A Performance Indicator for Interactive Evolutionary Multiobjective Optimization Methods: Supplementary materials

Pouya Aghaei Pour[1,*], Sunith Bandaru[2], Bekir Afsar[1], Michael Emmerich[3], Kaisa Miettinen[1]
[1]University of Jyvaskyla, Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland
[2]School of Engineering Science, University of Skövde, Box 408, 541 28 Skövde, Sweden
[3]Leiden Institute of Advanced Computer Science, Faculty of Science,
Niels Bohrweg 1, 2333CA Leiden, The Netherlands
[*]Corresponding author: Pouya Aghaei Pour, poaghaei@jyu.fi

1 Indicators for A Priori Methods

In this section, we provide a summary of indicators designed for a priori methods since these methods share some characteristics with interactive methods. Note that they incorporate the DM's preferences in the form of a *reference point* $\hat{z}$, which consists of the DM's desired objective values called aspiration levels $\hat{z}_i$ $(i = 1, \ldots, k)$.

The indicator proposed in [14] measures the convergence and diversity of a solution set generated by an a priori method. However, it relies on the knowledge of the Pareto front, which is usually not known when dealing with real-world problems. In [9], an indicator called *UPCF* is proposed to measure the diversity and convergence of a set of solutions without the knowledge of the Pareto front. Here, the basic idea is to use a parameter to define the desired region around the DM's preferences. Then the hypervolume indicator [13] or inverted generational distance (IGD) [2] of solutions that are within this region is calculated. However, the main issue is that for any set of solutions outside of the desired region, the performance assessment of UPCF is zero. Therefore, we cannot make a difference between the UPCF values of such solution sets even though they are different.

The so-called *R-metric* [8] is proposed to overcome the above-mentioned shortcoming of UPCF. First, we have to identify the "pivot point". The pivot point (the green dot in Figure 1) is the solution with the lowest ASF value. Then, Solutions are transferred into a "virtual position" as illustrated in Figure 1 (denoted as rectangles). The point of this transformation is to reposition the solutions so that at least one solution from each set would be in the desired region. Then, similar to [9], the desired region is identified (the blue rectangle in

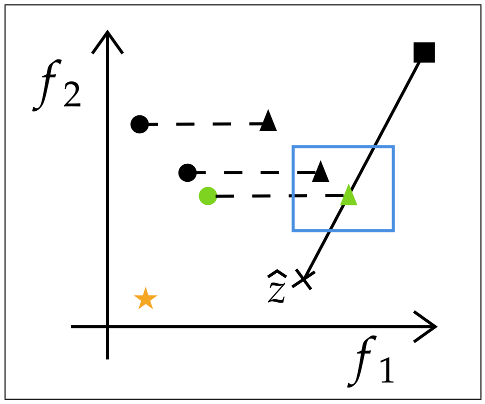


Figure 1: Simple example of how we can calculate R-metric. The solutions are shown as circles, the transferred solutions are denoted as triangles, the nadir point is shown as a black rectangle, and the desired region is denoted as a blue rectangle.

Figure 1), and an indicator developed for a posteriori methods (e.g., hypervolume indicator) is used to assess the solutions that are inside the desired region. Because of the transformation of solutions, the desired region is never empty (unlike UPCF); therefore, the issue of an indicator taking a value of zero does not occur. However, neither UPCF nor R-metric considers solutions outside of the desired region in their performance assessment, which can be misleading since those solutions also influence the DM.

Distance-based indicators proposed in [6, 12] penalize the performance of a priori methods based on how far the solutions are from the desired region. In PMDA [12], the desired region is identified based on a reference point and a parameter that determines the size of the desired region. Then, the mean of an angle-penalized function [3] is calculated for all the solutions. For a set of solutions P, the assessment of PMDA can be expressed as:

$$PMDA(P) = \frac{\sum_{i=n}^{n}(D_i + \theta_{x_i}/\pi)}{n}, \tag{1}$$

where n is the number of solutions in P, and *theta* is the penalty function for solutions outside the desired region.

An indicator called PMOD [6] was proposed, which uses three distance measures to assess the performance of a priori methods. Similar to R-metric, PMOD also transfers all of the solutions to a virtual point (black triangles in Figure 2) which are referred to as "mapping points". However, instead of using an ASF function, in PMOD, a hyperplane (illustrated as the black line in Figure 2) is constructed based on the position of the reference point (denoted as a black cross). Then, based on a parameter r that indicates a radius neighborhood of the reference point, the desired region is identified (black circle around the reference point). Next, we calculate the following distances:

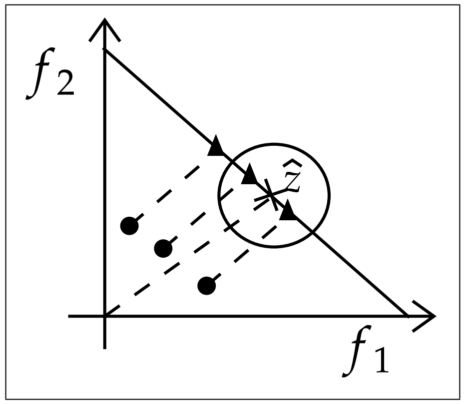


Figure 2: Simple example of how we can calculate PMOD. The solutions are shown as circles. the hyperplane

1. D_1: distance between the mapping points and the reference point;
2. D_2 standard deviation of the distance between each mapping point and the nearest mapping point;
3. $D3$ distance between the mapping points and the origin.

Finally, We can express PMOD as:

$$PMOD = \frac{\sum_1^n (D1 + kD3}{n} + D2, \tag{2}$$

where k is a penalty coefficient for mapping points outside the desired region. Note that

$$\begin{cases} k = 1 & if r < \text{D1} \\ k > 1 & otherwise. \end{cases}$$

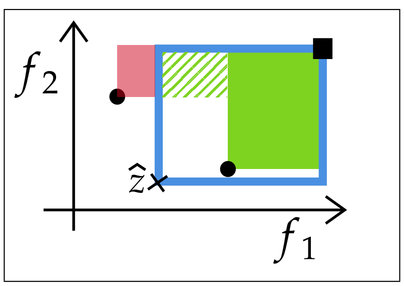
All the indicators mentioned so far rely on one or several parameters, like the size of the desired region or a penalty coefficient. This has two main issues. First, these parameters' role in the final performance assessment has not been studied. Second, setting these parameter values is challenging due to their unintuitive nature. Therefore, a parameterless indicator called EH-metric is proposed in [1]. It uses the concept of an *expanding hypercube*, which starts as a point at the reference point and expands (with the reference point at its center) until it envelops all solutions. The EH-metric value for an a priori method is calculated as the area under the curve generated by plotting the fraction of solutions enveloped by the hypercube as it expands versus the size of the hypercube. The EH-metric can measure convergence and diversity without relying on the knowledge of the Pareto front.

2 Analyzing the components of PHI

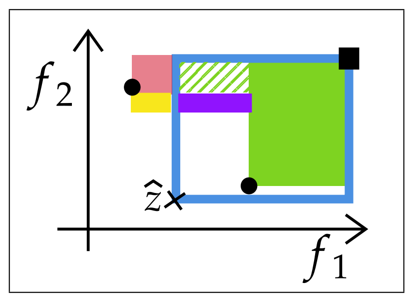
In this section, we go through analyzing different components of PHI. We briefly discuss the role of dystopian point in performance assessment, then we analyze how solutions inside and outside the desired region can affect the performance assessment.

2.1 Relations to $v^{\prec}$ and v^{-}

As we mentioned in the main paper, the assessment of PHI can be improved by either reducing the size of v^{+} or increasing the size of v^{-}. Here we study the effects of $v^{\prec}$ and v^{-} further. Figure 3a illustrates when the density of solutions is high. In this situation, we can observe that most of the contribution to $v^{\prec}$ comes from the solutions within the desired region. Meaning that if an interactive method can do a decent job in finding solutions that reflect the DM's preferences, the effect of solutions in $v^{\prec}$ diminishes. However, if the solutions have some distance from the boundaries of the desired region (see Figure 3b, then the solutions in v^{-} have a more significant role in covering the desired region.



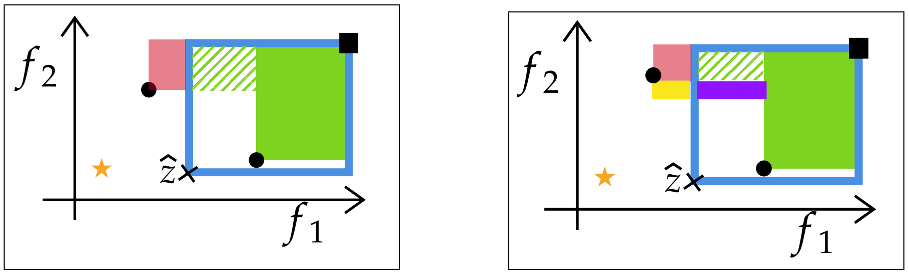
(a) High density solutions



(b) low density solutions

Figure 3: Demonstrating how solutions can contribute to $v^{\prec}$ and v^{-}. In (a) mostly the solutions inside the desired region contribute to the calculation of v^{-}, and in (b) because the generated solutions have some distance to the boundaries of the desired region, the contribution of solutions outside of the desired region has increased (the green dashed area).

Moreover, as one of the drawbacks of PHI, we mentioned that the performance assessment of PHI may increase or decrease when the volume of v^{-} decreases. In Figure 4, we provide a simple example to elaborate more. Here we can observe that when we move the solution outside of the desired region in Figure 4a a little up (see Figure 4b) the volume of $v^{\prec}$ that is decreased in $v^{\prec}$ (the purple rectangle) is more than the volume that is decreased in v^{-} (the yellow rectangle). This means that even though we reduced the volume of v^{-} and did not make any changes to the solution inside the desired region, the overall value of PHI decreased.

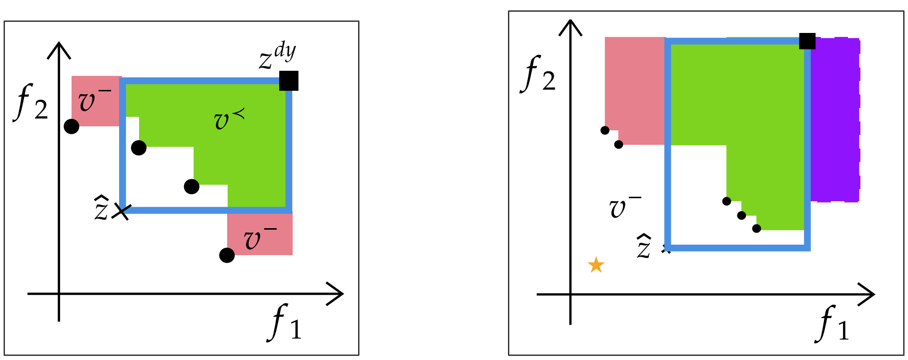


(a) (b)

Figure 4: An example where we reduce the volume of v^- but the performance worsens. We move the solution outside the desired region in (a) toward the f_2 axis. The green dashed area is the solution's contribution outside the desired region (blue rectangle) to calculate $v^{\prec}$ (green area). In (b), the purple rectangle illustrates the amount of coverage area we lose in $v^{\prec}$, and the yellow rectangle illustrates the amount of coverage we lose in v^- (the red area).

2.2 Relations to the Dystopian Point

There has been much research on how to set the dystopian point when assessing the performance of a posteriori methods [7]. However, to the best of our knowledge, no study has been explicitly dedicated to setting this point when the DM's preferences are involved. Even though this is not the main objective of this paper, here we briefly explain how the dystopian point can affect the value of PHI.



(a) (b)

Figure 5: An example where changing the dystopian point position only affects the volume of $v^{\prec}$. The green and red areas represent the $v^{\prec}$, v^-, respectively. The desired region is represented as a blue rectangle. The difference between the volume of $v^{\prec}$ in (a) and (b) is demonstrated as a purple rectangle.

We know the size of the desired region depends on the dystopian point and the reference point. However, in Figure 5, we have demonstrated that the dystopian point may also affect $v^{\prec}$, but v^{-} can stay unaffected. We have changed the position of the dystopian point from Figure 5a to Figure 5b. In Figure 5b we can observe that the size of $v^{\prec}$ (the green areas) has decreased but the size of v^{-} (the read areas in Figure 5a and 5b) is the same. Moreover, according to [7, 10], if the reference point is too far away from the solutions, the solutions close to the axis will contribute the most to the hypervolume calculation. PHI also inherits this issue from the hypervolume, and we should consider this when setting this point for PHI.

3 Numerical Results

Here we have used the engineering problems mentioned in [11] and some of DTLZ [4] problems to provide further results with our proposed indicator. For the engineering problems, we also analyze the v^{+} and v^{-} values to understand how PHI declares any of the methods as the winner. For the benchmark problems, we provide statistical analysis to show the consistency of PHI in identifying the best method.

The parameters of the algorithms and indicators are the same as in the main paper. We also have provided the results of other indicators for assessing the learning and decision phases here. However, we primarily focus on how we can use PHI to determine the best method since the other indicators are not designed for interactive methods.

3.1 Engineering problems

We have selected three engineering problems. The number of objectives and decision variables can be found in table 1. After showing the overall assessment of these three problems, we analyze the results v^{-} and v^{+} to understand better how the methods behave.

Table 1: Number of objectives (k) and decision variables (n) for the engineering problems.

Problem	k	n
RE34	3	4
RE37	3	7
RE9‘	9	7

3.1.1 Learning phase

Table 2 illustrates the results of the learning phase for these three engineering problems. Based on PHI, iRVEA has better performance for all three problems than IOPIS.

Table 2: Learning phase for the engineering problems

		PHI	R-metric	EH	PMOD	PMDA
RE34	IOPIS	520.26	4192.21	**318.89**	**13223.48**	**215.33**
	iRVEA	**566.46**	**6051.20**	290.67	17525.98	225.53
RE37	IOPIS	221.43	2825.09	165.00	**7751.71**	578.63
	iRVEA	**328.44**	**6650.99**	**457.88**	12124.32	**527.15**
RE91	IOPIS	901.95	**2011815.94**	211.04	6006.82	**1287.37**
	iRVEA	**903.56**	1540064.33	**287.63**	**8834.96**	1479.90

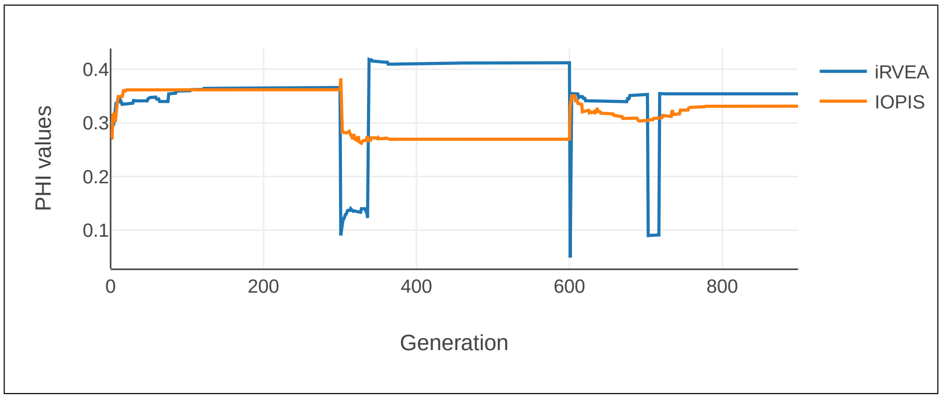


Figure 6: Tracking the values of PHI for iRVEA and IOPIS over the learning phase for RE34.

In Figure 6, we illustrate how the value of PHI changes over the generations for problem RE34. We can observe that iRVEA has almost the same performance as IOPIS during the first interaction, and during the second and third interactions, iRVEA performs better than IOPIS.

Moreover, as discussed in Section 3, PHI can provide additional information on its final performance. Figure 7 shows the v^+ values during the learning phase. During the first and third interactions, IOPIS had slightly better values than iRVEA. However, during the second interaction, iRVEA has noticeably better values in v^+ than IOPIS. So far, this analysis means that, except for the second interaction, both methods almost had similar diversity and convergence within the desired region.

Additionally, we can analyze the v^- values to understand the distribution of solutions outside the desired region. Figure 8 illustrates the v^- values for this problem. Here, we can see that IOPIS has lower v^- values than iRVEA for most of the generations. However, as we showed earlier, for the final performance assessment, iRVEA was the winner. This means that iRVEA had some solutions outside the desired region. However, the positive hypervolume contribution of these solutions to the desired region was higher than their negative hypervolume contribution. This is why iRVEA has been declared the winner by PHI.

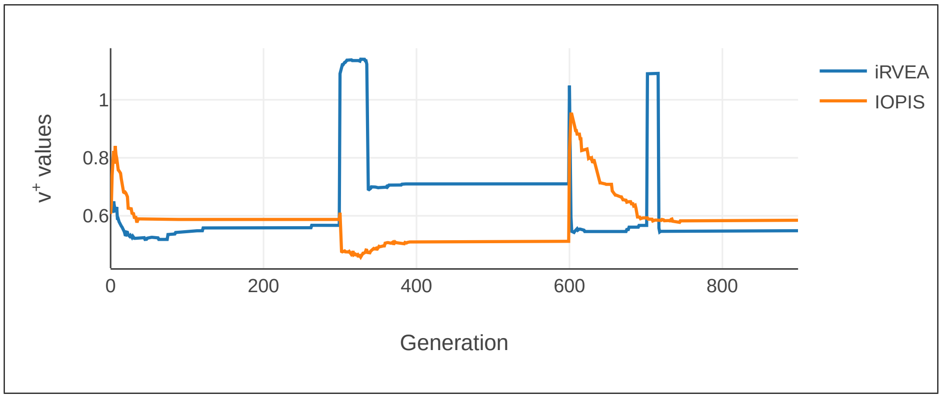


Figure 7: Tracking the values of v^+ for iRVEA and IOPIS over the learning phase for RE34.

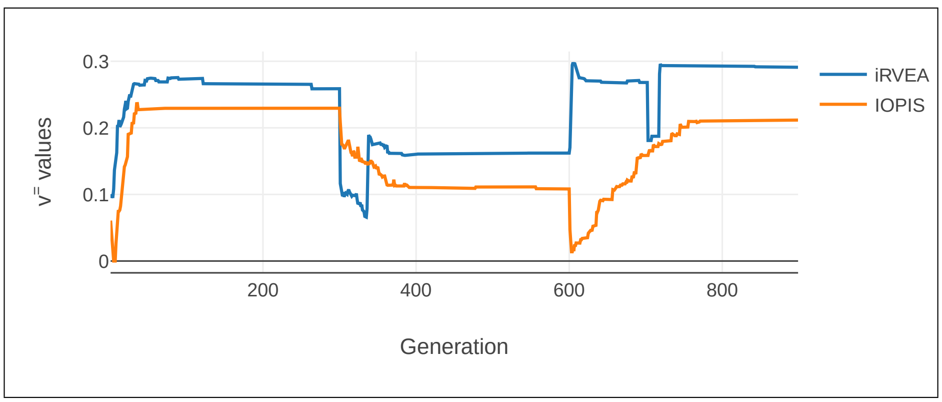


Figure 8: Tracking the values of v^- for iRVEA and IOPIS over the learning phase for RE34.

Next, we analyze the results of these methods for the learning phase of RE37. Figure 9 illustrates the values of PHI over the generations in the learning phase. Besides the second interaction, iRVEA and IOPIS have competitive results for the learning phase. Figure 10 shows the v^+ values for this problem during the learning phase. Here, we can observe that for most generations, iRVEA has had a better performance than IOPIS. Moreover, the v^+ values for IOPIS are quite low (almost zero). This means that the solutions that IOPIS has generated are far from the reference point and quite close to the dystopian point that makes the v^+ values small. Additionally, we have provided the values of v^- in Figure 11. Here we can observe that forever generation IOPIS has a better value for v^-. However, because of the low values of v^+ for IOPIS, we should declare iRVEA as the winner since it has better overall performance and the $v+$ values are much better than IOPIS.

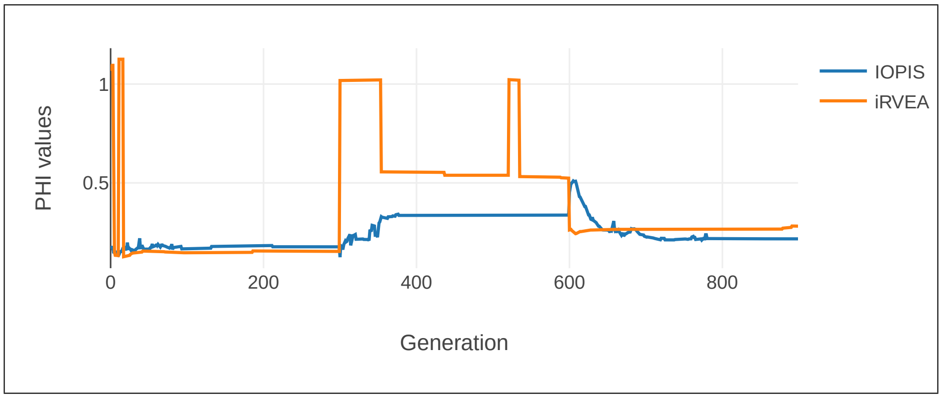


Figure 9: Tracking the values of PHI for iRVEA and IOPIS over the learning phase for RE37.

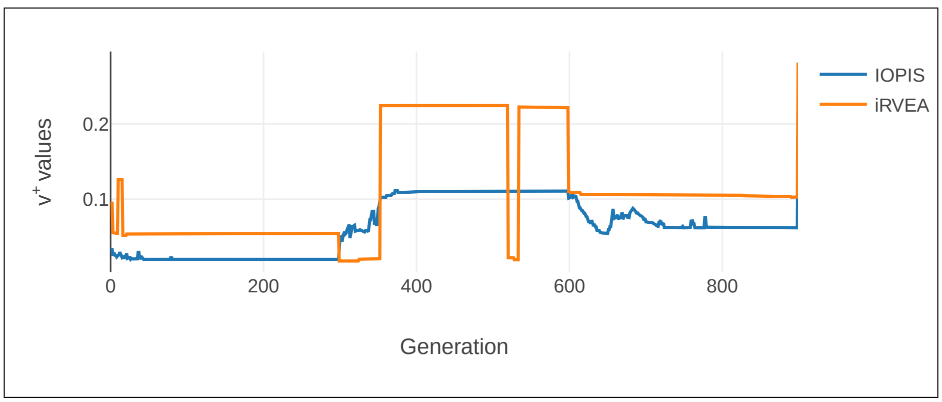


Figure 10: Tracking the values of v^+ for iRVEA and IOPIS over the learning phase for RE37.

Finally, we discuss the results of RE91. Figure 12 shows the PHI values during the learning phase. Even though iRVEA has better performance for most of the learning phase, you can see that the difference is not very large. In fact, their performance is different in the third decibel of the performance value. Therefore we have provided the v^+ and v^- values in Figures 13 and 14 to investigate the performance of iRVEA and IOPIS further.

In Figure 13, we can observe that the difference between v^+ values of the two interactive methods is more noticeable. Here for the majority of the learning phase, iRVEA has a better performance. Additionally, in Figure 14, we can observe that the difference between v^- values is relatively small. Hence, we can declare iRVEA as the winner of the learning phase for this problem with more confidence.

Thus far, we have analyzed the performance of the learning phase for engi-

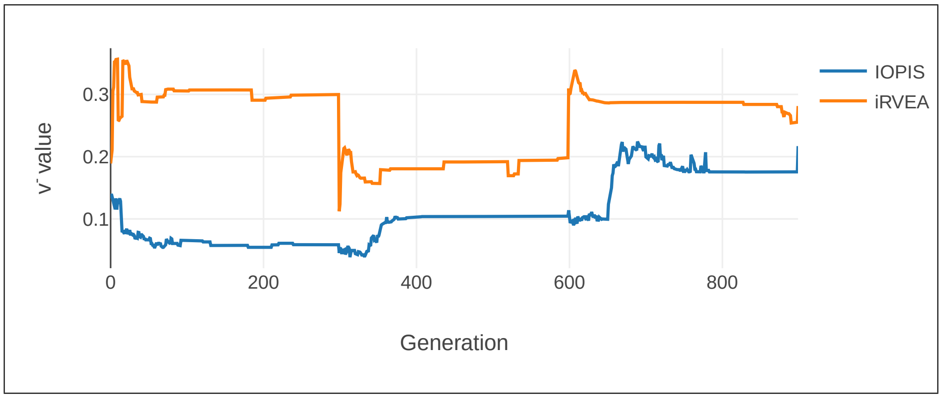


Figure 11: Tracking the values of v^- for iRVEA and IOPIS over the learning phase for RE37.

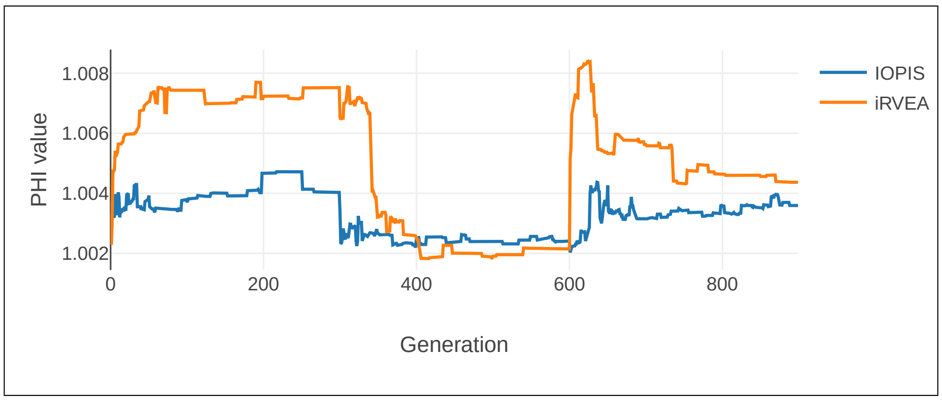


Figure 12: Tracking the values of PHI for iRVEA and IOPIS over the learning phase for RE91.

neering problems. Next, we take a look at their performance during the decision phase.

3.1.2 Decision phase

Table 3 shows the performance assessment of the decision phase for the engineering problems. For RE34, we can observe that PHI declares a draw between these two methods. However, we can investigate the values of PHI further to understand which method is more suitable for the decision phase.

In Table 4, we can observe that IOPIS only generates solutions that are inside the desired region ($v^- = 0$). On the other hand, v^- values for iRVEA indicate that some of the solutions are slightly outside of the desired region. Considering v^+ values and v^-, we can see that iRVEA is the winner for the

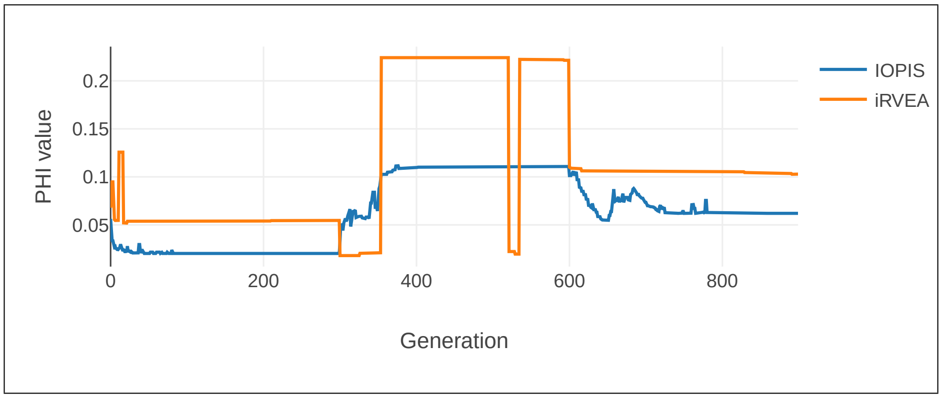


Figure 13: Tracking the values of v^+ for iRVEA and IOPIS over the learning phase for RE91.

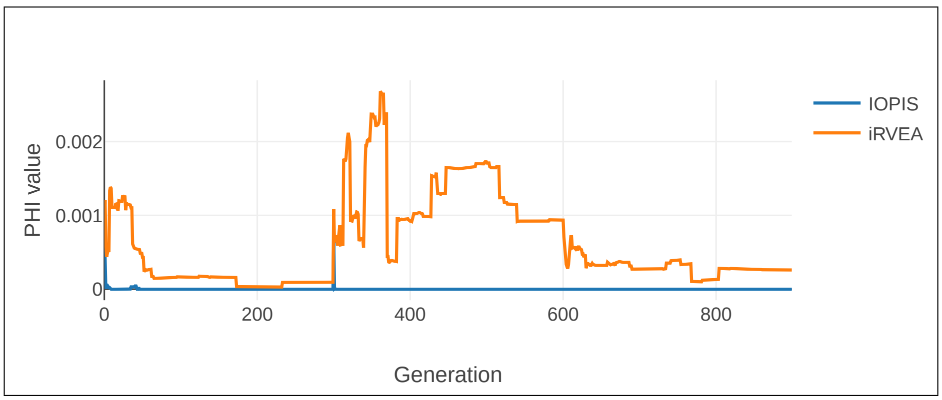


Figure 14: Tracking the values of v^- for iRVEA and IOPIS over the learning phase for RE91.

Table 3: Decision phase for the engineering problems

		PHI	R-metric	EH	PMOD	PMDA
RE34	IOPIS	0.8	4.06	**0.61**	54.63	**0.10**
	iRVEA	0.8	**5.38**	0.42	**41.51**	0.12
RE37	IOPIS	**0.67**	6.28	**0.73**	**23.88**	**0.44**
	iRVEA	0.63	**6.83**	0.65	27.56	0.45
RE91	IOPIS	0.002	1.936	**0.179**	4.578	**1.252**
	iRVEA	**0.007**	**3.345**	0.089	**3.853**	1.253

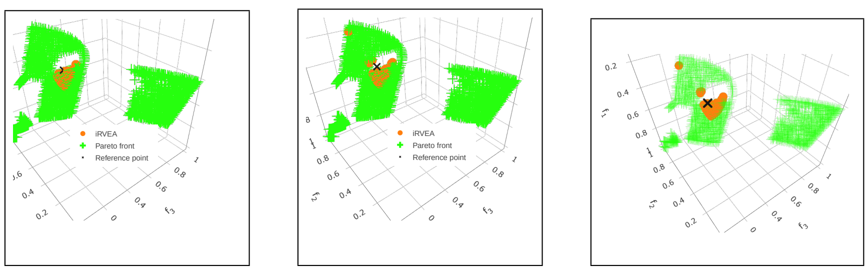
decision phase.

Moreover, Figures 16 and 15 illustrate the solutions IOPIS and iRVEA have generated during the decision phase. In these figures, we can observe that

Table 4: The values of v^+ and v^- for the three interactions during the decision phase.

Interaction	Measure	IOPIS	iRVEA
4	v^+	0.45	**0.46**
	v^-	**0.00**	0.15
5	v^+	0.46	**0.56**
	v^-	**0.00**	0.05
6	v^+	0.46	**0.55**
	v^-	**0.00**	0.05

iRVEA solutions have more diversity than solutions generated by IOPIS. This could be why some of the solutions are outside of the desired region and hence, the higher values of v^- of iRVEA. On the other hand, iRVEA solutions are too similar that we cannot detect any diversity between them in the Figure 16. This means that the DM has no flexibility to fine-tune solutions. Therefore, we can declare iRVEA as the winner for this problem even though the final performance assessments of both methods are quite similar.

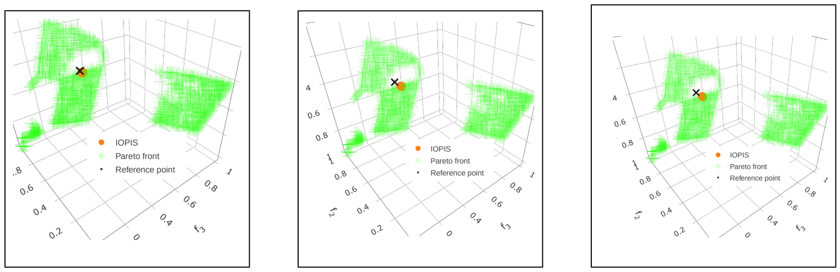


(a) Interaction 1. (b) Interaction 2. (c) Interaction 3.

Figure 15: iRVEA solutions during the decision phase. The green area is the Pareto front of RE34, the orange dots are the generated solutions by iRVEA, and the black cross is the reference point generated by ADM.

Next, we provide the decision phase results for the problem RE37 in Table 5. We can observe that IOPIS can stick to the inside of the desired region ($v^- = 0$), but iRVEA has some solutions outside the desired region. Because of this, the performance of IOPIS is slightly better than iRVEA (see Table 3) for the decision phase. For RE91, we can see in Table 3 that both IOPIS and iRVEA have relatively low performance for the decision phase (iRVEA is slightly better than IOPIS). This could be because they needed more function evaluations to converge toward the Pareto front. Moreover, we can analyze the results further to choose the best method based on this information.

Table 6 shows the v^+ and v^- values for problem RE91 that has been obtained by IOPIS and iRVEA. We can observe that Both methods have generated solutions inside the desired region ($v^- = 0$). However, the values of v^+ for both



(a) Interaction 1. (b) Interaction 2. (c) Interaction 3.

Figure 16: IOPIS solutions during the decision phase. The green area is the Pareto front of RE34, the orange dots are the generated solutions by IOPIS, and the black cross is the reference point generated by ADM.

Table 5: The values of v^+ and v^- for the decision phase of problem RE37.

Interaction	Measure	IOPIS	iRVEA
4	v^+	0.23	**0.30**
	v^-	**0**	0.11
5	v^+	0.22	**0.28**
	v^-	**0**	0.13
6	v^+	0.23	**0.30**
	v^-	**0**	0.11

methods are quite small. This means that these methods cannot converge toward the Pareto front during the decision phase, and they both need more function evaluations (or generations).

Table 6: The values of v^+ and v^- for the decision phase of problem RE91.

Interaction	Measure	IOPIS	iRVEA
4	v^+	0.001	**0.005**
	v^-	0	0
5	v^+	0.001	**0.004**
	v^-	0	0
6	v^+	0.001	**0.004**
	v^-	0	0

In the following subsection, we provide the results of performance assessments of benchmark problems and some statistical analysis.

Table 7: Learning phase for three objectives DTLZ1-4 problems

		PHI	R-metric	EH	PMOD	PMDA
DTLZ1	IOPIS	**1462.98**	3803.60	272.26	20467.95	131896.92
	iRVEA	977.07	**6947.00**	**385.89**	**10621.50**	**7309.03**
DTLZ2	IOPIS	**252.32**	5859.92	545.31	23481.08	**599.36**
	iRVEA	131.44	**7437.63**	**561.18**	**17790.88**	611.38
DTLZ3	IOPIS	**349.9976**	3095.605	**79.24834**	29180.17	316428.8
	iRVEA	254.5509	**6929.305**	64.46008	**15489.82**	**20279.23**
DTLZ4	IOPIS	**245.53**	5772.64	512.34	23134.02	**592.82**
	iRVEA	134.08	**7168.02**	**635.50**	**18612.57**	607.59

Table 8: Decision phase for 3 objectives DTLZ1-4 problems

		PHI	R-metric	EH	PMOD	PMDA
DTLZ1	IOPIS	**1.40**	**13.95**	**0.37**	22.06	**0.05**
	iRVEA	1.16	3.09	0.09	**6.43**	0.32
DTLZ2	IOPIS	**0.99**	7.96	**0.96**	554.40	**0.60**
	iRVEA	0.45	**8.25**	0.82	**27.67**	0.62
DTLZ3	IOPIS	**0.42**	5.67	0.72	**3.65**	1.21
	iRVEA	0.39	**8.53**	**0.76**	31.58	**0.68**
DTLZ4	IOPIS	**0.98**	7.94	**0.96**	954.78	**0.56**
	iRVEA	0.48	**8.54**	0.82	**33.86**	0.59

3.2 Benchmark problems

We have chosen DTLZ1, DTLZ2, DTLZ3, and DTLZ4 with 3,5 and 7 objectives. The number of decision variables is selected as follows:

$$n = k + 9,$$

Where k is the number of objectives, for each DTLZ problem, we have run IOPIS and iRVEA 11 times, and We conducted a pairwise Wilcoxon significance test [5]. If the difference between the results is significant, they are shown in bold font in the tables.

Tables 7, 9, and 11 shows the results of the learning phase for the benchmark problems. In addition, Tables 8, 10, and 12 shows the results of the decision phase for the benchmark problems.

For all of the benchmark problems, IOPIS has a better performance than iRVEA (both in the learning and decision phases). The reason could be that all of these DTLZ problems are well behaved, and the Pareto front continues, whereas in the engineering problems, this was not the case (see, e.g., Figure 15)

Table 9: Learning phase for five objectives DTLZ1-4 problems

		PHI	R-metric	EH	PMOD	PMDA
DTLZ1	IOPIS	**1464.57**	12612.81	283.33	6734.09	5301.25
	iRVEA	178.71	**33687.89**	**279.31**	**10614.67**	**5615.34**
DTLZ2	IOPIS	**326.36**	23394.84	0	23765.96	**648.42**
	iRVEA	174.07	**28987.16**	**594.89**	**17876.01**	667.87
DTLZ3	IOPIS	**461.72**	11417.15	**65.37**	14572.37	25576.89
	iRVEA	128.36	**26148.46**	296.34	**15207.47**	**22176.66**
DTLZ4	IOPIS	**270.86**	24001.33	58.23	17469.99	**651.05**
	iRVEA	177.98	**28825.84**	**482.77**	**16450.79**	668.02

Table 10: Decision phase for 5 objectives DTLZ1-4 problems

		PHI	R-metric	EH	PMOD	PMDA
DTLZ1	IOPIS	**0.91**	35.61	**0.58**	86.74	0.04
	iRVEA	0.88	**37.72**	0.54	**69.87**	0.04
DTLZ2	IOPIS	**0.96**	30.32	**0.88**	82.10	**0.66**
	iRVEA	0.38	**31.03**	0.75	**25.66**	0.68
DTLZ3	IOPIS	**0.45**	25.20	0.43	**9.47**	0.80
	iRVEA	0.37	**31.47**	**0.70**	29.19	**0.72**
DTLZ4	IOPIS	**0.94**	30.13	**0.86**	66.15	**0.69**
	iRVEA	0.36	**31.48**	0.73	**28.91**	0.71

Table 11: Learning phase for seven objectives DTLZ1-4 problems

		PHI	R-metric	EH	PMOD	PMDA
DTLZ1	IOPIS	**899.00**	35012.14	**791.83**	26718.90	333521.15
	iRVEA	780.14	**116777.86**	21.60	**7975.17**	**5023.56**
DTLZ2	IOPIS	**587.42**	97698.63	117.30	**12001.06**	**693.38**
	iRVEA	294.72	**109714.53**	**652.33**	13577.24	714.74
DTLZ3	IOPIS	**339.87**	**63070.15**	136.75	**8719.51**	**12963.34**
	iRVEA	131.87	74986.30	**217.52**	15735.29	23953.82
DTLZ4	IOPIS	**779.0993**	93572.91	0.751906	13860.88	**665.0891**
	iRVEA	250.17	**110945.3**	**628.57**	**11907.36**	699.37

Table 12: Decision phase for 7 objectives DTLZ1-4 problems

		PHI	R-metric	EH	PMOD	PMDA
DTLZ1	IOPIS	**0.45**	97.57	**0.58**	38.24	0.19
	iRVEA	0.21	**102.10**	0.41	**15.52**	**0.07**
DTLZ2	IOPIS	0.37	91.55	**0.88**	**17.93**	**0.65**
	iRVEA	**0.51**	**98.09**	0.78	19.74	0.68
DTLZ3	IOPIS	0.44	78.70	**0.65**	**12.31**	0.66
	iRVEA	**0.42**	**89.85**	0.63	16.03	**0.59**
DTLZ4	IOPIS	**0.70**	126.01	**0.89**	147.97	**0.68**
	iRVEA	0.53	**138.32**	0.74	**18.10**	0.71

References

[1] Sunith Bandaru and Henrik Smedberg. A parameterless performance metric for reference-point based multi-objective evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 499–506. Association for Computing Machinery, 2019.

[2] Peter AN Bosman and Dirk Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188, 2003.

[3] Ran Cheng, Yaochu Jin, Markus Olhofer, and B. Sendhoff. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791, 2016.

[4] K Deb, L Thiele, M Laumanns, and E Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02*, volume 1, pages 825–830. IEEE, 2002.

[5] Wilcoxon Frank. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.

[6] Zhanglu Hou, Shengxiang Yang, Juan Zou, Jinhua Zheng, Guo Yu, and Gan Ruan. A performance indicator for reference-point-based multiobjective evolutionary optimization. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI), Proceedings*, pages 1571–1578. IEEE, 2018.

[7] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary computation*, 26(3):411–440, 2018.

[8] Ke Li, Kalyanmoy Deb, and Xin Yao. R-metric: Evaluating the performance of preference-based evolutionary multiobjective optimization using reference points. *IEEE Transactions on Evolutionary Computation*, 22(6):821–835, 2017.

[9] Asad Mohammadi, Mohammad Nabi Omidvar, and Xiaodong Li. A new performance metric for user-preference based multi-objective evolutionary algorithms. In *2013 IEEE Congress on Evolutionary Computation, Proceedings*, pages 2825–2832. IEEE, 2013.

[10] Ke Shang, Hisao Ishibuchi, Linjun He, and Lie Meng Pang. A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(1):1–20, 2020.

[11] Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020.

[12] Guo Yu, Jinhua Zheng, and Xiaodong Li. An improved performance metric for multiobjective evolutionary algorithms with user preferences. In *2015 IEEE Congress on Evolutionary Computation, Proceedings*, pages 908–915. IEEE, 2015.

[13] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International Conference on Parallel Problem Solving from Nature, Proceedings*, pages 292–301. Springer, 1998.

[14] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.