A Cost Value Based Evolutionary Many-Objective Optimization Algorithm With Neighbor Selection Strategy

Jiawei Yuan, Hai-Lin Liu and Fangqing Gu
Guangdong University of Technology
Guangzhou, China
hlliu@gdut.edu.cn

***Abstract*—Based on the ideas of minimizing the loss of convergence and diversity of the candidate solution set, this paper proposes a cost value based evolutionary many-objective algorithm with neighbor selection strategy. In this work, the cost value of each solution is the mutual evaluation from other ones in current population. By this way, the proposed algorithm, named MEMO, can easily recognize the dominated and the non-dominated solutions and assess the contribution of convergence and diversity of each solution among the candidate solution set. To further enhance the performance of proposed algorithm, a neighbor selection strategy is also suggested in this paper. Simulation experiments on MaF series indicate that the proposed MEMO is superior to IBEA, MOEA/D, NSGA-III and RVEA in terms of effectiveness and robustness.**

***Index Terms*—Evolutionary algorithm; many-objective optimization; selection strategy.**

I. Introduction

Without loss of generality, a multi-objective optimization problem (MOP) can be stated as follows:

$$\begin{array}{lll} \min & f(\mathbf{x}) & = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})) \\ s.t. & \mathbf{x} & \in \prod_{i=1}^{n} [a_i, b_i] \subset \mathbf{R}^n \end{array} \tag{1}$$

where $\prod_{i=1}^{n}[a_i, b_i]$ is the feasible region in the decision space. $f : \prod_{i=1}^{n}[a_i, b_i] \to \mathbf{R}^m$ consists of m real objective functions, and $\mathbf{R}^m$ is called the objective space. For simplicity, we assume that all the objective functions $f_1, \cdots, f_m$ of (1) are nonnegative in this paper. When the number of objectives in problem (1) is greater than 3, it is called as a many-objective optimization problem (MaOP).

Since many problems in real-life can be be transformed into the MOP [1], [2], [3], many of evolutionary multi-objective optimization (EMO) algorithms have been presented and amply shown their advantage of finding a set of non-dominated solutions with well convergence and diversity in different kinds of MOPs in the last few decades [4], [5], [6], [7]. Nevertheless, most of these EMO algorithms still face a number of challenges in solving MaOPs. First, with an increase in the number of objectives, the proportion of non-dominated solutions in a randomly generated population becomes exponentially large. The Pareto-based EMO algorithms, such as NSGA-II [4] and SPEA2 [6], will have a little selection pressure to push the population towards the Pareto front (PF). Second, as the decomposition-based EMO algorithms' performance depends on the parameter setting, just as the weight vectors in MOEA/D [8] and the reference points in NSGA-III [9]. Furthermore, for indicator-based EMO algorithms, e.g. HypE [10], the high computational cost and intolerable time requirement are the serious problem in handling MaOPs. Therefore, the research on MaOPs is far from being fully explored.

In this paper, we propose a simple but efficient cost value based EMO algorithm, named MEMO, for MaOPs. The main ideas of this paper is to pick over some solutions to keep a good represention among the candidate set with the minimum loss of convergence and diversity. Therein, under the guidance of the thought that the candidate solutions are an approximation to the PF, all candidate solutions in objective space are regarded as the promising search points. Then the mutual evaluation of a solution assessed by the another is defined as the maximal ratio between their corresponding objectives. And the mutual evaluation of a solution in current population is defined as the worst assessment of the others. Under this definition, some solutions with good convergence and distribution among the candidate set will possess a better mutual evaluation. By this way, we can quickly recognize the dominated and non-dominated solutions in the candidate solution set. The solution whose cost value is larger (or smaller) than 1 is a non-dominated (or dominated) solution. Furthermore, since the definition of mutual evaluation is not sensitive to the scales of objective functions, it can work well in the problems with different objective scales. To further enhance the performance of proposed algorithm, we adopt a neighbor selection strategy in the selection process, in which a half of population will be firstly picked out to normalize all non-dominated solutions, and then non-dominated solutions with smaller mutual evaluation and distance will be omitted.

This work was supported by the National Science Foundation of China under Grant 61673121 and 61703108, the Natural Science Foundation of Guangdong Province (2017A030310467) and the Programme of Science and Technology of Guangzhou (201804010352).

Simulation results show that the proposed MEMO performs better than many state-of-the-art EMO algorithms, including IBEA [11], MOEA/D, NSGA-III and RVEA [12] on the benchmark functions for CEC'2018 competition on many-objective optimization [13].

The remainder of this paper is organized as follows. In Section II, we preset the definition of mutual evaluation and analyze its characteristics in a simple way. Section III describes the neighbor selection strategy in details. Besides, its processing and main techniques are also given in this section. The main framework of the proposed algorithm is shown in Section IV. Then in Section V, we conduct the experiments on the proposed MEMO, and take four state-of-the-art EMO algorithms into comparison. Finally, we draw our conclusions in Section VI.

II. Cost Value

It is important to evaluate the quality (i.e. convergence and diversity) of each solution and decide which one should be accepted or rejected during the evolution process in EMO algorithms. We use a cost function to measure the loss, which is named as mutual evaluation in this paper, of a solution evaluated by the others in the candidate solution set.

Let $\mathbf{f^i} = (f_1^i, f_2^i \cdots, f_m^i)$ be the objective vector of a candidate solution. Then the mutual evaluation of solution $\mathbf{x}_i$ evaluated by solution $\mathbf{x}_j$ is defined as:

$$cv_{ij} = \max_p f_p^j / f_p^i. \tag{2}$$

And its mutual evaluation among the candidate solution set $\mathbf{X}$ is calculated as:

$$CV_i = \min_{j \in \mathbf{I} \setminus i} cv_{ij}, \quad \mathbf{I} = \{1, 2, \cdots, N^*\}. \tag{3}$$

where N^* is the number of the solutions in the candidate solution set $\mathbf{X}$.

It is clear that CV_i is not sensitive to the scales of f_p ($p = 1, \cdots, m$) and it has the following two characteristics:

- $CV_i > 1 \Leftrightarrow \mathbf{x}_i$ is a non-dominated solution in $\mathbf{X}$;
- $CV_i \leq 1 \Leftrightarrow \mathbf{x}_i$ is a dominated solution in $\mathbf{X}$.

Figure 1 illustrates the mutual evaluation of four points in the case of $m = 2$. From this figure, we can see that a solution with good convergence and distribution among the candidate solutions will has a larger mutual evaluation. For example, as point B has the worst convergence and is dominated by A, its cost value is also the worst. Besides, due to the better performance on distribution, i.e. farther away from other solutions, the cost value of point A is larger than points C and D. In the above example, if we want to preserve only two solutions, then the solution B will be rejected firstly and C is secondly.

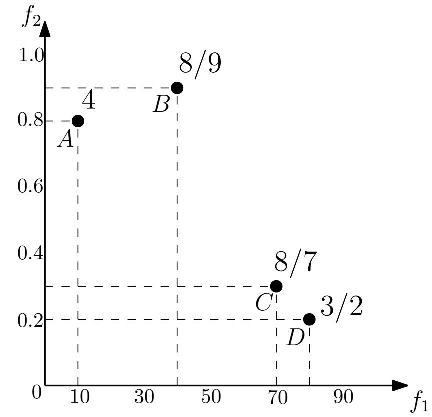

Fig. 1: The mutual evaluation of four points in two-dimension space.

III. Neighbor Selection Strategy

When the size N^* of the non-dominated solutions in current population $\mathbf{X}$ is smaller than the population size N, we select N solution with the greatest mutual evaluation. Otherwise, we first remove all dominated solutions from the candidate solution set. And then we design a neighbor selection strategy based on the information of mutual evaluation to remove the non-dominated solutions in this paper. The main idea of it is described as follows.

In the first step of the proposed strategy, a mutual evaluation matrix $CV = [cv_{ij}]_{N^* \times N^*}$ is calculated according to Eq. (2). After that, $N/2$ solutions with the best mutual evaluation in $\mathbf{X}$ will be preserved according to the following cycle operation: in each step, the solution with the worst mutual evaluation in $\mathbf{X}$ will be removed and any solutions assessed by this solution need to update their mutual evaluation. These $N/2$ solutions are used to normalize the objectives of candidate solutions for the next step:

$$\overline{f}_p = \frac{f_p - f_p^{min}}{\widetilde{f}_p^{max} - f_p^{min}}, \quad p = 1, \cdots, m. \tag{4}$$

where f_p^{min} is the minimum value of the pth objective found so far and $\widetilde{f}_p^{max}$ is the maximum value in the selected $N/2$ solutions. If any solution's objective $\overline{f}_p^i > 1$, it will be set as $\overline{f}_p^i = 1$ in this paper. Therefore, all candidate solutions have been mapped into the normal region $[0, 1]^m$.

In the second step, a parallel distance d_{ij} between any two solutions in candidate solution set is got by the Eq. (5), and its geometric significance in the case of two objectives has been shown in Fig. 2. And then, the two solutions with the minimum distance among all the candidate solutions are picked out and taken into comparison. The solution with a smaller mutual evaluation, which is determined by the cost matrix calculated in the first step, will be removed from the candidate solution set. The above procedure will be executed for $(N^* - N)$ times. After this, the N solutions with the better convergence and distribution in candidate solutions have been determined.

$$d_{ij} = [\sum_{p=1}^{m} (\overline{f}_p^i - \overline{f}_p^j)^2 - (\sum_{p=1}^{m} (\overline{f}_p^i - \overline{f}_p^j))^2 / m]^{1/2}. \tag{5}$$

The pseudocode of the selection mechanism in proposed MEMO is given in Algorithm 1.

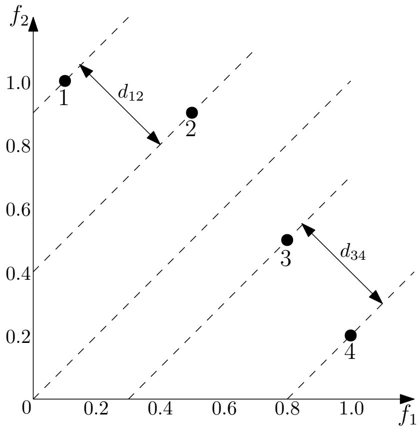

Fig. 2: The parallel distance displays in the case of $m = 2$.

Algorithm 1: Selection mechanism in MEMO

Input:
- The candidate solution set: $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{N^*}\}$;
- Ideal point in objective space: $\mathbf{f}^* = (f_1^{min}, \cdots, f_m^{min})$;
- The preset size of population: N;

Output: N solutions in $\mathbf{X}$.

1 Translate the objective of each solution $(f_p - f_p^{min})$, and calculate the mutual evaluation c_{ij} between any two candidate solutions by using the Eq. (2). Then we get a cost matrix $CV = [cv_{ij}]$;
2 Let $\widetilde{\mathbf{X}} = \mathbf{X}$, $\widetilde{CV} = CV$, $N_1 = N^* - N/2$;
3 **for** $i = 1 : N_1$ **do**
4   Compute the mutual evaluation $\widetilde{CV}_i$ of each solution in $\widetilde{\mathbf{X}}$ with Eq. (3);
5   Determine the worst solution who possesses the minimum mutual evaluation $k = \arg\min \widetilde{CV}_i$, and removed it from $\widetilde{\mathbf{X}}$: $\widetilde{\mathbf{X}} = \widetilde{\mathbf{X}} \setminus \widetilde{\mathbf{x}}_k$;
6   Update $\widetilde{CV}$: let $\widetilde{CV}[k,:] = \emptyset$, $\widetilde{CV}[:,k] = \emptyset$;
7 **end**
8 Using Eq. (4) to normalize the each objective. If any $\overline{f}_p^i > 1$, then let $\overline{f}_p^i = 0$;
9 Compute the parallel distance of any two solutions d_{ij} to form the distance matrix $D = [d_{ij}]$;
10 **for** $j = 1 : (N^* - N)$ **do**
11   Pick out the two solutions with the minimum distance;
12   Compare they mutual evaluation by using C and reject the worst one $\mathbf{x}_q$ from $\mathbf{X}$: $\mathbf{X} = \mathbf{X} \setminus \mathbf{x}_q$;
13   Update the matrixes C and D with the form in Line 6;
14 **end**

IV. Algorithm Framework

We name the solution $\mathbf{x}_k$ is the neighbor of $\mathbf{x}_i$ if $k = \arg\min_{j \in \mathbf{I} \setminus i} cv_{ij}$. For MaOPs, it is important to pick a reasonable partner for each solution to generate the offsprings. As the neighbor of a solution should have the similar property in solving the same problem, we let the neighbor be the partner of each solution in a big probability P_s. To further improve the quality of offsprings, the mating pool is confined to be composed by the non-dominated solutions in current population. More details of the proposed algorithm have been shown in Algorithm 2.

Algorithm 2: Pseudocode of MEMO

Input:
- The size of population: N;
- Probability in selecting partner: P_s;
- A stopping criterion;

Output: The non-dominated solutions in current population.

Initialize the current population with the size of N randomly;
while *The stopping criteria is unsatisfied* **do**
    Compute the mutual evaluation among any two solutions in current population, then pick out the non-dominated solutions to form the mating pool;
    for *Each solution* $\mathbf{x}_i$ *in mating pool* **do**
        if $rand < P_s$ **then**
            Let $\mathbf{x}_i$ cost neighbor $\mathbf{x}_{i'}$ be its spouse;
        else
            Randomly choose a solution $\mathbf{x}_j$ in mating pool and set it to be $\mathbf{x}_i$ spouse;
        end
        Generate a new offspring $\mathbf{y}_i$ by using genetic operators;
    end
    Mix the current population $\mathbf{X}$ and the offsprings $\mathbf{Y} = \{\mathbf{y}_i\}$.
    if *the number of solutions with* $CV > 1$ *is smaller than* N **then**
        Select N solutions with the greatest mutual evaluation
    else
        Select N solutions from it by the use of Algorithm 1
    end
end
Output the non-dominated solutions of the current population.

TABLE III: Statistically significant analysis. "+", "=" and "−" stand for the numbers of problems that the results obtained by MEMO are significantly better than, similar to and worse than that obtained by the other four algorithms.

Matrics	IBEA			MOEA/D			NSGA-III			RVEA		
	+	=	−	+	=	−	+	=	−	+	=	−
IGD	40	2	3	44	0	1	40	1	4	36	3	6
NHV	40	3	2	45	0	0	41	4	0	41	2	2

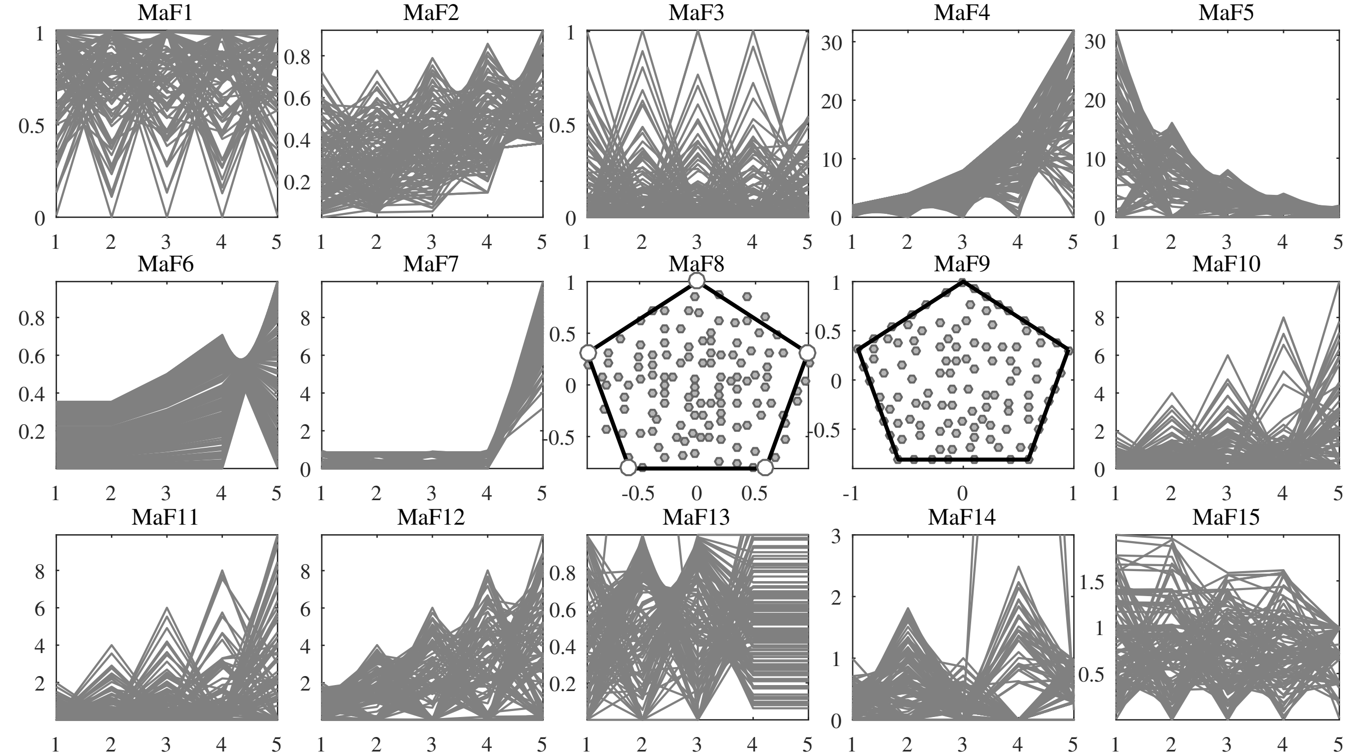

Fig. 3: Plot of the non-dominated solutions with median IGD-matric value found by MEMO in the case of $M = 5$.

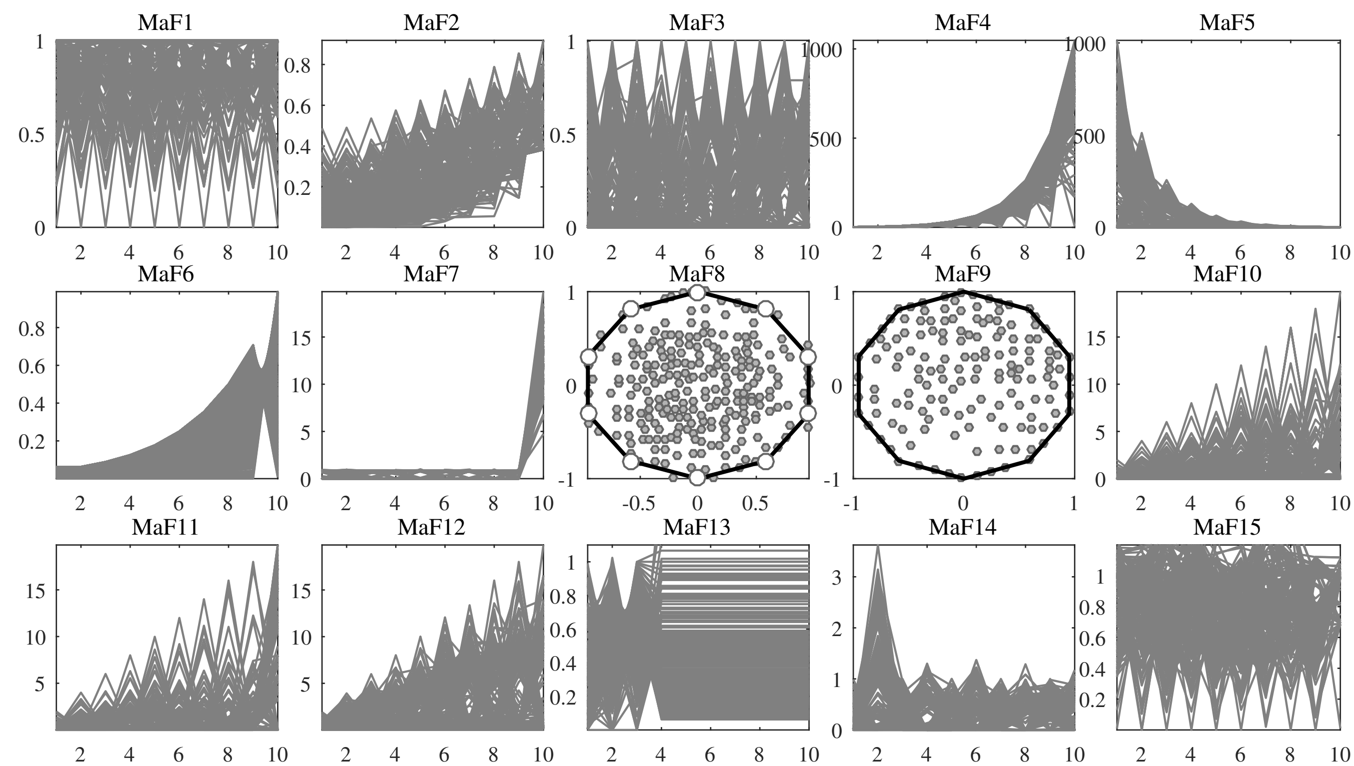

Fig. 4: Plot of the non-dominated solutions with median IGD-matric value found by MEMO in the case of $M = 10$.

TABLE I: The IGD results showed in PlatEMO platform after 20 runs of the five algorithms.Best performance is highlighted in bold face with gray background.

Problem	M	MEMO	IBEA	MOEA/D	NSGA-III	RVEA
MaF1	5	**1.3510e-01 (2.29e-03)**	1.4490e-01 (5.67e-03)	2.3196e-01 (1.50e-02)	2.0754e-01 (1.37e-02)	3.4280e-01 (1.26e-01)
	10	**2.4297e-01 (3.01e-03)**	2.7313e-01 (6.80e-03)	4.6981e-01 (1.55e-02)	2.9239e-01 (6.82e-03)	6.4577e-01 (6.62e-02)
	15	**2.8974e-01 (2.09e-03)**	3.1066e-01 (5.04e-03)	5.9072e-01 (2.69e-02)	3.1931e-01 (6.77e-03)	6.9098e-01 (5.27e-02)
MaF2	5	1.1088e-01 (2.67e-03)	**1.0682e-01 (1.62e-03)**	1.2438e-01 (7.10e-04)	1.3013e-01 (4.24e-03)	1.2775e-01 (1.78e-03)
	10	**1.6564e-01 (2.82e-03)**	1.9637e-01 (1.28e-02)	3.2426e-01 (1.50e-03)	2.2649e-01 (4.00e-02)	2.6243e-01 (9.06e-02)
	15	**1.7797e-01 (3.76e-03)**	3.0040e-01 (1.60e-02)	3.6363e-01 (3.35e-04)	2.1055e-01 (1.10e-02)	5.8313e-01 (1.88e-01)
MaF3	5	**7.1960e-02 (1.83e-03)**	4.7994e-01 (2.65e-01)	1.2401e-01 (1.79e-03)	9.7243e-02 (1.81e-03)	1.0170e-01 (1.08e-01)
	10	1.7473e-01 (1.26e-02)	3.4457e-01 (5.54e-02)	1.4498e-01 (8.93e-04)	5.2905e+03 (1.73e+04)	**9.7997e-02 (3.44e-02)**
	15	1.9199e-01 (7.17e-02)	3.4658e-01 (1.88e-01)	1.3056e-01 (4.99e-04)	1.7186e+06 (9.54e+06)	**9.6296e-02 (9.64e-03)**
MaF4	5	**2.0007e+00 (2.30e-02)**	5.6248e+00 (3.62e-01)	9.3763e+00 (5.14e-01)	3.2903e+00 (2.97e-01)	4.7958e+00 (1.15e+00)
	10	**4.8890e+01 (2.17e+00)**	1.7064e+02 (7.22e+00)	4.5532e+02 (1.74e+01)	1.0379e+02 (6.90e+00)	2.1172e+02 (5.27e+01)
	15	**1.3559e+03 (1.24e+02)**	4.5896e+03 (2.27e+02)	1.9003e+04 (1.08e+03)	4.0714e+03 (2.74e+02)	7.2753e+03 (2.27e+03)
MaF5	5	**2.1595e+00 (4.16e-02)**	2.2676e+00 (2.60e-01)	9.9957e+00 (2.69e+00)	2.2614e+00 (2.93e-01)	2.5311e+00 (9.79e-01)
	10	**5.1179e+01 (1.73e+00)**	5.8236e+01 (2.45e+00)	3.0107e+02 (2.09e+00)	8.7795e+01 (1.97e+00)	1.0031e+02 (6.76e+00)
	15	**1.2191e+03 (9.73e+01)**	1.3337e+03 (8.77e+01)	7.3232e+03 (1.04e+00)	2.6007e+03 (3.92e+02)	3.2798e+03 (5.24e+02)
MaF6	5	**3.7130e-03 (1.32e-04)**	5.6961e-02 (1.03e-02)	1.3404e-01 (1.85e-01)	4.6709e-02 (3.98e-02)	8.9045e-02 (2.61e-02)
	10	**1.8174e-03 (3.07e-05)**	2.4392e-01 (1.39e-01)	2.5018e-01 (2.72e-01)	3.1592e-01 (8.95e-02)	1.2802e-01 (2.33e-02)
	15	**1.2050e-03 (1.68e-05)**	4.4056e-01 (2.02e-01)	2.0944e-01 (2.30e-01)	3.2572e-01 (8.42e-02)	2.8566e-01 (2.21e-01)
MaF7	5	**2.6312e-01 (5.39e-03)**	3.9160e-01 (1.20e-01)	1.0613e+00 (1.11e-01)	3.3990e-01 (2.27e-02)	4.4829e-01 (9.48e-04)
	10	**8.3341e-01 (3.72e-03)**	9.0749e-01 (1.18e-01)	2.1991e+00 (4.15e-01)	1.0329e+00 (8.78e-02)	2.8793e+00 (2.78e-01)
	15	**1.4985e+00 (2.07e-02)**	2.2448e+00 (6.36e-01)	3.8367e+00 (8.18e-01)	4.4222e+00 (8.99e-01)	3.9586e+00 (6.72e-01)
MaF8	5	**1.0165e-01 (1.68e-03)**	9.3994e-01 (9.33e-02)	3.0586e-01 (2.38e-02)	2.5309e-01 (3.22e-02)	4.8398e-01 (7.30e-02)
	10	**1.1665e-01 (2.40e-03)**	9.7428e-01 (7.62e-02)	9.5270e-01 (1.78e-02)	4.1130e-01 (5.72e-02)	7.9558e-01 (8.32e-02)
	15	**1.2030e-01 (1.20e-03)**	1.0601e+00 (5.43e-02)	1.3770e+00 (1.25e-02)	3.7178e-01 (5.42e-02)	1.3057e+00 (1.97e-01)
MaF9	5	**1.0638e-01 (2.87e-03)**	9.6721e-01 (7.08e-02)	1.4602e-01 (9.14e-03)	3.1770e-01 (1.06e-01)	4.0006e-01 (7.21e-02)
	10	**1.8826e-01 (1.10e-01)**	9.8942e-01 (1.03e-01)	6.1117e-01 (9.16e-01)	6.5845e-01 (2.20e-01)	8.4125e-01 (2.25e-01)
	15	**1.1205e-01 (3.59e-03)**	1.0575e+00 (6.87e-02)	4.8232e+00 (5.30e+00)	3.9979e-01 (1.81e-01)	1.3677e+00 (2.85e-01)
MaF10	5	5.3300e-01 (2.23e-02)	5.6053e-01 (2.83e-02)	1.2239e+00 (5.02e-03)	4.6352e-01 (1.27e-02)	**4.5682e-01 (1.97e-02)**
	10	1.3300e+00 (4.63e-02)	**1.0786e+00 (3.27e-02)**	2.6677e+00 (1.65e-01)	1.0952e+00 (6.46e-02)	1.2840e+00 (5.29e-02)
	15	1.9228e+00 (8.99e-02)	**1.4744e+00 (5.31e-02)**	3.1112e+00 (1.75e-01)	1.5819e+00 (8.31e-02)	1.9319e+00 (2.74e-01)
MaF11	5	**5.4072e-01 (6.54e-02)**	1.3228e+00 (6.06e-02)	5.1286e+00 (2.87e-02)	8.2197e-01 (1.29e-02)	1.5654e+00 (5.34e-01)
	10	**1.5452e+00 (2.13e-01)**	5.7986e+00 (2.21e-01)	1.6660e+01 (4.63e-02)	4.9082e+00 (1.91e+00)	8.2964e+00 (2.19e+00)
	15	**2.8877e-03 (2.13e-03)**	1.0413e+01 (7.41e-01)	2.7294e+01 (8.07e-02)	1.2645e+01 (1.73e+00)	1.8549e+01 (3.83e+00)
MaF12	5	1.1350e+00 (9.57e-03)	1.1598e+00 (1.52e-02)	1.5373e+00 (6.31e-02)	**1.1197e+00 (9.82e-03)**	1.1233e+00 (4.13e-03)
	10	**4.0648e+00 (3.10e-02)**	4.1593e+00 (3.03e-02)	8.7864e+00 (8.84e-01)	4.5672e+00 (4.57e-02)	4.4967e+00 (7.00e-02)
	15	**6.5819e+00 (5.42e-02)**	6.7263e+00 (7.34e-02)	1.4930e+01 (8.91e-01)	8.2026e+00 (2.96e-01)	7.1267e+00 (2.11e-01)
MaF13	5	**1.0551e-01 (1.77e-02)**	7.4524e-01 (6.09e-02)	1.7142e-01 (2.01e-02)	2.3634e-01 (1.80e-02)	6.6051e-01 (1.66e-01)
	10	**1.1301e-01 (1.58e-02)**	1.1831e+00 (1.19e-01)	1.0383e+00 (1.18e-01)	2.5314e-01 (3.08e-02)	1.0034e+00 (2.89e-01)
	15	**1.0650e-01 (1.47e-02)**	1.4601e+00 (1.79e-01)	1.3207e+00 (1.82e-01)	2.8240e-01 (4.52e-02)	9.7000e-01 (3.63e-01)
MaF14	5	**3.4493e-01 (3.74e-02)**	1.6284e+00 (7.99e-01)	8.2302e-01 (2.01e-01)	6.9276e-01 (2.04e-01)	7.3626e-01 (1.98e-01)
	10	4.8353e-01 (4.56e-02)	1.0671e+00 (1.67e-01)	**4.2482e-01 (1.24e-01)**	2.1551e+00 (1.54e+00)	6.6011e-01 (5.48e-02)
	15	**6.3215e-01 (3.28e-02)**	1.1811e+00 (8.01e-02)	8.9476e-01 (2.18e-01)	1.2494e+00 (1.88e-01)	8.3580e-01 (1.79e-01)
MaF15	5	**2.5083e-01 (1.31e-02)**	8.6112e-01 (6.37e-02)	6.0662e-01 (6.27e-02)	1.2796e+00 (1.11e-01)	5.9689e-01 (3.92e-02)
	10	**5.1234e-01 (6.18e-02)**	7.9770e-01 (1.68e-01)	9.8162e-01 (4.18e-02)	1.1364e+00 (1.15e-01)	9.6567e-01 (4.36e-02)
	15	**6.0636e-01 (8.60e-02)**	1.3033e+00 (1.36e-02)	1.0783e+00 (3.37e-02)	4.5045e+00 (1.60e+00)	1.1602e+00 (4.05e-02)

V. Experiment Study

In order to verify the effectiveness of the proposed MEMO, 15 test problems in MaF series, which cover a good representation of various real-world scenarios and are set to be the benchmark functions for CEC'2018 competition, have been carried out in this paper. For a fair comparison, we execute the code of the proposed algorithm in official PlatEMO platform[1]. Four state-of-the-art EMO algorithms, including IBEA, MOEA/D, NSGA-III and RVEA, have been taken into comparison.

[1]Provided in http://bimk.ahu.edu.cn/index.php?s=/Index/Software/index.html

A. Parameter Setting

All the sensitive parameters for the competition have been set in the official PlatEMO platform as below:

- Number of objectives (M): 5, 10, 15;
- Maximum population size: 240;
- Maximum number of fitness evaluations (FEs): $\max\{10^5, 10^4 \times D\}$, where D is number of decision variables;

TABLE II: The NHV results showed in PlatEMO platform after 20 runs of the five algorithms.Best performance is highlighted in bold face with gray background.

Problem	M	MEMO	IBEA	MOEA/D	NSGA-III	RVEA
MaF1	5	**9.8786e-03 (2.45e-04)**	9.7648e-03 (2.09e-04)	5.5499e-03 (3.57e-04)	4.7281e-03 (5.30e-04)	2.2323e-03 (1.06e-03)
	10	**5.5000e-07 (7.59e-07)**	4.4395e-07 (2.89e-08)	3.5076e-08 (7.37e-09)	4.6122e-07 (2.70e-08)	5.6577e-09 (2.64e-09)
	15	0.0000e+00 (0.00e+00)	**7.5198e-12 (7.68e-13)**	8.3007e-14 (1.78e-14)	6.2224e-12 (7.74e-13)	2.6455e-14 (1.05e-14)
MaF2	5	**1.9164e-01 (1.84e-03)**	1.8561e-01 (1.99e-03)	1.6162e-01 (7.20e-04)	1.5661e-01 (5.11e-03)	1.5059e-01 (2.57e-03)
	10	**2.2014e-01 (2.54e-03)**	2.1466e-01 (3.76e-03)	2.0880e-01 (3.66e-04)	2.1327e-01 (5.67e-03)	1.8361e-01 (2.33e-02)
	15	2.1052e-01 (4.22e-03)	**2.1790e-01 (3.66e-03)**	1.8018e-01 (1.09e-03)	1.5724e-01 (1.23e-02)	6.7603e-02 (1.64e-02)
MaF3	5	**9.9925e-01 (6.56e-05)**	6.8150e-01 (3.24e-01)	9.8415e-01 (1.67e-03)	9.9876e-01 (1.54e-04)	9.7973e-01 (8.84e-02)
	10	**9.9992e-01 (1.10e-04)**	8.8154e-01 (4.58e-02)	9.6712e-01 (1.27e-03)	4.7047e-02 (1.94e-01)	9.9630e-01 (1.52e-02)
	15	9.9852e-01 (2.59e-03)	8.6950e-01 (2.15e-01)	9.6697e-01 (9.54e-04)	1.8113e-01 (3.78e-01)	**9.9910e-01 (8.11e-04)**
MaF4	5	**1.0696e-01 (2.57e-03)**	6.0564e-03 (9.00e-04)	1.2214e-02 (2.62e-03)	6.2747e-02 (7.19e-03)	1.4583e-02 (6.83e-03)
	10	1.5542e-04 (4.08e-05)	1.1767e-08 (3.62e-08)	3.2035e-08 (7.41e-09)	**1.8110e-04 (8.67e-06)**	1.9026e-07 (3.60e-07)
	15	6.9270e-08 (2.33e-07)	2.1095e-12 (8.17e-12)	9.5648e-14 (9.72e-15)	**1.9241e-07 (8.84e-09)**	8.9053e-12 (3.76e-11)
MaF5	5	**7.8926e-01 (1.59e-03)**	**7.8956e-01 (1.40e-02)**	4.2084e-01 (1.32e-01)	7.7516e-01 (2.53e-02)	7.5693e-01 (5.50e-02)
	10	9.6714e-01 (9.79e-04)	**9.7338e-01 (2.62e-04)**	4.0226e-01 (6.32e-02)	9.6404e-01 (2.08e-02)	9.5143e-01 (9.55e-03)
	15	**9.9507e-01 (2.43e-04)**	**9.9588e-01 (1.27e-04)**	2.7619e-01 (6.32e-02)	9.8580e-01 (1.29e-02)	9.4269e-01 (1.79e-02)
MaF6	5	**1.2918e-01 (3.72e-04)**	1.1679e-01 (3.53e-03)	1.1179e-01 (2.85e-02)	1.2314e-01 (4.76e-03)	1.1331e-01 (7.37e-03)
	10	**1.0094e-01 (2.88e-04)**	9.1381e-02 (2.47e-03)	9.1760e-02 (2.48e-02)	5.9664e-02 (2.59e-02)	8.5293e-02 (1.96e-02)
	15	**9.5369e-02 (3.20e-04)**	3.5853e-02 (3.76e-02)	7.2397e-02 (3.88e-02)	7.1649e-02 (1.39e-02)	9.1700e-02 (7.11e-04)
MaF7	5	**2.5414e-01 (3.11e-03)**	2.5324e-01 (1.07e-02)	1.5979e-02 (3.03e-02)	2.3169e-01 (8.04e-03)	2.1537e-01 (5.67e-04)
	10	1.5023e-01 (5.84e-03)	**1.9632e-01 (5.05e-03)**	2.5466e-04 (6.32e-04)	1.7554e-01 (9.16e-03)	1.4665e-01 (1.56e-02)
	15	9.8660e-02 (2.22e-03)	**1.5766e-01 (1.08e-02)**	9.8634e-06 (5.11e-05)	8.4640e-02 (3.57e-02)	7.4318e-02 (5.94e-02)
MaF8	5	**1.2064e-01 (4.46e-04)**	5.4868e-03 (9.78e-04)	1.0264e-01 (2.05e-03)	8.7729e-02 (5.92e-03)	7.2137e-02 (7.63e-03)
	10	**1.0788e-02 (1.17e-04)**	8.2153e-05 (1.83e-05)	6.4623e-03 (1.71e-04)	9.0767e-03 (2.33e-04)	4.5608e-03 (6.74e-04)
	15	**6.6672e-04 (2.46e-05)**	4.3839e-07 (3.67e-07)	3.1357e-04 (8.45e-06)	5.1175e-04 (2.67e-05)	1.2058e-04 (4.67e-05)
MaF9	5	**3.1326e-01 (1.37e-03)**	5.3416e-02 (1.37e-02)	2.8499e-01 (4.24e-03)	2.2374e-01 (2.97e-02)	1.8611e-01 (1.87e-02)
	10	**1.5451e-02 (3.18e-03)**	9.3614e-04 (2.16e-04)	1.3035e-02 (3.49e-03)	7.4695e-03 (2.47e-03)	4.1645e-03 (1.10e-03)
	15	**1.3677e-03 (3.22e-05)**	3.2432e-05 (6.55e-06)	4.9688e-04 (3.75e-04)	8.1968e-04 (1.61e-04)	2.1656e-04 (7.56e-05)
MaF10	5	**9.9773e-01 (1.91e-04)**	9.8481e-01 (2.79e-03)	9.3860e-01 (7.70e-03)	**9.9770e-01 (1.43e-04)**	9.9651e-01 (4.21e-04)
	10	**1.0000e+00 (2.38e-06)**	9.9290e-01 (1.22e-03)	6.0743e-01 (1.09e-01)	9.9891e-01 (4.25e-04)	9.9404e-01 (1.43e-02)
	15	**1.0000e+00 (1.35e-16)**	9.9542e-01 (1.09e-03)	5.9250e-01 (1.32e-01)	9.9957e-01 (2.94e-04)	9.9853e-01 (5.03e-04)
MaF11	5	**9.9700e-01 (3.54e-04)**	9.7411e-01 (5.12e-03)	9.5467e-01 (3.87e-03)	9.9588e-01 (6.18e-04)	9.8859e-01 (3.23e-03)
	10	**9.9925e-01 (4.42e-04)**	9.8569e-01 (3.21e-03)	9.3503e-01 (4.70e-03)	9.9767e-01 (1.01e-03)	9.8825e-01 (3.90e-03)
	15	**9.9918e-01 (6.41e-04)**	9.9100e-01 (2.12e-03)	9.3986e-01 (6.12e-03)	9.9831e-01 (9.17e-04)	9.7128e-01 (5.65e-03)
MaF12	5	**7.7489e-01 (4.55e-03)**	7.3758e-01 (9.22e-03)	5.8624e-01 (6.59e-02)	7.1918e-01 (2.85e-02)	7.3781e-01 (1.05e-02)
	10	**9.1331e-01 (1.19e-02)**	8.7939e-01 (2.82e-02)	3.4533e-01 (1.38e-01)	8.7015e-01 (4.97e-02)	8.7081e-01 (4.55e-02)
	15	**9.0784e-01 (5.71e-02)**	9.0690e-01 (6.42e-03)	1.9792e-01 (1.12e-01)	8.7450e-01 (5.46e-02)	8.5329e-01 (4.09e-02)
MaF13	5	**2.7792e-01 (1.17e-02)**	4.3240e-03 (4.95e-03)	2.3412e-01 (1.73e-02)	2.0025e-01 (1.11e-02)	1.5987e-01 (1.98e-02)
	10	**1.4303e-01 (2.37e-03)**	9.8552e-06 (3.42e-05)	6.5568e-02 (3.19e-02)	9.0861e-02 (2.88e-02)	8.5503e-02 (1.08e-02)
	15	**9.3742e-02 (2.50e-03)**	7.8168e-08 (3.71e-07)	3.4832e-02 (2.26e-02)	5.4503e-02 (1.67e-02)	5.0762e-02 (1.39e-02)
MaF14	5	**7.7298e-01 (3.77e-02)**	2.5516e-03 (1.16e-02)	1.6324e-01 (1.65e-01)	3.2717e-01 (2.09e-01)	2.6723e-01 (2.00e-01)
	10	**9.3634e-01 (5.24e-02)**	8.0883e-02 (9.39e-02)	9.1471e-01 (1.58e-01)	2.3786e-02 (4.55e-02)	6.3138e-01 (9.28e-02)
	15	**9.0037e-01 (3.40e-02)**	1.1664e-02 (2.24e-02)	2.4394e-01 (2.31e-01)	1.8445e-02 (3.13e-02)	3.2247e-01 (2.17e-01)
MaF15	5	**6.8974e-02 (6.48e-03)**	7.4603e-04 (5.53e-04)	3.5100e-02 (8.70e-03)	1.3565e-06 (5.25e-06)	1.8243e-02 (5.25e-03)
	10	**3.1246e-05 (2.15e-05)**	1.7557e-05 (2.20e-05)	3.1855e-06 (2.32e-06)	0.0000e+00 (0.00e+00)	2.5078e-06 (2.22e-06)
	15	**4.3607e-08 (1.95e-07)**	6.5838e-16 (6.35e-16)	1.7594e-10 (1.61e-10)	0.0000e+00 (0.00e+00)	1.9894e-12 (3.30e-12)

- Number of independent runs: 20;
- Parameter in EAreal operator: $proC = 1$, $disC = 20$, $proM = 1$, $disM = 20$.

For the proposed algorithm, the probability in selecting partner is set to be $P_s = 0.7$. And the other four algorithms are from the platform. Besides, two performance metrics, i.e., IGD and NHV is used.

B. Simulation Results and Analysis

The comparison results of the proposed MEMO with the other algorithms in IGD and NHV, which displayed in the platform, are listed in Table I and II respectively. Furthermore, we conduct the Wilcoxon Rank Test with 0.05 significance level between the performance metrics obtained by MEMO with those in other four algorithms to get a statistical conclusion. And the test results show in Table III. Figs. 3, 4 and 5 plot the parallel coordinate diagram of the non-dominated solutions with the median IGD-metric value obtained by the proposed

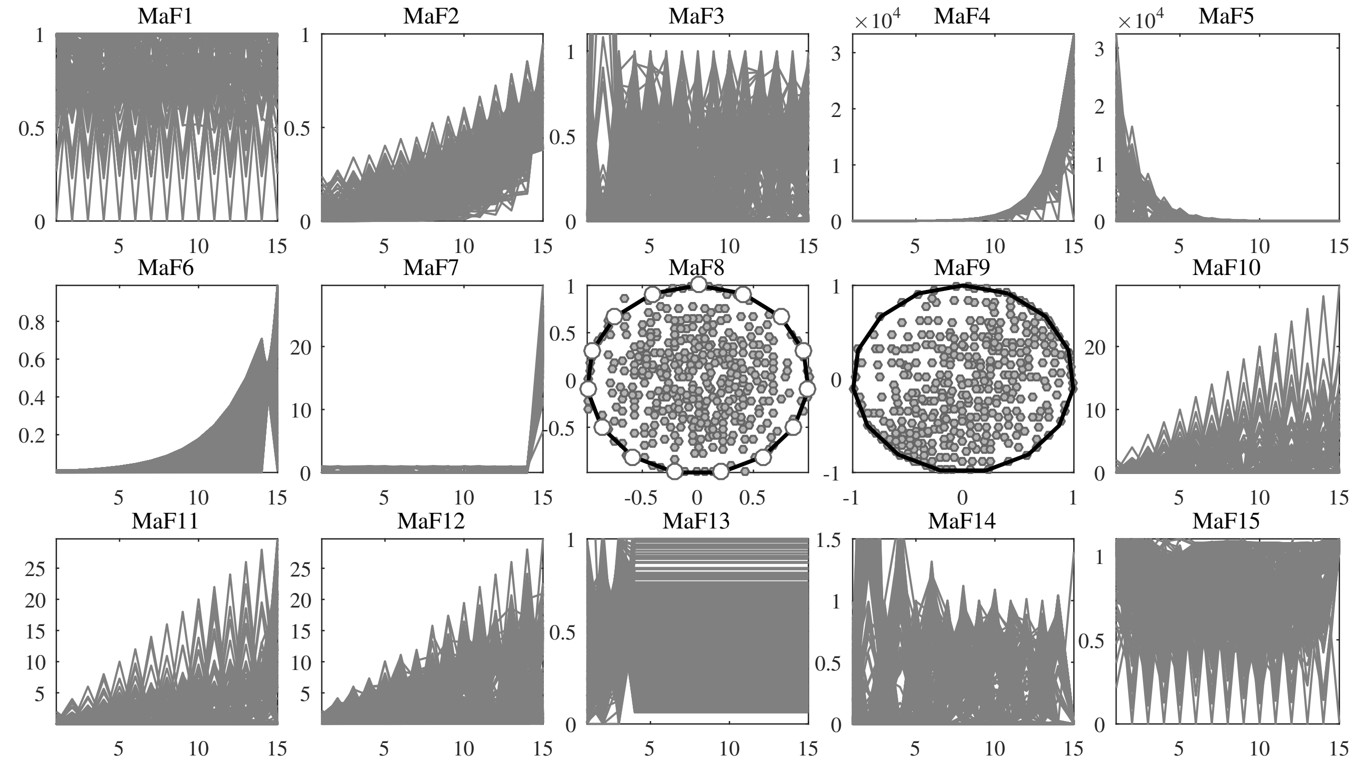

Fig. 5: Plot of the non-dominated solutions with median IGD-matric value found by MEMO in the case of $M = 15$.

MEMO.

From those tables, we can easily find that the proposed MEMO performs the best among these algorithms on the test problems, because the mutual evaluation is not sensitive to the scales of objectives. The proposed MEMO can work well on the problems with different objective scales. Besides, the performances of MOEA/D and NSGA-III are greatly depended on shape of the problem's PF. If the predefined reference points/weight vectors in them did not uniformly map into PF, they would have a poor performance. This will not happen in the proposed MEMO because the neighbor selection strategy can work well in identifying the distance between any two solutions in current population and then the solutions with better convergence and diversity will be preserved. Therefore, the proposed MEMO has a better performance and robustness in different kinds of MaOPs. This conclusion is also shown in Figs. 3, 4 and 5.

VI. Conclusion

In this paper, we proposed a simple and robust mutual evaluation based evolutionary many-objective optimization algorithm with neighbor selection strategy for MaOPs. Firstly, we presented the definition of mutual evaluation. The quality of a solution (i.e. convergence and distribution) is measured by the mutual evaluation that it assessed by the others in current population. As the mutual evaluation is not sensitive to the scales of objectives. Besides, a neighbor selection strategy based on mutual evaluation was introduced in selection operator to preserve the solutions with better convergence and distribution among the candidate solutions. Experimental results show that the proposed MEMO can obtain a promising results and is superior to the compared state-of-the-art EMO algorithms.

References

[1] N. Su, M. Zhang, M. Johnston, and K. C. Tan, "Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 193–208, 2014.

[2] H. L. Liu, F. Gu, Y. M. Cheung, and S. Xie, "On solving wcdma network planning using iterative power control scheme and evolutionary multiobjective algorithm," *Computational Intelligence Magazine IEEE*, vol. 9, no. 1, pp. 44–52, 2014.

[3] J. Yuan, H. L. Liu, and C. Peng, "Population decomposition-based greedy approach algorithm for the multi-objective knapsack problems," *International Journal of Pattern Recognition & Artificial Intelligence*, vol. 31, no. 04, pp. 1 759 006–, 2016.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.

[6] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," 2001.

[7] H. G. Wang and M. A. Liang, "Multi-objective particle swarm optimization," *Computer Engineering & Applications*, vol. 45, no. 4, pp. 82–85, 2008.

[8] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[9] 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 Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[10] J. Bader and E. Zitzler, "Hype: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2014.

[11] E. Zitzler and S. Knzli, "Indicator-based selection in multiobjective search," *Lecture Notes in Computer Science*, vol. 3242, pp. 832–842, 2004.

[12] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[13] 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.