Adaptive Neighborhood Selection for Multiobjective Optimization Problems

Yuping Zhang[a], Juan Zou[a,*], Jinhua Zheng[a,b], Yuan Liu[a], Shengxiang Yang[c]

[a] *Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, Information Engineering College of Xiangtan University, Xiangtan, Hunan Province, China*
[b] *Ministry of Education and School of Computer Science and Technology Hengyang Normal University, HengYang, Hunan Province, China*
[c] *School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K.*

Abstract

It is generally that the conflict between the convergence and distribution is deteriorated with the increase of the number of objectives. Furthermore, Pareto dominance loses its effectiveness in many objective problems. Therefore, a more valid selection mechanism should be proposed to maintain the two properties. This paper presents a multiobjective evolutionary algorithm, called *Adaptive Neighborhood Selection for Multiobjective evolutionary algorithm*(ANS-MOEA), to deal with multiobjectives optimization problems (MOPs). The method defines the ability of every individual with two categories of information, convergence information (CI) and distribution information (DI). In the critical layer, well-converged individual is selected firstly in the population, and its neighbors, calculated by DI, are pushed into neighbor collection soon afterwards. Finally, the proper distribution of the population is ensured by competition so that large DI goes back to the population and low DI remains in the collection. Four state-of-the-art multiobjective evolutionary algorithms are selected as the competitive algorithms to validate ANS-MOEA. The experimental results show that ANS-MOEA can solve a many objective optimization problems and generate a

☆Fully documented templates are available in the elsarticle package on CTAN.
*Corresponding author
Email address: support@elsevier.com (Shengxiang Yang)

set of remarkable solutions to balance convergence and distribution.

Keywords: Many objectives optimization problems, Multiobjective evolutionary algorithm, Distribution, Convergence, Neighborhoods, Selection mechanism

1. Introduction

In the real world, many problems have several objectives that must be met simultaneously. These problems are called multiobjective optimization problems (MOPs). Because objectives may have conflicting features, there is no single optimal solution for a MOP. In the early development, evolutionary algorithms (EAs) solve a MOP through some methods that integrate several objectives into a single objective to optimize. With the development of EAs, a number of state-of-the-art multiobjective evolutionary algorithms (MOEAs) have been proposed gradually, such as NSGAII [1], ϵ-MOEA [2], MOEAD [3], SPEA2 [4] etc. In fact, the applied research of MOEA is the hot topic in current world. In the academic field, a number of papers research how to solve the practical problems. MOEAs also have been successfully used in data mining [5, 6], flowshop schedule [7], wireless sensor networks [8, 9, 10], machine design [11], vehicle routing [12, 13], electric power dispatch [14], aircraft spare parts allocation problem [15], control system design [16, 17, 18]. In recent years, some related techniques have accelerated the development of MOEAs, including test problem [19, 20, 21], performance assessment metrics [22, 23], and experimental platform [24].

Without loss of generality, the form of a MOP can be stated as follows:

$$\begin{cases} Minimize \quad \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}))^T, \\ subject \quad to \quad \mathbf{x} \in \Omega \end{cases} \tag{1}$$

where $\mathbf{x} \in \Omega$ is the decision vector and Ω is the decision space: $\Omega = \{\mathbf{x} \in R^n | h_j \leq 0, j = 1, ..., m\}$. $f_j(x), j = 1, 2, ..., m$, is the objective function and m is the number of objectives, so R^m is defined as the objective space.The attainable objective set is described as the set $\{F(x)|x \in \Omega\}$.

In general, there is conflict among all the objectives which are described in equation (1). Therefore, there are no points in Ω that can optimize all of the objective functions simultaneously. When dealing with MOPs, convergence and distribution are the two significant indicators that should be considered simultaneously in the MOEAs. Convergence denotes the distance from PF_{known} (the Pareto front (PF) is calculated by algorithm) to PF_{true} (true PF). And distribution denotes the distribution of the population in the PF. It has been proven that convergence and distribution are conflict with each other and the convergent rate is bound to be affected when we consider the well performance of distribution [25]. With an increasing number of objectives for a MOP, these two indicators become a very challenging topic. Overall, an essential component of MOEAs is that they are able to design a selection mechanism to maintain the distribution without affecting the convergence as far as possible and then obtain a set of trade-off (balance the convergence and distribution) points. There are seven major categorizations of MOEAs based on the selection mechanism [26]:

1. Scalarizing-based method: All subgoals are either optimized combination or gathered into a single target. Thus, a MOP can be converted into single objective optimization problem. The typical representative of these kinds of methods are [3, 27].

2. Indicator-based method: The performance of indicators is used for fitness assignment. A typical algorithm like IBEA [28], which uses $I_{\epsilon+}-$indicator and $I_{HD}-$indicator to compare the quality of two Pareto set approximations to each other.

3. Relaxed dominance based method: In general, this kind of method aim to enlarge the dominating area of the non-dominated solutions by changing the objective values when they are in comparison so that some of them are more likely to be dominated by others [2, 29].

4. Diversity-based method: This method improves the performance by re-

ducing the adverse impact of diversity maintaince. Take SDE [30] as an example, it proposed a shift-based density estimation strategy in original paper. Briefly, it works by turning all of the superior objectives into the value which is the same as the detected individual and then calculates their density.

5. Reference set based method: In recent years, some typical methods have been proposed which are based on a reference set. One of the representative algorithms is NSGAIII [31]. These methods use a set of reference solutions to measure the quality of solutions. Thus, the search process is guided by the solutions in the reference solution set.
6. Preference-based method: In most real world problems, the decision maker (DM) would like to incorporate his/her preferences into the search process. These preferences are used to guide the search toward the preferred parts of the Pareto region. The typical methods are [32, 33, 47, 48].
7. Dimensionality reduction method: The dimensionality reduction approach aims to deal with MOPs by reducing the number of objectives. In general, researchers usually analyze problems based on principal component analysis (PCA) [34, 35].

In our proposed method ,called ANS-MOEA, the population is stratified into several layers with Pareto dominance and it is described as follows:

$F_1 \succ F_2 \succ \cdots \succ F_l$.

where l is the number of the layers and smaller subscripts dominate bigger subscripts. The individuals in the layers which corresponding smaller subscript priority to be selected into archive. When $F_k, (k \leq l)$ size is more than the remaining size of the archive, the F_k is defined as the critical layer (CL). In addition, every individual is endowed with convergence information (CI) and distribution information (DI). At the time of screening the critical layer, the elite point has a smaller CI in the CL will be selected into archive set firstly. Then, the neighborhood of the elite will be selected into the Neighbor Collection (NC), which is used to store worse points. The algorithm can retain well-distributed individual by comparing the DI when NC is saturation. The adaptive neighborhood selection mechanism adjusts the size of the neighborhood depend on the number of individuals in the CL. In the case of a low dimension, the CL has fewer individuals and sparse population density. If specification of the neighborhood is set too small, CL has to be compared too many times to obtain the well-distributed individual, whereas most individuals have well distribution so that the operation time is consumed. In contrast, almost all individuals are distributed on the critical layer in case of high dimensions. In order to obtain well-distributed individuals as well as avoid losing well-converged individuals, the smaller size for neighborhood is adjusted adaptively to find the well-performance trade-off individuals more exactly.

The rest of this paper is organized as follows. Section 2 is devoted to illustrating the proposed MOEA. Section 3 introduces the experimental design and four state-of-the-art algorithms, and these are GDE3 [36, 37], MOEA/D [3], IBEA [28], SMPSO [38]. Section 4 shows the experimental results. Section 5 draws the conclusions of this paper.

2. The Proposed Method

In this section, we describe the detail of our approach. Conveniently, a reference point, $z = (z_1, ..., z_m)^T$, is given the lowest value for each objective in this method and all individuals are translated into the first quadrant by subtracting the relevant $z_j, (j = 1, 2, ..., m)$. During the evolutionary process, the reference point must be updated constantly.

2.1. Convergence Information(CI)

Convergence information(CI) [25] reflects the convergence ability of an individual. In this paper, ANS-MOEA calculates the CI of an individual **P** in the population by summing its value in each objective:

$$CI(\mathbf{P}) = \sum_{i=1}^{m} f_i(p) \tag{2}$$

where $f_i(p)$ devotes the $\mathbf{i}_{th}$ objective value of $\mathbf{P}$ and m is the number of objectives. It can be noted that there are two information which equation (2) reflects:

1. m denotes the number of objectives and $f_i(p)$ reflects the performance for each point;
2. A point with good performance in most of the objectives probably has a lower value of CI.

The role of CI is that the individual which has the lowest CI value in the current population is selected into the archive first during the critical selection. The proposed selection mechanism in the critical layer retains well-converged individuals by this method in this paper.

It is worth noting that the CI cannot completely represent the convergence information of an individual. In a case where an individual is a DRS, which is called the dominance resistant solution (it has the lowest value on one or several objectives but performs the worst on the other objectives), it has a lower CI but does not approximate to the PF. In addition, several points with lower CI are often clustered. Therefore, in order to enable the final set to distribute on the PF uniformly, the distribution information(DI) is used.

2.2. Distribution Information(DI)

The calculation of DI for an individual occurs after a point $\mathbf{P}$, which has a lowest CI in current critical population, was put into the archive. The DI of an individual $\mathbf{p_1}$ is described as follows:

$$DI(\mathbf{p_1}) = \arccos \frac{\overrightarrow{\mathbf{OP}}\overrightarrow{\mathbf{Op_1}}}{|\overrightarrow{\mathbf{OP}}||\overrightarrow{\mathbf{Op_1}}|} \tag{3}$$

It can be seen in the left of Figure 1, when well-converged point $\mathbf{P_1}$ was selected into the archive, the DI of points $\mathbf{P_2}$ and $\mathbf{P_3}$ were θ_{12} and θ_{13} respectively. But the θ_{12} and θ_{13} are not the final distribution information of $\mathbf{P_2}$ and $\mathbf{P_3}$. In order to avoid a case in which $\mathbf{P}$ has a bigger angle with one point, but has a smaller angle with another point,the $DI(\mathbf{P})$ is updated and given the

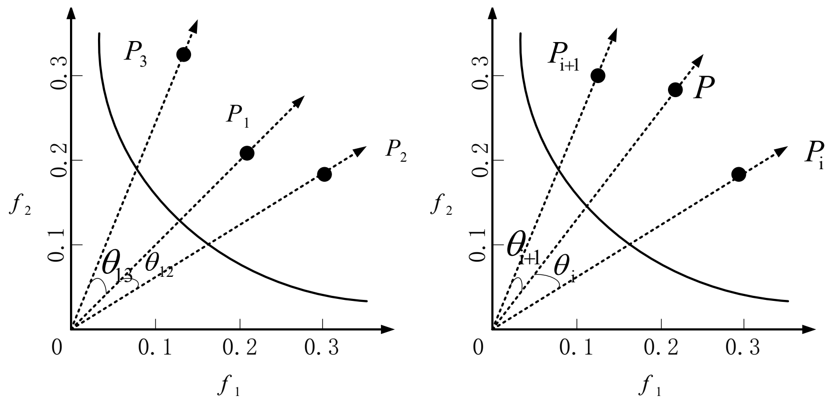

Figure 1: The definition of distribution information of an individual is described in the left picture and the update process about DI is described in the right.

latter value. For example, as the right of Figure 1 shows, P_i denotes itself is the $\mathbf{i}_{th}$ point which gets the nod into archive, and the P_{i+1} is the next point. When calculating the DI of point P, angle θ_i is indicating the current DI during P_i is the best point, and then the θ_{i+1} denotes the included angle between P and P_{i+1} when P_{i+1} was the next. The right shows that θ_{i+1} is less than θ_i, so the DI of P is updated from θ_i to θ_{i+1}.

2.3. Adaptive Neighborhood Selection

It is significant for the neighborhood selection mechanism to maintain trade-off points that adjust the size of the neighborhood appropriately. When rare individuals distribute in the CL, the distribution of individuals is relatively sparse and the population density in CL is small so that most individuals have better distribution. A large size of neighborhood can obtain the widely distributed individuals as much as possible. On the contrary, when there are a large number of individuals pushing and squeezing in the CL, the population density in the CL is large. A small size of neighborhood can avoid the influence of convergence speed as much as possible due to the lacking of well-converged individuals. In addition, outstanding individuals must be selected carefully with the comparing times of NC increases.

Algorithm 1 CriticalSelection

Require: P(critical population), $remain$(archive size);

$K = N - remain, T = \lceil K/remain \rceil$; /*Calculate the NC size and neighborhood size*/

$A = \emptyset, NC = \emptyset$; /*Initialize sets A, NC*/

for $j = 1$ **to** m **do**

    $z_j = \min\{f_j(x)|x \in \Omega\}$ /*Calculate the reference point*/

    $f_j^* = f_j(p) - z_j, p \in P$ /*Translation coordinates*/

end for

$CI_p = ConvergenceInformation()$ /*Calculate the CI of each individual*/

while $|A| < remain$ **do**

    $p^* = \{p_i|\forall j \neq i, CI_{p_i} < CI_{p_j}\}$

    $A = A.add(p^*), P = P.remove(p^*), frontSize = frontSize - 1$

    $DistributionInformation()$ /*Calculate the DI of each individual*/

    $Neighborhood(T)$

/*Calculate the neighborhood for p^**/

    $j = 0$

    while$(j < T)$ **do**

        $q_j = FindworstDI(Neighborhood)$ /*Find out the smallest DI in neighborhood*/

        if$(|NC| < K)$ **then**

            $NC.add(q_j)$

        else then

            $p^{'} = FindwellDI(NC)$ /*Find out the biggest DI in NC*/

            if$(DI_{p'} > DI_{p_j})$

                $P.add(p^{'})$,$frontSize = frontSize + 1$

                $NC.remove(p^{'})$

                $NC.add(p_j)$

            end if

        j++

    end while

end while

return A

Algorithm 1 gives the detailed procedure of the approach. In the CL, we define the number of individuals which are needed to screen out as *remain*, and the population in the CL is defined as $frontSize$. Then the size of NC is defined as follows:

$$K = fronSize - remain \tag{4}$$

and the neighborhood size of every individual is defined as:

$$T = \lceil K/remain \rceil \tag{5}$$

The function of NC is to reserve the individuals after the elimination whereas it was used to store neighbors at the beginning. Equation(5), which is shown in line 1 in Algorithm 1, divides the whole CL into the *remain* part. When $K \gg remain$, the density of the population in the CL is usually small and there are a few points needed to be selected. Well-converged individuals (the algorithm procedure is shown in line 9) were selected from each part in turn. NC and archive are saturated at the same time. In this case, no comparison occurred in the NC. But in most instances since dimensions exceed five or even more, archive and the NC are not saturated simultaneously. The saturation rate of the NC is superior to the archive. Line 13-26 describes the progress about arrow 3 presented in Figure 2 in detail, where line 16-18 indicates that individuals are inserted in the NC when the NC is unsaturated. In the case where the NC is saturated, shown in lines 19-24, the points, which in the neighborhood of the subsequent selected well-converged individuals, were added into the NC by comparing with the points in the NC. In this case, the point with the biggest DI was elected from the NC and is merged into the population subsequently when the DI of this point is bigger than the neighbor point.

Figure 2 shows an example of the process of critical selection. Two cases about arrows 3 probably occur and must be illustrated, shown in Figure 2:

1. No exchange occurs when the DI of the points in the neighborhood is bigger than all of the points in the NC.
2. On the contrary, the points which have a larger DI in the NC are replaced

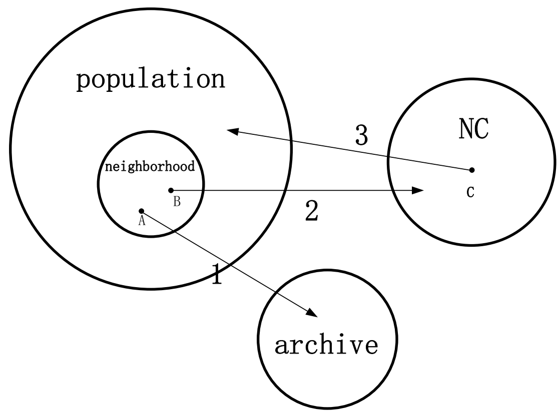

Figure 2: The procedure of critical selection, where A is the well-convergence individual in the current population.

1.As arrow 1 indicates, the well-converged individual was selected into the archive.

2.Arrow 2 expresses the process in which the neighbor B of A was put into the NC.

3.Subsequently, arrow 3 shows the comparison progress in the NC and superior individual C was selected to rejoin the population.

by several smaller DI points in the neighborhood one by one, and then the points take a part in the election of the well-convergence individual.

This process ensures that the worst distribution of individuals were embodied in the NC through comparative analysis by neighborhood selection mechanism and then they were eliminated in the end. The rest of the superior individuals were maintained in the archive as much as possible.

2.4. The General Framework

In this section, we introduce a general framework of algorithm which imitates the framework of NSGAII [1].

As Algorithm 2 shows that line 3 generates the offspring population by $GeneticOperation$, and then these two populations, called the offspring population and the parent population, are merged into one population named $Union_t$, shown in line 4. Line 5 divides the merged population into several non-domination layers by function $NondominateSort$. Subsequently, each layer

Algorithm 2 General Framework

Require: MOP, $MaxGeneration$, N(population size), $g = 0$, m(the number of objectives)

Ensure: P_{t+1}

$P_t = p_1, p_2, ..., p_N$, $A_t = \emptyset$
while$(g < MaxGeneration)$ **do**
    $Q_t = GeneticOperation(P_t)$
    $Union_t = P_t \bigcup Q_t$
    $\{F_1, F_2, ..., F_l\} = NondominateSort(Union_t)$, where $F_1 \succ F_2 \succ ... \succ F_l$
    $i = 0$
    while$(|A_t| \leq N)$ **do**
        $A_t = A_t \bigcup F_i$
        $i = i + 1$
    end while
    if $|A_t| = N$ **then**
        $P_{t+1} = A_t$
        break
    else
        $A_t = A_t - F_i$
        $P_{t+1} = A_t \bigcup CriticalSelection(F_i, N - |A_t|)$
    end if
end while

merges into the P_{t+1}, described in lines 6-10, according to the dominance sequence. In general, the number of points in the CL is more than P_{t+1} needs. And then as the line 16 shows the function $CriticalSelection$, which has a detailed description in Algorithm 1, screens out the proper number of points to merge into P_{t+1}.

3. Experimental Design

This section validates the performance of ANS-MOEA. We give the test problem and performance metrics involved in the experiments first. Then, we have a brief introduction about four other state-of-the-art MOEAs: GDE3 [36], MOEAD [3], IBEA [28], SMPSO [38]. Finally, the results of comparing the algorithms are provided.

In this paper, all of the algorithms which are in comparison are used within the jMetal [24] framework. The framework includes a number of classic and modern state-of-the-art optimizers, a wide set of benchmark problems and a set of well-known quality indicators to assess the performance of the algorithms. In the experiments, a crosser probability $p_c = 0.9$ and a mutation probability $p_m = 1/l$ (where l is the number of decision variables) are used. In addition, the operators for crossover and mutation are simulated binary crossover (SBX [39]) and polynomial mutation(PM [40])with both distribution indexes set to 20. For the proposed method and four other state-of-the-art MOEAs, the population size is 100, and all algorithms were run 30 times independently.

3.1. Test Problems And Performance Metrics

In this paper, DTLZ and the walking fish group (WFG) toolkit [19] were considered as the test suites for the MOEAs. DTLZ is a continuous problem suite that can adjust to any number of objectives and decision variables, commonly used in many-objective optimization. For the DTLZ suite, the problems can be classified into two groups roughly. One group, including DTLZ2, DTLZ4, DTLZ5, and DTLZ7, is used to investigate the ability of an algorithm to cope with problems with different shapes. The other group, including DTLZ1, DTLZ3 and DTLZ6, which creates more obstacles, is used to test the ability of convergence of an algorithm. In addition, WFG is also a continuous problem suite that can be scaled to any number of objectives and decision variables. The parameters k (position parameters) and l (distance parameters) in WFG are set to $2 \times (m-1)$ and 20, respectively, where m denotes the number of objectives. Table 1 describes the characteristics of the problem DTLZ1-DTLZ7 and Table 2 describes the details of WFG1-WFG9.

Table 1: The characteristic of DTLZ suite, where M and N denote the number of objectives and decision variables, respectively

Problem	Properties	M	N
DTLZ1	Linear,Multimodal	3, 5, 8, 10, 15	M+9
DTLZ2	Concave	3, 5, 8, 10, 15	M+9
DTLZ3	Concave,Multimodal	3, 5, 8, 10, 15	M+9
DTLZ4	Concave,Biased	3, 5, 8, 10, 15	M+9
DTLZ5	Concave,Degenerate	3, 5, 8, 10, 15	M+9
DTLZ6	Concave,Degenerate,Biased	3, 5, 8, 10, 15	M+9
DTLZ7	Mixed,Disconnected,Biased	3, 5, 8, 10, 15	M+9

Besides the test problem mentioned above, two widely acknowledged performance metrics, GD [22, 41] and IGD [23, 30], were used to assess the performance of the proposed method, ANS-MOEA, and the other four algorithms, GDE3 [36], MOEAD [3], IBEA [28], SMPSO [38].

GD(generational distance) is used to assess the ability of convergence by calculating the average Euclidean distance from the final solution set to the true Pareto front and is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{6}$$

where n is the number of solution in the final solution set PF_{kown}, and d_i denotes the minimum Euclidean distance between each of these solutions and

Table 2: The characteristic of WFG suite, where M, k and l denote the number of objectives, position parameters and distance parameters respectively)

Problem	Properties	M	k	l
WFG1	Convex, Mixed, Biased	5, 10, 15	$2 \times (M-1)$	20
WFG2	Convex, Nonseparable, Disconnected	5, 10, 15	$2 \times (M-1)$	20
WFG3	Linear, Nonseparable, Degenerate	5, 10, 15	$2 \times (M-1)$	20
WFG4	Concave, Separable	5, 10, 15	$2 \times (M-1)$	20
WFG5	Concave, Separable	5, 10, 15	$2 \times (M-1)$	20
WFG6	Concave, Nonseparable	5, 10, 15	$2 \times (M-1)$	20
WFG7	Concave, Separable	5, 10, 15	$2 \times (M-1)$	20
WFG8	Concave, Nonseparable	5, 10, 15	$2 \times (M-1)$	20
WFG9	Concave, Nonseparable, Multimodal	5, 10, 15	$2 \times (M-1)$	20

the point of the Pareto optimal set PF_{true}. The value of GD illustrates the deviation degree between PF_{kown} and PF_{true}. It is clear that a value of GD=0 indicates that all the generated elements are in the Pareto front.

IGD(invert generational distance) is selected since it can provide combined information about convergence and distribution of a solution set. IGD measures the average distance from the individuals in the Pareto front to the closest solution in the PF_{kown}. Mathematically, let P^* be a reference set representing the Pareto front, and the IGD value from P^* to the PF_{kown} is defined as follows:

$$IGD = \frac{\sum_{z \in P^*} d(z, P)}{|P^*|} \tag{7}$$

where P^* denotes the number of individuals in P^* and $d(z, P)$ is the nearest Euclidean distance from z to P (PF_{kown}). The value of IGD reflects the comprehensive performance of an algorithm. The lowest value of IGD is the best, which points out that PF_{kown} is close to PF_{true} and has a good distribution.

3.2. Description Of The Four Other Evolutionary Algorithms

In order to verify the proposed method ANS-EMOA, four state-of-the-art MOEAs were selected in the experiment. The brief introduction of the characteristics of these MOEAs are described as follows:

1. GDE3 [36]. A developed version of generalized differential evolution, and this is an extension of differential evolution for global optimization with a set of objectives and constraints. The basic idea GDE3 proposed is similar to the first GDE version in which the trial vector was selected to replace the old vector in the next generation if it weakly constraint-dominated the old vector.
2. MOEAD [3]. MOEAD is one of the most popular MOEAs. It is an aggregation-based algorithm that decomposes the many-objective problems into N scalar optimization subproblems with the N predefined well-distributed weight vectors. With the evolution of these subproblems, well individuals were integrated into the archive set. There are three aggregation function: 1) weight sum, 2) Tchebycheff and 3) penalty-based boundary intersection (PBI) approach are proposed in the original paper. In this paper, we chose the MOEA/D with the Tchebycheff.
3. IBEA [28]. In the original paper, the binary quality indicators, $I_{\epsilon+}-$indicator and $I_{HD}-$indicator, were used to compare the quality of two Pareto set approximations relative to each other. The paper mentioned that the binary quality indicators represent a natural extension of the Pareto dominance relation, and therefore can directly be used for fitness calculations similar to the common Pareto-based fitness assignment schemes. In this paper, $I_{HD}-$indicator was used for the IBEA in experiment.
4. SMPSO [38] uses the velocity constriction mechanism to limit the speed of the particles. In order to control the particle's velocity, the proposed idea adopted a constriction coefficient obtained from the constriction factor.

4. Experiments Results And Discussions

This section presents the experimental results of all the MOEAs compared and all the results were presented by tables and figures. In addition, some analysis and discussions of the results were also described after that.

Table 3: The GD values (mean and standard deviation) of the obtained solutions of the five algorithms on the DTLZ problems, where the best values are shown with a deep gray background and the second best with a light gray background, respectively.

Problem	Obj.	ANS-MOEA	GDE3	MOEAD	IBEA	SMPSO
DTLZ1	3	5.32E-4(1.1E-3)	2.42E-3(2.6E-4)	1.70E-2(4.8E-2)	8.67E-2(1.1E-1)†	1.17E-2(1.4E-2)
	5	3.12E-3(5.5E-3)	1.03E+0(4.0E-1)†	3.26E-3(1.0E-3)†	3.88E-2(4.5E-2)†	7.77E+0(4.4E+0)†
	8	5.28E-3(9.6E-3)	1.85E+1(2.1E+1)†	1.02E-1(1.1E-1)	7.66E-2(8.4E-2)	3.94E-2(1.1E-2)†
	10	2.80E-4(2.8E-4)	8.31E+0(4.3E-1)	1.25E+0(6.8E-1)†	6.15E-2(4.1E-2)†	4.48E-2(4.8E-3)†
	15	6.30E-4(1.3E-3)	1.77E+1(1.9E+0)†	8.34E-1(1.2E+0)	1.67E-3(1.5E-3)†	1.36E+1(9.1E-1)†
DTLZ2	3	3.90E-4(1.7E-4)	1.42E-3(3.5E-4)†	9.34E-5(2.7E-5)	1.22E-6(6.1E-7)	2.07E-3(1.9E-4)†
	5	4.12E-4(3.1E-4)	1.60E-2(4.8E-4)	1.42E-3(3.6E-4)†	7.77E-6(5.2E-6)†	4.88E-2(5.3E-3)†
	8	2.80E-4(3.4E-4)	6.49E-2(5.4E-3)†	3.92E-3(3.6E-3)†	2.02E-5(3.8E-6)†	1.31E-2(1.8E-3)†
	10	2.07E-4(4.1E-4)	8.38E-2(2.8E-3)†	9.63E-3(1.1E-2)†	7.21E-5(1.1E-6)†	1.28E-2(3.2E-3)†
	15	1.06E-4(1.0E-4)	3.92E-2(1.2E-3)	1.19E-3(6.6E-4)†	5.10E-6(1.1E-5)	2.98E-2(2.5E-3)†
DTLZ3	3	1.34E-4(1.1E-4)	8.04E-4(2.3E-5)†	1.22E-2(2.4E-2)	6.14E-2(7.1E-2)	1.52E+0(2.1E+0)
	5	1.19E-4(1.4E-4)	1.30E-2(1.9E-3)	6.84E-1(1.4E+0)	9.20E-4(3.5E-4)†	1.30E+1(6.7E-1)†
	8	4.71E-3(8.8E-3)	1.51E+1(4.6E+0)†	1.61E+0(2.3E+0)	9.45E-5(3.9E-5)	4.22E-2(2.2E-2)†
	10	3.12E-8(1.3E-8)	2.23E-1(1.9E-2)†	6.36E-3(7.7E-3)	1.52E-5(1.3E-5)	1.81E-2(3.9E-3)†
	15	3.42E-6(6.1E-6)	1.82E-1(2.0E-2)	1.60E-6(1.4E-6)	4.50E+2(1.3E+3)	3.63E-2(9.6E-3)†
DTLZ4	3	2.03E-4(1.8E-4)	1.27E-3(2.3E-4)	1.50E-4(9.8E-5)	1.09E-5(1.1E-6)	1.79E-3(2.5E-4)†
	5	2.07E-4(1.4E-4)	1.54E-2(1.1E-3)†	4.57E-4(4.3E-4)	1.72E-6(3.4E-6)†	2.65E-2(3.5E-3)
	8	2.74E-3(4.6E-3)	2.96E-2(4.6E-3)†	1.25E-4(1.5E-4)	1.73E-4(1.7E-7)	1.05E-2(5.5E-3)†
	10	1.84E-5(4.2E-7)	3.47E-2(6.1E-3)†	1.77E-5(3.1E-5)†	1.01E-5(2.0E-5)†	8.60E-3(1.9E-3)†
	15	3.80E-6(1.1E-5)	2.05E-2(2.5E-3)†	2.06E-8(4.7E-8)	0.00E+0(0.0E+0)	1.68E-2(1.6E-3)†
DTLZ5	3	1.00E-4(8.3E-5)	3.58E-9(5.2E-9)†	2.58E-5(6.1E-6)†	4.35E-8(2.3E-8)†	1.49E-4(9.2E-5)†
	5	3.68E-4(2.5E-4)	1.24E-2(2.4E-3)†	7.31E-4(6.8E-4)	1.03E-4(6.9E-4)	1.77E-2(4.4E-3)
	8	4.18E-4(4.7E-4)	3.72E-2(6.9E-3)†	2.69E-4(5.1E-4)	2.25E-4(2.7E-4†	5.50E-3(2.7E-3)†
	10	3.04E-4(2.6E-4)	2.26E-2(9.7E-4)†	2.69E-3(1.5E-4)†	5.38E-4(3.7E-4)†	2.84E-3(9.6E-4)†
	15	1.32E-4(9.8E-5)	3.96E-2(7.9E-3)†	1.98E-4(2.8E-4)	8.17E-9(2.4E-8)	1.52E-2(2.2E-3)
DTLZ6	3	1.62E-3(1.3E-3)	0.00E+0(0.0E+0)	0.00E+0(0.0E+0)	2.96E-3(2.7E-3)	6.08E-3(7.4E-5)
	5	1.15E-2(9.0E-3)	2.01E-2(7.0E-3)†	7.44E-2(7.4E-3)†	7.95E-2(2.4E-3)†	1.77E-2(4.4E-3)
	8	2.65E-3(3.9E-4)	4.35E-2(3.0E-3)†	1.73E-4(2.5E-4)†	7.60E-4(7.6E-4)†	1.39E-2(3.9E-3)†
	10	2.46E-3(6.4E-4)	4.20E-2(1.6E-3)†	1.35E-4(2.7E-4)†	1.69E-4(1.9E-4)†	1.28E-2(3.1E-3)†
	15	1.46E-3(5.6E-4)	1.39E-1(3.7E-3)	1.23E-4(2.8E-4)†	6.89E-3(5.6E-3)†	2.62E-2(2.8E-3)†
DTLZ7	3	9.10E-4(2.7E-4)	1.92E-3(6.0E-4)†	2.00E-4(8.8E-5)†	2.85E-3(1.8E-4)†	3.17E-3(6.4E-4)†
	5	9.85E-4(1.2E-3)	1.02E-2(5.8E-4)	3.67E-3(1.5E-3)	1.54E-3(9.7E-4)†	1.49E-2(3.5E-3)†
	8	7.55E-4(8.5E-4)	1.69E-2(2.9E-3)†	1.49E-2(9.4E-3)†	5.63E-6(1.1E-5)	7.12E-2(3.9E-2)†
	10	9.97E-4(7.6E-4)	7.15E-2(1.6E-2)†	2.07E-2(1.3E-2)†	1.35E-7(4.1E-7)†	4.82E-2(4.7E-3)†
	15	1.76E-4(3.5E-4)	6.23E-2(1.5E-2)†	0.00E+0(0.0E+0)†	4.52E-5(1.4E-4)	5.92E-2(1.5E-2)

Table 3 reveals the GD value of five algorithms after the experiments. As
265 can be seen from Table 2, the proposed method has a good convergence rate reflected by GD values. The application of the CI method assists the algorithm in

maintaining well-converged points and ensuring the rate of convergence without the influence of screen process as much as possible. In the table, ANS-MOEA performs the best on 3-, 5-, 10-objectives DTLZ3, 10-objectives DTLZ5, and both 3- and 5-objectives on DTLZ6, DTLZ7, and then it also has the best value for all considered numbers of objectives on DTLZ1. For the other algorithms, GDE3 have the best value on 3-objectives DTLZ5. MOEAD reaches the best results on 15-objectives DTLZ3, 8-, 15-objectives DTLZ4 and 8-, 10-, 15-objectives DTLZ6. In addition, IBEA also has a well converged performance on 8-objectives DTLZ3, 3-, 5-, 10-objectives DTLZ4, 5-, 8-, 15-objectives DTLZ5, 8-, 10-, 15-objectives DTLZ7 and DTLZ2 for all considered numbers of objectives except 8-objectives.

It is worth to noting that the $I_{HD}-$indicator (hypervolume indicator, called HV [42]) is used for IBEA in comparison. So it has good converged performance, but computing time is fairly long.

Table 4 shows the results of IGD for different dimensions on the DTLZ suites. Clearly, the five MOEAs have their own advantages on the test problems. ANS-MOEA performs the best on 10-objectives DTLZ4, DTLZ6, DTLZ7 and most considered numbers of objectives on the other problems except 5-objectives DTLZ1, 8-, 15-objectives DTLZ2, 10-objectives DTLZ3 and 3-, 5-objectives DTLZ5. As for the rest of the MOEAs, GDE3 has good performances on 8-objectives DTLZ3 and low-dimension DTLZ4-7. In addition, IBEA performs the best on 15-objectives DTLZ7. MOEAD obtains the best results on 5-objectives DTLZ1, 8-, 15- objectives DTLZ2, 5-, 8-, 15-objectives DTLZ4 and 8-objectives DTLZ6-7.

Note that MOEAD, which uses the Tchebycheff aggregation method, obtains the better IGD value on most test problems. An important reason is the use of a set of uniform weight vectors. Then, it can obtain a well-distributed solution set. But the result of MOEAD is affected by the shape of Pareto front.

Table 5 gives the performances of MOEAs on WFG. As shown, ANS-MOEA obtains the best results on all considered objectives WFG1, 15-objectives WFG2, 5-, 15-objectives WFG5, 10-objectives WFG7, 5-objectives WFG8 and 10-,

Table 4: IGD (mean and standard deviation) results of the five algorithms on the DTLZ suites, where the best mean is shown with a deep gray background and the second one with a light gray background, respectively.

Problem	Obj.	ANS-MOEA	GDE3	MOEAD	IBEA	SMPSO
DTLZ1	3	1.90E-3(2.4E-4)	2.00E-3(2.8E-5)	9.27E-2(1.8E-1)	3.07E-2(2.0E-2)†	6.60E-3(6.3E-3)
	5	5.04E-3(3.9E-4)	1.32E-1(1.4E-1)	3.13E-3(3.8E-5)†	1.61E-2(4.1E-3)†	4.25E-2(2.8E-2)
	8	6.26E-3(4.0E-4)	3.98E-2(4.3E-3)	7.14E-3(2.3E-3)	1.28E-2(5.0E-4)	2.53E-2(1.0E-3)†
	10	1.03E-2(3.2E-4)	5.33E-2(8.1E-3)†	1.99E-2(8.2E-3)†	2.38E-2(1.0E-2)	3.83E-2(6.9E-4)
	15	5.05E-3(4.7E-4)	3.21E-2(4.1E-3)†	5.08E-3(1.8E-3)	2.74E-2(1.0E-2)†	2.61E-2(2.6E-3)†
DTLZ2	3	1.46E-3(3.8E-5)	1.89E-3(8.6E-5)	3.68E-3(2.8E-5)†	3.02E-3(3.7E-5)†	2.70E-3(8.9E-5)†
	5	3.74E-3(2.9E-4)	6.89E-3(3.0E-4)	4.46E-3(2.3E-4)†	4.17E-3(4.1E-5)†	9.70E-3(3.3E-4)†
	8	6.19E-3(1.9E-3)	1.31FE-2(1.2E-3)†	5.42E-3(6.2E-4)	8.60E-3(3.3E-4)†	1.34E-2(2.3E-4)†
	10	5.51E-3(2.1E-3)	1.47E-2(1.7E-3)†	6.36E-3(1.1E-3)†	8.28E-3(4.5E-4)	1.50E-2(1.9E-4)†
	15	9.57E-3(7.1E-4)	1.13E-2(5.9E-4)†	9.28E-3(1.9E-4)	1.24E-2(5.7E-3)†	1.03E-2(6.0E-4)†
DTLZ3	3	1.80E-3(5.1E-5)	2.02E-3(5.0E-5)†	5.31E-0(2.7E-3)	1.64E-2(7.6E-4)	1.20E-2(8.3E-3)
	5	4.42E-3(1.1E-4)	6.94E-3(5.1E-4)†	6.15E-3(1.0E-3)†	1.89E-2(9.3E-3)†	1.15E-2(1.3E-3)†
	8	6.89E-3(8.7E-5)	1.30E-2(3.4E-4)†	8.85E-3(1.1E-3)†	1.35E-2(1.2E-6)	1.21E-2(3.4E-4)†
	10	9.93E-3(7.7E-8)	9.09E-3(1.2E-3)	9.41E-3(6.0E-4)†	9.94E-3(2.1E-6)†	1.23E-2(1.6E-4)†
	15	5.31E-3(8.6E-7)	6.08E-3(1.4E-3)	5.32E-3(7.2E-7)	2.58E-2(5.7E-3)†	3.40E-3(1.9E-4)†
DTLZ4	3	1.12E-2(1.2E-2)	1.92E-3(4.8E-5)†	4.70E-3(1.8E-3)†	8.61E-3(7.0E-3)	3.35E-3(9.1E-4)†
	5	5.32E-3(4.0E-3)	6.17E-3(1.4E-4)†	5.07E-3(2.1E-4)†	7.86E-3(2.5E-3)†	8.39E-3(8.9E-4)†
	8	1.12E-2(1.7E-3)	8.96eE-3(4.7E-4)†	5.41E-3(2.7E-5)†	6.49E-3(1.1E-3)†	9.39E-3(1.8E-4)†
	10	5.15E-3(2.4E-3)	1.14E-2(1.2E-3)	6.09E-3(3.4E-5)	5.97E-3(4.2E-4)†	1.12E-2(7.6E-5)†
	15	1.12E-2(9.8E-4)	9.09E-3(4.7E-4)†	7.56E-3(1.2E-5)†	1.81E-2(3.1E-3)†	7.71E-3(4.0E-4)
DTLZ5	3	2.05E-4(1.4E-5)	1.63E-4(2.6E-6)†	5.02E-4(8.8E-7)†	1.17E-3(6.4E-5)†	2.10E-4(4.8E-6)†
	5	8.47E-3(2.4E-4)	3.43E-3(2.0E-4)†	5.60E-3(3.1E-4)†	9.14E-3(1.7E-3)†	4.88E-3(2.3E-4)†
	8	4.72E-3(5.5E-4)	4.76E-3(2.5E-4)†	6.91E-3(4.5E-4)†	1.17E-2(2.7E-3)†	8.82E-3(1.9E-4)†
	10	4.81E-3(1.8E-4)	9.27E-2(2.0E-4)†	9.21E-3(2.1E-4)†	1.25E-2(3.1E-3)†	9.34E-3(2.0E-4)†
	15	1.08E-3(2.3E-4)	4.77E-3(1.4E-4)†	1.01E-2(1.2E-4)†	1.52E-2(3.8E-3)†	4.26E-3(8.5E-5)†
DTLZ6	3	3.96E-3(4.8E-3)	1.39E-4(1.4E-6)†	3.93E-4(1.9E-7)	1.72E-3(2.7E-4)†	1.73E-4(5.1E-6)
	5	4.72E-3(2.4E-4)	3.06E-3(2.7E-4)†	6.24E-3(2.7E-4)†	7.35E-3(1.1E-3)†	3.85E-3(2.9E-4)†
	8	7.93E-3(4.3E-4)	1.04E-2(1.4E-3)	7.29E-3(3.0E-5)†	8.72E-3(7.9E-5)†	9.83E-3(4.3E-4)†
	10	6.92E-3(3.2E-4)	1.13E-2(7.5E-4)	7.93E-3(8.1E-5)†	9.45E-3(7.8E-5)†	1.11E-2(3.6E-4)†
	15	8.73E-3(7.4E-5)	7.85E-3(7.7E-4)†	8.93E-3(5.7E-5)	1.03E-2(2.1E-3)†	3.56E-3(1.5E-4)†
DTLZ7	3	2.08E-3(4.7E-4)	1.17E-3(5.2E-5)	2.22E-3(2.5E-5)	1.28E-2(8.0E-3)†	2.21E-3(1.6E-4)
	5	6.09E-3(1.3E-4)	4.65E-3(1.9E-4)†	6.28E-3(2.8E-4)†	2.84E-2(1.4E-3)†	8.85E-3(2.4E-4)†
	8	2.54E-2(6.0E-3)	9.02E-2(2.8E-4)†	1.51E-2(1.7E-3)†	3.88E-2(2.6E-4)†	3.38E-2(6.9E-3)
	10	1.76E-2(2.2E-3)	1.78E-2(2.6E-3)†	3.79E-2(4.1E-3)†	4.71E-2(2.2E-4)†	2.99E-2(3.5E-4)†
	15	1.98E-2(1.4E-3)	1.92E-2(9.8E-4)†	2.06E-2(1.5E-3)†	8.34E-3(9.7E-4)†	1.77E-2(9.1E-4)†

15-objectives WFG9. MOEAD works fairly well on 5-, 10-objectives WFG2, 5-objectives WFG4, 10-objectives WFG5, 10-, 15-objectives WFG6, 5-, 15-objectives WFG7 and 15-objectives WFG8. Beyond that, GDE3 performs

Table 5: IGD (mean and standard deviation) results of the five algorithms on the WFG, where the best mean are shown with a deep gray background and the second best with a light gray background, respectively.

Problem	Obj.	ANS-MOEA	GDE3	MOEAD	IBEA	SMPSO
WFG1	5	5.17E-3(8.2E-4)	1.16E-1(8.3E-4)†	1.13E-1(6.4E-4)†	2.75E-2(7.2E-3)†	1.16E-1(1.2E-3)†
	10	1.00E-2(2.9E-3)	3.68E-1(7.2E-3)†	1.72E-1(1.5E-2)†	9.47E-2(2.9E-2)†	3.67E-1(1.2E-3)†
	15	1.38E-2(3.3E-3)	5.31E-1(5.5E-3)†	5.21E-1(1.4E-3)†	2.23E-1(4.6E-2)†	5.22E-1(2.5E-3)†
WFG2	5	5.56E-3(4.1E-4)	5.23E-3(3.0E-4)	3.11E-3(2.2E-4)†	8.14E-3(1.9E-3)†	4.65E-3(2.5E-4)†
	10	1.09E-2(1.0E-3)	1.81E-2(2.1E-3)†	9.22E-3(7.8E-4)†	1.41E-2(1.0E-3)†	1.23E-2(4.8E-4)†
	15	1.33E-2(6.8E-4)	3.61E-2(5.2E-3)†	2.55E-2(3.2E-3)†	2.46E-2(1.6E-3)†	2.33E-2(1.6E-3)†
WFG3	5	9.90E-3(2.3E-4)	2.25E-3(1.7E-4)†	2.02E-3(1.2E-4)†	1.79E-2(5.6E-4)†	1.79E-3(5.5E-5)†
	10	1.18E-2(1.6E-3)	2.61E-3(1.1E-4)†	7.64E-3(8.6E-4)†	1.91E-2(1.6E-4)†	3.29E-3(2.1E-4)†
	15	1.00E-2(1.7E-4)	2.84E-3(5.0E-5)†	5.37E-3(2.5E-4)	1.76E-2(8.6E-5)	3.39E-3(1.2E-4)†
WFG4	5	1.18E-2(8.0E-4)	6.36E-3(2.7E-4)†	5.93E-3(1.2E-4)†	1.92E-2(1.7E-3)†	6.58E-3(3.1E-4)†
	10	1.69E-2(1.8E-3)	7.67E-3(3.9E-4)†	1.10E-2(1.3E-3)†	1.87E-2(4.8E-6)†	6.83E-3(1.2E-4)†
	15	1.26E-2(3.5E-4)	8.28E-3(2.6E-4)†	1.03E-2(7.8E-4†)	1.83E-2(1.5E-5)†	8.19E-3(1.0E-4)†
WFG5	5	4.50E-3(7.8E-4)	6.15E-3(1.9E-4)†	5.07E-3(2.4E-4)†	4.51E-3(1.2E-4)†	7.88E-3(1.7E-4)†
	10	1.36E-2(3.9E-4)	6.56E-3(2.3E-4)	1.29E-2(2.0E-4)	1.60E-2(3.5E-6)†	7.00E-3(1.4E-4)†
	15	6.51E-3(1.1E-3)	7.35E-3(2.8E-4)†	9.90E-3(5.1E-4)	1.62E-2(1.7E-4)†	7.62E-3(1.2E-4)†
WFG6	5	5.38E-3(9.2E-4)	7.31E-3(4.6E-4)†	5.45E-3(1.7E-4)†	5.12E-3(1.1E-4)†	6.91E-3(2.4E-4)†
	10	1.63E-2(1.9E-3)	8.60E-3(2.8E-4)†	1.15E-2(1.3E-3)†	2.11E-2(6.2E-5)†	6.84E-3(1.1E-4)†
	15	1.58E-2(9.0E-4)	9.05E-3(2.2E-4)†	1.02E-2(3.5E-4)†	2.04E-2(5.8E-5)†	7.40E-3(3.3E-4)†
WFG7	5	7.18E-3(8.3E-4)	8.10E-3(2.7E-4)†	6.33E-3(1.5E-4)†	8.83E-3(4.9E-3)†	8.72E-3(2.7E-4)†
	10	1.55E-2(1.4E-3)	2.55E-2(3.2E-4)†	1.60E-2(5.3E-4)†	2.03E-2(1.3E-4)†	2.57E-2(5.3E-4)†
	15	1.30E-2(1.3E-3)	2.27E-2(1.6E-4)†	1.18E-2(4.1E-4)†	1.88E-2(1.3E-4)†	2.32E-2(2.0E-4)†
WFG8	5	8.51E-3(1.3E-3)	8.88E-3(2.6E-4)†	8.80E-3(3.3E-4)	1.45E-2(6.3E-3)†	9.56E-3(3.7E-4)†
	10	9.60E-3(7.7E-4)	7.35E-3(1.2E-4)†	1.51E-2(5.5E-4)†	1.94E-2(2.9E-6)†	7.27E-3(1.5E-4)†
	15	1.17E-2(5.7E-4)	2.56E-2(1.5E-4)	1.12E-2(5.8E-4)†	1.89E-2(2.0E-4)	2.15E-2(1.1E-4)†
WFG9	5	5.95E-3(7.6E-4)	6.87E-3(3.4E-4)†	4.90E-3(1.4E-4)†	4.18E-3(9.0E-5)†	7.95E-3(3.4E-4)†
	10	1.19E-2(1.9E-3)	2.77E-2(1.7E-4)†	1.21E-2(2.8E-4)†	1.59E-2(5.6E-6)†	2.08E-2(2.6E-4)†
	15	1.25E-2(1.7E-3)	2.41E-2(1.4E-4)†	1.29E-2(5.7E-4)†	1.91E-2(1.1E-4)†	2.39E-2(2.6E-4)†

the best on 10-, 15-objectives WFG3. IBEA has the best performances on 5-objectives WFG6 and WFG9. And SMPSO has the best values on 5-objectives WFG3 and 10-, 15-objectives WFG4, and 10-objectives WFG8.

In order to avoid the influence of extreme data and exactly reflect the general level of all the experimental results, five MOEAs were run 30 times independently. The median and interquartile range (IQR) values of IGD on DTLZ and WFG are shown in Table 6 and Table 7.

As we can see in the Table 6 and Table 7, ANS-MOEA performs well on

Table 6: The median and IQR of IGD for the five algorithms on the DTLZ suites.

Problem	Obj.	ANS-MOEA	GDE3	MOEAD	IBEA	SMPSO
DTLZ1	3	1.19E-3(9.3E-5)	1.42E-3(1.0E-4)	2.21E-3(1.3E-4)	1.14E-2(1.8E-3)	1.68E-3(3.6E-3)
	5	2.82E-3(1.9E-4)	4.02E-2(5.8E-2)	2.29E-3(7.1E-4)	9.41E-3(2.4E-3)	1.18E-2(4.6E-1)
	8	4.09E-3(2.0E-4)	2.62E-2(3.8E-3)	4.30E-3(1.5E-3)	9.13E-3(2.2E-4)	3.32E-2(6.9E-3)
	10	8.51E-3(9.7E-4)	6.24E-2(1.1E-2)	2.60E-2(1.3E-2)	2.01E-2(8.7E-4)	3.67E-2(2.9E-3)
	15	5.13E-3(4.0E-4)	2.65E-2(6.9E-3)	3.19E-3(1.7E-4)	7.87E-3(1.4E-2)	2.37E-2(3.5E-4)
DTLZ2	3	1.08E-3(3.5E-5)	1.39E-3(4.0E-5)	2.62E-3(1.9E-5)	2.18E-3(5.4E-5)	1.58E-3(1.4E-4)
	5	2.53E-3(7.0E-5)	4.78E-3(2.2E-4)	3.10E-3(2.1E-4)	3.23E-3(1.6E-4)	7.20E-3(9.6E-4)
	8	3.68E-3(3.1E-4)	9.67E-3(1.1E-3)	4.69E-3(8.6E-4)	7.34E-3(8.7E-4)	7.81E-3(8.3E-4)
	10	4.18E-3(1.2E-3)	1.19E-2(2.0E-3)	4.97E-3(1.3E-3)	7.52E-3(5.8E-4)	9.09E-3(1.0E-3)
	15	6.92E-3(5.8E-4)	9.06E-3(1.1E-3)	6.91E-3(9.7E-5)	7.78E-3(3.0E-4)	9.14E-3(6.7E-4)
DTLZ3	3	1.29E-3(5.6E-5)	1.46E-3(4.5E-5)	2.75E-3(5.1E-5)	1.19E-2(5.0E-4)	1.90E-3(5.9E-3)
	5	2.46E-3(1.9E-4)	3.64E-3(3.6E-4)	3.50E-3(3.5E-3)	1.06E-2(5.4E-3)	2.61E-2(1.2E-2)
	8	4.54E-3(8.9E-5)	8.89E-3(3.0E-4)	6.19E-3(9.3E-4)	9.01E-3(2.0E-6)	8.44E-3(8.0E-4)
	10	5.34E-3(1.1E-5)	6.54E-3(1.1E-3)	5.78E-3(2.2E-4)	5.95E-3(3.2E-6)	4.35E-3(8.7E-4)
	15	5.62E-3(3.9E-6)	6.11E-3(1.4E-3)	5.63E-3(5.6E-7)	5.63E-3(1.7E-6)	3.81E-3(2.3E-4)
DTLZ4	3	1.23E-2(1.7E-4)	1.44E-3(9.2E-5)	2.74E-3(9.1E-4)	1.23E-2(3.1E-4)	1.66E-3(5.9E-4)
	5	2.21E-3(7.5E-5)	3.70E-3(2.0E-4)	2.92E-3(4.4E-4)	5.07E-3(3.2E-3)	4.14E-3(1.2E-4)
	8	4.57E-3(8.1E-4)	6.21E-3(4.4E-4)	4.28E-3(7.5E-5)	4.63E-3(1.3E-3)	5.35E-3(2.6E-4)
	10	4.96E-3(3.0E-3)	7.91E-3(4.7E-4)	4.79E-3(2.9E-5)	4.87E-3(4.7E-4)	6.55E-3(8.1E-4)
	15	5.56E-3(1.8E-7)	7.80E-3(3.3E-4)	5.82E-3(2.1E-5)	6.23E-3(4.1E-4)	7.11E-3(8.6E-4)
DTLZ5	3	1.39E-4(1.1E-5)	1.10E-4(1.4E-6)	3.45E-4(4.6E-7)	8.14E-4(5.2E-5)	1.08E-4(4.7E-6)
	5	1.81E-3(3.6E-4)	2.46E-3(2.3E-4)	4.41E-3(1.2E-3)	6.54E-3(3.0E-4)	2.31E-3(3.1E-4)
	8	3.46E-3(1.9E-4)	3.69E-3(1.7E-4)	6.23E-3(1.4E-4)	8.45E-3(3.4E-3)	3.31E-3(2.2E-4)
	10	4.52E-3(2.8E-4)	3.70E-3(3.9E-4)	7.63E-3(3.1E-4)	1.07E-2(3.9E-3)	3.28E-3(1.6E-4)
	15	2.24E-3(1.3E-4)	3.79E-3(6.0E-4)	7.56E-3(2.5E-5)	1.14E-2(2.3E-3)	3.23E-3(1.2E-4)
DTLZ6	3	3.71E-4(3.6E-4)	9.21E-5(3.4E-6)	2.70E-4(1.1E-7)	1.33E-3(7.1E-4)	8.95E-5(3.7E-6)
	5	5.41E-3(2.3E-4)	2.26E-3(2.4E-4)	5.01E-3(1.4E-4)	7.11E-3(1.8E-3)	1.92E-3(1.4E-4)
	8	7.15E-3(1.0E-4)	7.51E-3(1.0E-3)	7.32E-3(1.5E-4)	8.27E-3(6.5E-5)	3.52E-3(5.6E-4)
	10	3.92E-3(5.0E-4)	8.05E-3(9.6E-4)	7.15E-3(9.7E-5)	8.40E-3(5.4E-5)	3.57E-3(6.7E-4)
	15	2.84E-3(2.2E-4)	7.17E-3(1.4E-3)	7.71E-3(1.3E-4)	8.24E-3(7.7E-5)	3.79E-3(3.7E-4)
DTLZ7	3	1.21E-3(1.6E-4)	8.60E-4(9.5E-5)	1.70E-3(2.3E-5)	9.50E-3(4.2E-3)	9.89E-4(1.8E-4)
	5	4.63E-3(8.6E-5)	3.44E-3(1.6E-4)	4.76E-3(3.0E-3)	1.92E-2(7.3E-4)	4.67E-3(5.2E-4)
	8	1.51E-2(6.7E-3)	6.50E-3(2.8E-4)	1.14E-2(2.6E-3)	2.71E-2(2.9E-4)	1.47E-2(7.2E-3)
	10	2.30E-2(3.6E-3)	1.23E-2(2.3E-3)	1.48E-2(2.6E-3)	3.28E-2(3.2E-4)	1.74E-2(4.1E-3)
	15	2.91E-2(2.9E-3)	1.83E-2(3.1E-3)	2.36E-2(5.5E-3)	3.97E-2(1.0E-4)	1.82E-2(1.3E-3)

most problems. It can be concluded that the IGD values on DTLZ and WFG, which ANS-MOEA obtains the results through the 30 times repeat and independent experiments, are relatively stable. In order to respond to the median and IQR directly, the 5-, 10-, 15-dimension boxplots for DTLZ2 and WFG9

Table 7: The median and IQR of IGD for the five algorithms on the WFG suites.

Problem	Obj.	ANS-MOEA	GDE3	MOEAD	IBEA	SMPSO
WFG1	5	4.94E-3(4.1E-4)	1.16E-1(1.5E-3)	1.12E-1(1.1E-3)	2.60E-2(1.3E-2)	1.16E-1(1.6E-3)
	10	8.99E-3(2.6E-3)	3.70E-1(3.9E-3)	1.72E-1(2.4E-2)	8.14E-2(3.7E-2)	3.67E-1(2.0E-3)
	15	1.26E-2(2.0E-3)	5.30E-1(9.7E-3)	5.21E-1(7.1E-4)	2.32E-1(6.2E-2)	5.22E-1(4.2E-3)
WFG2	5	5.60E-3(6.0E-4)	5.23E-3(3.8E-4)	3.09E-3(4.3E-4)	8.15E-3(3.4E-3)	4.59E-3(2.1E-4)
	10	1.09E-2(1.9E-3)	1.81E-2(1.5E-3)	9.22E-3(1.3E-3)	1.45E-2(2.0E-3)	1.22E-2(6.0E-4)
	15	1.34E-2(1.6E-3)	3.48E-2(1.1E-2)	2.54E-2(5.6E-3)	2.37E-2(2.2E-3)	2.35E-2(2.2E-3)
WFG3	5	6.00E-3(3.8E-4)	2.21E-3(2.3E-4)	2.01E-3(2.2E-4)	1.81E-2(1.9E-4)	1.79E-3(7.5E-5)
	10	6.10E-3(2.9E-4)	2.59E-3(1.5E-4)	7.63E-3(1.3E-3)	1.91E-2(2.9E-4)	3.30E-3(4.9E-4)
	15	6.01E-3(2.4E-4)	2.83E-3(1.0E-4)	5.34E-3(1.4E-4)	1.76E-2(1.5E-4)	3.39E-3(2.1E-4)
WFG4	5	1.19E-3(8.9E-4)	6.29E-3(3.4E-4)	5.94E-3(1.3E-4)	2.06E-2(3.5E-3)	6.57E-3(2.1E-4)
	10	1.73E-3(9.1E-4)	7.80E-3(6.3E-4)	1.08E-2(2.7E-3)	1.87E-2(9.0E-6)	6.85E-3(1.8E-4)
	15	1.26E-3(3.9E-4)	8.40E-3(4.5E-4)	9.91E-3(1.8E-3)	1.83E-2(2.6E-6)	8.19E-3(1.7E-4)
WFG5	5	7.67E-3(4.5E-4)	6.15E-3(3.3E-4)	5.02E-3(4.9E-4)	4.52E-3(2.3E-4)	7.85E-3(2.1E-4)
	10	1.36E-2(3.9E-4)	6.57E-3(2.3E-4)	1.29E-2(2.5E-4)	1.60E-2(4.0E-6)	7.02E-3(3.1E-4)
	15	1.52E-2(1.6E-3)	7.27E-3(2.0E-4)	9.92E-3(8.5E-4)	1.61E-2(1.1E-4)	7.61E-3(2.0E-4)
WFG6	5	8.63E-3(1.0E-3)	7.24E-3(4.1E-4)	5.42E-3(1.6E-4)	5.12E-3(1.4E-4)	6.89E-3(2.8E-4)
	10	1.56E-2(6.5E-4)	8.55E-3(3.4E-4)	1.10E-2(1.4E-3)	2.11E-2(4.0E-5)	6.87E-3(1.6E-4)
	15	6.56E-3(2.4E-4)	9.11E-3(1.7E-4)	1.00E-2(6.2E-4)	2.04E-2(5.6E-5)	7.27E-3(9.6E-5)
WFG7	5	1.18E-3(1.4E-3)	8.09E-3(3.4E-4)	6.27E-3(3.0E-4)	5.54E-3(6.2E-3)	8.65E-3(2.6E-4)
	10	1.51E-3(2.3E-3)	8.57E-3(5.7E-4)	1.61E-2(7.1E-4)	2.02E-2(1.6E-6)	8.51E-3(5.7E-4)
	15	1.26E-3(2.5E-3)	8.27E-3(1.1E-4)	1.19E-2(3.3E-4)	1.87E-2(1.3E-6)	8.31E-3(2.7E-4)
WFG8	5	8.32E-3(9.6E-4)	8.82E-3(3.0E-4)	8.81E-3(5.4E-4)	1.72E-2(1.4E-2)	9.70E-3(6.4E-4)
	10	9.17E-3(1.5E-3)	7.32E-3(1.9E-4)	1.50E-2(4.2E-4)	1.94E-2(2.0E-6)	7.28E-3(1.8E-4)
	15	1.16E-2(1.8E-4)	8.55E-3(2.4E-4)	1.14E-2(8.6E-4)	1.88E-2(1.8E-4)	8.16E-3(1.9E-4)
WFG9	5	1.71E-3(9.2E-4)	5.95E-3(5.3E-4)	3.89E-3(2.2E-4)	3.22E-3(1.3E-4)	7.94E-3(4.4E-4)
	10	1.20E-2(2.6E-3)	1.17E-2(2.8E-4)	1.21E-2(5.2E-4)	1.59E-2(5.7E-6)	7.12E-2(4.5E-4)
	15	1.24E-2(1.7E-3)	1.19E-2(1.2E-4)	1.31E-2(7.2E-4)	1.90E-2(3.0E-5)	1.42E-2(4.3E-4)

were presented in Figure 3 and Figure 4.

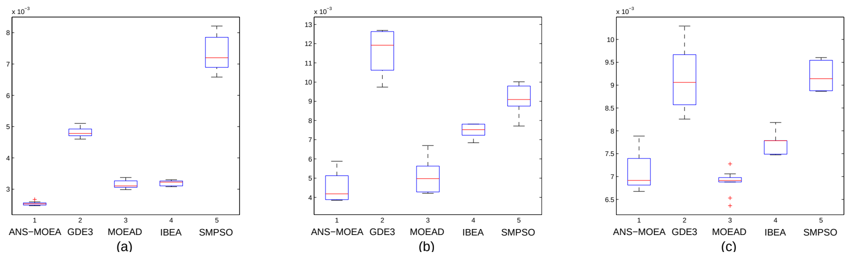

Figure 3: (a), (b), and (c) describe the boxplots of IGD value on 5-, 10-, 15-dimension DTLZ2 respectively.

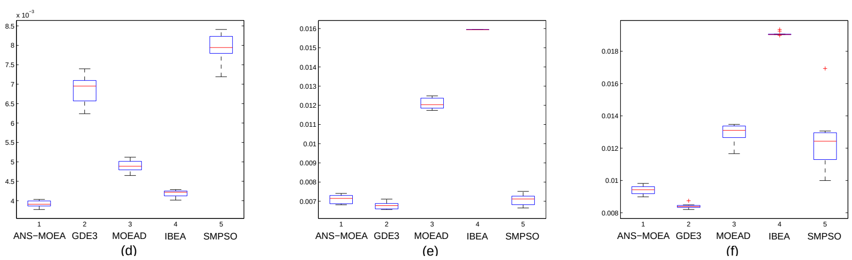

Figure 4: (d), (e), and (f) describe the boxplots of IGD value on 5-, 10-, 15-dimension WFG9 respectively.

In the boxplots, the size of rectangle indicates the IQR and the line in the rectangle represents the median. Take (a) in the Figure 3 as the case. It can be seen that the first rectangle, which represents the ANS-MOEA, has the lowest location in the figure. This indicates the overall data samples ANS-MOEA provide is superior than the other four state-of-the-art MOEAs. In addition, ANS-MOEA also have the smallest rectangle and IBEA has the second smallest. It denotes that the data samples of ANS-MOEA have the smaller otherness.

In addition, the distribution of the final solution sets in Pareto front are described in Figure 5. There are seveal points deserve elaboration:

1. As we can see in the Figure 5, ANS-MOEA can approximate the Pareto front more easily compared with the other MOEAs.
2. In the second column and third column of the Figure 5, which plot the

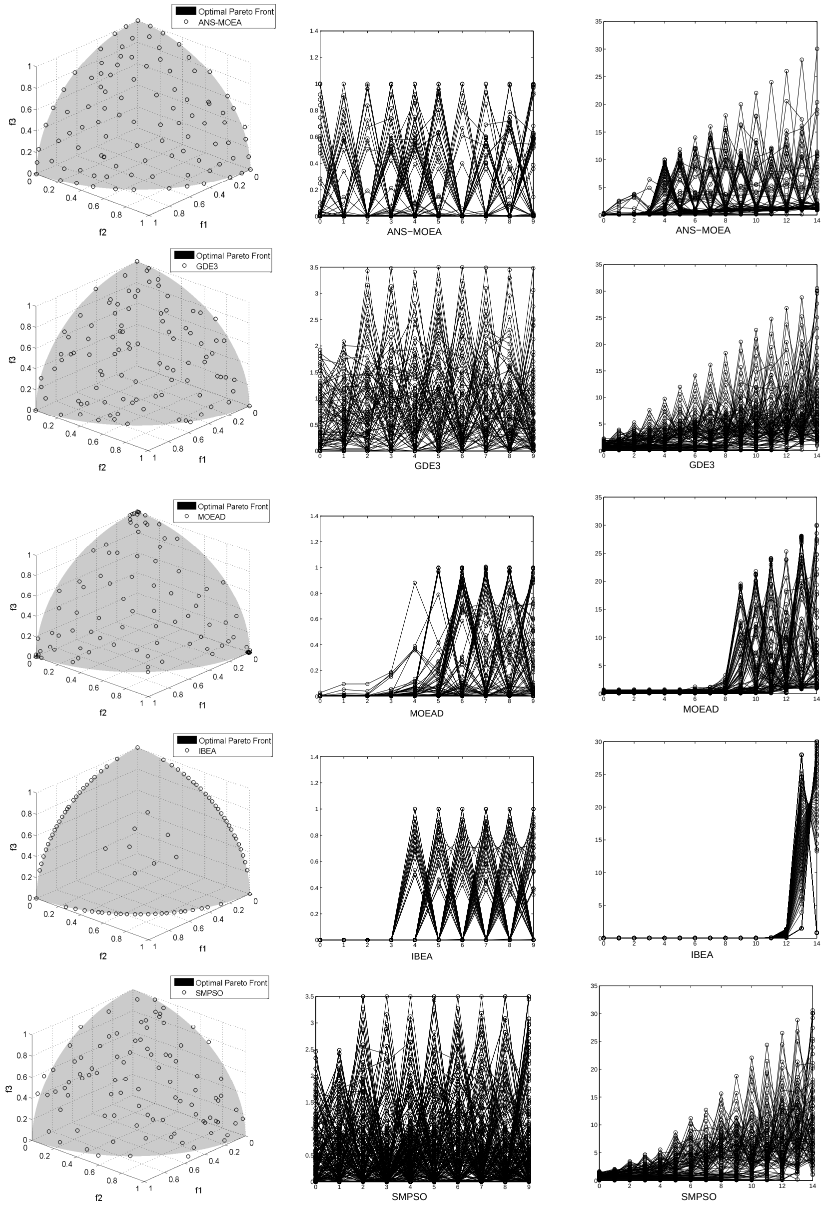

Figure 5: Distribution of final solution set on 3-, 10-objectives DTLZ2 and 15-objectives WFG9 for ANS-MOEA, GDE3, MOEAD, IBEA, SMPSO respectively.

10-objectives DTLZ2 and 15-objectives WFG9, GDE3 and SMPSO have the best performance on the WFG9, but they are non-convergent on the

DTLZ2. ANS-MOEA has the well-converged performance on the DTLZ2 and WFG9, but it has some sparse points on several objectives. The first reason we consider that it is affected by the distribution of the well-converged points and the second reason is that the size of neighborhood is constantly changing. Furthermore, it is one of the future work to be resolved.

5. Conclusions

This paper has presented an adaptive neighborhood selection strategy on critical-layer population to balance the convergence and distribution in many-objective [43] evolutionary optimization. The proposed algorithm has been used to judge the ability of convergence and distribution by CI and DI, which are called convergence information and distribution information, respectively. The adaptive neighborhood selection strategy adjusts the size of neighborhood for adapting to the different population densities on the critical layer. When the well-converged individual was selected into the archive, the NC, which is called the neighborhood collection, reserves its neighbor in order to avoid the over-close individuals were selected into archive and relatively better distribution individuals was selected into critical-layer population by competition.

Simulation experiments have been researched by providing several detailed comparisons with four other state-of-the-art evolutionary multiobjective algorithms (GDE3, MOEAD, IBEA, SMPSO). In the simulation experiments, DTLZ and WFG were selected to validate the ability of these five algorithms, and GD, IGD as the indicators to assess the performance intuitively. As the experimental results show, these five algorithms have their own superior performance. The results reflect that the proposed method has good performance on most problems.

Although the proposed method has superior performance on most problems, it also has some drawbacks. For example, as seen in the Table 4, ANS-MOEA has the worst performance on the biased and degenerate problem in cases of high-dimension. One major future work is to further investigate the ANS-MOEA in more multiobjective problems with different characteristics [44] and some real-world problems [45, 46].

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[2] K. Deb, M. Mohan, and S. Mishra, "Evaluating the -domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions," *Evolutionary Computation*, vol. 13, no. 4, pp. 501–525, 2005.

[3] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[4] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," 2010.

[5] M. L. Wong and G. Cui, "Data mining using parallel multi-objective evolutionary algorithms on graphics hardware," pp. 1–8, 2010.

[6] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "Survey of multiobjective evolutionary algorithms for data mining: Part ii," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 20–35, 2014.

[7] P. C. Chang, S. H. Chen, Q. Zhang, and J. L. Lin, "Moea/d for flowshop scheduling problems," in *Evolutionary Computation*, pp. 1433–1438, 2008.

[8] A. Konstantinidis and K. Yang, "Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific moea/d," *Applied Soft Computing*, vol. 11, no. 7, pp. 4117–4134, 2011.

[9] A. Konstantinidis, C. Charalambous, A. Zhou, and Q. Zhang, "Multi-objective mobile agent-based sensor network routing using moea/d," pp. 1–8, 2010.

[10] M. Vecchio, R. Lopez-Valcarce, and F. Marcelloni, "A study on the application of different two-objective evolutionary algorithms to the node localization problem in wireless sensor networks," pp. 1008–1013, 2011.

[11] E. J. S. Pires, P. B. D. M. Oliveira, and J. A. T. Machado, "Manipulator trajectory planning using a moea," *Applied Soft Computing*, vol. 7, no. 3, pp. 659–667, 2007.

[12] K. Akama, T. Matsuda, S. Takeda, K. Sakai, A. Nagasawa, and J. Kosaka, *MOEA/D for a Tri-objective Vehicle Routing Problem.* Springer Berlin Heidelberg, 2013.

[13] K. C. Tan, T. H. Lee, Y. H. Chew, and L. H. Lee, "A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems," in *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, pp. 855–885, 2006.

[14] M. A. Abido, "Multiobjective evolutionary algorithms for electric power dispatch problem," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 315–329, 2006.

[15] L. H. Lee, E. P. Chew, S. Teng, and Y. Chen, "Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem," *European Journal of Operational Research*, vol. 189, no. 2, pp. 476–491, 2008.

[16] K. C. Tan and Y. Li, "Automating control system design via a multiobjective evolutionary algorithm," *Intelligent Systems*, 2004.

[17] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," *Lecture Notes in Computer Science*, vol. 3410, pp. 14–32, 2005.

[18] J. G. Herrero, A. Berlanga, and J. M. M. Lopez, "Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters," *Evolutionary Computation IEEE Transactions on*, vol. 13, no. 1, pp. 151–168, 2009.

[19] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit.," *Evolutionary Computation IEEE Transactions on*, vol. 10, no. 5, pp. 477–506, 2006.

[20] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–95, 2000.

[21] F. Kursawe, *A variant of evolution strategies for vector optimization.* Springer Berlin Heidelberg, 1999.

[22] D. A. V. Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a pareto front," *Stanford University California*, pp. 221–228, 1999.

[23] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

[24] J. J. Durillo and A. J. Nebro, "jmetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, 2011.

[25] M. Li, S. Yang, and X. Liu, "Bi-goal evolution for many-objective optimization problems," *Artificial Intelligence*, vol. 228, no. C, pp. 45–65, 2015.

[26] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *Acm Computing Surveys*, vol. 48, no. 1, pp. 1–35, 2015.

[27] H. Ishibuchi, T. Doi, and Y. Nojima, "Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms.," in

Parallel Problem Solving From Nature - PPSN Ix, International Conference, Reykjavik, Iceland, September 9-13, 2006, Procedings, pp. 493–502, 2006.

[28] E. Zitzler and S. Knzli, "Indicator-based selection in multiobjective search," *Lecture Notes in Computer Science*, vol. 3242, pp. 832–842, 2015.

[29] H. Sato, H. E. Aguirre, and K. Tanaka, *Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs.* Springer Berlin Heidelberg, 2007.

[30] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.

[31] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[32] J. Molina, L. V. Santana, A. G. Hernndez-Daz, C. A. C. Coello, and R. Caballero, "g -dominance: Reference point based dominance for multiobjective metaheuristics," *European Journal of Operational Research*, vol. 197, no. 2, pp. 685–692, 2009.

[33] L. Ben Said, S. Bechikh, and K. Ghedira, "The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making," *Evolutionary Computation IEEE Transactions on*, vol. 14, no. 5, pp. 801–818, 2010.

[34] A. Sinha, D. K. Saxena, K. Deb, and A. Tiwari, "Using objective reduction and interactive procedure to handle many-objective optimization problems," *Applied Soft Computing*, vol. 13, no. 1, pp. 415–427, 2013.

[35] D. K. Saxena, J. A. Duro, A. Tiwari, and K. Deb, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, 2013.

[36] S. Kukkonen and J. Lampinen, "Gde3: the third evolution step of generalized differential evolution," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, pp. 443–450 Vol.1, 2005.

[37] A. C. Godinez, L. E. M. Espinosa, and E. M. Montes, "An experimental comparison of multiobjective algorithms: Nsga-ii and omopso," in *Electronics, Robotics and Automotive Mechanics Conference*, pp. 28–33, 2010.

[38] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, and C. A. C. Coello, "Smpso: A new pso-based metaheuristic for multi-objective optimization," pp. 66–73, 2009.

[39] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 3, pp. 115–148, 2000.

[40] K. Deb and A. Kumar, "Real-coded genetic algorithms with simulated binary crossover: Studies on multi-modal and multi-objective problems. complex systems 9(6), 431-454," in *Complex Systems*, 1995.

[41] A. J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J. J. Durillo, and A. Beham, "Abyss: Adapting scatter search to multiobjective optimization," *Evolutionary Computation IEEE Transactions on*, vol. 12, no. 4, pp. 439–457, 2008.

[42] J. Bader, K. Deb, and E. Zitzler, *Faster Hypervolume-Based Search Using Monte Carlo Sampling*. Springer Berlin Heidelberg, 2010.

[43] E. J. Hughes, "Evolutionary many-objective optimisation: Many once or one many?," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, pp. 222–227 Vol.1, 2005.

[44] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[45] H. K. Singh, A. Isaacs, and T. Ray, "A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *Evolutionary Computation IEEE Transactions on*, vol. 15, no. 4, pp. 539–556, 2011.

[46] Corne, W. David, and Knowles, "Techniques for highly multiobjective optimisation: some nondominated points are better than others," *Proceedings Gecco Acm*, pp. 773–780, 2009.

[47] Zheng J, Yu G, Zhu Q, et al. On decomposition methods in interactive user-preference based optimization[J]. Applied Soft Computing, 2016.

[48] Yu G, Zheng J, Shen R, et al. Decomposing the user-preference in multiobjective optimization[J]. Soft Computing, 2015: 1-17.