2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Ensemble R2-based Hypervolume Contribution Approximation

Guotong Wu, Tianye Shu, Yang Nan, Ke Shang*, Hisao Ishibuchi*
Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China

***Abstract*—The hypervolume-based multi-objective evolutionary algorithms (HV-MOEAs) have proven to be highly effective in solving multi-objective optimization problems. However, the computation time of the hypervolume calculation increases significantly as the number of objectives increases. To address this issue, an R2-based hypervolume contribution approximation (R2-HVC) method was proposed. Nevertheless, the original R2-HVC generates a large number of vectors and computes the HVC only once. In this study, we propose an ensemble method based on the R2-HVC method. By using a small number of vectors for repetitive computation and majority voting, the ensemble method can reduce the probability of making incorrect choices. Experimental results show that the proposed method can improve the approximation accuracy while maintaining a similar computation time to the original R2-HVC method.**

***Index Terms*—Multi-objective Optimization, Hypervolume Contribution, Ensemble Method, Majority Voting**

I. Introduction

In multi-objective optimization problems (MOPs), a set of m conflicting objectives $\boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$ are optimized simultaneously, where each objective is a function of d decision variables $(x_1, \ldots, x_d)$.

Over the years, various multi-objective evolutionary algorithms (MOEAs) have been proposed to solve MOPs. In the literature, MOEAs can be classified into three categories: Pareto dominance-based algorithms [1], decomposition-based algorithms [2], and indicator-based algorithms [3, 4]. In the conventional Pareto dominance-based algorithms such as NSGA-II [1] and SPEA2 [5], the quality of a solution is determined by its rank in the non-dominated sorting. However, as the number of objectives increases, the number of non-dominated solutions also increases, which leads to convergence issues. The decomposition-based algorithms (e.g., MOEA/D [2]) decompose MOPs into a set of subproblems and optimize them simultaneously, showing an excellent performance for both multi-objective and many-objective problems [6]. However, the MOEA/D's performance is highly dependent on the shape of the Pareto front [7]. The MOEA/D with a standard weight vector set is less effective on problems with an irregular Pareto front [8]. The indicator-based algorithms, such as IBEA [4] and SMS-EMOA [3], employ performance indicators like hypervolume (HV) [9], inverted generational distance (IGD) [10], and R2 indicator [11] to guide the evolutionary search. These indicators measure the quality of the current population and guide the algorithm to obtain the optimal set.

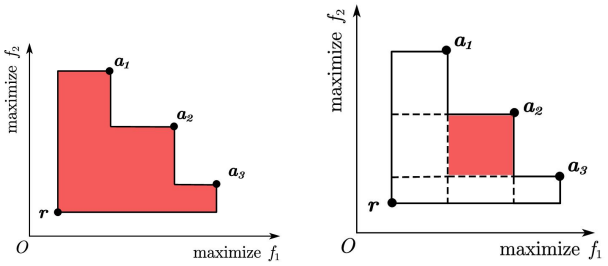
Among the indicator-based algorithms, hypervolume-based MOEAs convert the multi-objective problem into a single-objective problem, which performs exceptionally well on MOPs with two or three objectives. However, the computation of hypervolume is NP-hard, and its computation time grows exponentially with the increase of the dimension of objective space [12], making it impractical for many-objective problems. To deal with this issue, hypervolume approximation methods such as the Monte Carlo sampling method [13] and R2-based HV approximation (R2-HV) method [14] were introduced to speed up the computation.

The goal of SMS-EMOA is to maximize the hypervolume of the population. To achieve this goal, the algorithm computes the hypervolume contribution (HVC) of each solution individually. At each iteration, the solution with the lowest HVC is removed from the population. In the conventional method, the HVC of a solution $\boldsymbol{s}$ to a population A is calculated as the difference between $\mathrm{HV}(A)$ and $\mathrm{HV}(A \backslash \{\boldsymbol{s}\})$. To compute the HVC approximation directly, a new R2-based method called R2-based HVC approximation (R2-HVC) has been introduced [15]. Experimental results demonstrate that the R2-HVC outperforms both the Monte Carlo sampling method and the traditional R2-based method in terms of accuracy and efficiency [15].

In the original R2-HVC, a large number of vectors are utilized, and the worst solution is selected by estimating the HVC only once, leading to potential errors. To address this issue, we present an ensemble method based on the original R2-HVC in this paper. The main idea is to use a small number of vectors to perform multiple approximations and select the solution with the highest frequency through voting.

The remainder of this paper is organized as follows. In Section II, the concepts of HV and HVC are briefly explained. Then the R2-HVC and the correct identification rate (CIR)

This work was supported by the National Key R&D Program of China (Grant No. 2023YFE0106300), National Natural Science Foundation of China (Grant No. 62002152, 62250710163, 62250710682), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), The Stable Support Plan Program of Shenzhen Natural Science Fund (Grant No. 20200925174447003), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531).
*Corresponding authors: Ke Shang (kshang@foxmail.com), Hisao Ishibuchi (hisao@sustech.edu.cn)



(a) HV (b) HVC

Fig. 1. Illustration of (a) the HV of the solution set $A = \{\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}\}$, and (b) the HVC of the single solution $\boldsymbol{a_2}$ to A.

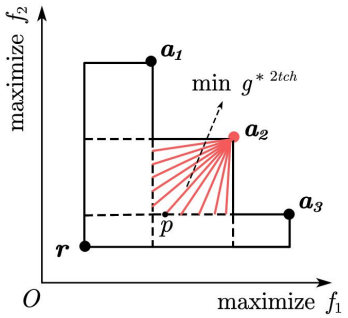


Fig. 2. A 2-objective case using R2-based method to approximate the HVC of solution $\boldsymbol{a_2}$.

indicator are introduced. In Section III, we first present a baseline ensemble method and then improve its ensemble process. Section IV compares our proposed method with the original R2-HVC to validate the effectiveness of the number of vectors and the number of iterations. In Section V, we make two improvements based on the sampling ensemble method and evaluate them experimentally. Finally, we conclude and discuss this paper in Section VI.

II. Preliminaries

In this section, we briefly explain the concepts of hypervolume (HV) and hypervolume contribution (HVC). Then, we introduce the R2-HVC, which is used to estimate the HVC and serves as the basis for our ensemble method. Additionally, we introduce the correct identification rate (CIR). We also explain the limitation of the R2-HVC.

A. Hypervolume and Hypervolume Contribution

Throughout this paper, we assume the maximization of m objectives. We define $A \subset \mathbb{R}^m$ as a solution set, $\boldsymbol{r} \in \mathbb{R}^m$ as a reference point for HV calculation, and $\boldsymbol{s}$ as a single solution in the set A. Figure 1 illustrates the concepts of HV and HVC for a 2-objective maximization problem. Specifically, Figure 1(a) depicts the HV as the area enclosed by the solution set A and the reference point $\boldsymbol{r}$. Figure 1(b) explains the HVC of $\boldsymbol{a_2}$ as the red region which is only dominated by $\boldsymbol{a_2}$.

The mathematical formula for the HV of the set A is defined as:

$$\mathrm{HV}(A, \boldsymbol{r}) = \mathcal{L}\left(\bigcup_{\boldsymbol{a} \in A} \{\boldsymbol{b} \mid \boldsymbol{a} \succ \boldsymbol{b} \succ \boldsymbol{r}\}\right), \tag{1}$$

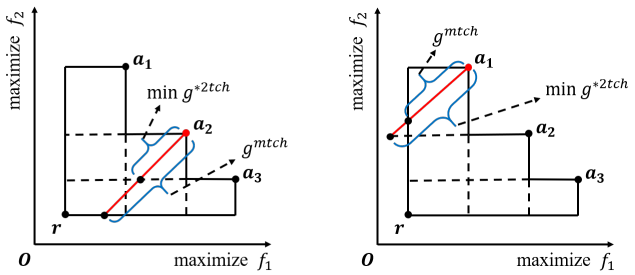
where $\mathcal{L}$ is the Lebesgue measure of a set, and $\boldsymbol{a} \succ \boldsymbol{b}$ means $\boldsymbol{b}$ is dominated by $\boldsymbol{a}$.

The HVC of a single point $\boldsymbol{s}$ to the set A is the difference of HV(A) and HV($A\backslash\{\boldsymbol{s}\}$), which can be written as:

$$\mathrm{HVC}(\boldsymbol{s}, A, \boldsymbol{r}) = \mathrm{HV}(A, \boldsymbol{r}) - \mathrm{HV}(A\backslash\{\boldsymbol{s}\}, \boldsymbol{r}). \tag{2}$$

B. R2-based Hypervolume Contribution Approximation

As computing the exact value of HVC can be time-consuming, Shang et al. [15] proposed a method that calculates HVC using the length of the vectors. Figure 2 depicts the usage of a set of vectors to estimate the area of HVC.



(a) Situation 1 (b) Situation 2

Fig. 3. Illustrations of (a) the vector which intersects with the attainment surface of the set $A\backslash\{s\}$, and (b) the vector intersects with the attainment of the reference point.

In situation 1 in Figure 3(a), each vector starts from the single point $\boldsymbol{s} = \boldsymbol{a_2}$ and intersects with the attainment surface of the set $A\backslash\{\boldsymbol{s}\}$ (i.e., in Figure 3(a)). The length is calculated using the following formula:

$$L_1(\boldsymbol{s}, A\backslash\{\boldsymbol{s}\}, \lambda) = \min_{\boldsymbol{a} \in A\backslash\{\boldsymbol{s}\}} \{g^{*2tch}(\boldsymbol{a}|\lambda, \boldsymbol{s})\}, \tag{3}$$

where $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)$ is one of the vectors, the minimum value in Equation (3) is the length from $\boldsymbol{s}$ to the attainment surface, and g^{*2tch} is the 2-Tch function defined as follows:

$$g^{*2tch}(\boldsymbol{a}|\lambda, \boldsymbol{s}) = \max_{j \in \{1,\ldots,m\}} \frac{s_j - a_j}{\lambda_j}, \tag{4}$$

In Situation 2 in Figure 3(b), the intersection point is located on the edge of the reference point, as shown in Figure 3(b). The length is calculated as follows:

$$L_2(\boldsymbol{s}, \boldsymbol{r}, \lambda) = g^{mtch}(\boldsymbol{r}|\lambda, \boldsymbol{s}) = \min_{j \in \{1,\ldots,m\}} \frac{|s_j - r_j|}{\lambda_j}, \tag{5}$$

At each calculation, the smallest value of L_1 and L_2 is chosen as the length of the vector λ:

$$L_\lambda = \min\{L_1(\boldsymbol{s}, A\backslash\{\boldsymbol{s}\}, \lambda), L_2(\boldsymbol{s}, \boldsymbol{r}, \lambda)\}. \tag{6}$$

The average of all vector lengths with m-th power is calculated in Equation (7) as the final HVC approximation, where V is a set of vectors and m is the number of objectives.

$$R_2^{HVC}(\boldsymbol{s}, A, V, \boldsymbol{r}) = \frac{1}{|V|} \sum_{\lambda \in V} L_\lambda^m. \tag{7}$$

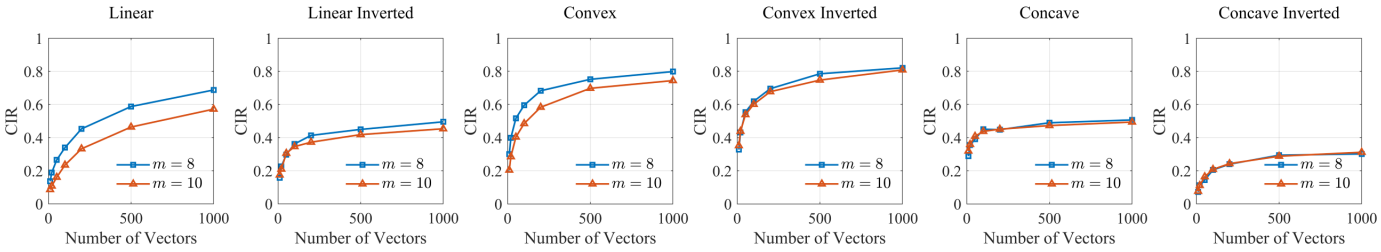


Fig. 4. Experimental results of the R2-HVC on different PF shapes.

Note that Equation (4) is only applicable for maximizing all objectives in MOPs, which is assumed throughout this paper. In the case of minimization, the positions of s_j and a_j should be swapped. For details. please refer to [15].

C. Correct Identification Rate

The correct identification rate (CIR) measures the ability to correctly identify the worst solution using the approximated HVC. The approximation method evaluates l solution sets, each containing n solutions (e.g., $S_1 = \{a_1^1, a_1^2, ..., a_1^n\}$). From each set S_i, we select k worst solutions according to the approximated HVC and name them S_i^{R2}. Furthermore, a_i^{Worst} represents the worst solution calculated according to the true HVC. A correct identification occurs if a_i^{Worst} is included in S_i^{R2}. CIR is the ratio of the total number of correct identifications to the total number of solution sets.

R2HCA-EMOA [16] is a multi-objective evolutionary algorithm that uses the R2-HVC to estimate the HVC. In each iteration, R2HCA-EMOA searches for the solution with the smallest HVC and eliminates it. In this process, we only focus on the worst solution. Consequently, this paper solely considers the scenario where $k = 1$.

To evaluate the R2-HVC, we conduct a straightforward experiment and measure its performance using the CIR indicator. We generate solution sets from m-objective maximization problem with various Pareto front (PF) shapes, including linear, convex, concave, linear inverted, convex inverted, and concave inverted. Each Pareto front shape contains 100 solution sets, and each solution set contains 100 solutions. We use the R2-HVC to select the worst solution from each solution set, and then calculate the CIR value based on the different shapes. Let us take the 8-objective and 10-objective problems as examples and observe the change in the CIR value.

The formulations of the linear, concave and convex PFs are as follows: $f_1+f_2+...+f_m = 1$, $f_1^2+f_2^2+...+f_m^2 = 1$, $\sqrt{f_1}+\sqrt{f_2}+...+\sqrt{f_m} = 1$, where $f_i \in [0, 1]$. For inverted cases, we invert the solution and remap it to $[0, 1]$ using the formula $f_i = 1 - f_i$, where $i = 1, 2, ..., m$. We use the Unit Norm Vector (UNV) method to generate vectors in the R2-HVC, as recommended in [17]. The UNV method firstly samples a set of vectors from a normal distribution and then transforms these vectors to satisfy $\sum_{i=1}^{M} \lambda_i = 1$ [18]. The reference point is fixed at $(-0.1, ..., -0.1)$. We specify the number of vectors as $|V| \in \{10, 20, 50, 100, 200, 500, 1000\}$ for each experiment and repeat 21 runs to obtain the average CIR value.

Figure 4 illustrates the change of the CIR value with the number of vectors for 8-objective and 10-objective problems, respectively. Different subplots represent different PF shapes. A general trend in Figure 4 is that increasing the number of vectors improves the identification accuracy. However, this trend is not effective for some specific shapes of solutions. For instance, the CIR values for inverted concave PFs (the rightmost figure in Figure 4) are about 0.3 even when the number of vectors is 1000. In general, better results are obtained for 8-objective problems than 10-objective problems.

D. The Limitation of R2-HVC

In the original R2-HVC, the length of vectors is used to approximate the area of the HVC, which means the R2-HVC depends on the choice of vectors. As we have already explained in Figure 4, the increase in the number of vectors improves the accuracy of R2-HVC. However, even when we use 1000 vectors, the accuracy of R2-HVC is not so high in some cases. Moreover, the increase in the number of vectors also extends the computation time. Thus, we cannot use a large number of vectors. To further examine the characteristic features of R2-HVC, we generate a set of 100 solutions from the 8-objective linear triangular PF. Then we apply R2-HVC with 1000 vectors 1000 times to select the worst solution. Due to the randomness in creating vectors in R2-HVC, a different solution can be selected as the worst solution in each of the 1000 runs. We count the number of runs where each solution is selected as the worst solution. Experimental results are summarized in Figure 5 where the true worst solution is solution 11. In Figure 5, this solution is correctly selected in about 300 runs. This means that the worst solution selection is incorrect in about 700 runs. From Figure 5, we choose the top eight solutions and show the true HVC of each solution in Table I. From Table I, we can see that the difference of HVC between the true worst solution (i.e., solution 11) and the second most frequently selected solution (i.e., solution 76) is very small. That is, we need a very accurate HVC approximation for correctly selecting solution 11. Whereas the correct selection of solution 11 is about 300 runs (i.e., about 30% accuracy), we can see that the correct solution (i.e., solution 11) is most frequently selected in Figure 5. This observation motivates us to use a voting scheme. If we use a simple majority vote in Figure 5, solution 11 is correctly selected whereas the average accuracy of individual runs is about 30%.

TABLE I
THE TRUE HYPERVOLUME CONTRIBUTION (HVC) OF THE TOP 8 SOLUTIONS IN FIGURE 5

Index	Solution 11	Solution 76	Solution 68	Solution 71	Solution 13	Solution 80	Solution 93	Solution 20
HVC (1e-6)	0.1293	0.1354	0.1468	0.1682	0.1719	0.1798	0.1911	0.1925

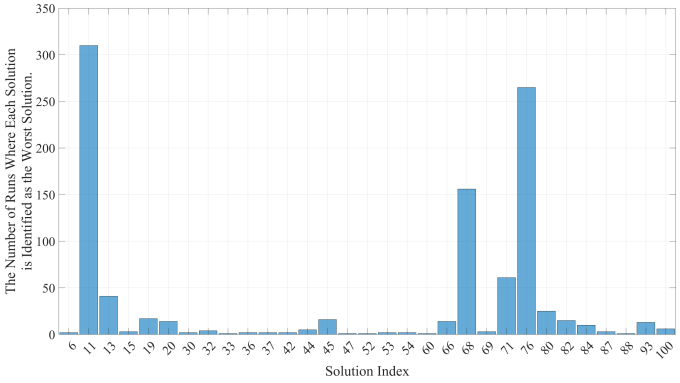
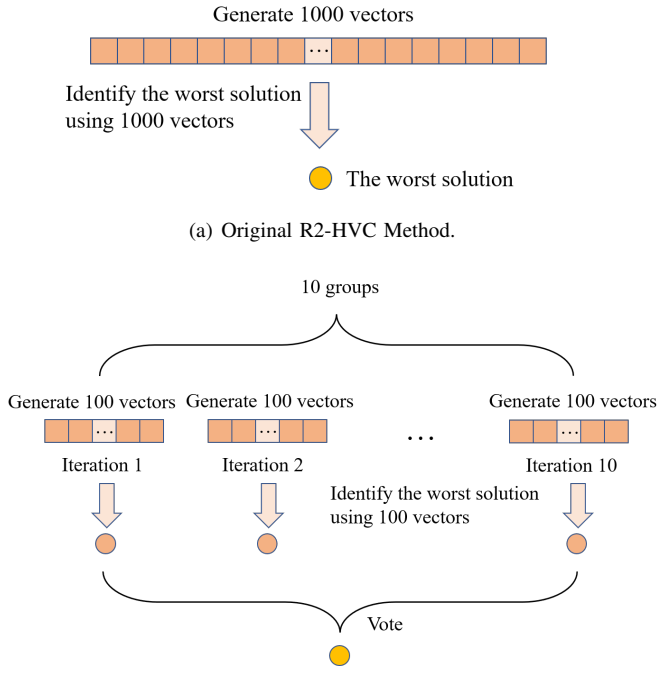


Fig. 5. Illustration of an 8-objective set using the R2-HVC to select the worst solution in 1000 runs. The x-axis is the solution index, and the y-axis is the number of runs where each solution is identified as the worst solution.

III. PROPOSED METHOD

In this section, we present our ensemble R2-HVC method. The first is a baseline ensemble method, which simply repeats R2-HVC and conducts majority voting. Based on the experimental results, we enhance the ensemble process to improve its performance.



(a) Original R2-HVC Method.

(b) Baseline Ensemble Method: Generate Different Vectors.

Fig. 6. The simple example to illustrate the different processes of the original R2-HVC method and the baseline ensemble method.

A. Baseline Ensemble Method

The baseline method repeats the original R2-HVC multiple times but uses fewer vectors generated for each iteration. In each iteration, new vectors are generated and R2-HVC is applied to estimate the HVC of each solution. The solution with the lowest HVC value is selected in each iteration. The final result is the solution that is most frequently selected. When multiple solutions are equally chosen, we randomly select one of them.

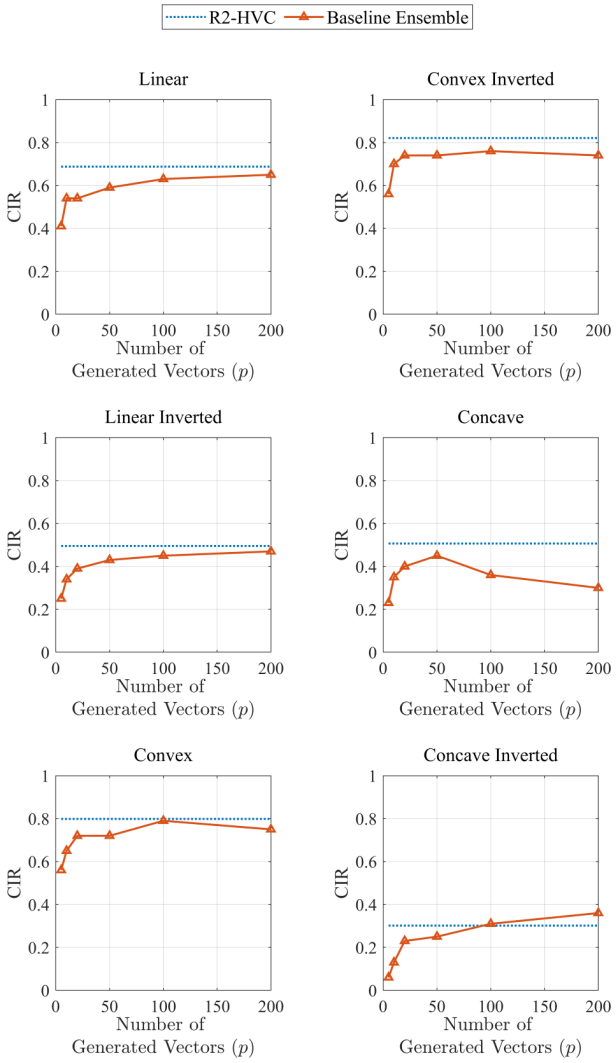


Fig. 7. CIR values of the original R2-HVC and the baseline ensemble method in the 10-objective problem.

The difference between the original R2-HVC and the baseline ensemble method is illustrated in Figure 6. Specifically, Figure 6(a) shows that the original R2-HVC determines the

worst solution based on a single calculation using 1000 vectors. In contrast, Figure 6(b) shows that the baseline ensemble method generates 100 vectors in each iteration, repeats the process 10 times, and then selects the most frequently selected solution.

To verify the performance of this baseline method, we use the solution sets with six PF shapes. Same as in Section II-C, each shape contains 100 sets, and each set contains 100 solutions. We generate different vectors in different iterations and set the number of vectors p and the number of iterations q as $(p, q) \in \{(5, 200), (10, 100), (20, 50), (50, 20), (100, 10), (200, 5)\}$. Thus, the total number of generated vectors in each case (i.e., $p_i \times q_i$) is fixed as 1000.

We repeat the experiment in 21 runs, and show the average result in Figure 7. Unfortunately, only one combination ($p = 200, q = 5$) for the inverted concave set outperforms the original R2-HVC. This suggests that simply repeating the process of generating vectors does not improve the performance.

B. Sampling Ensemble Method

In Figure 7, we fix the total number of generated vectors to 1000. However, in the following, We will now break this constraint. Figure 8 shows CIR values of the ensemble method where the number of vectors p is fixed (i.e., 100) and the number of iterations q increases from 1 to 500 (i.e, 1,10,20,50,100,200,500). When q is larger than 100, the ensemble method performs better than the original R2-HVC with 1000 vectors. Therefore, we can conclude that the poor performance of the baseline method is due to the limited number of vectors or iterations, which is required from the time limitation.

To address the above issue, we propose an improved version named the sampling ensemble method. First, a pre-specified number of vectors (e.g., 1000 vectors) are generated in the same manner as in the original R2-HVC algorithm. The length of each vector for each solution is calculated in the same manner as in the original R2-HVC. Then, we iterate to sample a subset of vectors and select the worst solution in q times. It should be noted that the length of each vector for each solution has already been calculated. Thus the worst solution can be easily found based on the average length calculation through the sampled vectors. After iterating this procedure q times, we can use the simple majority voting to choose the final solution. The computation time of the R2-HVC mainly depends on the length calculation for each vector, while the calculation of the average length is not time-consuming. Thus the time complexity of the ensemble method is very similar to the original R2-HVC.

Figure 9 illustrates the sampling ensemble method. Initially, 1000 vectors are generated and their lengths are calculated, which takes the same computation time as the original R2-HVC in Figure 6(a). In each iteration, 100 vectors are randomly sampled from the 1000 vectors. Since the time on average length calculation is small, this method can repeat more times than the baseline ensemble method.

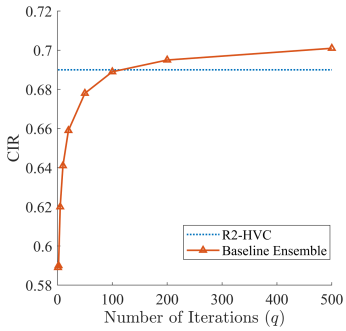


Fig. 8. The performance of the baseline ensemble method with 100 vectors for various numbers of iterations q and the performance of the original R2-HVC with 1000 vectors.

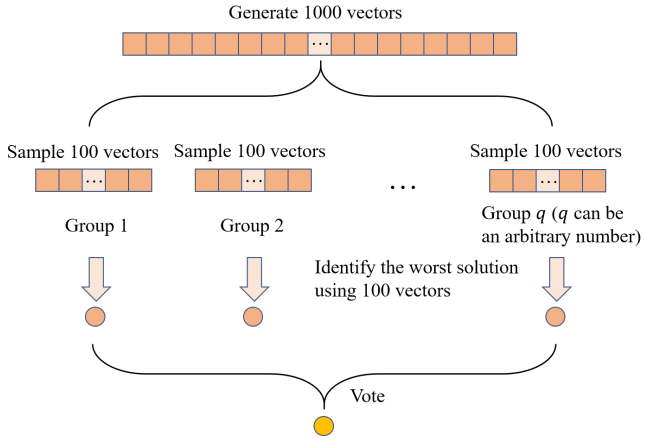


Fig. 9. A simple example to illustrate the process of the sampling ensemble method.

IV. Experiments

In this section, we perform a detailed experimental analysis of the sampling ensemble method, focusing on CIR value. We investigate the effects of the number of iterations q and the number of vectors p.

Specifically, solution sets of 10-objective problems with six different PF shapes (i.e., linear, convex, concave, linear inverted, convex inverted, concave inverted) are considered in this section. We evaluate the sampling ensemble method under various combinations of the number of iterations q and the number of vectors p. In addition to the sampling ensemble method, we also include the original R2-HVC for comparison. The total number of vectors used by the original R2-HVC is set to 1000. The same number of vectors is also used in the sampling ensemble method, where a subset of p vectors is randomly selected in each of q iterations.

The experiment is conducted on a CPU of Intel(R) Core(TM) i7-8700k CPU @ 3.70GHZ. All codes are implemented in MATLAB R2022b. It is also repeated 21 times to calculate the average result.

Since the sampling ensemble method is fast, we can try various combinations of p and q and analyze its performance in detail. We set the number of sampled vectors as $p \in \{5, 10, 20, 50, 100, 200\}$, and the number of iterations as

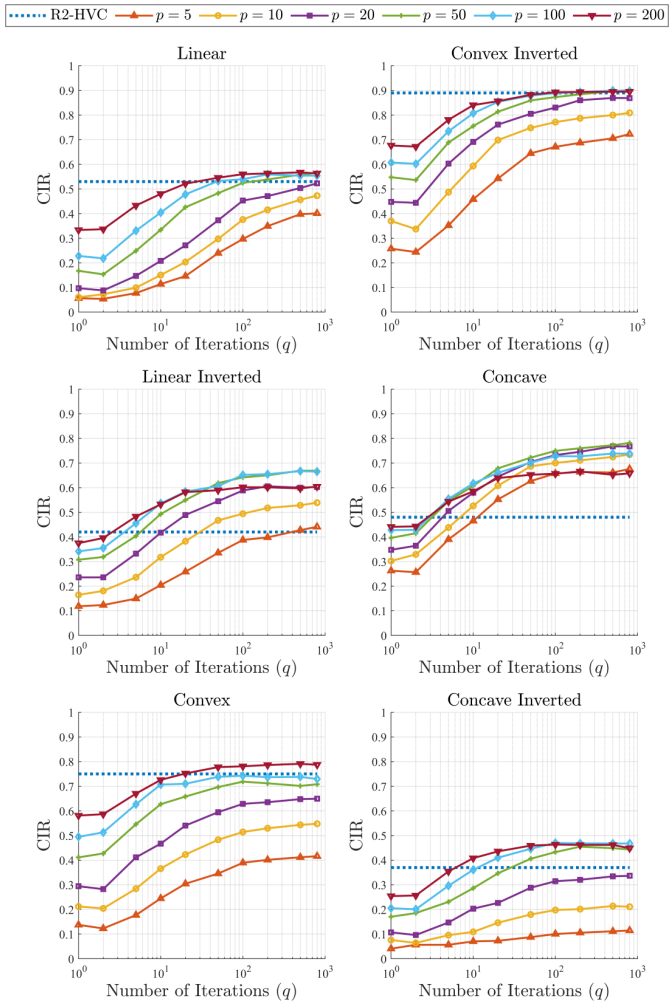


Fig. 10. CIR values of the sampling ensemble method and the original R2-HVC with different combinations of $\{p, q\}$ in the 10-objective problem. Different lines correspond to different values of p and the x-axis represents the value of q.

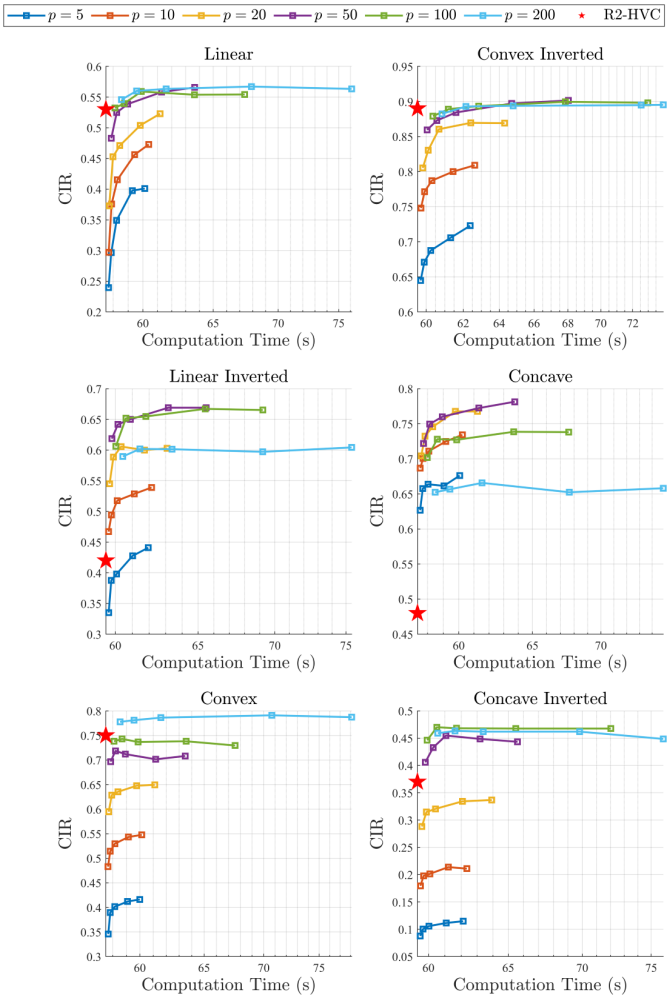


Fig. 11. CIR values and computation time of the original R2-HVC (red pentagram) and the sampling ensemble method (dots) under different combinations of p and q in the 10-objective problems.

$q \in \{1, 2, 5, 10, 20, 50, 100, 200, 500, 800\}$. In each iteration, p vectors are sampled randomly without replacement. Thus, there are no duplications among the sampled p vectors.

Figure 10 shows CIR values of the sampling ensemble method and the original R2-HVC. The x-axis denotes the value of q, and the y-axis is CIR value. Points with the same color on each connected line show the CIR values obtained from the same value of p. The results of the R2-HVC are shown in dotted blue. From Figure 10, we can observe that when the number of iterations q is 1 (same as the original R2-HVC with a smaller number of vectors), the proposed method is always inferior to the original R2-HVC. As q increases from 1 to 100, the CIR values of all parameter combinations improve. In general, several parameter combinations of the sampling ensemble method outperform the original R2-HVC in Figure 10. The computation time of the proposed method with $p = 200$ and $q = 1000$ (the largest specification in Figure 10) is about 1.5 times longer than the original R2-HVC (see Figure 11).

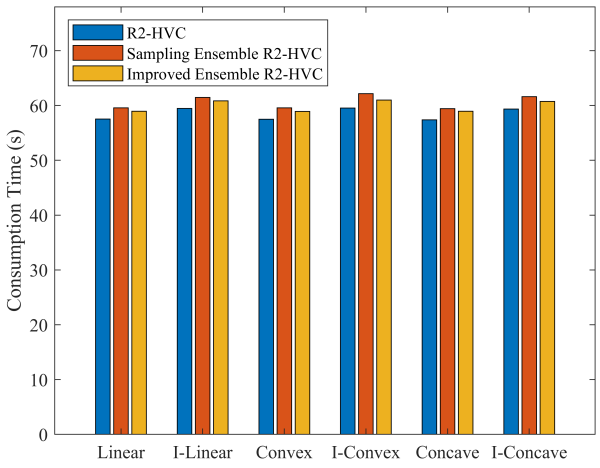
The CIR values and computation time of the sampling ensemble method for each combination of $\{p, q\}$ are shown in Figure 11. Results from different values of p are shown by different colors. Each connected line has 5 points corresponding to different specifications of $q \in \{50, 100, 200, 500, 800\}$. When q ranges from 50 to 100, the CIR values increase while the time consumption remains stable. However, when q is larger than 100, the computation time significantly increases while the CIR value remains unchanged in some cases (especially when $p = 100$ and $p = 200$). The original R2-HVC is illustrated by a red pentagram. Since the R2-HVC method does not involve sampling and voting, it consumes the least time among all the points.

For a decision-maker, selecting an appropriate combination of parameters to achieve higher accuracy and lower time consumption is crucial. From Figure 11, we recommend choosing the points that are vertically above the red pentagram. Specifically, for the concave, inverted linear, and inverted concave

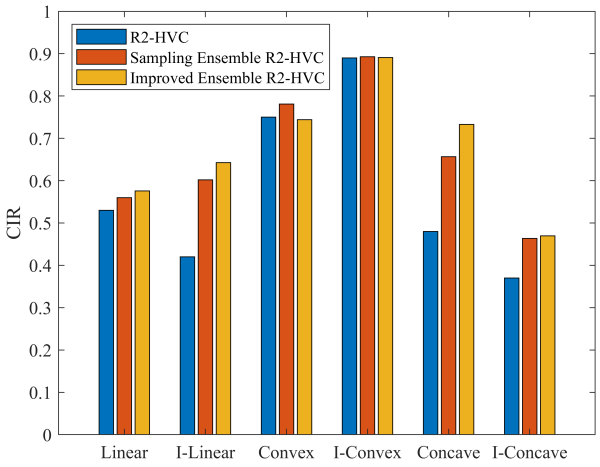
cases, selecting the green or blue points ($q = 50$ or $q = 100$) in Figure 11 can achieve better results than the original R2-HVC method with almost the same time consumption. For the other three cases, a slightly longer time (approximately 3 seconds) is needed to obtain similar or better results than the original R2-HVC method. In the inverted linear and concave cases in Figure 11, it can be observed that using 200 vectors is less effective than using 50 vectors. This suggests that each vector plays a different role in estimating HVC, and some may even have negative effects.

V. Further Enhancement on the Proposed Method

In the above experiments for the sampling ensemble method, we find that many combinations of parameters $\{p, q\}$ can exceed the performance of the original R2-HVC. Therefore, in this section, we continue to investigate whether the sampling ensemble method can be further improved. The enhanced approach has two aspects: changing the number of vectors in each iteration, and assigning a different weight to the vote (i.e., use of the weighted voting).



(a) Computation Time



(b) CIR

Fig. 12. Performance comparison between the improved version, the sampling ensemble method, and the original R2-HVC on a 10-objective problem.

In the previous experiment, the same number of vectors were sampled in each iteration, and the worst solution is selected. To increase the diversity, we change the number of sampled vectors p_i from a set of $\{5, 10, 20, 50, 100, 200\}$ in each iteration. For example, $\{p_1 = 100, p_2 = 20\}$ means we will randomly sample 100 vectors in the first iteration and 20 vectors in the second iteration. And this value p_i is randomly chosen from the pre-defined set $\{5, 10, 20, 50, 100, 200\}$.

The previous experiment only focuses on the worst solution in each iteration. Moreover, only the worst solution in an iteration could be recorded for voting. However, as discussed in Section II-D, the R2-HVC can be easily confused when two solutions have very similar HVC values. Therefore, the correct solution may be the second or third worst solution in a calculation, and the previous voting schema lost some valuable information. To improve the method, we can assign weights to a few worst solutions in each iteration. For instance, we set the weights $[5, 4, 3, 2, 1]$ to the top 5 worst solutions. Then we store these 5 solutions with their accumulated weights. Finally, in the voting session, the solution with the highest accumulated weight is selected.

Figure 12 compares the improved version with the R2-HVC and the sampling ensemble method on the 10-objective set. In the sampling ensemble method, we select a relatively good parameter combination ($p = 200, q = 100$) according to Figure 10. And in the improved method, we fix the number of iterations $q = 800$.

As it can be seen from Figure 12, even we repeat the computation 800 times in the sampling ensemble R2-HVC, the time is nearly the same as the R2-HVC. Due to the effect of the ensemble, the overall performance of the two ensemble methods is better than the R2-HVC in all solution sets. In addition to the convex case, the improved version performs better than the sampling ensemble method and its computation is faster.

VI. Conclusions

In this paper, we proposed an ensemble-based R2-HVC method for selecting the worst solution from a solution set. In comparison to the original R2-HVC, the ensemble method aims to reduce the probability of errors by performing repetitive calculations and utilizing majority voting. However, since repetitive calculations consume additional time, we used fewer vectors in this step. We focused on improving the process of sampling and voting and recommended the best parameter combination (i.e., the number of vectors p and the number of iterations q) for decision-makers. The experimental results indicated that our ensemble method outperforms the original R2-HVC.

In the future, we hope to optimize our ensemble method further by assigning dynamic weights to each vector and performing put-back-free sampling. This will make it easier to choose the best parameter combination, leading to more interesting results. We believe that such improvements to the R2-HVC can enhance its performance in various applications.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] E. Zitzler, S. Künzli *et al.*, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature - PPSN VIII*, vol. 4. Springer, 2004, pp. 832–842.

[5] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[6] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[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] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[9] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *Parallel Problem Solving from Nature — PPSN V*. Springer, 1998, pp. 292–301.

[10] C. A. Coello Coello and M. Reyes Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *MICAI 2004: Advances in Artificial Intelligence: Third Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, April 26-30, 2004. Proceedings 3*. Springer, 2004, pp. 688–697.

[11] 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.

[12] K. Bringmann and T. Friedrich, "Approximating the least hypervolume contributor: NP-hard in general, but fast in practice," *Theoretical Computer Science*, vol. 425, pp. 104–116, 2012.

[13] 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: Proceedings of the 19th International Conference on Multiple Criteria Decision Making, Auckland, New Zealand, 7th-12th January 2008*. Springer, 2010, pp. 313–326.

[14] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima, "Hypervolume approximation using achievement scalarizing functions for evolutionary many-objective optimization," in *2009 IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 530–537.

[15] K. Shang, H. Ishibuchi, and X. Ni, "R2-based hypervolume contribution approximation," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 185–192, 2020.

[16] K. Shang and H. Ishibuchi, "A new hypervolume-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 839–852, 2020.

[17] Y. Nan, K. Shang, and H. Ishibuchi, "What is a good direction vector set for the R2-based hypervolume contribution approximation," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 2020, pp. 524–532.

[18] J. Deng and Q. Zhang, "Approximating hypervolume and hypervolume contributions using polar coordinate," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 913–918, 2019.