An adaptive reference point technique to improve the quality of decomposition based multi-objective evolutionary algorithm

Tran Binh Minh[1*], Nguyen Long[2], Thai Trung Kien[1]

[1]Institue of Information Technology, Academy of Military Science and Technology;
[2]National Defense Academy.
[*]Corresponding author: minhchip79@gmail.com

Received 12 Oct. 2023; Revised 10 Dec. 2023; Accepted 12 Dec. 2023; Published 30 Dec. 2023.
DOI: https://doi.org/10.54939/1859-1043.j.mst.CSCE7.2023.3-14

ABSTRACT

Applying multi-objective evolutionary optimization algorithms in solving multi-objective optimization problems is a research field that has received attention recently. In the literature of this research field, many studies have been carried out to propose multi-objective evolutionary algorithms or improve published algorithms. However, balancing the exploitation and exploration capabilities of the algorithm during the evolution process is still challenging. This article proposes an approach to solve that equilibrium problem based on analyzing population distribution during the evolutionary process to identify empty regions in which no solutions are selected. After that, information about empty regions with the most significant area will be combined with the current reference point to create a new reference point to prioritize choosing solutions in those regions. Experiments on 10 test problems of 2 typical benchmark sets showed that this mechanism increases the diversity of the population, thereby contributing to a balance between the algorithm's abilities in the evolutionary process and enhancing the algorithm.

Keywords: Evolutionary multi-objective optimization; Balance of exploration and exploitation; Population distribution; Empty region; Adaptive reference point; MOEA/D.

1. INTRODUCTION

The need to solve classes of optimization problems is enormous in many fields. Optimization problems with conflicting objectives are called multi-objective optimization problems (MOPs). When solving the MOPs, it is necessary to solve the objectives simultaneously, and the result of MOPs is a set of compromise solutions. The set of optimal solutions in the decision space is called the Pareto optimal set (PS), and the set of points mapped from the PS to the objective space forms the Pareto optimal front (PF). An unconstrained minimization MOP is defined as follows:

$$\begin{gathered} minimize\ \vec{F}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_M(\vec{x})] \\ \vec{x} = [x_1, x_2, \dots, x_v] \in \Omega \end{gathered} \tag{1}$$

In which $\vec{F}$ is the objective function, $\vec{x}$ is the vector of decision variables, M is the number of objectives, Ω is the feasible region in v-dimensional $\mathbb{R}^v$.

Solving MOPs with conventional optimization methods is challenging because of conflicting objectives. Multi-objective evolutionary algorithms (MOEAs) have affirmed their superiority due to using the evolutionary mechanism, working based on populations so that it can solve most MOPs. MOEAs can be classified into the following groups [1]:

(i) Domination-based MOEAs: The Pareto-dominance concept is used to calculate the fitness value to choose more suitable individuals;

(ii) Indicator-based MOEAs: The value of quality metrics are combined with MOEAs to guide the search progress in the desired direction;

(iii) Decomposition-based MOEAs: Divide a MOP into minor problems and optimize them simultaneously and collaboratively by using the neighboring information;

(iv) Improvement direction-based MOEAs: Using the direction of convergence and spread as preference information to product solutions for generations.

(v) Hybrid MOEAs: Combine advantage techniques of different MOEAs to solve complicated MOPs and environmental fluctuations in evolution.

The ***convergence*** and ***diversity*** of the population are core elements to evaluate MOEAs quality [2]. The closer the solutions reach PF, the better the population convergence quality. Similarly, the more widely and evenly distributed solutions along the PF, the higher the diversity quality. To keep the equilibrium of convergence and diversity of the population, MOEAs need mechanisms to search by depth (called ***exploitation*** capacity) to find solutions near the PF and search by breadth (called ***exploration*** capacity) to find solutions widely distributed over the PF. Solving the balance problem between exploration and exploitation is a complicated research issue in the MOEAs research field. If we concentrate on exploitation, the found solutions will be challenging to distribute widely and evenly. On the contrary, the attainable solutions will not be close to the PF if we focus on exploration.

Many works have been done to solve the above research problem, and adaptive techniques such as measurement-based and spatial partitioning techniques are often used.

(i) Measurement-based technique: The basic idea of this technique is to analyze the trend of quality change over a period to determine if the algorithm is skewed towards exploration/exploitation and then control priority in the opposite direction to ensure balance. Some typical studies have been conducted in this direction, such as: In 2011, D. H. Phan *et al* [3] proposed the BIBEA algorithm, in which the selection mechanism is based on combining measures using the AdaBoost algorithm to increase the convergence speed and improving the distribution according to the PF. In 2017, R. H. Gomez *et al* [4] suggested an environmental selection mechanism based on the R2 measure combined with the Riesz energy to improve the diversity of solutions and maintain convergence. In 2018, J. G. Falcon-Cardona *et al* [5] introduced the MIHPS algorithm by proposing a meta-heuristic method based on taking advantage of measures R2, IGD+, €+ and Δp to choose the solution that matches the search state and the values of the measures. In 2020, J. G. Falcon-Cardona *et al* [6] proposed the IBDE algorithm based on a density estimator using a combination of five differential measures.

(ii) Space partitioning technique: This technique often divides the search space into relatively sub-spaces. In each sub-space, the solutions will be assigned a fitness value and selected based on the priority of fitness value from high to low. Sub-space division is typically performed by grid division or using a ray system from one or more reference points. The ray set is often evenly distributed and is responsible for guiding the process of finding solutions in a priority direction. Some typical studies have been conducted in this direction, such as: In 2007, Zh. Qingfu *et al* proposed the MOEA/D algorithm [7] with the main idea of dividing a MOP into sub-problems and optimizing them simultaneously and collaboratively by using the information of neighboring sub-problems. A uniformly distributed weight vector set originating from a reference point is used to maintain the diversity of the population. In 2014, K. Deb *et al* introduced a reference point-based algorithm called NSGA-III [9] in which the best members from the last non-dominated front are selected based on the supplied reference points created by Das and Dennis's procedure. The system of reference lines from the origin passing through the reference points will be used to determine the selected solutions according to diverse priorities. In 2018, A. Masood *et al* suggested the ANSGA-III algorithm [10] to address the issue that some reference points are never associated with any of the Pareto-optimal solutions in NSGA-III by using the reference points adaptation mechanism.

In MOEAs using space partitioning technique, especially in decomposition-based MOEAs, the reference point, which serves as the original point of the division vector set, plays a significant role. However, their effect on the balance of exploration and exploitation has yet to be studied well. In 2017, W. Rui *et al* [11] studied the effect of the reference point in the MOEA/D algorithm and suggested three methods to approximate the idea point based on offset values called optimistic, pessimistic, and dynamic methods, respectively (the optimistic method uses significant deviation; the pessimistic method uses slight deviation; the dynamic method uses gradually decreases deviation value from a start value to a stop value). In 2019, J. Zou *et al* [12] suggested an adaptation reference point-based algorithm called ARMA, in which the population was partitioned into *N* sub-populations using the approximation of proportion and angles, and the selection strategy is implemented separately in each sub-population.

To the best of our knowledge, few studies use population distribution information to control the exploration and exploitation capabilities. In addition, no studies have adjusted the reference point (RP) based on empty regions in the population distribution. This research was conducted to clarify empty regions' role and find an adaptive mechanism to change RP. The main contributions of this proposal include: (i) Propose an adaptive mechanism to adjust RP based on the population distribution, which directs the algorithm to search for regions that do not have a selected solution; (ii) Apply the proposed mechanism to enhance the balance between exploitation and exploration in the MOEA/D algorithm, thereby enhancing the population's quality.

The remainder of this proposal is organized as follows. Section 2 analyzes the empty regions and gives some background knowledge of the MOEA/D algorithm. Section 3 illustrates our methodology to adjust RP adaptively. Experiments, results, and discussion are introduced in section 4. The conclusion and future work are given in section 5.

2. PROBLEM

2.1. Empty region in MOEA

In the evolutionary process, depending on each MOP, empty regions formed by generation will appear (excluding problems with discrete PF). The lack of solutions in the empty regions leads to a decrease in diversity because the solutions are then concentrated in some areas, leading to an unevenly distributed population along the PF. In the previous study [14], we analyzed the role of empty regions in the distribution of the solution set on the balance between the explorative and exploitative capabilities of MOEAs. If instructed to prioritize solutions in or near those areas, the algorithm will improve diversity quality. Obviously, there are many empty regions, especially in the early stages of evolution, so we should only focus on a set of the largest empty regions.

The demonstration of empty regions in two-objective space is shown figure 1.

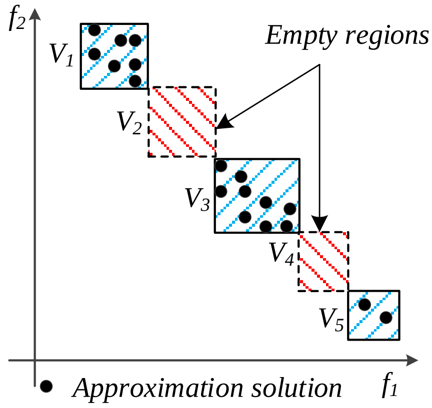


Figure 1. *Demonstration of empty regions in population distribution.*

2.2. The MOEA/D Algorithm

Theoretically, information about empty regions can be applied to enhance different MOEA categories. In this study, we chose the MOEA/D algorithm because it is the representative algorithm for decomposition-based MOEAs.

2.2.1. The MOEA/D algorithm

The MOEA/D algorithm using Tchebycheff scalarizing methods is as follows [7]:

Input: a MOP with a number of objectives is *M*; *N*: the number of sub-problems; $\lambda^1, \dots, \lambda^N$: a uniformly distributed weight vectors set; *T*: the neighborhood size; a stopping criterion.

Ouput: External population *EP* contain**s** solutions of MOP.

Algorithm:

Step 1. Initialization:

Step 1.1: Set EP = ∅.

Step 1.2: Compute the Euclidean distances between any two weight vectors and then work out the T closest weight vectors to each weight vector. For each *i = 1, . . . N*, set $B(i) = \{i_1, \dots, i_T\}$, where $\lambda^{i1}, \dots, \lambda^{iT}$ are the T-closest weight vectors to λ^i.

Step 1.3: Generate an initial population $x^1, \dots, x^N$ randomly or by a problem-specific method. Set $FV^i = F(x^i)$.

Step 1.4: Initialize reference point $z^{min} = (z_1, \dots, z_M)^T$ by a problem specific method.

Step 2. Evolution:

For *i = 1, . . . N*, do

Step 2.1. Reproduction:

Matting selection: Randomly select two indexes *k*, *l* from *B(i)*

Reproduction: Generate a new solution *y* from x^k and x^l by using genetic operators.

Step 2.2. Improvement: Apply a problem-specific repair/improvement heuristic *y* on to produce y'.

Step 2.3. Update of z^{min} by the minimum value of each objective so far

For each *j = 1, . . . , M*

If $z_j > f_j(y')$ then set $z_j = f_j(y')$.

Step 2.4. Update of Neighboring Solutions:

For each index $j \in B(i)$

If $g^{te}(y'|\lambda^j, z) \leq g^{te}(x^j|\lambda^j, z)$ then set $x^j = y'$ and $FV^j = F(y')$.

Step 2.5. Update of EP: Remove from *EP* all the vectors dominated by $F(y')$. Add $F(y')$ to *EP* if no vectors in *EP* dominate it.

Step 3. Stopping Criteria:

If the stopping criteria are satisfied, output EP. Otherwise, go to ***Step 2***.

The Tchebycheff approach for dividing MOP into a number of sub-problems in the form:

$$minimize\ g^{te}(x|\lambda, z) = max_{1<i<M}\{\lambda_i|f_i(x) - z\} \tag{2}$$

Where z is RP and is used as a condition to replace a parent with a child when the child performs better than the parent concerning the j^{th} sub-problem (*Step 2.4*).

2.2.2. The Role of Reference Point in MOEA/D

In the MOEA/D algorithm, the RP could be the ideal point z^{ide} (minimum optimization) or the nadir point z^{nad} (maximum optimization). In practice, when the z^{ide} and z^{nad} are unknown, z^{min} and z^{max} can be used as an alternative solution.

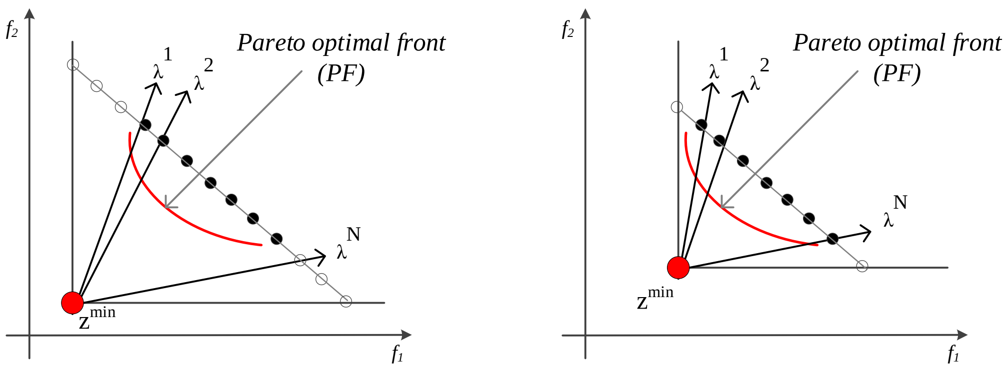
(i) z^{min} denotes the vector that the i^{th} component is the best (minimum) value obtained so far for the i^{th} objective.

$$z^{min} = (z_1^{min}, \dots, z_M^{min})^T$$
$$z_i^{min} = min\{f_i(x)\,|x \in \Omega\}\ for\ each\ i = 1, \dots, M \tag{3}$$

(i) z^{max} denotes the vector that the i^{th} component is the worst (maximum) value obtained so far for the i^{th} objective.

$$z^{max} = (z_1^{max}, \dots, z_M^{max})^T$$
$$z_i^{max} = max\{f_i(x)\,|x \in \Omega\}\ for\ each\ i = 1, \dots, M \tag{4}$$

For ease of interpretation and representation, in the following sections of the paper, the symbol z is used instead z^{min}. The set of weight vectors in the MOEA/D includes vectors whose origin is RP and whose endpoints are uniformly distributed points.



a) Reference point far from the PF　　　b) Reference point near the PF

Figure 2. *Demonstration the role of the reference point.*

RP in the MOEA/D algorithm, to some extent, influences the algorithm's exploitation and exploration performance [8]. It determines the search region of the MOEA/D in the objective space. More clearly, RP controls the proportion of search capacity distributed to explore undiscovered regions or exploit existing regions. Therefore, the location of RP in the objective space is very important. The example illustrated in figure 2 explains the role of RP. In figure 2(a), some weight vectors (e.g. λ^N) do not intersect with the PF, so it is beneficial to enhance exploration performance. On the contrary, in figure 2(b), all weight vectors cross with the PF. Therefore, it is valuable to improve exploitation performance.

3. METHODOLOGY

We hypothesize that if we combine the use of information about empty regions in solution distribution to adjust RP, thereby directing the weight vectors to these regions to prioritize the selection of solutions for the population, we will guarantee the balance between exploration and exploitation capacities and lead to better quality in convergency and diversity of population. In the enhanced algorithm of MOEA/D (called MOEA/D++), RP's position will be adaptively adjusted based on information about empty regions. Changing the position of RP appropriately contributes to the balance between exploration and exploitation capabilities.

Our specific method is as follows:

Step 1. Identify empty regions:

- *Step 1.1*: Detect and select n_{Ept} largest area regions based on the method mentioned in our previous study [14]. If no empty regions are found, or the areas of empty regions are too small (this often occurs in later generations of the evolutionary process), run as the original MOEA/D algorithm; otherwise, carry out the adjustment process.

Pseudocode of the identify empty regions algorithm (called *FindEmptyRegion*):

Input: *P:* population with size *N*; *M*: number of objectives; *r*: minimum edge size; n_{Ept}: number of empty regions to be taken.

Output: Set A consists of large empty regions in population distribution.

Algorithm:

1: *P'* ← *Sort(P); B* ← ∅.
2: for *i* = 2 *to N*
3: *isEmptyRegion* ← 1; *V* ← 1;
4: for *j* = 1 *to M*
5: $d[j] = Abs(P'[i].f_j - P'[i-1].f_j)$. /* j^{th} edge */
6: if $d[j] < r$ then *isEmptyRegion* ← 0; break.
7: else $V = V * Abs(P'[i].f_j - P'[i-1].f_j)$.
8: if (*isEmptyRegion* = 1) then
9: $Center_p \leftarrow (P'[i] - P'[i-1])/2$; $B[|B| + 1] \leftarrow \{P'[i-1], P'[i], Center_p, V\}$.
10: *A* ← *Sort(B,V)* /* *Sort by area in descending order* */
11: if $|A| > n_{Ept}$ then
12: for $i = |A|$ *downto* $n_{Ept} + 1$
13: *Remove(A[i])*. /* *Take* n_{Ept} *of the largest regions* */

- *Step 1.2*: Compute the bounding region center (denoted *C*) created by empty regions.

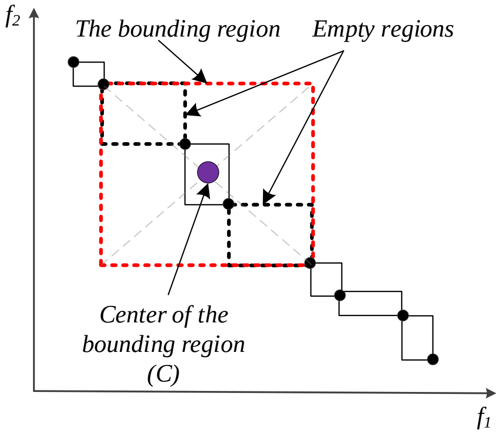


Figure 3. *Center of the bounding region.*

Step 2. Adjust the RP position to guide the algorithm to focus on empty regions:

This step will adjust the RP position to focus the weight vectors on the bounding box and lead to concentration on empty regions to prioritize the selection of solutions.

- Move the current RP (denoted z_{cur}) along the vector (z_{cur}, C) in the opposite direction to C with step-length value k to create a new RP (denoted z_{new}) by the fomula (5).

$$z_{new} = z_{cur} - (C - z_{cur})x\ k \tag{5}$$

Moving RP in the vector (z_{cur}, C) direction focuses the weight vector set into empty regions while moving RP in the opposite direction to the bounding region center to increase the ability to explore spatial regions. The displacement also adapts to the distance from the bounding region center and the current RP. This shows that significant adjustments are needed in the early stages of the evolution process when solutions are far from the PF. In the final stages, only minor adjustments are needed when solutions are close to the PF. The illustration of the proposed method is presented in figure 4.

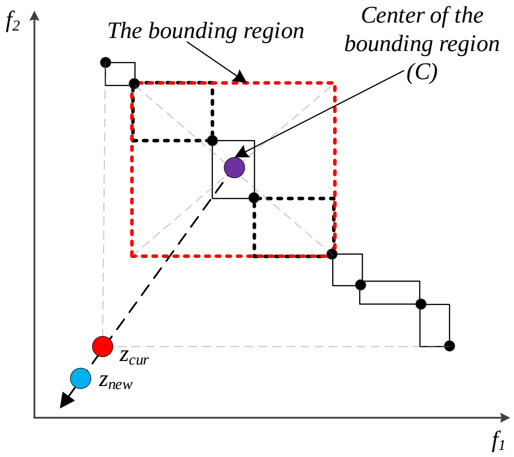


Figure 4. *Illustration of creating the new RP.*

Pseudocode of the MOEA/D++ algorithm is as follows:

Input: A MOP with a number of objectives is M; N: The number of sub-problems; $\lambda^1, \ldots, \lambda^N$: A uniformly distributed weight vectors set; T: The neighborhood size; a stopping criterion; *r*: Minimum edge size; n_{Ept}: Number of empty regions to be taken.

Output: External population *EP* contain**s** solutions of MOP.

Algorithm:

Step 1: Initialization (same as MOEA/D)

Step 2. Evolution

For $i = 1, . . . N$, do

Step 2.1. Reproduction (same as MOEA/D)

Step 2.2. Improvement (same as MOEA/D)

Step 2.3. Update of z^{min} by the minimum value of each objective so far

For each $j = 1, . . . , M$

if $z_j > f_j(y')$ then set $z_j = f_j(y')$.

X ← *FindEmptyRegion. /* Identify empty regions */*

***if* |X| > 0.**

C ← *Center(X) /* Center of bounding box */*

$\boldsymbol{z^{min} \leftarrow z^{min} - (C - z^{min}) \times k}$.

Step 2.4. Update of Neighboring Solutions (same as MOEA/D).

Step 2.5. Update of EP (same as MOEA/D).

Step 3. Stopping Criteria (same as MOEA/D).

The adjustment is executed in Step 2.3 in the primary MOEA/D algorithm, directly impacts Step 2.5 of the current iteration, and indirectly impacts Step 2.1 and Step 2.2 of the next iteration. It should be noted that this method is most effective at the early stage of the evolutionary process (G_e generation) because, at that time, the solutions are quite far apart, leading to many empty regions. In the later stage, the locations of the solutions are close to each other, so the number of large empty regions will gradually decrease.

4. RESULTS AND DISCUSSION

4.1. Experiments

To evaluate the performance of the proposed method, experiments were conducted with the following conditions:

(i) Benchmark set: This paper considered 10 test problems from the ZDT benchmark set [15] and the UF benchmark set [16] with different classes of problem complexity. All test problems use the real value encoding.

(ii) Indicators: The generational distance (GD) and the inverse generational distance (IGD) indicators were used because they are typical and effective metrics to evaluate population quality. In addition, we also used the Hypervolume (HV) measure to evaluate the suitability when solving real-world problems. The references [17] and [18] present detailed formulas to compute those indicator values. GD is used to evaluate the convergency quality; the smaller the GD value, the better the population quality. IGD is used to evaluate the convergency and diversity quality; the smaller the IGD value, the better the population quality. HV is used to evaluate the convergency and diversity quality; the larger the value, the better the population quality.

(iii) Test problems parameter: Population size was 100; objective number was 2.

(iv) Algorithm parameters: The number of empty regions N_{emp} was *1/10* of population size, the step-length *k* was *0.01*, and the number of adjustment generation G_e was *1000*. Other parameters, such as evolution operators, crossover rate, and mutation rate are the same between MOEA/D and MOEA/D++.

(v) Simulation software: PlatEMO 3.5 implements on Matlab R2020b.

Each test problem will be performed 30 times and the results are expressed in the mean value and standard deviation value.

4.2. Results

In order to illustrate the performance of our proposal, we recorded the GD and IGD values of the last population in the primary algorithm and the new one after independent runs. The results are shown in table 1, with the mean in the above row and the standard deviation in the row below for each test problem. Numbers in bold are significantly better, and numbers in bold italics are marginally better.

Through the results, it is clear that: (i) MOEA/D++ is better than MOEA/D on 4/5 tested problems of the ZDT benchmark set and 3/5 tested problems of the UF benchmark set (in which the difference in diversity is significant, convergence is not too great); (ii) MOEA/D++ worse than MOEA/D on ZDT3 and UF5 (in which the difference in UF5 is significant); (iii) in all cases, MOEA/D++ is better than MOEA/D on the HV indicator.

Table 1. *The mean and standard deviation values for GD, IGD and HV.*

MOPs	**GD**		**IGD**		**HV**	
	MOEA/D	MOEA/D++	MOEA/D	MOEA/D++	MOEA/D	MOEA/D++
ZDT1	0.001210	**0.001057**	0.028927	**0.012603**	0.691293	**0.705667**
	0.000450	0.000193	0.021927	0.001612	0.691624	0.705759
ZDT2	0.001033	**0.000687**	0.092657	**0.019780**	0.350657	**0.421170**
	0.000629	0.000220	0.108098	0.030822	0.348052	0.420979
ZDT3	**0.002897**	0.004810	**0.040597**	0.089733	0.615480	**0.655433**
	0.002624	0.004879	0.017365	0.044008	0.617307	0.650848
ZDT4	0.037560	**0.033380**	0.172780	**0.155783**	0.516543	**0.537320**
	0.052287	0.054870	0.096698	0.080356	0.513290	0.534569
ZDT6	0.004467	**0.002203**	0.015953	**0.014023**	0.371500	**0.372063**
	0.009036	0.003991	0.020476	0.003237	0.371462	0.371866
UF1	**0.004890**	0.005673	0.270033	**0.218820**	0.439817	**0.476803**
	0.006256	0.004436	0.093864	0.092298	0.060519	0.066073
UF2	0.002547	**0.002450**	0.102777	**0.093280**	0.634567	**0.641663**
	0.001411	0.000855	0.058300	0.036248	0.027454	0.017345
UF3	0.007617	**0.006943**	0.328127	**0.315387**	0.349887	**0.352767**
	0.006160	0.004541	0.027585	0.024403	0.048931	0.033912
UF4	0.010193	**0.009593**	0.090943	**0.089667**	0.310513	**0.311857**
	0.000739	0.001046	0.004840	0.004007	0.005600	0.005490
UF5	**0.166790**	0.226397	**1.013513**	1.134670	0.002253	**0.004930**
	0.052171	0.060500	0.232276	0.371195	0.005157	0.012611
+/-/=		**7/3/0**		**8/2/0**		**10/0/0**

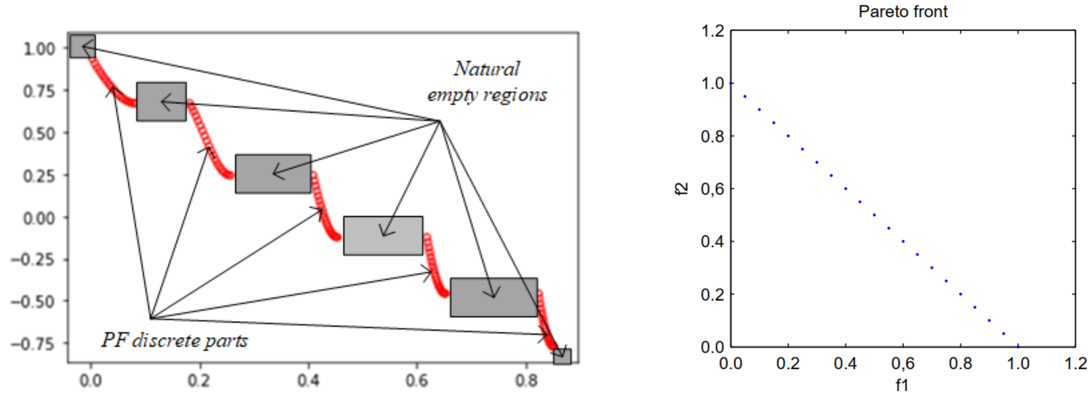
4.3. Discussions

Based on the experimental results, some specific discussions are as follows:

(i) The MOEA/D++ algorithm is better than the primary algorithm MOEAD in terms of diversity and conversity quality on most test problems. The reason is that adjustment RP leads to expanding the search region over different regions in the objective space while prioritizing the selection of solutions in empty regions. The convergence quality of the MOEA/D++ is still guaranteed because RP is shifted within the allowable range.

(ii) The PF of ZDT3 is discontinuous and consists of some discrete parts [15], so there are several natural empty regions in the PF. When the algorithm focuses its solution on PF, it will always create empty regions in the population distribution according to the natural empty regions of the PF. Similarly, the PF of UF5 is discontinuous and consists of discrete points [16], so there are many natural empty regions between those points. The inferior case for the ZDT3 and UF5 test problem is illustrated in figure 5.

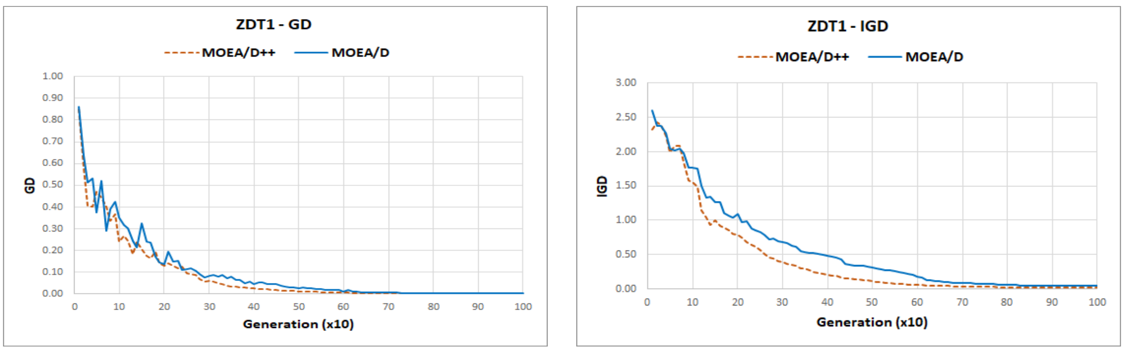
(iii) The HV value of the MOEA/D++ algorithm is always better, proving that it is suitable for real-world problems (the PF has not been determined).



a) ZDT3 b) UF5

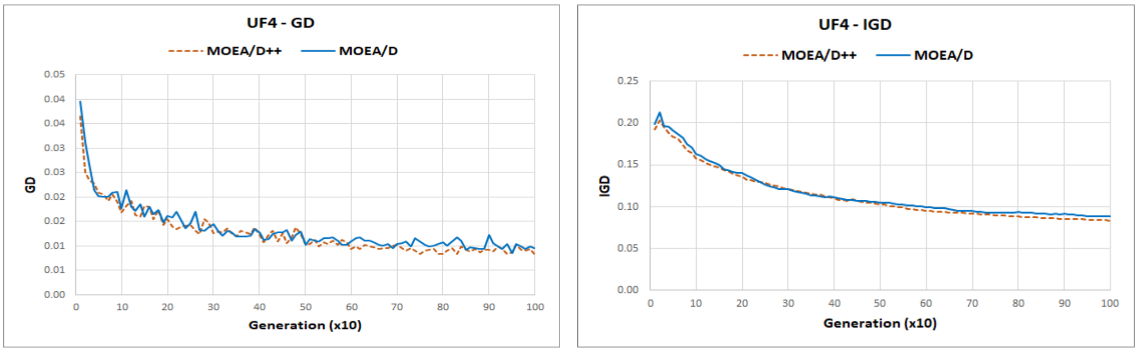
Figure 5. *Characteristics of PF of ZDT3 and UF5 test problem.*

The comparison results of GD and IGD values in some cases are shown visually in figure 6 and figure 7, corresponding to some typical cases. The primary *MOEA/D was presented in solid lines*, and the enhanced MOEA/D++ *was drawn with dash lines*. The vertical axis is the GD and IGD values, and the horizontal axis is the generation.



a) GD value b) IGD value

Figure 6. *Comparison of GD and IGD values on ZDT1.*



a) GD value b) IGD value

Figure 7. *Comparison of GD and IGD values on UF4.*

5. CONCLUSIONS

In this paper, we presented a novel method to improve the equilibrium between the exploitative and explorative aspects of MOEAs. In this, the evolutionary process adjusted to prioritize the selection of solutions located in empty regions in the population distribution, thereby increasing the diversity of the population. The proposed technique was applied to enhance the MOEA/D algorithm to adjust the position of RP when calculating the neighborhood. Experimental results show that the MOEA/D++ enhanced the balance between exploration and exploitation and improved the diversity and convergency quality of most tested problems in case the PF of the test problem is continuous. In the future, more research must be done to solve MOPs in which the PF has natural empty regions. Some of the techniques expected to be used are the forbidden zone technique based on the DM's interaction or the method suggested in the research [13] to detect natural gaps in the PF of problems.

REFERENCES

[1]. H. Wenlan, Zh. Yu, L. Lan, *"Survey on multi-objective evolutionary algorithms"*, J. of Physics: Conference Series, Conference Series, 1288, 012057, (2019).

[2]. K. Deb and J. Sundar, *"Reference point based multi-objective optimization using evolutionary algorithms"*, Proc. of the 8th annual conference on Genetic and evolutionary computation, pp.635-642, (2006).

[3]. D. H. Phan, J. Suzuki, I. Hayashi, *"Leveraging IndicatorBased Ensemble Selection in Evolutionary Multiobjective Optimization Algorithms"*, Proc. of 2012 Genetic and Evolutionary Computation Conference, pp.497-504, (2012).

[4]. R. H. Gomez, C. A. C. Coello, *"A Hyper- Heuristic of Scalarizing Functions"*, Proc. of 2017 Genetic and Evolutionary Computation Conference, pp.577-584, (2017).

[5]. J. G. Falcón-Cardona and C. A. C Coello, *"A Multi-Objective Evolutionary Hyper-Heuristic Based on Multiple Indicator Based Density Estimators"*, Proc. of 2018 Genetic and Evolutionary Computation Conference, pp.633-640, (2018).

[6]. J. G. Falcón-Cardona and C. A. C Coello, *"An ensemble indicator-based density estimator for evolutionary multi-objective optimization"*, Proc. of International Conference on Parallel Problem Solving from Nature, pp.201-214, (2020).

[7]. Zh. Qingfu and L. Hui, *"MOEA/D: A multiobjective evolutionary algorithm based on decomposition"*, J. of Evolutionary Computation, Vol 11, pp. 712-731, (2008).

[8]. Q. Xu, Z. Xu and T. Ma, *"A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition: Variants, Challenges and Future Directions"*, J. of IEEE Access, Vol. 8, pp.41588-41614, (2020).

[9]. K. Deb, H. Jain, *"An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints",* J. of IEEE Transactions on Evolutionary Computation, Vol 18(4), pp.577–601, (2014).

[10]. A. Masood, G. Chen, Y. Mei, M. Zhang, "*Adaptive Reference Point Generation for Many-Objective Optimization Using NSGA-III",* Lecture Notes in Computer Science, pp.358-370, (2018).

[11]. W. Rui, X. Jian, I. Hisao, W. Guohua, Zh. Tao, *"On the effect of reference point in MOEA/D for multi-objective optimization"*, J. of Applied Soft Computing, pp.25-34, (2017).

[12]. J. Zou, L. Fu, S. Yang, J. Zheng, G. Ruan, T. Pei, L. Wang, *"An adaptation reference-point-based multiobjective evolutionary algorithm"*, J. of Information Sciences, Vol 488, pp.41-57, (2019).

[13]. P. V. Pellicer, M. I. Escudero, S. F. Alzueta, K. Deb, *"Gap finding and validation in evolutionary multi- and many-objective optimization",* Proc. of the 2020 Genetic and Evolutionary Computation Conference, pp. 578-586, (2020).

[14]. T. B. Minh, N. Long, N. D. Dinh, *"Using bliss points to enhance direction based multi-objective algorithms"*, Proc. of 14th International Conference on Knowledge and Systems Engineering, pp.1-6, (2022).

[15]. E. Zitzler, K. Deb, L. Thiele, *"Comparison of multiobjective evolutionary algorithms: Empirical results"*, J. of Evolutionary Computation, Vol 8(2); pp.173-195, (2000).

[16]. Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, *"Multiobjective optimization test instances for the CEC 2009 special session and competition",* University of Essex and NanyangTechnological University, Tech. Rep. CES-487, (2008).

[17]. D.A.V. Veldhuizen, "*Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovation*", PhD thesis, Airforce Institute of Technology, Ohio (1999).

[18]. E. Zitzler, L. Thiele, and K. Deb, *"Comparision of multiobjective evolutionary algorithms: Empirical results",* J. of Evolutionary Computation, Vol 8(1), pp.173-195, (2000).

TÓM TẮT

Kỹ thuật điểm tham chiếu thích ứng để cải thiện chất lượng của giải thuật tiến hóa tối ưu đa mục tiêu dựa trên phân hoạch

Ứng dụng giải thuật tiến hóa tối ưu đa mục tiêu để giải các bài toán tối ưu đa mục tiêu là một lĩnh vực nghiên cứu nhận được nhiều sự quan tâm của các nhà khoa học. Mặc dù đã có nhiều công trình nghiên cứu trong lĩnh vực này, tuy nhiên, việc cân bằng giữa khả năng khai thác và thăm dò của giải thuật trong quá trình tiến hóa vẫn là một vấn đề thách thức trong lĩnh vực nghiên cứu. Bài báo đề xuất một cách tiếp cận mới nhằm giải quyết vấn đề cân bằng đó trên cơ sở phân tích phân bố của quần thể để xác định các vùng trống trong đó chưa có giải pháp được chọn. Thông tin về các vùng trống được sử dụng để điều chỉnh thích nghi vị trí điểm tham chiếu nhằm ưu tiên lựa chọn giải pháp tại các vùng trống. Thí nghiệm trên các bài toán mẫu cho thấy, phương pháp đề xuất làm tăng tính đa dạng của quần thể, từ đó cân bằng giữa các khả năng của giải thuật trong quá trình tiến hóa và cải thiện chất lượng của giải thuật.

Từ khóa: Tối ưu tiến hóa đa mục tiêu; Cân bằng giữa thăm dò và khai thác; Phân bố quần thể; Vùng trống, điểm tham chiếu thích nghi; MOEA/D.