



Improving the Efficiency of R2HCA-EMOA

Ke Shang, Hisao Ishibuchi^(✉), Longcan Chen, Weiyu Chen,
and Lie Meng Pang

Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation,
Department of Computer Science and Engineering, Southern University of Science
and Technology, Shenzhen 518055, China

kshang@foxmail.com, {hisao,panglm}@sustech.edu.cn,
{11813009,11711904}@mail.sustech.edu.cn

Abstract. R2HCA-EMOA is a recently proposed hypervolume-based evolutionary multi-objective optimization (EMO) algorithm. It uses an R2 indicator variant to approximate the hypervolume contribution of each solution. Meanwhile, it uses a utility tensor structure to facilitate the calculation of the R2 indicator variant. This makes it very efficient for solving many-objective optimization problems. Compared with HypE, another hypervolume-based EMO algorithm designed for many-objective problems, R2HCA-EMOA runs faster and at the same time achieves better performance. Thus, R2HCA-EMOA is more attractive for practical use. In this paper, we further improve the efficiency of R2HCA-EMOA without sacrificing its performance. We propose two strategies for the efficiency improvement. One is to simplify the environmental selection, and the other is to change the number of direction vectors depending on the state of evolution. Numerical experiments clearly show that the efficiency of R2HCA-EMOA is significantly improved without deteriorating its performance.

Keywords: Hypervolume indicator · R2HCA-EMOA · Evolutionary multi-objective optimization

1 Introduction

In the field of Evolutionary Multi-objective Optimization (EMO), new algorithms are proposed every year. In general, these algorithms are classified into three categories: Pareto dominance-based (e.g., NSGA-II [5]), decomposition-based (e.g., MOEA/D [18]), and indicator-based (e.g., SMS-EMOA [2, 7]). Among these three categories, indicator-based EMO algorithms transform a multi-objective optimization problem into a single-objective optimization problem where the single-objective is to optimize the value of an indicator (e.g., GD [16], IGD [3], Hypervolume [19], R2 [8]). The performance of EMO algorithms is usually evaluated by indicators. Each indicator-based EMO algorithm is likely to achieve a good performance with respect to the indicator used in

the algorithm. That is, an IGD-based algorithm is likely to achieve a good IGD performance, and a hypervolume-based algorithm is likely to achieve a good hypervolume performance. This is the main advantage of the indicator-based algorithms.

Among different indicators, the hypervolume indicator is the most popular one in the EMO community since it has solid theoretical foundations and is comprehensively investigated [14]. The hypervolume-based EMO algorithms use the hypervolume indicator as the optimization criterion, which have been successfully applied to solve multi- and many-objective optimization problems. Some representative hypervolume-based algorithms are SMS-EMOA [2, 7], FV-MOEA [11], HypE [1], and R2HCA-EMOA [12]. SMS-EMOA and FV-MOEA use the exact hypervolume calculation. The advantage of these two algorithms is that a good hypervolume performance can be achieved. The main disadvantage of these two algorithms is that the exact hypervolume calculation is very time-consuming in high-dimensional objective spaces. Thus, their efficiency is very low when solving many-objective problems.

In order to solve the above issue, HypE and R2HCA-EMOA are proposed. These two algorithms use hypervolume approximation methods in the algorithm implementations, and thus they can solve many-objective problems efficiently. Since the hypervolume is approximated in these two algorithms, their hypervolume performance is worse than SMS-EMOA and FV-MOEA. As reported in [12], R2HCA-EMOA is able to run faster and at the same time achieve better hypervolume performance than HypE. Thus, R2HCA-EMOA is more efficient and effective than HypE for solving many-objective problems.

However, compared with other popular EMO algorithms (e.g., NSGA-II, MOEA/D, NSGA-III [4]), R2HCA-EMOA is still not very efficient (i.e., slower than those algorithms). Thus, the question is whether we can further improve the efficiency of R2HCA-EMOA without sacrificing its high performance. In this paper, we investigate this issue and propose two strategies to further improve the efficiency of R2HCA-EMOA. The improved version of R2HCA-EMOA is more efficient and its performance is not deteriorated. That is, the improved version is more attractive for practical use.

The rest of the paper is organized as follows. R2HCA-EMOA is briefly introduced in Sect. 2. Two efficiency improvement strategies are presented in Sect. 3. Numerical experiments are conducted in Sect. 4. The conclusions are drawn in Sect. 5.

2 R2HCA-EMOA

R2HCA-EMOA follows the framework of SMS-EMOA. In each generation, one offspring is generated and added into the population. Then one individual with the least hypervolume contribution in the last front is removed from the population. The main difference between R2HCA-EMOA and SMS-EMOA is that the hypervolume contribution is approximated in R2HCA-EMOA instead of exactly calculated. The pseudocode of R2HCA-EMOA is shown in Algorithm 1.

Algorithm 1. R2HCA-EMOA**Input:** Population Size N , Maximum Function Evaluations FEs^{\max} .**Output:** Final Population P .

```

1: Initialize Population  $P$ , Direction Vector Set  $\Lambda$ , Utility Tensor  $\mathbf{T}$ , reference point
    $\mathbf{r}$ ,  $FEs = N$ ;
2: while  $FEs \leq FEs^{\max}$  do
3:    $\mathbf{q} = \text{GenerateOffspring}(P)$ ;
4:    $P = P \cup \{\mathbf{q}\}$ ;
5:    $P' = \text{Normalize}(P)$ ;
6:    $\mathbf{T} = \text{UpdateUtilityTensor}(\mathbf{T}, P', \Lambda)$ ;
7:    $\{F_1, F_2, \dots, F_l\} = \text{NondominatedSort}(P')$ ;
8:   if  $|F_l| == 1$  then
9:      $R_2^{\text{HCA}}(\mathbf{s}) = 0, \forall \mathbf{s} \in F_l$ ;
10:  else if  $|F_l| == N + 1$  then
11:     $R_2^{\text{HCA}}(\mathbf{s}) = \text{CalculateR2HCA}(\mathbf{T}), \forall \mathbf{s} \in F_l$ ;
12:  else
13:     $\mathbf{T}' = \text{ExtractUtilityTensor}(\mathbf{T}, F_l)$ ;
14:     $R_2^{\text{HCA}}(\mathbf{s}) = \text{CalculateR2HCA}(\mathbf{T}'), \forall \mathbf{s} \in F_l$ ;
15:  end if
16:   $\mathbf{s}_{\text{worst}} = \arg \min_{\mathbf{s} \in F_l} R_2^{\text{HCA}}(\mathbf{s})$ ;
17:   $P = P \setminus \{\mathbf{s}_{\text{worst}}\}$ ;
18:   $FEs = FEs + 1$ ;
19: end while

```

In R2HCA-EMOA, the hypervolume contribution is approximated by an R2 indicator variant, and a utility tensor structure is used to facilitate the calculation of the R2 indicator variant. Next, the R2 indicator variant and the utility tensor structure are briefly explained.

2.1 R2-Based Hypervolume Contribution Approximation

In R2HCA-EMOA, an R2 indicator variant is used to approximate the hypervolume contribution of a solution \mathbf{s} to a solution set A in an m -dimensional objective space¹:

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

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

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

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

¹ In this paper, the solutions (individuals) are assumed to be points in the objective space, i.e., a solution \mathbf{s} denotes an objective vector.

For maximization problems, it is defined as

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

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

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

The geometric meaning of the R2 indicator variant is illustrated in Fig. 1. Given a non-dominated solution set $\{\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3\}$ and a set of line segments with different directions in the hypervolume contribution region of \mathbf{a}^2 , the R2 indicator variant in (1) is calculated as $R_2^{\text{HCA}}(\mathbf{a}^2) = \frac{1}{|A|} \sum_{i=1}^{|A|} l_i^m$ where l_i is the length of the i th line segment. For more detailed explanations of the R2 indicator variant, please refer to [13].

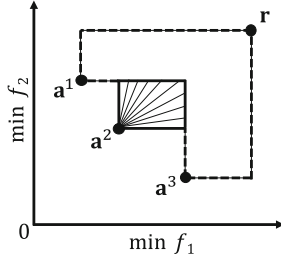


Fig. 1. An illustration of the geometric meaning of R_2^{HCA} for the hypervolume contribution approximation.

2.2 Utility Tensor Structure

In R2HCA-EMOA, a utility tensor structure is introduced to facilitate the calculation of R_2^{HCA} . From the definition of R_2^{HCA} in (1), we can see that in order to compute the R_2^{HCA} value of a solution $\mathbf{s} \in A$, we need to compute $g^{*2\text{tch}}(\mathbf{a}|\boldsymbol{\lambda}, \mathbf{s})$ for all $\mathbf{a} \in A \setminus \{\mathbf{s}\}$ and all $\boldsymbol{\lambda} \in \Lambda$, and $g^{\text{mtch}}(\mathbf{r}|\boldsymbol{\lambda}, \mathbf{s})$ for all $\boldsymbol{\lambda} \in \Lambda$. Thus, we can store $g^{*2\text{tch}}$ and g^{mtch} values in a matrix \mathbf{M} as follows:

$$\mathbf{M} = \begin{bmatrix} g^{*2\text{tch}}(\mathbf{a}_1|\boldsymbol{\lambda}_1, \mathbf{s}) & g^{*2\text{tch}}(\mathbf{a}_1|\boldsymbol{\lambda}_2, \mathbf{s}) & \dots & g^{*2\text{tch}}(\mathbf{a}_1|\boldsymbol{\lambda}_{|A|}, \mathbf{s}) \\ \vdots & \vdots & \ddots & \vdots \\ g^{*2\text{tch}}(\mathbf{a}_{|A|-1}|\boldsymbol{\lambda}_1, \mathbf{s}) & g^{*2\text{tch}}(\mathbf{a}_{|A|-1}|\boldsymbol{\lambda}_2, \mathbf{s}) & \dots & g^{*2\text{tch}}(\mathbf{a}_{|A|-1}|\boldsymbol{\lambda}_{|A|}, \mathbf{s}) \\ g^{\text{mtch}}(\mathbf{r}|\boldsymbol{\lambda}_1, \mathbf{s}) & g^{\text{mtch}}(\mathbf{r}|\boldsymbol{\lambda}_2, \mathbf{s}) & \dots & g^{\text{mtch}}(\mathbf{r}|\boldsymbol{\lambda}_{|A|}, \mathbf{s}) \end{bmatrix}_{|A| \times |A|}$$

After obtaining the above matrix, we can calculate $R_2^{\text{HCA}}(\mathbf{s})$ as follows:

$$R_2^{\text{HCA}}(\mathbf{s}) = \frac{1}{|A|} \sum_{j=1}^{|A|} \left(\min_{i=\{1,2,\dots,|A|\}} M_{ij} \right)^m. \quad (5)$$

Since we need to calculate R_2^{HCA} for each $\mathbf{s} \in A$, we need to have $|A|$ matrices for all solutions in A . These $|A|$ matrices form a tensor

$$\mathbf{T} = [\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{|A|}],$$

where \mathbf{M}_k is the matrix for $\mathbf{a}_k \in A$.

We use tensor \mathbf{T} as a memory to store useful information for the calculation of R_2^{HCA} . As shown in Algorithm 1, since only one offspring is generated and one individual is removed in each generation, we do not need to recalculate tensor \mathbf{T} in each generation. Instead, we only update \mathbf{T} with respect to the changed elements to maximize the efficiency. Furthermore, if all the solutions in the current population are non-dominated, we can use (5) to calculate R_2^{HCA} . However, if there are some dominated solutions, we need to extract useful information from \mathbf{T} to calculate R_2^{HCA} for the last front. For more detailed explanation of tensor \mathbf{T} and its operations, please refer to [12].

3 Efficiency Improvement Strategies

In this section, we propose two strategies to further improve the efficiency of R2HCA-EMOA. The first strategy is to simplify the environmental selection, and the second strategy is to change the number of direction vectors depending on the state of evolution.

3.1 Simplify the Environmental Selection

The first strategy is to simplify the environmental selection in R2HCA-EMOA. As explained in the previous section, we use tensor \mathbf{T} to calculate R_2^{HCA} values, i.e., the approximated hypervolume contributions. If there are dominated solutions, we cannot directly use tensor \mathbf{T} to calculate R_2^{HCA} for the last front. Thus, a tensor extraction is performed to get a subtensor \mathbf{T}' . This will lower the efficiency of R2HCA-EMOA.

The idea is to simplify the environmental selection when there are dominated solutions. Instead of extracting a subtensor \mathbf{T}' and calculating R_2^{HCA} for the last front, we simply randomly remove one solution from the last front. In this way, we can avoid the tensor extraction procedure and the algorithm efficiency can be improved. To be more specific, lines 8–15 in Algorithm 1 are replaced by the following lines:

- 1: **if** $|F_l| == N + 1$ **then**
- 2: $R_2^{\text{HCA}}(\mathbf{s}) = \text{CalculateR2HCA}(\mathbf{T}), \forall \mathbf{s} \in F_l;$
- 3: **else**
- 4: $R_2^{\text{HCA}}(\mathbf{s}) = \text{rand}(0, 1)^2, \forall \mathbf{s} \in F_l;$
- 5: **end if**

The question is whether this simple strategy can deteriorate the performance of R2HCA-EMOA. The answer is NO (which will be verified in the experiments).

² $\text{rand}(0, 1)$ means a random number between 0 and 1.

The reason can be explained as follows. In many-objective optimization, all or almost all solutions in the current population are non-dominated. In this situation, it is not likely that dominated solutions play an important role in multi-objective evolution. That is, it is likely that all dominated solutions in the current population are almost equally useless (if compared with the non-dominated solutions). As a result, it is not needed to carefully evaluate each dominated solution to identify the worst one with the least contribution.

3.2 Change the Number of Direction Vectors

The second strategy is to dynamically change the number of direction vectors used in R2HCA-EMOA. As discussed in [12], the worst-case time complexity of one generation of R2HCA-EMOA is $O(N^2|A|)$. We can see that the time complexity is related to the number of direction vectors $|A|$. We can improve the efficiency of R2HCA-EMOA by decreasing the value of $|A|$. However, a smaller value of $|A|$ means a lower approximation quality of R_2^{HCA} . Thus, if we simply set $|A|$ as a small value, the performance of R_2^{HCA} may be deteriorated.

The idea is to use a small number of $|A|$ in the early stage of the evolution and use a large number of $|A|$ in the late stage of the evolution. In the early stage of the evolution, we do not need a very high approximation quality of R_2^{HCA} since the main focus is the convergence of the population. Also, as we discussed in the first strategy, the solutions in the last front are randomly removed. In this case, we do not need to calculate R_2^{HCA} . In the late stage of the evolution, we use a large number of $|A|$ to achieve a good hypervolume performance.

The question is the timing to change the number of direction vectors. Ideally, the timing for the change is when the population has converged to the Pareto front. However, the Pareto front is usually unknown in practice. Thus, we need to have a mechanism to decide the timing for the change. In this paper, we use the following mechanism:

$$|A| = \begin{cases} \text{a small number, if } \text{FEs} \leq \theta \text{FEs}^{\max} \\ \text{a large number,} & \text{otherwise} \end{cases} \quad (6)$$

where $\theta \in [0, 1]$ is a timing control parameter. The small number and the large number are set to 10 and 100, respectively, in our experiments.

We need to note that when $|A|$ is changed, the direction vector set A is regenerated, and tensor \mathbf{T} needs to be reinitialized since the size of \mathbf{T} is also changed.

4 Experiments

In this section, we examine the effect of the proposed two strategies on the performance of R2HCA-EMOA through computational experiments.

4.1 Experimental Settings

1. **Platforms.** Our experiments are performed on PlatEMO [15], which is a MATLAB-based open source platform for evolutionary multi-objective optimization. The source code of R2HCA-EMOA is available at <https://github.com/HisaoLabSUSTC/R2HCA-EMOA>. The hardware platform is a PC equipped with Intel Core i7-8700K CPU@3.70 GHz, 16 GB RAM.
2. **Test Problems.** DTLZ1-4 [6], WFG1-9 [9], and their minus versions (i.e., MinusDTLZ1-4 and MinusWFG1-9) [10] are selected as test problems in our experiments. The number of objectives is set to $m = 5$. The number of decision variables is set to $m + 4$ for DTLZ1 and MinusDTLZ1, and $m + 9$ for the other DTLZ and MinusDTLZ problems. For WFG and MinusWFG problems, the distance and position decision variables are set to 24 and $m - 1$, respectively.
3. **Parameter Settings.** The population size N is set to 100. 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. The simulated binary crossover (SBX) and the polynomial mutation are adopted in the algorithms 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 run 20 time independently on each test problem.
4. **Performance Metrics.** The hypervolume indicator is used to evaluate the performance of the algorithms, and the runtime is recorded to evaluate the efficiency of the algorithms. We employ the WFG algorithm [17] to calculate the exact hypervolume of the obtained solution sets. The hypervolume is calculated as follows. First, using the true ideal point \mathbf{z}^* and the true nadir point \mathbf{z}^{nad} calculated from the true PF, the objective space is normalized, i.e., the two points \mathbf{z}^* and \mathbf{z}^{nad} are $(0, \dots, 0)$ and $(1, \dots, 1)$ after the normalization. Then, the reference point \mathbf{r} is specified as $(1.1, \dots, 1.1)$ to calculate the hypervolume of the obtained solution set in the normalized objective space.
5. **Algorithm Configurations.** The compared algorithms are R2HCA-EMOA and its improved version (denoted as R2HCA-EMOA-II) equipped with the two strategies. For R2HCA-EMOA, the number of direction vectors $|\Lambda|$ is set to 100. For R2HCA-EMOA-II, $|\Lambda|$ is initially set to 10, and finally changed to 100 according to (6). The timing control parameter θ is set to $\{0.0, 0.3, 0.5, 0.8\}$, i.e., we examine the effect of different θ values. All the results are analyzed by the Wilcoxon rank sum test with a significance level of 0.05. ‘+’, ‘−’ and ‘ \approx ’ indicate that R2HCA-EMOA-II is ‘significantly better than’, ‘significantly worse than’ and ‘statistically similar to’ the original R2HCA-EMOA, respectively.

4.2 Results

The mean hypervolume results and the mean runtime results over 20 runs are shown in Table 1 and Table 2, respectively.

Table 1. Statistical results of hypervolume values obtained by R2HCA-EMOA and R2HCA-EMOA-II. The best result for each test problem is shaded.

Problem	R2HCA-EMOA	R2HCA-EMOA-II			
		$\theta = 0.0$	$\theta = 0.3$	$\theta = 0.5$	$\theta = 0.8$
DTLZ1	1.5657e+0	1.5656e+0 \approx	1.5647e+0 \approx	1.5636e+0 \approx	1.5651e+0 \approx
DTLZ2	1.2852e+0	1.2861e+0 \approx	1.2844e+0 \approx	1.2852e+0 \approx	1.2850e+0 \approx
DTLZ3	1.2826e+0	1.2823e+0 \approx	1.2825e+0 \approx	1.2826e+0 \approx	1.2797e+0 $-$
DTLZ4	1.1013e+0	1.1123e+0 \approx	1.1372e+0 \approx	1.1152e+0 \approx	1.1387e+0 \approx
MinusDTLZ1	1.6317e-2	1.6310e-2 \approx	1.6233e-2 \approx	1.6327e-2 \approx	1.6294e-2 \approx
MinusDTLZ2	2.0310e-1	2.0346e-1 \approx	2.0264e-1 \approx	2.0282e-1 \approx	2.0270e-1 \approx
MinusDTLZ3	2.0302e-1	2.0287e-1 \approx	2.0283e-1 \approx	2.0298e-1 \approx	2.0280e-1 \approx
MinusDTLZ4	2.0184e-1	2.0203e-1 \approx	2.0190e-1 \approx	2.0208e-1 \approx	2.0185e-1 \approx
WFG1	1.6027e+0	1.6005e+0 \approx	1.5981e+0 \approx	1.5945e+0 \approx	1.5888e+0 \approx
WFG2	1.5915e+0	1.5921e+0 \approx	1.5913e+0 \approx	1.5902e+0 \approx	1.5872e+0 $-$
WFG3	3.1829e-1	2.7968e-1 \approx	2.9585e-1 \approx	2.6692e-1 $-$	2.6957e-1 $-$
WFG4	1.2607e+0	1.2636e+0 \approx	1.2593e+0 \approx	1.2585e+0 \approx	1.2546e+0 $-$
WFG5	1.1983e+0	1.1988e+0 \approx	1.1987e+0 \approx	1.1979e+0 \approx	1.1972e+0 \approx
WFG6	1.2002e+0	1.2066e+0 \approx	1.2061e+0 $+$	1.1985e+0 \approx	1.2016e+0 \approx
WFG7	1.2767e+0	1.2774e+0 \approx	1.2768e+0 \approx	1.2760e+0 \approx	1.2739e+0 $-$
WFG8	1.1532e+0	1.1527e+0 \approx	1.1462e+0 $-$	1.1434e+0 $-$	1.1419e+0 $-$
WFG9	1.2124e+0	1.2186e+0 $+$	1.2149e+0 \approx	1.2145e+0 \approx	1.1968e+0 $-$
MinusWFG1	4.1141e-3	4.1346e-3 \approx	4.1124e-3 \approx	4.0299e-3 $-$	4.0278e-3 $-$
MinusWFG2	1.3725e-2	1.3776e-2 \approx	1.3681e-2 \approx	1.3743e-2 \approx	1.3651e-2 \approx
MinusWFG3	1.6000e-2	1.5931e-2 \approx	1.5991e-2 \approx	1.5972e-2 \approx	1.5887e-2 \approx
MinusWFG4	2.0259e-1	2.0243e-1 \approx	2.0226e-1 \approx	2.0247e-1 \approx	2.0230e-1 \approx
MinusWFG5	1.9853e-1	1.9895e-1 \approx	1.9904e-1 \approx	1.9814e-1 \approx	1.9717e-1 $-$
MinusWFG6	2.0067e-1	2.0059e-1 \approx	2.0092e-1 \approx	2.0039e-1 \approx	2.0101e-1 \approx
MinusWFG7	2.0072e-1	2.0093e-1 \approx	2.0006e-1 \approx	2.0056e-1 \approx	2.0029e-1 \approx
MinusWFG8	2.0230e-1	2.0211e-1 \approx	2.0221e-1 \approx	2.0231e-1 \approx	2.0270e-1 \approx
MinusWFG9	1.9961e-1	1.9952e-1 \approx	1.9956e-1 \approx	1.9945e-1 \approx	1.9858e-1 $-$
+ / - / \approx	-	1/0/25	1/1/24	0/3/23	0/10/16

From Table 1, we can see that R2HCA-EMOA-II achieves a comparable hypervolume performance to R2HCA-EMOA when $\theta = 0.0, 0.3$. When $\theta = 0.0$ (i.e., we only consider the first strategy), R2HCA-EMOA-II is significantly better on WFG9. When $\theta = 0.3$, it is significantly better on WFG6 and significantly worse on WFG8, and has no significant difference on all the other test problems. It should be noted that the difference in the average hypervolume value between the two algorithms is very small even when there exists a statistically significant difference (see the results on WFG6, WFG8 and WFG9). When $\theta = 0.5$, R2HCA-EMOA-II is slightly worse than R2HCA-EMOA. It is significantly worse on 3 out of the 26 test problems, and has no significant difference on all the other test problems. When $\theta = 0.8$, the performance of R2HCA-EMOA-II is clearly deteriorated. It is significantly worse on 10 out of the 26 test problems. From these results, we can see that the first strategy does not degrade the performance of R2HCA-EMOA-II (i.e., when $\theta = 0.0$). We can also see that the second strategy has a large effect on the performance of R2HCA-EMOA-II depending on

Table 2. Statistical results of runtime (in seconds) of R2HCA-EMOA and R2HCA-EMOA-II. The best result for each test problem is shaded.

Problem	R2HCA-EMOA	R2HCA-EMOA-II			
		$\theta = 0.0$	$\theta = 0.3$	$\theta = 0.5$	$\theta = 0.8$
DTLZ1	4.5574e+2	4.5307e+2 \approx	3.5623e+2 +	2.7796e+2 +	1.6856e+2 +
DTLZ2	1.6576e+2	1.6766e+2 \approx	1.3070e+2 +	1.0260e+2 +	6.1985e+1 +
DTLZ3	4.0426e+2	3.8419e+2 +	3.2252e+2 +	2.5434e+2 +	1.5958e+2 +
DTLZ4	1.6334e+2	1.6439e+2 \approx	1.3528e+2 +	1.0462e+2 +	6.2214e+1 +
MinusDTLZ1	5.3837e+2	5.3888e+2 \approx	4.3997e+2 +	3.4244e+2 +	2.0172e+2 +
MinusDTLZ2	1.6480e+2	1.7038e+2 \approx	1.3480e+2 +	1.0591e+2 +	6.1981e+1 +
MinusDTLZ3	5.2682e+2	5.2273e+2 \approx	4.9828e+2 +	3.3254e+2 +	1.9823e+2 +
MinusDTLZ4	1.8391e+2	1.7922e+2 +	1.4052e+2 +	1.1022e+2 +	6.5740e+1 +
WFG1	4.6995e+2	4.7799e+2 \approx	3.9469e+2 +	3.1191e+2 +	1.8881e+2 +
WFG2	1.5686e+2	1.5491e+2 \approx	1.3662e+2 +	1.0389e+2 +	6.2514e+1 +
WFG3	1.7011e+2	1.6653e+2 \approx	1.3885e+2 +	1.0886e+2 +	6.4457e+1 +
WFG4	1.5867e+2	1.5813e+2 \approx	1.3048e+2 +	1.0211e+2 +	6.2384e+1 +
WFG5	1.6327e+2	1.8309e+2 \approx	1.3231e+2 +	1.0401e+2 +	6.6410e+1 +
WFG6	1.4974e+2	1.5208e+2 \approx	1.2421e+2 +	9.5330e+1 +	6.2164e+1 +
WFG7	1.7443e+2	1.7106e+2 \approx	1.3610e+2 +	1.0749e+2 +	6.4252e+1 +
WFG8	1.3313e+2	1.3389e+2 \approx	1.1160e+2 +	8.9537e+1 +	5.6189e+1 +
WFG9	1.7798e+2	1.7765e+2 \approx	1.4325e+2 +	1.1197e+2 +	6.6564e+1 +
MinusWFG1	4.6240e+2	4.6057e+2 \approx	3.6451e+2 +	2.9485e+2 +	1.8118e+2 +
MinusWFG2	1.5324e+2	1.5700e+2 \approx	1.2389e+2 +	9.8582e+1 +	6.0794e+1 +
MinusWFG3	1.7885e+2	1.7575e+2 \approx	1.4634e+2 +	1.1536e+2 +	6.9738e+1 +
MinusWFG4	1.7316e+2	1.7272e+2 \approx	1.4242e+2 +	1.1180e+2 +	6.7144e+1 +
MinusWFG5	1.6915e+2	1.6888e+2 \approx	1.3963e+2 +	1.0950e+2 +	6.5649e+1 +
MinusWFG6	1.6638e+2	1.6679e+2 \approx	1.3882e+2 +	1.0950e+2 +	6.6004e+1 +
MinusWFG7	1.8269e+2	1.8287e+2 \approx	1.4910e+2 +	1.1705e+2 +	7.8022e+1 +
MinusWFG8	1.7359e+2	1.7494e+2 \approx	1.4802e+2 +	1.1267e+2 +	6.7384e+1 +
MinusWFG9	1.8258e+2	1.8270e+2 \approx	1.4986e+2 +	1.1747e+2 +	7.0116e+1 +
+ / - / \approx		2/0/24	26/0/0	26/0/0	26/0/0

the value of θ . A too large θ value can deteriorate the performance of R2HCA-EMOA-II. In our results, $\theta = 0.5$ is an acceptable value since the performance of R2HCA-EMOA-II is almost the same as that of R2HCA-EMOA.

From Table 2, we can see that the efficiency of R2HCA-EMOA-II is slightly improved by the first strategy when $\theta = 0.0$. The runtime of R2HCA-EMOA-II is significantly better on 2 out of the 26 test problems. This is because the first strategy only works when there exist dominated solutions. If all the solutions are non-dominated, the two algorithms are exactly the same. It is very likely that the first strategy only works for a small number of generations. For the second strategy, the efficiency of R2HCA-EMOA-II is significantly improved compared with R2HCA-EMOA. We can also see that R2HCA-EMOA-II runs faster as θ increases. Take DTLZ1 as an example, when $\theta = 0.3$, the runtime of R2HCA-EMOA-II is about 0.78 times that of R2HCA-EMOA. When $\theta = 0.5$, the runtime of R2HCA-EMOA-II is about 0.61 times that of R2HCA-EMOA. When $\theta = 0.8$, this number is about 0.37. Similar results are obtained for other test problems.

Based on Table 1 and Table 2, we can see that if θ is properly specified (e.g., $\theta = 0.3, 0.5$), we can improve the efficiency of R2HCA-EMOA without sacrificing its performance, which suggests the usefulness of the proposed two strategies.

5 Conclusions

In this paper, we proposed two strategies to further improve the efficiency of R2HCA-EMOA. The first strategy is to randomly remove one individual in the last front when there are dominated solutions. The second strategy is to use a small number of direction vectors in the early stage and a large number of direction vectors in the late stage of the evolution. The experimental results verified the effectiveness of the proposed two strategies. The efficiency of R2HCA-EMOA was improved without sacrificing its hypervolume performance. The results in this paper make R2HCA-EMOA more attractive for practical use.

In the future, the improved R2HCA-EMOA can be tested on more test problems and real-world problems. It is also interesting to propose other strategies to further improve the efficiency of R2HCA-EMOA. For example, removing the non-dominated sorting procedure and using the R2 indicator variant (i.e., R_2^{HCA}) to evaluate all the solutions (not only the solutions in the last front) is an interesting way to go.

Acknowledgments. This work was supported by National Natural Science Foundation of China (Grant No. 62002152, 61876075), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531), the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008).

References

1. Bader, J., Zitzler, E.: HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.* **19**(1), 45–76 (2011)
2. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **181**(3), 1653–1669 (2007)
3. Coello Coello, C.A., Reyes Sierra, M.: A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 688–697. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24694-7_71
4. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.* **18**(4), 577–601 (2013)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
6. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) *Evolutionary Multiobjective Optimization*. AI&KP, pp. 105–145. Springer, London (2005). https://doi.org/10.1007/1-84628-137-7_6

7. Emmerich, M., Beume, N., Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 62–76. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31880-4_5
8. Hansen, M.P., Jaszekiewicz, A.: Evaluating the quality of approximations to the non-dominated set. IMM, Department of Mathematical Modelling, Technical University of Denmark (1998)
9. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comput.* **10**(5), 477–506 (2006)
10. Ishibuchi, H., Setoguchi, Y., Masuda, H., Nojima, Y.: Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Trans. Evol. Comput.* **21**(2), 169–190 (2017)
11. Jiang, S., Zhang, J., Ong, Y.S., Zhang, A.N., Tan, P.S.: A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. *IEEE Trans. Cybern.* **45**(10), 2202–2213 (2015)
12. Shang, K., Ishibuchi, H.: A new hypervolume-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **24**(5), 839–852 (2020)
13. Shang, K., Ishibuchi, H., Ni, X.: R2-based hypervolume contribution approximation. *IEEE Trans. Evol. Comput.* **24**(1), 185–192 (2020)
14. Shang, K., Ishibuchi, H., He, L., Pang, L.M.: A survey on the hypervolume indicator in evolutionary multi-objective optimization. *IEEE Trans. Evol. Comput.* **25**(1), 1–20 (2021)
15. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput. Intell. Mag.* **12**(4), 73–87 (2017)
16. Van Veldhuizen, D.A.: Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Ph.D. thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio (1999)
17. While, L., Bradstreet, L., Barone, L.: A fast way of calculating exact hypervolumes. *IEEE Trans. Evol. Comput.* **16**(1), 86–95 (2012)
18. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007)
19. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **7**(2), 117–132 (2003)