

# A New Dynamically Reference Point Adaptation Mechanism in indicator-based EMOA based on weak convergence detection

Weiduo Liao, Ke Shang and Hisao Ishibuchi<sup>(✉)</sup>

Department of Computer Science and Engineering  
Southern University of Science and Technology, Shenzhen, China  
Email: {11849249@mail., hisao@}sustech.edu.cn

**Abstract**—ABSTRACT HERE

**Keywords**—keyword 1; keyword 2

## I. INTRODUCTION

This demo file is intended to serve as a “starter file” for IEEE conference papers produced under L<sup>A</sup>T<sub>E</sub>X using IEEEtran.cls version 1.8a and later. I wish you the best of success.

mds

September 17, 2014

### A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

## II. REFERENCE POINT ADAPTATION

When hypervolume(HV) is used in indicator-based algorithms, one important thing to be considered is that how to specified the reference point. Before calculating the HV values, reference point needs to be chosen in advance. However, it is not suggested that the reference point is set only once at the beginning. [1] This may cause a very far away reference point from solutions for those problems with a very large feasible space(As shown in Fig. 1), as the solutions set is gradually converging to the pareto front during the iteration of the algorithm process.

There is a big problem when applying this strategy to some problems with specific pareto front shape, for example, the inverted-DTLZ1 problem with a inverted-triangular pareto front in 3 dimensions, that many solutions in the final solutions set will distribute at the boundary of the pareto front (Fig. 2a comparing with Fig. 2b)[2], [3]. Although it has no effect on the distributions of solutions set in problems with triangular pareto front in 3 dimensions (Fig. 2c comparing with Fig. 2d), it is necessary using reference point adaptation during the algorithm progress. And the reason is illustrated detailedly in hisao Ishibuchi et al. [4].

This work was supported by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Peacock Plan (Grant No. KQTD2016112514355531), the Science and Technology Innovation Committee Foundation of Shenzhen (Grant No. ZDSYS201703031748284), the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008), and National Natural Science Foundation of China (Grant No. 61876075).

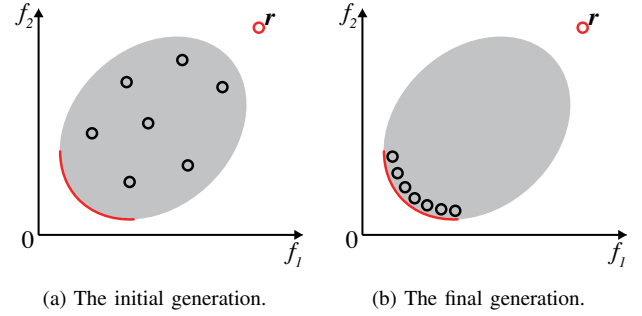


Fig. 1: The reference point is set with a large feasible space. pareto front can be far away from reference point. The gray region shows the feasible region and the red arc is the corresponding pareto front. The red circle  $r$  is the reference point calculated by the initial solutions in (1a) which is randomly generated. After some generations, the current solutions reach to the five black circles in (1b), which is far away from the reference point.

In many algorithms including SMS-EMOA[5], the reference point is adapted based on the following rules:

$$RP = r * ENP, r = 1.1. \quad (1)$$

Note that the estimated nadir point( $ENP$ ) is the nadir point in current population. When the solutions in the current population is obtained, we use hypervolume as indicator to evaluate the performance of the solutions set. Then the reference point used to calculate the hypervolume is calculated by the formula above.

## III. DYNAMIC MECHANISM

Basically the process of Evolutionary Multi-objective Optimization Algorithm can be separated into two stages:

1) *Early Stage*: In this stage, all the solutions are far away from pareto front. The main task is to converge the solutions to pareto front. We also call this stage the convergence stage.

2) *Final Stage*: In this stage, all the solutions are in or near the pareto front. So the main task is to make the distribution of solutions more evenly in the pareto front. We also call this stage the diversity stage.

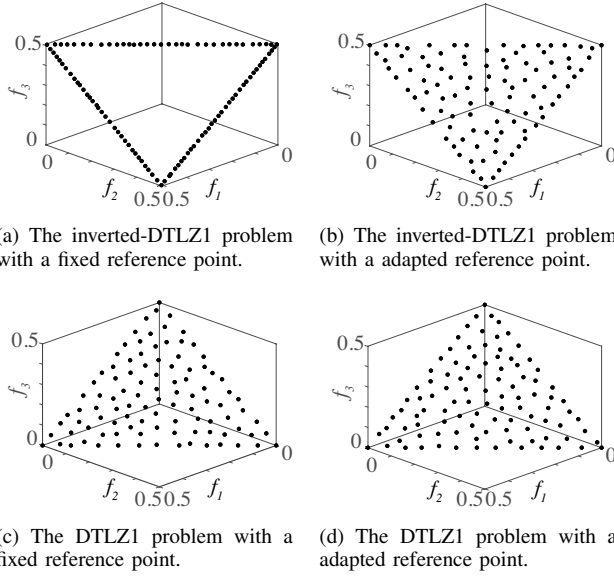


Fig. 2: The final distribution of solutions set in the inverted-DTLZ1(2a and 2b) and the DTLZ1 problem(2c and 2d). The algorithm is FV-EMOA with population size = 100, evaluation number = 20000 and  $r = 1.1$ . (2a and 2c): the reference point is calculated only once at the initial step; (2b and 2d): the reference point is adapted based on the formula (1). All the solutions in the final distribution are at the boundary of the pareto front in (2a), which shows the bad effect of a far away reference point on the final distribution of inverted-triangular problems. This bad effect can not be observed on triangular problems in (2c).

For different purposes in these two stages, the  $r$  should be treated differently[1]. Not only the reference point but also the value of  $r$  needs to be adapted in each iteration of algorithm. This is called dynamically reference point adaptation.

Unfortunately, the research on how to specified  $r$  is limited. Only a few papers[2], [3], [6], [7] did some research on the reference point. The reason is that, the effect of the location of the reference point on the pareto front is not fatal on some benchmark problems, especially triangular pareto front. But in fact, on some specific problems, the distribution of solutions on pareto front strongly depends on the location of the reference point. The sensitivity about value of  $r$  for solutions is also observed on some real world problems, for example, distance minimization problems. This observation Potential shows the usefulness of the dynamically reference point adaptation [8].

In this section, we will specify the suggested  $r$  values seperately in the early and final stages.

#### A. Reference Point Specification for Optimal Distribution

In the final stages, the major purpose is to augment the diversity of solutions set. More specifically, for inverted-triangular problems, the interval between two boundary so-

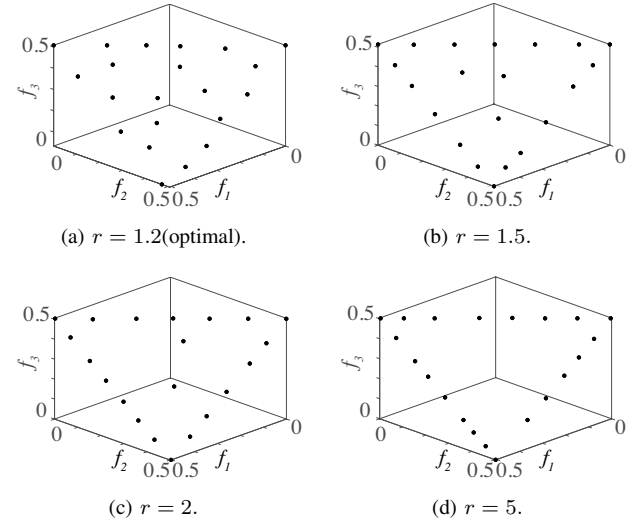


Fig. 3: The final distribution of solutions set in the inverted-DTLZ1 problem. The algorithm is FV-EMOA with population size = 21( $H = 5$ ) and evaluation number = 20000.  $r = 1.2$  is the optimal setting and we observed a evenly distribution in (3a). As the increasing of  $r$ , solutions are more likely to be at the boundary (3b-3d).

lutions should be same as that between two inner solutions. In hisao[8].

$$r = 1 + \frac{1}{H}, \quad (2)$$

is the optimal setting for flat(not concave or convex) pareto front problem (As shown in Fig. 3).  $H$  is the number of solutions intervals in 2-dimension and the number of interval at each boundary of pareto front in many-dimension. In Fig. 3a,  $r = 1.2(H = 5)$  is the optimal setting for a 21 individuals inverted-DTLZ1 problem and a evenly distribution is observed. In Fig. 3b - Fig. 3d, the inner solutions are decreased and move to the boundary of pareto front.

For simply illustration, when algorithm reach to the final stage, all solutions are near the pareto front,  $r$  should be specified as  $1 + 1/H$ , as in equation 2.

#### B. Reference Point Specification for Convergence

However in the early stage, solutions set is not close to the pareto front, which makes the estimated ideal and nadir points far away from true ideal and nadir points, for the reason that they are calculated by current nondominated solutions in each generation. As a result, the search behavior will be poor for the poor normalization[8], [9].

In [8], a larger value of  $r$  than the suggested value  $1 + 1/H$  is used in the early stage.

#### C. Linearly Decrease Mechanism

Based on the theory above,  $r$  is suggested to be specified dynamically at different stages of the algorithm(at early stage, a slightly larger  $r$  is chosen; at final stage,  $r = 1 + 1/H$  is chosen). But unfortunately, there is no best mechanism on how to specify value of  $r$  dynamically outperforms the others in all

problems and all experiments settings. One mechanism may be the best when work on some specific experiments conditions, but may be not good on other conditions.

In [8], a linearly decrease mechanism has been proposed:

$$r(t) = r_{Initial} \frac{(T-t)}{T} + (1 + 1/H) \frac{t}{T}, t = 0, 1, \dots, T, \quad (3)$$

where  $T$  is the total number of generations, and  $r_{Initial}$  is the initial value of  $r$ , which is larger than  $1 + 1/H$ . It is a simple and practical mechanism. In (3), the value of  $r$  starts from  $r_{Initial}$ , then gradually decrease to the suggested value in a linearly decrease process.

In next section, we will propose another good dynamic mechanism based on weak convergence detection criterion outperform simple linearly decrease mechanism on some constraint condition.

#### IV. NEW DYNAMIC MECHANISM

In this section, we will introduce a new mechanism that uses a weak convergence detection criterion to decide whether to change the value of  $r$  from  $r_{Initial}$  to  $1 + 1/H$ .

As we have explained before, a slightly larger  $r$  is suggested at the initial stage of the algorithms. But for well diversity at the final stage, it is needed to set  $r$  to it's optimal value ( $1 + 1/H$ ). For this purpose, we detect whether algorithm has converged or not. If solutions are all close to the pareto front, we change the value of  $r$  to  $r_{Optimal}$ ; otherwise, we set value of  $r$  to  $r_{Initial}$ . The mechanism is also shown below:

$$r(t) = r_{Initial} \mathbb{I}[t < t_{Convergent}] + (1 + 1/H) \mathbb{I}[t \geq t_{Convergent}], t = 0, 1, \dots, T, \quad (4)$$

where  $\mathbb{I}$  is the indicator function returning 1 if argument is true and 0 otherwise.  $r(t)$  equals to  $r_{Initial}$  before reaching to the convergent generation  $t_{Convergent}$ , and changes to  $1 + 1/H$  after  $t_{Convergent}$ . The  $t_{Convergent}$  is determined by a weak convergence detection criterion.

##### A. Weak Convergence Detection

Consider many convergence detection paper [?], various indicators including convergence detection indicators are using to detect the stagnation. They focus on accuracy of convergence, which is not the purpose in our approach for the reason that after algorithm convergent, we still need some generations in order to get evenly distribution of solutions set. We summarize our weak convergence detection criterions as follow:

1) *inaccuracy*: It is no need to have a accurate convergence detection. The convergence can be reported if current solutions are close to the pareto front. In other words, the estimated ideal and nadir points based on the current solutions are close to the true ideal and nadir points.

2) *saving time*: We should not spend too much time in convergence detection for the reason that the state-of-the-art indicator-based algorithms such as SMS-EMOA and HypE, are time consuming when dimension is very high.

We are discussing the effect of reference points in calculating hypervolume. It seems to be a good idea to use progress

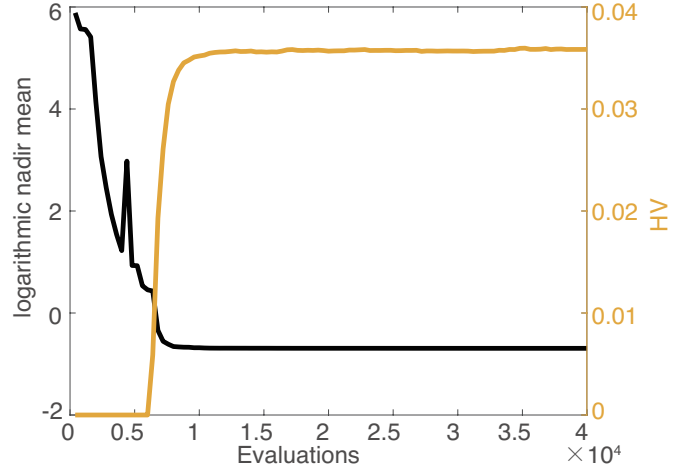


Fig. 4: Example of nadir point and hypervolume(HV) value on inverted-DTLZ1 3-dimension problem with FV-EMOA algorithm. The yellow curve is the change of hypervolume while the black curve is the change of logarithmic nadir point.

indicator hypervolume as our convergence detection indicator, for that we have calculated hypervolume in each generation in hypervolume-based evolutionary multi-objective optimization algorithm. But during the process of algorithm, the reference point is calculated by (1) in each generation. So we can not just simply compare hypervolume calculated in algorithm among different generations.

We are trying on some other good indicators satisfying our convergence detection criterions. Fig. 4 shows the change of hypervolume(HV) and nadir point on the FV-EMOA algorithm with 3-dimensional inverted-DTLZ1 problem. When current solutions are close to the pareto front, the estimated nadir point is close to the true nadir points. This imply that the estimated nadir point can be a good indicator for our purpose. But for some problems with large feasible region, the moving distance of estimated nadir point in the early generations is larger than that near convergence. So we consider the logarithm. And consider the possible bad variance of the indicator, finally we use best logarithmic nadir point so far as our indicator. more specifically, for a minimization problem, we consider the indicator as follows:

$$\begin{aligned} ENP_t &= [f_{t1}, f_{t2}, \dots, f_{tm}]^T \in \mathbb{R}^m, \\ I_0 &= \frac{1}{m} \sum_{i=1}^m \ln f_{0i}, \\ I_t &= \min(I_{t-1}, \frac{1}{m} \sum_{i=1}^m \ln f_{ti}), t = 1, 2, \dots, T, \end{aligned} \quad (5)$$

where  $T$  is the total number of generations,  $ENP_t$  is the estimated nadir point at the  $t$ th generation with  $m$  objectives  $f_{t1}, f_{t2}, \dots, f_{tm}$ .  $I_0$  is the initial indicator calculated by the initial population. And  $I_t$  is the minimum value before the  $t$ th generation (including the  $t$ th generation).

After chosen the indicator, the next step is to detect the

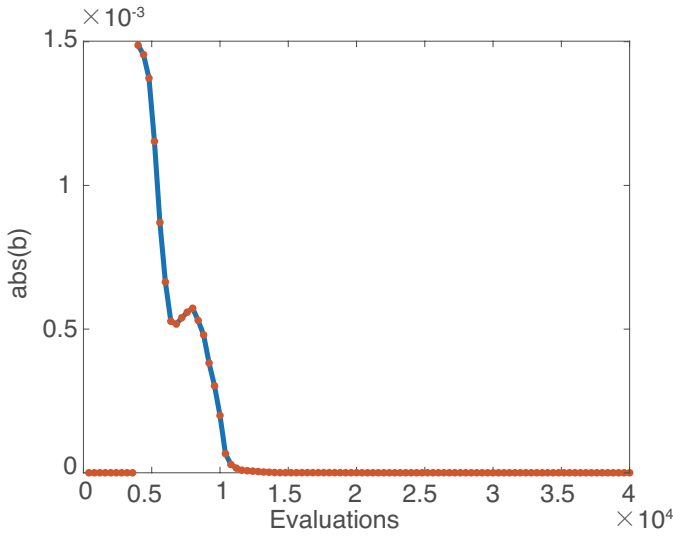


Fig. 5: Example of  $|b|$  on inverted-DTLZ1 3-dimension problem (window size = 4000 evaluations with population size = 100).

stagnation of indicator. We use a basic linear regression method called Simple Least Squares[10] with a simple least squares convergence detection strategy introduced in [11]. If the absolute value of slope of indicator is below a threshold, the convergence is reported. Briefly speaking, for a simple linear regression  $I = a + bt$ , the intercept  $a$  and slope  $b$  of the  $t$ th generation can be calculated with the following matrix-based formula:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum t_i^2 & \sum t_i \\ \sum t_i & w\_l \end{bmatrix}^{-1} * \begin{bmatrix} \sum t_i * I_{t_i} \\ \sum I_{t_i} \end{bmatrix} \quad (6)$$

where  $w\_l$  is the length of the chosen window and  $t_i$  is the evaluated number in the chosen window. The value of slope  $b$  is shown in Fig. 5 (Note that the value in the first  $w\_l$  evaluations is 0, and we should not consider the first  $w\_l$  evaluations).

With the above formula (6), the convergence detection criterion is defined as:

$$convergence = |b| < thres \quad (7)$$

The chosen of the  $thres$  value is not so important as the report ahead or delay is not fatal to the algorithm or to the final solutions set. We choose the  $thres$  value as  $10^{-5}$  after some experimental computation with the window size  $w\_l = 4000$  evaluations. If we do not change the window size, this threshold can be applied to other problems or indicator-based algorithms because of a weak convergence detection purpose.

## V. COMPUTATIONAL EXPERIMENTS

Population size (l): 100, Termination condition: 100,000 solution evaluations, Crossover: SBX (Crossover probability: 1.0, Distribution index: 20), Mutation: PM (Mutation proba-

bility:  $1/(\text{String length})$ , Distribution index: 20), Number of runs: 11 runs.

To clearly represent the superiority of dynamically reference point adaptation and to ease the comparison process of two different dynamic mechanisms (linearly decrease mechanism and weak convergence detection mechanism), the mechanisms is going to be tested with the algorithm FV-EMOA[12]. The problems include: DTLZ series[13], WFG series[?] and their minus versions[?] and two inverted-triangular pareto front problems: Inverted-DTLZ1[14], MaF1[15]. To show the performance both on multi-objective and many-objective, we tested in 3 and 5 dimensions.

### A. Computational Results

Each experiment has been run for twenty times independently. And the population size is 100,  $r_{Initial} = 2$ . The detailed algorithms are as following: FV-EMOA-2(the FV-EMOA[12] algorithm with the reference point adaptation with  $r = 2$ ), FV-EMOA-Opt(the FV-EMOA algorithm with the reference point adaptation with  $r = 1 + 1/H$ ), FV-EMOA-LD(the FV-EMOA algorithm with a linearly decrease mechanism), and FV-EMOA-CD(the FV-EMOA algorithm with a weak convergence detection mechanism, proposed in this paper). We obtain the hypervolume results after 40000 evaluations for each algorithm. The computational results is shown in TABLE I and TABLE II.

In the basic test problems(DTLZ1-4, WFG1, 2, 4 with the triangular pareto front except WFG3), we can not tell the differences among the four algorithms in TABLE I, for that all the four algorithms are more or less the best or the worst and the Wilcoxon rank sum tests show that almost all the results from the other three algorithms are not significantly different from FV-EMOA-CD (except 2 results from FV-EMOA-Opt are worse than results from FV-EMOA-CD).

But in the results of inverted-triangular pareto front problems (MinusDTLZ1-4, MinusWFG1-4, Inverted-DTLZ1, 4, MaF1, 2), FV-EMOA-2 performs almost the worst in all experiments (23 out of 24), and the Wilcoxon rank sum tests show that all the results from FV-EMOA-2 are significantly worse than FV-EMOA-CD. The reason is that the  $r$  is 2 all the process of FV-EMOA-2 algorithm. Comparing with FV-EMOA-CD, the final hypervolume values of FV-EMOA-Opt are not significantly different. And it is difficult to say FV-EMOA-CD is better than FV-EMOA-LD or vice versa, for that near one third of results are better, one third are worse, one third are not significantly different.

### B. The Importance of Using Dynamic Mechanism

We plot the final solutions distributions of some experiments including DTLZ4 and minus-DTLZ4 in 3-dimension (as shown in Fig. 6) and in 5-dimension (as shown in Fig. 7). The solutions distributions of Fig. 6a - 6c hint that the FVEMOA do not have good performance on concave pareto front problems. In Fig. 6d, the solutions are poor distributed comparing with Fig. 6a-6c. This phenomenon is also observed in Fig. 7d. The standard deviations of FV-EMOA-Opt on

TABLE I: HV mean and standard deviation over 20 independent runs for triangular pareto front problems.

Problem	$M$	$D$	FV-EMOA-2	FV-EMOA-Opt	FV-EMOA-LD	FV-EMOA-CD
DTLZ1	3	7	<b>1.4026e-1 (6.25e-5) <math>\approx</math></b>	1.4020e-1 (1.14e-4) $\approx$	1.4021e-1 (1.15e-4) $\approx$	1.4022e-1 (7.38e-5)
DTLZ1	5	9	<b>4.8962e-2 (8.83e-6) <math>\approx</math></b>	4.8955e-2 (1.55e-5) $\approx$	4.8957e-2 (1.35e-5) $\approx$	4.8952e-2 (2.06e-5)
DTLZ2	3	12	7.5684e-1 (1.10e-4) $\approx$	7.5683e-1 (1.48e-4) $\approx$	<b>7.5686e-1 (1.13e-4) <math>\approx</math></b>	7.5681e-1 (1.32e-4)
DTLZ2	5	14	1.2934e+0 (3.12e-4) $\approx$	1.2934e+0 (2.65e-4) $\approx$	<b>1.2935e+0 (3.11e-4) <math>\approx</math></b>	1.2935e+0 (2.62e-4)
DTLZ3	3	12	7.1789e-1 (7.31e-2) $\approx$	7.0047e-1 (1.65e-1) $\approx$	<b>7.4029e-1 (8.35e-3) <math>\approx</math></b>	7.3655e-1 (9.43e-3)
DTLZ3	5	14	1.1264e+0 (3.87e-1) $\approx$	<b>1.1957e+0 (2.82e-1) <math>\approx</math></b>	1.1351e+0 (3.89e-1) $\approx$	1.1931e+0 (2.84e-1)
DTLZ4	3	12	<b>6.9813e-1 (1.21e-1) <math>\approx</math></b>	6.0515e-1 (2.14e-1) $\approx$	6.8340e-1 (1.31e-1) $\approx$	6.6637e-1 (1.76e-1)
DTLZ4	5	14	1.2381e+0 (9.98e-2) $\approx$	1.2308e+0 (1.01e-1) $\approx$	1.2424e+0 (7.21e-2) $\approx$	<b>1.2644e+0 (6.05e-2)</b>
WFG1	3	12	5.9236e+1 (9.45e-1) $\approx$	5.9204e+1 (7.24e-1) $\approx$	<b>5.9303e+1 (4.79e-1) <math>\approx</math></b>	5.9298e+1 (8.50e-1)
WFG1	5	14	5.9946e+3 (3.02e+0) $\approx$	5.9908e+3 (1.09e+1) $\approx$	5.9801e+3 (6.90e+1) $\approx$	<b>5.9954e+3 (1.12e+0)</b>
WFG2	3	12	5.9711e+1 (1.94e-2) $\approx$	<b>5.9720e+1 (2.92e-2) <math>\approx</math></b>	5.9712e+1 (2.54e-2) $\approx$	<b>5.9703e+1 (3.49e-2)</b>
WFG2	5	14	<b>6.0208e+3 (2.44e+0) <math>\approx</math></b>	6.0170e+3 (4.04e+0) $\approx$	6.0186e+3 (3.72e+0) $\approx$	6.0195e+3 (5.24e+0)
WFG3	3	12	<b>6.6296e+0 (1.23e-2) <math>\approx</math></b>	6.6238e+0 (1.16e-2) $\approx$	6.6222e+0 (9.24e-3) $\approx$	6.6229e+0 (1.46e-2)
WFG3	5	14	<b>3.3164e+0 (4.12e-2) <math>\approx</math></b>	3.3137e+0 (3.46e-2) $\approx$	3.2991e+0 (3.42e-2) $\approx$	3.3105e+0 (3.46e-2)
WFG4	3	12	<b>3.6259e+1 (1.29e-2) <math>\approx</math></b>	3.6252e+1 (2.87e-2) $\approx$	3.6258e+1 (1.43e-2) $\approx$	<b>3.6251e+1 (2.37e-2)</b>
WFG4	5	14	<b>4.9536e+3 (3.75e+0) <math>\approx</math></b>	4.9503e+3 (3.64e+0) $\approx$	4.9531e+3 (2.55e+0) $\approx$	4.9523e+3 (2.25e+0)
+/- / $\approx$			0/0/16	0/2/14	0/0/16	

TABLE II: HV mean and standard deviation over 20 independent runs for inverted-triangular pareto front problems.

Problem	$M$	$D$	FV-EMOA-2	FV-EMOA-Opt	FV-EMOA-LD	FV-EMOA-CD
MinusDTLZ1	3	7	4.8083e+7 (1.08e+5) $-$	<b>4.9842e+7 (2.58e+4) <math>\approx</math></b>	4.9832e+7 (2.04e+4) $\approx$	4.9842e+7 (2.51e+4)
MinusDTLZ1	5	9	5.6448e+11 (8.23e+9) $-$	8.5886e+11 (4.19e+9) $\approx$	8.3910e+11 (5.37e+9) $-$	<b>8.5919e+11 (5.29e+9)</b>
MinusDTLZ2	3	12	3.0452e+1 (3.82e-2) $-$	3.0690e+1 (7.55e-2) $\approx$	<b>3.0919e+1 (5.32e-2) <math>+</math></b>	3.0696e+1 (9.18e-2)
MinusDTLZ2	5	14	8.5606e+1 (1.02e+0) $-$	1.0680e+2 (4.56e-1) $\approx$	<b>1.0766e+2 (3.15e-1) <math>+</math></b>	1.0657e+2 (5.69e-1)
MinusDTLZ3	3	12	7.5985e+9 (7.41e+6) $-$	7.6308e+9 (2.89e+7) $\approx$	<b>7.7119e+9 (1.32e+7) <math>+</math></b>	7.6232e+9 (3.43e+7)
MinusDTLZ3	5	14	8.5019e+15 (1.14e+14) $-$	1.0539e+16 (4.69e+13) $\approx$	<b>1.0603e+16 (7.31e+13) <math>+</math></b>	1.0526e+16 (4.87e+13)
MinusDTLZ4	3	12	3.0434e+1 (3.77e-2) $-$	3.1092e+1 (2.34e-2) $\approx$	<b>3.1116e+1 (1.64e-2) <math>+</math></b>	3.1089e+1 (2.57e-2)
MinusDTLZ4	5	14	8.4994e+1 (1.50e+0) $-$	<b>1.0797e+2 (1.95e-1) <math>\approx</math></b>	1.0756e+2 (2.10e-1) $-$	1.0795e+2 (1.89e-1)
MinusWFG1	3	12	6.9856e+0 (2.77e-2) $-$	7.1742e+0 (4.52e-2) $\approx$	<b>7.1857e+0 (2.74e-2) <math>+</math></b>	7.1844e+0 (3.78e-2)
MinusWFG1	5	14	1.4147e+1 (2.48e-1) $-$	<b>1.5635e+1 (2.33e-1) <math>\approx</math></b>	1.5284e+1 (2.01e-1) $-$	1.5539e+1 (2.02e-1)
MinusWFG2	3	12	1.8711e+1 (1.19e-2) $-$	1.8965e+1 (4.48e-3) $\approx$	1.8958e+1 (4.32e-3) $-$	<b>1.8967e+1 (3.80e-3)</b>
MinusWFG2	5	14	4.7715e+1 (1.85e-1) $-$	<b>5.6183e+1 (1.78e-1) <math>\approx</math></b>	5.5939e+1 (1.85e-1) $-$	5.6168e+1 (1.71e-1)
MinusWFG3	3	12	1.3765e+1 (3.31e-2) $-$	1.4269e+1 (9.29e-3) $\approx$	<b>1.4275e+1 (6.86e-3) <math>\approx</math></b>	1.4274e+1 (7.04e-3)
MinusWFG3	5	14	4.2737e+1 (6.41e-1) $-$	6.3482e+1 (7.07e-1) $\approx$	6.2143e+1 (7.62e-1) $-$	<b>6.3513e+1 (6.15e-1)</b>
MinusWFG4	3	12	3.4059e+1 (4.33e-2) $-$	3.4700e+1 (4.83e-2) $\approx$	<b>3.4783e+1 (3.16e-2) <math>+</math></b>	3.4725e+1 (5.17e-2)
MinusWFG4	5	14	6.2416e+2 (1.05e+1) $-$	7.9011e+2 (1.26e+0) $\approx$	7.8781e+2 (2.02e+0) $-$	<b>7.9023e+2 (1.63e+0)</b>
MaF1	3	12	2.8634e-1 (7.00e-4) $-$	<b>2.9746e-1 (1.36e-4) <math>\approx</math></b>	2.9736e-1 (1.36e-4) $\approx$	2.9736e-1 (1.87e-4)
MaF1	5	14	1.1011e-2 (1.83e-4) $-$	1.6331e-2 (1.39e-4) $\approx$	1.6138e-2 (1.73e-4) $-$	<b>1.6331e-2 (9.90e-5)</b>
MaF4	3	12	4.0852e+1 (9.56e+0) $-$	<b>3.9125e+1 (1.44e+1) <math>\approx</math></b>	4.1884e+1 (1.07e+1) $\approx$	<b>4.4345e+1 (2.82e+0)</b>
MaF4	5	14	3.7215e+3 (1.64e+3) $-$	<b>6.2601e+3 (2.81e+2) <math>\approx</math></b>	6.2386e+3 (3.23e+2) $\approx$	6.1540e+3 (4.87e+2)
Inverted-DTLZ1	3	7	3.5744e-2 (1.47e-4) $-$	3.6520e-2 (1.11e-3) $\approx$	<b>3.6961e-2 (2.51e-4) <math>\approx</math></b>	3.6897e-2 (4.23e-4)
Inverted-DTLZ1	5	9	3.1124e-4 (4.25e-5) $-$	<b>4.7494e-4 (4.71e-5) <math>\approx</math></b>	4.5874e-4 (6.97e-5) $\approx$	4.2499e-4 (1.00e-4)
Inverted-DTLZ2	3	12	7.0894e-1 (1.17e-3) $-$	7.1559e-1 (2.72e-3) $\approx$	<b>7.2070e-1 (1.16e-3) <math>+</math></b>	7.1612e-1 (2.45e-3)
Inverted-DTLZ2	5	14	1.6127e-1 (2.18e-3) $-$	2.0287e-1 (8.88e-4) $\approx$	<b>2.0452e-1 (6.98e-4) <math>+</math></b>	2.0326e-1 (5.90e-4)
+/- / $\approx$			0/24/0	0/0/24	9/8/7	

the DTLZ4 problem (shown in the TABLE I) are both higher than other three algorithms in 3- and 5-dimension ( $2.14e-1$  comparing with  $1.21e-1$ ,  $1.31e-1$ ,  $1.76e-1$  in 3-dimension and  $1.01e-1$  comparing with  $9.98e-2$ ,  $7.21e-2$ ,  $6.05e-2$  in 5-dimension). This observation also shows that the solutions distributions of some runs in FV-EMOA-Opt are poor over the total 20 runs.

In FV-EMOA-Opt, the algorithm applies  $r_{Initial} = 1 + 1/H$  mechanism at the early stage of algorithm while other three algorithms apply  $r_{Initial} = 2$ . Even though the  $r_{final}$  of FV-EMOA-LD, FV-EMOA-CD and FV-EMOA-Opt are all equal to  $1 + 1/H$ , FV-EMOA-Opt can not jump out from the local minimum and estimate the pareto front well due to the pool

searching behavior at the early stage.

The above examples clearly show the importance of using dynamic mechanism, that a slightly larger  $r$  than  $1 + 1/H$  can make the algorithm with a better searching behavior in convergence stage where the solutions have not reached to the pareto front especially for the problems which is difficult to find the whole solutions space like the DTLZ4 problem.

### C. Comparison of Two Dynamic Mechanisms

We want to further investigate the differences between two dynamic mechanisms (the linearly decrease mechanism and the weak convergence detection mechanism). Here is the examples of the final distribution on MaF1 problem (as shown in Fig. 8). The total evaluation number is set to 3500



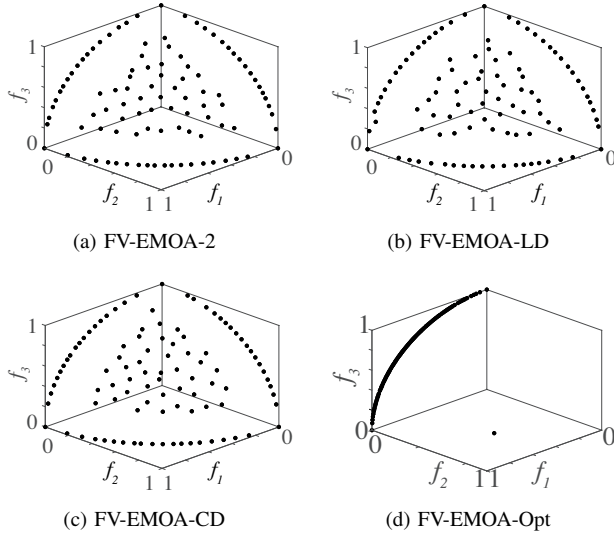


Fig. 6: The final distribution of 4 reference point strategies on 3-dimensional DTLZ4 problems.

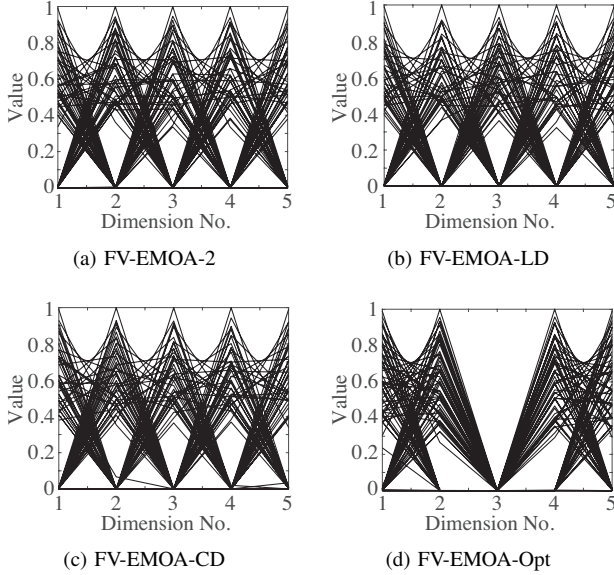


Fig. 7: The final distribution of 4 reference point strategies on 5-dimensional DTLZ4 problems.

and the convergence is detected at 3400 evaluations. The solutions distribution of FV-EMOA-CD(Fig. 8c) is similar to the that of FV-EMOA-Opt(Fig. 8d) while there are still overmany solutions at the boundary of pareto front in FV-EMOA-LD(Fig. 8b).

The reason is quite intuitionistic. In the linearly decrease mechanism, the total evaluation number is so small that it is too late for the solutions to distribute evenly when reaching to the final generation. But in the weak convergence detection mechanism, a convergence is reported after 3400 evaluations and the algorithm still has 100 evaluations to make a well solutions distribution.

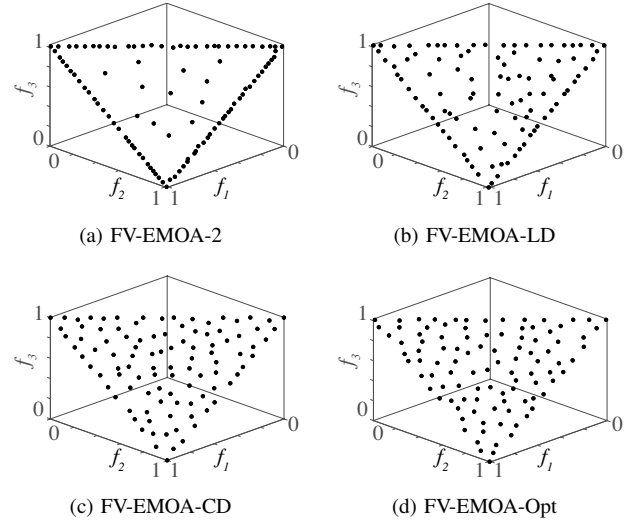


Fig. 8: The final distribution of 4 reference point strategies on MaF1 problem. The results are obtained after 3500 evaluations( $t_{Convergent} = 3400$ ).

## VI. CONCLUSION

In this paper, we emphasize the importance of reference point adaptation in indicator-based EMOA. Without a reference point adaptation mechanism, the diversity of the final solutions will be poor on inverted-shape pareto front problems.

## REFERENCES

- [1] C. Liu, Q. Zhao, B. Yan, S. Elsayed, T. Ray, and R. Sarker, "Adaptive sorting-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2018, [J]TEVC.
- [2] O. P. Jones, J. E. Oakley, and R. C. Purshouse, "Component-level study of a decomposition-based multi-objective optimizer on a limited evaluation budget," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2018, pp. 689–696, [C]GECCO.
- [3] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "How to specify a reference point in hypervolume calculation for fair performance comparison," *Evolutionary computation*, pp. 1–29, 2018, [J]EC.
- [4] —, "Reference point specification in hypervolume calculation for fair comparison and efficient search," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2017, pp. 585–592, [C]GECCO.
- [5] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2017, [J]TEVC.
- [6] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.
- [7] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "Hypervolume subset selection for triangular and inverted triangular pareto fronts of three-objective problems," in *Proceedings of the 14th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*. ACM, 2017, pp. 95–110.
- [8] H. Ishibuchi, K. Doi, and Y. Nojima, "On the effect of normalization in moea/d for multi-objective and many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 4, pp. 279–294, 2017.
- [9] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016, [J]TEVC.

- [10] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach." *IEEE Trans. Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014, [J]TEVC.
- [11] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints." *IEEE Trans. Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014, [J]TEVC.
- [12] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86–95, 2012, [J]TEVC.
- [13] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary computation*, vol. 19, no. 1, pp. 45–76, 2011, [J]EC.
- [14] N. Beume, B. Naujoks, and M. Emmerich, "Sms-emoa: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.
- [15] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006, [J]TEVC.