Weight Vector Grid with New Archive Update Mechanism for Multi-Objective Optimization

Xizi Ni
Southern University of Science and Technology
Shenzhen, Guangdong, China
11510602@mail.sustc.edu.cn

Hisao Ishibuchi
Southern University of Science and Technology
Shenzhen, Guangdong, China
hisao@sustc.edu.cn

Kanzhen Wan
Southern University of Science and Technology
Shenzhen, Guangdong, China
11749197@mail.sustc.edu.cn

Ke Shang
Southern University of Science and Technology
Shenzhen, Guangdong, China
kshang@foxmail.com

Chukun Zhuang
Southern University of Science and Technology
Shenzhen, Guangdong, China
11749278@mail.sustc.edu.cn

ABSTRACT

Currently, most of the decomposition-based multi-objective evolutionary algorithms (MOEA) are based on a number of prespecified weight vectors. However, when the shape of the Pareto front is inconsistent with the distribution of weight vectors, only a small number of non-dominated solutions can be obtained inside the Pareto front. Moreover, if an external archive with a dominance-based update mechanism is used to overcome this difficulty, a large computational time is needed which is often unpractical. In this paper, we propose a new archive update mechanism with a new archive structure. A large weight vector grid is used to update the archive by using a scalarizing function. The proposed archive update mechanism can be applied to any MOEA with an external archive. We examine the effectiveness of the proposed mechanism on MOEA/D. Our experimental results show that MOEA/D with the proposed new archive update mechanism is able to find more solutions inside the Pareto front compared to MOEA/D without the archive. In addition, it needs less computational time compared to MOEA/D with the dominance-based archive update mechanism.

CCS CONCEPTS

• **Mathematics of computing → Evolutionary algorithms**;

KEYWORDS

Evolutionary algorithm, multiobjective optimization, MOEA/D, archive update mechanism

ACM Reference Format:
Xizi Ni, Hisao Ishibuchi, Kanzhen Wan, Ke Shang, and Chukun Zhuang. 2018. Weight Vector Grid with New Archive Update Mechanism for Multi-Objective Optimization. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3205651.3208225

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

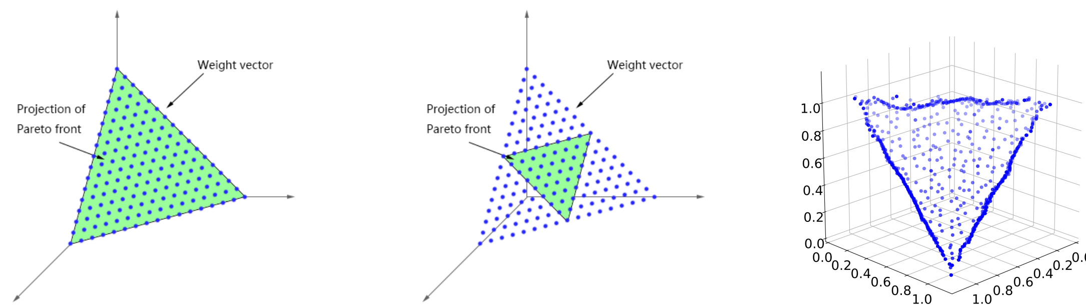
ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208225

1 INTRODUCTION

MOEA/D [12] is a representative decomposition-based algorithm. In MOEA/D, the multi-objective problem is decomposed into a series of single-objective subproblems and each solution is associated with a subproblem. If a set of weight vectors is properly defined, MOEA/D is able to obtain a set of non-dominated solutions which are uniformly distributed across the entire Pareto Front (PF). As discussed in [9], the weight vector grid in MOEA/D is a nice way to maintain the diversity of the population. Most of the decomposition-based MOEAs are using the same or similar mechanism [7, 8, 10, 12].

However, a crucial issue of using the weight vector grid is that when the shape of the PF is inconsistent with the shape of the weight vectors, only a small portion of solutions are obtained inside the PF [5–7]. Fig. 1 shows the relation between the weight vectors and the PF as well as the result of MOEA/D for a three-dimension inverted DTLZ1 (I-DTLZ1) problem [8]. The shape of PF in Fig. 1(a) is the same as the distribution of weight vectors. However, they are different in Fig. 1(b). When MOEA/D is applied to a three-dimension DTLZ1 problem [2], the obtained solutions are uniformly distributed. However, Fig. 1(c) shows an example of the obtained nondominated solutions by MOEA/D for a three-dimension I-DTLZ1 problem. The PF shape of the inverted DTLZ1 is inverted triangular, which is inconsistent with the distribution of the weight vectors. It shows that for the inverted DTLZ1, a lot of the obtained solutions are on the boundary of the PF, and only a few solutions are inside the PF. The reason for this phenomenon is that when the shape of the PF is different from the distribution of the weight vectors, MOEA/D will push the solutions associated with the weight vectors out of the shaded region (Fig. 1(b)) to the boundary of the PF [4].

The external archive is needed to overcome the above issue. Here some external archive update mechanisms have been proposed. MODE [3] is a dominated based archive, the size of the archive is controlled by the crowding distance. JADE [11] stores the recently explored inferior solutions, whose difference from the current population is utilized as the directions of the searching process.



(a) Consistent distributions (b) Inconsistent distributions (c) Inverted DTLZ1

Figure 1: Relation between a set of weight vector grid and the PF and a result in I-DTLZ1

However, the majority of external archives are based on Pareto dominance. In this paper, a new external archive update mechanism with a weight vector grid structure is proposed. In our experiment, we examine the effectiveness of the proposed mechanism on MOEA/D. Our experimental results show that the proposed new weight vector based external archive update mechanism is able to find more solutions inside the PF compared with MOEA/D without an archive and its computational time is less than MOEA/D with dominance-based archives.

The remainder of this paper is organized as follows. Section 2 presents the proposed new weight vector based external archive update mechanism. Numerical results are given in Section 3. Section 4 will discuss the further study. Finally, Section 5 concludes this paper.

2 PROPOSED EXTERNAL ARCHIVE UPDATE MECHANISM

2.1 Basic Idea of New External Archive Update Mechanism

In order to overcome the issue described in Section 1, a simple method is to use an external archive with a dominance-based update mechanism [12], which stores all the non-dominated solutions. When a new solution is generated, it removes the solutions which are dominated by the new generated one and add the new generated solution to the archive. However, as the population size increases, the additional computation load for updating the external archive increases exponentially.

Our idea is to use an external archive with a large weight vector grid structure. When a new solution is generated, it is assigned to the closest weight vector in the weight vector grid. Then the new solution is compared with the current solution of the assigned weight vector using a scalarizing function. If the new solution has a better scalarizing function value than the current solution, the new solution will replace the current solution. If not, the current solution stays the same.

2.2 Update Procedure

To update the solutions in the weight vector based external archive, there are mainly two steps:

1. Weight vector searching, which finds the nearest weight vector for the new generated solution.

2. External archive solution update, which updates the solutions by a scalarizing function.

2.2.1 *Weight vector searching.*

First, we need to find the closest weight vector in the weight vector grid of the external archive, and then assign the generated solutions to that weight vector. The cosine similarity value between the weight vector and the new generated solution is calculated as the closeness indicator which defined as follows:

$$\cos\theta = \frac{\mathbf{wv} \cdot \mathbf{s}}{||\mathbf{wv}||||\mathbf{s}||} \tag{1}$$

Where $\mathbf{wv}$ is the weight vector in the weight vector grid of external archive and $\mathbf{s}$ is the new generated solution.

2.2.2 *The external archive solution update.*

After finding the closest weight vector for the new generated solution, we update the weight vector based external archive by using a scalarizing function such as PBI and Tchebycheff function. In our experiments, we use Tchebycheff as the scalarizing function. The current solution which associated with the weight vector and the new generated solution are compared through their scalarizing function values. The better one will be kept for this weight vector.

3 EXPERIMENTAL RESULTS

The parameter setting of the different versions of MOEA/D algorithm in our experiments are set as follows:

- Initialization method: Randomly Initialization
- Mutation probability: 0.01 (Randomly Generated)
- Crossover probability: 1 (Blending Crossover)
- Scalarizing function: Tchebycheff
- Neighborhood size: 2% of population size

The population size in different experiments is specified in the experiment part. We use the number of examined solutions as the terminate condition to make sure that the computational load among different versions of MOEA/D algorithm are the same.

Our computational experiments are performed on a PC with Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20GHz and 128 GB RAM.

The result and source code can be found in https://github.com/nixizi/Weight-Vector-Grid-Based-Archive

Table 1: Hypervolume comparison

Hypervolume				
Algorithm	DTLZ1	I-DTLZ1	DTZL2BZ	Avg.
MOEA/D-WV [1]	1.1460	0.3081	0.5773	0.6771
MOEA/D-NA [2]	1.0628	0.2717	0.5660	0.6335
MOEA/D-DB [3]	1.1509	0.3169	0.5827	0.6835
MOEA/D-BA [4]	1.1509	0.3169	0.5827	0.6835

[1] MOEA/D with the weight vector based archive
[2] MOEA/D without the archive
[3] MOEA/D with the dominance-based archive
[4] MOEA/D with the bounded dominance-based archive

3.1 Solution Set Comparison

As we have discussed in Section 1, the majority of archive mechanisms are based on Pareto dominance. Therefore, in the experiment, we compare the two major types of the dominated-based archive, i.e., the bounded dominated-based archive and the unbounded dominated-based archive. We embedded those two types of dominated-based archive and the new proposed weight vector grid based archive into MOEA/D to generate three algorithms, the MOEA/D with the dominance-based archive (MOEA/D-DB), MOEA/D with bounded dominated-based archive (MOEA/D-BA) and MOEA/D with the weight vector based archive (MOEA/D-WV). When the bounded archive is full, we remove the worst solution in the archive and replace it with the new one. Besides, we use MOEA/D without archive (MOEA/D-NA) for the comparison.

We compare MOEA/D-DB, MOEA/D-NA, MOEA/D-WV and MOEA/D-BA by applying those algorithms to different types of test problems. The first one is DTLZ1 [2] which has a regular triangle PF and all the weight vectors are within the PF. The second one is the inverted DTLZ1 [8], which has inverted triangular PF with only part of weight vectors are within the PF. The third one is DTLZ2BZ [1], which has concave PF and only part of weight vectors are within the PF.

Fig. 2 shows that the result of the MOEA/D-NA and MOEA/D-WV on DTLZ1 and I-DTLZ1. From the Fig. 2, we can observe that these two algorithms are able to find the PF of both problems. However, when the shape of the PF is inconsistent with the weight vector grid, those two algorithms perform totally different. The MOEA/D-NA generates more solution on the boundary and relative fewer solutions inside of inverted triangle PF. However, MOEA/D-WV generates a uniformly distributed solutions in the inverted triangle PF.

In order to show the difference more precisely, we apply MOEA/D-WV, MOEA/D-NA, MOEA/D-DB and MOEA/D-BA to the three types of test problem set in three dimension. The experiment settings are as follows:

- Dimension: 3
- Population size: 100
- External Archive size: 1000
- Number of evaluated solutions: 100000
- Total Runs: 100 (Average result return)

To be more rigorous, we update all the external archive structures in the same run, therefore the individuals generated by the MOEA/D algorithm are the same. The results are shown in Table 1.

Table 2: Computational time comparison

Computational Time (sec)				
Algorithm	3D	4D	5D	Avg.
MOEA/D-WV [1]	397	1260	2572	1410
MOEA/D-DB [2]	1083	>7200	>7200	>7200
MOEA/D-BA [3]	401	>7200	>7200	>7200

[1] MOEA/D with the weight vector based archive
[2] MOEA/D with the dominance-based archive
[3] MOEA/D with the bounded dominance-based archive

In Table 1, the results show that MOEA/D-DB and MOEA/D-BA perform best among those four algorithms. MOEA/D-NA performs the worst. Besides, there is only less than 1% difference in hypervolume between MOEA/D-WV and MOEA/D-DB. Therefore, the quality of solution set generated by MOEA/D-DB, MOEA/D-BA and MOEA/D-WV are relatively same. Also, MOEA/D without archive shows poor performance on the test problems where a large number of points are distributed on the boundaries of the PF.

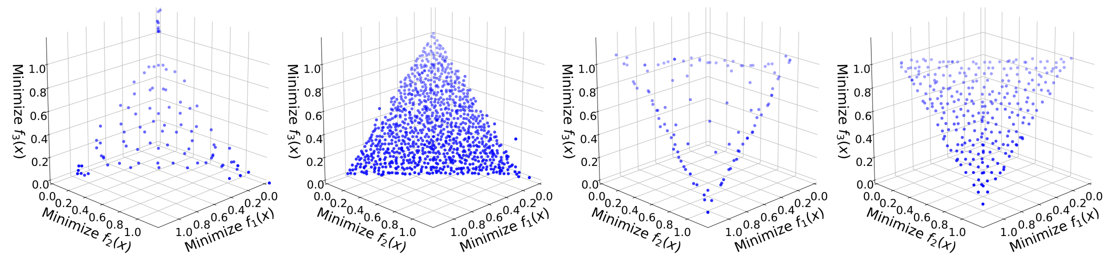
3.2 Computational Time Comparison

In Section 3.1, MOEA/D-WV, MOEA/D-DB, and MOEA/D-BA show similar performance on three test problems, DTLZ1, I-DTLZ1, and DTZL2BZ. The only difference is the computational time. To compare the computational time of the MOEA/D-DB, MOEA/D-WV and MOEA/D-BA, we apply those three algorithms to I-DTLZ1 with different dimensions. The experiment settings are as follows:

- Dimension: 3, 4, 5
- Population size: 100, 200, 300
- External Archive size: 1000, 2000, 3000
- Number of evaluated solutions: 100000, 200000, 300000
- Total Runs: 100 (Average Result Return)

Table 2 shows the results. When the dimension is three, the MOEA/D-DB takes the longest time to complete, while the computational time of MOEA/D-WV and MOEA/D-BA are similar. However, when the dimension is higher than three, the computational time for MOEA/D-BA and MOEA/D-DB are more than 7200 seconds which is much longer than the computational time for MOEA/D-WV. The computational time for the weight vectors update mechanism depends on the size of the archive. It will not increase computational complexity. However, the computational time for the unbounded dominance-based archive and bounded dominance-based archive depend largely on the dimension. For MOEA/D-DB, the computational time increases a lot when increase dimension, since more nondominated solutions are generated in the late generations and more time is needed in the update process. For MOEA/D-BA, when the dimension increase, the computational time for select the worst solutions increases.

From Table 1 & 2, although the quality of MOEA/D-DB, MOEA/D-BA, and MOEA/D-WV are similar, the MOEA/D-WV can generate a good solution set in relatively short time. Nevertheless, MOEA/D-WV spends most of the time in updating solutions to the external archive, so if we can reduce the updating computational time or reduce the number of updating, we can reduce a lot of computational time for MOEAs with the weight vector based external archive.



(a) MOEA/D **(b) MOEA/D-WV** **(c) MOEA/D** **(d) MOEA/D-WV**

Figure 2: Experimental result by two algorithms on DTLZ1 and I-DTLZ1

4 FURTHER STUDY DISCUSSION

In this section, we mainly focus on how to decrease the computational time for updating the weight vector based archive and some research topics we can study further.

For decreasing computational time, we can use several ways to speed up the updating of the weight vector based archive. First, not the all new generated solutions are needed to update to the archive. The solution generated in the early generation have higher chance to be replaced with a new one. Therefore, if we only update the last part of the generation, we can obtain a solution set with similar quality. Second, after updating part of solutions, we can possibly obtain the shape of PF and then remove the weight vectors which are out of the PF, in order to reduce the computational time for the weight vector search process. Third, how to search the nearest weight vector is essential for deducing the computational time.

For our further research, first, we will adaptively change the weight vector grid of the external archive in the MOEA/D to generate more weight vectors inside of PF. This may be another way to solve the problem we mentioned in Section 1 (i.e., the nonuniformity of the solution distribution). Second, we can use weighted sum function as the scalarizing function for MOEA/D and Tchebycheff or PBI for the weight vector based archive, due to the reason that the weighted sum approach can push solutions to PF faster than PBI and Tchebycheff approaches, while the PBI and Tchebycheff can be used to maintain diversity of the solutions.

5 CONCLUSION

Weight vector grid is often used in decomposition-based MOEA. However, when the shape of PF is inconsistent with the distribution of prespecified weight vectors, a nonuniform solution set could be obtained. Therefore, we proposed a new external archive update mechanism with a large weight vector based archive. The new external archive update mechanism can be applied to any MOEA with an external archive. In our experiment, we applied our new external archive update mechanism in MOEA/D to demonstrate the effectiveness of the proposed mechanism. We showed that MOEA/D-WV is able to generate a uniformly distributed solution set, and its computational time is much shorter than MOEA/D-DB and MOEA/D-BA. In the future, we will focus on how to reduce the update times of the weight vector based external archive and how to reduce the computational time on searching the nearest weight vector.

ACKNOWLEDGMENTS

This work was supported by the Science and Technology Innovation Committee Foundation of Shenzhen
(Grant No. ZDSYS201703031748284).

REFERENCES

[1] Dimo Brockhoff and Eckart Zitzler. 2009. Objective Reduction in Evolutionary Multiobjective Optimization: Theory and Applications. *Evol. Comput.* 17, 2 (June 2009), 135–166. https://doi.org/10.1162/evco.2009.17.2.135

[2] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2005. *Scalable Test Problems for Evolutionary Multiobjective Optimization.* Springer London, London, 105–145. https://doi.org/10.1007/1-84628-137-7_6

[3] VL Ling Huang, Ponnuthurai Nagaratnam PN Suganthan, AK Alex Kai Qin, and S Baskar. 2005. Multiobjective differential evolution with external archive and harmonic distance-based diversity measure. *Sch. Electr. Electron. Eng. Nanyang, Technol. Univ. Tech. Rep.* July (2005), 1–25. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.6982

[4] Hisao Ishibuchi, Naoya Akedo, and Yusuke Nojima. 2013. Relation between neighborhood size and MOEA/D performance on many-objective problems. In *International Conference on Evolutionary Multi-Criterion Optimization.* Springer, 459–474.

[5] Hisao Ishibuchi, Yuji Sakane, Noritaka Tsukamoto, and Yusuke Nojima. 2009. Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on.* IEEE, 1758–1763.

[6] Hisao Ishibuchi, Yu Setoguchi, Hiroyuki Masuda, and Yusuke Nojima. 2017. Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes. *IEEE Transactions on Evolutionary Computation* 21, 2 (2017), 169–190.

[7] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. 2008. Evolutionary many-objective optimization: A short review. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on.* IEEE, 2419–2426.

[8] Himanshu Jain and Kalyanmoy Deb. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Trans. Evolutionary Computation* 18, 4 (2014), 602–622.

[9] Hui Li and Qingfu Zhang. 2009. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on evolutionary computation* 13, 2 (2009), 284–302.

[10] Yuan Yuan, Hua Xu, Bo Wang, and Xin Yao. 2016. A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 20, 1 (2016), 16–37.

[11] Jingqiao Zhang and Arthur C Sanderson. 2009. JADE: adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* 13, 5 (2009), 945–958.

[12] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.