# A new Ensemble Framework based on MOEA/D

Jiayi Han and Shinya Watanabe[1]

***Abstract***: Multi-objective evolutionary algorithm based on decomposition (MOEA/D) is a powerful algorithm and provides a framework for solving multi-objective optimization problems (MOPs). Differential evolution (DE) algorithm and its variants are often used to generate new solutions in MOEA/D heuristically. However, based on the "No Free Lunch" theory, only a fixed algorithm for generating new solutions in the original MOEA/D cannot efficiently solve all MOPs.

Therefore, in this paper, we propose a new framework based on MOEA/D named MOEA/D-EF (Ensemble Framework), which can contain a variety of new-solutions generating algorithms (candidate algorithms) with different search capabilities to improve the overall universality of the algorithm. In the new approach, the whole iteration is divided into the evaluation generation (EG) and the implementation generation (IG). We provide a fair evaluation environment for each candidate algorithm at the beginning of each generation belonging to the EG and evaluate their performance by using the Hypervolume indicator. The algorithm with the best performance in one EG will be chosen and executed in the following IG.

Also, we believe that some historical information representing evolutionary details can help generate superior new solutions. Thus, in numerical experiments, we take our original DE variant based on the ideal point and historical information as one of the candidate algorithms for generating new solutions. The numerical experiments show that the new framework has broader universality.

***Keywords***: MOEA/D, MOPs, ensemble framework, historical information

## 1. Introduction

As one of the current state-of-the-art multi-objective evolutionary algorithms, Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [1] provides a framework for solving MOPs. In MOEA/D, a set of evenly weight vectors in the objective space is used to decompose one MOP into multiple subproblems. For each of the subproblems, scalarization function (such as Weighted Sum Approach or Tchebycheff Approach) and weight vectors are used to coordinate each objective function's relationship, and each subproblem can be considered as a scalar optimization problem. Since that, it is natural to use genetic evolution algorithms those were originally designed to solve single-objective optimization problem such as the Differential Evolution (DE) [2] algorithm and its variants to generate new solutions.

Due to different parameter selection strategies, the DE algorithm and its variants exhibit different search capabilities, which can be described as exploitation and exploration. Exploration is the process of visiting entirely new regions of a search space, while exploitation is visiting those regions within the neighborhood of previously visited points [3]. However, according to the "No Free Lunch" theory, it is difficult for one new-solutions generating algorithm to exhibit both exploitation and exploration capabilities.

To efficiently solve the optimization problem through evolutionary computation, in this paper, we propose a new Ensemble Framework based on MOEA/D(MOEA/D-EF), which can contain a variety of new-solutions generating algorithms with different search capabilities and has a mechanism to switch new-solutions generating algorithms by the current search situation.

We inherit part of the idea from HMJCDE [4] and MVC [5] framework. However, most similar attempts are designed for single-objective optimization problems. Different from that, one of the critical problems of multi-objective optimization is how to balance the relationship between multiple objective functions. Even though the MOEA/D framework can import modified single-objective optimization algorithms, it is necessary to use a comprehensive indicator for evaluating a multi-objective optimization algorithm. Thus, the Hypervolume indicator, widely used in solving MOPs, is introduced for switching candidate algorithms in our approach.

In addition, we strongly believe that not only the current population, but also the past population should have the information that can contribute to the evolution. In this paper, we also propose a DE variant with historical items. We named the new variant as DE-IDEAL, because the historical item satisfies the geometric relation with respect to the ideal point in MOEA/D.

Since our motivation is to improve the universality of the overall algorithm, we expect candidate algorithms to have different search capabilities. Although, in theory, our new approach can contain any number of different new-solutions generation algorithms. In the current study, we chose the classic DE algorithm, JADE [6] with applicable modification, and DE-IDEAL for numerical experiments.

## 2. DE-IDEAL

A well-known DE variant with the mutation strategy DE/best/1 is defined as: of

$$V_i^G = X_i^G + F \cdot (X_{best}^G - X_{r1}^G) \tag{1}$$

where $X_{best}^G$ is best solution in G generation, and $X_{r1}^G$ is randomly selected from the current population $N$. In MOEA/D framework, $X_i^G$ always has a minimum fitness-value to $\lambda_i$ in neighborhood $B(i)$, and $X_{r1}^G$ should be randomly selected from the neighborhood $B(i)$, the mutation strategy of DE-IDEAL should be modified as:

1 Muroran Institute of Technology, Muroran Hokkaido 050-8585, Japan.

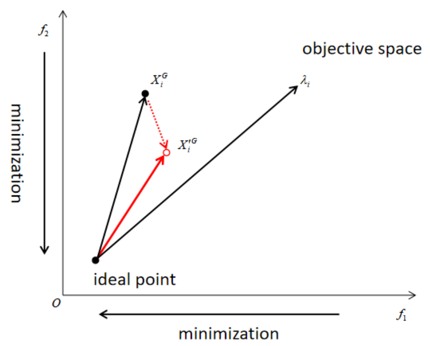
$$V_i^G = X_i^G + F \cdot (X_i^G - X_{r1}^G) + F' \cdot H_i \quad (2)$$

where $H_i$ is a historical vector with the form as same as $X_{best}^G$ or $X_{r1}^G$ in the design space, and randomly selected from set $(\widetilde{H}_i \in HIP)$.

Historical information pool (set $HIP$) is used to store historical details. Specifically, when the parent individual $X_i^G$ is successfully replaced by its child $X_i^{G+1}$, $\tilde{V} = X_i^{G+1} - X_i^G$ will be append to set $HIP$. To be clear, $\tilde{V}$ is a vector with the same form as $X_i^G$ and $X_i^{G+1}$ in the design space, but it also represents the vector $\tilde{V}` = F(X_i^{G+1}) - F(X_i^G)$ in the objective space. $HIP$ can hold as many items as the population size $N$. If the number of items in $HIP$ reaches $N$, the item which was first added to $HIP$ will be removed, and the new $\tilde{V}$ will be added in. $\widetilde{H}_i$ is a subset of $HIP$, and the corresponding $\tilde{V}`$ of $\tilde{V}$ in $\widetilde{H}_i$ should satisfy certain geometric relations in the objective space.

We assume an individual $X_i^G$, and its position in the objective space is shown in Fig. 1. Ideally, it would be alluring for individual $X_i^G$ to approach the ideal point strictly in the opposite direction of the weight vector $\lambda_i$. Still, the reality is that there is usually some distance between individual $X_i^G$ and the weight vector $\lambda_i$. We expect that the individual $X_i^G$ to be in the position of $X`_i^G$, because $X`_i^G$ is closer to the weight vector $\lambda_i$ and closer to the ideal point than $X_i^G$, simultaneously. The key to getting from $X_i^G$ to $X`_i^G$ is to find a vector represents $X`_i^G - X_i^G$. Let $\tilde{V}` = X`_i^G - X_i^G$, $\tilde{V}`$ must satisfy two conditions:

- Based on simple geometry, $\angle VIW$(the angle from $-\tilde{V}`$ to $\lambda_i$) must bigger than $\angle XIW$ (the angle from $X_i^G$ to $\lambda_i$). $X_i^G$ and $X`_i^G$ represent a positional relationship in the




  objective space, and the red dotted arrow represents the vector $v`$. It is important to note that vectors $X_i^G$, $X`_i^G$, and $\lambda_i$ here all start from ideal point.

- $\angle VIX$ (the angle from $-\tilde{V}`$ to $X_i^G$) must be less than a threshold $\theta`$. Because an oversize $\angle VIX$ will still lead $X`_i^G$ closer to the weight vector $\lambda_i$ but it will take $X`_i^G$ further away from the ideal point.

## 3. The Ensemble Framework Based on MOEA/D

Penalty-based boundary intersection (PBI) approach was used as the scalarization function in our current study, which can decompose a MOP into several subproblems (scalar optimization subproblems) as the following consideration:

$$\text{minimize } g^{bip}(x|\lambda, z*) = d_1 + \theta d_2$$
$$\text{subject to } x \in R^m \quad (3)$$

where $$d_1 = \frac{||(F(x)-z*)^T\lambda||}{||\lambda||}$$

$$d_2 = ||F(x) - (z* + d_1\lambda)||$$

$\lambda$ represents even spread weight vectors $\lambda_1, \lambda_2, \ldots \lambda_N$; $z^*$ is the ideal point; $\theta > 0$ is a preset penalty parameter.

In MOEA/D-EF, we divide all generations into two parts, the evaluation generation ($EG$) and the implementation generation ($IG$). $EG$ and $IG$ match each other, and multiple $EG - IG$ pairs will be presented in the whole iteration process. $EG$ aims to evaluate the performance of different new-solutions generating algorithms (candidate algorithm) fairly by providing the same initial environment. However, it leads $EG$ to consume several times the computing resources of its corresponding $IG$. Therefore $EG$ is usually set to have a small capacity. In this study, all $EG$s including the initial $EG_1$ were set to contain 5 generations, and the initial $IG_1$ was set to contain 10 generations. The capacity of the following $IG_k|_{k>1}$ has the capacity as $len(IG_k) = 2len(IG_{k-1}|_{k>1})$ until $len(IG_k) \geq 40$.

In $EG_k$, we first copy the population at generation $G$ into $q$ copies (in this paper $q = 3$) and assign the identical population as an initial population to each candidate algorithm. When all candidate algorithms have completed the calculations for generation $G$, the Hypervolume value $HV_j^G|_{1\leq j\leq q}$ of each algorithm are recorded. The population in the next generation $G+1$ will be generated by the following mechanism:

$$X_i^{G+1} = X_{i,j}^G$$
$$fit_i(X_{i,j}^G) = min\, f\, it_i(X_{i,j}^G)|_{1\leq j\leq q} \quad (4)$$

where $X_{i,j}^G$ represents the $i$th individual generated by the $j$th candidate algorithm. In this paper, $X_{i,1}^G$, $X_{i,2}^G$ and $X_{i,3}^G$ represent the individual generated by JADE, DE-IDEAL, and DE1, respectively. $fit_i(X_{i,j}^G)$represents the fitness-value of $X_{i,j}^G$ to $\lambda_i$, which can be simply calculated by (10). If generation $G+1$ still in$EG_k$, the population in generation $G+1$ will be copy into $q$ copies as same as the description at the beginning of this paragraph, and repeat the process; Otherwise, the regnant candidate algorithm will be determined and be executed in the following $IG_k$ The algorithm works as follows:

**Input:** A MOP; A stopping criterion; $N$: the number of subproblems; A set of even spread weight vectors:$\lambda_1, \lambda_2, \ldots \lambda_N$; $T$: the number of the weight vectors in the neighborhood of each weight vector.

**Output:** EP, which is an external population used to store non-dominated solutions found during the search.

**Step 1) Initialization:**

**Step1.1)** Set $EP = \emptyset$.

**Step1.2)** Compute the Euclidean distances between any two weight vectors and then work out the $T$ closest weight vectors to each weight vector. For each $i = 1, \ldots, N$, set $B(i) = \{i_1, \ldots, i_T\}$, where $\lambda_{i_1}, \ldots, \lambda_{i_T}$ are the $T$ closest weight vectors to $\lambda_i$.

**Step1.3)** Generate an initial population $S: X_1, \ldots X_N$ randomly.

**Step1.4)** Initialize $z = (z_1, \dots, z_{1m})$ where $z = (z_1, \dots, z_{1m})$ $z_i = minf_i(x)|_{1 \le i \le m}$.

**Step1.5)** Initialize the capacity of $EG$ and $IG$ as the method mentioned above.

**Step2) Switch:**

If the current generation is in $EG$: set $S_1 = S_2 = S_3 = S$, and assign $S_1, S_2, S_3$ to JADE, DE-IDEAL, and DE, respectively.

**Step3) Update:**

For $i = 1, \dots, N$:

**Step3.1) Reproduction:**

If the current generation is in $EG$:

generate $X_{i,1}^G$, $X_{i,2}^G$ and $X_{i,3}^G$ by JADE, DE-IDEAL, and DE, respectively.

Else:

generate $X_{i,R}^G$which represents the individual generated by the regnant candidate algorithm determined during last $EG$.

**Step 3.2) Update of** $z$:

For $j = 1, \dots, m$:

If $z_j > min(f_j(X_{i,1}^G), f_j(X_{i,2}^G), f_j(X_{i,3}^G)$ or $f_j(X_{i,R}^G))$:

$z_j = min(f_j(X_{i,1}^G), f_j(X_{i,2}^G), f_j(X_{i,3}^G)$ or $f_j(X_{i,R}^G))$, depends on the current generation $G$ is in $EG$ or not.

**Step 3.3) Update of Neighboring Solutions:**

Mark the basic vector of $X_{i,1}^G$ as $X_{i,1}^{1,G}$, where the individual $X_{i,1}^{1,G} \in S_1$. Let $j = 1, \dots, T$, the neighborhood of the current subproblem can be considered as $B(i) = \{i_j|_{1 \le j \le T}\}$, where $i_j$ is an index represents the $i_j$thsubproblem, and also it can represent a weight vector $\lambda_{i_j}$. The neighboring solutions update strategy can be considered as:

For $j = 1, \dots, T$:

If $fit_{i_j}(X_{i,1}^G) \le fit_{i_j}(X_{i,1}^{j,G})$:

$X_{i,1}^{j,G} = X_{i,1}^G$; break.

$X_{i,2}^G$, $X_{i,3}^G$ and $X_{i,R}^G$ have similar neighboring solutions update strategy with the basic vector $X_{i,1}^{2,G}, X_{i,1}^{3,G}$ and $X_{i,R}^{R,G}$.

**Step 4) *Update of the population[1]:**

At the end of generation G, there are three populations $S_1, S_2, S_3$.

**Step 4.1) Unification:**

For $j = 1, \dots, q$:

Do the mechanism shown as (4), get $S_j^{G+1}$.

**Step 4.2) Calculation of Hypervolume value:**

Calculate $HV_1^G$, $HV_2^G$ and $HV_3^G$ which represent the Hypervolume values of JADE, DE-IDEAL, and DE, respectively.

**Step 5) *End of EG:**

If the current generation is the last generation in $EG$, **a**ssuming that generation $G$ is the beginning of the current EG, calculate $V_1, V_2$ and $V_3$, where $V_j = \sum_G^{G+5} V_j^G |_{1 \le j \le q}$. The cumulative Hypervolume value represents the overall performance of each candidate algorithm in the current EG respectively. The candidate algorithm has the biggest V will be determined as the regnant algorithm and will be executed in the following EG.

**Step 6) Update of EP:**

Remove all the dominated vectors in union $EP \cup S^{G+1}$.

**Step 7) Stopping Criteria:**

If stopping criteria is satisfied, then output EP. Otherwise, go back to **Step 2)**.

## 4. Numerical experiments

Comprehensive experiments are conducted to evaluate our new framework and compare the effectiveness of MOEA/D-EF with the approaches that only use pure DE, JADE and DE-IDEAL. Since Hypervolume values were used as the switching indicator in the iteration of MOEA/D-EF, the IGD value was selected as the evaluation indicator in the comprehensive experiment.

Three-objective WFG[2] series problems with 30, 50 and 100 design variables are taken as the test instances. In addition to the conventional numerical analysis, we discuss the universality of the new algorithm utilizing horizontal comparison.

### 4.1 Parameters setting

For each approach, the subproblem number $N$ is set to 300, and the neighborhood size $T$ is set to 21. The scalarization function PBI is imported to decompose a MOP. The preset penalty parameter $\theta = 5$ is set throughout. For DE and DE-IDEAL, the scaling factor $F$ and the crossover rate $CR$ are set as 0.5 and 0.9, respectively. The unique scaling factor $F`$ of DE-IDEAL is set as 0.1. As the definition given in Section 2, the $\angle VIX$ in DE-IDEAL must be less than a threshold $\theta`$, we subjectively relate $\theta`$ to $\theta$ as: $\theta` = arctan\frac{1}{\theta}$.

### 4.2 The numerical evaluation of IGD

Numerical experiments with 30, 50 and 100 design vectors were repeated 11 times, respectively. The result of IGD values is shown in Table 1 to 3, where MOEA/D-EF is called EF and DE-IDEAL is called IDEAL for short. V30 represents each approach contains 30 design variables e.g. The MV column represents the average IGD value, and the R column represents the ranking of the approach in the current problem. The R column in the last row represents the average score of the ranking. Obviously, the smaller the average score, the better the approach performs.

As shown from Table 1 to 3, DE-IDEAL always takes the lead in WFG 4,7,8. JADE takes the lead in WFG 5,6,9 but always performs worst in other problems. MOEA/D-EF has not took the lead on any problems but never has a worst-case performance.

DE-IDEAL had the best overall performance in ranking score, followed by MOEA/D-EF. With the number of design variables increases to 100, the difference between those two approaches became very limited. To some extent, this result supports our

1 Step 4)* and Step 5)* are only for the situation that the current generation is in EG.

2 The IGD values of all approaches on WFG2 showed drastic fluctuations. We consider that the results of WFG2 are not representative, and the report related to WFG2 will not be listed in current.

TABLE 1. AVERAGE IGD VALUE AND RANKING SCORE FOR PROBLEMS WITH 30 VECTORS.

| V30 | EF | | DE | | IDEAL | | JADE | |
|---|---|---|---|---|---|---|---|---|
| | MV | R | MV | R | MV | R | MV | R |
| WFG1 | 1.4676 | 3 | **1.4607** | **1** | 1.4613 | 2 | 1.4814 | 4 |
| WFG3 | 0.1679 | 2 | **0.1566** | **1** | 0.1723 | 3 | 0.6122 | 4 |
| WFG4 | 0.3100 | 2 | 0.3239 | 3 | **0.2899** | **1** | 0.4765 | 4 |
| WFG5 | 0.1821 | 2 | 0.2253 | 4 | 0.1877 | 3 | **0.1760** | **1** |
| WFG6 | 0.1940 | 2 | 0.2518 | 4 | 0.2118 | 3 | **0.1885** | **1** |
| WFG7 | 0.3562 | 3 | 0.3553 | 2 | **0.3495** | **1** | 0.5983 | 4 |
| WFG8 | 0.4414 | 3 | 0.4440 | 2 | **0.4201** | **1** | 0.6483 | 4 |
| WFG9 | 0.1991 | 2 | 0.2489 | 4 | 0.2202 | 3 | **0.1916** | **1** |
| | | 2.4 | | 2.8 | | **2.1** | | 2.9 |

conjecture about the new-solutions generating algorithm using historical information.

### 4.3 Universality analysis

Although the average IGD value and ranking score intuitively reflect the performance of each approach on specific problems, they cannot intuitively reflect the universality of a certain approach. Universality represents a spirit with equilibrium that does not require the approach to be dominant in a particular problem but requires the approach should not perform poorly in any situation. Based on the above thinking, we made a horizontal comparison of the four different approaches` IGD value. The specific evaluation methods are as follows:

$$E_j|_{1\le j\le 4} = \sum_{i=1,i\neq 2}^{i=9} \left(\frac{IGD_j^i}{IGD_{mean}^i} - IGD_{mean}^i\right)$$

$$IGD_{mean}^i = \frac{\sum_{i=1,i\neq 2}^{i=9} IGD_j^i}{4}|_{1\le j\le 4} \qquad (5)$$

where $j$ is the index represents MOEA/D-EF, DE, IDEAL and JADE, respectively; $i$ is the index of WFG series problems; $\frac{IGD_j^i}{IGD_{mean}^i}$ scales the average IGD values of each approach in the same problem to the same scale, The result of the summation $E_j$, is the final universality degree of the approach. The results are listed in Table 4. According to the definition given in (5), a negative $E_j$ indicates that this approach performs better than the average of all. The smaller the value of $E_j$, the more obvious the advantage of this approach.

TABLE 2. AVERAGE IGD VALUE AND RANKING SCORE FOR PROBLEMS WITH 50 VECTORS.

| V50 | EF | | DE | | IDEAL | | JADE | |
|---|---|---|---|---|---|---|---|---|
| | MV | R | MV | R | MV | R | MV | R |
| WFG1 | 1.4676 | 3 | 1.4663 | 2 | **1.4635** | **1** | 1.4925 | 4 |
| WFG3 | 0.2628 | 2 | 0.2880 | 3 | **0.2617** | **1** | 0.7330 | 4 |
| WFG4 | 0.3375 | 2 | 0.3627 | 3 | **0.3177** | **1** | 0.5045 | 4 |
| WFG5 | 0.1896 | 2 | 0.2531 | 4 | 0.1903 | 3 | **0.1814** | **1** |
| WFG6 | 0.1730 | 2 | 0.2827 | 4 | 0.2090 | 3 | **0.1680** | **1** |
| WFG7 | 0.4240 | 2 | 0.4410 | 3 | **0.4060** | **1** | 0.7042 | 4 |
| WFG8 | 0.4794 | 2 | 0.4982 | 3 | **0.4702** | **1** | 0.7507 | 4 |
| WFG9 | 0.1842 | 2 | 0.2906 | 4 | 0.2052 | 3 | **0.1763** | **1** |
| | | 2.1 | | 3.3 | | **1.8** | | 2.9 |

TABLE 3. AVERAGE IGD VALUE AND RANKING SCORE FOR PROBLEMS WITH 100 VECTORS.

| V100 | EF | | DE | | IDEAL | | JADE | |
|---|---|---|---|---|---|---|---|---|
| | MV | R | MV | R | MV | R | MV | R |
| WFG1 | 1.4760 | 3 | **1.4684** | **1** | 1.4688 | 2 | 1.4965 | 4 |
| WFG3 | 0.3995 | 2 | 0.4166 | 3 | **0.3917** | **1** | 0.8600 | 4 |
| WFG4 | 0.3622 | 2 | 0.4094 | 3 | **0.3447** | **1** | 0.5995 | 4 |
| WFG5 | 0.1922 | 2 | 0.1961 | 3 | 0.1984 | 4 | **0.1865** | **1** |
| WFG6 | 0.1811 | 2 | 0.3307 | 4 | 0.2051 | 3 | **0.1652** | **1** |
| WFG7 | 0.4796 | 2 | 0.5117 | 3 | **0.4560** | **1** | 0.7669 | 4 |
| WFG8 | 0.5199 | 2 | 0.5470 | 3 | **0.4956** | **1** | 0.8012 | 4 |
| WFG9 | 0.1776 | 2 | 0.3238 | 4 | 0.2034 | 3 | **0.1687** | **1** |
| | | 2.1 | | 3 | | **2** | | 2.9 |

TABLE 4. THE UNIVERSALITY DEGREE CALUCLATED FROM AVERAGE IGD VALUE

| | EF | DE | IDEAL | JADE |
|---|---|---|---|---|
| V30 | **-0.9589** | -0.2320 | -0.8520 | 2.0432 |
| V50 | **-1.0821** | -0.2639 | -0.9193 | 1.5496 |
| V100 | **-1.0087** | 0.6011 | -0.8942 | 1.3016 |

## 5. Conclusion

In this paper, a new framework MOEA/D-EF, which can contain multi new-solutions generating algorithms with different search characteristics, is proposed to solve multi-objective optimization problems efficiently. Compared with some existed frameworks with similar motivation, we made adaptive modification on MOPs for candidate algorithms. At the same time, we introduce historical information into the process of generating new solutions. To a certain extent, the results of numerical experiments support our view that historical information con-tributes to efficient generation of new solutions; The results of the universality degree indicate that the proposed new framework is more widely applicable to different problems.

### References

[1] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on evolutionary computation, 2007, 11(6): 712-731.

[2] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997).

[3] M. Crepinsek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Comput. Surv. 45 (3) (2013).

[4] Li G, Lin Q, Cui L, et al. A novel hybrid differential evolution algorithm with modified CoDE and JADE[J]. Applied Soft Computing, 2016, 47: 577-599.

[5] Zhang S X, Zheng S Y, Zheng L M. An efficient multiple variants coordination framework for differential evolution[J]. IEEE transactions on cybernetics, 2017, 47(9): 2780-2793.

[6] J.Q. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945-958.