An Enhanced Direction based Multi-objective Evolutionary Algorithms using Rank Sum

Dinh Nguyen Duc
Military Information Technology Institute
Academy of Military Science and Technology
Hanoi, Vietnam
nddinh76@gmail.com

Long Nguyen
National Defense Academy
Hanoi, Vietnam
longit76@gmail.com

Hung Nguyen Xuan
Le Quy Don Technical University
Military Science Academy
Hanoi, Vietnam
nguyenxuanhung@outlook.com

ABSTRACT

This paper suggests to use rank sum technique to enhance multi-objective evolutionary algorithms (MOEAs), specifically for MOEAs using directional information. We introduce Rank based Converge Direction (RCD) and Rank based Spread Direction (RSD) for DMEA-II (DMEA-II is a popular directional based MOEA), it called DRS. In which, we use rank sum technique in convergence direction and spread direction in place of dominated-non-dominated direction and non-dominated-non-dominated direction. The performance of the proposal is validated and compared with the original algorithm NSGA-II, DMEA-II on well-known benchmark sets. With respect to result, DRS is compared favorably with its origin and competitive with others.

Keywords

MOPs, MOEAs, DMEA-II, Rank sum, Convergence Direction, Spread Direction.

1. INTRODUCTION

Evolutionary Algorithms (EAs) is used a lot in solving Multi-objective Problems (MOPs), which are known Multi-objective Evolutionary Algorithms (MOEAs). MOEAs can offer a set of trade-offspring solutions. Some related works is presented in Section 2. Of them, The Direction–based Multi–objective Evolutionary Algorithm (DMEA) [10] adopts two types of direction: (1) the convergence direction (CD) between a non-dominated solution and a dominated solution from the current population; and, (2) the spread direction (SD) between two non-dominated solutions in the archive. At each generation, these directions are used to perturb the current parental population from which offspring are produced. Instead of using fixed ratio 1:1 between CD and SD, DMEA-II [11] use an adaptive ratio basing on scale number of non-dominated and number of individual in population each generation.

According to work in [11], the paper showed that the better performance of DMEA–II in comparison with the most popular MOEAs. However, DMEA–II is nearly the same as other MOEAs using Pareto-dominance, there is a small number of non-dominated solution in the early stages, while the vast majority of the population is non-dominated in the later stages. With DMEA-II, in the early stages, in the step of producing offspring using CD direction, many directions are approximately the same; moreover, there are few SD directions. Consequently, it leads to a result may fall into a local optimum area. With the desire to make sure not to fall into traps in local during perturb parents to produce offspring, we focused on increasing the real number of CD and SD directions. We use of rank sum [12] mechanism hoping to overcome these weakness.

The remainder of this paper is organized as follows. Section 2 describes Related Works. Methodology proposed in Section 3. Experiments is presented in section 4 to examine the effective of proposed technique. Conclusion and future works are given in section 5.

2. RELATED WORKS

In this section, we first give common concepts of Evolutionary Multi-objective Optimization. Then we review some related works about MOEAs recently.

2.1 Common Concepts

In many areas of social life, optimization problem has gathered significant research interests. There often exists more than one objective which conflicts to each other. This class of problem is defined as MOPs. The problem is defined as follows [1]:

$$\text{minimize } \{ f_1(x), f_2(x), \dots, f_k(x) \} \quad (1)$$
$$\text{subject to } x \in S$$

in which k (≥2) objective functions $f_i: R^n \rightarrow R$. Vector $f(x) = (f_1(x), f_2(x), \dots, f_k(x))^T$ is denoted as objective functions. The decision (variable) vectors $x = (x_1, x_2, \dots, x_n)^T$ belong to the (nonempty) feasible region (set) S, which is a subset of the decision variable space R^n.

The word *minimize* means that minimization is taken all the objective functions simultaneously. If there is no conflict between the objective functions, then a solution can be found where every objective function attains its optimum. In this case, no special methods are needed. To avoid such trivial cases, we assume that there does not exist a single solution that is optimal with respect to every objective function. This means that the objective functions are at least partly conflicting. They may also be incommensurable.

In evolutionary computation, x represents an individual in the population to be evolved. The value $f_i(x)$ describes the performance of individual x as evaluated against the i^{th} objective in the MOP. For each of k objectives, there exists one different optimal solution. An objective vector constructed with these individual optimal objective values constitutes the ideal objective vector. Unlike the ideal objective vector which represents the lower bound of each objective in the entire feasible search space,

978-1-7281-5009-3/19/$31.00 ©2019 IEEE

the nadir objective vector, represents the upper bound of each objective in the entire Pareto-optimal set [2].

An individual x_1 is said to dominate individual x_2 if x_1 is not worse than x_2 on all k objectives and is better than x_2 on at least one objective. If x_1 does not dominate x_2; and x_2 also does not dominate x_1, then x_1 and x_2 are said non-dominated with respect to each other.

A solution is considered Pareto-optimal if we cannot find any feasible solution which would decrease some criterion without causing a simultaneous increase in at least one criterion. The set of solutions that satisfies the Pareto–optimality definition is called Pareto optimal set (POS).

There have been many methods to solve these problems, among which Multi-objective Evolutionary Algorithms (MOEAs) have been widely and effectively used. The MOEAs target two goals: (1) obtaining the solutions as close to the Pareto optimal solution as possible and (2) retrieving the solutions as diverse as possible along Pareto optimal front (POF).

2.2 Related Works

Issues in EMO to balance between diversity and convergence. While diversity– to maintain a diverse Pareto set approximation, convergence–to guide the population towards the Pareto set.

In the past decades, many MOEAs have proposed to deal with MOPs, there existed many ways to categorize them. One of those can be categorized into two classes. They are Pareto-based and nonPareto-based. NonPareto-based class includes: aggregation-based, indicator-based, and preference-based. Some MOEAs will be listed below.

In Pareto-based approach, each solution is compared with the other one using Pareto-dominance. The authors suggested a non-dominated sorting-based multi-objective EA. Non-dominated Sorting Genetic Algorithm (NSGA) [3], NSGA–II [4], the improved Strength Pareto Evolutionary Algorithm (SPEA2) [5] are typical examples of this approach.

For the aggregation-based approach, the key point is the setting of weighting vectors. Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [6] uses decomposition method to decompose a multi-objective optimization problem into a number of scalar optimization sub-problems and optimize them simultaneously.

Indicator-based approaches, which use an indicator as the fitness function in MOEAs, represent a different type of non-Pareto-based approaches. Indicator-Based Evolutionary Algorithm (IBEA) [7] is the first indicator-based MOEA, it can be adapted to the preferences of the user and moreover does not require any additional diversity preservation mechanism such as fitness sharing to be used.

Improved Two-Archive Algorithm (Two-Arch2) [8] used indicator-based for convergence archive and Pareto-based for diversity archive, these two archives are maintained during search process.

Reference approach is another one, NSGA-III [9], an improved NSGA-II. In order to improve the disadvantages of NSGA-II from the Pareto dominance, a set of reference points is used in the diversity maintenance of NSGA-III. NSGA-III adopts minimal perpendicular distances to the lines (between ideal point and these reference points) as a measure in selecting individuals. In other words, NSGA-III assigns a set of uniformly distributed reference points in advance, which act as a reference distribution for the final output.

The two versions of DMEA [10], [11], which are Direction based MOEAs, adopt two types of direction: (1) the convergence direction (CD) between a non-dominated solution and a dominated solution from the current population; and, (2) the spread direction (SD) between two non-dominated solutions in the archive. At each generation, these directions are used to perturb the current parental population from which offspring are produced. Instead of using fixed ratio 1:1 between CD and SD, DMEA-II use an adaptive ratio basing on scale number of non-dominated and number of individual in population each generation. This paper concentrates on this direction-based.

3. METHODOLOGY

3.1 Rank-sum method

As mentioned before, DMEAs use convergence and spread directions (CD & SD directions) for guiding the search process. Instead of using a fixed ratio between CD and SD (namely 1:1) in DMEA, DMEA-II maintains an adaptive ratio between them. In which, CD is defined as the direction from a solution to a better one, CD in MOPs is a vector that direct from dominated solution to non-dominated solution. An SD is defined as the direction between two equivalent solutions, SD in MOPs is a vector that points from one non-dominated solution to other non-dominated one.

Two algorithms employ Pareto-dominance to maintain elitists. At each generation, elitists are non-dominated solutions that are maintained. In order to generate new population, the algorithms use archive set (elitists solutions) and main population. During the optimization process the archive can be combined with the current population to form next generation. However, with two algorithms, only two layers of solution, that are non-dominated solutions and dominated ones. As a result, there is few real directions between in CD directions.

Table 1: An example of 12 points in 2-objective space

Point	Objective 1	Objective 2
A	1.26	1.25
B	1.66	4.34
C	2.86	1.64
D	1.54	0.73
E	2.52	3.13
F	1.61	2.96
G	1.28	4.03
H	3.85	3.46
I	1.89	1.64
J	3.17	2.08
K	2.90	4.93
L	0.93	3.05

We consider the following example, assuming that, there are 12 points in two-objective space as in Table 1. It is expressed in space as in Figure 1. It shows that there exist 3 points that are non-dominated: A, D, and L, while 9 other points are dominated. As a result, there may be 9 multiple 3 convergence directions. If there do not exist point D and L, there are only directions from other points to unique point A. This may lead to results can be trapped in local optima, so it takes long time to overcome, even it cannot escape from those areas. Moreover, as defined, SD direction is between to non-dominated solutions, therefore, there exist only three SD directions. Even there does not exist SD direction in case of existing one non-dominated solution. Instead of using CD and SD direction for producing offspring based on two kinds of solution (non-dominated and dominated), we use rank sum in order to classify solutions into more classes in hoping eliminate or alleviate the weakness of Pareto-dominance in DMEAs.

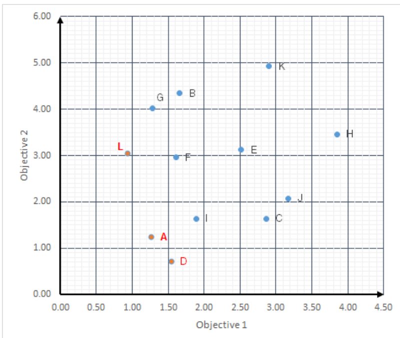

Figure 1: Scatter of 12 points in two-objective space

Building on the idea of rank sum in [12], the objective space is divided into grid, in which, each objective is divided into number of equal parts. Every point in each objective is identified which cell it belongs to and assign corresponding rank to the point for that objective. Rank sum of each solution is calculated by sum up all ranks in each objective.

We consider the data in Table 1. The real values are mapped to rank objective space as in Table 2 and they are illustrated in Figure 2

Table 2: Rank sum of 12 points

Point	rank 1	rank 2	rank sum
A	1	1	2
B	2	8	10
C	6	2	8
D	2	0	2
E	5	5	10
F	2	5	7
G	1	7	8
H	9	6	15
I	3	2	5
J	7	3	10
K	6	9	15
L	0	5	5

After performance of rank sum process, sorting is carried out. The points in same rank sum is considered in the same diagonal line as showed in Figure 3. With rank sum data in Table 2, the points with same rank sum level are drawn in Figure 4

Ranksum sorting is used to produce offspring in CD and SD direction in next section.

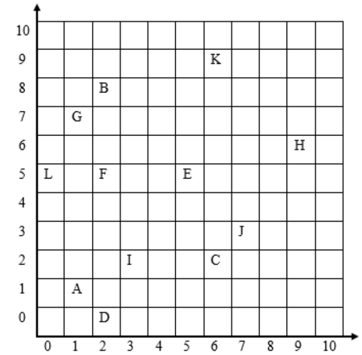

Figure 2: Illustration of 12 points in rank space

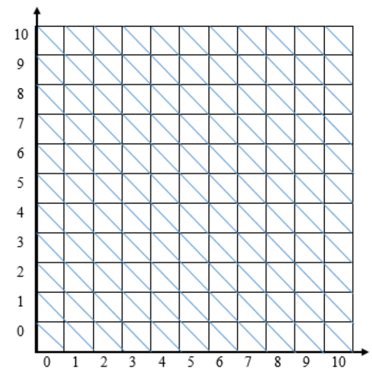

Figure 3: Cells with same ranksum in diagonal lines

3.2 Enhanced Direction-based MOEAs using rank sum

EAs work on population and stochastic mechanism so they can be effectively used to solve difficult problems which have complex optimal set in objective space. Some MOEAs combine them with search mechanism to improve the performance quality of the algorithms. These are MOEAs using directional information. They include gradient based direction, differential directions, and directions of improvement. Gradient based direction method determines search directions based on gradient vector of objective functions a selected point. Since gradient vectors are fast increasing directions of objective at a point, so the idea of using negative derivative directions is the fast decreasing direction of objective functions. Other words, solutions are moved by these directions allow to find an optimal point. Another method uses directional information to guide MOEAs is differential evolution method. In this method, guided directions are determined on the differences of selected individuals in population. In which, the differences of weighted vectors in decision space are used to perturb the population. Besides gradient based and differential evolutions, directions of improvement are used. The directions of improvement as 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 and diversity. A direction based multi-objective evolutionary algorithm (DMEA) [10] is an example of direction of improvement method. In this section, we focus on this DMEA.

DMEAs, which adopted an elitist mechanism, addressed interaction between archive and main population; and archive update. 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. DMEAs use 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.

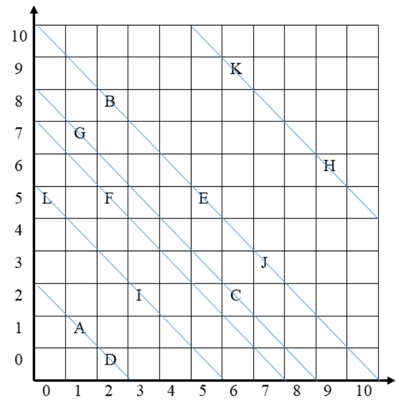

Figure 4: Rank sums of 12 points are classified into 6 layers

As presented, in early stages there may be few non-dominated solution, which leads to few real convergence directions existed. Therefore, it takes long time (number of generations) in evolving to converge, and it may also guide direction to a local optimum area. Then, it may take more time to overcome this local optimum, even it may be stuck here when the number of non-dominated is too small. Furthermore, it causes there are few SD direction (between a non-dominated solution and another one), so it does not ensure diversity in result.

We use rank sum to classify population into many layers in order to alleviate these disadvantages when dividing it by only 2 types of non-dominated and dominated solutions. Solutions with the same ranksum are considered the same layers. The effect is there existed more layers than Pareto-dominance.

With data as in Table 1, which is mapped in rank sum space (in 10 divisions), the result is showed in Figure 4. We have 6 layers totally while Pareto-dominance has 2 layers.

Rank of Converge Direction (RCD) is direction from one solution at one layer towards another solution at better layer. Here, "better" means in the layer has smaller ranksum value.

Rank of Spread Direction (RSD) is direction from one solution to another at the same layer (they have same ranksum value–in the same diagonal line in the ranksum space).

In DRS we use RCD and RSD stand for CD and SD in DMEA, DMEA-II in step 4.

The following is structure of algorithm using RCD and RSD

***Algorithm:* Calculate rank sum**

Input

Div: number of divisions in each objective

// calculate the rank sum of each individual

Step 1. Select one objective which has not ranked

Step 2. Find max, min values of individual in the objective chosen in **Step 1**

Step 3. Calculate lowerbound, upperbound, dimension of cell

lowerbound =min - (max - min) / (2 * div);

upperbound= max - (max - min) / (2 * div);

d= (upperbound- lowerbound)/div;

Step 4. Calculate rank of each individual's objective

individual_rank= (individual_objective –lowerbound)/d

Step 5. If all objectives are judged go to Step 6 else go **Step 1**

Step 6. Calculate ranksum of each individual by summing up of individual's objective rank.

***Algorithm:* Generate Rank of Convergence Direction**

Step 1 if random() >0.5 go to Step 3

Step 2. Select randomly front1, front2 different

Go to Step 4

Step 3. Select randomly front1, front2 adjacent (rank front1 < rank front2)

Step 4. Select the first individual in front1 randomly

Select the second individual in front2 randomly

Step 5. Calculate RCD from the first and the second individual (found in **Step 4**)

***Algorithm:* Generate Rank of Spread Direction**

Step 1. Select randomly a front which has more than one individual

Step 2. Select randomly two individuals in front (found in Step 1)

Step 3. Calculate RSD from the two individuals (in Step 2)

4. EXPERIMENTS

4.1 Testing problem

In order to test the performance of the modified algorithm, we choose a set of 12 continuous benchmark problems from 3 well-known benchmark sets, namely UF[13], ZDT [14], DTLZ [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/nonconvexity, uniformity/non-uniformity, single-modality/multimodality, linearity/non-linearity, interdependency, and continuity/discontinuity. They are detailed in Table 3.

Table 3. List of problem tests

Problems	Number of variables	Number of objectives	Decision Space
ZDT1	30	2	$[0, 1]^{30}$
ZDT2	30	2	$[0, 1]^{30}$
ZDT3	30	2	$[0, 1]^{30}$
ZDT4	10	2	$[0, 1] \times [-5, 5]^9$
ZDT6	10	2	$[0, 1]^{10}$
DTLZ2	12	3	$[0, 1]^{12}$
DTLZ3	12	3	$[0, 1]^{12}$
DTLZ7	22	3	$[0, 1]^{22}$
UF1	10	2	[-1, 1]
UF2	10	2	[-1, 1]
UF3	10	2	$[0, 1]^{10}$
UF4	10	2	$[-2, 2]^{10}$
UF5	10	2	$[-1, 1]^{10}$
UF6	10	2	$[-1, 1]^{10}$
UF7	10	2	$[-1, 1]^{10}$
UF8	10	3	$[-2, 2]^{10}$
UF9	10	3	$[-2, 2]^{10}$
UF10	10	3	$[-2, 2]^{10}$

4.2 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. 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). The first-norm equation is defined as

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

where di 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

$$IGD = \frac{\sum_{i=1}^{\overline{N}} \overline{d_i}}{\overline{N}} \quad (3)$$

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 third measure is Hypervolume indicator (HYP), which is also named as S Metric. Being different from GD and IGD, HYP is a unary measure. Both GD and IGD use the POS 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.

$$HYP = Volume(\bigcup_{i=1}^{N} v_i) \quad (4)$$

Where v_i is volume which is determined by solution i and the ideal point.

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 are the anti-optimal point or worst-possible point in objective space. In our experiments with 8 MOEAs with 12 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.

4.3 Parameter Settings

The experiments for DRS and other popular algorithms including DMEA-II, MOEA/D-DE, NSGA-II-DE, and SPEA2 were carried out on all problems in List of problem tests Table 3. For these algorithms, we used the most stable settings that we found for algorithms as follows: The crossover rate, mutation rate, the number of generations to 0.4, 0.01, 1000; the population size was set to 100 for two objective problems, 300 for three objective ones, respectively.

4.4 Results

Though the results which was reported in Tables 4, 5, 6 we can indicate some new findings:

With the proposed technique, on the GD metric, the metric measures the convergence of obtained solutions after number of runs, the modified algorithm is the best on ZDT1, UF7 and is the second rank on UF1, UF2, UF3, UF5, UF8, UF9 and the algorithm is not good with ranks of 5 in ZDT3. The overall rank is the best in the comparison on GD metric with value is 2.78. On this metric, DMEA-II is difficult to solve ZT6, DTLZ2, UF3.

On IGD metric, the metric measures both convergence and diversity of the obtained solutions we can find that: the algorithm well performance on DTLZ2, DTLZ3, DTLZ7, UF1, UF2, UF6, UF7 an UF9 with ranks 1 or 2. Anyways, the algorithm is not good to solve problems ZDT3 with ranks 5. The overall rank is second with the value 3.00.

On HYP metrics, the another metric for convergence, the algorithm have good results on ZDT1. DTLZ3, UF1, UF2, UF3, UF5, UF6, UF7 and UF9. It also difficult to solve problems: ZDT3, ZDT6. The overall rank is the second.

Through the comparison, based on the features of the test problems, the algorithm can be good convergence on convex, problems, and it is difficult to solve disconnected, non-convex, concave, scalable and multi-modal problems. The new proposed technique helps the built-up algorithm to keep the balance of convergence and spread of the obtained solutions, make it can be competitive with technique of rank sum, make DMEA-II more flexible with difference technique to classify population during the search.

Table 4. The recorded of GD values for each algorithm

Problems	DMEA-II	MOEA/D-DE	NSGA-II-DE	SPEA2	DRS
ZDT1	0.0049 (3)	***0.0041 (2)***	0.0052 (4)	0.0053 (5)	**0.0034 (1)**
ZDT2	0.0043 (4)	0.0044 (5)	***0.0041 (2)***	***0.0041 (2)***	0.0042 (3)
ZDT3	**0.0038 (1)**	***0.0051 (2)***	0.0062 (3)	0.0063 (4)	0.0074 (5)
ZDT4	***0.0050 (2)***	**0.0040 (1)**	0.0775 (5)	0.0743 (4)	0.0053 (3)
ZDT6	0.0070 (5)	***0.0033 (2)***	***0.0033 (2)***	0.0042 (3)	0.0046 (4)
DTLZ2	0.3046 (5)	**0.0751 (1)**	***0.0806 (2)***	0.0823 (3)	0.1602 (4)
DTLZ3	0.2670 (4)	***0.0760 (2)***	**0.0622 (1)**	1.3801 (5)	0.1720 (3)
DTLZ7	**0.0551 (1)**	0.1990 (3)	***0.1438 (2)***	2.0372 (5)	0.2955 (4)
UF1	0.0115 (4)	0.0678 (5)	0.0115 (4)	**0.0076 (1)**	***0.0097 (2)***
UF2	**0.0084 (1)**	0.0347 (5)	0.0100 (3)	0.0102 (4)	***0.0087 (2)***
UF3	0.1045 (5)	**0.0245 (1)**	0.0670 (3)	0.0748 (4)	***0.0441 (2)***
UF4	**0.0340 (1)**	0.0601 (5)	***0.0372 (2)***	0.0375 (3)	0.0426 (4)
UF5	0.2674 (3)	0.5649 (5)	0.3153 (4)	**0.1357 (1)**	***0.1441 (2)***
UF6	0.2429 (4)	0.3067 (5)	**0.0522 (1)**	***0.1115 (2)***	0.1864 (3)
UF7	***0.0064 (3)***	0.0432 (5)	0.0069 (4)	***0.0064 (3)***	**0.0048 (1)**
UF8	1.2094 (3)	**0.4294 (1)**	2.8583 (4)	5.6946 (5)	***0.7636 (2)***
UF9	0.4888 (3)	**0.3973 (1)**	1.6661 (4)	6.0749 (5)	***0.4194 (2)***
UF10	***1.0306 (2)***	**0.3967 (1)**	6.4336 (5)	3.9500 (4)	1.1242 (3)
Eve.rank	2.89	2.83	2.89	3.39	2.78

Table 5. The recorded of IGD values for each algorithm

Problems	DMEA-II	MOEA/D-DE	NSGA-II-DE	SPEA2	DRS
ZDT1	**0.0036 (1)**	*0.0041 (4)*	0.0045 (5)	***0.0038 (2)***	0.0039 (3)
ZDT2	0.0042 (3)	***0.0041 (2)***	*0.0045 (4)*	**0.0038 (1)**	0.0122 (5)
ZDT3	0.0082 (3)	*0.0092 (4)*	***0.0061 (2)***	**0.0046 (1)**	0.0439 (5)
ZDT4	**0.0038 (1)**	***0.0040 (2)***	0.0780 (5)	*0.0449 (4)*	0.0056 (3)
ZDT6	***0.0131 (2)***	**0.0032 (1)**	*0.0137 (4)*	0.0134 (3)	0.0221 (5)
DTLZ2	*0.1933 (4)*	0.2874 (5)	0.0371 (3)	**0.0307 (1)**	***0.0320 (2)***
DTLZ3	**0.3916 (1)**	0.5238 (5)	0.4438 (3)	*0.4455 (4)*	***0.3940 (2)***
DTLZ7	**0.0420 (1)**	2.4404 (5)	*2.3089 (4)*	1.0575 (3)	***0.2714 (2)***
UF1	**0.0127 (1)**	0.0649 (5)	0.0424 (3)	*0.0553 (4)*	***0.0202 (2)***
UF2	**0.0082 (1)**	0.0298 (5)	0.0153 (3)	*0.0175 (4)*	***0.0143 (2)***
UF3	0.2915 (5)	**0.0298 (1)**	*0.2478 (4)*	0.2453 (3)	***0.2356 (2)***
UF4	**0.0344 (1)**	0.0585 (5)	***0.0353 (2)***	0.0364 (3)	*0.0535 (4)*
UF5	***0.0355 (2)***	*0.1132 (4)*	**0.0349 (1)**	0.0400 (3)	0.2338 (5)
UF6	0.2632 (3)	*0.2671 (4)*	**0.1049 (1)**	0.2991 (5)	***0.2118 (2)***
UF7	**0.0082 (1)**	*0.0411 (4)*	0.0223 (3)	0.1368 (5)	***0.0150 (2)***
UF8	*0.7817 (4)*	**0.1223 (1)**	1.0365 (5)	***0.5984 (2)***	0.7637 (3)
UF9	0.1832 (3)	0.1932 (5)	***0.0710 (2)***	*0.1906 (4)*	**0.0416 (1)**
UF10	0.4616 (5)	***0.3670 (2)***	**0.3616 (1)**	0.4344 (3)	*0.4460 (4)*
Eve.rank	2.33	3.56	3.06	3.06	3.00

Table 6. The recorded of HYP values for each algorithm

Problems	DMEA-II	MOEA/D-DE	NSGA-II-DE	SPEA2	DRS
ZDT1	**1.000000 (1)**	0.998135 (5)	*0.998916 (4)*	0.999493 (3)	***0.999930 (2)***
ZDT2	**1.000000 (1)**	0.997409 (5)	0.998485 (3)	***0.999011 (2)***	*0.997515 (4)*
ZDT3	0.998806 (3)	*0.996938 (4)*	***0.999817 (2)***	**1.000000 (1)**	0.822000 (5)
ZDT4	**1.000000 (1)**	***0.999922 (2)***	0.995575 (5)	*0.997560 (4)*	0.999200 (3)
ZDT6	***0.861066 (2)***	**1.000000 (1)**	0.860767 (3)	*0.860383 (4)*	0.836000 (5)
DTLZ2	*0.938457 (4)*	0.937462 (5)	***0.994776 (2)***	**1.000000 (1)**	0.944616 (3)
DTLZ3	*0.999991 (4)*	0.999988 (5)	1.000000 (3)	**1.000000 (1)**	***1.000000 (2)***
DTLZ7	*0.901907 (4)*	0.808884 (5)	***0.939440 (2)***	**1.000000 (1)**	0.920000 (3)
UF1	**1.000000 (1)**	*0.985580 (4)*	0.992589 (3)	0.958074 (5)	***0.998000 (2)***
UF2	0.800410 (3)	**1.000000 (1)**	*0.796948 (4)*	0.788557 (5)	***0.809502 (2)***
UF3	0.555766 (3)	**1.000000 (1)**	*0.533404 (4)*	0.490834 (5)	***0.828000 (2)***
UF4	0.487673 (3)	**1.000000 (1)**	***0.501934 (2)***	0.482333 (5)	*0.486318 (4)*
UF5	0.987142 (3)	0.875493 (5)	**1.000000 (1)**	*0.934920 (4)*	***0.995054 (2)***
UF6	*0.578012 (4)*	**1.000000 (1)**	0.618105 (3)	0.518117 (5)	***0.935880 (2)***
UF7	***0.909091 (2)***	*0.898015 (4)*	0.905329 (3)	0.825228 (5)	**1.000000 (1)**
UF8	*0.997487 (4)*	0.901243 (5)	***0.999410 (2)***	**1.000000 (1)**	0.998111 (3)
UF9	*0.866369 (4)*	0.781912 (5)	***0.869565 (2)***	0.868377 (3)	**1.000000 (1)**
UF10	**1.000000 (1)**	0.968568 (5)	***0.999472 (2)***	*0.990738 (4)*	0.998653 (3)
Eve.rank	2.67	3.56	2.78	3.28	2.72

5. CONCLUSIONS AND FUTURE WORKS

In this paper, we introduced a new technique to enhance directional information MOEAs, namely RCD rank and RSD rank. The ranks are used in direction-based EA, which is called DRS. The technique helps balance between convergence and spread. Experiments on 12 well-known benchmark problems were carried out to investigate the performance and behavior of DRS. The results showed a better performance of DRS over the original DMEA-II and competitive with other MOEAs. We compared its performance on three metrics: GD, IGD, and HYP. DRS with our proposed techniques was competitive in comparison with its predecessor with respect to both solution convergence and spread. Several analyses on the behaviors of the algorithm were thoroughly investigated.

The rank sum has experimented on problems with two or three objectives. Our next study is employment of rank sum on problems with number of objective greater than three.

REFERENCES

[1]. Kaisa Miettinen, "*Nonlinear Multi-objective Optimization*", Kluwer Academic Publishers, 1999.

[2]. K. Deb, *"Multi-objective Optimization using Evolutionary Algorithms"*, John Wiley & Sons, Ltd, 2001.

[3]. N. Srinivas and K. Deb, "Multi-objective function optimization using non-dominated sorting genetic algorithms," Evol. Comput., vol. 2, no. 3, pp. 221–248, Fall 1995

[4]. K. Deb et al, *"A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II"*, IEEE Vol. 6, No. 2, April 2002.

[5]. E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm for multi-objective optimization," in Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, 2001, pp. 95–100

[6]. Q. Zhang and H. Li, "*MOEA/D: a multi-objective evolutionary algorithm based on decomposition*," IEEE Transactions on Evolutionary Computation, vol. 11, no. 6, pp. 712–731, 2007.

[7]. E. Zitzler and S. Künzli, "Indicator-based selection in multi-objective search," in Parallel Problem Solving from Nature-PPSN VIII. Springer, 2004, pp. 832–842.

[8]. H. Wang, L. Jiao, and X. Yao, "*An Improved Two-Archive Algorithm for Many Objective Optimization. Evolutionary Computation*," IEEE Transactions on (2014). DOI: http://dx.doi.org/10.1109/TEVC.2014.2350987, 2014.

[9]. K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach", IEEE Transactions on, vol. 18, no. 4, pp. 577–601, 2014

[10]. Lam Thu Bui, Jing Liu, Axel Bender, Michael Barlow, Slawomir Wesolkowski, and Hussein A. Abbass. DMEA: a direction-based multi-objective evolutionary algorithm. Memetic Computing, pages 271-285, 2011

[11]. Long Nguyen. et al, "DMEA-II: The Direction-based Multi-objective Evolutionary Algorithm - II", Journal of Soft Computing, Volume 18, Issue 11, pp. 2119-2134, 2014, Springer-Verlag

[12]. B. Y. Qu, P. N. Suganthan, "Multi-objective Evolutionary Programming without Non-domination Sorting is up to Twenty Times Faster", CEC'09, 2009

[13]. Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multi-objective optimization test instances for the CEC 2009 special session and competition, Tech. Rep. CES-487, 2008.

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

[15]. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. "Scalable test problems for evolutionary multi-objective optimization", TIK-Report no. 112. Technical report,