Swarm and Evolutionary Computation 83 (2023) 101403

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Differential evolution with alternation between steady monopoly and transient competition of mutation strategies

Chenxi Ye [a], Chengjun Li [a,b], Yang Li [c,*], Yufei Sun [d], Wenxuan Yang [a], Mingyuan Bai [e], Xuanyu Zhu [a], Jinghan Hu [a], Tingzi Chi [a], Hongbo Zhu [a], Luqi He [f]

[a] School of Computer Science, China University of Geosciences, Wuhan 430078, PR China
[b] Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430078, PR China
[c] School of Geography and Information Engineering, China University of Geosciences, Wuhan 430078, PR China
[d] Engineering - Master of information technology, University of New South Wales, Kensington 2052, Australia
[e] RIKEN AIP, Tokyo 1030027, Japan
[f] School of Mechanical Electronic and Electronic Information, China University of Geosciences, Wuhan 430074, PR China

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Differential evolution
Monopoly and competition
Ensemble
Diversification

ABSTRACT

Real parameter single objective optimization has been studied for decades. In recent, long-term search is emphasized based on the consideration that, in the field, solving difficulty often scales exponentially with the increase of function dimensionality. For long-term search, Differential Evolution (DE) still performs outstanding among types of population-based metaheuristics. In this paper, based on IMODE – a DE algorithm with three mutation strategies, we propose AMCDE – Differential Evolution with Alternation between steady Monopoly and transient Competition of mutation strategies. Our algorithm has two states. In the steady state – monopoly, a selected mutation strategy controls the whole population. Once improvement in fitness becomes difficult, the transient state – competition – arises. In the competition state, similar with IMODE, each of the mutation strategies controls a proportion of positions in the population and competes with the others. For enhancement, we propose that positions controlled by the winner among the three mutation strategies continue to be controlled by the mutation strategy in the next generation. Besides, in the competition state, the original selection strategy of IMODE is revised by us for diversification, while adaptation of the crossover rate is updated. Our experiment is based on three CEC benchmark test suites and the CEC 2011 suite of real world optimization problems. AMCDE is compared with seven peers. Based on experimental results, our algorithm demonstrates superior or at the very least comparable performance for long-term search compared to the peers. Moreover, we do experimental observation on AMCDE.

1. Introduction

Real parameter single objective optimization is for searching the best decision vector to minimize (or maximize) objective function. In recent decades, multiple types of population-based metaheuristic are proposed for such optimization. Among the types, Differential Evolution (DE) [1] has gained significant attention from researchers due to its high performance.

Mutation, crossover and selection are the main operators in DE. At the beginning of execution, individuals $\vec{x}_{i,0} = (x_{1,i,0}, x_{2,i,0}, \dots, x_{D,i,0})$, ($i = \{1, 2, \dots, NP\}$), where NP denotes the population size and D denotes dimensionality, are initialized. Individuals of DE are also called target vectors. During the gth generation, mutant vectors, denoted as $\vec{v}_{i,g}$, are generated through mutation based on target vectors. We list one of the basic mutation strategies DE/rand/1 below for example.

$$\vec{v}_{i,g} = \vec{x}_{r1,g} + F \cdot (\vec{x}_{r2,g} - \vec{x}_{r3,g}), \tag{1}$$

where $r1 \in \{1, 2, \dots, NP\}$, $r2 \in \{1, 2, \dots, NP\}$, $r3 \in \{1, 2, \dots, NP\}$, and $r1 \neq r2 \neq r3 \neq i$. F represents the important parameter of mutation – scaling factor. Then, based on $\vec{x}_{i,g}$ and $\vec{v}_{i,g}$, $\vec{u}_{i,g} = (u_{1,i,g}, u_{2,i,g}, \dots, u_{D,i,g})$ – trial vectors – are generated by crossover. Binomial crossover,

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & if\ rand(0,1) \leq Cr\ or\ j = j_{rand}, \\ x_{j,i,g}, & otherwise, \end{cases} \tag{2}$$

is used in majority of DE algorithms. In Eq. (2), $j \in \{1, 2, \dots, D\}$, the crossover rate $Cr \in [0, 1]$, and $j_{rand} = randint(1, D)$ is an integer randomly generated from the range $[1, D]$ to ensure that $\vec{u}_{i,g}$ has at least

* Corresponding author.
E-mail address: liyang_cs@cug.edu.cn (Y. Li).

https://doi.org/10.1016/j.swevo.2023.101403
Received 4 March 2023; Received in revised form 21 September 2023; Accepted 23 September 2023
Available online 26 September 2023

one component from $\vec{v}_{i,g}$. On many occasions, crossover and mutation of DE together are specified as trial vector generation strategy. The widely used method for selection is

$$\vec{x}_{i,g+1} = \begin{cases} \vec{u}_{i,g}, & if\ f(\vec{u}_{i,g}) \le f(\vec{x}_{i,g}), \\ \vec{x}_{i,g}, & otherwise, \end{cases} \tag{3}$$

where $f(\cdot)$ represents fitness obtained by function evaluation. During execution, the three operators are repeated in a loop until the maximum number of function evaluations, denoted as $MaxFES$, is reached.

Under the control of the operators, provided that the whole population is initialized within a sufficiently large attraction basin of a single local optimum, the population can hardly leave this basin. Formal proof of this statement is given by [2]. The above feature makes that DE is prone to stagnation. Based on such a background, DE has been studied for decades in at least four key aspects,

- Improvement of the operators;
- Adaptation of parameter;
- Hybridization with other search techniques; and
- Ensemble of operators or even DE algorithms.

In fact, in the whole community of evolutionary computation, the second aspect is attracting attention from researchers [3,4]. In Section 2, existing DE algorithms reviewed by us are categorized into the above four aspects. In addition, DE has been revised for tasks more complex than real parameter single objective optimization, e.g., multimodal optimization [5], constrained optimization [6], large-scale optimization [7] and multi-objective optimization [8].

Recently, long-term search of real parameter single objective optimization is concerned based on the consideration that, in this field, solving difficulty often scales exponentially with the increase of D. Here, long-term search means that, with the increase of D, $MaxFES$ scales exponentially. In fact, for years, $MaxFES$ is set just linearly with the increase of D. For example, from the first CEC competition on real parameter single objective optimization held in 2013 [9] to the CEC 2018 competition, $MaxFES = D \cdot 1.00E{+}04$. So far, in literature, $MaxFES$ is set as above on most occasions. However, from 2020, long-term search becomes the new focus in CEC competition on real parameter single objective optimization [10] and then is studied elsewhere.

According to our data analysis, to fit for long-term search, population-based metaheuristic suited to traditional search requires to be revised greatly. Otherwise, algorithm converges quickly in the beginning, and then stagnates soon. Recently, population-based metaheuristics for long-term search are proposed in literature. Among the proposed algorithms, IMODE [11], a type of DE ensemble, has good performance. In fact, the algorithm is the winner of the CEC 2020 competition.

Among the three operators of DE, mutation plays a vital role [12]. IMODE possesses three mutation strategies, DE/current-to-ϕbest/1 with archive,

$$\vec{v}_{i,g} = \vec{x}_{i,g} + F_i \cdot (\vec{x}_{\phi,g} - \vec{x}_{i,g} + \vec{x}_{r1,g} - \vec{x}_{r3,g}), \tag{4}$$

DE/current-to-ϕbest/1,

$$\vec{v}_{i,g} = \vec{x}_{i,g} + F_i \cdot (\vec{x}_{\phi,g} - \vec{x}_{i,g} + \vec{x}_{r1,g} - \vec{x}_{r2,g}), \tag{5}$$

and DE/weighted-rand-to-ϕbest/1,

$$\vec{v}_{i,g} = F_i \cdot \vec{x}_{r1,g} + F_i \cdot (\vec{x}_{\phi,g} - \vec{x}_{r2,g}). \tag{6}$$

The parameters in the equations different with the ones in Eq. (1) are explained below. Here, $\vec{x}_{r3,g}$ are randomly chosen from population or the external archive A. That is, $r3 \in \{1, 2, \ldots, NP + |A|\}$. In fact, A is used to store individuals eliminated from the population. $|A|_{min} = 0$, while $|A|_{max} = NP$. After $|A| = |A|_{max}$, random individual in A is replaced by new comer. Moreover, $\vec{x}_{\phi,g}$ is among the best ϕ individuals in the gth generation. In addition, F_i is F for the ith position of the population. Meanwhile, IMODE employs the binomial or exponential crossover,

$$u_{j,i,g} = \begin{cases} \begin{cases} v_{j,i,g}, & if\ rand(0,1) \le Cr\ or\ j = j_{rand}, \\ x_{j,i,g}, & otherwise, \end{cases} & if\ rand(0,1) \le p \\ \begin{cases} v_{j,i,g}, & for\ j = \langle l \rangle_D, \langle l+1 \rangle_D, \ldots, \langle l+L-1 \rangle_D \\ x_{j,i,g}, & for\ all\ other\ j \in [1, D], \end{cases} & otherwise \end{cases} \tag{7}$$

The parameters in Eq. (7) different with the ones in Eq. (2) are explained below. p is the probability of binomial manner in crossover. Both l and L are integers in the range $[1, NP]$. l is randomly chosen, while L requires to be set. In IMODE, the two parameters, F and Cr, are set according to [13], while the external archive is maintained according to [14]. For selection, Eq. (3) is used. Let $\overline{\Delta f}_k$ be the average improvement from target vector to trial vector among all the positions in the population controlled by the kth mutation strategy. The controlling ratio of the kth mutation strategy c_k is computed as below,

$$c_k = \frac{\overline{\Delta f}_k}{\sum_{i=1}^{3} \overline{\Delta f}_i}, \tag{8}$$

where $k = \{1, 2, 3\}$. Then, c_k is adjusted to ensure $c_k \ge 0.1$ on the constant premise of $\sum_{i=1}^{3} c_k = 1$. In detail, if $c_m < 0.1$ ($m \in 1, 2, 3$), c_m is turned to 0.1 by reducing c_{max}. In the next generation, $c_k \cdot NP$ positions are allocated to the kth mutation strategy randomly. At the last stage of execution, sequential quadratic programming is called in successive generations to the best individual as local search for further improvement.

Our motivation is given below. Although IMODE performs well for long-term search, there is still room for further improvement. For example, in IMODE, positions in the population are seized by the mutation strategies without any consideration on historical ownership. In fact, ownership of position is not a problem in most DE algorithms. Just for ensemble-based DE algorithms, it is possible that the ownership is changed during execution. Even in this case, the stability in the ownership needs to be considered. In many types of ensemble with DE, e.g., either MPEDE [15] and EDEV [16], which are suited to traditional search, or APGSK-IMODE [17] and APGSK-IMODE-FL [18], which are suited to long-term search, each mutation strategy or constituent DE algorithm has its relatively stable subpopulation. That is, most of positions are possessed stably by a mutation strategy or constituent DE algorithm. Moreover, in other types of ensemble, e.g., L-SHADE-E [19], where subpopulations are not divided from the population, only if no better offspring can be obtained, another constituent DE algorithm replaces the current one to control the whole population. In brief, beside IMODE, it is hardly to find another DE ensemble with no consideration on historical ownership of position. We think that, for enhancing IMODE, the historical ownership of position needs to be considered to some extent.

In this paper, we propose a new type of ensemble named Differential Evolution with Alternation between steady Monopoly and transient Competition of mutation strategies – AMCDE – based on IMODE. AMCDE still employs the three mutation strategies used by IMODE. However, different with IMODE, our AMCDE has two alternative states, monopoly and competition. The monopoly state means that a selected mutation strategy controls the whole population, while the competition state means that the three mutation strategies share the population and compete with each others. Undoubtedly, in the monopoly state, the historical ownership of position is not a problem. To consider the historical ownership of position in the competition state, we propose that positions controlled by the winner among the three mutation strategies continue to be controlled by the mutation strategy in the next generation. For matching the above competition, the original selection strategy of IMODE is revised by us for diversification, while adaptation of the crossover rate is updated. As soon as the best fitness is updated, competition terminates while monopoly goes back. In the new round of monopoly, the mutation strategy leading to the breakthrough in fitness becomes the new monopolizer.

Our experiment is carried out based on the CEC 2017, 2020 and 2022 benchmark test suites [10,20,21] and the CEC 2011 suite of real world optimization problems [22]. For comparison, we select excellent algorithms for long-term search as peers. Based on experimental results, our algorithm demonstrates superior or at the very least comparable performance compared to the peers. Moreover, based on our experimental observation on AMCDE, our algorithm is better understood.

The rest of this paper is organized as below. Section 2 provides a review of related work. In Section 3, our AMCDE is presented. In Section 4, we demonstrate and analyze the experimental results. Finally, Section 5 concludes the paper.

2. Related work

This section consists of three parts. Firstly, we review recent DE algorithms suited to traditional search. Then, we review population-based metaheuristics suited to long-term search. Finally, adjustment of population size for DE is reviewed because, for real parameter single objective optimization, the technique is widely considered in types of population-based metaheuristics including DE.

2.1. Recent DE algorithms suited to traditional search

So far, research on traditional search is still carried out extensively. EFADE [23] employs triangular mutation. Meanwhile, adaptation schemes are used to update F and Cr, respectively. BSDE [24] requires no parameter setting. In [25], CJADE is obtained by incorporating chaotic local search mechanisms into DE. PaDE [26] includes three aspects of improvement – an adaptation scheme for crossover rate, the parabolic population size reduction scheme, and the mutation strategy based on enhance time stamp. In RAM-JAPDE [27], DE/pbest/1 or DE/current-to-pbest/1 is employed with a probability decided by performance, while joint adaptation of parameters is considered. DMCDE [28] calls DE/e-rand/2 and DE/e-best/2 based on a collaboration scheme. CSDE [29] calls two mutation strategies based on historical success. Meanwhile, parameters are set independently for individual. In TVDE [30], both mutation operator creating and parameters setting are based on three time-varying functions. MBADE [31] is a version of bat algorithm is hybridized with DE. FDDE [32] is with mutation based on both fitness and diversity ranking. In such mutation, both fitness and diversity contribution of individual is considered. BeSDE [33] is with the partially elitist mutation and employs the crossover based on Bernstein polynomials. In [34], a framework based on search space adjustment is proposed for improving DE algorithm. Provided that an individual cannot be improved for several generations, an adjustment mechanism is triggered to generate a substitute individual. In [35], adaptive selection of mutation strategy is realized based on deep reinforcement learning. Then, based on the above technique, DEDQN is proposed. FLDE [36] is with adaptive mutation based on fitness landscape. In [19], ensemble named L-SHADE-E based on L-SHADE-EpSin and L-SHADE-RSP is proposed. In L-SHADE-E, one of the constituent algorithms is chosen randomly to control the population at the beginning. Provided that, for generations, progress in fitness cannot be obtained, the other constituent algorithm takes over the population. In FADE [37], the whole population is divided into multiple swarms, and three popular trial vector generation strategies can be utilized by the swarms. In addition, the population size is adjusted adaptively based on performance. In APSDE [38], beside the population, an accompanying population is maintained. Individuals in the accompanying population are composed of suboptimal solutions. Based on the accompanying population, mutation is realized, while parameters are optimized. In addition, reverse individuals are generated at occasions for diversity.

The above research findings can be categorized into the four study aspects. PaDE, TVDE, FDDE, BeSDE, ARSA, FLDE, and APSDE are with improvement of operators. RAM-JAPDE, PaDE, CSDE and TVDE embody adaptation of parameter. BSDE, CJADE, and MBADE belong to hybridization with other search techniques. RAM-JAPDE, CSDE, DEDQN, L-SHADE-E, FADE are ensemble of operators or even DE algorithms.

Our AMCDE belongs to ensemble of multiple trial vector generation strategies. More exactly, AMCDE is based on adaptation of mutation strategies. The direction has been studied for years. ZEPDE [39] is one of the early DE algorithms with adaptation of mutation strategies. In the algorithm, five mutation strategies – DE/rand/1, DE/current-to-best/2, DE/current-to-best/1, DE/best/2, and DE/rand/2 – seize individuals randomly at the late stage. Moreover, the algorithms mentioned above – RAM-JAPDE, DMCDE, CSDE, DEDQN, and FLDE – are also with adaptation of mutation strategies.

2.2. Population-based metaheuristics suited to long-term search

So far, algorithms suited to long-term search are mainly proposed in the latest CEC competitions. However, recently, we can find algorithms suited to long-term search elsewhere. Details are given below.

As mentioned before, the winner of the CEC 2020 competition is IMODE. AGSK [40], an adaptive variant of GSK [41], ranks second in the CEC 2020 competition. In AGSK, junior gaining-sharing knowledge phase and senior gaining-sharing knowledge phase occur in different dimensions, respectively, under the control of an adaptive scheme. The algorithm j2020 [42], which is adapted from jDE100 [43] by employing a crowding mechanism and a scheme to choose individual for mutation, ranks third in the competition.

APGSK-IMODE, the winner of non-shifted cases in the CEC 2021 competition, is ensemble of APGSK [44] – another adaptive variant of GSK – and IMODE. In the type of ensemble, each of the two constituent algorithms is allocated a subpopulation. At intervals, the current best individual is shared by the two subpopulations. NL-SHADE-RSP [45], the winner of both shift cases and rotated shifted cases in the competition, is revised based on L-SHADE-RSP [46]. In the algorithm, selective pressure is adjusted by changing setting of mutation strategy slightly. Furthermore, usage probability of the archive is tuned automatically. The winner for non-rotated shifted cases, jDE-21 [47], employs a restart mechanism for diversity maintenance. Meanwhile, the crowding mechanism and the scheme to choose individual for mutation proposed in j2020 is used here after revision.

EA4eig [48] ranks first in the CEC 2022 competition. Based on the ensemble of four population-based metaheruistics – CoBiDE [49], IDEbd [50], CMA-ES [51], and jSO [52] – proposed in [53], EA4eig is obtained by extending the application range of the Eigen approach from just one constituent algorithm – CoBiDE – to all the ones. NL-SHADE-LBC [54], the algorithm ranks second in the competition, is with mutation and crossover modified based on those of L-SHADE-RSP and NL-SHADE-RSP. Meanwhile, parameters are based on the linear bias change. NL-SHADE-RSP-MID [55] ranks third and revised from NL-SHADE-RSP. To improve NL-SHADE-RSP, the algorithm adopts following measures – updating the best individual by considering the midpoint of the population, a triggering scheme for restart, dividing the population by clustering based on k-means.

APGSK-IMODE-FL, an improved version of APGSK-IMODE, demonstrates that research on long-term search is no longer just developed in CEC competition. In the algorithm, the communication scheme between the two constituent algorithms is updated. In detail, at intervals, APGSK and IMODE attempt to exchange individuals among the worst ones having lived for enough generations. If such an individual cannot be found in at least one constituent algorithm, the original method of communication in APGSK-IMODE is still used.

Also, the above algorithms for long-term search can be categorized into the four study aspects. Firstly, j2020, NL-SHADE-RSP, jDE-21, NL-SHADE-LBC, and NL-SHADE-RSP-MID focus on improvement of the operators. Then, NL-SHADE-LBC embodies adaptation of parameter. After that, APGSK-IMODE, EA4eig, and APGSK-IMODE-FL belong to hybridization with other search techniques. Finally, as before mentioned, IMODE belongs to ensemble of operators, and is based on adaptation of mutation strategies. In addition, the variant of GSK – AGSK – is not based on DE although has operators similar to ones used in DE.

2.3. Population size adjustment

In recent years, most population-based metaheuristics are with population size adjustment to save times of function evaluation. In the majority of the above algorithms listed as related work, population size is reduced during execution according to an equation. However, in a few algorithms, population size is adaptively adjusted according to performance. Details are given below.

Among all of the algorithms, most ones suited to traditional search and a few ones suited to long-term search employ the linear population size reduction. However, the other ones suited to long-term search, e.g. AGSK, APGSK-IMODE, NL-SHADE-RSP, NL-SHADE-LBC, NL-SHADE-RSP-MID, and APGSK-IMODE-FL, employ the non-linear population size reduction [40]. That is, in execution, population size decreases rapidly in the early stage but slowly in the late stage. For long-term search, much more times of function evaluation are given than before. In this case, the non-linear population size reduction leads to superior solution compared to the linear one [40]. According to our analysis, the former scheme saves more times of function evaluation than the latter one, especially when $MaxFES$ is set large enough. Besides, adaptive adjustment is adopted in the two DE algorithms suited to traditional search, PaDE and FADE.

3. AMCDE

In IMODE, the three mutation strategies seize positions in the population according to performance. Compared with most existing types of DE ensemble, a significant difference of IMODE is that position in the population is seized by mutation strategy without any consideration about historical ownership.

The two mutation strategies in IMODE – DE/current-to-ϕbest/1 with archive and DE/current-to-ϕbest/1 – are good in performance [14] and cooperate well with each other in ensemble [56]. Nevertheless, the two mutation strategies are similar in action manner. To further reflect the difference in action manner of mutation strategies, DE/weighted-rand-to-ϕbest/1, which is very different in action manner with most of other existing mutation strategies, is adopted beside the above two ones. In fact, the mutation strategy is seldom used independently in algorithm because is structurally biased in searching. However, it is reasonable that the mutation strategy is integrated into ensemble as a measure for diversification. In IMODE, each mutation strategy controls a number of positions and thus affects the distribution of individuals. Although DE/weighted-rand-to-ϕbest/1 may not lead to approaching the best decision vector at all on occasions, the mutation strategy does make a contribution to change the distribution of individuals.

Both position seized by the three mutation strategies without consideration on historical ownership and the use of the third mutation strategy slow down convergence velocity and broaden searching range. Thus, IMODE is good at long-term search and performs best in the CEC 2020 competition. However, we think that a search method should not be disturbed when is working efficiently. Thus, monopoly of a mutation strategy should be the normal or steady state in ensemble, while competition of mutation strategies should just be the transient response to the phenomenon that the current mutation strategy cannot make progress any more. Moreover, we believe that the distribution of mutation strategy in the population is excessively random in IMODE. Thus, proper consideration should be given to historical ownership of position in the competition state.

In this paper, we propose AMCDE based on IMODE. In most of generations, just one of the three mutation strategies is used. Such a steady state – monopoly – is broken if the best fitness in the population has never been improved for G_n generations. Then, the three mutation strategies work together to get out of the dilemma. That is, the transient state – competition – arises. In the state, after every generation, the average improvement from target vector to trial vector made by each mutation strategy $\overline{\Delta f}_k$ is calculated as in IMODE. Based on the calculation, the mutation strategy best in performance is found. In each generation with competition, one third positions are randomly allocated to each of the three mutation strategies. Moreover, excepting in the first generation of a round of competition, positions controlled by the mutation strategy with the best performance in the previous generation are given back to the mutation strategy as a reward if are allocated to the other mutation strategies. As a result, the mutation strategy with the best performance in the previous generation may control more positions than the other two mutation strategies. AMCDE also employs Eq. (7) for crossover as IMODE. However, in the competition state, the crossover rate of the binomial mode in the binomial or exponential crossover in Eq. (7), Cr, is adaptively set as below,

$$Cr = \begin{cases} 0, & if\ FES < MaxFES \cdot 0.5 \\ (\frac{FES}{MaxFES} - 0.5) \cdot 2, & otherwise. \end{cases} \tag{9}$$

The equation is coming from NL-SHADE-RSP. During competition, based on Eq. (3), selection is revised as below. At a probability p_r, the worst p_w target vectors in the population is replaced by their worse trial vector. As soon as the best fitness in the population is updated, competition is halted and then replaced by monopoly in the next generation. Naturally, the mutation strategy obtaining the new best fitness becomes the new monopolizer.

The revisions based on IMODE for our AMCDE can be explained as below. We give a chance to just one of the mutation strategies to control the whole population. However, if the mutation strategy has lost its ability to obtain better solution, all the mutation strategies compete for the next chance of controlling. In the competitions state, if a mutation strategy performs best, as reward, the strategy is given more positions including all the ones having controlled by itself. Moreover, it is for diversification that, during competition, at a probability, the worst individuals are replaced by their offspring with even worse fitness. Then, we study crossover parameter adaptation in the competition state. In brief, the schemes proposed by us include the main one – alternation between monopoly and competition, and the other two ones – the revision in selection and our Cr adaptation.

The pseudo-code of AMCDE is shown in Algorithm 1. In Algorithm 1, Algorithm 2 is called. Among the input items of Algorithm 1, NP_{init} and NP_{min} represent the initial and the final population size, respectively. Moreover, p_{bc1} and p_{bc2} are the probability of the binomial crossover in Algorithm 1 and Algorithm 2, respectively. The output S is the final solution. It can be seen that, the mutation strategy – DE/current-to-ϕbest/1 with archive – is the initial monopolizer. After all, the mutation strategy is good at maintaining diversity because an external archive is employed to store individuals eliminated from the population for following difference operation in mutation.

4. Experiment

We choose the CEC 2020 and 2022 benchmark test suites for the first experiment. There are 10 and 12 functions in the two suites designed for long-term search, respectively. The reason why the CEC 2021 benchmark test suite, which is also designed for long-term search, is not used by us is given below. For the first time, in the suite, functions are parameterized by bias, rotation, and translation. In practice, each function has five possible transformations – basic, shift, bias & shift, shift & rotation, and bias & shift & rotation. However, the parameterization may be not a successful attempt and was given up in 2022.

Algorithm 1 Pseudo-Code of AMCDE

```
Input: NP_init, NP_min, MaxFES, G_n, p_bc1, p_bc2, p_w, and p_r;
Parameter: g_n, g, FES;
Output: S
Obtain the initial generation of the population P_0
g_n = 0, g = 0
Evaluate individuals in the population
FES = FES + NP_init and g = g + 1
Choose DE/current-to-φbest/1 with archive shown in Eq. (4) as the monopolizer
while FES < MaxFES do
    if g_n < G_n then
        Execute the chosen mutation strategy
        Execute the binomial or exponential crossover in Eq. (7) where p = p_bc1
        Evaluate offspring
        Execute Eq. (3) for selection
        if the best fitness is not improved then
            g_n = g_n + 1
        else
            g_n = 0
        end if
    else
        Execute Algorithm 2 with parameters, (P_g, p_bc2, p_w, p_r), for a generation
        if the best fitness is improved then
            g_n = 0 and G_n = G_n + 1
            Choose the mutation strategy obtaining the new best fitness as the new monopolizer
        end if
    end if
    FES = FES + NP and g = g + 1
    if FES ≥ 0.85 · MaxFES then
        Apply the local search method based on sequential quadratic programming
        Update FES
    end if
    Decrease NP according to the linear population size reduction
end while
Report the final solution S
```

Algorithm 2 Pseudo-Code of a generation with competition

```
Input: P_g, p_bc2, p_w, p_r and Cr;
Parameter: S_k,g (k = 1, 2, 3), and S_win,g
Output: P_g+1
Equal number of positions are randomly allocated to the three mutation strategies and grouped into S_k,g
c_k is computed according to Eq. (8)
win = 1
for x = 1 to 3 do
    if c_x > c_win then
        win = x
    end if
end for
if g is not the first generation in the current round of competition then
    for x = 1 to 3 do
        if x ≠ win then
            S_x,g = S_x,g \ S_win,g−1
        else
            S_x,g = S_x,g ∪ S_win,g−1
        end if
    end for
end if
Execute mutation strategy of each position
Execute the binomial or exponential crossover in Eq. (7) where p = p_bc2 and Cr is set based on Eq. (9)
Evaluate offspring
Execute the revised selection method where, at p_r, each of the worst p_w% target vectors are replaced by its worse trial vector
Compute c_k according to Eq. (8)
Find win ∈ {1, 2, 3} to make c_win = max(c_1, c_2, c_3)
```

Table 1
Setting of the CEC 2020 and CEC 2022 benchmark test suites.

D	MaxFES	
	2020	2022
5	5.00E+04	–
10	1.00E+06	1.00E+06
15	3.00E+06	–
20	1.00E+07	1.00E+07

Supported values of D and corresponding values of $MaxFES$ of the CEC 2020 and CEC 2022 benchmark test suites are shown in Table 1. It can be seen that the maximum of D is just 20. According to the behind thinking of long-term search – $MaxFES$ scaling exponentially with the increase of D, if $D > 20$, the large value of $MaxFES$ makes experiment infeasible at least on commonly used experiment platforms.

The second experiment is based on the CEC 2011 suite of real world optimization problems. We just select the problems whose $D \le 15$ for the above mentioned reason. In addition, $MaxFES$ is adjusted for long-term search. Table 2 shows all the problems selected by us. Table 3 gives $MaxFES$ designed by us, which scales exponentially with the increase of D, for the selected problems.

In the third experiment, the CEC 2017 benchmark test suite is employed although the suite is designed for traditional search. There are 30 functions in the suite. However, F2 is removed from the CEC 2017 competition because the function is unstable. Therefore, we also just use F1 and F3–F30 in this experiment. Supported values of D and corresponding values of $MaxFES$ of the suite are shown in Table 4. It can be seen that $MaxFES = D \cdot 1.00E{+}04$.

Beside the three experiments, we do experiment observation on our AMCDE based on the CEC 2020 and 2022 benchmark test suites. Four aspects – effectiveness of the combination of the three mutation strategies, usage of mutation strategy in execution, effectiveness of the proposed schemes, and difference in diversity brought by the revision in selection are observed for understanding AMCDE better.

In our experiments, IMODE, AGSK, j2020, NL-SHADE-RSP, EA4eig, NL-SHADE-LBC, and APGSK-IMODE-FL are selected as peers. The peers are introduced in Section 2. In the third experiment, each algorithm is executed 51 times for a case of comparison, respectively. In the other experiments, the number is 30.

4.1. Experimental setting

The original setting for the involved peers are still used in our experiment. Details are shown in Table 5. For our AMCDE, the original value of all the parameters inherited from IMODE is still used. Furthermore, the new parameters – G_n, p_{bc1}, p_{bc2}, p_r, and p_w – are set as below. G_n, the threshold to reflect whether the current mutation strategy is still efficient, is set 5 at first. After each round of competition, $G_n = G_n + 1$. The probability of binomial crossover in the monopoly state – p_{bc1} – is set to 0.4, which is the same value as p in IMODE. Meanwhile, the probability in the competition state – p_{bc2} – is left for testing. Moreover, the other two parameters, the probability of the worst individuals replaced by their worse offspring in selection – p_r, and the ratio of worst individuals – p_w, are also left for testing. In our experiment for parameter testing based on the CEC 2020 benchmark test suite when $D = 10$, the three parameters, p_{bc2}, p_r, and p_w, are set three values, respectively. For p_{bc2}, the three values are 0.40, 0.50, and 0.60. Meanwhile, for p_w, those are 0.10, 0.20, and 0.30. In addition, for p_r, those are 0.005, 0.010, and 0.050. Altogether, 27 combinations of the three parameters are implemented in AMCDE to find the best one. The experimental results are shown in Tables S1–S3 in our supplementary material. In Table 6, we give the outcome of the Friedman Test based on

Table 2
The problems selected from the CEC 2011 suite.

Problem number	Problem name	Dimensionality
P1	Parameter estimation for frequency modulated sound waves	6
P3	Bifunctional catalyst blend optimal control problem	1
P4	Optimal control of a non-linear stirred tank teacto	1
P8	Transmission network expansion planning problem	7
P10	Circular antenna array design problem	12
P11.3	Static economic load dispatch instance 1	6
P11.4	Static economic load dispatch instance 2	13
P11.5	Static economic load dispatch instance 3	15

Table 3
Value of $MaxFES$ for the problems selected from the CEC 2011 suite.

D	$MaxFES$
$D \leq 5$	5.00E+04
$5 < D \leq 10$	1.00E+06
$10 < D \leq 15$	3.00E+06

Table 4
Setting of the CEC 2017 benchmark test suite.

D	$MaxFES$
10	1.00E+05
30	3.00E+05
50	5.00E+05
100	1.00E+06

Tables S1–S3. It can be seen that $p_{bc2} = 0.40$, $p_w = 0.20$, and $p_r = 0.010$ lead to the best performance. Therefore, we adopt the combination for AMCDE.

4.2. Comparison based on the CEC 2020 and 2022 benchmark test suites

In this experiment, AMCDE and the seven peers are compared based on the CEC 2020 and the CEC 2022 benchmark test suites. Both 10 and 20 are set in D for functions. The results for the CEC 2020 suite when $D = 10$ are listed in Table S4, while those when $D = 20$ are listed in Table S5. Meanwhile, the results for the CEC 2022 suite are listed in Tables S6 and S7. Based on the tables, we show outcome of the Wilcoxon rank sum test at a 0.05 significance level in Table 7. Also, we execute the Friedman test and give outcome in Table 8.

For the CEC 2020 functions when $D = 10$, it can be seen that, from the view of the Wilconxon rank sum test as shown in Table 7, the proposed algorithm is comparable with j2020 and NL-SHADE-RSP, but defeats the other five algorithms. Meanwhile, from the view of the Friedman test as shown in Table 8, AMCDE ranks second and is defeated only by NL-SHADE-RSP in the comparison.

For the CEC 2020 functions when $D = 20$, it can be seen that, from the view of the Wilconxon rank sum test as shown in Table 7, the proposed algorithm is defeated by j2020 and NL-SHADE-RSP but defeats the other algorithms. Meanwhile, from the view of the Friedman test as shown in Table 8, AMCDE ranks second and is defeated only by NL-SHADE-RSP in the comparison.

For the CEC 2022 functions when $D = 10$, it can be seen that, from the view of the Wilconxon rank sum test as shown in Table 7, the proposed algorithm is defeated by EA4eig but defeats the other algorithms. Meanwhile, from the view of the Friedman test as shown in Table 8, AMCDE ranks second and is defeated only by EA4eig in the comparison.

For the CEC 2022 functions when $D = 20$, it can be seen that, from the view of the Wilconxon rank sum test as shown in Table 7, the proposed algorithm is defeated by EA4eig but defeats the other algorithms. Meanwhile, from the view of the Friedman test as shown in Table 8, AMCDE ranks first although ties EA4eig in the comparison.

Furthermore, by two tables – Tables 9 and 10, we further summarize the outcomes of the Wilcoxon rank sum test and those of the Friedman test, respectively. On the whole, according to Table 9, from the angle of the Wilconxon rank sum test, our algorithm ties EA4eig and outperforms the other peers. Meanwhile, according to Table 10, from the angle of Friedman test, our algorithm ties NL-SHADE-RSP and outperforms the other peers. Based on an overall consideration, AMCDE perform better than or at least comparable to the peers.

Based on the CEC 2022 suite, we study convergence manner of algorithm, When $D = 10$, for the four functions – F6, F8, F9, and F12, all the algorithms never obtain the optimal solution. When $D = 20$, all the algorithms never obtain the optimal solution for the eight functions – F2, F4, F6–F9, F11, and F12. Here, based on the average in the 30 executions of average fitness, we give convergence graph of all the algorithms for the four functions when $D = 10$ in Figure S1 and that for the eight functions when $D = 20$ in Figure S2, respectively.

According to Subfigures (a) and (b) in Figure S1 and Subfigures (b) and (c) in Fiugre S2, AMCDE may converge sharply at the final stage of execution. Furthermore, Subfigure (a) of Figure S1 and Subfigures (c) and (e) of Figure S2 significantly demonstrate that the average of fitness may go worse for generations during execution of our algorithm. The phenomenon can be also seen in convergence curve of a few peers, such as j2020, but cannot be observed in that of IMODE.

4.3. Comparison based on the selected CEC 2011 real world optimization problems

This experiment is based on the selected CEC 2011 real world optimization problems. Here, we just use IMODE, NL-SHADE-RSP, and EA4eig as peers. IMODE is selected because AMCDE is revised from the algorithm. The reason of selecting NL-SHADE-RSP and EA4eig is that the two algorithms are closer to our AMCDE in performance than the other peers according to the previous experiment based on the CEC 2014 and 2017 suites. Now that just four algorithms participate in this experiment, only the Wilcoxon rank sum test is used for comparison.

The results are listed in Table S8. In the table, the average solution of L-SHADE-cnEpSin-PWI [57] is listed as reference value. According to Table S8, for five problems – P1, P4, P10, P11.3, and P11.5 – out of the all eight ones, average solution of our AMCDE or the three peers is better than reference value. That is, most of the selected problem may not be completely solved by traditional search and deserve to be considered in long-term search. Based on Table S8, we show outcome of the Wilcoxon rank sum test at a 0.05 significance level in Table 11. According to Table 11, our algorithm is comparable with EA4eig, but defeats the other two algorithms in the comparison.

4.4. Comparison based on the CEC 2017 benchmark test suite

This experiment is based on the CEC 2017 benchmark test suite, which is designed for traditional search. All the four values of D – 10, 30, 50, 100 – are involved in experiment. Here, NL-SHADE-RSP and EA4eig are still used as peers. However, IMODE is replaced by APGSK-IMODE-FL. After all, among the peers, just APGSK-IMODE-FL does not come from the CEC competitions. Now that just four algorithms participate in this experiment, only the Wilcoxon rank sum test is used for comparison.

Table 5
Settings of the involved peers.

Algorithm	Parameters
IMODE	$NP_{max} = D^2 \cdot 6$, $NP_{min} = 4$, $A_{rate} = 2.60$, $H = D \cdot 20$, $p = 0.4$, and $FES_{LS} = MaxFES \cdot 0.85$
AGSK	$NP_{max} = D \cdot 20$, $NP_{min} = 12$, $p = 0.05$, and $c = 0.05$
j2020	$bNP = D \cdot 7$, $sNP = D$, $\tau_1 = \tau_2 = 0.1$, $ageLmt = \frac{MaxFES}{10}$, $eps = 1E-16$, and $myEqs = 25$
NL-SHADE-RSP	$NP_{max} = 30D$, $M_{f,r} = 0.2$, $M_{Cr,r} = 0.2$, and $n_A = 0.5$
EA4eig	$NP_{max} = 100$ and $NP_{min} = 10$
NL-SHADE-LBC	$NP_{max} = D \cdot 23$, $NP_{min} = 4$, $H = D \cdot 20$, $M_{F,r} = 0.5$, $M_{Cr,r} = 0.9$, $k = 1$, $n_A = 0$
APGSK-IMODE-FL	$NSP_a^{max} = NSP_i^{max} = D \cdot 30 \cdot 0.5$, $NSP_a^{min} = 12$, $NSP_i^{min} = 4$ $[T_a^{2CS}, T_i^{2CS}, T_a^{CS}, T_i^{CS}] = [70, 85, 37, 42]$ in the first 5 cycles, then, $[100, 100, 100, 100]$, $[70, 85, 35, 40]$, and $[70, 85, 37, 42]$ are used one by one

Table 6
The Friedman test outcome for the testing of the probability parameters.

p_{bc2}					0.40				
p_w		0.10			0.20			0.30	
p_r	0.005	0.010	0.050	0.005	0.010	0.050	0.005	0.010	0.050
Outcome	11.85	11.50	13.00	14.95	9.30	17.25	12.65	10.65	22.00
p_{bc2}					0.50				
p_w		0.10			0.20			0.30	
p_r	0.005	0.010	0.050	0.005	0.010	0.050	0.005	0.010	0.050
Outcome	16.05	13.60	13.20	14.15	9.90	19.20	10.15	15.15	21.15
p_{bc2}					0.60				
p_w		0.10			0.20			0.30	
p_r	0.005	0.010	0.050	0.005	0.010	0.050	0.005	0.010	0.050
Outcome	10.95	12.30	13.75	13.10	11.90	19.10	10.35	9.95	20.90

Table 7
Outcome of the Wilcoxon rank sum test of the results for the CEC 2020 and 2022 benchmark test suites. "+" or "−" denotes that, for a function, the result of the current algorithm is significantly better or statistical worse than the result of AMCDE in terms of the Wilcoxon rank sum test at a 0.05 significance Level, respectively. Meanwhile, "≈" represents that there is no significant difference. The four numbers separated by "/" are for the CEC 2020 suite when $D = 10$, that when $D = 20$, the CEC 2022 suite when $D = 10$, and that when $D = 20$, respectively.

Difference	IMODE	AGSK	j2020	NL-SHADE -RSP	EA4eig	NL-SHADE -LBC	APGSK _IMODE_FL
+	1/3/1/0	0/1/1/3	3/4/3/3	2/5/1/1	3/1/4/4	2/2/1/2	0/2/1/3
−	5/5/3/3	5/5/6/5	3/3/8/4	2/1/2/2	5/6/2/3	6/5/3/4	5/3/3/4
≈	4/2/8/9	5/4/5/4	4/3/1/5	6/4/9/9	2/3/6/5	2/3/8/6	5/5/8/5

Table 8
Outcome of the Friedman test of the results for the CEC 2020 and 2022 benchmark test suites. The four numbers separated by "/" are for the CEC 2020 suite when $D = 10$, that when $D = 20$, the CEC 2022 suite when $D = 10$, and that when $D = 20$, respectively.

Algorithm	Ranking	Algorithm	Ranking
IMODE	4.25/4.05/4.29/4.75	EA4eig	5.25/6.05/3.33/4.13
AGSK	4.95/5.30/5.17/4.88	NL-SHADE-LBC	5.90/5.75/5.21/4.50
j2020	3.95/3.95/5.96/4.54	APGSK-IMODE-FL	5.25/4.55/4.00/4.29
NL-SHADE-RSP	2.85/2.70/4.17/4.79	AMCDE	3.60/3.65/3.88/4.13

We use four tables – Tables S9–S12 to list the results. Based on the tables, we show outcome of the Wilcoxon rank sum test at a 0.05 significance level in Table 12. According to Table 12, our algorithm performs much worse than all of the peers for traditional search. The fact is also reflected by Figures S1 and S2 for showing convergence graph in long-term search. In the figures, at initial, AMCDE does not converge most quickly. For example, according to Subfigure (a), (c), (e) and (f) in Figure S2, AMCDE even converges most slowly at initial. As mentioned in Section 1, quick convergence at initial is not a good phenomenon for long-term search.

4.5. Experimental observation on our algorithm

Here, we observe our AMCDE at four aspects based on experiment. Based on the observation, AMCDE can be better understood.

4.5.1. Effectiveness of the combination of the three mutation strategies

We observe effectiveness of the combination of the three mutation strategies in AMCDE by experiment. Firstly, AMCDE with only DE/current-to-ϕbest/1 with archive, AMCDE with only DE/current-to-ϕbest/1, AMCDE with both DE/current-to-ϕbest/1 with archive and DE/current-to-ϕbest/1, AMCDE with DE/current-to-ϕbest/1 with archive, DE/current-to-ϕbest/1, and DE/rand/1, and the formal version of AMCDE are compared based on the CEC 2022 benchmark test suite. Results are given in Tables S13 and S14. Table S13 is for $D = 10$, while Table S14 is for $D = 20$. Based on the tables, we show outcome of the Friedman test in Table 13. Then, the two versions with three mutation strategies are further compared based on the selected CEC 2011 real world optimization problems. Results are given in Table S15. Based on Table S15, we show outcome of the Wilcoxon rank sum test at a 0.05 significance level in Table 14.

According to Table 13, it can be seen that the two versions with just one mutation strategy show worse performance than the formal version. Besides, AMCDE with both DE/current-to-ϕbest/1 with archive and DE/current-to-ϕbest/1 perform a little worse than the two versions of AMCDE with three mutation strategies. Furthermore, AMCDE with DE/current-to-ϕbest/1 with archive, DE/current-to-ϕbest/1, and DE/rand/1 performs a little better than the formal version. However, Table 14 demonstrates that, for the selected CEC 2011 real world optimization problems, the formal version performs much better than the other version with three mutation strategies in which DE/weighted-rand-to-ϕbest/1 is replaced by DE/rand/1. In brief, the combination of the three mutation strategies used in the formal version of AMCDE bears examination.

Table 9
The summary of the Wilconxon rank sum test of the results for the CEC 2020 and 2022 benchmark test suites.

Time (s)	IMODE	AGSK	j2020	NL-SHADE -RSP	EA4eig	NL-SHADE -LBC	APGSK _IMODE_FL
Defeating	4	4	2	2	2	4	4
Defeated by	0	0	1	1	2	0	0
Tying	0	0	1	1	0	0	0

Table 10
The summary of the Friedman test of the results for the CEC 2020 and 2022 benchmark test suites.

Time (s)	IMODE	AGSK	j2020	NL-SHADE -RSP	EA4eig	NL-SHADE -LBC	APGSK _IMODE_FL
Defeating	4	4	4	2	2	4	4
Defeated by	0	0	0	2	1	0	0
Tying	0	0	0	0	0	0	0

Table 11
Outcome of the Wilcoxon rank sum test for the selected CEC 2011 real world optimization problems. “+” or “−” denotes that, for a problem, the result of the current algorithm is significantly better or statistical worse than the result of AMCDE in terms of the Wilcoxon rank sum test at a 0.05 significance Level, respectively. Meanwhile, “≈” represents that there is no significant difference.

Difference	IMODE	NL-SHADE -RSP	EA4eig
+	1	0	2
−	4	4	2
≈	3	4	4

Table 12
Outcome of the Wilcoxon rank sum test for the CEC 2017 benchmark test suite. “+” or “−” denotes that, for a function, the result of the current algorithm is significantly better or statistical worse than the result of AMCDE in terms of the Wilcoxon rank sum test at a 0.05 significance Level, respectively. Meanwhile, “≈” represents that there is no significant difference. The four numbers separated by “/” are for $D = 10$, $D = 30$, $D = 50$, and $D = 100$, respectively.

Difference	NL-SHADE-RSP	EA4eig	APGSK-IMODE-FL
+	9/25/27/29	17/26/26/29	8/16/21/24
−	4/1/0/0	6/2/1/0	10/8/4/2
≈	16/3/2/0	6/1/2/0	11/5/4/3

4.5.2. Usage of mutation strategy in execution

Furthermore, based on the CEC 2022 suite when $D = 10$, we observe usage of the mutation strategies in execution of AMCDE with both DE/current-to-ϕbest/1 with archive and DE/current-to-ϕbest/1, AMCDE with DE/current-to-ϕbest/1 with archive, DE/current-to-ϕbest/1, and DE/rand/1, and the formal version of AMCDE. In Figures S3 and S4, based on the average of 30 executions of AMCDE with both DE/current-to-ϕbest/1 with archive and DE/current-to-ϕbest/1, we show the number of consumed function evaluations of each mutation strategy at 10 stages for each function, respectively. Figures S5 and S6 are for AMCDE with DE/current-to-ϕbest/1 with archive, DE/current-to-ϕbest/1, and DE/rand/1, while Figures S7 and S8 are for the formal version of AMCDE.

According to Figures S3 and S4, in the version of AMCDE with the two mutation strategies, DE/current-to-ϕbest/1 with archive dominates in execution for the majority of the functions, while DE/current-to-ϕbest/1 is executed more times for just one function. According to Figures S5 and S6, in the version of AMCDE in which DE/weighted-rand-to-ϕbest/1 is replaced by DE/rand/1, both DE/current-to-ϕbest/1 with archive and DE/current-to-ϕbest/1 dominates for six functions. DE/rand/1 is executed more frequently than one of the other to mutation strategies twice. According to Figures S7 and S8, in the formal version of AMCDE, DE/current-to-ϕbest/1 with archive dominates in execution for eight of the functions, while DE/current-to-ϕbest/1 performs not worse than DE/current-to-ϕbest/1 with archive for the rest four functions. Compared with the above two mutation strategies, for every function, DE/weighted-rand-to-ϕbest/1 is called much less times, especially in the late stages. That is, DE/current-to-ϕbest/1 with archive and DE/current-to-ϕbest/1 often become the monopolizer, while it is difficult for DE/weighted-rand-to-ϕbest/1 to become that.

In short, AMCDE with DE/current-to-ϕbest/1 with archive, DE/current-to-ϕbest/1, and DE/rand/1 shows the best balance in usage of mutation strategies, while the formal version of AMCDE demonstrates the worst balance.

4.5.3. Effectiveness of the proposed schemes

We observe effectiveness of the schemes proposed by us based on the CEC 2020 benchmark test suite when $D = 10$. In Table S16, results of IMODE, AMCDE without the alternation, AMCDE without the revision in selection, AMCDE without our Cr adaptation, and AMCDE are given. Based on Table S16, outcome of the Friedman test is given in Table 15. It can be seen from Table 15 that, if the alternation between monopoly and competition or the revision in selection is removed from AMCDE, solution goes even worse than that of IMODE. However, although the absence of our Cr adaptation has negative effect, cooperation of the alternation and the revision in selection is enough to make solution better than that of IMODE.

4.5.4. Difference in diversity brought by the revision in selection

Finally, we observe difference in diversity brought by the revision in selection based on the CEC 2022 suite when $D = 20$. In Figures S9 and S10, for each function, based on the average diversity in the 30 executions at 11 stages, we show diversity curve of both AMCDE without the revision in selection and AMCDE. According to the figures, for F1, F3–F4, F6–F8, and F12, the two algorithms show little difference in diversity curve. Meanwhile, for F5, AMCDE without the revision in selection keeps diversity better in the first half of execution but shows no advantage in the second one. However, for four functions – F2 and F9–11, AMCDE performs better in diversity maintaining in the second half of execution than AMCDE without the revision in selection.

4.6. Discussion

In fact, our AMCDE is revised from IMODE. Our schemes – the alternation between monopoly and competition, the revision in selection, and our Cr adaptation – lead to improvement in performance after parameter optimization. Therefore, AMCDE performs better than IMODE in long-term search. Moreover, AMCDE demonstrates superior or at the very least comparable performance compared to all the peers in long-term search. In fact, a DE algorithm suitable for long-term search may be not fit for traditional search at all. Our AMCDE is an example.

In such a type of ensemble, balance in usage of mutation strategies cannot reflect performance at all. AMCDE with DE/current-to-ϕbest/1 with archive, DE/current-to-ϕbest/1, and DE/rand/1 is much better in the balance than the formal version of AMCDE. However, performance of our formal version of AMCDE is at least comparable with that of

Table 13
The Friedman test outcome of comparison among versions of AMCDE with different combination of mutation strategy for the CEC 2022 benchmark test suite. The two numbers separated by "/" are for $D = 10$ and $D = 20$, respectively.

Algorithm	Ranking (average)
AMCDE with only DE/current-to-ϕbest/1 with archive	3.04/3.12 (3.08)
AMCDE with only DE/current-to-ϕbest/1	3.04/3.38 (3.21)
AMCDE with both DE/current-to-ϕbest/1 with archive and DE/current-to-ϕbest/1	3.04/2.92 (2.98)
AMCDE with DE/current-to-ϕbest/1 with archive, DE/current-to-ϕbest/1, and DE/rand/1	2.96/2.58 (2.77)
AMCDE	2.92/3.00 (2.96)

Table 14
The Friedman test outcome of comparison among the two versions of AMCDE with three mutation strategies for the selected CEC 2011 real world optimization problems.

Difference	AMCDE with DE/current-to-ϕbest/1 with archive, DE/current-to-ϕbest/1, and DE/rand/1
+	1
−	4
≈	3

Table 15
The Friedman test outcome of ablation experiment.

Algorithm	Ranking
IMODE	3.25
AMCDE without the alternation	3.70
AMCDE without the revision in selection	3.55
AMCDE without our Cr adaptation	2.35
AMCDE	2.15

the version. It can be analyzed that, in AMCDE, DE/current-to-ϕbest/1 with archive and DE/current-to-ϕbest/1 serves as the main mutation strategies, while DE/weighted-rand-to-ϕbest/1, which is very different from the other existing mutation strategies in action manner, can be regarded as a method for improving diversity. In fact, the individuals produced by DE/weighted-rand-to-ϕbest/1 do affect the distribution in solution space of the population. Compared with restart and other simple methods for improving diversity, the method may result in less degeneration.

Nevertheless, in AMCDE, DE/weighted-rand-to-ϕbest/1 used for improving diversity can hardly become the monopolizer and may be used just in the competition state. Therefore, the revision in selection is proposed by us as another method for diversification. Although the scheme may bring that the average fitness go worse for generations, does not affect the survival of individuals with better fitness at all. Therefore, the scheme is the important cooperator of our main scheme – the alternation between monopoly and competition. Only if the two schemes are employed at the same time, solution is improved. In addition, our Cr adaptation is helpful for further improving solution.

5. Conclusion

From 2020, long-term search is emphasized in the field of real parameter single objective optimization. A number of population-based metaheuristics suitable for long-term search have been proposed in literature. Among the algorithms, IMODE is ensemble of three mutation strategies.

Based on IMODE, we propose AMCDE. Our ensemble has two alternative states, monopoly and competition. The former is a steady state, while the latter is a transient one. In the monopoly state, one of the mutation strategies controls the whole population. If improvement in fitness becomes difficult, monopoly is replaced by competition. In the competition state, similar with IMODE, each of the mutation strategies controls a proportion of positions. However, we propose that the best performer among the three mutation strategies not only randomly seizes positions, but also continues to occupy the positions controlled by itself in the previous generation. Besides, for the competition state, we revise selection strategy and update Cr adaptation. After parameter optimization, we compare our AMCDE with the peers with good performance. The experimental results show that the proposed algorithm is very competitive for long-term search.

So far, many mutation strategies have been proposed in literature. Although we have proven that the combination of the three mutation strategies is not bad, it is reasonable that more combinations of mutation strategies are further tested. We should focus on this direction in the future.

CRediT authorship contribution statement

Chenxi Ye: Methodology, Writing – original draft. **Chengjun Li:** Conceptualization. **Yang Li:** Supervision. **Yufei Sun:** Writing – review & editing. **Wenxuan Yang:** Writing – review & editing. **Mingyuan Bai:** Writing – reviewing. **Xuanyu Zhu:** Validation. **Jinghan Hu:** Validation. **Tingzi Chi:** Software. **Hongbo Zhu:** Validation. **Luqi He:** Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We gratefully acknowledge the help of Professor Ponnuthurai Nagaratnam Suganthan, who offered the source code of most of the peers to us via GitHub.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.swevo.2023.101403.

References

[1] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[2] Z. Hu, Q. Su, X. Yang, Z. Xiong, Not guaranteeing convergence of differential evolution on a class of multimodal functions, Appl. Soft Comput. 41 (2016) 479–487.
[3] J. Gomez, Self adaptation of operator rates in evolutionary algorithms, in: Genetic and Evolutionary Computation Conference, Springer, 2004, pp. 1162–1173.
[4] A.E. Eiben, S.K. Smit, Evolutionary algorithm parameters and methods to tune them, in: Autonomous Search, Springer, 2011, pp. 15–36.
[5] B.-Y. Qu, P.N. Suganthan, J.-J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, IEEE Trans. Evol. Comput. 16 (5) (2012) 601–614.
[6] R. Storn, System design by constraint adaptation and differential evolution, IEEE Trans. Evol. Comput. 3 (1) (1999) 22–34.
[7] Y.-F. Ge, W.-J. Yu, Y. Lin, Y.-J. Gong, Z.-H. Zhan, W.-N. Chen, J. Zhang, Distributed differential evolution based on adaptive mergence and split for large-scale optimization, IEEE Trans. Cybern. 48 (7) (2017) 2166–2180.

[8] K. Sindhya, K. Miettinen, K. Deb, A hybrid framework for evolutionary multi-objective optimization, IEEE Trans. Evol. Comput. 17 (4) (2012) 495–511.
[9] J.J. Liang, B. Qu, P.N. Suganthan, A.G. Hernández-Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, Tech. Rep., Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, 2013.
[10] C.T. Yue, K.V. Price, J.J. Liang, M.Z. Ali, B.Y. Qu, N.H. Awad, P.P. Biswas, Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization, Tech. Rep., School of Electrical Engineering, Zhengzhou University, Zhengzhou, China and 836 Owl Circle, Vacaville, CA, U.S.A. and School of EEE, Nanyang Technological University, Singapore and School of Computer Information Systems, Jordan University of Science and Technology, Jordan and School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou, China and Department of Computer Science, Faculty of Engineering, University of Freiburg and Advanced Digital Science Center, Singapore, 2019.
[11] K.M. Sallam, S.M. Elsayed, R.K. Chakrabortty, M.J. Ryan, Improved multi-operator differential evolution algorithm for solving unconstrained problems, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8.
[12] A.W. Mohamed, A.A. Hadi, A.K. Mohamed, Differential evolution mutations: taxonomy, comparison and convergence analysis, IEEE Access 9 (2021) 68629–68662.
[13] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: Proc of CEC, IEEE, 2014, pp. 1658–1665.
[14] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.
[15] G. Wu, R. Mallipeddi, P.N. Suganthan, R. Wang, H. Chen, Differential evolution with multi-population based ensemble of mutation strategies, Inform. Sci. 329 (2016) 329–345.
[16] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, P. Suganthan, Ensemble of differential evolution variants, Inform. Sci. 423 (2018) 172–186.
[17] A.W. Mohamed, A.A. Hadi, P. Agrawal, K.M. Sallam, A.K. Mohamed, Gaining-sharing knowledge based algorithm with adaptive parameters hybrid with IMODE algorithm for solving CEC 2021 benchmark problems, in: 2021 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2021, pp. 841–848.
[18] X. Zhu, C. Ye, L. He, H. Zhu, T. Chi, J. Hu, Ensemble of differential evolution and gaining–sharing knowledge with exchange of individuals chosen based on fitness and lifetime, Soft Comput. (2023) 1–16.
[19] X. Wang, C. Li, J. Zhu, Q. Meng, L-SHADE-E: Ensemble of two differential evolution algorithms originating from L-SHADE, Inform. Sci. 552 (2021) 201–219.
[20] N.H. Awad, M.Z. Ali, P.N. Suganthan, J.J. Liang, B.Y. Qu, Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Tech. Rep., School of EEE, Nanyang Technological University, Singapore and School of Computer Information Systems, Jordan University of Science and School of Electrical Engineering, Zhengzhou University, Zhengzhou, China, 2016.
[21] A. Kumar, K.V. Price, A.W. Mohamed, A.A. Hadi, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2022 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization, Tech. Rep., Department of Electrical Engineering, Indian Institute of Technology (BHU), Varanasi, Varanasi, India and Vacaville, California, USA and Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza, Egypt and Department of Mathematics and Actuarial Science, School of Sciences & Engineering, The American University in Cairo, Giza, Egypt and Faculty of Computer Studies, Arab Open University, Riyadh, Saudi Arabia and School of EEE, Nanyang Technological University, Singapore, 2021.
[22] S. Das, P.N. Suganthan, Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems, Tech. Rep., Dept. of Electronics and Telecommunication Engg. , Jadavpur University, Kolkata, India and School of EEE, Nanyang Technological University, Singapore, 2010.
[23] A.W. Mohamed, P.N. Suganthan, Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation, Soft Comput. 22 (2018) 3215–3235.
[24] P. Civicioglu, E. Besdok, Bernstain-search differential evolution algorithm for numerical function optimization, Expert Syst. Appl. 138 (2019) 112831.
[25] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, M. Zhou, Chaotic local search-based differential evolution algorithms for optimization, IEEE Trans. Syst. Man Cybern.: Syst. 51 (6) (2019) 3954–3967.
[26] Z. Meng, J.-S. Pan, K.-K. Tseng, PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization, Knowl.-Based Syst. 168 (2019) 80–99.
[27] M. Leon, N. Xiong, Adaptive differential evolution with a new joint parameter adaptation method, Soft Comput. 24 (17) (2020) 12801–12819.
[28] Y. Li, S. Wang, B. Yang, An improved differential evolution algorithm with dual mutation strategies collaboration, Expert Syst. Appl. 153 (2020) 113451.
[29] G. Sun, B. Yang, Z. Yang, G. Xu, An adaptive differential evolution with combined strategy for global numerical optimization, Soft Comput. 24 (9) (2020) 6277–6296.
[30] G. Sun, G. Xu, N. Jiang, A simple differential evolution with time-varying strategy for continuous optimization, Soft Comput. 24 (4) (2020) 2727–2747.
[31] G. Yildizdan, Ö.K. Baykan, A novel modified bat algorithm hybridizing by differential evolution algorithm, Expert Syst. Appl. 141 (2020) 112949.
[32] J. Cheng, Z. Pan, H. Liang, Z. Gao, J. Gao, Differential evolution algorithm with fitness and diversity ranking-based mutation operator, Swarm Evol. Comput. 61 (2021) 100816.
[33] P. Civicioglu, E. Besdok, Bezier search differential evolution algorithm for numerical function optimization: A comparative study with CRMLSP, MVO, WA, SHADE and LSHADE, Expert Syst. Appl. 165 (2021) 113875.
[34] G. Sun, C. Li, L. Deng, An adaptive regeneration framework based on search space adjustment for differential evolution, Neural Comput. Appl. 33 (15) (2021) 9503–9519.
[35] Z. Tan, K. Li, Differential evolution with mixed mutation strategy based on deep reinforcement learning, Appl. Soft Comput. 111 (2021) 107678.
[36] Z. Tan, K. Li, Y. Wang, Differential evolution with adaptive mutation strategy based on fitness landscape analysis, Inform. Sci. 549 (2021) 142–163.
[37] X. Xia, L. Gui, Y. Zhang, X. Xu, F. Yu, H. Wu, B. Wei, G. He, Y. Li, K. Li, A fitness-based adaptive differential evolution algorithm, Inform. Sci. 549 (2021) 116–141.
[38] M. Wang, Y. Ma, P. Wang, Parameter and strategy adaptive differential evolution algorithm based on accompanying evolution, Inform. Sci. 607 (2022) 1136–1157.
[39] Q. Fan, X. Yan, Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies, IEEE Trans. Cybern. 46 (1) (2015) 219–232.
[40] A.W. Mohamed, A.A. Hadi, A.K. Mohamed, N.H. Awad, Evaluating the performance of adaptive gainingsharing knowledge based algorithm on cec 2020 benchmark problems, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8.
[41] A.W. Mohamed, A.A. Hadi, A.K. Mohamed, Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm, Int. J. Mach. Learn. Cybern. 11 (7) (2020) 1501–1529.
[42] J. Brest, M.S. Maučec, B. Bošković, Differential evolution algorithm for single objective bound-constrained optimization: Algorithm j2020, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8.
[43] J. Brest, M.S. Maučec, B. Bošković, The 100-digit challenge: Algorithm jde100, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 19–26.
[44] A.W. Mohamed, H.F. Abutarboush, A.A. Hadi, A.K. Mohamed, Gaining-sharing knowledge based algorithm with adaptive parameters for engineering optimization, IEEE Access 9 (2021) 65934–65946.
[45] V. Stanovov, S. Akhmedova, E. Semenkin, NL-SHADE-RSP algorithm with adaptive archive and selective pressure for CEC 2021 numerical optimization, in: 2021 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2021, pp. 809–816.
[46] V. Stanovov, S. Akhmedova, E. Semenkin, LSHADE algorithm with rank-based selective pressure strategy for solving CEC 2017 benchmark problems, in: 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2018, pp. 1–8.
[47] J. Brest, M.S. Maučec, B. Bošković, Self-adaptive differential evolution algorithm with population size reduction for single objective bound-constrained optimization: Algorithm j21, in: 2021 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2021, pp. 817–824.
[48] P. Bujok, P. Kolenovsky, Eigen crossover in cooperative model of evolutionary algorithms applied to CEC 2022 single objective numerical optimisation, in: 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2022, pp. 1–8.
[49] Y. Wang, H.-X. Li, T. Huang, L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Appl. Soft Comput. 18 (2014) 232–247.
[50] P. Bujok, J. Tvrdík, Enhanced individual-dependent differential evolution with population size adaptation, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2017, pp. 1358–1365.
[51] N. Hansen, A. Ostermeier, Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, in: 1996 IEEE Congress on Evolutionary Computation (CEC), IEEE, 1996, pp. 312–317.
[52] J. Brest, M.S. Maučec, B. Bošković, Single objective real-parameter optimization: Algorithm jSO, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2017, pp. 1311–1318.
[53] P. Bujok, A. Zamuda, Cooperative model of evolutionary algorithms applied to CEC 2019 single objective numerical optimization, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 366–371.
[54] V. Stanovov, S. Akhmedova, E. Semenkin, NL-SHADE-LBC algorithm with linear parameter adaptation bias change for CEC 2022 numerical optimization, in: 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2022, pp. 01–08.
[55] R. Biedrzycki, J. Arabas, E. Warchulski, A version of NL-SHADE-RSP algorithm with midpoint for CEC 2022 single objective bound constrained problems, in: 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2022, pp. 1–8.
[56] K.M. Sallam, S.M. Elsayed, R.A. Sarker, D.L. Essam, Landscape-based adaptive operator selection mechanism for differential evolution, Inform. Sci. 418 (2017) 383–404.
[57] A.P. Piotrowski, L-SHADE optimization algorithms with population-wide inertia, Inform. Sci. 468 (2018) 117–141.