

A New Selection Strategy for the Direction-based Multi-objective Evolutionary Algorithm

Long Nguyen, Lam Thu Bui
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam

Email: longit76@mta.edu.vn, lam.bui@mta.edu.vn

Abstract—The direction of improvement has been discussed and used to guide MOEAs during the search process towards the area of Pareto optimal set. One of typical examples using direction of improvement is the Direction based Multi-objective Evolutionary Algorithm (DMEA). For DMEA, its authors introduced a novel algorithm incorporating the concept of direction of improvement. Our preliminary analysis showed that DMEA uses a selection procedure based on a *weighted sum scheme* for a half of the population. This is good for convergence, but it might make the population quickly losing diversity. We propose a new selection strategy to avoid this issue. With the new selection strategy, we make DMEA to be better in balance between exploration and exploitation.

To validate the performance of our proposed selection strategy for DMEA, we carried out a case study on a wide range of test problems and comparison with other MOEAs. We obtained quite good results on primary performance metrics, namely the generation distance, inverse generation distance and hypervolume. Our analysis on the results indicates the better performance of DMEA with the new selection strategy in comparison with the most popular MOEAs.

Keywords—MOEA, Improvement Direction; Selection Strategy; DMEA.

I. INTRODUCTION

The direction of improvement has been discussed, conceptualized and used to guide MOEAs during the search process towards Pareto Optimal Front (POF) [15],[2],[4]. Recently, the authors in [5] introduced an algorithm incorporating the concept of direction of improvement, called Direction based Multi-objective Evolutionary Algorithm (DMEA). With DMEA, a population of solutions is evolved over time under the guidance of directions of improvement. Two types of directions are used including Convergence direction (from a dominated solution to a non-dominated one) and Spreading direction (between two non-dominated solutions) for generation of offspring along those directions. An archive is maintained over time. This archive is not only used for contributing elite solutions for the next generation, but

also for derivation the directions. In order to fill the populations of the next generation as well as the archive, DMEA gets solutions from the combined population (resulted from combining the current archive and the offspring). For the next generation, the first half of the population is filled by the non-dominated solutions using niching information in the decision space, while the second half is filled by using a weighted-sum technique for all remaining solutions in the combined population. The selection of non-dominated solutions for updating the archive is also assisted by a niching mechanism, in which neighborhoods of rays in objective space serve as niches. These rays originate from the current estimate of the POFs ideal point and emit uniformly into the hyperquadrant that contains the current POF estimate.

DMEA was shown competitive with other popular MOEAs. However, we found an issue that relates to the performance of DMEA: The main population might be quickly losing diversity by using a weighted sum scheme. DMEA uses a selection procedure based on a weighted sum scheme for the second half of the population. Because weighted values are used to control population to be convergence, so it might makes a good convergence, but it will cause the population quickly losing diversity.

In this paper, we deal with the above issue and introduce a new selection strategy for DMEA. The improved version with the new selection strategy is called IDMEA. We propose to use a new selection strategy using the ray-based niching to select dominated solutions and low ranked non-dominated solutions to enter the next generation. To validate the proposed IDMEA, we carried experiments on 18 problems from 3 well-known benchmark sets. We also make comparisons with 6 existing MOEAs on 3 performance metrics. The result strongly suggests a superior of IDMEA over the original version DMEA. It also shows a better performance of IDMEA in comparison to other MOEAs.

The remainder of this paper is organized as follows. A brief summary of MOEAs is given in Section II and

the description of DMEA in Section III. Detail of our proposed selection strategy is shown in Section IV. The experimental results is presented in Section V to examine the effectiveness and efficiency of proposed techniques. Conclusion and future work are given in Section VI.

II. MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Practical problems in real-life usually possess the feature of multi-objectivity having multiple competing objectives (or criteria). Their solutions therefore describe alternatives, each of which represents a different compromise between the conflicting objectives. The set of optimal solutions to the problem are called *Pareto optimal set*. Its projection in objective space is known as the *Pareto Optimal Front* (POF). The *ideal point* of the POF is the vector whose components contain the result of minimizing each objective individually.

Mathematically, in a k -objective unconstrained (bound constrained) minimization problem, a vector function $\vec{f}(\vec{x})$ of k objectives is defined as:

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})] \quad (1)$$

in which \vec{x} is a vector of decision variables in v -dimensional \mathbb{R}^v . In Evolutionary Computing, \vec{x} represents an individual in the population to be evolved. The value $f_j(\vec{x})$, then, describes the performance of individual \vec{x} as evaluated against the j th objective in the Multi-Objective Optimization Problem (MOP).

An individual \vec{x}_1 *dominates* \vec{x}_2 if \vec{x}_1 is not worse than \vec{x}_2 on all k objectives and is better than \vec{x}_2 on at least one objective. If \vec{x}_1 does not dominate \vec{x}_2 and \vec{x}_2 also does not dominate \vec{x}_1 , then \vec{x}_1 and \vec{x}_2 are said to be *non-dominated* with respect to each other. If we use the symbol “ \preceq ” to denote that $\vec{x}_1 \preceq \vec{x}_2$ means \vec{x}_1 dominates \vec{x}_2 , and the symbol “ $\not\preceq$ ” between two scalars a and b to indicate that $a \not\preceq b$ means a is not worse than b , then *dominance* can be formally defined as [8]:

Definition 1 (Dominance): $\vec{x}_1 \preceq \vec{x}_2$ if the following conditions are held:

1. $f_j(\vec{x}_1) \not\preceq f_j(\vec{x}_2) \forall j \in \{1, 2, \dots, k\}$; and,
2. $\exists j \in \{1, 2, \dots, k\} : f_j(\vec{x}_1) \prec f_j(\vec{x}_2)$.

In general, if an individual is not dominated by any other individual in the population, it is called a non-dominated solution. All non-dominated solutions in a population form the non-dominated set as formally described in the following definition:

Definition 2 (Non-Dominated Set): A set S is said to be the non-dominated set of a population P if the following conditions are met:

1. $S \subseteq P$; and,
2. $\forall \vec{s} \in S \nexists \vec{x} \in P : \vec{x} \preceq \vec{s}$.

If P represents the entire search space, then S is referred to as the *global Pareto optimal set*. If P represents only a sub-space, then S is called the *local Pareto optimal set*. While there can be multiple local Pareto optimal sets, there exists only one global one.

Multi-objective evolutionary algorithms (MOEAs) are stochastic techniques being used to find Pareto optimal solutions for MOPs. There are two key problems that MOEAs have to deal with [8]. The first one is how to get as close as possible to the POF. This is challenging because of the stochasticity of the convergence process. The second one is how to keep solutions diverse. A diverse set of solutions will provide decision makers, designers, etc with more choice. However, working on a set of solutions instead of only one makes the measurement of MOEA convergence more difficult because one individual’s closeness to the POF does not act as a measure for the entire set. Unsurprisingly, then, convergence and diversity are commonly used as performance criteria when optimization algorithms are assessed and compared with each other [21].

To date, many MOEAs have been developed (i.e Pareto Archived Evolution Strategy (PAES)[12], Strength Pareto EA 2 (SPEA2) [20], Pareto frontier DE (PDE)[2], NSGA-II [9], Decomposition based Multi-objective Evolutionary Algorithm (MOEA/D) [18] and Multi-Objective Particle Swarm Optimization (MOPSO) [7], MODE-LD+SS [3] and the Direction based Multi-objective Evolutionary Algorithm (DMEA)[5] are typical examples of elitist MOEAs) and they are usually classified into two broad categories: with and without elitism. Elitist approach is a mechanism to preserve the best individuals, once found, during the optimization process. The concept of elitism was established at an early stage of EC (see, for example, [11]); and to date, it has been widely used in EAs. Elitist approach can be realized either by placing one or more of the best parents directly into the next generation of individuals, or by replacing only those parents that are dominated by their offspring [16]. Elitist MOEAs usually (but not necessarily) employ an external set called the archive to store the non-dominated solutions after each generation.

III. DIRECTION-BASED MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM - DMEA [5].

Elitism is a very useful mechanism to enhance MOEAs. In DMEA the authors adopt an elitist mechanism in their methodology. In particular, they address both issues: *interaction between archive and main population* and *archive update*.

In DMEA, an external archive is being maintained over time. Its task is not only to store elitist solutions

but also to contribute directional information for guiding the evolutionary process. Knowing how solutions have improved from one iteration to the next is useful information in any iterative optimization approach. DMEA uses this information during the reproduction phase. At every generation, the archive is exploited to determine directions of improvement. The main population is then perturbed along those directions in order to produce offspring. Subsequently, the offspring are merged with the current archive to form a *combined population*, from which the next generation's archive and parental pool are derived.

The second unique feature of DMEA entails the deterministic control of some aspects of the selection of non-dominated solutions for archive and main population. Augmenting MOEAs with deterministic steps is not uncommon. In DMEA solutions are placed into two categories: non-dominated and dominated solutions. The archive is updated by using niching in objective space, while up to half of the next-generation main population is filled by applying niching criteria in decision variable space.

A. Directional information

In DMEA two types of directional information are used to perturb the parental population prior to offspring production: convergence and spread (see Fig. 1). **Convergence direction (CD):** In general defined as the direction from a solution to a better one, CD in MOP is a normalized vector that points from dominated to non-dominated solutions. If non-dominated solutions are maintained globally, CD corresponds to the global direction of convergence. In unconstrained MOP, a dominated solution guided by this direction is more likely to find a better area in the decision space than an unguided solution. **Spread direction (SD):** Generally defined as the direction between two equivalent solutions, SD in MOP is an unnormalized vector that points from one non-dominated solution to another. If solutions are perturbed along the SD, a better spread within the population should be obtained.

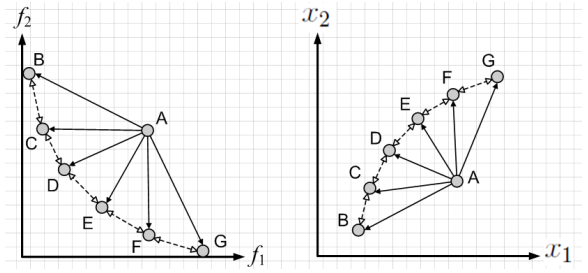


Fig. 1. Illustration of convergence (black arrows) and spread (hollow arrows) directions in objective space (left) and decision variable space (right).

B. Niching information

A character of quality in MOP is the even spread of non-dominated solutions across the POF [8]. In DMEA a bundle of rays are used either emitting uniformly from the estimated ideal point into the part of objective space that contains the POF estimate, or being parallel as depicted in Fig. 2. The number of rays equals the number of non-dominated solutions wanted by the user. Rays emit into a “hyperquadrant” of objective space, i.e. the subspace that is bounded by the k hyperplanes $f_i = f_{i,\min}$, $i \in \{1, 2, \dots, k\}$ and described by $f_i \geq f_{i,\min} \forall i \in \{1, 2, \dots, k\}$ where $f_{i,\min} \approx \min_{\text{all } A_1, A_2, \dots} f_i$ with A_1, A_2, \dots being the solutions stored in the current archive. By their construction, the hyperquadrant contains the estimated POF.

During the archival update (interesting non-dominated solutions), the rays are used as reference lines to select particular non-dominated solutions from the combined population. One by one, the rays are scanned and the non-dominated solution closest to a given ray is selected and archived.

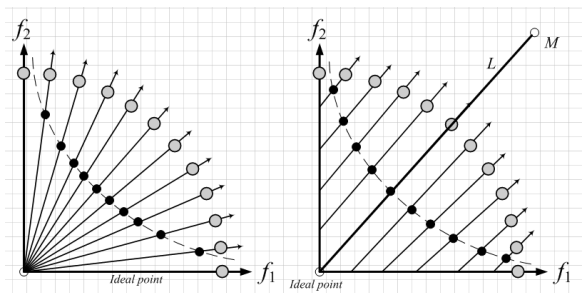


Fig. 2. Illustration of the ray system in a 2-dim MOP. The left graph: origin of the bundle is collocated with the estimated ideal point. The ray bundle is bounded by the two lines $f_1 = f_{1,\min}$ and $f_2 = f_{2,\min}$ and it emits uniformly into the top right quadrant which contains the POF estimate. The right graph: The rays start from generated points and parallel with the central lines of the top right quadrant.

A niching operator is used for the main population. From the second generation onward, the population is

composed from two equal parts: one part for convergence, and the other one for diversity. The first part is filled by non-dominated solutions up to a maximum of $n/2$ solutions from the combined population, where n is the population size. This filling task is based on niching information in the decision space.

C. General structure of algorithm

The step-wise structure of the DMEA algorithm [5] is as follows:

- **Step 1.** Initialize the main population P with size n .
- **Step 2.** Evaluate the population P .
- **Step 3.** Copy non-dominated solutions to the archive A .
- **Step 4.** Generate an interim mixed population (M) of the same size n as P
 - Loop {
 - * Select a random parent Par without replacement.
 - * If Par is dominated, $j = 1$. Else $j = 2$.
 - * Generate a solution S_j using Convergence Direction ($j = 1$) or Spread Direction ($j = 2$) information [5].
 - * Add S_j to M .
 - } Until (the mixed population is full).
- **Step 5.** Perform the polynomial mutation operator [8] on the mixed population M with a small rate.
- **Step 6.** Evaluate the mixed population M .
- **Step 7.** Identify the estimated ideal point of the non-dominated solutions in M and determine a list of n rays R (starting from the ideal point and emitting uniformly into the hyperquadrant that contains the non-dominated solutions of M) [1]
- **Step 8.** Combine the interim mixed population M with the current archive A to form a combined population C (i.e. $M + A \rightarrow C$).
- **Step 9:** Create new members of the archive A by copying non-dominated solutions from the combined population C
 - Loop{
 - * Select (without replacement) a ray $R(i)$.
 - * In C , find the non-dominated solution whose distance to $R(i)$ is minimum.
 - * Select (without replacement) this solution and copy it to the archive.
 - } Until (all n rays are scanned)
- **Step 10:** Determine the new population P for the next generation.
 - Empty P .

- Determine the number m of non-dominated solutions in C . Using either niching in decision space (function C1 in [5]) or in objective space (function C2 in [15]) to select $n/2$ solutions from C to P .

- Apply a *weighted-sum scheme* to copy $\max\{n - m, n/2\}$ solutions to P .

- **Step 11:** Go to Step 4 if stopping criterion is not satisfied.

Function C1 and C2 are described as follow:

- Function C1 in Decision Space [5]:

- Determine niching value (average Euclidean distance to other non-dominated solutions in decision space) for all non-dominated solutions in C .
- Sort non-dominated solutions in C according to niching values.
- Copy (without replacement) the $n/2$ solutions with highest niching value to P .

- Function C2 in Objective Space [15]:

- Determine density-based niching value for all non-dominated solutions in C in objective space.
- Sort non-dominated solutions in C according to niching values.
- Copy the $n/2$ solutions with highest niching value to P .

We use function C2 in [15] for experiments in this paper.

IV. THE PROPOSED SELECTION STRATEGY

In DMEA: for the second half of the main population, a selection procedure based on a *weighted sum scheme* is used, the solutions are sorted based on the normalized weighted-sum objective value. After sorting, $n - \min\{m, n/2\}$ solutions with the highest scores F are chosen and copied into the main population. This *weighted-sum scheme* allows dominated and (according to the niching value) low ranked non-dominated solutions to enter the next generation. Its purpose is to strike a balance between exploitation and exploration. However, the weighted values are used to control the population to be converged, so it might make a good convergence, but it might also make the population quickly losing diversity.

We design a new selection strategy to avoid this effect.

We propose to use a new selection strategy using the ray-based niching in the same way as DMEA does for the archive. The first we scan all rays to calculate distance to all solutions in combined population (between current archive and offspring). Then we get the closest solutions for each scanned ray and put to main population for next generation. It stops until $\max(n-m, n/2)$ dominated solutions are selected and put to main population. Here m is the number of non-dominated solutions in the archive.

We do steps bellow for the new selection strategy:

- **Step 1:** Scan all rays to do following tasks:
 - *Step 1.1:* Calculate the distance from all solutions in the combined population.
 - *Step 1.2:* Make a sorted index list of population by above distance values.
 - *Step 1.3:* Get the solution with index for the lowest item from sorted index list.
- **Step 2:** Copy selected solution to the main population.

The pseudo code for proposed selection strategy is shown in Algorithm 1.

Algorithm 1: Ray-based niching selection strategy.

Input: The combined population (C) at current generation

Output: Select $\max\{n - m, n/2\}$ solutions to the next generation P. (Step 10)

- Set $counter = 0$.

while $counter < ctr_size$ **do**

- Set $min_{distance} = MAX_{INT}$.
- Set $choice_{index} = -1$.

For $i \leftarrow 0$ **to** n_r (number of rays)

- For $j \leftarrow 0$ **to** n_m (n_m is mixed population's size.)
 - Calculate distance d_i from solution s_j to ray r_i
 - If $d_i < min_{distance}$ **Then**
 - * Set $min_{distance} = d_i$
 - * Set $choice_{index} = i$

End For(j)

- Get solution $s_{[choice_{index}]}$ from mixed population and put it to main population P.

End For(i)

return P;

In DMEA, the computational complexity of the weighted sum scheme is $O(n \log n)$, the cost mainly comes from the task of using the sorting function. The computational complexity for the new selection strategy

is $O(n^2)$ the cost mainly comes from the task of scanning n rays to find the closest solutions for each ray and put them into the main population.

V. EXPERIMENTAL STUDIES

A. Testing problems and parameter settings

This paper considers a large set of 18 continuous benchmark problems from 3 well-known benchmark sets, namely ZDT [19], DTLZ [10] and UF [14]. For these problems, the number of variables are between 10 and 30 while the number of objectives are 2 or 3. The reason for us to select these benchmark problems is that each benchmark illustrates a different class of problem complexity such as *convexity/non-convexity, uniformity/non-uniformity, single-modality/multi-modality, linearity/non-linearity, interdependency, and continuity/discontinuity* and they are widely analyzed in the literature. The testing parameters for these problems are reported in Table I.

TABLE I
PARAMETER SETTINGS FOR THE EXPERIMENTS.

Prbs	Objs/Vars	P.size	N.gens	N.runs	POF.size
ZDT1	2/30	100	1000	30	100
ZDT2	2/30	100	1000	30	100
ZDT3	2/30	100	1000	30	100
ZDT4	2/10	100	1000	30	100
ZDT6	2/10	100	1000	30	100
DTLZ2	3/10	300	1000	30	300
DTLZ3	3/10	300	1000	30	300
DTLZ7	3/10	300	1000	30	300
UF1	2/10	100	1000	30	100
UF2	2/10	100	1000	30	100
UF3	2/10	100	1000	30	100
UF4	2/10	100	1000	30	100
UF5	2/10	100	1000	30	100
UF6	2/10	100	1000	30	100
UF7	2/10	100	1000	30	100
UF8	3/10	300	1000	30	300
UF9	3/10	300	1000	30	300
UF10	3/10	300	1000	30	300

The experiments were carried out on all problems in Table I. For DMEA and IDMEA experiments, the mutation rate was kept at the same small rate of 0.01, and the perturbation rate was 0.4. Other MOEAs included: MOEA/D [18], MOEA/D-DE [13], NSGA-II [9], NSGA-II-DE [6], SPEA2 [20] on the same experimental environment. All algorithms are ran 30 times with different random seeds.

B. Performance metrics

Performance metrics are usually used to compare algorithms in order to form an understanding of which algorithm is better and in what aspects. However, it is hard to define a concise definition of algorithmic

performance. In general, when doing comparisons, a number of criteria are employed [21]. Since each metric has both advantage and disadvantage, we will look at all three popular criteria: the generational distance (GD), the inverse generational distance (IGD) and hypervolume (HYP).

The GD measure is defined as the average distance from a set of solutions, denoted P , found by evolution to the global Pareto optimal set (POS) [17]. The first-norm equation is defined as

$$GD = \frac{\sum_{i=1}^n d_i}{n} \quad (2)$$

where d_i is the Euclidean distance (in objective space) from solution i to the nearest solution in the POS, and n is the size of P . This measure is considered for convergence aspect of performance. Therefore, it could happen that the set of solutions is very close to the POF, but it does not cover the entire the POF.

The measure IGD takes into account both convergence and spread to all parts of the POS. The first-norm equation for IGD is as follows

$$IGD = \frac{\sum_{i=1}^{\bar{N}} \bar{d}_i}{\bar{N}} \quad (3)$$

where \bar{d}_i is the Euclidean distance (in objective space) from solution i in the POS to the nearest solution in P , and \bar{N} is the size of the POS. In order to get a good value for IGD (ideally zero), P needs to cover all parts of the POS. However, this method only focuses on the solution that is closest to the solution in the POS indicating that a solution in P might not take part in this calculation.

The HYP[21] is also named as S Metric. Being different from IGD, HYP is a unary measure. IGD uses the POF as a reference, which is not practical for real-world applications. Thus, HYP attracts increasing attentions recently. HYP is a measure of the hypervolume in objective space that is dominated by a set of non-dominated points. In the following experiments, before computing HYP, the values of all objectives are normalized to the range of a reference point for each test problem. The reference points normally is the worst-possible point in objective space. In our experiments with 8 MOEAs with 18 test problems, we choose the reference points by the following way: With minimizing test problems, the reference points are taken from the maximize values of each objective on all of MOEAs results. Otherwise, the reference points are taken from the minimum ones. Not to change the properties of HYP, we normalize the HYP

values (in range [0,1]) by formula:

$$HYP_k = \frac{HYP_k^*}{\max_{1 \leq i \leq N} (HYP_i^*)} \quad (4)$$

There, HYP_k^* is the HYP value for a test problem of MOEA k^{th} , $k \in 1, \dots, N$, HYP_k is the normalized value of HYP_k^* .

In our test, standard deviations are calculated when averaging values of GD, IGD and HYP for 30 independence runs with different random seeds. In our experiments, For all these results, we used statistical t-tests for significant difference between results for IDMEA and other MOEAs. T-test values (Ps) are calculated by formula:

$$P = \frac{x_1 - x_2}{se(x_1 - x_2)} \quad (5)$$

Where $se(x_1 - x_2)$ is the standard deviation of the difference $(x_1 - x_2)$. We tested the significant difference between IDMEA with other MOEAs on GD, IGD, HYP for 18 test problems in 30 independence runs. In tables: II III, IV and V, (+) indicates that the difference between IDMEA and other MOEAs is significant (Using t-test with $\alpha = 0.05$).

C. Results and Analysis

1) IDMEA's performance in comparison with DMEA:

In order to analyze the performance of IDMEA and DMEA, we recorded all non-dominated solutions and calculated values of GD, IGD, HYP (together with standard deviation values - in second lines). These values were reported in Table II.

We first visualized all non-dominated solutions obtained by IDMEA and compare them to that of DMEA (see some typical snapshots in Figs. 3 to 12). All graphs clearly show the better performance of IDMEA over DMEA. In particular, with DMEA, the final non-dominated solutions were not always well-located uniformly in the area of POF; and for some problems, some parts of POFs are more dense than others. An example can be seen in the case of ZDT1 and ZDT4, where the central area of POFs are more much crowded than other areas. In contrast, for IDMEA, the final solutions are distributed more uniformly along POF than that from DMEA's. This finding shows the strong effect of our new selection strategy: the use of ray-based niching values pushed solutions spreading along the area of POF.

For a more concrete comparison between IDMEA and DMEA, we focus on the numerical results with three metrics in Table II. We highlighted (with a bold font) all cases where IDMEA or DMEA was the better one. The

results indicate that IDMEA is always ranked above the original DMEA. In more details, among 18 problems, IDMEA was better in 12 problems with regards to GD and HYP and 13 problems with IGD. The statistical t-tests confirmed this finding (see Table II). Also, IDMEA obtained better results in most of difficult problems like multi-modal problems of ZDTs, DTLZs or UFs. Together with graphical results, these findings confirm the advantage of IDMEA with the new selection strategy.

2) *Behavior of the algorithm over time:* To get a full understanding of the proposed selection strategy, we also analyze the behavior of IDMEA over time. There are several ways for understanding the behavior of IDMEA, here we decided to record the values of GD and IGD over time and plotted them on time dependent graphs.

With almost all test problems (as examples, they were all visualized in Figs: 13 and 17 on GD and IGD metrics with ZDT1, and ZDT3 and for both DMEA and IDMEA), we found that for the following results on the GD metric, there is a common pattern that at the earlier stage, DMEA was quite fast in convergence, getting slightly better value of GD than IDMEA did. However, at the latter stage, IDMEA was getting better as the search progressed. That might be the case that IDMEA maintained better the balance between convergence and spreading of the population. At the later time, population obtained by DMEA might have the low diversity affecting the searching ability of DMEA. This finding is confirmed when looking at the graphs of IGD metric. IDMEA always showed better values of IGD than DMEA.

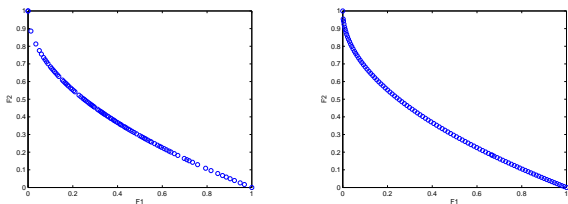


Fig. 3. ZDT1 results in DMEA and proposed IDMEA in the objective space.

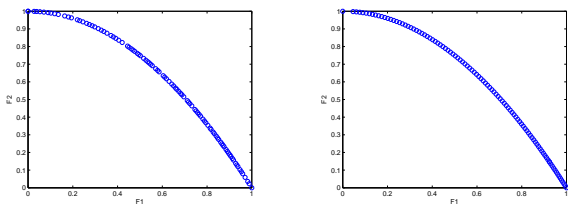


Fig. 4. ZDT2 results in DMEA and proposed IDMEA in the objective space.

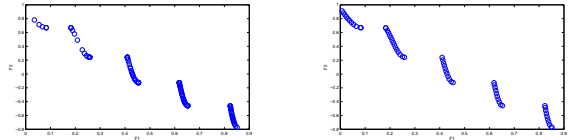


Fig. 5. ZDT3 results in DMEA and proposed IDMEA in the objective space.

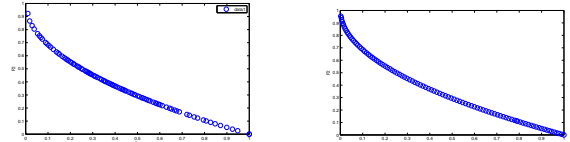


Fig. 6. ZDT4 results in DMEA and proposed IDMEA in the objective space.

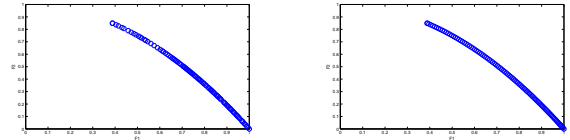


Fig. 7. ZDT6 results in DMEA and proposed IDMEA in the objective space.

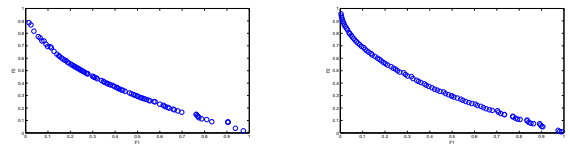


Fig. 8. UF2's non-dominated set obtained by DMEA (left) and IDMEA (right) in the objective space.

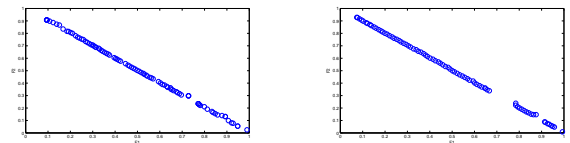


Fig. 9. UF7's non-dominated set obtained by DMEA (left) and IDMEA (right) in the objective space.

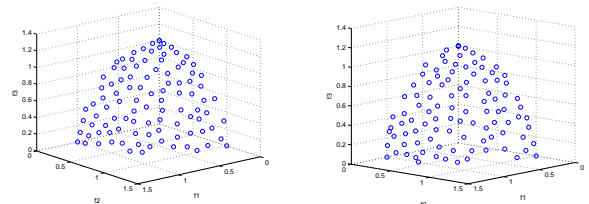


Fig. 10. DTLZ2's non-dominated set obtained by DMEA (left) and IDMEA (right) in the objective space.

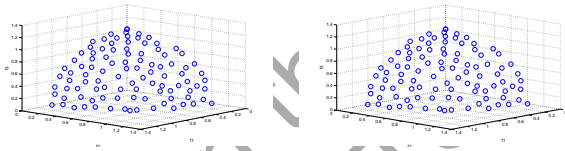


Fig. 11. DTLZ3's non-dominated set obtained by DMEA (left) and IDMEA (right) in the objective space.

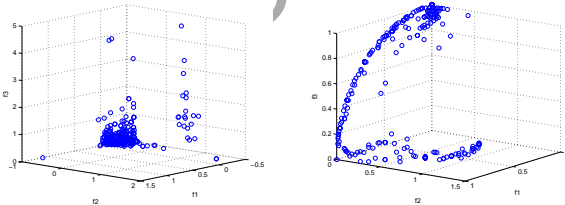


Fig. 12. UF8's non-dominated set obtained by DMEA (left) and IDMEA (right) in the objective space.

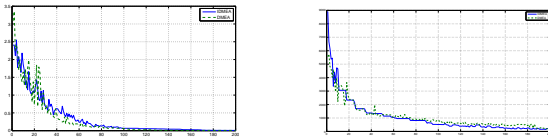


Fig. 13. Visualization of GD (left) and IGD (right) over time for ZDT1.

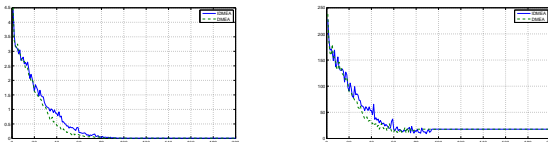


Fig. 14. Visualization of GD (left) and IGD (right) over time for ZDT2.

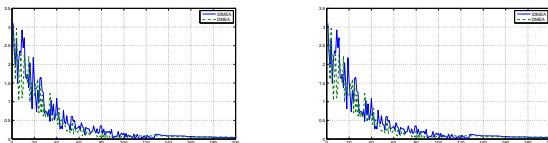


Fig. 15. Visualization of GD (left) and IGD (right) over time for ZDT3.

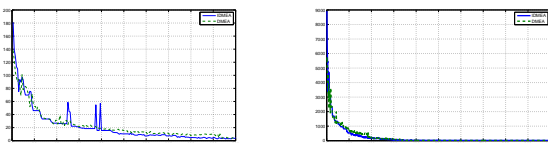


Fig. 16. Visualization of GD (left) and IGD (right) over time for ZDT4.

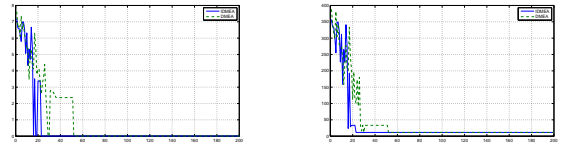


Fig. 17. Visualization of GD (left) and IGD (right) over time for ZDT6.

3) *IDMEA's performance in comparison with other MOEAs:* In order to make comparisons, we calculated values of GD, IGD, HYP (together with standard deviation values - in second lines). These values were reported in Tables: III, IV, V. For tables with GD, IGD and HYP, we ranked all algorithms and calculated the average ranks in all problems.

With regards to GD, we see that NSGA-II and SPEA2 obtained the worst results. While the use of DE operations made NSGA-II better, it made MOEA/D worse. IDMEA obtained quite comparable GD values with that of MOEA/D, in which IDMEA was the best in four problems and MOEA/D had 6 problems. According to the t-test results, the four best results of IDMEA were significantly better than that of MOEA/D. Meanwhile the six best results of MOEA/D were significantly better than that of IDMEA. In general, the best average ranks belong to MOEA/D (2.89), IDMEA (3.39) and NSGA-II-DE (3.28). This means the good convergence of those algorithms over all tested problems.

However, for IGD, the standings of algorithms were quite different. MOEA/D became the worst performers. That is because the non-dominated solutions found by MOEA/D lost of spreading factor. Meanwhile, IDMEA and NSGA-II-DE still showed good performance on IGD, in which IDMEA demonstrated the better performance with the highest rank (average rank 2.72) among MOEAs. it was the best on test problems: ZDT1, DTLZ3, DTLZ7, UF1, UF2, UF4 and UF7. The statistical t-tests confirmed this finding (see Table IV). The finding is strengthened by the results on HYP metric. In Table V we also have the similar results for IDMEA. It got the best average rank (2.50) with nine times getting the best HYP values (on test problems: ZDT1, ZDT2, DTLZ3, UF1, UF2, UF4, UF6, UF7 and UF10). IDMEA was able to find better spreading POFs than other algorithms. The statistical t-tests confirmed this finding (see Table V).

In overall, out of three metrics, IDMEA was the best algorithms in 2, while MOEA/D and NSGA-II-DE were the best in 1 metric respectively. It again shows the strength of our new selection strategy on DMEA. The

TABLE II

THE AVERAGE VALUES OF GD, IGD AND HYP WITH STANDARD DEVIATION VALUES FOR ORIGINAL DMEA AND IDMEA. (+) INDICATES THAT THE DIFFERENCE BETWEEN IDMEA AND DMEA IS SIGNIFICANT (USING T-TEST WITH $\alpha = 0.05$).

Problems	GD(STDEV)		IGD(STDEV)		HYP(STDEV)	
	DMEA	IDMEA	DMEA	IDMEA	DMEA	IDMEA
ZDT1	0.003625 0.00022(+)	0.004861 0.00006	0.004847 0.00035(+)	0.003558 0.00002	0.997387 0.00049(+)	1.000000 0.00003
ZDT2	0.004301 0.00023	0.004325 0.00005	0.004239 0.00025	0.00419 0.00005	0.999011 0.00013(+)	1.000000 0.00001
ZDT3	0.005564 0.00036(+)	0.00383 0.0001	0.011544 0.00142(+)	0.008216 0.00006	0.995634 0.00135(+)	0.998823 0.00005
ZDT4	0.008269 0.02189	0.00541 0.00067	0.0092 0.0216	0.004166 0.00069	0.999654 0.03089	0.999984 0.00163
ZDT6	0.004838 0.00415(+)	0.004432 0.01062	0.013425 0.00077(+)	0.013128 0.00001	0.860705 0.00241	0.860935 0.00001
DTLZ2	0.366645 0.16195(+)	0.263353 0.1755	0.052655 0.00169(+)	0.200311 0.06542	1.000000 0.17997(+)	0.967400 0.67391
DTLZ3	0.33232 0.27025	0.267104 0.18145	0.338842 0.14589	0.413976 0.13801	1.000000 0.78454(+)	0.999999 0.17931
DTLZ7	1.200727 0.82222(+)	0.057542 0.01859	1.401515 0.82239(+)	0.038833 0.01497	0.934889 0.47604(+)	0.902587 0.41984
UF1	0.008775 0.00168(+)	0.012066 0.00468	0.011829 0.00284	0.01202 0.00374	1.000000 0.08042	0.998649 0.00747
UF2	0.007396 0.0014(+)	0.008042 0.00099	0.009971 0.00116(+)	0.008078 0.00112	0.799653 0.00872	1.000000 0.00673
UF3	0.110821 0.08758	0.127535 0.12159	0.276663 0.08614	0.267 0.11157	0.545415 0.60079	0.973358 0.69025
UF4	0.035069 0.00174(+)	0.034264 0.00141	0.034973 0.0015	0.034687 0.00199	0.486353 0.00387	1.000000 0.00337
UF5	0.222777 0.17586(+)	0.264984 0.13262	0.529945 0.30579(+)	0.036908 0.0109	0.965000 1.70164	0.994899 0.96067
UF6	0.243305 0.13363	0.234867 0.13257	0.222589 0.13599	0.258278 0.15383	0.584803 0.96996	1.000000 1.07568
UF7	0.007284 0.00137(+)	0.006527 0.00108	0.010739 0.00175(+)	0.008157 0.00132	0.999235 0.00484(+)	1.000000 0.00236
UF8	1.485403 0.01165(+)	1.160744 0.16451	0.905205 0.21213(+)	0.778199 0.22296	1.000000 0.46996(+)	0.998214 0.70687
UF9	2.335798 0.22714(+)	0.771123 0.32922	0.1347 0.06342	0.202848 0.14098	1.000000 0.17586(+)	0.995617 0.18544
UF10	1.225281 0.15671(+)	1.042361 0.32582	0.63601 0.38463(+)	0.453612 0.15468	0.999284 0.51534	1.000000 0.62054
Best values	6	12	5	13	6	12

new selection strategy helped DMEA being balanced between two aspects: convergence and spreading. In other words, with a balance between exploration and exploitation, performance of DMEA was improved and was shown by all three popular performance metrics. The results are better than original one in all metrics.

TABLE III

THE RANK (AVERAGE VALUE OF GD) AND STANDARD DEVIATION VALUES FOR EACH ALGORITHM. (+) INDICATES THAT THE DIFFERENCE BETWEEN IDMEA AND OTHER MOEAs IS SIGNIFICANT (USING T-TEST WITH $\alpha = 0.05$)

Problems	IDMEA	MOEA/D	MOEA/D-DE	NSGA-II	NSGA-II-DE	SPEA2
ZDT1	(3)0.004861 0.00006	(1)0.003766 0.00001(+)	(2)0.00413 0.00008(+)	(6)0.00532 0.00041(+)	(4)0.005186 0.00028(+)	(5)0.005302 0.00023(+)
ZDT2	(5)0.004325 0.00005	(4)0.004303 0.00001(+)	(6)0.004422 0.00006(+)	(3)0.004273 0.00032	(2)0.004133 0.0002(+)	(1)0.004105 0.00022(+)
ZDT3	(1)0.00383 0.0001	(2)0.004884 0.00001(+)	(3)0.005129 0.00011(+)	(5)0.006076 0.00043(+)	(4)0.0062 0.00041(+)	(6)0.006349 0.00032(+)
ZDT4	(3)0.00541 0.00067	(1)0.003898 0.00014(+)	(2)0.003971 0.00009(+)	(4)0.056643 0.0892(+)	(6)0.077546 0.09002(+)	(5)0.074256 0.10337
ZDT6	(6)0.004432 0.01062	(1)0.00325 0.00003	(2)0.003322 0.00024(+)	(4)0.003942 0.00019	(3)0.003273 0.00027	(5)0.00422 0.00014
DTLZ2	(6)0.263353 0.1755	(1)0.066486 0.00007(+)	(2)0.075109 0.00011(+)	(4)0.081473 0.00264(+)	(3)0.080561 0.00261(+)	(5)0.08225 0.00127
DTLZ3	(5)0.267104 0.18145	(1)0.060372 0.00006(+)	(4)0.075961 0.01769(+)	(3)0.635451 0.7024(+)	(2)0.062155 0.00166(+)	(6)1.380131 0.13851(+)
DTLZ7	(1)0.057542 0.01859	(2)0.133183 0.02399(+)	(5)0.199028 0.02351(+)	(4)0.148025 0.00526(+)	(3)0.143845 0.00355(+)	(6)2.037204 0.21199(+)
UF1	(4)0.012066 0.00468	(5)0.027819 0.0065(+)	(6)0.067801 0.00703(+)	(2)0.011192 0.0082	(3)0.011549 0.00548(+)	(1)0.007646 0.00323(+)
UF2	(1)0.008042 0.00099	(5)0.026468 0.00348(+)	(6)0.034738 0.00555(+)	(4)0.011934 0.00329(+)	(2)0.010039 0.00127	(3)0.010162 0.00172(+)
UF3	(6)0.127535 0.12159	(5)0.076694 0.02588	(1)0.02453 0.02798(+)	(3)0.068729 0.02138	(2)0.06703 0.03799(+)	(4)0.074826 0.03048
UF4	(1)0.034264 0.00141	(5)0.046091 0.01508(+)	(6)0.060091 0.00819(+)	(4)0.038245 0.00127(+)	(2)0.037238 0.00146	(3)0.037543 0.00156(+)
UF5	(3)0.264984 0.13262	(6)0.697828 0.24986(+)	(5)0.564877 0.22891(+)	(1)0.131011 0.05435(+)	(4)0.315271 0.21944	(2)0.135676 0.0431(+)
UF6	(5)0.234867 0.13257	(3)0.090533 0.04907(+)	(6)0.306746 0.06107	(2)0.087615 0.03462(+)	(1)0.052228 0.09148(+)	(4)0.111452 0.06212
UF7	(2)0.006527 0.00108	(5)0.019532 0.00785(+)	(6)0.043162 0.00624(+)	(3)0.006533 0.00166	(4)0.006895 0.00133	(1)0.006379 0.0047
UF8	(3)1.160744 0.16451	(2)0.577568 0.00795(+)	(1)0.429362 0.08932(+)	(5)3.073705 0.72738(+)	(4)2.858333 0.28673(+)	(6)5.694581 0.13567(+)
UF9	(3)0.771123 0.32922	(1)0.303064 0.0419(+)	(2)0.397282 0.04172	(5)4.546364 0.12622(+)	(4)1.666083 0.42798(+)	(6)0.074895 0.27739(+)
UF10	(3)1.042361 0.32582	(2)0.925704 0.33758	(1)0.39666 0.10734(+)	(4)1.926261 0.21823(+)	(6)6.433633 0.54881(+)	(5)3.949969 0.52153(+)
Average ranks	3.39	2.89	3.67	3.67	3.28	4.11

Bản quyền thuộc
 Tạp chí CNTT & TT

TABLE IV

THE RANK (AVERAGE VALUE OF IGD) AND STANDARD DEVIATION VALUES FOR EACH ALGORITHM. (+) INDICATES THAT THE DIFFERENCE BETWEEN IDMEA AND OTHER MOEAs IS SIGNIFICANT (USING T-TEST WITH $\alpha = 0.05$)

Problems	IDMEA	MOEA/D	MOEA/D-DE	NSGAI1	NSGAI1-DE	SPEA2
ZDT1	(1) 0.003558 0.00002	(2)0.003765 0.00001(+)	(4)0.004136 0.00009(+)	(6)0.005365 0.00022(+)	(5)0.004501 0.00028(+)	(3)0.003825 0.00021(+)
ZDT2	(4)0.00419 0.00005	(2)0.003942 0.00001(+)	(3)0.0041 0.00007(+)	(6)0.005389 0.00032(+)	(5)0.004533 0.00026(+)	(1) 0.003823 0.00022(+)
ZDT3	(4)0.008216 0.00006	(6)0.009299 0.00001(+)	(5)0.009168 0.00007(+)	(3)0.006261 0.00027(+)	(2)0.00607 0.00037(+)	(1) 0.004594 0.00028(+)
ZDT4	(3)0.004166 0.00069	(1) 0.003896 0.00015	(2)0.003965 0.00009	(5)0.054586 0.08894(+)	(6)0.077979 0.08876(+)	(4)0.044885 0.06456(+)
ZDT6	(3)0.013128 0.00001	(1) 0.003081 0.00002(+)	(2)0.003167 0.00024(+)	(6)0.014388 0.00017(+)	(5)0.013749 0.00021(+)	(4)0.013442 0.00014(+)
DTLZ2	(5)0.200311 0.06542	(4)0.038215 0.00017(+)	(6)0.287387 0.0004(+)	(3)0.040401 0.00211(+)	(2)0.037124 0.00188(+)	(1) 0.030672 0.00079(+)
DTLZ3	(1) 0.413976 0.13801	(4)0.452131 0.00012(+)	(5)0.523787 0.00059(+)	(6)0.534521 0.22421(+)	(2)0.443797 0.00238(+)	(3)0.445538 0.02807
DTLZ7	(1) 0.038833 0.01497	(6)2.630811 0.47013(+)	(5)2.440367 0.19182(+)	(4)2.309717 0.00322(+)	(3)2.308873 0.00452(+)	(2)1.057455 0.55163(+)
UF1	(1) 0.01202 0.00374	(6)0.123914 0.06439(+)	(5)0.064926 0.00164(+)	(3)0.045684 0.0146(+)	(2)0.042377 0.01228(+)	(4)0.055278 0.01871(+)
UF2	(1) 0.008078 0.00112	(6)0.042847 0.01712(+)	(5)0.029768 0.00031(+)	(4)0.0186 0.00266	(2)0.015293 0.00203(+)	(3)0.017456 0.00343(+)
UF3	(5)0.267 0.11157	(6)0.320518 0.05536	(1) 0.029823 0.03411(+)	(4)0.251112 0.05183	(3)0.247844 0.08995	(2)0.24529 0.06358
UF4	(1) 0.034687 0.00199	(6)0.085313 0.00479(+)	(5)0.058549 0.00675(+)	(3)0.036283 0.00158	(2)0.035261 0.00089(+)	(4)0.036382 0.0017(+)
UF5	(2)0.036908 0.0109	(6)0.184736 0.04882(+)	(5)0.113243 0.03029(+)	(3)0.038716 0.00375	(1) 0.034925 0.00328	(4)0.039982 0.00413(+)
UF6	(2)0.258278 0.15383	(6)0.609237 0.20551(+)	(3)0.267101 0.0812	(5)0.320063 0.13201	(1) 0.104944 0.04595(+)	(4)0.29912 0.093
UF7	(1) 0.008157 0.00132	(6)0.264516 0.20345(+)	(3)0.041148 0.00039(+)	(4)0.087453 0.12556(+)	(2)0.022305 0.0033(+)	(5)0.136817 0.15367(+)
UF8	(5)0.778199 0.22296	(2)0.128006 0.00129(+)	(1) 0.122289 0.00103(+)	(4)0.665988 0.21907	(6)1.036505 0.29149(+)	(3)0.598389 0.19417(+)
UF9	(5)0.202848 0.14098	(4)0.195354 0.13059	(3)0.193243 0.03263	(6)0.206027 0.06688	(1) 0.070979 0.0161(+)	(2)0.190566 0.0675
UF10	(4)0.453612 0.15468	(6)0.707529 0.35677(+)	(2)0.366985 0.03651(+)	(5)0.464939 0.27704	(1) 0.361609 0.11337(+)	(3)0.434427 0.15571
Average ranks	2.72	4.44	3.44	4.44	2.89	3.50

TABLE V

THE RANK (AVERAGE VALUE OF HYP) AND STANDARD DEVIATION VALUES FOR EACH ALGORITHM. (+) INDICATES THAT THE DIFFERENCE BETWEEN IDMEA AND OTHER MOEAs IS SIGNIFICANT (USING T-TEST WITH $\alpha = 0.05$)

Problems	IDMEA	MOEA/D	MOEA/D-DE	NSGA-II	NSGA-II-DE	SPEA2
ZDT1	(1)1.000000 0.00003	(2)0.999585 0.00001(+)	(5)0.998191 0.00022(+)	(6)0.997184 0.0003(+)	(4)0.998972 0.00021(+)	(3)0.999549 0.00006(+)
ZDT2	(1)1.000000 0.00001	(2)0.999542 0.00001(+)	(5)0.997409 0.00042(+)	(6)0.995538 0.0003(+)	(4)0.998485 0.00015(+)	(3)0.999011 0.00007(+)
ZDT3	(4)0.998823 0.00005	(5)0.997982 0.00004(+)	(6)0.996937 0.0005(+)	(3)0.999682 0.00011(+)	(2)0.999817 0.00016(+)	(1)1.000000 0.00007(+)
ZDT4	(2)0.999984 0.00163	(1)1.000000 0.0005	(3)0.999942 0.00061(+)	(5)0.996991 0.1255(+)	(6)0.995595 0.12516(+)	(4)0.997580 0.09134(+)
ZDT6	(3)0.860935 0.00001	(1)1.000000 0.00006(+)	(2)0.999849 0.00058(+)	(6)0.859987 0.00016(+)	(4)0.860637 0.00018(+)	(5)0.860253 0.00013(+)
DTLZ2	(5)0.967400 0.67391	(2)0.992243 0.00034(+)	(6)0.931695 0.00071	(4)0.984460 0.04731(+)	(3)0.988656 0.02372(+)	(1)0.993848 0.00326(+)
DTLZ3	(1)1.000000 0.17931	(2)0.996448 0.02636(+)	(6)0.996436 0.24168	(5)0.996446 0.33548(+)	(2)0.996448 0.32494(+)	(2)0.996448 0.78384(+)
DTLZ7	(4)0.902587 0.41984	(5)0.865833 0.2037	(6)0.808884 0.78652(+)	(3)0.938659 0.10408(+)	(2)0.939440 0.0799(+)	(1)1.000000 0.39924(+)
UF1	(1)0.998649 0.00747	(6)0.913480 0.45674(+)	(3)0.984840 0.00181(+)	(4)0.977471 0.15586(+)	(2)0.991844 0.06064(+)	(5)0.957355 0.20179(+)
UF2	(1)1.000000 0.00673	(6)0.748446 0.18892(+)	(2)0.987835 0.01807(+)	(4)0.779132 0.05664	(3)0.787253 0.02209(+)	(5)0.778964 0.0606
UF3	(2)0.973358 0.69025	(6)0.465220 0.39825(+)	(1)1.000000 0.32984(+)	(5)0.489033 0.60832(+)	(3)0.533404 0.60353(+)	(4)0.490834 0.51486(+)
UF4	(1)1.000000 0.00337	(6)0.413188 0.03411(+)	(2)0.966642 0.01582(+)	(5)0.466194 0.00427(+)	(3)0.485191 0.12385	(4)0.466244 0.00361(+)
UF5	(2)0.994899 0.96067	(6)0.682235 0.16353(+)	(6)0.875493 0.11361(+)	(4)0.933887 0.39681(+)	(1)1.000000 0.69284	(3)0.934920 0.58363(+)
UF6	(1)1.000000 1.07568	(6)0.289360 0.70966(+)	(2)0.639717 0.54559	(4)0.334982 0.70895(+)	(3)0.395412 0.34921(+)	(5)0.331448 0.69588(+)
UF7	(1)1.000000 0.00236	(6)0.834652 0.71561(+)	(3)0.987835 0.00318(+)	(4)0.942910 0.51499(+)	(2)0.995882 0.00606(+)	(5)0.907769 0.58755(+)
UF8	(4)0.998214 0.70687	(5)0.930923 0.0529(+)	(6)0.900919 0.77999(+)	(1)0.999940 0.49383(+)	(3)0.999050 0.6183	(2)0.999640 0.49547(+)
UF9	(4)0.995617 0.18544	(5)0.898024 0.20075(+)	(6)0.896711 0.25812(+)	(2)0.996987 0.18158(+)	(1)0.997234 0.1989(+)	(3)0.995871 0.22474(+)
UF10	(1)1.000000 0.62054	(5)0.968500 0.12719(+)	(6)0.967627 0.50178(+)	(3)0.990061 0.82967(+)	(2)0.998501 0.33404(+)	(4)0.989775 0.95831(+)
Average ranks	2.17	4.28	4.22	4.17	2.78	3.28

VI. CONCLUSIONS

In this paper, we introduced the new selection strategy for DMEA to control balance between convergence and spread. Experiments on 18 well-known benchmark problems with 7 well-known MOEAs have been carried out to investigate the performance and behavior of IDMEA. We compared its performance with all other 7 MOEAs on three popular metrics: GD, IGD and HYP. IDMEA with our new selection strategy showed to be competitive in comparison with these algorithms with respect to both solution convergence and spread. Several analyses on the behaviors of components of the algorithm were thoroughly investigated.

ACKNOWLEDGMENTS

We acknowledge the financial support from Vietnam's National Foundation for Science and Technology.

REFERENCES

- [1] H.A. Abbass. An evolutionary artificial neural network approach for breast cancer diagnosis. *Artificial Intelligence in Medicine*, 25(3):265–281, 2002.
- [2] H.A. Abbass, R.A. Sarker, and C.S. Newton. PDE: A pareto-frontier differential evolution approach for multi-objective optimization problems. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC2001)*, volume 2, pages 971–978, Piscataway, NJ, 2001. IEEE Press.
- [3] Coello Coello C.A. Mezura-Montes E.. Arias Montano, A. Mode-ld+ss: A novel differential evolution algorithm incorporating local dominance and scalar selection mechanisms for multi-objective optimization. In: *2010 IEEE Congress on Evol Comp (CEC2010)*, 2010.
- [4] L. T. Bui, H. A. Abbass, and D. Essam. Local models: An approach to distributed multi-objective optimization. *Journal of Computational Optimization and Applications*, Springer., pages 105–139, 2007.
- [5] Lam Thu Bui, Jing Liu, Axel Bender, Michael Barlow, Slawomir Wesolkowski, and Hussein A. Abbass. Dmea: a direction-based multiobjective evolutionary algorithm. *Memetic Computing*, pages 271–285, 2011.
- [6] Fan Yang Chung Kwan and Che Chang. A differential evolution variant of nsga ii for real world multiobjective optimization. *Proceeding ACAL'07 Proceedings of the 3rd Australian conference on Progress in artificial life*, pages 345–356, 2007.
- [7] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Trans. one Evolutionary Computation*, 8(3):256–279, 2004.
- [8] K. Deb. *Multiobjective Optimization using Evolutionary Algorithms*. John Wiley and Son Ltd, New York, 2001.
- [9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 6(2):182–197, 2002.
- [10] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization, TIK-Report no. 112. Technical report, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, 2001.
- [11] Kenneth Alan DeJong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, Ann Arbor, 1975.
- [12] J. Knowles and D. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
- [13] Hui Li and Qingfu Zhang. Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Trans. Evolutionary Computation*, pages 284–302, 2009.
- [14] Yu Chen Minzhong Liu, Xiufen Zou and Zhijian Wu. Performance assessment of dmoea-dd with cec 2009 moea competition test instances. *Proceeding CEC'09 Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, pages 2913–2918, 2009.
- [15] Long Nguyen, Lam Thu Bui, and Hussein Abbass. A new niching method for the direction-based multi-objective evolutionary algorithm. In *2013 IEEE Symposium Series on Computational Intelligence*, Singapore, 4 2013.
- [16] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. technical report tr-95-012. Technical report, ICSI, 1995.
- [17] D.A.V. Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovation*. PhD thesis, Department of Electrical Engineering and Computer Engineering, Airforce Institute of Technology, Ohio, 1999.
- [18] Q. F. Zhang and H. Li. Moea/d: A multi-objective evolutionary algorithm based on decomposition. 2007.
- [19] E. Zitzler, K. deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
- [20] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Periaux, K. D. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100. Int. Center for Numerical Methods in Engineering (Cmine), 2001.
- [21] E. Zitzler, L. Thiele, and K. Deb. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(1):173–195, 2000.

AUTHORS BIOGRAPHIES



Long NGUYEN has his B.Sc (2000) and M.Tech (2007) degree from Le Quy Don Technical University. Currently he is a PhD Candidate at the Faculty of Information Technology, Le Quy Don Technical University. His research interests are in the areas of Evolutionary Multi-objective Optimizations, Computational Intelligence, Multi-agent systems and their applications to defence and security domain.



Lam Thu BUI received the Ph.D. degree in computer science from the University of New South Wales, Kensington, Australia, in 2007. He did postdoctoral training at UNSW from 2007 until 2009.

He has been involved with academics including teaching and research since 1998. Currently, he is an Associate Professor and Deputy Dean of IT Faculty, Le Quy Don Technical University, Hanoi, Vietnam. He is researching in the field of evolutionary computation, specialized with evolutionary multiobjective optimization. He is the co-editor of the book *Multiobjective Optimization in Computational Intelligence: Theory and Practice* (IGI Global Information Science Reference Series); and the General Chair of the Ninth International Conference on Simulated evolution and Learning - SEAL2012.

Prof. Bui is a member of the Editorial Board, *International Journal of Computational Intelligence and Applications* (IJCIA), and the Vice-Chair of the Evolutionary Computation Technical Committee (ECTC), IEEE Computational Intelligence Society.