A New Niching Method for the Direction-based Multi-objective Evolutionary Algorithm

Long Nguyen
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
Email: longit76@gmail.com

Lam Thu Bui
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
Email: lam.bui07@gmail.com

Hussein Abbass
University of New South Wales
Australian Defence Force Academy
Canberra, Australia
Email: h.abbass@adfa.edu.au

***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 the performance of DMEA is also dependent on the way niching is implemented. In this paper, we propose a new niching approach for DMEA. The main idea of proposed approach is to define a new concept of ray-based density within the framework of DMEA and then use it as niching information. With this method, we hope to give more control on the balance between exploration and exploitation.**

To validate the performance of the new improved version of DMEA, we carried out a case study on several test problems and comparison with some other MOEAs, it obtained quite good results on primary performance metrics, namely the generation distance, inverse generation distance and hypervolume.

***Keywords*-Multi-objective Evolutionary Algorithms, Direction based EMO; EMO Performance Measurement; DMEA.**

I. Introduction

Approximating solutions of multi-objective optimization problems (MOPs) using evolutionary algorithms (EAs) has been a popular topic since EAs can offer simultaneously a set of trade-off solutions. Note that in the case of multi-objective minimization problems (MOPs), a solution is considered Pareto optimal if we can not find any feasible solution which would decrease some criterion without causing a simultaneous increase in at least one other criterion [6]. The set of solutions that satisfies the Pareto optimality definition is called the *Pareto optimal set*. Its image in objective space is known as the *Pareto optimal front* (POF). Mathematically, in a k-objective unconstrained (bound constrained) minimization problem, a vector function $\vec{f}(\vec{x})$ of k objectives is defined as:

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

In which $\vec{x}$ is a vector of decision variables in v-dimensional $\mathbb{R}^v$. In evolutionary computation (EC), $\vec{x}$ 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 jth objective in the MOP.

An individual $\vec{x}_1$ is said to *dominate* $\vec{x}_2$ 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 "$\ntriangleright$" between two scalars a and b to indicate that $a \ntriangleright b$ means a is not worse than b, then *dominance* can be formally defined as [8]. Most of the modern multi-objective evolutionary algorithms (MOEAs) are influenced by using the concept of Pareto dominance to assign fitness values to candidate solutions and then to select solutions for the production process.

Recently, the direction of improvement has been discussed and used to guide MOEAs during the search process towards POF [2],[3]. The authors in [4] introduced a novel algorithm incorporating the concept of direction of improvement, called Direction based Multi-objective Evolutionary Algorithm (DMEA). The uniqueness specific feature of DMEA is about the way of defining and using directional vectors as well as niching information. With DMEA, a population of solutions is evolved over time under the guidance of directions of improvement. The authors used two types of directions: Convergence direction (from a dominated solution to a non-dominated one)and Spread direction (between two non-dominated solutions) for generation of offsprings along those directions. Further, an archive is maintained over time. At each iteration, this archive is combined with the offspring population for forming a mixed population and then producing the next generation. In order to fill the population of the next generation, DMEA gets solutions from the combined population. Half of the population is filled the non-dominated solutions using niching information, while the other half is filled solutions using a weighted-sum technique for all remaining solutions in the combined population.

In DMEA, niching is implemented for both the archival population and the main one. While the archive is using an explicit niching in the objective space, the main population is forced to maintain the diversity of non-dominated solutions in the decision space. Our analysis showed that DMEA's niching scheme might make DMEA become off-balanced between exploration and exploitation. In this paper we propose a new niching approach for DMEA. The main idea of proposed approach is to define a new concept of ray-based density within the framework of DMEA and then use it as niching information.

To validate the proposed technique, we carried experiments on 17 problems from 3 well-known benchmark sets. We also make comparisons with 6 existing MOEAs on 3 performance metrics. The results strongly suggest that our new niching techniques made DMEA performs well in both convergence and solution spreading. The results indicate that new niching technique is suitable for making DMEA competitive.

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 technique is shown in Section IV. The experimental results is presented in Section V to examine the effectiveness and efficiency of proposed technique. Conclusion and future work are given in Section VI.

II. Multi-objective Evolutionary Algorithms

Multi-objective evolutionary algorithms (MOEAs) are stochastic techniques being used to find Pareto optimal solutions for a particular problem. The majority of existing MOEAs employs the concept of dominance; therefore, in our brief summary of MOEAs we focus mainly on this class of dominance-based MOEAs. 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 performance criteria when optimization algorithms are assessed and compared with each other [22].

To date, many MOEAs have been developed and there are several ways to classify them. Here we follow the classification of [6], in which MOEAs fall into two broad categories: Non-elitist Elitist approaches. 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 [17].

Elitist MOEAs usually (but not necessarily) employ an external set called the archive to store the non-dominated solutions after each generation. In general, when using an archive, there are two important aspects to consider [6]:

- *Interaction between archive and main population*: During the optimization process the archive can be combined with the current population to form the population for the next generation as in [21]. However, the archive is more than just a gene pool. It also contains information about the best performance of the algorithm so far. Exploiting this rich archival information should enhance the optimization process and is the main motivation for the research reported in this paper.
- *Updating the archive*: The method by which the archive is built also plays an important role. In some approaches the neighborhood relationship between individuals is used; e.g. in the form of geographical grid [12], crowded dominance [9], and clustering [21]. Others entail controlling the size of the archive through truncation when the number of non-dominated individuals exceeds a predefined threshold. In this paper we will pursue a different approach to maintaining the archive. Details will be given in the next section.

How archive and main population interact and how the archive is being updated differ from one MOEA to another. The general elitist principle is to preserve each generation's best individuals. This helps algorithms to get closer to the POF. A proof of convergence for MOEAs using elitist approach can be found in [16]. Algorithms such as Pareto Archived Evolution Strategy (PAES)[12], Strength Pareto EA 2 (SPEA2) [21], Pareto frontier DE (PDE)[2], NSGA-II [9], Decomposition based Multi-objective Evolutionary Algorithm (MOEA/D) [19] and Multi-Objective Particle Swarm Optimization (MOPSO) [7], MODE-LD+SS [14] and the Direction based Multi-objective Evolutionary Algorithm (DMEA)[4] are typical examples of elitist MOEAs.

III. Direction-based Multi-objective Evolutionary Algorithm - DMEA

To paraphrase the previous section, 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 mentioned above: *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 Figure 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.

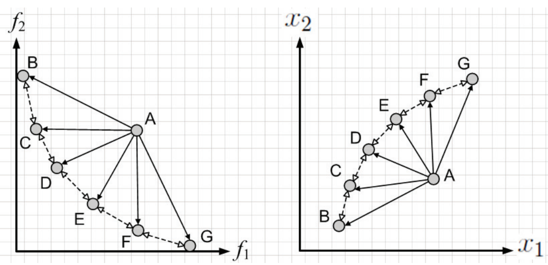

Figure 1. Illustration of convergence (black arrows) and spread (hollow arrows) directions in objective space (left) and decision variable space (right).

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 Figure 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 $f_i = f_{i,\min}$, $i \in \{1, 2, \ldots, k\}$ and described by $f_i \geq f_{i,\min} \forall i \in \{1, 2, \ldots, k\}$ where $f_{i,\min} \approx \min_{\text{all} A_1, A_2, \ldots} f_i$ with $A_1, A_2, \ldots$ being the solutions stored in the current archive. By their construction, the hyperquadrant contains the estimated POF.

During the archival update (insterting 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.

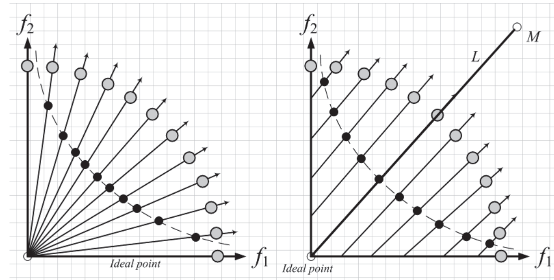

Figure 2. Illustration of the ray system in a 2-dim MOP. The left graph: origin of the bundle is collocated with the estimated ideal point. The ray bundle is bounded by the two lines $f_1 = f_{1,\min}$ and $f_2 = f_{2,\min}$ and it emits uniformly into the top right quadrant which contains the POF estimate. 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 [4] 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 [4].
    - 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. $M + A \to C$).
- **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.
    * $*$ If $m < n/2$, select (without replacement) all non-dominated solutions from C and copy to P.
    * $*$ Else,
      · 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.
  - 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.

There are some differences between DMEA and other MOEAs: (1)The selection of non-dominated solutions to fill the archive and the next population is assisted by a new technique of explicit niching in the objective space by using a system of straight lines or rays starting from the current estimation of the ideal point and dividing the space evenly. (2)An external archive of non-dominated solutions is maintained over time. DMEA is based on the effective use and refinement of information contained in the archive. The archive not only contribute solutions to the next generation, but also supports the derivation of directions for offspring production.

IV. Methodology

As indicated above, the main population is maintained in *Step 10* with two components (ideally their sizes are equal). The first component is composed of the non-dominated solutions, while the second component is composed of either dominated or non-dominated solutions. When selecting non-dominated solutions for the first component, DMEA uses a diversity-based criterion (being measured in the decision space). Note that the non-dominated solutions are drawn from the combined population which also contains solutions of the current archive. We also notice that the current archive is also maintained by a niching process (in the objective space). This means the non-dominated set in the main population is somehow affected by both niching schemes (in both spaces).

Further, through an analysis we found that, in some cases, DMEA strongly converges towards the central area of POF and non-dominated solutions are sparsely located at ends of POF. It means the solutions are not distributed uniformly on the entire POF. This affects the coverage of the obtained non-dominated solutions. It is largely because of the ray-based niching technique in DMEA. When considering non-dominated solutions for the archive, we scan the rays and find the closest solutions to rays respectively. As demonstrated in Figure 3, solution a is far from ray *r1*, but is still considered as the closest one since no other solutions closer to *r1* than solution a. In a long run, this might make the solutions grouped. To avoid this effect, it is necessary to include a concept of density when considering non-dominated solutions.

Our proposal is that we still keep ray-based niching for the archive, but when considering niching for the first component of the main population, we use the density information making sure the non-dominated solutions placed evenly in the objective space and discouraging grouped solutions. Our new measure is called 'Ray-based Density' counting the number of rays that a solution is the closest.

For each ray, we find a non-dominated solution that is closest to the ray. A solution may be the closest point for several rays, may be not be the closest for any rays. We call the number of rays being the closest as the ***Ray-based Density*** (see Figure 3); and replace the average Euclidean distance in DMEA by this density value.

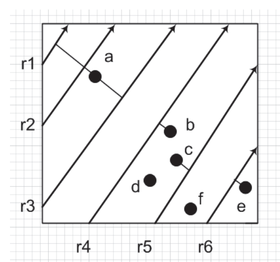

Figure 3. Illustration of the Ray-based Density: solution a has density value is 3 (closest to $r1, r2, r3$); solution b has density value is 1 (closest to only $r4$); solution d has crowded value zero (not closest to any rays).

A. Computational Complexity

In this technique we scan each ray and find a solution that is closest to it. This means requiring two loop when calculating density to all given rays. Hence the the computational complexity here is $O(n^2)$; and it has the same computational complexity as the original DMEA does.

V. Experiments

A. Testing problems

This paper considered a set of 17 continuous benchmark problems from 3 well-known benchmark sets, namely ZDT [20], DTLZ [10] and UF [15]. 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 benchmarks 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. The parameters for these problems are reported in Table I.

Table I
PARAMETER SETTINGS FOR THE EXPERIMENTS.

Test.P	N.Objs	N.Vars	Pop.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

B. Performance measurement methods

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 [22]. We will look at 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) [18]. The first-norm equation is defined as

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

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

$$IGD = \frac{\sum_{i=1}^{\overline{N}} \overline{d_i}}{\overline{N}} \tag{3}$$

where $\overline{d_i}$ is the Euclidean distance (in objective space) from solution *i* in the POS to the nearest solution in P, and $\overline{N}$ 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[22] 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 ant-optimal point or worst-possible point in objective space. In our experiments with 8 MOEAs with 17 test problems, we choosing the reference points by the 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 compact the HYPs to be $HYP^{*}s$ (in range [0,1]) by formula:

$$HYP_k^{*} = \frac{HYP_k}{\max\limits_{1 \le i \le N}(HYP_i)} \tag{4}$$

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

The experiments for proposed DMEA and existed MOEAs were carried out and used problems in Table I. We call the new improved version of DMEA as IDMEA. For DMEA and IDMEA experiments, the mutation rate was kept at the same small rate of 0.01, and the perturbation rate was a relatively small 0.4. Other MOEAs included: MOEA/D [19], MOEA/D-DE [13], NSGAII [9], NSGAII-DE [5], SPEA2 [21] on the same experimental environment. All algorithms are ran 30 times with different randomize seeds.

C. Results and Comparison

To analyze the performance of the new version IDMEA, we recorded all non-dominated solutions and calculate values of GD, IGD and HYP. These values were reported in Tables: II, III and IV.

First, we visualized all non-dominated solutions obtained by IDMEA and compare them to that of DMEA (see some typical snapshots in Figures 4 to 8). The results clearly show the better performance of IDMEA over DMEA. 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 quite dense in comparison to others; typically the central area of POF. In contrast, for IDMEA, the final solutions are distributed uniformly along POF. This shows the strong effect of our new niching scheme: the use of density values pushed solutions spreading along the area of POF. Further, we have summary results with comparison to other MOEAs with three metrics in tables:II, III and IV with average ranks on each metric. In comparison to

DMEA, our new IDMEA is always ranked above the original one.

With regards to other MOEAs, the experimental results were categorized as follows:

- GD: On Table II, we see that IDMEA obtained quite comparable GD values. It was ranked better than DMEA, MOEA/D-DE, NSGAII, and SPEA2. Also it was the best on test problems: ZDT3, DTLZ7 and UF7.
- IGD: On Table III, IDMEA demonstrated the better performance with the highest rank (averaged rank 3.17) among MOEAs. it was the best on test problems: ZDT1, DTLZ7, UF1, UF2 and UF7.
- HYP: On Table IV we also have the similar results for IDMEA. It got the best averaged rank (2.78) with 6 times getting the best HYP values (on test problems: ZDT1, ZDT2, UF2. UF4. UF6 an UF7).

The above finding on GD, IGD and HYP again shows the strength of our new design on niching for DMEA. This new technique 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.

D. *Behavior of the algorithm over time*

To get a full understanding of our new design, we also analyze the behavior of IDMEA over time. There are several ways for understanding the behavior of IDMEA; here we decided to recorded the values of GD and IGD over time and plotted them on a time dependent graph.

With almost test problems (as examples, they were all visualized in Figures: 9 and 10 on GD and IGD metrics with ZDT1, and ZDT3 and for both DMEA and IDMEA), we found that:

- For the results on the GD metric: There is a common pattern that At the eirlier stage (first several generations), DMEA is 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.
- For the results on the IGD metric: IDMEA got better IGD values all the time.

Through the above comparison using our experimental results, we see that by applying our new technique, DMEA's performance is greatly improved, especially the spacing of non-dominated solutions (being shown via IGD values).

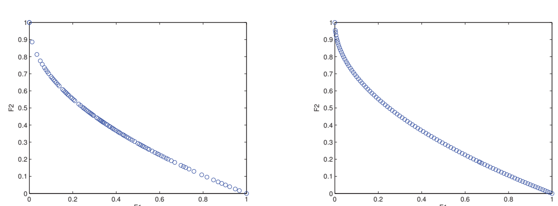

Figure 4. ZDT1 results in DMEA and proposed IDMEA in the objective space.

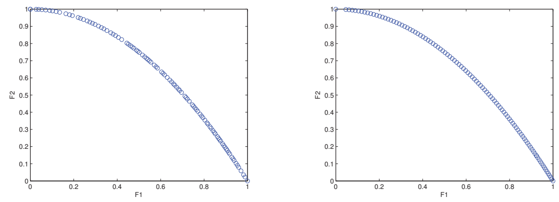

Figure 5. ZDT2 results in DMEA and proposed IDMEA in the objective space.

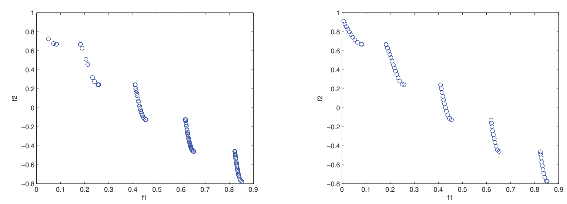

Figure 6. ZDT3 results in DMEA and proposed IDMEA in the objective space.

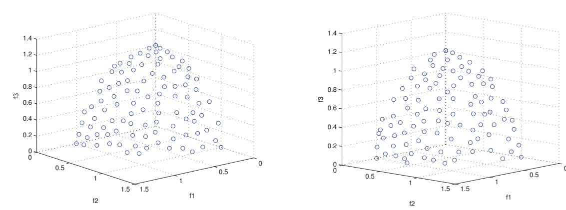

Figure 7. DTLZ2 results in DMEA and proposed IDMEA in the objective space.

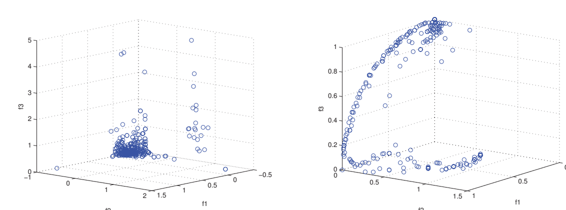

Figure 8. UF8 results in DMEA and proposed IDMEA in the objective space.

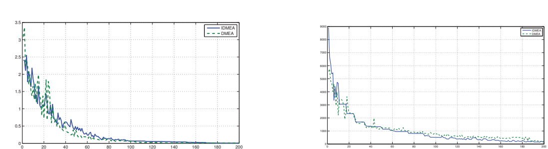

Figure 9. Visualization of GD (left) and IGD (right) over time for ZDT1.

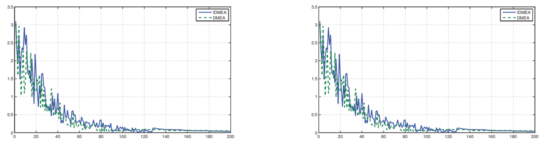

Figure 10. Visualization of GD (left) and IGD (right) over time for ZDT3.

VI. Conclusions

In this paper, we introduced a new niching technique to improve performance of DMEA. This technique used ray-based density information to support selection of non-dominated solutions during preparation of the main population for the next generation.

Experiments on 17 well-known benchmark problems with 7 well-known MOEAs have been carried out to investigate the performance and behavior of the new niching techniques to improve DMEA. We compared its performance with all other 7 MOEAs on three metrics: GD, IGD and HYP. DMEA with new control techniques 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 Evol. Compt (CEC2001)*, volume 2, pages 971–978, Piscataway, NJ, 2001. IEEE Press.

[3] L. T. Bui, H. A. Abbass, and D. Essam. Local models: An approach to disibuted multi-objective optimization. *Journal of Comp. Opt. and Apps, Springer. Accepted to appear*, 2007.

[4] 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.

[5] 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.

[6] C. A. C. Coello. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Comp. Intell. Mag*, 1(1):28–36, 2006.

[7] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Trans. one Evol. Comp*, 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 Evol. Comp*, 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. *Evol. Comp*, 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. Evol. Comp*, pages 284–302, 2009.

[14] Arias Montano A. Coello Coello C.A. Mezura-Montes. 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.

[15] 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 11th conference on Congress on Evol. Comp*, pages 2913–2918, 2009.

[16] G. Rudolph and A. Agapie. Convergence properties of some multi-objective evolutionaryalgorithms. In *Proceedings of the Congress on Evol. Comp*, pages 1010–1016. IEEE Press, 2000.

[17] 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.

[18] D.A.V. Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovation*. PhD thesis, Department of Electrical Engineering and Computer Engineering, Airforce Institue of Technology, Ohio, 1999.

[19] Q. F. Zhang and H. Li. Moea/d: A multi-objective evolutionary algorithm based on decomposition. 2007.

[20] E. Zitzler, K. deb, and L. Thiele. Comparison of multi-objective evolutionary algorithms: empirical results. *Evol. Comp*, 8(2):173–195, 2000.

[21] 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. CMINE, 2001.

[22] E. Zitzler, L. Thiele, and K. Deb. Comparision of multi-objective evolutionary algorithms: Emprical results. *Evol. Comp*, 8(1):173–195, 2000.

Table II
THE RANK (AVERAGE VALUE OF GD) FOR EACH ALGORITHM.

Problems	**IDMEA**	**DMEA**	**MOEA/D**	**MOEA/D-DE**	**NSGA-II**	**NSGA-II-DE**	**SPEA2**
ZDT1	(4)0.004866	(1)0.003625	(2)0.003766	(3)0.00413	(7)0.00532	(5)0.005186	(6)0.005302
ZDT2	(6)0.004343	(4)0.004301	(5)0.004303	(7)0.004422	(3)0.004273	(2)0.004133	(1)0.004105
ZDT3	(1)0.003888	(4)0.005564	(2)0.004884	(3)0.005129	(5)0.006076	(6)0.0062	(7)0.006349
ZDT4	(7)0.009115	(6)0.008269	(1)0.003898	(2)0.003971	(3)0.056643	(5)0.077546	(4)0.074256
ZDT6	(6)0.004791	(7)0.004838	(1)0.00325	(3)0.003322	(4)0.003942	(2)0.003273	(5)0.00422
DTLZ2	(6)0.333037	(7)0.366645	(1)0.066486	(2)0.075109	(4)0.081473	(3)0.080561	(5)0.08225
DTLZ3	(4)0.284596	(5)0.33232	(1)0.060372	(3)0.075961	(6)0.635451	(2)0.062155	(7)1.380131
DTLZ7	(1)0.06427	(6)1.200727	(2)0.133183	(5)0.199028	(4)0.148025	(3)0.143845	(7)2.037204
UF1	(5)0.011741	(2)0.008775	(6)0.027819	(7)0.067801	(3)0.011192	(4)0.011549	(1)0.007646
UF2	(2)0.00771	(1)0.007396	(6)0.026468	(7)0.034738	(5)0.011934	(3)0.010039	(4)0.010162
UF3	(6)0.104425	(7)0.110821	(5)0.076694	(1)0.02453	(3)0.068729	(2)0.06703	(4)0.074826
UF4	(1)0.034826	(2)0.035069	(6)0.046091	(7)0.060091	(5)0.038245	(3)0.037238	(4)0.037543
UF5	(4)0.272366	(3)0.222777	(7)0.697828	(6)0.564877	(1)0.131011	(5)0.315271	(2)0.135676
UF6	(6)0.243736	(5)0.243305	(3)0.090533	(7)0.306746	(2)0.087615	(1)0.052228	(4)0.111452
UF7	(2)0.006428	(5)0.007284	(6)0.019532	(7)0.043162	(3)0.006533	(4)0.006895	(1)0.006379
UF8	(3)1.264394	(4)1.485403	(2)0.577568	(1)0.429362	(6)3.073705	(5)2.858333	(7)5.694581
UF9	(3)0.606963	(5)2.335798	(1)0.303064	(2)0.397282	(6)4.546364	(4)1.666083	(7)6.074895
UF10	(3)1.150331	(4)1.225281	(2)0.925704	(1)0.39666	(5)1.926261	(7)6.433633	(6)3.949969
Averaged Ranks	3.89	4.33	**3.28**	4.11	4.17	3.67	4.56

Table III
THE RANK (AVERAGE VALUE OF IGD) FOR EACH ALGORITHM.

Problems	**IDMEA**	**DMEA**	**MOEA/D**	**MOEA/D-DE**	**NSGAII**	**NSGAII-DE**	**SPEA2**
ZDT1	(1)0.00356	(6)0.004847	(2)0.003765	(4)0.004136	(7)0.005365	(5)0.004501	(3)0.003825
ZDT2	(4)0.004188	(5)0.004239	(2)0.003942	(3)0.0041	(7)0.005389	(6)0.004533	(1)0.003823
ZDT3	(4)0.008241	(7)0.011544	(6)0.009299	(5)0.009168	(3)0.006261	(2)0.00607	(1)0.004594
ZDT4	(3)0.008046	(4)0.0092	(1)0.003896	(2)0.003965	(6)0.054586	(7)0.077979	(5)0.044885
ZDT6	(3)0.013128	(4)0.013425	(1)0.003081	(2)0.003167	(7)0.014388	(6)0.013749	(5)0.013442
DTLZ2	(6)0.185108	(5)0.052655	(3)0.038215	(7)0.287387	(4)0.040401	(2)0.037124	(1)0.030672
DTLZ3	(2)0.435732	(1)0.338842	(5)0.452131	(6)0.523787	(7)0.534521	(3)0.443797	(4)0.445538
DTLZ7	(1)0.049834	(3)1.401515	(7)2.630811	(6)2.440367	(5)2.309717	(4)2.308873	(2)1.057455
UF1	(1)0.011328	(2)0.011829	(7)0.123914	(6)0.064926	(4)0.045684	(3)0.042377	(5)0.055278
UF2	(1)0.007636	(2)0.009971	(7)0.042847	(6)0.029768	(5)0.0186	(3)0.015293	(4)0.017456
UF3	(6)0.286392	(5)0.276663	(7)0.320518	(1)0.029823	(4)0.251112	(3)0.247844	(2)0.24529
UF4	(3)0.03547	(1)0.034973	(7)0.085313	(6)0.058549	(4)0.036283	(2)0.035261	(5)0.036382
UF5	(2)0.037113	(7)0.529945	(6)0.184736	(5)0.113243	(3)0.038716	(1)0.034925	(4)0.039982
UF6	(3)0.263039	(2)0.222589	(7)0.609237	(4)0.267101	(6)0.320063	(1)0.104944	(5)0.29912
UF7	(1)0.007884	(2)0.010739	(7)0.264516	(4)0.041148	(5)0.087453	(3)0.022305	(6)0.136817
UF8	(4)0.805489	(6)0.905205	(2)0.128006	(1)0.122289	(5)0.665988	(7)1.036505	(3)0.598389
UF9	(7)0.215859	(2)0.1347	(5)0.195354	(4)0.193243	(6)0.206027	(1)0.070979	(3)0.190566
UF10	(5)0.526397	(6)0.63601	(7)0.707529	(2)0.366985	(4)0.464939	(1)0.361609	(3)0.434427
Averaged Ranks	**3.17**	3.89	4.94	4.11	5.11	3.33	3.44

Table IV
THE RANK (AVERAGE VALUE OF HYP*) FOR EACH ALGORITHM.

Problems	**IDMEA**	**DMEA**	**MOEA/D**	**MOEA/D-DE**	**NSGA-II**	**NSGA-II-DE**	**SPEA2**
ZDT1	(1)1.000000	(6)0.997404	(2)0.999602	(5)0.998209	(7)0.997202	(4)0.998990	(3)0.999567
ZDT2	(1)1.000000	(3)0.999040	(2)0.999571	(6)0.997438	(7)0.995567	(5)0.998515	(4)0.999040
ZDT3	(4)0.998830	(7)0.995634	(5)0.997982	(6)0.996937	(3)0.999682	(2)0.999817	(1)1.000000
ZDT4	(3)0.999748	(4)0.999674	(1)1.000000	(2)0.999942	(6)0.996991	(7)0.995595	(5)0.997580
ZDT6	(3)0.860936	(4)0.860705	(1)1.000000	(2)0.999849	(7)0.859987	(5)0.860637	(6)0.860254
DTLZ2	(6)0.955526	(1)1.000000	(3)0.992243	(7)0.931695	(5)0.984460	(4)0.988656	(2)0.993848
DTLZ3	(5)0.999990	(1)1.000000	(1)1.000000	(7)0.999988	(6)0.999998	(1)1.000000	(1)1.000000
DTLZ7	(5)0.885958	(4)0.934889	(6)0.865833	(7)0.808884	(3)0.938659	(2)0.939440	(1)1.000000
UF1	(2)0.998607	(1)1.000000	(7)0.913480	(4)0.984840	(5)0.977471	(3)0.991844	(6)0.957355
UF2	(1)1.000000	(3)0.789999	(7)0.748515	(2)0.987926	(5)0.779204	(4)0.787326	(6)0.779036
UF3	(2)0.971189	(3)0.545415	(7)0.465220	(1)1.000000	(6)0.489033	(4)0.533404	(5)0.490834
UF4	(1)1.000000	(4)0.469768	(7)0.412871	(2)0.965900	(6)0.465836	(3)0.484819	(5)0.465886
UF5	(2)0.997652	(3)0.965000	(7)0.682235	(6)0.875493	(5)0.933887	(1)1.000000	(4)0.934920
UF6	(1)1.000000	(4)0.580872	(7)0.449284	(2)0.993278	(5)0.520121	(3)0.613950	(6)0.514634
UF7	(1)1.000000	(2)0.999160	(7)0.834573	(4)0.987742	(5)0.942821	(3)0.995788	(6)0.907683
UF8	(5)0.998685	(1)1.000000	(6)0.930923	(7)0.900919	(2)0.999941	(4)0.999050	(3)0.999640
UF9	(5)0.995868	(1)1.000000	(6)0.898024	(7)0.896711	(3)0.996987	(2)0.997234	(4)0.995871
UF10	(2)0.999196	(1)1.000000	(6)0.968748	(7)0.967875	(4)0.990314	(3)0.998757	(5)0.990028
Averaged Ranks	**2.78**	2.94	4.89	4.67	5	3.33	4.06