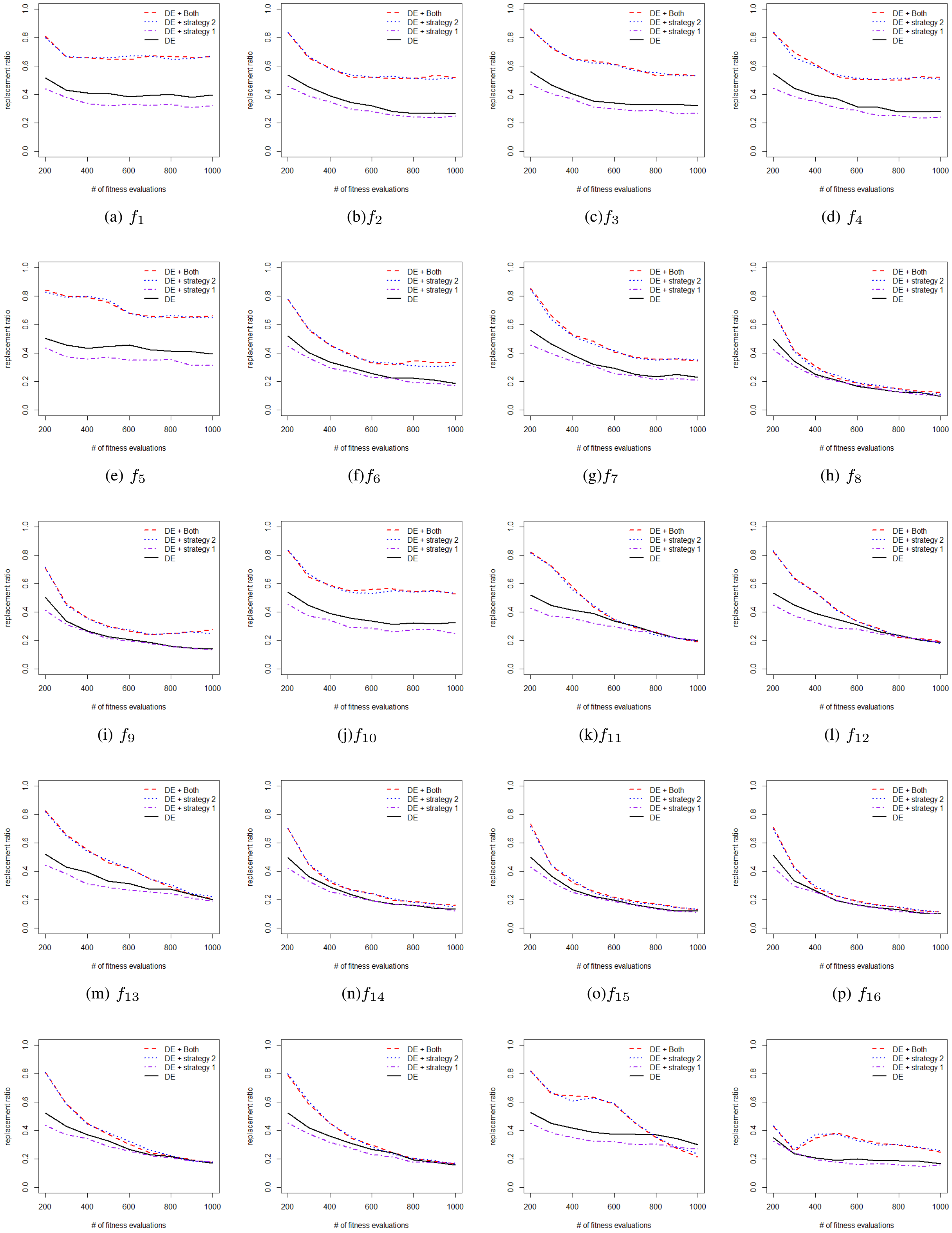
(e) f_5 (f) f_6 (g) f_7 (h) f_8

(i) f_9 (j) f_{10} (k) f_{11} (l) f_{12}

(m) f_{13} (n) f_{14} (o) f_{15} (p) f_{16}

(q) f_{17} (r) f_{18} (s) f_{19} (t) f_{20}

Fig. 2. Convergence curves of 2-D f_1–f_{20} benchmark functions. We can observe that DE with proposed two competitive strategies can accelerate DE search.



(a) f_1 (b)f_2 (c)f_3 (d) f_4

(e) f_5 (f)f_6 (g)f_7 (h) f_8

(i) f_9 (j)f_{10} (k)f_{11} (l) f_{12}

(m) f_{13} (n)f_{14} (o)f_{15} (p) f_{16}

(q) f_{17} (r)f_{18} (s)f_{19} (t) f_{20}

Fig. 3. Average ratio of replacing parent individuals for 2-D f_1–f_{20} benchmark functions. We can observe that the ratio of replacing parent individuals of DE with strategy 2 is higher than that of DE with strategy 1.