Journal of Systems Engineering and Electronics
Vol. 33, No. 1, February 2022, pp.156 – 169

An ε-domination based two-archive 2 algorithm for many-objective optimization

WU Tianwei[1,2], AN Siguang[1,2,*], HAN Jianqiang[1,2], and SHENTU Nanying[1,2]

1. College of Mechanical and Electrical Engineering, China Jiliang University, Hangzhou 310018, China;
2. Key Laboratory of Intelligent Manufacturing Quality Big Data Tracing and Analysis of Zhejiang Province, China Jilang University, Hangzhou 310018, China

Abstract: The two-archive 2 algorithm (Two_Arch2) is a many-objective evolutionary algorithm for balancing the convergence, diversity, and complexity using diversity archive (DA) and convergence archive (CA). However, the individuals in DA are selected based on the traditional Pareto dominance which decreases the selection pressure in the high-dimensional problems. The traditional algorithm even cannot converge due to the weak selection pressure. Meanwhile, Two_Arch2 adopts DA as the output of the algorithm which is hard to maintain diversity and coverage of the final solutions synchronously and increase the complexity of the algorithm. To increase the evolutionary pressure of the algorithm and improve distribution and convergence of the final solutions, an ε-domination based Two_Arch2 algorithm (ε-Two_Arch2) for many-objective problems (MaOPs) is proposed in this paper. In ε-Two_Arch2, to decrease the computational complexity and speed up the convergence, a novel evolutionary framework with a fast update strategy is proposed; to increase the selection pressure, ε-domination is assigned to update the individuals in DA; to guarantee the uniform distribution of the solution, a boundary protection strategy based on $I_{\varepsilon+}$ indicator is designated as two steps selection strategies to update individuals in CA. To evaluate the performance of the proposed algorithm, a series of benchmark functions with different numbers of objectives is solved. The results demonstrate that the proposed method is competitive with the state-of-the-art multi-objective evolutionary algorithms and the efficiency of the algorithm is significantly improved compared with Two_Arch2.

Keywords: many-objective optimization, ε-domination, boundary protection strategy, two-archive algorithm.

DOI: 10.23919/JSEE.2022.000016

1. Introduction

Classic multi-objective evolutionary algorithms (MOEAs) [1], especially the Pareto dominance based ones, can efficiently handle multi-objective optimization problems (MOPs) [2], such as nondominated sorting genetic algorithm II (NSGA-II) [3] and strength Pareto evolutionary algorithm 2 (SPEA2) [4]. However, it has been found through experiments that these Pareto-based MOEAs show decreasing performance when they encounter many-objective optimization problems (MaOPs) [5,6], a category of MOPs with more than three objectives. The main reason for this phenomenon is that most individuals are in non-dominated relationship based on Pareto dominance in high-dimensional objective space. Therefore, the classic Pareto dominance based evolutionary algorithms cannot compare the better solutions which should survive into the next generation.

Recently, researchers have proposed a series of algorithms and techniques to deal with MaOPs, which can be roughly divided mainly into four categories. The first category is domination-based approaches that aim to reduce the number of non-dominated individuals or solutions by expanding the Pareto dominating area or proposing a new domination relationship. The ε-domination is used assigned to relax the dominance relationship which can enhance the selection pressure [7,8]. Wang et al. [9] studied the concept of coevolving a family of decision-maker preferences together with a population of candidate solutions. Li et al. [10] proposed a shift-based density estimation (SDE) strategy which aims to develop a general modification of density estimation. Dai et al. [11] proposed an improved α-dominance strategy which assigned α values based on an elliptic function to enhance the convergence pressure. Liu et al. [12] proposed an angle dominance which can provide sufficient selection pressure towards the Pareto front (PF) and can be exempt from the parameter tuning.

The second category is decomposition-based methods which decompose the MaOPs into a set of single-objec-

Manuscript received December 09, 2020.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China, Natural Science Foundation of Zhejiang Province (52077203; LY19E070003) and the Fundamental Research Funds for the Provincial Universities of Zhejiang (2021YW06).

tive subproblems by weight preference or reference points. Reference points are used to maintain the diversity of the solutions and minimal perpendicular distance from the current solutions to the reference points acts as a measure in selection strategy in NSGA-III [13]. Yang et al. [14] aimed to exploited the potential of the grid-based approach to strengthen the selection pressure toward the optimal direction while maintaining an extensive and uniform distribution among solutions. Asafuddoula et al. [15] proposed uniformly distributed reference points generated via systematic sampling and a maintenance that balance between convergence and diversity using two independent distance measures. Yuan et al. [16] exploited perpendicular distance from the individual to the weight vector in the objective space to maintain the desired diversity of solutions. Lyu et al. [17] proposed a bidirectional decomposition which contains two searching directions to provide a uniform distribution regardless of the problems' feature.

The third type of approach is the indicator-based evolutionary algorithm (IBEA) which uses the quality indicator that could compare individuals or solutions to guide the search towards a PF. Bader et al. [18,19] focused on the hypervolume indicator which is a metric to evaluate both convergence and diversity. However, the computation of the hypervolume indicator is very complex resulting in low efficiency. To enhance the efficiency of the indicator calculation, a novel indicator R2 [20] is used in MOEA [21]. Sun et al. [22] employed the inverted generational distance (IGD) in each generation to select the solutions with favorable convergence and diversity.

The last category is an objective reduction-based approach with the purpose of reducing objective space dimension by removing the redundant objectives. In order to determine redundant objectives, dimension reduction techniques in machine learning such as principal component analysis was also adopted [23]. Freitas et al. [24] presented a non-parametric harmony-based approach which can visualize conflict and combine the objective function based on the conflict. Yuan et al. [25] suggested viewing objective reduction as a multi-objective search problem which consists of three multi-objective formulations. A sampling method was used to collect points that can represent objectives by calculating objectives' improvements and affinity propagation was adopted to cluster the objectives [26].

With the development and improvement of these algorithms and techniques, the two-archive algorithm (Two_Arch) [27] firstly proposed a framework that contains two independent archives for convergence and diversity respectively. The convergence archive (CA) is updated by Pareto dominance so that the solutions could pursue to the PF. The diversity archive (DA) contains the non-dominated solutions without domination and consequently aims for diversity. Two_Arch removes crowded solutions from DA according to their distances to CA when the total size of two archives overflows. However, due to the decreasing evolutionary pressure of the Pareto dominance in high-dimensional objective space, the number of non-comparable individuals rises dramatically. Inspired by the idea of the two_arch, several improved Two_Arch algorithms were proposed. Li et al. [28] proposed an improved Two_Arch algorithm (IT-AA) which incorporates a ranking mechanism and a shifted density estimation technique in CA and DA, respectively. Dai [29] presented a multi-search strategy of selecting convergent solutions from offspring obtained from CA and DA as parents to enhance the convergence.

Wang et al. [30] proposed a two-archive 2 algorithm (Two_Arch2) which develops indicator-based and Pareto-based selection principles to the two archives. Compared with the original Two_Arch, Two_Arch2 can handle MaOPs with better performance on convergence and diversity of the final solutions in PF. However, Two_Arch2 tends to fall into a premature condition in the high-dimensional problems due to the weak selection pressure. Also, the distribution of the final solutions searched by Two_Arch2 is not always uniform, because Two_Arch2 adopts the population in DA as the final output which suffers with unacceptable quality of environmental selection. Furthermore, the complexity of the algorithm is increased because of the DA truncation strategy. To enhance the convergence ability and promote the convergence speed, an ε-domination based Two_Arch2 algorithm is proposed, namely, ε-Two_Arch2. The main contributions of this paper can be summarized as follows:

(i) A novel evolutionary framework with a fast update strategy is proposed to decrease the computational complexity.

(ii) ε-dominance is assigned to update the DA to enhance the quality of selection strategy and the affection of the vital parameter ε is analyzed.

(iii) A boundary protection strategy based on $I_{\varepsilon+}$ indicator is assigned to update the CA which can improve the diversity of CA.

The rest of this paper is organized as follows. The Two_Arch2 algorithm is firstly introduced in Section 2. After that, the proposed algorithm is introduced in details in Section 3. In Section 4, MOEAs are used to compare the proposed algorithm with others on a series of benchmark functions of different numbers of objectives. Experimental results and analysis are included in Section 4. Finally, the conclusions and the future work are provided in Section 5.

2. Two_Arch2 algorithm

The Two_Arch proposes a framework that divides the set of non-dominated solutions into two separate archives, which are used for convergence and diversity. Inspired by Two_Arch, different selection principles and a new L_P-norm-based diversity maintenance scheme are designed in Two_Arch2.

2.1 Basic flow

The basic evolutionary process of Two_Arch2 is given in Fig. 1. The initial non-dominated solution set is divided into CA and DA. Crossover is operated in both CA and DA; mutation is only implemented in CA during the reproduction. CA is updated by the quality indicator in IBEA which aims to guide the population to converge to the PF quickly. The purpose of DA is to add more diversity to the population in the high-dimensional objective space. Once DA overflows, the new L_P-norm-based diversity maintenance scheme is assigned to truncate DA. The sizes of CA and DA are fixed by their updating strategy respectively. With the less encouraged diversity of CA, Two_Arch2 uses DA which maintains better balance on diversity and convergence as the final output.

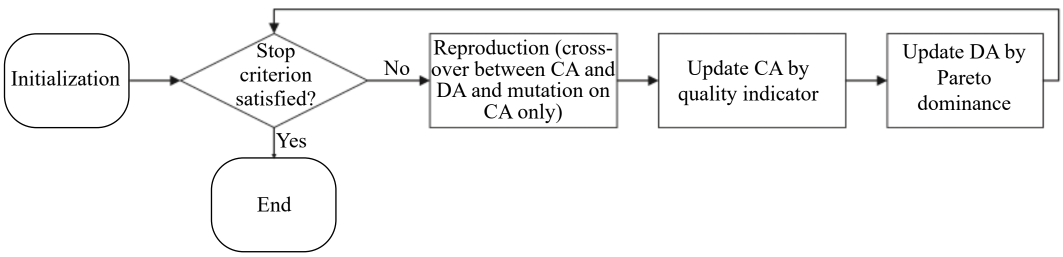


Fig. 1 Flow chart of Two_Arch2

2.2 Strengths and drawbacks

Two_Arch2 follows the idea of CA and DA in Two_Arch, and it assigns different selection strategies to update two archives. Two_Arch2 is a hybrid MOEA based on indicator and Pareto dominance. As a result, Two_Arch2 has improved the convergence ability by the quality indicator in CA and maintain approving diversity by the Pareto-based DA compared with Two_Arch. However, DA is updated by Pareto-dominance which is known as losing selection pressure in high-dimensional objective space. Meanwhile, CA cannot maintain in wide diversity because the environmental selection encourages convergence so that Two_Arch2 can only use DA as the final output. Although some individuals in CA may be selected into DA, DA cannot converge because there is no convergence maintenance within DA, which immediately impacts the performance of the algorithm. The last drawback is that Two_Arch2 needs to assign the third selection strategy in DA when DA overflows, which leads to more computational complexity. In order to solve these problems, an ε-dominance based Two_Arch2 algorithm is proposed in the following section.

3. Proposed algorithm

3.1 Framework

Balancing convergence and diversity is the main goal of an effective many-objective evolutionary algorithm (MaOEA). It is found that the ε-dominance relationship can encourage a good performance on diversity which is more effective than Pareto dominance relationship in high-dimensional objective space. Then, the boundary protection strategy based on the $I_{\varepsilon+}$ indicator is assigned in updating CA which could guide the population to converge to the PF and promote a good diversity simultaneously. Because of the more efficient environmental selection mechanism in coevolution of two archives, the simulated binary crossover (SBX) and polynomial mutation (PM) are both adopted to generate offspring population in reproduction for searching more individuals in the real PF. With the higher selection pressure of ε-dominance relationship, effective solutions can be selected into DA. As the iteration progresses, the size of population can expend in a fast speed which results in increasing of the algorithm complexity.

To solve this problem, a fast-update strategy including the Two_Arch algorithm framework is proposed and shown in Fig. 2. Different from the Two_Arch2, a novel iteration framework is developed to obtain a well distributed PF. Offspring generated by crossover and mutation operation in reproduction are firstly added to DA rather than CA, then the union set of another offspring and the updated DA selected based on the ε-dominance are added to CA for strengthening the diversity of CA population. In next iteration, the solutions deleted by the selection strategy in CA will be removed in DA synchronously (the details of selection in DA and CA are shown in the following subsections). Therefore, the size of DA is settled and fixed by fast-update strategy. Because the more distributional individuals of DA would be added to CA as parent, CA could maintain better diversity and coverage. Consequently, ε-Two_Arch2 uses CA as the final output. The pseudo-code of the framework is shown in Algorithm 1. With the proposed fast-update strategy, it is unnecessary to adopt additional environmental selection methods for population truncation when DA overflows, which leads to a better performance of algorithm efficiency.

Algorithm 1 Pseudo-code of the framework

Parameters: $Iter$-iteration, O-offspring, A_D-DA, A_C-CA. N-the size of A_D and A_C.
Initialize A_D and A_C
for $i = 1 : Iter$
  Generate offspring population O by making crossover between A_D and A_C but mutation on A_C only
    $A_D \leftarrow A_D \bigcup O$
  A_D = **Updating_DA**(A_D)
    $A_C \leftarrow A_C \bigcup A_D$
  (A_C, A_D) = **Updating_CA**(A_C, A_D, N)
end
return A_C

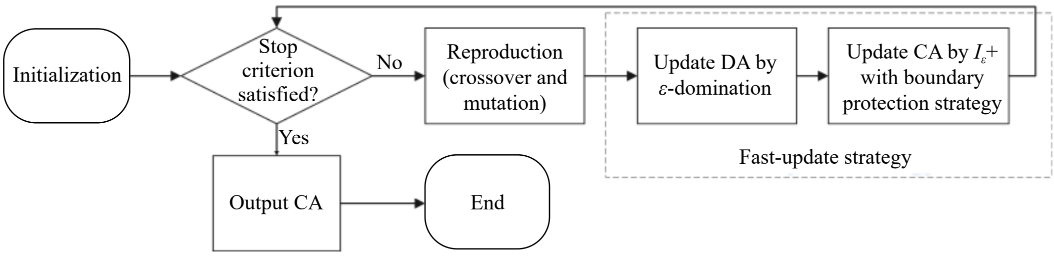


Fig. 2 Flow chart of ε-Two_Arch2

3.2 Diversity archive

Since the Pareto-dominance loses the quality of its environmental selection on high-dimensional objective space, the ε-dominance in ε-MOEA [8] is assigned as the selection strategy for DA in this paper which could provide sufficient selection pressure towards the PF. For minimization problems, let $f, g \in \mathbf{R}^{+m}$, $\forall i \in \{1, \cdots, m\}$, f is said to ε-dominate g ($\varepsilon_i > 0$, denoted by $f \succ_\varepsilon g$) if and only if

$$(1-\varepsilon_i) \cdot f_i \leqslant g_i \tag{1}$$

where m is the number of objectives, $\mathbf{R}^{+m}$ is the normalized objectives space.

In the step of updating DA in ε-Two_Arch2, the offspring generated by crossover in reproduction are firstly added to DA. Then, the proposed algorithm deletes the extra solutions in DA according to the ε-dominance. With the enhancing quality of the environmental selection, more solutions would be selected in DA, and the size of DA would not expend by adopting the fast-update strategy, which leads to lower complexity. In each iteration, DA is added to CA as parent, and the solutions that are deleted by selection strategy in CA would also be deleted in next iteration. The pseudo-code of the updating strategy in DA is shown in Algorithm 2.

Algorithm 2 Updating_DA(A_D)
Input: A_D-DA
Output: UA_D-updated A_D

Set an empty temporary population UA_D
for $i = 1 : |A_D|$
  Randomly choose x and y from A_D
  if $x \succ_\varepsilon y$
    Add x to UA_D
  else $y \succ_\varepsilon x$
    Add y to UA_D
  end
end
return UA_D

3.3 Convergence archive

To achieve a better performance of the algorithm, ε-Two_Arch2 needs to assign an archive with more balance on convergence and diversity as final output. Inspired by the indicator-based MaOEA with boundary protection MaOEA-IBP [31], a boundary protection strategy based on $I_{\varepsilon+}$ is assigned as the selection strategy in CA in this paper, which is expected to ensure convergence of the population to the PF and promote a better distribution synchronously.

The pseudo-code of the updating strategy in CA is shown in Algorithm 3. To be specific, after the updating in DA which selects individuals by ε-dominance, the union of the offspring generated during reproduction and the updated DA is added to CA. The individuals in CA are sorted by nondominated sorting, and the individuals with the top rank level are selected to a temporary population T. Then, the quality indicator $I_{\varepsilon+}$ is assigned as the first selection criterion, the $I_{\varepsilon+}$ indicator value of each two solutions are calculated as follows:

$$I_{\varepsilon+}(x,y) = \min_{\varepsilon}(f_i(x) - \varepsilon \leqslant f_i(y)), i \in \{1,\cdots,m\}. \quad (2)$$

By calculating $I_{\varepsilon+}$ indicator value of each two individuals in T based on (2), the pair of individuals x and y which have the minimum $I_{\varepsilon+}$ value are recognized. If $I_{\varepsilon+}(x,y) < 0$ which means $x \succ y$, or $I_{\varepsilon+}(y,x) < 0$ which means $x \prec y$, then the dominated individual y or x is deleted from the population T. Otherwise, the second selection strategy should be assigned to select individuals in CA.

Algorithm 3 Updating_CA(A_C,A_D,N)
Input: A_C-CA, A_D-DA, N-size of A_C
Output: UA_C-updated A_C, UA_D-updated A_D

T = ParetoNondominationRank(A_C)
Calculate the Euclidean distance $d(x,C)$ between each individual in T and the curve or surface C
while $|A_D| > N$
Calculate the $I_{\varepsilon+}$ value of each two individuals by (2) and then find a pair of the individuals with the minimum $I_{\varepsilon+}$ value in T
    if $I_{\varepsilon+}(x,y) < 0$
        Delete y from T
    else if $I_{\varepsilon+}(y,x) < 0$
        Delete x from T
    else
        if $d(x,C) > d(y,C)$
            Delete y from T
        else if $d(x,C) < d(y,C)$
        Delete x from T
        else
        Randomly delete x or y
        end
    end
    $UA_C \leftarrow T$
    Delete x or y if it is also in A_D
    $UA_D \leftarrow A_D$
end
return UA_C, UA_D

The boundary protection strategy is the second selection criterion in environmental selection of MaOEA-IBP which is inspired by the knee point-driven evolutionary algorithm (KnEA) [32]. The Euclidean distance $d(x,C)$ between the individuals and a curve or surface C is assigned to measure the fitness of the individuals. The curve or surface C is characterized as follows:

$$\left(\sum_{i=1}^{m}(f_i(x))^P\right)^{\frac{1}{P}} = 1 \quad (3)$$

where $f_i(x)(i \in \{1,2,\cdots,m\})$ is a vector on the curve or surface C, and P is the curvature of C. Then, the L_P-norm distance [33] is adopted to approximate $d(x,C)$. The L_P-norm distance between the points on C and the origin is 1, and $d(x,C)$ is approximated as follows:

$$d(x,C) \approx \left(\sum_{i=1}^{m}(f_i(x))^P\right)^{\frac{1}{P}} - 1. \quad (4)$$

The smaller value of $d(x,C)$ means that it is closer between the individual and the curve or surface C. And $d(x,C)$ is assigned a negative value for individuals below C. In the updating process of CA, the individuals with larger $d(x,C)$ value are removed from temporary population T if the quality indicator $I_{\varepsilon+}$ cannot judge which is superior. If both selection strategies cannot distinguish the pair of individuals, an individual is randomly deleted from T. When the size of population T reaches N, the updating process stops.

4. Experimental design and results

4.1 Test problems and performance indicator

In order to evaluate the performance of ε-Two_Arch2 on MaOPs, DTLZ (proposed by Deb, Thiele, Laumanns, and Zitzler) benchmark problems [34] and walking fish group (WFG) benchmark problems [35] are assigned in this paper, which also are the two most widely used tunable benchmark problems. More specifically, the number of objectives is set in $\{4,5,\cdots,10\}$ which satisfies the definition of MaOPs. Each objective function of m-objective DTLZ test problem has $n = k+m-1$ decision variables, and k is set to be 5 for DTLZ1 and DTLZ3, 10 for DTLZ2 and DTLZ4. Furthermore, each objective function of m-objective WFG test problem has $n = k+m-1$ decision variables, and k is set to 20 based on the suggestion from [36].

The IGD [36] and hypervolume (HV) [37] both are widely used as the performance indicators. Because the computational complexity of the HV indicator is too high for the MaOPs, the IGD is assigned as the performance indicators in this paper. IGD evaluates the convergence and diversity of the algorithm by calculating the minimum distance between each individual in the real Pareto optimal solution set and the optimal solution set obtained by the algorithm. The smaller IGD values indicate a better convergence and diversity performance of the algorithm.

4.2 Compared algorithms and parameter settings

Two_Arch2 and other four state-of-the-art MaOEAs are selected for comparison to assess the performance of the proposed ε-Two_Arch2. The four state-of-the-art MaOEAs respectively are NSGA-III, goals-considered preference-inspired coevolutionary algorithm (PICEA-g) [9], shift-based density estimation incorporated in SPEA2 (SPEA2+SDE) [10], and grid-based evolutionary algorithm (GrEA) [14]. Two_Arch2 is selected as a reference to show the improvement of ε-Two_Arch2 on MaOPs. NSGA-III is the newly-proposed NSGA algorithm with reference point for MaOPs. PICEA-g is a specific algorithm that coevolves a family of preferences with candidate solutions. SPEA2+SDE assigns the shift-based density estimation strategy in the framework of SPEA2. GrEA exploits the potential of a grid to handle many-objective optimization problems.

Because of the particularity of the reference point in NSGA-III, the population sizes of other compared algorithms are set the same as that of NSGA-III. The termination criterion of all algorithm is appointed as the maximum generation (GenMax) which is set for 300. Each algorithm runs 20 times independently on each test problem. The SBX and PM [38] operators are adopted to generate offspring population, and its parameter settings are shown in Table 1. According to [8] and [32], the settings of ε in (1) for ε-dominance in updating DA and P in (4) for the boundary protection strategy in updating CA are also shown in Table 1.

Table 1 Parameter settings of algorithm units

Parameter	Value
Crossover probability	1.0
Mutation probability	$1/n$
ε *in* (1)	0.06
P in (4)	0.3

4.3 Experimental results and analysis

The experimental results are shown in this section. The average values of IGD results which are generated by competing algorithms over DTLZ1-DTLZ4 and WFG1-WFG9 test problems are listed in following tables. In addition, the numbers with bold face imply the best average values over the comparative test problems with a given objective number. And the numbers in brackets following the average values are the ranking numbers of the performance of competing algorithms.

Table 2 shows the average values of IGD results of compared algorithms on DTLZ1 test problems. NSGA-III and SPEA2+SDE show the best performance on DTLZ1. PICEA-g and GrEA have the worst IGD values with more than six objectives. However, GrEA has better IGD values than ε-Two_Arch2 with four and five objectives. Compared with Two_Arch2, ε-Two_Arch2 has better IGD values on DTLZ1.

Table 2 Average values of IGD results of the compared algorithms on DTLZ1

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
DTLZ1	4	0.261 5(5)	0.273 4(6)	**0.040 1(1)**	0.260 8(4)	0.133 6(3)	0.041 8(2)
	5	0.251 8(4)	0.270 0(5)	0.062 0(2)	0.299 1(6)	0.198 5(3)	**0.061 9(1)**
	6	0.242 0(3)	0.266 2(5)	0.117 6(2)	0.329 1(6)	0.252 1 (4)	**0.079 3(1)**
	7	0.257 2(3)	0.263 6(4)	0.107 8(2)	0.385 7 (6)	0.327 3(5)	**0.098 1(1)**
	8	0.251 7(3)	0.262 1(4)	0.130 4(2)	0.397 9(6)	0.389 1(5)	**0.106 8(1)**
	9	0.260 5(3)	0.267 7(4)	0.247 6(2)	0.446 7(6)	0.401 2(5)	**0.136 7(1)**
	10	0.273 2(3)	0.273 6(4)	**0.109 8(1)**	0.369 7(5)	0.393 7(6)	0.120 4(2)

Table 3 shows the average values of IGD results of compared algorithms on DTLZ2 test problems. PICEA-g has the worst performance on DTLZ2 with four to eight objectives, and GrEA has the worst IGD values on DTLZ2 with more than eight objectives. ε-Two_Arch2 shows the best performance on all the DTLZ2. NSGA-III has a better performance than Two_Arch2 on all the DTLZ2 which can prove the improvement of ε-Two_Arch2.

Table 3 Average values of IGD results of the compared algorithms on DTLZ2

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
DTLZ2	4	**0.105 5(1)**	0.137 7(4)	0.116 1(2)	0.188 0(6)	0.124 7 (3)	0.141 8(5)
	5	**0.139 4(1)**	0.218 3(5)	0.189 7(2)	0.258 3(6)	0.190 2(3)	0.208 9(4)
	6	**0.173 6(1)**	0.294 0(5)	0.257 2(3)	0.321 8(6)	0.254 9(2)	0.266 5(4)
	7	**0.202 9(1)**	0.362 7(5)	0.331 8(3)	0.410 1(6)	0.326 2(2)	0.333 1(4)
	8	**0.227 3(1)**	0.424 8(5)	0.355 8(3)	0.460 9(6)	0.401 5(4)	0.345 5(2)
	9	**0.233 1(1)**	0.480 4(3)	0.492 4(4)	0.716 9(5)	0.812 1(6)	0.414 0(2)
	10	**0.280 8(1)**	0.519 2(5)	0.380 6(3)	0.451 3(4)	0.978 1(6)	0.356 1(2)

Table 4 shows the average values of IGD results of compared algorithms on DTLZ3 test problems. GrEA and NSGA-III have the worst performance on DTLZ3, although NSGA-III is in rank 2 with less than six objectives. SPEA2+SDE is in rank 2 with more than five objectives. PICEA-g only has a better performance than GrEA on DTLZ3. Two_Arch2 has a better performance than NSGA-III with more than five objectives, and ε-Two_Arch2 shows the best performance on all the DTLZ3.

Table 4 Average values of IGD results of the compared algorithms on DTLZ3

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
DTLZ3	4	**0.0958(1)**	0.1335(3)	0.1170(2)	0.5167(5)	0.8396(6)	0.1409(4)
	5	**0.1384(1)**	0.2091(3)	0.1927(2)	0.5899(5)	0.8499(6)	0.2071(4)
	6	**0.1815(1)**	0.2827(3)	1.3700(6)	0.7024(4)	1.1058(5)	0.2683(2)
	7	**0.2069(1)**	0.3481(3)	0.7919(4)	0.8489(5)	1.9463(6)	0.3425(2)
	8	**0.2200(1)**	0.4129(3)	1.8506(5)	0.8660(4)	3.7406(6)	0.3454(2)
	9	**0.2409(1)**	0.4692(3)	8.7805(6)	0.9951(4)	2.5892(5)	0.4314(2)
	10	**0.2585(1)**	0.5175(3)	1.6335(6)	0.9302(5)	1.5928(5)	0.3538(2)

Table 5 shows the average values of IGD results of compared algorithms on DTLZ4 test problems. PICEA-g has the worst performance on DTLZ4 with less than 10 objectives and GrEA shows the worst IGD results on DTLZ4 with 10 objectives. SPEA2+SDE is in rank 2 on DTLZ4 with more than seven objectives. NSGA-III has better IGD results than Two_Arch2 on DTLZ4 with four and five objectives. Two_Arch2 shows a better performance than NSGA-III on DTLZ4 with more than five objectives. For DTLZ4 with more than seven objectives, ε-Two_Arch2 is the best-performance algorithm on all the DTLZ4.

Table 5 Average values of IGD results of the compared algorithms on DTLZ4

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
DTLZ4	4	**0.1015(1)**	0.1388(2)	0.1498(4)	0.2107(6)	0.1932(5)	0.1438(3)
	5	**0.1309(1)**	0.2222(4)	0.1953(3)	0.3015(6)	0.1926(2)	0.2282(5)
	6	**0.1621(1)**	0.3016(4)	0.3032(5)	0.3402(6)	0.2570(2)	0.2811(3)
	7	**0.1873(1)**	0.3727(5)	0.3647(4)	0.4894(6)	0.3071(2)	0.3408(3)
	8	**0.1890(1)**	0.4360(6)	0.3944(4)	0.4261(5)	0.3855(3)	0.3499(2)
	9	**0.2596(1)**	0.4997(5)	0.4695(3)	0.5554(6)	0.4862(4)	0.4424(2)
	10	**0.3079(1)**	0.5349(5)	0.4077(3)	0.4107(4)	0.5389(6)	0.3714(2)

Table 6 shows the average values of IGD results of compared algorithms on WFG1 test problems. WFG1 is the problem that has the most transformation functions among the WFG benchmark problems, so that performing a good diversity is a challenge for the MaOEAs. NSGA-III shows the best performance on all the objectives of WFG1 test problems. Unexpectedly, SPEA2+SDE is in rank 2 on WFG1 with four to seven objectives and nine to ten objectives, PICEA-g is in rank 2 on WFG1 with eight objectives. ε-Two_Arch2 has the worst performance on all of the WFG1 test problems, and Two_Arch2 has better IGD values than ε-Two_Arch2 on all of the WFG1 test problems.

Table 6 Average values of IGD results of the compared algorithms on WFG1

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
WFG1	4	1.9249(6)	1.3681(5)	**0.2915(1)**	0.3620(3)	0.4261(4)	0.3124(2)
	5	2.1207(6)	1.6050(5)	**0.4355(1)**	0.5168(3)	0.5635(4)	0.4803(2)
	6	2.3325(6)	1.7400(5)	**0.6402(1)**	0.7368(3)	0.8636(4)	0.6932(2)
	7	2.5507(6)	2.0164(5)	**0.8739(1)**	0.9932(3)	1.2685(4)	0.9732(2)
	8	2.8387(6)	2.1617(5)	**1.0830(1)**	1.3254(2)	1.9771(4)	1.4427(3)
	9	3.2123(6)	2.4915(5)	**1.2732(1)**	2.1385(4)	2.1214(3)	1.7623(2)
	10	3.4066(6)	2.7621(4)	**1.3463(1)**	2.8271(5)	2.1551(3)	1.8376(2)

Table 7 shows the average values of IGD results of compared algorithms on WFG2 test problems. In WFG benchmark problems, WFG2 is the only disconnected problem. NSGA-III shows the best performance on WFG2 with four objectives. ε-Two_Arch2 has the best IGD values on WFG2 with six, seven and ten objectives. GrEA shows the best IGD results on WFG2 with five objectives and Two_Arch2 has the best performance on WFG2 with eight and nine objectives, followed by ε-Two_Arch2, but Two_Arch2 shows the worst IGD results on WFG2 with four and five objectives. PICEA-g shows the worst IGD values on WFG2 with six and eight to ten objectives, SPEA2+SDE has the worst performance on WFG2 with seven objectives.

Table 7 Average values of IGD results of the compared algorithms on WFG2

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
	4	0.603 5(4)	0.759 8(6)	**0.354 2(1)**	0.679 9(5)	0.464 9(3)	0.432 5(2)
	5	0.671 8(3)	1.061 0(6)	0.626 9(2)	0.837 9(5)	**0.600 3(1)**	0.690 8(4)
	6	**0.833 2(1)**	1.069 2(4)	0.973 8(3)	1.108 5(6)	0.896 7(2)	1.085 1(5)
WFG2	7	**1.024 5(1)**	1.101 6(2)	1.132 6(3)	1.673 2(5)	1.334 6(4)	1.678 4(6)
	8	1.191 8(2)	**1.086 1(1)**	1.314 8(3)	2.129 1(6)	1.450 4(4)	1.879 2(5)
	9	1.424 4(2)	**1.374 6(1)**	1.424 5(3)	2.689 5(6)	1.664 1(4)	2.349 8(5)
	10	**1.549 6(1)**	1.591 3(3)	1.575 8(2)	2.827 1(6)	1.879 6(4)	2.472 3(5)

Table 8 shows the average values of IGD results of compared algorithms on WFG3 test problems. WFG3 is the connected version of WFG2. GrEA shows the best and outstanding performance than other compared algorithms on WFG3 with more than six objectives, and PICEA-g has the best performance on WFG3 with fewer than seven objectives. SPEA2+SDE shows the worst performance on WFG3 with more than four objectives. ε-Two_Arch2 shows the worst performance on WFG3 with four objectives, but it has better IGD values than Two_Arch2 on WFG3 with more than six objectives and ε-Two_Arch2 is in rank 2 on WFG3 with seven objectives.

Table 8 Average values of IGD results of the compared algorithms on WFG3

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
	4	0.401 5(6)	0.230 2(3)	0.259 5(4)	**0.105 1(1)**	0.188 6(2)	0.391 7(5)
	5	0.464 1(4)	0.436 8(3)	0.482 6(5)	**0.161 9(1)**	0.175 0(2)	0.747 6(6)
	6	0.693 9(5)	0.668 0(4)	0.431 6(3)	**0.136 7(1)**	0.157 6(2)	0.957 3(6)
WFG3	7	0.702 1(2)	0.855 8(4)	0.887 4(5)	0.830 1(3)	**0.238 5(1)**	1.463 2(6)
	8	1.060 5(3)	1.127 4(4)	0.963 7(2)	1.354 4(5)	**0.280 2(1)**	1.738 3(6)
	9	1.183 7(3)	1.341 5(4)	0.967 4(2)	1.476 2(5)	**0.306 7(1)**	1.932 5(6)
	10	1.216 2(3)	1.481 1(4)	0.993 3(2)	1.919 6(5)	**0.464 9(1)**	1.969 2(6)

Table 9 shows the average values of IGD results of compared algorithms on WFG4 test problems. WFG4 is a multimodal problem so that aggregation functions cannot jump out of those local optimal sets, which results in the worst performance of PICEA-g on all the WFG4 test problems. SPEA2+SDE and Two_Arch2 also show worse IGD values on all the WFG4 test problems. NSGA-III has the best performance on WFG4 with ten objectives, however, ε-Two_Arch2 is the best-performing algorithm on WFG4 with four to nine objectives and performs better than Two_Arch2 and GrEA on all the WFG4 test problems.

Table 9 Average values of IGD results of the compared algorithms on WFG4

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
	4	**0.539 6(1)**	0.653 9(4)	0.604 7(3)	0.777 6(6)	0.590 4(2)	0.705 0(5)
	5	**1.065 9(1)**	1.238 1(4)	1.124 3(3)	1.341 4(6)	1.067 5(2)	1.241 4(5)
	6	**1.487 8(1)**	1.904 7(5)	1.762 6(3)	3.469 4(6)	1.622 7(2)	1.841 1(4)
WFG4	7	**2.113 2(1)**	2.688 1(5)	2.473 5(2)	3.984 2(6)	2.532 1(3)	2.674 2(4)
	8	**2.049 4(1)**	3.470 3(5)	2.987 9(3)	4.370 5(6)	2.853 4(5)	2.989 1(4)
	9	**2.571 7(1)**	4.349 9(5)	2.873 4(2)	5.031 2(6)	3.159 5(3)	3.684 9(4)
	10	3.069 4(2)	5.196 0(5)	**2.798 9(1)**	5.250 1(6)	3.825 1(3)	3.838 7(4)

Table 10 shows the average values of IGD results of compared algorithms on WFG5 test problems. WFG5 is a deceptive problem. PICEA-g shows the worst IGD values on WFG5 with fewer than eight objectives and Two_Arch2 has the worst performance on WFG5 with more than seven objectives. SPEA2+SDE also shows a worse performance on all the WFG5 test problems which rank in 3, 4, and 5. ε-Two_Arch2 shows the best IGD values on all the WFG5 test problems with significant advantage, followed by NSGA-III and GrEA which rank in 2 and 3.

Table 10 Average values of IGD results of the compared algorithms on WFG5

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
WFG5	4	**0.2172(1)**	0.6619(4)	0.5955(3)	0.8014(6)	0.5940(2)	0.7157(5)
	5	**0.7704(1)**	1.2246(4)	1.1179(3)	1.3701(6)	1.0834(2)	1.2542(5)
	6	**0.8705(1)**	1.8981(5)	1.7167(3)	2.0773(6)	1.6535(2)	1.8369(4)
	7	**1.3184(1)**	2.6287(5)	2.1356(2)	2.8904(6)	2.4573(3)	2.4874(4)
	8	**1.3281(1)**	3.4438(6)	2.9538(4)	3.3167(5)	2.8551(2)	2.9097(3)
	9	**1.6306(1)**	4.2972(6)	2.8381(2)	3.5982(5)	3.3904(3)	3.4928(4)
	10	**2.1201(1)**	5.0838(6)	2.7024(2)	3.8423(4)	3.8612(5)	3.7335(3)

Table 11 shows the average values of IGD results of compared algorithms on WFG6 test problems. WFG6 is a nonseparable-reduced problem. PICEA-g and Two_Arch2 both perform badly on this problem and show worse IGD values on all the WFG6 test problems. GrEA is in rank 2 on WFG6 with fewer than nine objectives and NSGA-III is in rank 2 on WFG6 with more than eight objectives. SPEA2+SDE is in rank 3 on WFG6 with more than six objectives. ε-Two_Arch2 has the best performance on all the WFG6 test problems with considerable preponderance compared with other algorithms.

Table 11 Average values of IGD results of the compared algorithms on WFG6

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
WFG6	4	**0.3768(1)**	0.6751(4)	0.6175(3)	0.8370(6)	0.6159(2)	0.7534(5)
	5	**0.7905(1)**	1.2527(4)	1.1377(3)	1.4275(6)	1.1080(2)	1.3163(5)
	6	**0.8715(1)**	1.9374(4)	1.7318(3)	2.1173(6)	1.6431(2)	1.9203(5)
	7	**1.3695(1)**	2.6722(4)	2.6723(5)	2.8952(6)	2.3649(2)	2.4875(3)
	8	**1.5943(1)**	3.4910(5)	3.0176(4)	3.5134(6)	2.8809(2)	2.9825(3)
	9	**2.0326(1)**	4.3084(6)	2.8931(2)	3.7686(5)	3.4784(4)	3.2718(3)
	10	**2.6002(1)**	5.0426(6)	2.7913(2)	3.9433(5)	3.8699(4)	3.6443(3)

Table 12 shows the average values of IGD results of compared algorithms on WFG7 test problems. WFG7 is both uni-modal and separable. NSGA-III shows the best IGD values on WFG7 with four, nine, and ten objectives, GrEA has the best performance on WFG7 with five, six, and eight objectives, PICEA-g shows the best IGD results on WFG7 with seven objectives. SPEA2+SDE also has well performance that is in rank 3 on WFG7 with fewer than seven objectives and is in rank 2 on WFG7 with more than six objectives. ε-Two_Arch2 shows the worst IGD values on WFG7 with less than seven objectives, and it has a better performance than Two_Arch2 on WFG7 with more than six objectives.

Table 12 Average values of IGD results of the compared algorithms on WFG7

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
WFG7	4	1.2137(6)	0.9708(5)	**0.5841(1)**	0.8554(4)	0.6048(2)	0.7127(3)
	5	2.0960(6)	1.6853(5)	1.1099(2)	1.4142(4)	**1.0844(1)**	1.2511(3)
	6	2.4758(6)	2.0541(4)	1.6978(2)	2.2881(5)	**1.6093(1)**	1.8145(3)
	7	3.4406(5)	3.6137(6)	2.4391(3)	**2.3874(1)**	2.4639(4)	2.3982(2)
	8	3.0700(4)	4.3854(6)	2.9771(3)	3.3989(5)	**2.8392(1)**	2.8838(2)
	9	3.8203(5)	4.6156(6)	**2.9010(1)**	3.4756(3)	3.5735(4)	3.3873(2)
	10	4.6485(5)	5.6048(6)	**2.8390(1)**	3.7319(3)	3.7766(4)	3.6307(2)

Table 13 shows the average values of IGD results of compared algorithms on WFG8 test problems. WFG8 is a hard nonseparable problem. GrEA shows the best IGD values on WFG8 with four to seven objectives. ε-Two_Arch2 has the worst performance on WFG8 with four and five objectives, and PICEA-g shows the worst IGD values on WFG8 with six and eight objectives. However, ε-Two_Arch2 has the best IGD values on WFG8 with eight objectives. And NSGA-III shows the best performance on WFG8 with nine and ten objectives. SPEA2+SDE is in rank 2 on WFG8 with four to eight and ten objectives. ε-Two_Arch2 shows better IGD values than Two_Arch2 on WFG8 with more than five objectives.

Table 13 Average values of IGD results of the compared algorithms on WFG8

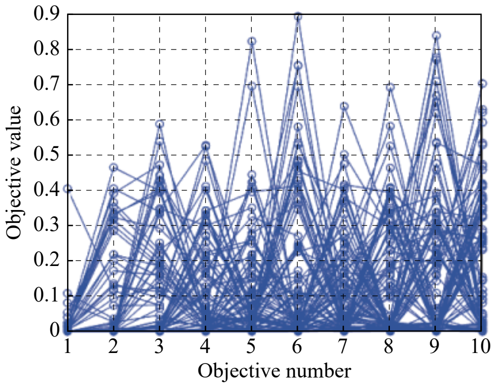
Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
WFG8	4	1.478 7(6)	1.251 1(4)	1.387 7(5)	0.889 9(3)	**0.762 5(1)**	0.816 6(2)
	5	2.278 9(6)	1.907 2(5)	1.872 0(4)	1.607 1(3)	**1.279 2(1)**	1.406 5(2)
	6	2.544 3(4)	2.692 5(5)	2.322 2(3)	2.836 8(6)	**1.792 3(1)**	2.031 8(2)
	7	3.327 4(4)	3.507 5(6)	2.831 9(3)	3.489 2(5)	**2.587 1(1)**	2.677 2(2)
	8	**3.014 2(1)**	4.187 3(5)	3.187 2(3)	4.222 3(6)	3.251 0(4)	3.069 6(2)
	9	3.551 1(2)	5.191 1(6)	**3.391 8(1)**	5.190 1(5)	3.789 3(4)	3.675 8(3)
	10	4.028 4(3)	6.123 0(6)	**3.708 4(1)**	5.575 7(5)	4.168 2(4)	3.915 9(2)

Table 14 shows the average values of IGD results of compared algorithms on WFG9 test problems. WFG9 is a nonseparable-reduced problem. Two_Arch2 shows the worst performance on WFG9 with more than four objectives and PICEA-g also shows the worst IGD values on WFG9 with four objectives. ε-Two_Arch2 has the best performance on WFG9 with four, six, and seven objectives. Furthermore, NSGA-III shows the best IGD values on WFG9 with five, nine, and ten objectives, and GrEA has the best performance on WFG9 with eight objectives. However, SPEA2+SDE has a better performance than ε-Two_Arch2 on WFG9 with more than seven objectives.

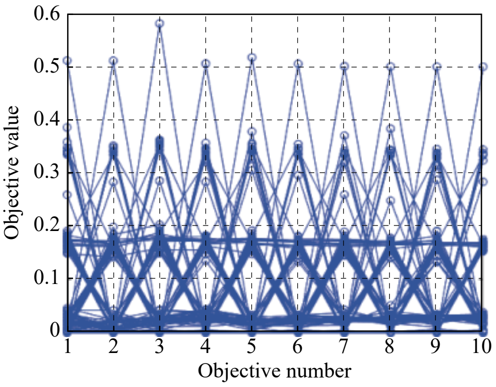
Table 14 Average values of IGD results of the compared algorithms on WFG9

Problem	Number of objectives	ε-Two_Arch2	Two_Arch2	NSGA-III	PICEA-g	GrEA	SPEA2+SDE
WFG9	4	**0.572 4(1)**	0.780 1(5)	0.615 6(3)	0.780 5(6)	0.594 5(2)	0.691 2(4)
	5	1.166 2(2)	1.391 5(6)	**1.158 9(1)**	1.288 3(5)	1.122 2(3)	1.232 3(4)
	6	**1.384 5(1)**	2.093 0(6)	1.825 7(3)	1.897 8(5)	1.724 4(2)	1.851 3(4)
	7	**2.151 7(1)**	2.932 2(6)	2.456 3(3)	2.773 9(4)	2.344 8(2)	2.683 2(5)
	8	3.453 2(5)	3.634 8(6)	3.025 8(3)	3.287 7(4)	**2.999 8(1)**	3.019 2(2)
	9	4.431 8(5)	4.544 0(6)	**3.032 1(1)**	3.976 3(4)	3.464 4(2)	3.647 0(3)
	10	5.280 6(5)	5.598 1(6)	**3.039 9(1)**	4.191 4(4)	3.930 1(3)	3.783 1(2)

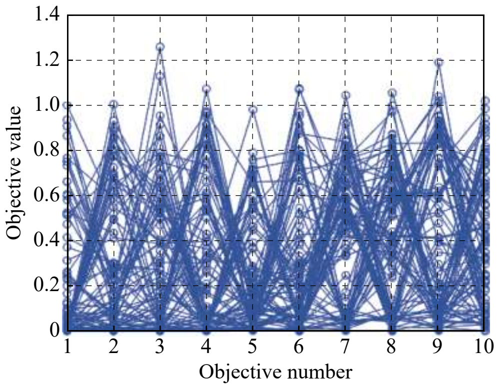
In summary, ε-Two_Arch2 and NSGA-III are the two best-performing algorithms on DTLZ test problems. NSGA-III has the best IGD values on all the DTLZ1 test problems, and ε-Two_Arch2 shows the best performance on all the DTLZ2, DTLZ3 and DTLZ4 test problems. The parallel coordinate plots of the best solutions sets obtained by ε-Two_Arch2 and NSGA-III on the DTLZ test problems with ten problems are shown in Fig. 3. It is clear to find that ε-Two_Arch2 extends more areas of the entire PF than NSGA-III on DTLZ2-4 test problems. Furthermore, ε-Two_Arch2 outperforms Two_Arch2 on IGD values of all the DTLZ test problems which can demonstrate the effectiveness of the ε-dominance and the boundary protection strategy. Focused on the experiments on the WFG problems, it is harder to maintain the diversity and convergence for the selected algorithms because of more transformation functions. Except from the WFG1 and WFG3 test problems, ε-Two_Arch2 has better distribution and convergence, and it shows the best IGD values on all the WFG5 and WFG6 test problems. Compared with Two_Arch2, ε-Two_Arch2 has better IGD values except from WFG1.



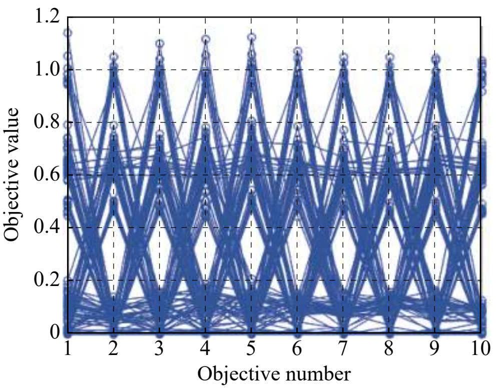
(a) ε-Two_Arch2 on DTLZ1 with ten objectives



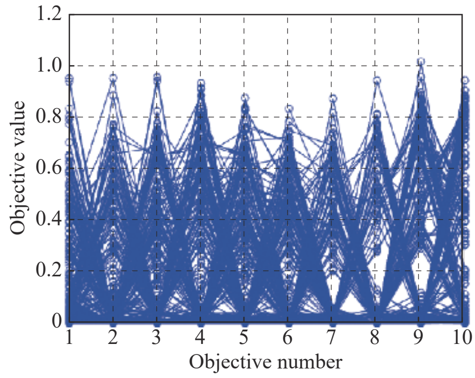
(b) NSGA-III on DTLZ1 with ten objectives



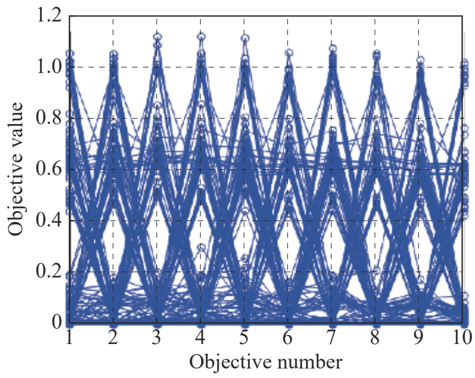
(c) ε-Two_Arch2 on DTLZ2 with ten objectives



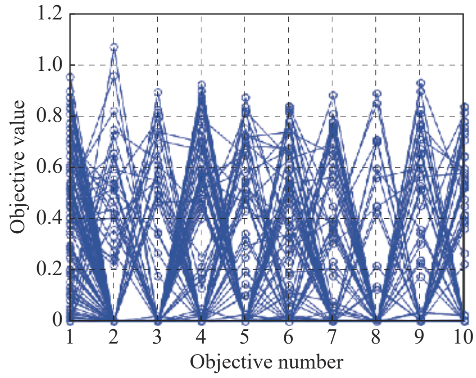
(d) NSGA-III on DTLZ2 with ten objectives



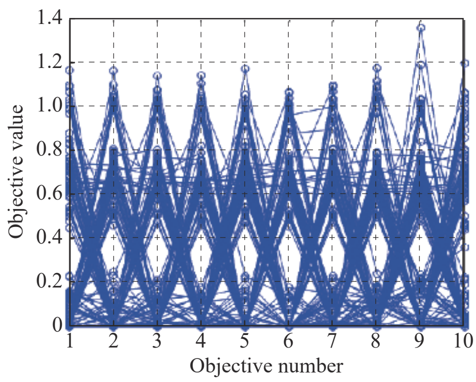
(e) ε-Two_Arch2 on DTLZ3 with ten objectives



(f) NSGA-III on DTLZ3 with ten objectives



(g) ε-Two_Arch2 on DTLZ4 with ten objectives

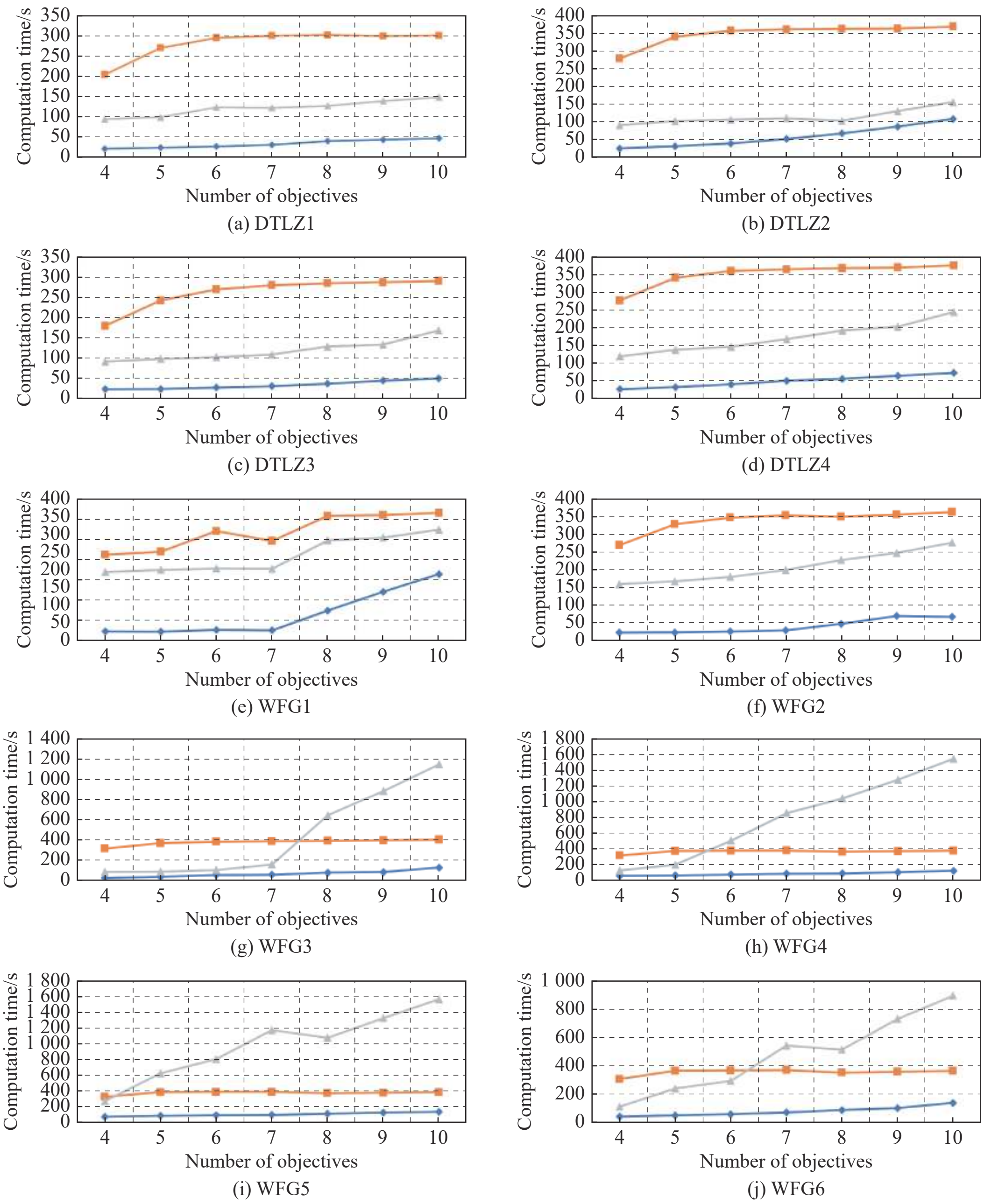


(h) NSGA-III on DTLZ4 with ten objectives

Fig. 3 Parallel coordinate plot of the best solution set obtained by ε-Two_Arch2 and NSGA-III on DTLZ test problems with ten objectives

4.4 Effectiveness of the fast-update strategy

The purpose of the fast-update strategy is decreasing the computational complexity which contains the procedure of updating DA and CA. The proposed ε-Two_Arch2 is mainly composed of three parts: reproduction, updating DA, and updating CA. In reproduction, the SBX and PM operators are adopted to generate offspring, which consumes a computational complexity of $O(DN)$, where D and N are the number of decision variables and the population size, respectively. In updating DA, the procedure of ε-dominance requires $O(N\log N)$ computations. In updating CA, the computational complexity of $I_{\varepsilon+}(x,y)$ and $d(x,C)$ calculation is $O(mN^2)$ and $O(mN)$, respectively. Overall, the computational complexity of the ε-Two_Arch2 is $O(mN^2)$. As a contrast, the complexity of Two_Arch2 is $\max\{O\left(N\log^{m-2}N\right), O(mN^2)\}$. The computation times comparison between ε-Two_Arch2, Two_Arch2 and ε-Two_Arch2 without fast-update strategy (*ε-Two_Arch2) on DTLZ and WFG test problems is shown in Fig. 4. It can be easily observed that ε-Two_Arch2 costs less computation time than Two_Arch2 and *ε-Two_Arch2 on all the DTLZ and WFG test problems which can demonstrate the effectiveness of the fast-update strategy.



(a) DTLZ1 (b) DTLZ2 (c) DTLZ3 (d) DTLZ4 (e) WFG1 (f) WFG2 (g) WFG3 (h) WFG4 (i) WFG5 (j) WFG6

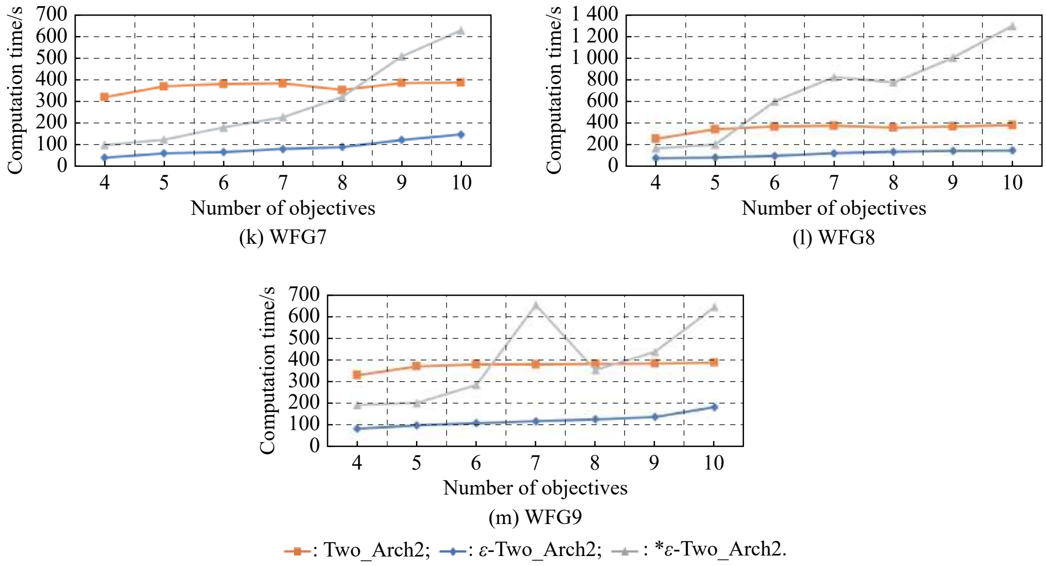


Fig. 4 Computation time comparison between ε-Two_Arch2, Two_Arch2, and *ε-Two_Arch2 on DTLZ and WFG test problems

5. Conclusions

In this paper, an ε-dominance based Two_Arch2 algorithm is proposed to solve MaOPs, called ε-Two_Arch2. Firstly, ε-Two_Arch2 adopts ε-dominance as the selection strategy in DA to decrease the loss of the individuals on PF and enhance the selection pressure. To obtain a more balanced archive as final output of the proposed algorithm, a boundary protection strategy based on $I_{\varepsilon+}$ indicator is assigned to update the CA which can maintain satisfactory distribution and convergence. In addition, a fast-update strategy is proposed to preserve the population size and decrease the computational complexity. Finally, ε-Two_Arch2 shows a competitive performance with the compared MaOEAs through comparative experiments. However, ε-Two_Arch2 has no satisfactory performance on several WFG problems, because the allocations of ε and P are not adaptive. The more efficient and non-parametric environmental selection scheme would be assigned on two archives in the future work.

References

[1] DEB K. Multi-objective optimization using evolutionary algorithms. New York: John Wiley & Sons, 2001.

[2] MIETTINEN K. Nonlinear multiobjective optimization. Berlin: Springer, 1999.

[3] DEB K, AGARWAL S, PRATAP A, et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Proc. of the 6th International Conference on Parallel Problem Solving from Nature, 2000: 849–858.

[4] ZITZLER E, LAUMANNS M, THIELE L. SPEA2: improving the strength pareto evolutionary algorithm. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.5073&rep=rep1&type=pdf. DOI: 10.3929/ethz-a-004284029.

[5] PRADITWONG K, YAO X. How well do multi-objective evolutionary algorithms scale to large problems. Proc. of the IEEE Congress on Evolutionary Computation, 2007: 3959–3966.

[6] ISHIBUCHI H, TSUKAMOTO N, NOJIMA Y. Evolutionary many-objective optimization: a short review. Proc. of the IEEE Congress on Evolutionary Computation, 2008: 2419–2426.

[7] HERNANDEZ-DIAZ A G, SANTANA-QUINTERO L V, COELLO COELLO C A, et al. Pareto-adaptive ε-dominance. Evolutionary Computation, 2007, 15(4): 493–517.

[8] DEB K, MOHAN M, MISHRA S. Evaluating the ε-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. Evolutionary Computation, 2005, 13(4): 501–525.

[9] WANG R, PURSHOUSE R C, FLEMING P J. Preference-inspired coevolutionary algorithms for many-objective optimization. IEEE Trans. on Evolutionary Computation, 2013, 17(4): 474–494.

[10] LI M, YANG S X, LIU X H. Shift-based density estimation for pareto-based algorithms in many-objective optimization. IEEE Trans. on Evolutionary Computation, 2014, 18(3): 348–365.

[11] DAI C, WANG Y P, HU L J. An improved α-dominance strategy for many-objective optimization problems. Soft Computing, 2016, 20(3): 1105–1111.

[12] LIU Y, ZHU N B, LI K L, et al. An angle dominance criterion for evolutionary many-objective optimization. Information Sciences, 2020, 509: 376–399.

[13] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans. on Evolutionary Computation, 2014, 18(4): 577–601.

[14] YANG S X, LI M Q, LIU X H, et al. A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans. on Evolutionary Computation, 2013, 17(5): 721–736.

[15] ASAFUDDOULA M, RAY T, SARKER R. A decomposition-based evolutionary algorithm for many objective optimization. IEEE Trans. on Evolutionary Computation, 2015, 19(3): 445–460.

[16] YUAN Y, XU H, WANG B, et al. Balancing convergence and diversity in decomposition-based many-objective optimizers. IEEE Trans. on Evolutionary Computation, 2016, 20(2): 180–198.

[17] LYU C Z, LI W M. Evolutionary many objective optimization based on bidirectional decomposition. Journal of Systems Engineering and Electronics, 2019, 30(2): 319–326.

[18] BADER J, ZITZLER E. HypE: an algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation, 2011, 19(1): 45–76.
[19] BADER J, ZITZLER E. A hypervolume-based optimizer for high-dimensional objective spaces. JONES D, TAMIZ M, RIES J. ed. New developments in multiple objective and goal programming. Berlin: Springer, 2010.
[20] BROCKHOFF D, WAGNER T, TRAUTMANN H. On the properties of the R2 indicator. Proc. of the 14th Annual Conference on Genetic and Evolutionary Computation, 2012: 465–472.
[21] GOMEZ R H, COELLO COELLO C A. Improved metaheuristic based on the R2 indicator for many-objective optimization. Proc. of the Annual Conference on Genetic and Evolutionary Computation, 2015: 679–686.
[22] SUN Y N, YEN G G, YI Z. IGD indicator-based evolutionary algorithm for many-objective optimization problems. IEEE Trans. on Evolutionary Computation, 2019, 23(2): 173–187.
[23] SAXENA D K, DEB K. Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: employing correntropy and a novel maximum variance unfolding. Proc. of the 4th International Conference on Evolutionary Multi-criterion Optimization, 2007: 772–787.
[24] FREITAS A R R, FLEMING P J, GUIMARAES F G. A non-parametric harmony-based objective reduction method for many-objective optimization. Proc. of the IEEE International Conference on Systems, 2013: 651–656.
[25] YUAN Y, ONG Y, GUPTA A, et al. Objective reduction in many-objective optimization: evolutionary multiobjective approaches and comprehensive analysis. IEEE Trans. on Evolutionary Computation, 2018, 22(2): 189–210.
[26] LI M H, WEI J X, SONG A, et al. Objective reduction using objective sampling and affinity propagation for many-objective optimization problems. IEEE Access, 2019, 7: 68392–68403.
[27] PRADITWONG K, YAO X. A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm. Proc. of the International Conference on Computational Intelligence and Security, 2006: 286–291.
[28] LI B D, LI J L, TANG K, et al. An improved two archive algorithm for many-objective optimization. Proc. of the IEEE Congress on Evolutionary Computation, 2014: 2869–2876.
[29] DAI C. Two-archive evolutionary algorithm based on multi-search strategy for many-objective optimization. IEEE Access, 2019, 7: 79277–79286.
[30] WANG H D, JIAO L C, YAO X. Two_Arch2: an improved two-archive algorithm for many-objective optimization. IEEE Trans. on Evolutionary Computation, 2015, 19(4): 524–541.
[31] LIANG Z P, LUO T T, HU K F, et al. An indicator-based many-objective evolutionary algorithm with boundary protection. IEEE Trans. on Cybernetics, 2021, 51(9): 4553–4566.
[32] ZHANG X Y, TIAN Y, JIN Y C. A knee point-driven evolutionary algorithm for many-objective optimization. IEEE Trans. on Evolutionary Computation, 2015, 19(6): 761–776.
[33] WANG R, ZHANG Q F, ZHANG T. Pareto adaptive scalarising functions for decomposition based algorithms. Proc. of the International Conference on Evolutionary Multi-Criterion Optimization, 2015: 248–262.
[34] DEB K, THIELE L, LAUMANNS M, et al. Scalable multi-objective optimization test problems. Proc. of the Congress on Evolutionary Computation, 2002: 825–830.
[35] HUBAND S, HINGSTON P, BARONE L, et al. A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans. on Evolutionary Computation, 2006, 10(5): 477–506.
[36] ZHANG Q F, ZHOU A, ZHAO S Z, et al. Multiobjective optimization test instances for the CEC 2009 special session and competition. http://www.al-roomi.org/multimedia/CEC_Database/CEC2009/MultiObjectiveEA/CEC2009_MultiObjectiveEA_TechnicalReport.pdf.
[37] ZITZLER E, THIELE L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. on Evolutionary Computation, 1999, 3(4): 257–271.
[38] DEB K, AGRAWAL R B. Simulated binary crossover for continuous search space. Complex Systems, 1994, 9(3): 115–148.

Biographies

WU Tianwei was born in 1996. He received his B.S. degree in engineering degree from Hefei Normal University, China in 2017. He is currently working toward his M.S. degree in control engineering at China Jiliang University, Hangzhou, China. His research interests include many-objective optimization problems and objective reduction optimization algorithms.
E-mail: 554729930@qq.com

AN Siguang was born in 1981. She received her M.S. and Ph.D. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 2006 and 2012. From 2006 to 2014, she was a research assistant in China Jiliang University, Hangzhou, China. Since 2015, she has been an assistant professor with the Electrical Engineering Department, China Jiliang University. Her research interests include optimization algorithms, electromagnetic inverse problems, and the computational method on electromagnetic devices.
E-mail: annsg@126.com

HAN Jianqiang was born in 1970. He received his Ph.D. degree in electron science and technology from Xi'an Jiaotong University in 2003. He was a post-doctoral fellow with the State Key Laboratory of Transducer Technology, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, from 2004 to 2005. Since 2006, he has been working as a professor with the College of Mechanical and Electrical Engineering, China Jiliang University. His research interests are micromachined resonant accelerometers, resonant IR detectors, and thin-film thermal converter.
E-mail: hjqsmx@sina.com

SHENTU Nanying was born in 1977. She received her B.S. and M.S. degrees from Chongqing University, Chongqing, China, in 1999 and 2003, respectively. She is currently pursuing her Ph.D. degree at National Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, Zhejiang, China. She is a lecturer at China Jiliang University, Hangzhou, Zhejiang. Her research interests cover industrial field parameter measurement, design, fabrication and theoretical modeling of sensors, and geohazard monitoring.
E-mail: stnying@163.com