

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/338022641>

A New Hypervolume-based Evolutionary Algorithm for Many-objective Optimization

Preprint · December 2019

DOI: 10.36227/techrxiv.11381016

CITATION

1

READS

260

2 authors, including:



[Ke Shang](#)

Southern University of Science and Technology

60 PUBLICATIONS 211 CITATIONS

SEE PROFILE

A New Hypervolume-based Evolutionary Algorithm for Many-objective Optimization

Ke Shang, and Hisao Ishibuchi, *Fellow, IEEE*

Abstract—In this paper, a new hypervolume-based evolutionary multi-objective optimization algorithm (EMOA), namely R2HCA-EMOA (R2-based Hypervolume Contribution Approximation EMOA), is proposed for many-objective optimization. The core idea of the algorithm is to use an R2 indicator variant to approximate the hypervolume contribution. The basic framework of the proposed algorithm is the same as SMS-EMOA. In order to make the algorithm computationally efficient, a utility tensor structure is introduced for the calculation of the R2 indicator variant. Moreover, a normalization mechanism is incorporated into R2HCA-EMOA to enhance the performance of the algorithm. Through experimental studies, R2HCA-EMOA is compared with three hypervolume-based EMOAs and several other state-of-the-art EMOAs on 5-, 10- and 15-objective DTLZ, WFG problems and their minus versions. Our results show that R2HCA-EMOA is more efficient than the other hypervolume-based EMOAs, and is superior to all the compared state-of-the-art EMOAs.

Index Terms—Many-objective optimization, Evolutionary algorithms, Hypervolume contribution approximation.

I. INTRODUCTION

MULTI-OBJECTIVE optimization problems (MOPs) have two or more objectives that need to be optimized simultaneously. When the number of objectives is more than three, an MOP is often referred to as a many-objective optimization problem (MaOP) [1]. In the evolutionary multi-objective optimization (EMO) community, MaOPs have posed a number of challenges to the development of evolutionary multi-objective optimization algorithms (EMOAs) [2].

EMOAs are often classified into three categories: Pareto-based, decomposition-based and indicator-based EMOAs. The Pareto-based EMOAs (e.g., NSGA-II [3] and SPEA2 [4]) use the Pareto dominance as the main selection criterion in environmental selection. The convergence of the population is achieved by Pareto-based selection criterion. However, the main challenge of the Pareto-based EMOAs for solving MaOPs is that the selection pressure towards the Pareto front (PF) is weakened in high-dimensional objective spaces (e.g.,

10-dimensional objective space). Almost all individuals in the population are non-dominated with each other in high-dimensional objective spaces, which makes the Pareto-based selection inefficient. The decomposition-based EMOAs (e.g., MOGLS [5] and MOEA/D [6]) decompose an MOP into a set of single-objective sub-problems. Then the sub-problems are optimized simultaneously in a collaborative manner. The convergence and diversity of the population are achieved by the scalarizing function criterion. However, for those decomposition-based EMOAs using a fixed weight vector grid, usually the obtained solutions cannot be uniformly distributed on the PF when the shape of the PF is irregular (e.g., the inverted triangular PF [7]). Thus, adaptively adjusting weight vectors in decomposition-based EMOAs is a hot research topic to deal with this issue [8], [9]. The indicator-based EMOAs (e.g., IBEA [10] and SMS-EMOA [11], [12]) use performance indicators (e.g., hypervolume [13], inverted generational distance (IGD) [14], R2 indicator [15]) as the main selection criterion. By properly setting the parameters of the indicators (e.g., the reference point for the hypervolume indicator and the reference point set for the IGD indicator) and optimizing the indicator values of the population, a solution set with good convergence and diversity can be obtained by the indicator-based EMOAs.

Among the indicator-based EMOAs, the hypervolume-based EMOAs (e.g., SMS-EMOA, FV-MOEA [16], HypE [17]) adopt the hypervolume indicator within the algorithms. The general idea of the hypervolume-based EMOAs is to transform an MOP into a single-objective optimization problem, where the single-objective is to maximize the hypervolume value of the population. The hypervolume indicator is widely used for performance evaluation of EMOAs. The advantages of the hypervolume-based EMOAs over the other EMOAs are twofold. One is that the hypervolume indicator is strictly Pareto compliant. Thus the hypervolume-based EMOAs are sensitive to any improvement to a set with respect to Pareto dominance. The other is that the hypervolume-based EMOAs can directly optimize the hypervolume of the population. Thus, good performance comparison results are likely to be obtained by the hypervolume-based EMOAs.

The key procedure in the hypervolume-based EMOAs is the calculation of the hypervolume contribution of each individual. It has been proven that the exact calculation of the hypervolume and hypervolume contributions is #P-hard [18], [19]. Even though there exist $O(N \log N)$ algorithms to compute the hypervolume and hypervolume contributions in two- and three-dimensional objective spaces [20], [21], where N is the population size, the computational time of the hypervolume

This work was supported by National Natural Science Foundation of China (Grant No. 61876075), 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) and the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008). (*Corresponding Author: Hisao Ishibuchi.*)

K. Shang and H. Ishibuchi are with Shenzhen Key Laboratory of Computational Intelligence, University Key Laboratory of Evolving Intelligent Systems of Guangdong Province, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: kshang@foxmail.com; hisao@sustc.edu.cn).

and hypervolume contributions increases exponentially with the increase in the number of objectives. This limited the applicability of the hypervolume-based EMOAs for solving MaOPs with many objectives (e.g., more than 10 objectives). In order to overcome this drawback, the use of hypervolume approximation (instead of exact calculation) has been proposed for hypervolume-based EMOAs. HypE is a representative hypervolume-based EMOA which uses a Monte Carlo sampling method for hypervolume approximation. However, in order to achieve a good hypervolume approximation, a large number of sampling points are needed in HypE. Thus, it is still a time-consuming algorithm.

The above discussions motivate us to develop a new EMOA which inherits the advantage of the hypervolume-based EMOAs and at the same time is computationally efficient for solving MaOPs. In order to fulfil this goal, an efficient hypervolume approximation method is needed. Currently, there are mainly two methods to approximate the hypervolume. One is the Monte Carlo sampling method which is adopted in HypE. The other method is the achievement scalarizing function method proposed in [22], [23]. The basic idea of this method is to use the average length of a set of line segments from the reference point to the hypervolume attainment surface to approximate the hypervolume. Recently, a new R2 indicator is proposed to approximate the hypervolume [24], which is an extension of the achievement scalarizing function method. Based on [24], an R2-based hypervolume contribution approximation method is proposed in [25]. The proposed method shows its superiority over the Monte Carlo sampling method [26] for the hypervolume contribution approximation in terms of both the runtime and the approximation accuracy.

In this paper, a new hypervolume-based evolutionary algorithm, namely R2HCA-EMOA, is proposed for many-objective optimization. The proposed algorithm, which is based on the SMS-EMOA framework, employs the R2-based hypervolume contribution approximation method in [25] in environmental selection. Our algorithm has the following two advantages: 1) It inherits the advantage of the hypervolume-based EMOAs, i.e., it can directly optimize the hypervolume of the population; 2) It has a worst-case time complexity of $O(N^2|\Lambda|)$ in one generation where N is the population size and $|\Lambda|$ is the number of the direction vectors. Since the worst-case time complexity does not increase exponentially with the number of objectives, it is suitable for solving MaOPs. The main contributions of this paper are listed as follows:

- Using the R2-based hypervolume contribution approximation method in [25], a new hypervolume-based EMOA, namely R2HCA-EMOA, is proposed for many-objective optimization. A normalization mechanism is incorporated into R2HCA-EMOA to enhance its performance.
- In order to make R2HCA-EMOA computationally efficient, a utility tensor structure is introduced for the calculation of the R2 indicator variant. By efficiently updating the utility tensor, we can avoid the repeated calculation and ensure the efficiency of the algorithm.
- R2HCA-EMOA is compared with three hypervolume-based EMOAs and several other state-of-the-art EMOAs on 5-, 10- and 15-objective DTLZ and WFG problems

and their minus versions. The results demonstrate the efficiency and effectiveness of R2HCA-EMOA for solving MaOPs.

The rest of the paper is organized as follows: In Section II, we briefly review three representative hypervolume-based EMOAs. The proposed algorithm R2HCA-EMOA is elaborated in Section III. Experiments are given in Section IV. We conclude the paper in Section V.

II. RELATED WORKS

A. SMS-EMOA

SMS-EMOA [11], [12] is a classical hypervolume-based EMOA. In SMS-EMOA, the population evolves in a steady-state manner. In each generation, a single offspring is generated and added to the population, then the non-dominated sorting is applied to the updated population, and the individual with the smallest hypervolume contribution among the last front individuals is removed from the population. In this manner, SMS-EMOA ensures that the hypervolume of the whole population monotonically increases as the number of generations increases.

SMS-EMOA already showed its promising performance on MOPs and MaOP with 4-6 objectives [27]. However, the main drawback of SMS-EMOA is that it can hardly be applied to MaOPs with many objectives (e.g., more than 10 objectives) because the exact hypervolume calculation is time-consuming in high-dimensional spaces. This drawback has been addressed in the following two ways: One is to apply a faster hypervolume calculation method, and the other way is to apply a hypervolume approximation method. These two research directions lead to the following two algorithms: FV-MOEA [16] and HypE [17].

B. FV-MOEA

FV-MOEA [16] is a recently proposed hypervolume-based EMOA. Differs from SMS-EMOA, FV-MOEA has the framework of a $(\mu + \mu')$ evolution strategy. In each generation, a set of offspring is generated and added to the population. Next, non-dominated sorting-based environmental selection is performed to identify the last front to be included in the next generation. Then individuals in the last front are selected based on a fast hypervolume update procedure. The general idea of this procedure is to delete the individual with the smallest hypervolume contribution one by one. The hypervolume contribution of each of the remaining individuals in this front is updated once an individual is removed.

As reported in [16], FV-MOEA is much faster than SMS-EMOA on up to 5-objective MaOPs. This is due to the following two reasons:

- 1) FV-MOEA adopts a $(\mu + \mu')$ evolution strategy while SMS-EMOA adopts a $(\mu + 1)$ evolution strategy. It has been shown that a $(\mu + 1)$ EMOA takes much longer runtime than its $(\mu + \mu')$ counterpart when they are compared under the same number of function evaluations [28]. For this reason, FV-MOEA is more efficient than SMS-EMOA.

2) FV-MOEA has a fast hypervolume contribution update procedure. Once a solution is removed from the population, the hypervolume contribution of each individual in the remaining population is efficiently updated instead of recalculated. In this manner, FV-MOEA can achieve a fast computational performance. However, there is no such an update procedure in SMS-EMOA.

FV-MOEA shows a good performance on two- and three-objective MOPs and up to five-objective MaOPs. However, because of the #P-hardness of the exact hypervolume calculation [18], FV-MOEA becomes time-consuming in much higher dimensional spaces (e.g., 10-dimensional space), which limits the applicability of FV-MOEA for solving MaOPs with 10 or more objectives.

To overcome the limitation of the exact hypervolume calculation, while keeping the advantage of the hypervolume-based EMOAs, hypervolume approximation can be applied instead of the exact hypervolume calculation.

C. HypE

HypE [17] is a hypervolume estimation-based EMOA with a $(\mu + \mu')$ evolution strategy. The framework of HypE is similar to that of FV-MOEA. After a set of offspring is generated and added to the population, the non-dominated sorting is performed to identify the last front to be included. Instead of using the exact hypervolume calculation for selecting individuals from the last front as in FV-MOEA, HypE uses a Monte Carlo sampling method to estimate the expected hypervolume loss (i.e., fitness value) attributed to each individual in the last front to be included. The individual with the smallest fitness value is removed one by one, and the fitness value of each of the remaining individuals is updated once an individual is removed.

HypE has been tested on up to 50-objective MaOPs and showed its effectiveness for solving MaOPs. However, HypE is still very time-consuming if a large number of sampling points are used in order to achieve a good hypervolume approximation.

III. R2HCA-EMOA

A. R2-based Hypervolume Contribution Approximation

In our previous work [25], we defined the following R2 indicator variant to approximate the hypervolume contribution of a solution s to a solution set A in an m -dimensional objective space¹:

$$R_2^{\text{HCA}}(s, A, \Lambda, \mathbf{r}, \alpha) = \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} \min \left\{ \min_{\mathbf{a} \in A \setminus \{s\}} \{g^{*2\text{ch}}(\mathbf{a}|\lambda, s)\}, g^{\text{mtch}}(\mathbf{r}|\lambda, s) \right\}^\alpha, \quad (1)$$

where A is a non-dominated solution set and $s \in A$, Λ is a direction vector set and each direction vector $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m) \in \Lambda$ satisfies $\|\lambda\|_2 = 1$ and $\lambda_i \geq 0$, $i = 1, \dots, m$, \mathbf{r} is the reference point, α is a parameter that is

¹In this paper, the solutions (individuals) are discussed in the objective space, i.e., a solution s denotes an objective vector.

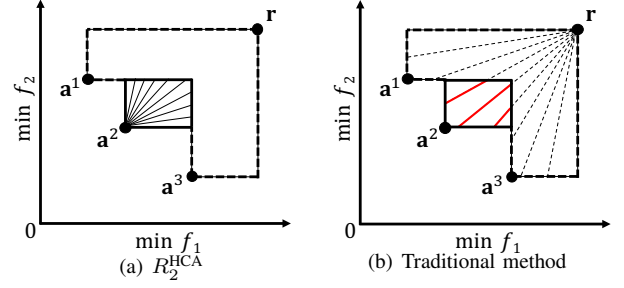


Fig. 1. An illustration of the geometric meaning of R_2^{HCA} and a traditional method for the hypervolume contribution approximation.

chosen to be $\alpha = m$ as suggested in [25]. For convenience, we use $R_2^{\text{HCA}}(s)$ to denote $R_2^{\text{HCA}}(s, A, \Lambda, \mathbf{r}, \alpha)$ in the remaining of this paper.

The $g^{*2\text{ch}}(\mathbf{a}|\lambda, s)$ function is defined for minimization problems as

$$g^{*2\text{ch}}(\mathbf{a}|\lambda, s) = \max_{j \in \{1, \dots, m\}} \left\{ \frac{a_j - s_j}{\lambda_j} \right\}, \quad (2)$$

For maximization problems, it is defined as

$$g^{*2\text{ch}}(\mathbf{a}|\lambda, s) = \max_{j \in \{1, \dots, m\}} \left\{ \frac{s_j - a_j}{\lambda_j} \right\}. \quad (3)$$

The g^{mtch} function is defined for both minimization and maximization problems as

$$g^{\text{mtch}}(\mathbf{r}|\lambda, s) = \min_{j \in \{1, \dots, m\}} \left\{ \frac{|s_j - r_j|}{\lambda_j} \right\}. \quad (4)$$

The R2 indicator variant in Eq. (1) has a clear geometric meaning to approximate the hypervolume contribution which is illustrated in Fig. 1 (a). Suppose we have three non-dominated solutions \mathbf{a}^1 , \mathbf{a}^2 and \mathbf{a}^3 , a set of direction vectors Λ , and the reference point \mathbf{r} . If we draw different line segments with different directions starting from \mathbf{a}^2 and ending on the boundary of the hypervolume contribution region of \mathbf{a}^2 as shown in Fig. 1 (a), then $R_2^{\text{HCA}}(\mathbf{a}^2)$ is the hypervolume contribution of \mathbf{a}^2 approximated by $\frac{1}{|\Lambda|} \sum_{i=1}^{|\Lambda|} l_i^m$ where l_i is the length of the line segment associated with the i th direction vector in Λ .

A traditional method for the hypervolume contribution approximation which is based on the achievement scalarizing function method in [22] is illustrated in Fig. 1 (b). The hypervolume contribution of \mathbf{a}^2 is approximated by the hypervolume approximation of solution set $\{\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3\}$ minus the hypervolume approximation of solution set $\{\mathbf{a}^1, \mathbf{a}^3\}$, where the hypervolume of a solution set is approximated by the line segments starting from the reference point \mathbf{r} and ending on the hypervolume attainment surface. As shown in Fig. 1 (b), the hypervolume contribution of \mathbf{a}^2 is approximated by the red lines in the hypervolume contribution region of \mathbf{a}^2 . Comparing with the method in Fig. 1 (a), the traditional method is less accurate if the same finite number (especially a small number) of direction vectors are used in both methods. This is because R_2^{HCA} can utilize all direction vectors whereas the traditional method can utilize only a part of them which go through the hypervolume contribution region. The approximation accuracy

of these two methods are compared in [25], and the results clearly show that R_2^{HCA} is much more accurate than the traditional method.

The R2 indicator variant defined in Eq. (1) is the foundation for the development of the new algorithm. For more detailed explanations of this R2 indicator variant, please refer to [25]. Based on this R2 indicator variant, we will introduce R2HCA-EMOA in the following subsections.

B. General Framework

Algorithm 1 R2HCA-EMOA

Input: Population Size N , Maximum Function Evaluations FEs^{max} .

Output: Final Population P .

- 1: Initialize Population P , Direction Vector Set Λ , Utility Tensor \mathbf{T} , reference point \mathbf{r} , $\text{FEs} = N$;
 - 2: **while** $\text{FEs} \leq \text{FEs}^{\text{max}}$ **do**
 - 3: $\mathbf{q} = \text{GenerateOffspring}(P)$;
 - 4: $P = P \cup \{\mathbf{q}\}$;
 - 5: $[P, \mathbf{T}] = \text{EnvironmentalSelection}(P, \Lambda, \mathbf{T}, \mathbf{r})$;
 - 6: $\text{FEs} = \text{FEs} + 1$;
 - 7: **end while**
-

The general framework of R2HCA-EMOA follows a steady-state $(\mu + 1)$ evolution strategy, which is similar to SMS-EMOA. Algorithm 1 gives the pseudocode of R2HCA-EMOA. In each generation, an offspring \mathbf{q} is generated from population P (Line 3). Then the population P is updated by incorporating offspring \mathbf{q} (Line 4). After that, an environmental selection procedure is performed to remove one individual from population P to keep the population size constant (Line 5). Lastly, the current function evaluations FEs is incremented by 1 since only one offspring is produced in each generation (Line 6).

In *GenerateOffspring* procedure, the offspring is generated as follows: two parents are randomly selected from population P , and the offspring is produced based on its two parents by applying simulated binary crossover (SBX) [29] and polynomial mutation [30] operators, which are commonly used in the EMO community. With respect to other procedures in R2HCA-EMOA, detailed descriptions are presented in the following subsections.

C. Reference Point Specification in R2HCA-EMOA

In fact, R2HCA-EMOA is a hypervolume-based EMOA where its objective is to maximize the hypervolume of the final population. As suggested in [31], the specification of the reference point for the hypervolume indicator should be carefully treated. The specification of the reference point has a large effect on the optimal distribution of the final population on the PF in hypervolume-based EMOAs (e.g., SMS-EMOA, FV-MOEA). If the reference point is too close to the PF, then the solutions cannot widely distribute on the PF. However, if the reference point is too far away from the PF, then the solutions tend to distribute on the boundaries of the PF when the shape of the PF is inverted triangular. In order to obtain widely and evenly distributed solutions on the PF, the reference

point $\mathbf{r} = (r, r, \dots, r)$ can be specified in the normalized objective space with the ideal point $(0, 0, \dots, 0)$ and the nadir point $(1, 1, \dots, 1)$ as follows as suggested in [31]:

$$r = 1 + \frac{1}{H}, \quad (5)$$

where H is an integer satisfying $C_{m-1}^{H+m-1} \leq N < C_{m-1}^{H+m}$, and C_m^n is the total number of combinations for choosing m elements from a set of n elements, i.e., $C_m^n = \frac{n!}{m!(n-m)!}$.

In each generation of R2HCA-EMOA, the current population P will be normalized to the objective space $(0, 0, \dots, 0)$ to $(1, 1, \dots, 1)$, and the reference point \mathbf{r} is specified according to Eq. (5).

D. Environmental Selection

Algorithm 2 EnvironmentalSelection($P, \Lambda, \mathbf{T}, \mathbf{r}$)

Input: $P, \Lambda, \mathbf{T}, \mathbf{r}$.

Output: Updated P and \mathbf{T} .

- 1: $P' = \text{Normalize}(P)$;
 - 2: $\mathbf{T} = \text{UpdateUtilityTensor}(\mathbf{T}, P', \Lambda)$;
 - 3: $\{F_1, F_2, \dots, F_l\} = \text{NondominatedSort}(P')$;
 - 4: **if** $|F_l| == 1$ **then**
 - 5: $R_2^{\text{HCA}}(\mathbf{s}) = 0, \forall \mathbf{s} \in F_l$;
 - 6: **else if** $|F_l| == N + 1$ **then**
 - 7: $R_2^{\text{HCA}}(\mathbf{s}) = \text{CalculateR2HCA}(\mathbf{T}), \forall \mathbf{s} \in F_l$;
 - 8: **else**
 - 9: $\mathbf{T}' = \text{ExtractUtilityTensor}(\mathbf{T}, F_l)$;
 - 10: $R_2^{\text{HCA}}(\mathbf{s}) = \text{CalculateR2HCA}(\mathbf{T}'), \forall \mathbf{s} \in F_l$;
 - 11: **end if**
 - 12: $\mathbf{s}_{\text{worst}} = \arg \min_{\mathbf{s} \in F_l} R_2^{\text{HCA}}(\mathbf{s})$;
 - 13: $P = P \setminus \{\mathbf{s}_{\text{worst}}\}$;
-

Algorithm 2 gives the pseudocode of the environmental selection procedure. Firstly, the population P is normalized to P' (Line 1), then the utility tensor \mathbf{T} is updated based on P' (Line 2). After that, the non-dominated sorting is performed to divide population P' into different front levels (Line 3). Lastly, one individual with the smallest R_2^{HCA} value from the last front F_l is removed from P (Line 4-13). If there is only one individual in F_l , then this individual can be directly removed without calculating its R_2^{HCA} value (Line 4-5). If there is more than one individual in F_l , then the R_2^{HCA} values of all individuals in F_l need to be calculated first (Line 6-10).

In Algorithm 2, a utility tensor \mathbf{T} is used to assist the calculation of the R_2^{HCA} values. In the next subsection, we will explain tensor \mathbf{T} and its operations in detail.

E. Utility Tensor \mathbf{T}

1) *Definition of Tensor \mathbf{T}* : In R2HCA-EMOA, the most time-consuming part is the calculation of the R_2^{HCA} values. In order to maximize the computational efficiency of this part, a utility tensor \mathbf{T} is introduced to assist the calculation of R_2^{HCA} . Suppose the current population with the offspring included is $P = \{\mathbf{s}^1, \mathbf{s}^2, \dots, \mathbf{s}^{N+1}\}$. \mathbf{T} is a 3-order tensor with the following elements

$$T_{ijk} = \begin{cases} g^{*2\text{th}}(\mathbf{a}^i | \lambda^j, \mathbf{s}^k), & i = 1, \dots, N \\ g^{\text{mth}}(\mathbf{r} | \lambda^j, \mathbf{s}^k), & i = N + 1 \end{cases}, \quad (6)$$

where $\mathbf{s}^k \in P, k = 1, \dots, N+1$, $\mathbf{a}^i \in P \setminus \{\mathbf{s}^k\}, i = 1, \dots, N$, $\lambda^j \in \Lambda, j = 1, \dots, |\Lambda|$. So the size of \mathbf{T} is $(N+1) \times |\Lambda| \times (N+1)$.

Once the tensor \mathbf{T} is obtained, its information can be used to calculate the R_2^{HCA} values. Suppose P is a non-dominated solution set, then for a solution $\mathbf{s}^k \in P$, its R_2^{HCA} value can be calculated as follows:

$$R_2^{\text{HCA}}(\mathbf{s}^k) = \frac{1}{|\Lambda|} \sum_{j=1}^{|\Lambda|} (\min_{i=1}^{N+1} T_{ijk})^m. \quad (7)$$

Notice that P should be a non-dominated solution set. If P contains dominated solutions, then we cannot directly use \mathbf{T} to calculate R_2^{HCA} values by Eq. (7). We will discuss how to deal with this situation later in this section.

Next we give a toy example to illustrate the tensor \mathbf{T} defined above, and how it can be used to calculate R_2^{HCA} .

Example 1. Suppose we have a solution set $A = \{\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3\}$, a direction vector set $\Lambda = \{\lambda^1, \lambda^2, \lambda^3\}$, and the reference point \mathbf{r} . Then the tensor \mathbf{T} is depicted in Fig. 2. Suppose A is a non-dominated solution set, then the R_2^{HCA} values of the solutions in A are calculated as follows:

$$\begin{cases} R_2^{\text{HCA}}(\mathbf{a}^1) = \frac{1}{3} \sum_{j=1}^3 (\min_{i=1}^3 T_{ij1})^m \\ R_2^{\text{HCA}}(\mathbf{a}^2) = \frac{1}{3} \sum_{j=1}^3 (\min_{i=1}^3 T_{ij2})^m \\ R_2^{\text{HCA}}(\mathbf{a}^3) = \frac{1}{3} \sum_{j=1}^3 (\min_{i=1}^3 T_{ij3})^m \end{cases} \quad (8)$$

T_{ij1}	$g^{*2tch}(\mathbf{a}^2 \lambda^1, \mathbf{a}^1)$	$g^{*2tch}(\mathbf{a}^2 \lambda^2, \mathbf{a}^1)$	$g^{*2tch}(\mathbf{a}^2 \lambda^3, \mathbf{a}^1)$
	$g^{*2tch}(\mathbf{a}^3 \lambda^1, \mathbf{a}^1)$	$g^{*2tch}(\mathbf{a}^3 \lambda^2, \mathbf{a}^1)$	$g^{*2tch}(\mathbf{a}^3 \lambda^3, \mathbf{a}^1)$
	$g^{mtch}(\mathbf{r} \lambda^1, \mathbf{a}^1)$	$g^{mtch}(\mathbf{r} \lambda^2, \mathbf{a}^1)$	$g^{mtch}(\mathbf{r} \lambda^3, \mathbf{a}^1)$
T_{ij2}	$g^{*2tch}(\mathbf{a}^1 \lambda^1, \mathbf{a}^2)$	$g^{*2tch}(\mathbf{a}^1 \lambda^2, \mathbf{a}^2)$	$g^{*2tch}(\mathbf{a}^1 \lambda^3, \mathbf{a}^2)$
	$g^{*2tch}(\mathbf{a}^3 \lambda^1, \mathbf{a}^2)$	$g^{*2tch}(\mathbf{a}^3 \lambda^2, \mathbf{a}^2)$	$g^{*2tch}(\mathbf{a}^3 \lambda^3, \mathbf{a}^2)$
	$g^{mtch}(\mathbf{r} \lambda^1, \mathbf{a}^2)$	$g^{mtch}(\mathbf{r} \lambda^2, \mathbf{a}^2)$	$g^{mtch}(\mathbf{r} \lambda^3, \mathbf{a}^2)$
T_{ij3}	$g^{*2tch}(\mathbf{a}^1 \lambda^1, \mathbf{a}^3)$	$g^{*2tch}(\mathbf{a}^1 \lambda^2, \mathbf{a}^3)$	$g^{*2tch}(\mathbf{a}^1 \lambda^3, \mathbf{a}^3)$
	$g^{*2tch}(\mathbf{a}^2 \lambda^1, \mathbf{a}^3)$	$g^{*2tch}(\mathbf{a}^2 \lambda^2, \mathbf{a}^3)$	$g^{*2tch}(\mathbf{a}^2 \lambda^3, \mathbf{a}^3)$
	$g^{mtch}(\mathbf{r} \lambda^1, \mathbf{a}^3)$	$g^{mtch}(\mathbf{r} \lambda^2, \mathbf{a}^3)$	$g^{mtch}(\mathbf{r} \lambda^3, \mathbf{a}^3)$

Fig. 2. An illustration of tensor \mathbf{T} in Example 1.

2) *Update of Tensor \mathbf{T}* : In R2HCA-EMOA, only one offspring is generated in each generation. Thus we do not have to recalculate all elements of the tensor \mathbf{T} in each generation. We update only a small number of elements in \mathbf{T} for computational efficiency. Once an individual is removed from the population, a new offspring is added. This generation update mechanism can be viewed as the replacement of the removed individual with the new offspring. Thus, all elements in \mathbf{T} related to the removed individual are updated with the new offspring (Line 2 in Algorithm 2).

To be more specific, suppose $\mathbf{s}^{k'} \in P$ is replaced by a new solution \mathbf{q} , then $P = \{\mathbf{s}^1, \dots, \mathbf{s}^{k'-1}, \mathbf{q}, \mathbf{s}^{k'+1}, \dots, \mathbf{s}^{N+1}\}$. \mathbf{T} will be updated as follows:

For all $k = 1, \dots, N+1$ and $j = 1, \dots, |\Lambda|$,

$$\begin{aligned} \text{If } k = k', \text{ then } T_{ijk} &= \begin{cases} g^{*2tch}(\mathbf{a}^i|\lambda^j, \mathbf{q}), & i = 1, \dots, N \\ g^{mtch}(\mathbf{r}|\lambda^j, \mathbf{q}), & i = N+1 \end{cases} \\ \text{If } k < k', \text{ then } T_{ijk} &= g^{*2tch}(\mathbf{q}|\lambda^j, \mathbf{s}^k), \quad i = k' - 1, \\ \text{If } k > k', \text{ then } T_{ijk} &= g^{*2tch}(\mathbf{q}|\lambda^j, \mathbf{s}^k), \quad i = k', \end{aligned} \quad (9)$$

where $\mathbf{a}^i \in P \setminus \{\mathbf{q}\}, i = 1, \dots, N$.

From the above equation we can see that totally $(N+1)|\Lambda| + N|\Lambda|$ elements of tensor \mathbf{T} need to be updated. We illustrate this update procedure by the following toy example.

Example 2. Let us consider the case in Example 1. Suppose \mathbf{a}^2 is removed and \mathbf{q} is added, then the tensor \mathbf{T} is updated as depicted in Fig. 3 where the updated elements are shaded.

T_{ij1}	$g^{*2tch}(\mathbf{q} \lambda^1, \mathbf{a}^1)$	$g^{*2tch}(\mathbf{q} \lambda^2, \mathbf{a}^1)$	$g^{*2tch}(\mathbf{q} \lambda^3, \mathbf{a}^1)$
	$g^{*2tch}(\mathbf{a}^3 \lambda^1, \mathbf{a}^1)$	$g^{*2tch}(\mathbf{a}^3 \lambda^2, \mathbf{a}^1)$	$g^{*2tch}(\mathbf{a}^3 \lambda^3, \mathbf{a}^1)$
	$g^{mtch}(\mathbf{r} \lambda^1, \mathbf{a}^1)$	$g^{mtch}(\mathbf{r} \lambda^2, \mathbf{a}^1)$	$g^{mtch}(\mathbf{r} \lambda^3, \mathbf{a}^1)$
T_{ij2}	$g^{*2tch}(\mathbf{a}^1 \lambda^1, \mathbf{q})$	$g^{*2tch}(\mathbf{a}^1 \lambda^2, \mathbf{q})$	$g^{*2tch}(\mathbf{a}^1 \lambda^3, \mathbf{q})$
	$g^{*2tch}(\mathbf{a}^3 \lambda^1, \mathbf{q})$	$g^{*2tch}(\mathbf{a}^3 \lambda^2, \mathbf{q})$	$g^{*2tch}(\mathbf{a}^3 \lambda^3, \mathbf{q})$
	$g^{mtch}(\mathbf{r} \lambda^1, \mathbf{q})$	$g^{mtch}(\mathbf{r} \lambda^2, \mathbf{q})$	$g^{mtch}(\mathbf{r} \lambda^3, \mathbf{q})$
T_{ij3}	$g^{*2tch}(\mathbf{a}^1 \lambda^1, \mathbf{a}^3)$	$g^{*2tch}(\mathbf{a}^1 \lambda^2, \mathbf{a}^3)$	$g^{*2tch}(\mathbf{a}^1 \lambda^3, \mathbf{a}^3)$
	$g^{*2tch}(\mathbf{q} \lambda^1, \mathbf{a}^3)$	$g^{*2tch}(\mathbf{q} \lambda^2, \mathbf{a}^3)$	$g^{*2tch}(\mathbf{q} \lambda^3, \mathbf{a}^3)$
	$g^{mtch}(\mathbf{r} \lambda^1, \mathbf{a}^3)$	$g^{mtch}(\mathbf{r} \lambda^2, \mathbf{a}^3)$	$g^{mtch}(\mathbf{r} \lambda^3, \mathbf{a}^3)$

Fig. 3. An illustration of tensor update in Example 2.

3) *Extraction of Tensor \mathbf{T}* : In R2HCA-EMOA, one individual in the last front F_l is removed. If F_l is the same as the whole population (i.e., $l = 1$, which means all individuals in the population are non-dominated with each other), then the tensor \mathbf{T} can be directly used to calculate the R_2^{HCA} values of all individuals (Line 7 in Algorithm 2). However, F_l usually contains only a part of the population (i.e., $l > 1$). In this case, a subtensor \mathbf{T}' is extracted from \mathbf{T} (Line 9 in Algorithm 2) in order to calculate the R_2^{HCA} values of the individuals in F_l . The extraction is straightforward, i.e., elements in \mathbf{T} that are related only to individuals in F_l (i.e., those that are not related to any individual in the other fronts) are extracted to form the subtensor \mathbf{T}' .

To be more specific, suppose $P = \{\mathbf{s}^1, \mathbf{s}^2, \dots, \mathbf{s}^{N+1}\}$, $F_l = \{\mathbf{s}^{I_1}, \mathbf{s}^{I_2}, \dots, \mathbf{s}^{I_M}\}$ and $I = \{I_1, I_2, \dots, I_M\}$, where $1 \leq I_1 < I_2 < \dots < I_M \leq N+1$ and $|F_l| = M$ with $1 < M < N+1$, then the subtensor \mathbf{T}' is extracted from \mathbf{T} as follows:

$$\begin{aligned} \text{For all } k = 1, \dots, M, i = 1, \dots, M, \text{ and } j = 1, \dots, |\Lambda|, \\ T'_{ijk} = T_{i'j'k'}, \end{aligned} \quad (10)$$

where $k' = I_k$, $j' = j$, and

$$\begin{aligned} \text{If } k = 1, \text{ then } i' &= \begin{cases} I'_i - 1, & \text{if } i < M \\ N + 1, & \text{if } i = M \end{cases} \\ \text{If } k = M, \text{ then } i' &= \begin{cases} I'_i, & \text{if } i < M \\ N + 1, & \text{if } i = M \end{cases} \\ \text{If } 1 < k < M, \text{ then } i' &= \begin{cases} I'_i, & \text{if } i < k \\ I'_i - 1, & \text{if } k \leq i < M \\ N + 1, & \text{if } i = M \end{cases} \end{aligned} \quad (11)$$

where $I' = I \setminus \{I_k\}$, and I'_i denotes the i th element in I' .

From the above equation we can see that the size of \mathbf{T}' is $M \times |\Lambda| \times M$, which means that totally $M^2|\Lambda|$ elements need to be extracted from \mathbf{T} .

Once the subtensor \mathbf{T}' is extracted from \mathbf{T} , we can use it to calculate the R_2^{HCA} values of the individuals in F_l as follows:

$$R_2^{\text{HCA}}(\mathbf{s}^{I_k}) = \frac{1}{|\Lambda|} \sum_{j=1}^{|\Lambda|} (\min_{i=1}^M T'_{ijk})^m. \quad (12)$$

We will use the following toy example to illustrate this procedure.

Example 3. Let us consider the case in Example 1 again. Suppose $F_l = \{\mathbf{a}^1, \mathbf{a}^3\}$, i.e., \mathbf{a}^2 dominates \mathbf{a}^1 and \mathbf{a}^3 while \mathbf{a}^1 and \mathbf{a}^3 are non-dominated with each other. Thus all elements related to \mathbf{a}^1 and \mathbf{a}^3 and not related to \mathbf{a}^2 are extracted from \mathbf{T} to form \mathbf{T}' as shown in Fig. 4, where the extracted elements in \mathbf{T} are shaded. Then the R_2^{HCA} values of the solutions in F_l are calculated as follows:

$$\begin{cases} R_2^{\text{HCA}}(\mathbf{a}^1) = \frac{1}{3} \sum_{j=1}^3 (\min_{i=1}^2 T'_{ij1})^m \\ R_2^{\text{HCA}}(\mathbf{a}^3) = \frac{1}{3} \sum_{j=1}^3 (\min_{i=1}^2 T'_{ij2})^m \end{cases}. \quad (13)$$

F. Normalization Mechanism

In R2HCA-EMOA, the normalization mechanism is incorporated before the update of the tensor \mathbf{T} (Line 1 in Algorithm 2). The normalization works as follows: First, we estimate the ideal point \mathbf{z}^* and the worst point \mathbf{z}^{wst} from the current population P as $z_i^* = \min_{\mathbf{s} \in P} s_i$ and $z_i^{\text{wst}} = \max_{\mathbf{s} \in P} s_i$, $i = 1, \dots, m$. Then each solution $\mathbf{s} \in P$ is normalized as $\mathbf{s}'_i = (s_i - z_i^*) / (z_i^{\text{wst}} - z_i^*)$, $i = 1, \dots, m$ where $\mathbf{s}' \in P'$ is the normalized solution. After the normalization, the population P' is within the objective space $(0, 0, \dots, 0)$ to $(1, 1, \dots, 1)$.

The normalization mechanism is used to tackle the MOPs and MaOPs with a totally different scale in each objective. For such a problem, the hypervolume contribution approximation by R_2^{HCA} can be inaccurate if no normalization is performed. The normalization mechanism remedies this potential difficulty. However, the normalization will bring an issue for the update of the tensor \mathbf{T} . In the last subsection, we explained that only the elements in the tensor \mathbf{T} related to the removed individual are updated with those to the newly added offspring in order to improve the computational efficiency. If the normalization is performed, the locations of all solutions in the objective space can be changed. If this happens, we need to update all elements instead of updating only a small number of

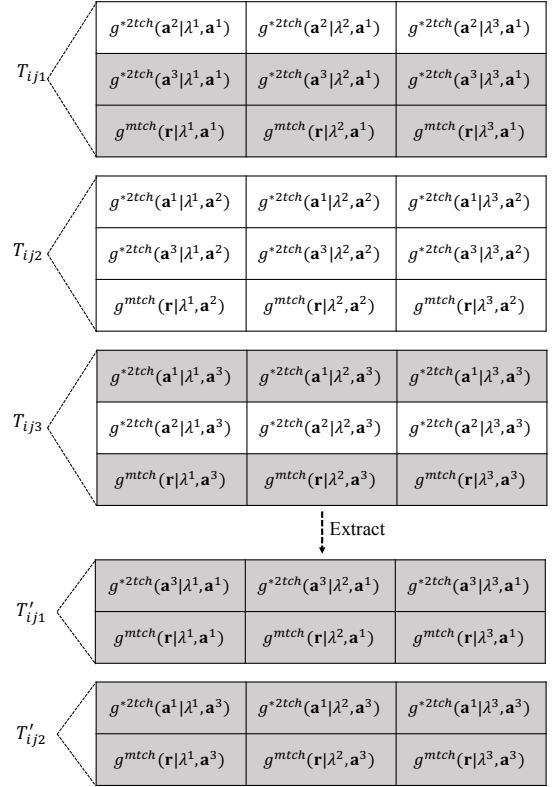


Fig. 4. An illustration of tensor extraction in Example 3.

elements in the tensor \mathbf{T} . Actually, this can happen frequently in the early generations of the optimization process. Thus the tensor \mathbf{T} needs to be recalculated frequently which leads to low computational efficiency.

In this paper, in order to incorporate the normalization mechanism in R2HCA-EMOA while keeping the algorithm computationally efficient, we simply ignore the normalization in the update phase of the tensor \mathbf{T} . That is, we only replace the elements in \mathbf{T} related to the removed individual with those to the new offspring (even if other solutions are changed in the objective space after the normalization). In this manner, the computational efficiency of the proposed R2HCA-EMOA is not degraded. This simple approach is based on the following considerations. In early generations, accurate approximation of hypervolume contributions is not important since the search is mainly driven by non-dominated sorting and/or the differences of the hypervolume contributions among the individuals are large. In late generations, the update of the estimated ideal and worst points does not happen frequently. In Section IV-D1, we will show the effectiveness of the normalization mechanism together with the simple update strategy in R2HCA-EMOA.

G. Computational Complexity

The computational complexity of one generation of R2HCA-EMOA is analyzed as follows. The main computational cost is from the environmental selection procedure (line 5 in Algorithm 1). The normalization of population P (line 1 in Algorithm 2) requires $O(mN)$ computations. In the tensor update procedure, totally $(N + 1)|\Lambda| + N|\Lambda|$ elements

of tensor \mathbf{T} need to be updated. Therefore, update of tensor \mathbf{T} (line 2 in Algorithm 2) requires $m(2N+1)|\Lambda| = O(mN|\Lambda|)$ computations. The non-dominated sorting procedure used here is the T-ENS algorithm [32] with a time complexity of $O(mN \log N / \log m)$. The calculation of the R_2^{HCA} values using tensor \mathbf{T} (line 7 in Algorithm 2) requires $O(N^2|\Lambda|)$ computations. The extraction of tensor \mathbf{T} requires $O(N^2|\Lambda|)$ operations in the worst case. Thus, the calculation of the R_2^{HCA} values using the extracted subtensor \mathbf{T}' (line 10 in Algorithm 2) requires $O(N^2|\Lambda|)$ computations in the worst case. Finally, finding the worst solution (line 12 in Algorithm 2) requires $O(N)$ comparisons in the worst case. In summary, because usually $m < N$, the overall worst-case time complexity of one generation of R2HCA-EMOA is $O(N^2|\Lambda|)$.

If the tensor \mathbf{T} is not used in R2HCA-EMOA, that is, the R_2^{HCA} values of the individuals in F_t are recalculated in each generation, it requires $O(mN^2|\Lambda|)$ computations in the worst case. So the overall worst-case time complexity of R2HCA-EMOA will be $O(mN^2|\Lambda|)$. By introducing the tensor \mathbf{T} and its operations, we can improve the time complexity of R2HCA-EMOA by a factor of m in each generation.

IV. EXPERIMENTAL STUDIES

A. Experiment Settings

1) *Platforms*: The software platform we use in our experiments is PlatEMO [33], which is a MATLAB-based open source platform for evolutionary multi-objective optimization. The R2HCA-EMOA is implemented in PlatEMO. All the other EMOAs used in our experiments are based on their implementations in PlatEMO². The hardware platform is a PC equipped with Intel Core i7-8700K CPU@3.70GHz, 16GB RAM.

2) *Test Problems*: We choose DTLZ1-4 [34], WFG1-9 [35], and their minus version MinusDTLZ1-4, MinusWFG1-9 [7] for testing. The number of objectives is set to $m = 5, 10, 15$. The number of decision variables is set to $m + 4$ for DTLZ1 and MinusDTLZ1, and $m + 9$ for other DTLZ and MinusDTLZ problems. The distance- and position-related decision variables are set to 24 and $m - 1$, respectively, for WFG and MinusWFG problems.

3) *Parameter Settings*: The population size N is set to 100 for all test problems. For DTLZ1, DTLZ3, WFG1 and their minus versions, the maximum function evaluations FEs^{\max} is set to 100,000, while FEs^{\max} is set to 30,000 for the other test problems. These values of FEs^{\max} are used as the termination condition. Simulated binary crossover (SBX) and polynomial mutation are used where the distribution index is specified as 20 in both operators. The crossover and mutation rates are set to 1.0 and $1/n$, respectively, where n is the number of decision variables. Each algorithm is applied to each test problem 20 times (i.e., 20 independent runs).

4) *Performance Metrics*: The hypervolume indicator is employed to evaluate the performance of the compared EMOAs. For 5- and 10-objective problems, the WFG algorithm [36] is employed to calculate the exact hypervolume of the obtained

solution set in each run of each algorithm on each test problem, whereas for 15-objective problems, the Monte Carlo method is employed to estimate the hypervolume. We calculate the hypervolume as follows. First, using the true ideal point \mathbf{z}^* and the true nadir point \mathbf{z}^{nad} obtained from the true PF, the objective values of the obtained solutions are normalized so that two points \mathbf{z}^* and \mathbf{z}^{nad} are $(0, \dots, 0)$ and $(1, \dots, 1)$, respectively. Then the reference point \mathbf{r} is set to $(1.1, \dots, 1.1)$ to calculate the hypervolume. Furthermore, the results are analyzed by the Wilcoxon rank sum test with a significance level of 0.05 to determine whether one algorithm shows a statistically significant difference with the other, where '+', '-', and ' \approx ' indicate that the compared EMOA is 'significantly better than', 'significantly worse than' and 'statistically similar to' R2HCA-EMOA, respectively.

5) *Compared Algorithms*: First, three hypervolume-based EMOAs (i.e., SMS-EMOA, FV-MOEA and HypE) are compared with R2HCA-EMOA. Both SMS-EMOA and FV-MOEA employ the WFG algorithm for the hypervolume calculation inside their implementations. In HypE, the number of sampling points is set to 10,000, which is the same as in [17]. The reference point is specified as $1 + 1/H$ in the normalized objective space in the three hypervolume-based EMOAs, which follows the same manner as in R2HCA-EMOA.

We also compare R2HCA-EMOA with several other state-of-the-art EMOAs. They are briefly introduced as follows:

- AR-MOEA [37] is an indicator-based EMOA, which is based on an enhanced inverted generational distance indicator (IGD-NS). The reference points are adaptively adjusted based on the IGD-NS contributions of the candidate solutions in an external archive, thus taking both uniform distribution and Pareto front approximation into consideration.
- SPEA2+SDE [38] is a combination of SPEA2 and the shift-based density estimation (SDE) strategy. SDE modifies the density estimation strategy in traditional Pareto-based EMOAs and is able to cover both the distribution and convergence information of individuals. Thus, it can make Pareto-based algorithms suitable for MaOPs.
- GFM-MOEA [39] is an EMOA based on PF modeling. The shape of the PF is estimated by training a generalized simplex model, then the algorithm is driven by this approximated PF during the optimization process. The penalty parameter θ and the frequency f_r in GFM-MOEA are set to 0.2 and 0.1 respectively, which are the same as in [39].
- BCE-IBEA [40] is an EMOA with a bi-criterion evolution (BCE) framework. BCE-IBEA integrates IBEA [10] into the BCE framework, where the Pareto criterion evolution and the non-Pareto criterion evolution (i.e., IBEA) work collaboratively to facilitate each other's evolution. IBEA is based on $I_{\epsilon+}$ indicator [41] and the scaling factor κ in IBEA is set to 0.05.

The parameters involved in R2HCA-EMOA are set as follows: The number of the direction vectors $|\Lambda|$ is set to 100. The direction vectors used in R_2^{HCA} are generated by sampling

²FV-MOEA is implemented by ourselves according to [16] since it is not implemented in PlatEMO.

points uniformly on the unit hypersphere [42]³.

For all the EMOAs involved in our experiments, SMS-EMOA and R2HCA-EMOA follow the $(\mu + 1)$ evolution strategy, while the other EMOAs follow the $(\mu + \mu')$ evolution strategy where $\mu = \mu'$ (i.e., N offspring are generated from N parents in each generation).

B. Comparison with the hypervolume-based EMOAs

In this subsection, we compare R2HCA-EMOA with the three representative hypervolume-based EMOAs: SMS-EMOA, FV-MOEA and HypE.

First we compare the runtime among the four algorithms to evaluate their computational efficiency. As SMS-EMOA and FV-MOEA become very time-consuming for solving 10- and 15-objective problems, we only test the four algorithms on DTLZ2 with 5, 10 and 15 objectives by a single run, and set FES_{\max} to 3000 (i.e., 1/10 of the original FES_{\max} setting for DTLZ2).

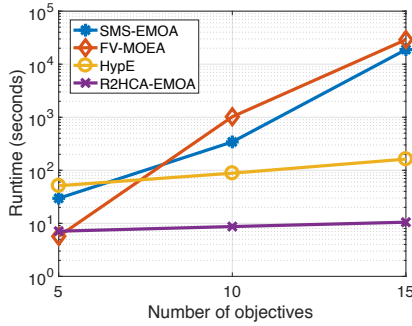


Fig. 5. Runtime comparison among SMS-EMOA, FV-MOEA, HypE and R2HCA-EMOA in a single run on DTLZ2 with 5, 10 and 15 objectives and 3000 FES_{\max} (in seconds). Notice that the number of the direction vectors (i.e., $|\Lambda|$) is 100 in our algorithm.

Figure 5 shows the runtime comparison of the four algorithms in a single run on DTLZ2 with 5, 10 and 15 objectives. From Figure 5 we can observe that, for the 5-objective DTLZ2, FV-MOEA is the fastest algorithm. The runtime of R2HCA-EMOA is comparable to that of FV-MOEA. SMS-EMOA and HypE consume much longer time than the other two algorithms. However, for the 10- and 15-objective DTLZ2, the runtime of SMS-EMOA and FV-MOEA increase exponentially while the runtime of HypE and R2HCA-EMOA increase linearly. It should be noted that SMS-EMOA and FV-MOEA take several hours for the 15-objective DTLZ2 even when we decrease the termination condition from 30,000 function evaluations to 3,000. R2HCA-EMOA is the fastest algorithm for the 10- and 15-objective DTLZ2. Because 10,000 sampling points are used in HypE while only 100 direction vectors are used in R2HCA-EMOA, HypE is much slower than R2HCA-EMOA in all the cases.

From Figure 5 we can also observe that FV-MOEA is faster than SMS-EMOA for the 5-objective DTLZ2, while it is slower than SMS-EMOA for the 10- and 15-objective

DTLZ2. The main reason is that with the offspring included in each generation, the population size of SMS-EMOA and FV-MOEA is $N + 1$ and $2N$ respectively. The almost 2 times larger population size of FV-MOEA will lead to much longer computational time of the hypervolume in 10- and 15-objective space. The observation suggests that FV-MOEA is more efficient than SMS-EMOA for 5-objective problems which is consistent with the results in [16]. However, FV-MOEA is less efficient than SMS-EMOA for 10- and 15-objective problems due to the exponentially increasing time in the hypervolume calculation.

On account of the very long runtime of SMS-EMOA and FV-MOEA on 10- and 15-objective problems, we will only compare the four algorithms on 5-objective problems. The results are shown in Table I. In Table I, we only show the Wilcoxon rank sum test results. Detailed numerical results are provided in Appendix A in the supplementary material.

TABLE I
WILCOXON RANK SUM TEST RESULTS BASED ON HYPERVOLUME VALUES OBTAINED BY SMS-EMOA, FV-MOEA, HYPE AND R2HCA-EMOA ON 5-OBJECTIVE DTLZ & WFG, MINUSDTLZ & MINUSWFG PROBLEMS.

+ / - / \approx	SMS-EMOA	FV-MOEA	HypE
DTLZ & WFG	12/1/0	10/2/1	0/9/4
MinusDTLZ & MinusWFG	13/0/0	11/2/0	0/13/0

The results show that SMS-EMOA and FV-MOEA outperform HypE and R2HCA-EMOA, which is consistent with our intuition. Since SMS-EMOA and FV-MOEA are based on exact hypervolume calculation, it is likely that they can achieve better hypervolume results than HypE and R2HCA-EMOA with hypervolume approximation. However, SMS-EMOA and FV-MOEA become very time-consuming for solving 10- and 15-objective problems, which limited their applicability for solving MaOPs with more than 10 objectives.

We can also observe that R2HCA-EMOA clearly outperforms HypE (+/-/ \approx is 0/9/4 for DTLZ & WFG and 0/13/0 for MinusDTLZ & MinusWFG as shown in Table I). HypE employs the Monte Carlo sampling method while R2HCA-EMOA employs the R2-based hypervolume contribution approximation method. Notice that 10,000 sampling points are used in HypE while only 100 direction vectors are used in R2HCA-EMOA. Comparing with HypE, R2HCA-EMOA consumes much less runtime and is able to achieve better hypervolume results, which shows that R2HCA-EMOA is an efficient and effective hypervolume-based EMOA for solving MaOPs.

In addition, the hypervolume variations over function evaluations obtained by the four hypervolume-based EMOAs on all 5-objective problems are shown in Figure 6. Instead of showing the hypervolume values directly, we show the hypervolume difference from the maximum value $maxVal$, i.e., each hypervolume value is subtracted from $maxVal$. For each test problem, the maximum value $maxVal$ is determined as the maximum mean plus standard deviation of all the hypervolume values obtained by all the algorithms over all the function evaluations. In each figure, the solid line represents the mean

³First we randomly sample points \mathbf{x} according to the normal distribution $\mathcal{N}_m(0, I_m)$, then the corresponding direction vectors are obtained by $\lambda = |\mathbf{x}| / \|\mathbf{x}\|_2$.

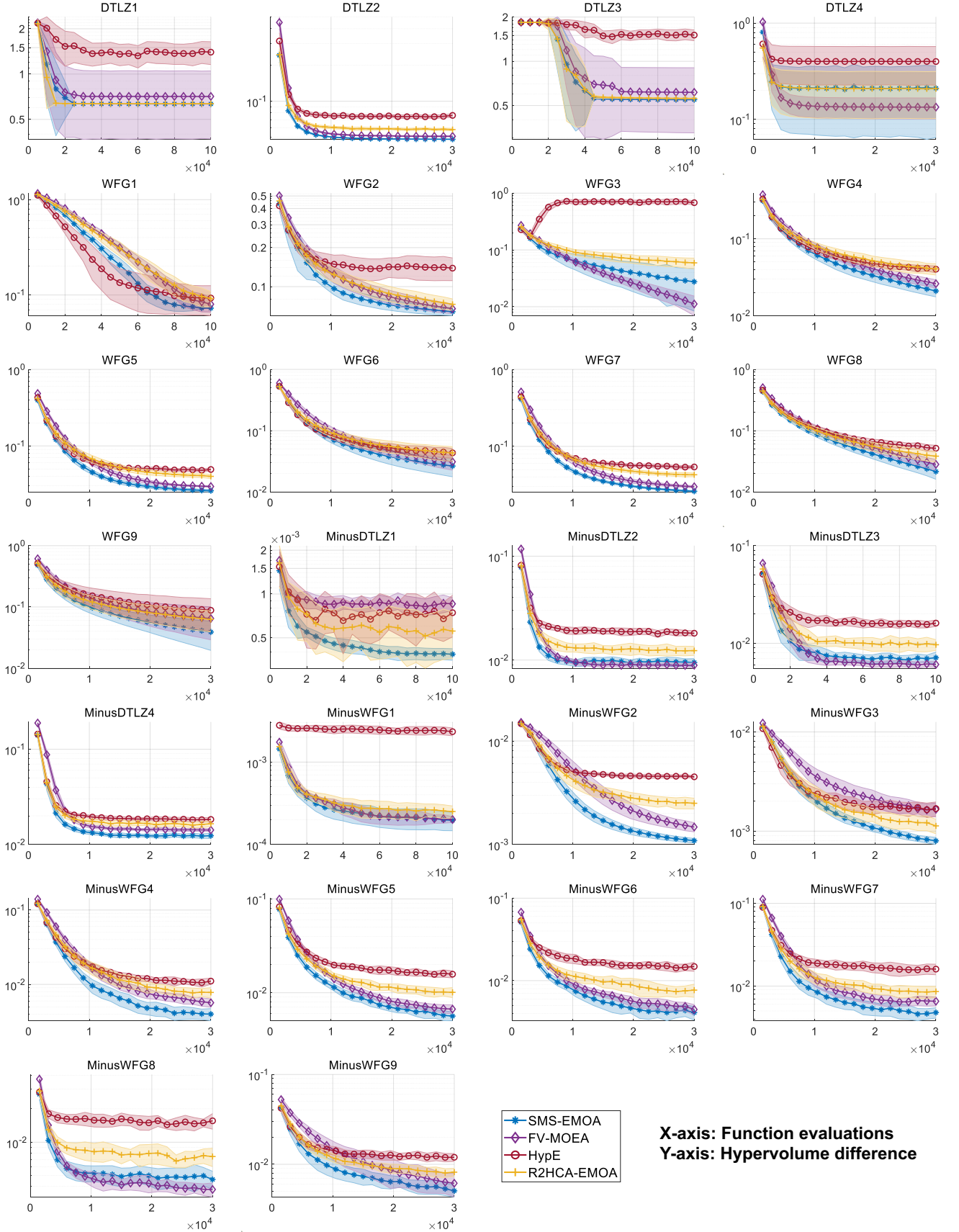


Fig. 6. Hypervolume variations over function evaluations obtained by SMS-EMOA, FV-MOEA, HypE and R2HCA-EMOA on the 5-objective problems. The solid line represents the mean values while the transparent area around the line represents the standard deviations.

TABLE II

WILCOXON RANK SUM TEST RESULTS BASED ON HYPERVOLUME VALUES OBTAINED BY AR-MOEA, SPEA2+SDE, GFM-MOEA, BCE-IBEA AND R2HCA-EMOA ON 5-, 10- AND 15-OBJECTIVE DTLZ & WFG, MINUSDTLZ & MINUSWFG PROBLEMS.

+ / - / \approx	AR-MOEA	SPEA2+SDE	GFM-MOEA	BCE-IBEA
DTLZ & WFG	2/36/1	1/34/4	1/29/9	3/28/8
MinusDTLZ & MinusWFG	2/35/2	3/33/3	3/35/1	5/33/1

values while the transparent area around the line represents the standard deviations. The Y-axis is shown in log scale for better discrimination of different lines.

From the figures we can observe that in general, the mean hypervolume difference decreases with the increase of FEs for all the four algorithms (except for HypE on WFG3 where the hypervolume difference increases with the increase of FEs). For most of the problems, R2HCA-EMOA has a competitive performance compared with SMS-EMOA and FV-MOEA, and clearly outperforms HypE. On some problems (e.g., DTLZ1, DTLZ3, MinusDTLZ1, MinusWFG3), R2HCA-EMOA performs even better than FV-MOEA. Meanwhile, R2HCA-EMOA has a relatively small standard deviation on most of the problems (except for MinusDTLZ1), which shows that R2HCA-EMOA has a stable performance in general. Figure 6 provides us a deeper insight into the behavior of R2HCA-EMOA, which clearly shows the ability of R2HCA-EMOA for achieving a good hypervolume performance.

C. Comparison with other state-of-the-art EMOAs

In this subsection, we compare R2HCA-EMOA with the four state-of-the-art EMOAs: AR-MOEA, SPEA2+SDE, GFM-MOEA and BCE-IBEA.

The results are shown in Table II where only the Wilcoxon rank sum test results are shown and detailed numerical results are provided in Appendix B in the supplementary material. From Table II we can observe that R2HCA-EMOA clearly outperforms all the other algorithms (+/-/ \approx is 2/36/1, 1/34/4, 1/29/9, 3/28/8 on DTLZ & WFG, and 2/35/2, 3/33/3, 3/35/1, 5/33/1 on MinusDTLZ & MinusWFG as shown in Table II). This is because only R2HCA-EMOA is a hypervolume-based EMOA whereas the other algorithms are not. We need to mention that in terms of the hypervolume metric, AR-MOEA outperforms MOEA/DD [43], NSGA-III [44], RVEA [45] and MOMBI-II [46] as reported in [37], and GFM-MOEA outperforms MOEA/DD, RVEA, MOEA/D-PaS [47] and VaEA [48] as reported in [39]. However, these two algorithms are both defeated by R2HCA-EMOA. Our results clearly demonstrate the effectiveness of R2HCA-EMOA for solving MaOPs in terms of the hypervolume metric.

We also show the hypervolume variations over function evaluations obtained by the five EMOAs on all the 10-objective problems in Figure 7 (The figures related to 5- and 15-objective problems are provided in Appendix B in the supplementary material). Same as in Figure 6, the hypervolume difference is shown in Figure 7. For each figure, the Y-axis is shown in linear scale if the lines are easy to discriminate, otherwise it is shown in log scale. From the figures we can observe that R2HCA-EMOA outperforms the

other compared algorithms on most of the problems (except for DTLZ4, WFG4, MinusDTLZ1, MinusWFG3). However, as will be seen in Section IV-E, R2HCA-EMOA outperforms the other algorithms on all minus problems based on a different reference point specification for hypervolume calculation. We can also observe that R2HCA-EMOA has a stable performance on most of the problems. The results in Figure 7 clearly show the superiority of R2HCA-EMOA over the other algorithms for achieving high average hypervolume values.

D. Further investigations on R2HCA-EMOA

1) *The effect of the normalization mechanism:* R2HCA-EMOA has introduced the normalization mechanism to enhance the performance of the algorithm. In order to investigate the effectiveness of this mechanism, we compare R2HCA-EMOA with and without the normalization mechanism.

The results are shown in Table III for 5-objective DTLZ and WFG problems. From Table III we can see that R2HCA-EMOA with normalization clearly outperforms that without normalization (+/-/ \approx is 1/5/7 as shown in Table III), which means that the effect of the normalization mechanism is significant on the performance of the algorithm.

TABLE III

STATISTICAL RESULTS OF HYPERVOLUME VALUES OBTAINED BY R2HCA-EMOA WITH AND WITHOUT NORMALIZATION ON DTLZ1-4 AND WFG1-9 WITH 5 OBJECTIVES.

Problem	Without Normalization	With Normalization
DTLZ1	1.5654e+0 (6.30e-4) \approx	1.5657e+0 (5.17e-4)
DTLZ2	1.2855e+0 (1.40e-3) \approx	1.2862e+0 (1.35e-3)
DTLZ3	1.2828e+0 (2.95e-3) \approx	1.2829e+0 (3.30e-3)
DTLZ4	1.0925e+0 (1.96e-1) \approx	1.2122e+0 (1.06e-1)
WFG1	1.5808e+0 (3.13e-2) \approx	1.5895e+0 (1.67e-2)
WFG2	1.5848e+0 (5.11e-3) \approx	1.5849e+0 (7.37e-3)
WFG3	1.0541e+0 (1.31e-2) +	1.0430e+0 (1.26e-2)
WFG4	1.2357e+0 (2.54e-2) -	1.2571e+0 (6.84e-3)
WFG5	1.1872e+0 (5.62e-3) -	1.1980e+0 (2.49e-3)
WFG6	1.1977e+0 (9.40e-3) \approx	1.2057e+0 (1.16e-2)
WFG7	1.2663e+0 (5.49e-3) -	1.2758e+0 (2.34e-3)
WFG8	1.1450e+0 (1.19e-2) -	1.1592e+0 (7.56e-3)
WFG9	1.1874e+0 (2.88e-2) -	1.2023e+0 (2.37e-2)
+ / - / \approx	1/5/7	

We can also observe that R2HCA-EMOA with normalization clearly achieves better results than that without normalization on almost all problems (except for WFG3). Moreover, R2HCA-EMOA with normalization is significantly better than that without normalization on many WFG problems which are MaOPs with a different scale in each objective. As discussed in Section III-F, the normalization mechanism is introduced to tackle this type of problems. We also use a simple update

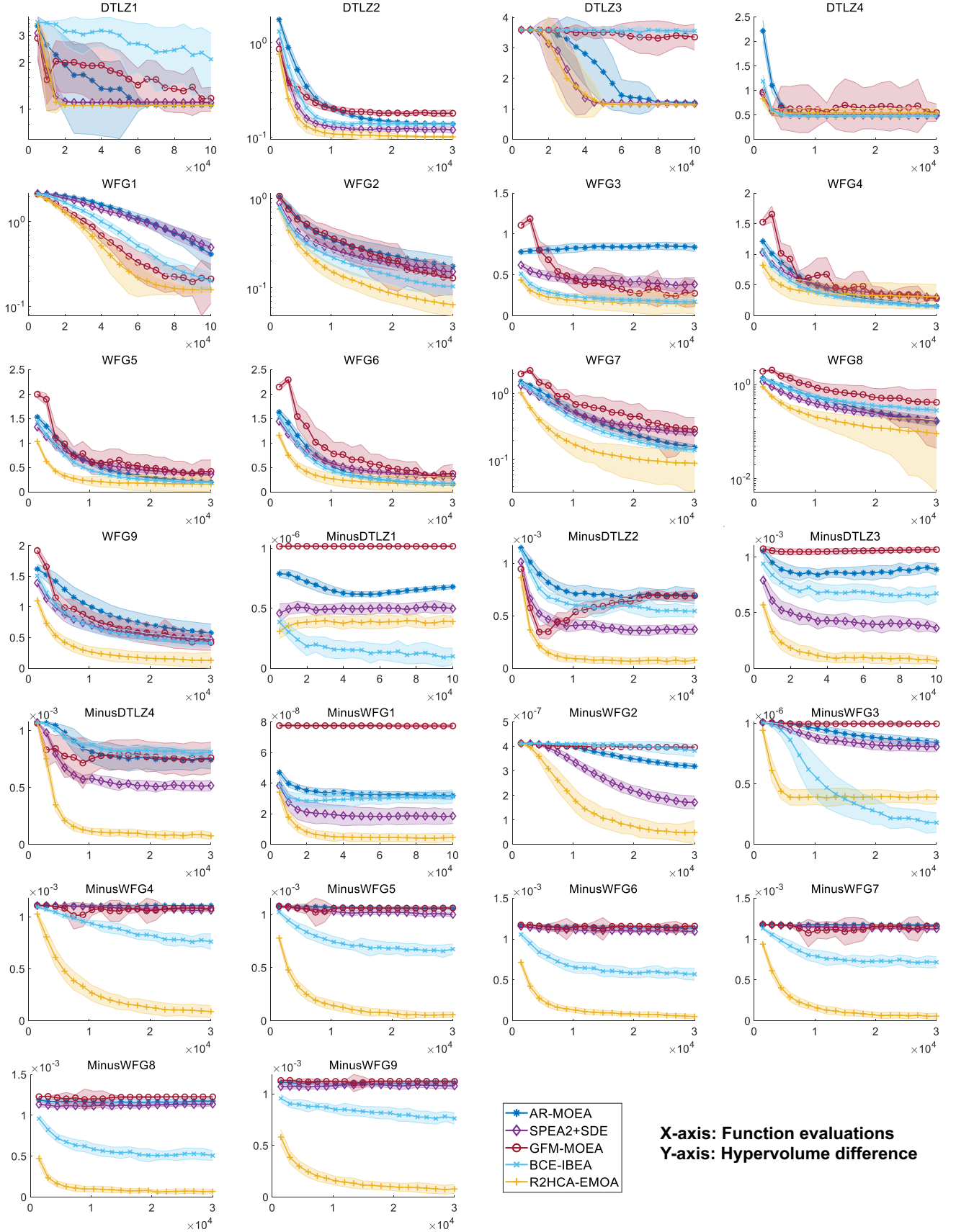


Fig. 7. Hypervolume variations over function evaluations obtained by AR-MOEA, SPEA2+SDE, GFM-MOEA, BCE-IBEA and R2HCA-EMOA on the 10-objective problems. The solid line represents the mean values while the transparent area around the line represents the standard deviations.

strategy to update the tensor \mathbf{T} in order to keep the algorithm computationally efficient. The results in Table III demonstrate the effectiveness of the normalization mechanism together with the simple update strategy in R2HCA-EMOA.

2) *The influence of the number of the direction vectors:* In our experiments, we only use 100 direction vectors in R2HCA-EMOA. Intuitively, a larger number of the direction vectors can improve the performance of R2HCA-EMOA. In order to investigate this issue, we additionally use 50 and 200 direction vectors in R2HCA-EMOA. We test R2HCA-EMOA with 50, 100 and 200 direction vectors on the 5-objective DTLZ and WFG problems. The other settings are the same as in Section IV-A. The results are shown in Table IV.

From Table IV we can see that the results are consistent with our intuition. R2HCA-EMOA with 200 direction vectors outperforms that with 100 direction vectors, and R2HCA-EMOA with 100 direction vectors outperforms that with 50 direction vectors. However, a larger number of the direction vectors means longer runtime. There is a tradeoff between the runtime and the performance of R2HCA-EMOA. The principle to set the number of the direction vectors should depend on the computing power, available time and performance requirement, etc.

TABLE IV
STATISTICAL RESULTS OF HYPERVOLUME VALUES OBTAINED BY R2HCA-EMOA WITH 50, 100 AND 200 DIRECTION VECTORS ON DTLZ1-4 AND WFG1-9 WITH 5 OBJECTIVES.

Problem	$ \Lambda = 50$	$ \Lambda = 200$	$ \Lambda = 100$
DTLZ1	1.5640e+0 −	1.5666e+0 +	1.5657e+0
DTLZ2	1.2808e+0 −	1.2888e+0 +	1.2862e+0
DTLZ3	1.2784e+0 −	1.2855e+0 +	1.2829e+0
DTLZ4	1.2008e+0 ≈	1.1902e+0 ≈	1.2122e+0
WFG1	1.5901e+0 ≈	1.5885e+0 ≈	1.5895e+0
WFG2	1.5824e+0 ≈	1.5861e+0 ≈	1.5849e+0
WFG3	1.0268e+0 −	1.0493e+0 ≈	1.0430e+0
WFG4	1.2469e+0 −	1.2645e+0 +	1.2571e+0
WFG5	1.1927e+0 −	1.2029e+0 +	1.1980e+0
WFG6	1.2047e+0 ≈	1.2135e+0 +	1.2057e+0
WFG7	1.2692e+0 −	1.2808e+0 +	1.2758e+0
WFG8	1.1487e+0 −	1.1617e+0 ≈	1.1592e+0
WFG9	1.2007e+0 −	1.2147e+0 +	1.2023e+0
+ / − / ≈	0/9/4	8/0/5	

3) *The influence of the population size:* In our experiments, we set the population size N to 100 for all test problems. In order to investigate the influence of the population size on the performance of R2HCA-EMOA, we also perform our computational experiments with $N = 50$ and $N = 200$. R2HCA-EMOA is compared with AR-MOEA, SPEA2+SDE, GFM-MOEA and BCE-IBEA on the 5-objective DTLZ and WFG problems. The other settings are the same as in Section IV-A. The results are shown in Table V. Detailed numerical results are provided in Appendix C in the supplementary material.

From Table V we can see that R2HCA-EMOA outperforms all the compared algorithms in terms of the hypervolume metric in both cases: $N = 50$ and $N = 200$. This observation further demonstrate the effectiveness of R2HCA-EMOA for solving MaOPs under different population sizes.

TABLE V
WILCOXON RANK SUM TEST RESULTS BASED ON HYPERVOLUME VALUES OBTAINED BY AR-MOEA, SPEA2+SDE, GFM-MOEA, BCE-IBEA AND R2HCA-EMOA ON 5-OBJECTIVE DTLZ & WFG PROBLEMS.

+ / − / ≈	AR-MOEA	SPEA2+SDE	GFM-MOEA	BCE-IBEA
$N = 50$	1/10/2	0/12/1	1/11/1	1/12/0
$N = 200$	0/13/0	1/12/0	2/10/1	2/11/0

E. Additional Results

In our experiments, the hypervolume indicator is employed to evaluate the performance of R2HCA-EMOA. Since R2HCA-EMOA is a hypervolume-based EMOA, the hypervolume indicator is able to directly evaluate the ability of R2HCA-EMOA for maximizing the hypervolume value of the population. In addition to the hypervolume indicator, the inverted generational distance (IGD) indicator [14] is another widely used performance indicator for the performance evaluation. We provide the IGD results of all the compared EMOAs in this paper in Appendix D in the supplementary material. Interested readers can refer to the detailed discussions there.

In addition, the hypervolume indicator is evaluated based on the reference point specification $r = 1.1$ as explained in Section IV-A4. This reference point specification is commonly used in the EMO community for the hypervolume performance evaluation of EMOAs. However, different reference point specifications may affect the hypervolume evaluation results [31]. Thus, in order to fairly compare the hypervolume performance of different EMOAs, we provide the hypervolume results of all the compared EMOAs based on the reference point specification $r = 1 + 1/H$ in Appendix E in the supplementary material. Detailed discussions can be found there.

Lastly, in order to visually explain the behavior of R2HCA-EMOA, we show the final solution distributions on some 3-objective test problems obtained by R2HCA-EMOA and the other three hypervolume-based EMOAs. The results are provided in Appendix F in the supplementary material.

V. CONCLUSIONS

In this paper, we proposed R2HCA-EMOA, a new hypervolume-based EMOA for many-objective optimization. By using the R2-based hypervolume contribution approximation method in environmental selection, R2HCA-EMOA is able to inherit the advantage of the hypervolume-based EMOAs. At the same time, it is efficient for solving MaOPs. We demonstrated the efficiency and the effectiveness of the proposed algorithm through comparative experiments. First, we showed that R2HCA-EMOA is slightly outperformed by SMS-EMOA and FV-MOEA, both of which need much longer computation time than R2HCA-EMOA. Comparing with HypE, R2HCA-EMOA is superior in terms of both the runtime and the hypervolume metric. Lastly, we compared R2HCA-EMOA with several state-of-the-art EMOAs. Our results showed that R2HCA-EMOA outperforms all the compared algorithms in terms of the hypervolume metric.

As future research, we have the following three research directions: 1) It has been shown that a dynamic reference point specification mechanism can enhance the performance of the hypervolume-based EMOAs [49]. In the future, we will investigate the effect of the dynamic reference point mechanism on the performance of the proposed algorithm; 2) It is interesting to test the proposed algorithm on real-world problems; 3) It is also interesting to extend the proposed algorithm for solving large-scale MOPs [50] and multi-modal MOPs [51].

The source code of R2HCA-EMOA is available at <https://github.com/HisaoLabSUSTC/R2HCA-EMOA>.

REFERENCES

- [1] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in *2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings. NAFIPS-FLINT 2002 (Cat. No. 02TH8622)*. IEEE, 2002, pp. 233–238.
- [2] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, p. 13, 2015.
- [3] 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.
- [4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.
- [5] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 3, pp. 392–403, 1998.
- [6] 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.
- [7] 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.
- [8] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.
- [9] M. Asafuddoula, H. K. Singh, and T. Ray, "An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2321–2334, 2017.
- [10] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 832–842.
- [11] M. Emmerich, N. Beume, and B. Naujoks, "An EMO algorithm using the hypervolume measure as selection criterion," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 62–76.
- [12] 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.
- [13] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms - a comparative case study," in *International Conference on Parallel Problem Solving from Nature*. Springer, 1998, pp. 292–301.
- [14] C. A. C. Coello and M. R. Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *Mexican International Conference on Artificial Intelligence*. Springer, 2004, pp. 688–697.
- [15] M. P. Hansen and A. Jaszkiewicz, *Evaluating the quality of approximations to the non-dominated set*. IMM, Department of Mathematical Modelling, Technical University of Denmark, 1994.
- [16] S. Jiang, J. Zhang, Y.-S. Ong, A. N. Zhang, and P. S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2202–2213, 2015.
- [17] 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.
- [18] K. Bringmann and T. Friedrich, "Approximating the volume of unions and intersections of high-dimensional geometric objects," *Computational Geometry Theory and Applications*, vol. 43, no. 6, pp. 601–610, 2010.
- [19] —, "Approximating the least hypervolume contributor: NP-hard in general, but fast in practice," *Theoretical Computer Science*, vol. 425, no. 2, pp. 104–116, 2012.
- [20] N. Beume, C. M. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1075–1082, 2009.
- [21] M. T. Emmerich and C. M. Fonseca, "Computing hypervolume contributions in low dimensions: Asymptotically optimal algorithm and complexity results," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2011, pp. 121–135.
- [22] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima, "Hypervolume approximation using achievement scalarizing functions for evolutionary many-objective optimization," in *IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 530–537.
- [23] —, "Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2010, pp. 527–534.
- [24] K. Shang, H. Ishibuchi, M.-L. Zhang, and Y. Liu, "A new R2 indicator for better hypervolume approximation," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2018, pp. 745–752.
- [25] K. Shang, H. Ishibuchi, and X. Ni, "R2-based hypervolume contribution approximation," *IEEE Transactions on Evolutionary Computation (Early Access)*, pp. 1–1, 2019.
- [26] J. Bader, K. Deb, and E. Zitzler, "Faster hypervolume-based search using Monte Carlo sampling," in *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*. Springer, 2010, pp. 313–326.
- [27] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *International Conference on Evolutionary Multi-criterion Optimization*. Springer, 2007, pp. 742–756.
- [28] J. J. Durillo, A. J. Nebro, F. Luna, and E. Alba, "On the effect of the steady-state selection scheme in multi-objective genetic algorithms," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2009, pp. 183–197.
- [29] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.
- [30] K. Deb and M. Goyal, "A combined genetic adaptive search (geneas) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.
- [31] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "How to specify a reference point in hypervolume calculation for fair performance comparison," *Evolutionary Computation*, vol. 26, no. 3, pp. 411–440, 2018.
- [32] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2016.
- [33] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.
- [34] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*. Springer, 2005, pp. 105–145.
- [35] 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.
- [36] 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.
- [37] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 609–622, 2018.
- [38] 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.
- [39] Y. Tian, X. Zhang, R. Cheng, C. He, and Y. Jin, "Guiding evolutionary multiobjective optimization with generic front modeling," *IEEE Transactions on Cybernetics*, 2018.

- [40] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 645–665, 2016.
- [41] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [42] J. E. Gentle, *Random number generation and Monte Carlo methods*. Springer Science & Business Media, 2006.
- [43] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.
- [44] 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.
- [45] 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.
- [46] R. Hernández Gómez and C. A. Coello Coello, "Improved metaheuristic based on the R2 indicator for many-objective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2015, pp. 679–686.
- [47] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.
- [48] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 131–152, 2017.
- [49] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, "Dynamic specification of a reference point for hypervolume calculation in SMS-EMOA," in *IEEE Congress on Evolutionary Computation*. IEEE, 2018, pp. 1–8.
- [50] R. Cheng, Y. Jin, M. Olhofer *et al.*, "Test problems for large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4108–4121, 2017.
- [51] Y. Liu, G. G. Yen, and D. Gong, "A multi-modal multi-objective evolutionary algorithm using two-archive and recombination strategies," *IEEE Transactions on Evolutionary Computation*, 2018.