

A new balanced direction for multi-objective evolutionary algorithms

Nguyen Long⁽¹⁾, Nguyen Xuan Hung⁽²⁾, Nguyen Thi Hien⁽²⁾, Bui Thu Lam⁽²⁾

(1) National Defense Academy, Hanoi, Vietnam

(2) Le Quy Don Technical University, Hanoi, Vietnam

Emails: longit76@mta.edu.vn, hungk18@outlook.com, {hien_cpqn, lam.bui}@mta.edu.vn

Abstract - This paper suggests to use a new improvement direction for multi-objective evolutionary algorithms. In DMEA-II, two improvement directions (convergence and spread) are used for the guidance during evolutionary processes. Based on those directions and the balance between exploration and exploitation, we determined a new improvement direction to keep DMEA-II to be better on the balance of convergence and diversity.

To validate the performance of the new improved version of DMEA-II, we carried out a case study on several test problems and comparison with well-known MOEAs, it obtained quite good results on primary performance metrics, namely the generation distance, inverse generation distance and hypervolume. Our analysis on the results indicates that, the usage of proposed direction may make DMEA-II to be improved in keeping balanced between convergence and diversity at each generation during the search.

keywords: moea, dmea-ii, directions of improvement, niching, ray based density, convergence direction, spread direction.

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 minimized problems, a solution is considered a Pareto optimal if we cannot find any feasible solution which would decrease some criterion without causing a simultaneous increase in at least one other criterion [8]. 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:

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

In which \vec{x} is a vector of decision variables in v -dimensional \mathbb{R}^v . In evolutionary 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 j th 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 “ $\not\preceq$ ” 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 [9]. 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. Working on population, evolutionary algorithms have been extensively used to solve multi-objective optimization problems. However, most of the current multi-objective evolutionary algorithms (MOEAs) cannot strike a good balance between the closeness to the true Pareto front and the uniform distribution of non-dominated solutions. To resolve this issue, researchers discussed and proposed techniques to keep the balancing between convergence and diversity of MOEAs and it has still been a hot topic on MOEA research area.

Recently, the direction of improvement has been discussed and used to guide MOEAs during the search process towards POF PDE [1], DMEA [2], DMEA-II[17]. DMEA-II is a new version of DMEA, some new features were introduced: (1) an adap-

tation of the balance between convergence and spreading by using an adaptive ratio between the convergence and spreading directions being selected over time; (2) a new concept of ray-based density for niching; and (3) a new selection scheme based on the ray-based density for selecting solutions for the next generation.

DMEA-II uses an adaptive ratio between convergence and spreading directions to control the balance between exploration and exploitation of the evolutionary process: There are two types of direction of improvement for guiding the search process in DMEA-II: convergence and spread directions. The convergence direction (CD) is defined as the direction from a solution to a better one, CD in MOPs is a 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 MOPs, a dominated solution guided by this direction is more likely to find a better area in the decision space than an unguided solution. The spread direction (SD) is defined as the direction between two equivalent solutions, SD in MOPs is a vector that points from one non-dominated solution to the other non-dominated one. If solutions are perturbed along SD, a better spread within the population should be obtained. In DMEA-II, the authors suggest a thought on an adaptation of the balance between convergence and spreading. Instead of using a fixed ratio of 0.5 as in DMEA, an adaptive ratio based on the number of non-dominated solution in the archive at current generation was suggested to used.

Focus on the usage of directions of improvement, our analysis showed that, each individual is generated by perturbation on a selected direction (CD or SD), it depends on the adaptive ratio at current generation. In those cases, the generating of new individual is only effected by the selected direction, it means the new individual might lacked the effectiveness of remaining direction. We hypothesised that, if an individual is effected by both directions, it will make the population to be balanced between convergence and diversity better. From this idea, we suggest to use a new improvement direction namely balanced directions, which is determined from both CD and SD directions and the balance between exploration and exploitation.

To validate the proposed technique, we carried out experiments on 10 problems from 3 well-known

benchmark sets. We also make comparisons with origin and a modified version of DMEA-II and 3 well-known MOEAs on 3 performance metrics. The results suggest that our new improvement direction made DMEA -II performs well in both convergence and spreading. The results indicate that new improvement direction is suitable to used in DMEA-II.

The remainder of this paper is organized as follows. A brief of improvement direction concept is given in Section II and the description of DMEA-II in Section III. Detail of our proposed technique is shown in Section IV. The experimental results are presented in Section V to examine the effectiveness and efficiency of proposed technique. Conclusion and future work are given in Section VI.

II DIRECTION OF IMPROVEMENTS

The effectiveness of using directional information has been discussed in multi-objective evolutionary algorithmic design. Beside of gradient based technique and differential evolution to guide MOEAs, the use of directions of improvement is raised and indicated as a promising technique to obtain set of good approximated Pareto optimal solutions. This technique helps MOEAs to maintain exploitation and exploration characteristics. Following the aspect of improving MOEAs, directions of improvement are defined as directions which are determined from population in decision space or objective space. These directions are archived and used for moving solutions towards the directions to make MOEAs to be improved in convergence rate and diversity. There are some proposals which use directions of improvement:

In [13], based on the concept of differential evolution, the authors propose the usage of directions of improvement for two important characteristics of MOEAs: convergence and diversity. In this proposal, two types of direction are defined for NSGA-II are directional convergence and directional spread, those directions are used to guide the evolutionary processes during generations:

In [5], [3], [4] the authors introduce a guidance technique for multi-objective algorithm in both decision space and objective space in a local search model. In this proposal, decision space is divided into several non-overlap spheres, each sphere is defined that it includes a pair of a centroid and a radius. Each sphere is defined as a local area in decision space. In this proposal, the movement

direction of a sphere is determined by the vector which is created from a pair of an old centroid and a new one, this direction is defined as a direction of improvement. The combined direction of the system is defined as a combination of all directions of improvement of the spheres.

Based on the concept of particle swarm optimization (PSO) rules [12], a new individual is generated by an adaption of using the vector of a direction. Directions of improvement are created by movement direction of spheres, they are used to guide the evolutionary process towards difference areas in POFs in objective space, it helps to maintain the diversity of the population during generations.

The authors in [6], [17] introduce a direction based multi-objective evolutionary algorithm (DMEA) and next version DMEA-II, two types of improvement directions are used. These directions of improvement are defined as follows:

Convergence direction: This direction is defined as a direction from a solution to a better solution in objective space, in MOPs, the convergence direction is a direction from a dominated solution to a non-dominated solution. If non-dominated solutions are maintained globally, this direction is global direction. In unconstraint MOPs, a guidance to move a dominated solution towards this direction helps the evolutionary process to find better area in decision space. An illustration of convergence direction is shown in Fig.1, A is a dominated solution, B, C, D, E, F, G are non-dominated solutions, all directions from A to B, C, D, E, F, G are determined as convergence directions.

Spread direction: This direction is defined as a direction from two equivalence solutions, in MOPs, the spread direction is a direction from a non-dominated solution to other non-dominated solution. If a solution is perturbed by this direction, it helps the population to be more spreading, in other words, the obtained solutions will be distributed towards the POF in objective space. An illustration of convergence direction is shown in Fig.1, all directions between B, C, D, E, F, G are determined as spread directions.

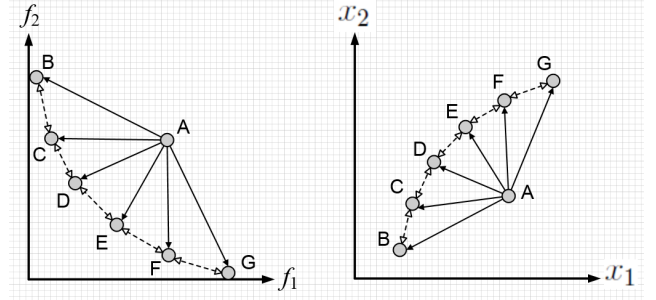


Fig. 1: Illustration of convergence (black arrows) and spread (hollow arrows) directions in objective space (left) and decision variable space (right).

These directions of improvement are used as follows:

A dominated solution Par is selected with variables $Par(i)$ in decision space $i \in \{1, 2, \dots, n\}$, a perturbation rate $0 < p < 1$ and a convergence d_1 , an individual $S_1(i)$ is perturbed by convergence direction:

$$S_1(i) \equiv Par(i) + RND_i(p)\sigma_1 d_1(i). \quad (2)$$

σ_1 is a scalar parameter which is randomly picked in a distribution $U(0, 2)$, this is a step-length to avoid local optimums, $RND_i(p)$ is a randomized number in distribution $U(0, 1)$, it equals 1 if $U(0, 1) < p$ and 0 otherwise. A non-dominated solution Par is selected with variables $Par(i)$ is perturbed by a spread direction d_2 :

$$S_2(i) \equiv Par(i) + RND_i(p)\sigma_2 d_2(i). \quad (3)$$

In this proposal, directions of improvement are calculated by:

$$d_1 \equiv \frac{A_1 - Par}{|A_1 - Par|} \quad (4)$$

$$d_2 \equiv \frac{A_2 - A_3}{|A_2 - A_3|} \quad (5)$$

there, A_1, A_2 and A_3 are three randomly selected solutions in the archive (an external population is used to keep elitist solutions during generations). Those directions are used to effectively guide the evolutionary process for maintaining the exploitation and exploration of the algorithm.

Through the survey of using improvement directions, we found that, a new individual are generated from the perturbation by only one improvement direction on different selection techniques. We assumed that the new individual is lacked the effectiveness of remaining directions. It is might make new population is imbalanced between convergence

and diversity. If the perturbation is effected by both directions simultaneously, then, the new individual will be effected by all directions, it will make the new population to be balanced between convergence and diversity better.

DMEA-II is introduced in 2014 [17] which is a typical MOEA of direction based approach, we will discuss above idea in using directions of improvement on this algorithm. Then, we designed a new improvement direction which is determined from both directions and suggested to used to improve the algorithm's performance. The details of the technique are shown in next section.

III METHODOLOGY

A. Balanced directions

In DMEA-II, at each generation an adaptive ratio is calculated to determine number of new individuals are generated belong each directions of improvement by perturbations 2, 3. In our design, we used this adaptive ratio with CD and SD (as temporary directions) to determine a new direction and perturb individuals on it in 3 steps at each generation as follows:

- Step 1: Calculate CD (d_1) and SD (d_2) as equation 4 and 5. These directions are temporary kept to calculate a new direction.
- Step 2: A dominated solution Par is selected with variables $Par(i)$ in decision space $i \in \{1, 2, \dots, n\}$, a perturbation rate $0 < p < 1$ and a balanced direction d . Calculate a new direction d from CD and SD with the adaptive ratio as follows: We have two experimental cases: the first, the adaptive ratio (AR) is not used, the new direction (d) is determined as a direction which is created between CD (d_1) and SD (d_2) (see Fig 2). The second, we use the adaptive ratio to calculate an angle to rotate the direction d to have a new direction d' . We call α is the angle between d_1 and d_2 , α_1 is the angle between d_1 and new direction d' , α_2 is the angle between d_2 and new direction, then we can calculate:

$$\alpha_1 = AR * \alpha \quad (6)$$

$$\alpha_2 = (1 - AR) * \alpha \quad (7)$$

Then, we use d and α_1 or α_2 ($\alpha_1 + \alpha_2 = \alpha$) to calculate the rotate angle $\Delta\alpha_r$:

$$\Delta\alpha_r = \frac{\|\alpha_2 - \alpha_1\|}{0.5} * (-1)^{rd} \quad (8)$$

Here, rd is equal to 1 when number of non-dominated solutions in current population greater than a half of population size, otherwise rd equal to 0. This parameter is used to calculate the direction for d to rotate. After the rotation, we get the new direction d' (see Fig 2). The new direction of improvement (d in the first case, d' in the second one) we called a **balanced direction**.

- Step 3: An individual $S(i)$ is perturbed by the balanced direction:

$$S(i) \equiv Par(i) + RND_i(p)\sigma d'(i). \quad (9)$$

Here, σ is a scalar parameter which is randomly picked in a distribution $U(0, 2)$, this is a step-length to avoid local optimums, $RND_i(p)$ is a randomized number in distribution $U(0, 1)$, it equals 1 if $U(0, 1) < p$ and 0. In the first case, we use d instead of d' in equation 9.

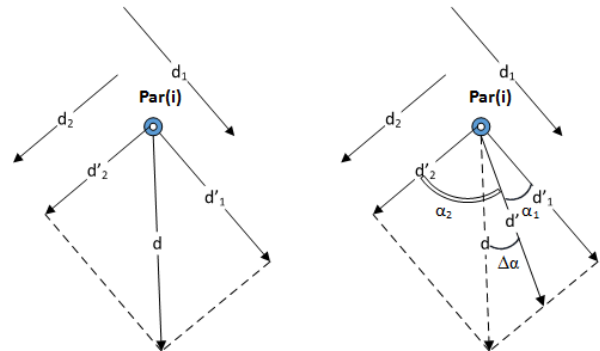


Fig. 2: An illustration of balanced directions with adaptive ratio (right) and without adaptive ratio (left).

B. Computational Complexity

In this technique we still use CD and SD as temporary variables to determine the balanced direction, so the computational complexity is not changed. Hence the the computational complexity here is $O(n^2)$.

IV EXPERIMENTS

A. Testing problems

This paper considered a set of 18 continuous benchmark problems from 3 well-known benchmark

sets, namely ZDT [22], DTLZ [11] and UF [16]. 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.

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 is employed [22]. We will look at three popular criteria: the generational distance (GD), the inverse generational distance (IGD) and hypervolume (HYP).

The measure of GD 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

$$GD = \frac{\sum_{i=1}^n d_i}{n} \quad (10)$$

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

$$IGD = \frac{\sum_{i=1}^{\bar{N}} \bar{d}_i}{\bar{N}} \quad (11)$$

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[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 point normally is the ant-optimal point or worst-possible point in objective space. In our experiments with 6 MOEAs with 18 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:

$$HYP_k^* = \frac{HYP_k}{\max_{1 \leq i \leq N} (HYP_i)} \quad (12)$$

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

We considered a large set of 18 continuous benchmark problems from 3 well-known benchmark sets, namely ZDT [22], DTLZ [11] and UF [16]. For these problems, the number of variables is 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*. The benchmarks ZDT, DTLZ are used in many proposals [10], [19], [6], [20] and the benchmark set UF is used in the CEC 2009 for the MOEAs contest. It confirms that selected benchmark sets are widely analyzed in the literature. The features for these test problems are shown in Table I.

Table II: Common parameter settings for the experiments.

Problems	Objs/Vars	P. size	N. gens	N. runs	POF
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

Table I: The main features of test problems

Sets	Problems	N. Objs	POF
ZDT	ZDT1	2	Convex.
	ZDT2	2	Concave.
	ZDT3	2	Disconnected and convex.
	ZDT4	2	Multi-modal.
	ZDT6	2	Non-convex.
DTLZ	DTLZ1	3	Separable, multi-modal.
	DTLZ2	3	Concave, scalable, multi-modal.
	DTLZ3	3	Concave.
	DTLZ7	3	Disconnected.
UF	UF1	2	Concave.
	UF2	2	Concave.
	UF3	2	Concave.
	UF4	2	Convex.
	UF5	2	Disconnected.
	UF6	2	Multi-modal.
	UF7	2	Convex.
	UF8	3	Convex.
	UF9	3	Convex, multi-modal.
	UF10	3	Convex.

The common testing parameters for these problems are reported in Table II. The experiments were carried out on all problems with parameters settings in Table II. For DMEA-II and modified versions 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-DE [15] using differential evolution (DE) operators, NSGA-II-DE [7], SPEA2 [20] on the same experimental environment. All algorithms are run 30 times with different random seeds. The detailed parameters settings for compared MOEAs are showed in Table III.

C. Results and Discussion

To analyze the performance of the algorithms, we recorded all non-dominated solutions and calculate values of popular performance measurements: GD, IGD and HYP. These values were reported in Tables: IV, V and VI.

The first, we visualized all non-dominated solutions obtained by DMEA-II and compare them to that of two versions (DMEA-II-NAD: a modified version of DMEA-II which uses the balanced directions without adaptive method; DMEA-II-AD: a modified version of DMEA-II which uses adaptive balanced directions) and three popular MOEAs: NSGA-II-DE [7], MOEA/D-DE [14], SPEA2 [20]. In this set, MOEA/D-DE, NSGA-II-DE are two modified versions of MOEA/D [19] and NSGA-II [10]. SPEA2 [20] an elitist MOEA version of original SPEA [21] which is employing an external population (archive) to preserve primarily non-dominated solutions. This population is combined with the current population to form the next archive for creating offspring for the next generation. Some typical snapshots in Figures 3, 4, 5, 6).

The results clearly show the better performance of DMEA-II-AD over DMEA-II and DMEA-II-NAD and it is competitive with other MOEAs. With DMEA-II, some parts of POFs are quite dense in comparison to MOEAs; typically the central area of POF. In contrast, for DMEA-II-AD, the final solutions are distributed uniformly along POF.

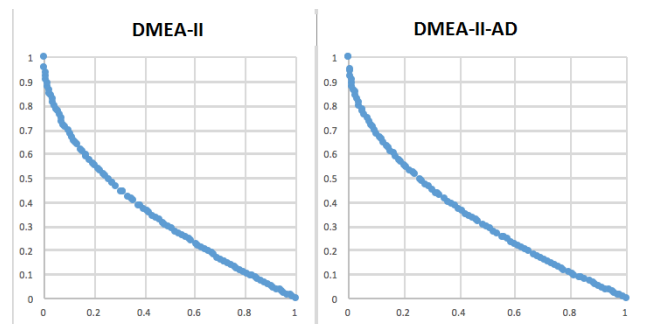


Fig. 3: ZDT1 results in DMEA-II and DMEA-II-AD in the objective space.

Table III: Parameters settings for compared MOEAs in the experiments

Parameters	DMEA-II, DMEA-II-NAD, DMEA-II-AD	MOEA/D-DE	NSGA-II-DE	SPEA2
Rdm.seed	[0-1]	[0-100]	[0-1]	[0-1]
Prob.xov	0.4	N/A	0.4	0.4
Prob.mut	0.01	N/A	0.01	0.01
Dist.idx.xov	0.4	20	0.4	0.4
Dist.idx.mut	0.5	20	0.5	0.5
Neigh.size	N/A	20	N/A	N/A
Prob.sel.mat	N/A	0.9	N/A	N/A

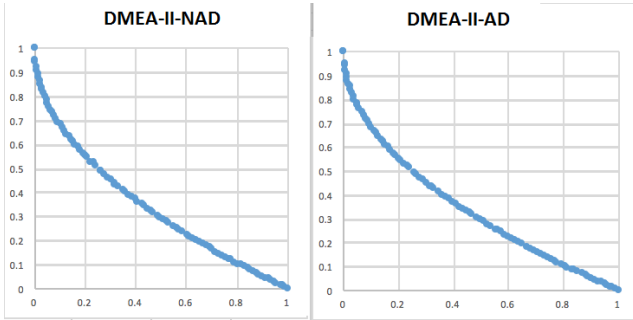


Fig. 4: ZDT1 results in DMEA-II-NAD and DMEA-II-AD in the objective space.

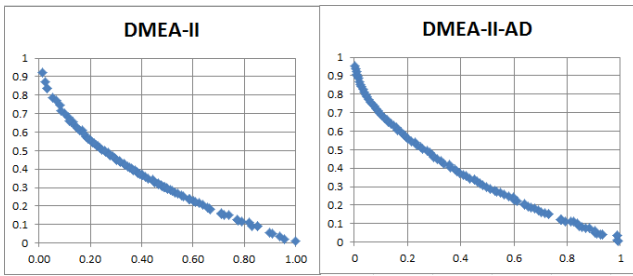


Fig. 5: UF2 results in DMEA-II and DMEA-II-AD in the objective space.

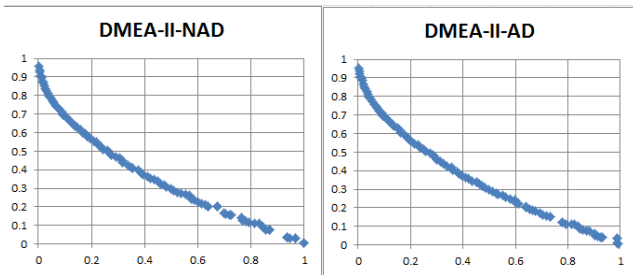


Fig. 6: UF2 results in DMEA-II-NAD and DMEA-II-AD in the objective space.

With regards to DMEA-II, DMEA-II-NAD, the experimental results were categorized as follows:

- GD: On Table IV, we see that DMEA-II-AD obtained quite comparable GD values. It was the best than others on ZDT4, ZDT6, DTLZ2, DTLZ7, UF2, UF7 and UF10. DMEA-II-NAD was the best on UF3, UF4 and UF8, DMEA-II was the best on DTLZ3, UF1, UF5, UF6 and UF9. Three algorithms got the same results on ZDT1, ZDT2.
- IGD: On Table V, DMEA-II demonstrated the better performance. it was the best on test problems: ZDT4, ZDT6, DTLZ2, DTLZ7, UF2, UF7 and UF10. DMEA-II was the best on DTLZ3, UF1, UF5, UF6 and UF9. Three algorithms got the same results on ZDT1, ZDT2, ZDT3.
- HYP: On Table VI we also have the similar results for DMEA-AD. we also have the similar results for DMEA-AD. It got the best HYP values on test problems: ZDT1, ZDT2, ZDT3, DTLZ2, UF2, UF3, UF4, UF5, UF6 and DMEA-II was the best on test problems: ZDT4, ZDT6, DTLZ3, DTLZ7, UF1, UF7, UF8, UF9, UF10.

With regards to all MOEAs in the experiments, the results were categorized as follows:

- GD: On Table IV, the number times that algorithm were the best on others in 18 test problems: (the results are bold): DMEA-II (2 times); DMEA-II-NAD (3 times); DMEA-II-AD (6 times), MOEA/D-DE (4 times); NSGA2-DE (3 times) and SPEA2 (3 times)
- IGD: On Table V, we got: DMEA-II (4 times); DMEA-II-NAD (once); DMEA-II-AD (6 times); MOEA/D-DE (2 times); NSGA2-DE (3 times); SPEA2 (3 times).
- HYP: On Table VI, we got: DMEA-II (4 times); DMEA-II-NAD (none); DMEA-II-AD (6 times); MOEA/D-DE (2 times); NSGA2-DE (once); SPEA2 (3 times).

Overall, the above finding on GD, IGD and HYP

Table IV: The recorded of GD values for each algorithm.

Problems	DMEA-II	DMEA-II-NAD	DMEA-II-AD	MOEA/D-DE	NSGA2-DE	SPEA2
ZDT1	0.0049	0.0049	0.0049	0.0041	0.0052	0.0053
ZDT2	0.0043	0.0043	0.0043	0.0044	0.0041	0.0041
ZDT3	0.0038	0.0038	0.0039	0.0051	0.0062	0.0063
ZDT4	0.0050	0.0041	0.0038	0.0040	0.0775	0.0743
ZDT6	0.0070	0.0035	0.0035	0.0033	0.0033	0.0042
DTLZ2	0.3046	0.2753	0.2533	0.0751	0.0806	0.0823
DTLZ3	0.2670	0.6973	0.5520	0.0760	0.0622	1.3801
DTLZ7	0.0551	0.0463	0.0412	0.1990	0.1438	2.0372
UF1	0.0115	0.0250	0.0417	0.0678	0.0115	0.0076
UF2	0.0084	0.0079	0.0078	0.0347	0.0100	0.0102
UF3	0.1045	0.0859	0.1440	0.0245	0.0670	0.0748
UF4	0.0340	0.0308	0.0315	0.0601	0.0372	0.0375
UF5	0.2674	0.3233	0.3039	0.5649	0.3153	0.1357
UF6	0.2429	0.3979	0.3244	0.3067	0.0522	0.1115
UF7	0.0064	0.0117	0.0059	0.0432	0.0069	0.0064
UF8	1.2094	1.0457	1.0813	0.4294	2.8583	5.6946
UF9	0.4888	0.7070	0.8531	0.3973	1.6661	6.0749
UF10	1.0306	1.0243	1.0135	0.3967	6.4336	3.9500

Table V: The recorded of IGD values for each algorithm.

Problems	DMEA-II	DMEA-II-NAD	DMEA-II-AD	MOEA/D-DE	NSGA2-DE	SPEA2
ZDT1	0.0036	0.0036	0.0036	0.0041	0.0045	0.0038
ZDT2	0.0042	0.0042	0.0042	0.0041	0.0045	0.0038
ZDT3	0.0082	0.0082	0.0082	0.0092	0.0061	0.0046
ZDT4	0.0038	0.0036	0.0033	0.0040	0.0780	0.0449
ZDT6	0.0131	0.0131	0.0130	0.0032	0.0137	0.0134
DTLZ2	0.1933	0.1755	0.1693	0.2874	0.0371	0.0307
DTLZ3	0.3916	0.3900	0.3876	0.5238	0.4438	0.4455
DTLZ7	0.0420	0.0507	0.0534	2.4404	2.3089	1.0575
UF1	0.0127	0.0212	0.0104	0.0649	0.0424	0.0553
UF2	0.0082	0.0094	0.0095	0.0298	0.0153	0.0175
UF3	0.2915	0.3210	0.3245	0.0298	0.2478	0.2453
UF4	0.0344	0.0412	0.0318	0.0585	0.0353	0.0364
UF5	0.0355	0.0454	0.0444	0.1132	0.0349	0.0400
UF6	0.2632	0.2311	0.2213	0.2671	0.1049	0.2991
UF7	0.0082	0.0118	0.0117	0.0411	0.0223	0.1368
UF8	0.7817	0.7554	0.7580	0.1223	1.0365	0.5984
UF9	0.1832	0.2863	0.3114	0.1932	0.0710	0.1906
UF10	0.4616	0.4217	0.4181	0.3670	0.3616	0.4344

Table VI: The recorded of HYP values for each algorithm.

Problems	DMEA-II	DMEA-II-NAD	DMEA-II-AD	MOEA/D-DE	NSGA2-DE	SPEA2
ZDT1	0.9999427	0.9999966	1.0000000	0.9981353	0.9989163	0.9994931
ZDT2	0.9999852	0.9866544	1.0000000	0.9973939	0.9984705	0.9989958
ZDT3	0.9988063	0.9988680	0.9988703	0.9969375	0.9998171	1.0000000
ZDT4	1.0000000	0.9955446	0.9895887	0.9999222	0.9955751	0.9975598
ZDT6	0.8610655	0.8610629	0.8610627	1.0000000	0.8607670	0.8603833
DTLZ2	0.9338714	0.9952644	1.0000000	0.9328805	0.9899145	0.9951133
DTLZ3	0.9999908	0.9999541	0.9999610	0.9999882	0.9999995	1.0000000
DTLZ7	0.9019065	0.8963248	0.8939041	0.8088841	0.9394397	1.0000000
UF1	1.0000000	0.9934165	0.9927002	0.9855799	0.9925886	0.9580743
UF2	0.7910147	0.9995331	1.0000000	0.9882614	0.7875931	0.7793007
UF3	0.5557659	0.9801744	0.9806898	1.0000000	0.5334038	0.4908343
UF4	0.4753205	0.9991141	1.0000000	0.9746712	0.4892211	0.4701164
UF5	0.9871417	0.9843168	0.9888487	0.8754930	1.0000000	0.9349201
UF6	0.5682043	0.9949914	1.0000000	0.9830315	0.6076164	0.5093252
UF7	1.0000000	0.9989987	0.9990176	0.9878160	0.9958622	0.9077507
UF8	0.9974866	0.9830237	0.9845790	0.9012432	0.9994098	1.0000000
UF9	0.9963245	0.9794874	0.9801831	0.8991988	1.0000000	0.9986336
UF10	1.0000000	0.9937050	0.9971402	0.9685680	0.9994719	0.9907375

again shows the strength of our new design on the usage of the balanced directions. In this technique, the perturbation is effected by both directions simultaneously, the new individual will be effected by all directions, it makes the new population to be more balanced between convergence and diversity. The usage of the balanced directions with adaptive method as a guidance for the evolutionary process to make the MOEA balanced between exploitation and exploration automatically.

To get a full understanding of our new design, we also analyze the behaviour of DMEA-II-AD over time. There are several ways for understanding the behaviour of DMEA-II-AD; here we decided to recorded the values of GD and IGD on ZDT1, UF1 over time for the first 500 generations and plotted them on a time dependent graph. On ZDT1, UF2 problem (as examples, they were all visualized in Figures: 7, 8, 9,10 on GD and IGD metrics with DMEA-II, DMEA-II-NAD and DMEA-II-AD), we found that:

- For the results on the GD metric: There is a common pattern that, at the earlier stage (first several generations), DMEA-AD is quite fast in convergence (getting slightly better value of GD than DMEA-II and DMEA-II-NAD did. At the latter stage, DMEA-II-AD was getting better as the search progressed.
- For the results on the IGD metric: At the earlier stage (first several generations), DMEA-II-AD is quite fast in convergence (getting slightly better value of GD than DMEA-II-NAD and DMEA-II-AD did. However, at the latter stage, DMEA-II-AD was getting better as the search progressed.

Through the above comparison using our experimental results, we see that by applying our new technique, DMEA-II's performance is greatly improved, especially the spacing of non-dominated solutions (being shown via IGD values). On 18 well-known test problems, DMEA-II-AD might be a good algorithm on 2 objective problems, but it seems difficult to be solve 3 objective problems (DTLZ3, DTLZ7, UF7). We will investigation this issue in the further research.

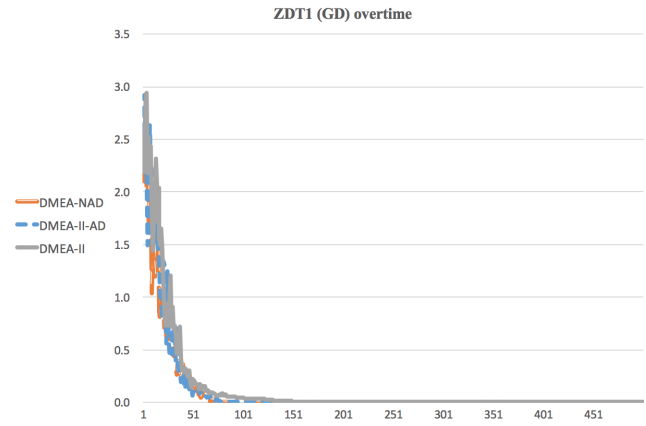


Fig. 7: Visualization of GD over time for ZDT1.

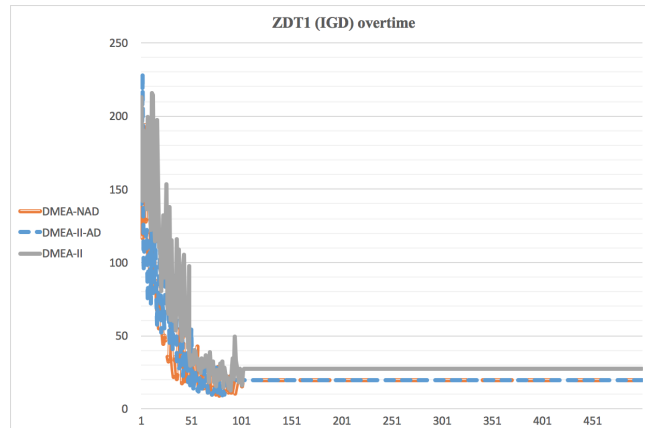


Fig. 8: Visualization of IGD over time for ZDT1.

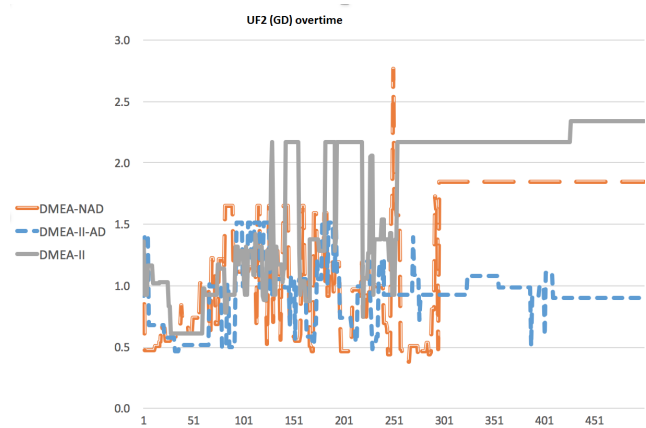


Fig. 9: Visualization of GD over time for UF2.

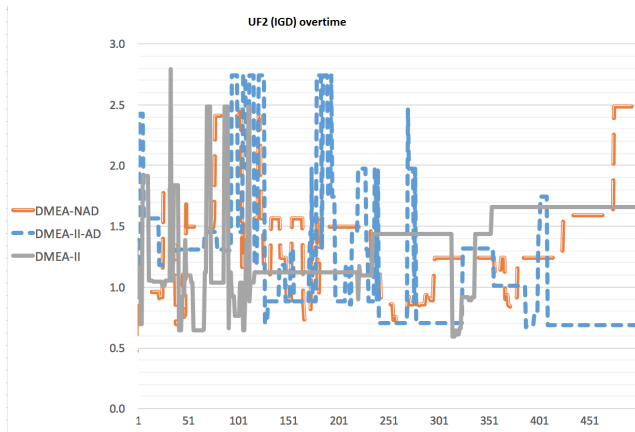


Fig. 10: Visualization of IGD over time for UF2.

V CONCLUSIONS

In this paper, we introduced a new direction of improvement, called balanced directions which is used to improve performance of DMEA-II. The balanced directions are defined from two kinds of improvement direction in DMEA-II: Convergence and Spread direction. By the usage of the balanced directions, the perturbation is effected by both directions simultaneously so the new individuals are effected by all directions, it will make the balanced between convergence and diversity is kept during generations. Experiments on 18 well-known benchmark problems with 5 other MOEAs have been carried out to investigate the performance and behaviour of the new balanced directions to improve DMEA-II. We compared its performance with others on three metrics: GD, IGD and HYP. DMEA-II with new balanced directions showed to be competitive in comparison with all others with respect to both solution convergence and spread. Several analyses on the behaviours of components of the algorithm were thoroughly investigated.

ACKNOWLEDGMENTS

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2015.12.

References

[1] 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.

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

[3] Lam T Bui, Hussein A Abbass, and Daryl Essam. Local models-an approach to distributed multi-objective optimization. *Computational Optimization and Applications*, 42(1):105–139, 2009.

[4] Lam T Bui, Hussein A Abbass, and Daryl Essam. Localization for solving noisy multi-objective optimization problems. *Evolutionary computation*, 17(3):379–409, 2009.

[5] Lam Thu Bui, Kalyanmoy Deb, Hussein A Abbass, and Daryl Essam. Interleaving guidance in evolutionary multi-objective optimization. *Journal of Computer Science and Technology*, 23(1):44–63, 2008.

[6] 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.

[7] 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.

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

[9] K. Deb. *Multiobjective Optimization using Evolutionary Algorithms*. John Wiley and Son Ltd, New York, 2001.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[11] 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.

[12] R. C. Eberhart and Y. Shi. Particle swarm optimization: developments, applications and resources. In *Proceedings of the Congress on Evolutionary Computation*. IEEE Press, 2001.

[13] Antony W Iorio and Xiaodong Li. Incorporating directional information within a differential evolution algorithm for multi-objective optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 691–698. ACM, 2006.

[14] Hui Li and Qingfu Zhang. Comparison between nsga-ii and moea/d on a set of multiobjective optimization problems with complicated pareto sets.

[15] 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.

[16] 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.

[17] Long Nguyen, Lam T Bui, and Hussein A Abbass.

- Dmea-ii: the direction-based multi-objective evolutionary algorithm-ii. *Soft Computing*, 18(11):2119–2134, 2014.
- [18] 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.
- [19] Q. F. Zhang and H. Li. Moea/d: A multi-objective evolutionary algorithm based on decomposition. 2007.
- [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. CMINE, 2001.
- [21] E. Zitzler and L. Thiele. Multi-objective optimization using evolutionary algorithms - a comparative case study. In *Parallel Problem Solving from Nature*, volume 1498 of *Lecture Notes in Computer Science*, pages 292–304. Springer, 1998.
- [22] E. Zitzler, L. Thiele, and K. Deb. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comp.*, 8(1):173–195, 2000.



Bui Thu Lam 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) 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.

AUTHORS BIOGRAPHIES



Nguyen Long has his B.Sc (2000) and M.Tech (2007) degree from Le Quy Don Technical University. He received the Ph.D. degree in Fundamentals of Mathematics for Informatics from the Le Quy Don Technical University, in 2014.

Currently he is a teacher, researcher at the Department of Information Technology, National Defense Academy. His research interests are in the areas of Evolutionary Multi-objective Optimization, Computational Intelligence, Multi-agent systems, cyber warfare, network security and their applications to defense and security domain.



Nguyen Xuan Hung received his B.Sc (1998) and M.Sc (2008) degrees in computer science and technology from Le Quy Don Technical University.

Currently he is a PhD Candidate at the Faculty of Information Technology, Le Quy Don Technical University. His current research interest is Evolutionary Manyobjective Optimization.



Nguyen Thi Hien received her BSc Degree in Computer Science from Hanoi University of Science, MSc Degree in IT from School of Information Technology, Hanoi National University of Vietnam and her PhD in Computer Science from Le Quy Don Technical University in 2015.

Since 2002, she is a lecturer at Faculty of IT, Le Quy Don Technical University. Her field of research interest is in the domain of Genetic Programming and Machine Learning.