MOEA/D with Uniformly Randomly Adaptive Weights

Lucas R. C. de Farias
Universidade Federal de Pernambuco
Recife, Brazil
lrcf@cin.ufpe.br

Pedro H. M. Braga
Universidade Federal de Pernambuco
Recife, Brazil
phmb4@cin.ufpe.br

Hansenclever F. Bassani
Universidade Federal de Pernambuco
Recife, Brazil
hfb@cin.ufpe.br

Aluizio F. R. Araújo
Universidade Federal de Pernambuco
Recife, Brazil
aluizioa@cin.ufpe.br

ABSTRACT

When working with decomposition-based algorithms, an appropriate set of weights might improve quality of the final solution. A set of uniformly distributed weights usually leads to well-distributed solutions on a Pareto front. However, there are two main difficulties with this approach. Firstly, it may fail depending on the problem geometry. Secondly, the population size becomes not flexible as the number of objectives increases. In this paper, we propose the MOEA/D with Uniformly Randomly Adaptive Weights (MOEA/D-URAW) which uses the Uniformly Randomly method as an approach to subproblems generation, allowing a flexible population size even when working with many objective problems. During the evolutionary process, MOEA/D-URAW adds and removes subproblems as a function of the sparsity level of the population. Moreover, instead of requiring assumptions about the Pareto front shape, our method adapts its weights to the shape of the problem during the evolutionary process. Experimental results using WFG41-48 problem classes, with different Pareto front shapes, shows that the present method presents better or equal results in 77.5% of the problems evaluated from 2 to 6 objectives when compared with state-of-the-art methods in the literature.

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms**;

KEYWORDS

Many-objective optimization, Multi-objective optimization, Evolution strategies, Decomposition methods

ACM Reference Format:
Lucas R. C. de Farias, Pedro H. M. Braga, Hansenclever F. Bassani, and Aluizio F. R. Araújo. 2018. MOEA/D with Uniformly Randomly Adaptive Weights. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Kyoto, Japan,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205455.3205648

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

ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205648

1 INTRODUCTION

Decomposition-based evolutionary multiobjective optimization (EMO) algorithms decomposes a multiobjective optimization problem (MOOP) [2] into a number of single-objective optimization problems using a set of weight vectors [10]. Each subproblem or weight vector is associated with a solution in the population, and the diversity of the evolutionary population is controlled explicitly by a set of weight vectors [6]. Thus, an appropriate set of weights can increase the quality of the final solution.

A set of uniformly distributed weights usually leads to well-distributed solutions on a Pareto front (PF). However, there are two main difficulties with this approach. Firstly, it may fail depending on the problem geometry. When dealing with a complex Pareto Front (e.g., disconnected, degenerate, badly-scaled or inverted simplex-like), the final solution set can present results do not meet the expectations [6–8]. Secondly, the population size becomes not flexible as the objectives number grows, because it behaves non-linearly, that is, when working with Many-objectives Optimization Problem (MaOP), the computational cost significantly increases [8].

To handle such limitations, we propose a method called MOEA/D with Uniformly Randomly Adaptive Weights (MOEA/D-URAW) that adapts the subproblems based on the sparsity level of the population. This approach allows a flexible population size for the MaOPs. We compared our method with other MOEA/D's variants that use fixed weight vectors for a set of test problems with different PF geometries. Experimental results show that the MOEA/D-URAW presents better or equal results in 77,5% of the problems evaluated from 2 to 6 objectives.

The rest of this paper is organized as follows: Section 2 discusses the background knowledge used in this work. Section 3 introduces the proposed method MOEA/D-URAW. Thereafter, Section 4 and Section 5 present the experimental setup and results analysis respectively. Finally, the conclusions are presented in Section 6.

2 BACKGROUND

This section presents briefly the Tchebycheff (TCH) decomposition approach and the fixed weights generation methods used in the experiments of this paper. Such a decomposition is the basis for the rest of the paper.

2.1 Tchebycheff Decomposition Approach

In this approach, the scalar optimization problems are defined as:

$$\begin{aligned} &\text{minimize } g^{TCH}(\boldsymbol{x}|\boldsymbol{\lambda}, \boldsymbol{z^*}) = \max_{1\leq j\leq m}(\lambda_j|f_j(\boldsymbol{x}) - z_j^*|), \\ &\text{subject to } \boldsymbol{x} \in \Omega \end{aligned} \tag{1}$$

where the utopian objective vector $\boldsymbol{z^*} = (z_1^* \ldots z_m^*)^T$ is the reference point, i. e. $z_j^* = \min\{f_j(\boldsymbol{x})|\boldsymbol{x} \in \Omega\}$, for each $j = 1, \ldots, m$. The m-dimensional weight vector is defined as $\boldsymbol{\lambda} = (\lambda_1 \ldots \lambda_m)^T$, $\sum_i^m \lambda_i = 1$ and $\lambda_i \geq 0$, for all $i \in 1, \ldots, m$ [10]. By altering weight vectors, different Pareto-optimal solutions can be obtained by the TCH approach [9]

2.2 Fixed Weights Generation Methods

Three different methods to generate a set of fixed weight vectors are described in the following Section 2.2.1, Section 2.2.2 and Section 2.2.3. Section 2.2.4 presents the WS-tranformation used in the three described methods.

2.2.1 Das and Dennis (DD). Most decomposition-based EMO algorithms use the method proposed by Das and Dennis [1] to systematically generate a set of fixed weight vectors uniformly distributed over a unit simplex. Let H be the number of divisions of each axis, totally $N = \binom{H+m-1}{m-1}$ weight vectors can be generated using this approach. Since H should be no smaller than m to prevent intermediate points being created. The number of generated weight vectors can be very high for more than three objectives.

2.2.2 Uniform Randomly (UR). A set of N weight vectors W are generated as follows. Firstly, 5000 weight vectors are uniformly random generated for forming the set W_1. W is initialized as the set containing all the weight vector (1 0 ... 0 0), (0 1 ... 0 0), ... , (0 0 ... 0 1). Secondly, find the weight vector in W_1 with the largest distance to W, add it to W, and remove it from W_1. Then, if the size of W is N, stop and return W. Otherwise, go to the second item and repeat the process [11].

2.2.3 TCH Scalarizing Function (TSF). Given a reference point, $\boldsymbol{z^*}$, the optimal weight vector to a solution with respect to Tchebycheff scalarizing function can be easily generated. This is a frequent approach in the weight vector adaptation [3, 6, 7]. Formally, let $\boldsymbol{z^*} = (z_1^* \ldots z_m^*)^T$ be the reference point and $\boldsymbol{w} = (\lambda_1\ \lambda_2 \ldots \lambda_m)^T$ be the optimal weight vector to a solution $\boldsymbol{q}$. Then Eq. 2 holds:

$$\frac{f_1(\boldsymbol{q}) - z_1^*}{\lambda_1} = \frac{f_2(\boldsymbol{q}) - z_2^*}{\lambda_2} = \cdots = \frac{f_m(\boldsymbol{q}) - z_m^*}{\lambda_m} \tag{2}$$

Since $\lambda_1 + \lambda_2 + \ldots + \lambda_m = 1$, we have Eq. 3.

$$\boldsymbol{w} = (\lambda_1, \cdots, \lambda_m) = \left(\frac{f_1(\boldsymbol{q}) - z_1^*}{\sum_{i=1}^m f_i(\boldsymbol{q}) - z_i^*}, \cdots, \frac{f_m(\boldsymbol{q}) - z_m^*}{\sum_{i=1}^m f_i(\boldsymbol{q}) - z_i^*}\right) \tag{3}$$

2.2.4 WS-Transformation. It maps the weight vector of a scalar subproblem to its solution mapping vector [7]. If $\boldsymbol{\lambda}$ is a weight vector, $\boldsymbol{\lambda} = (\lambda_1 \ldots \lambda_m)^T \in \mathbb{R}^m$, satisfying $\sum_{i=1}^m \lambda_i = 1$, $\lambda_i \geq 0$, $i = 1$, ..., m. Then the WS-transformation, giving rise to $\boldsymbol{\lambda'}$, on $\boldsymbol{\lambda}$ can be defined as:

$$\boldsymbol{\lambda'} = WS(\boldsymbol{\lambda}) = \left(\frac{\frac{1}{\lambda_1}}{\sum_{i=1}^m \frac{1}{\lambda_i}}, \cdots, \frac{\frac{1}{\lambda_m}}{\sum_{i=1}^m \frac{1}{\lambda_i}}\right) \tag{4}$$

3 PROPOSED METHOD

This section presents in details the proposed method for adapting weights during the evolutionary process called MOEA/D-URAW.

3.1 Weights Generation

MOEA/D-URAW uses the Uniformly Randomly method presented in Section 2.2.2 for generating weight vectors. In this approach, the population size is flexible, that is, it independs on the number of objectives. However, maintain a fixed set of weights throughout the evolutionary process means assuming that the PF follows a given geometry. This is a problem because the use of a fixed set of weights does not guarantee to find well-distributed solutions for all PF shapes. Adapting weights helps to deal with this problem [6, 7, 9]. An adaptation method is presented in Section 3.2.

3.2 Weights Adaptation

The method proposed in this paper is based on the methodology presented in the MOEA/D with Adaptive Weight Adjustment (MOEA/D-AWA)[7] that uses the sparsity level among individuals of the population to indicate which subproblem should be removed and which is apt to be added.

The sparsity level is based on the vicinity distance [4]. This approach is defined in Equation (5), where $L_2^{NN_i^j}$ is the j-th individual euclidean distance, ind^j, along with its i-th nearest neighbor of the population, pop. The m closest euclidean distances are used, where m is the number of objectives [7].

$$SL(ind^j, pop) = \prod_{i=1}^{m} L_2^{NN_i^j} \tag{5}$$

The individual with the lowest level of sparsity or overcrowded is removed. That is, the sparsity level of each of the individuals in the population is calculated among the population itself using Equation (5). If 5% of the sub-problems (nus) have not been removed yet of the population, then, repeat the process of calculating the sparsity levels and remove the most overcrowded subproblem [7]. An important observation is that the MOEA/D-URAW does not update the current evolutionary population by checking which individual is best suited to which weight vector as the MOEA/D-AWA does.

In addition to the evolutionary population, the MOEA/D framework has an external population (EP) that is used to store non-dominated solutions found during the search. The procedure to create new subproblems consists of calculating the sparsity level of each individual in an EP with respect to the population itself, using Equation (5). Then, it generates a new subproblem using the individual $\boldsymbol{ind^{sp}} = (\boldsymbol{x^{sp}}\ \boldsymbol{FV^{sp}})$ which has the highest sparsity level. The weight vector $\boldsymbol{\lambda^{sp}}$ of the new constructed subproblem can be calculated as follows, in which $\boldsymbol{FV^{sp}} = (f_1^{sp} \ldots f_m^{sp})$,

$$\boldsymbol{\lambda}^{sp} = \left(\frac{\frac{1}{f_1^{sp} - z_1^*}}{\sum_{k=1}^{m} \frac{1}{f_k^{sp} - z_k^*}}, \cdots, \frac{\frac{1}{f_m^{sp} - z_m^*}}{\sum_{k=1}^{m} \frac{1}{f_k^{sp} - z_k^*}}\right), \prod_{j=1}^{m} (f_j^{sp} - z_j^*) \neq 0. \tag{6}$$

At last, the solution of the new constructed subproblem as ind^{sp} is set and added to the current population. The process is repeated until *nus* subproblems are added to the population.

There is no consensus in the literature about the best moment for adaptation [6, 7, 9]. In a preliminary evaluation the MOEA/D-URAW performed better when the weight update operation is conducted every 5% of the total generations/evaluations. We followed it, and MOEA/D-URAW does not change the weight vectors during the last 10% generations/evaluations.

3.3 Algorithm Framework

Alg. 1 presents the main procedure of MOEA/D-URAW. As can be seen, it has the same framework as MOEA/D version in [5]. However, it employs the weight vector update using sparsity level (line 26-37), initialization of weights using uniformly randomly method (line 2) and the limitation of the size of EP (line 24-25).

4 EXPERIMENTAL SETUP

4.1 Test Problems

In the experimental study, we used eight modified WFG4 test problems, i.e., WFG41 to WFG48 [8], from 2 to 6 objectives. They have different PF characteristics, namely continuous and discontinuous, convex, concave, strong convex, strong concave and mixed PFs with different shapes. Each WFG problem had 100 executions. The number of decision variables is defined as $n = k + l$, and m determines the number of objectives. The others settings are described in Table 1. Note that, the population size was thus defined, in order to have a fair comparison between the test algorithms, since Das and Dennis method does not provide a flexible value when treating many-objectives.

4.2 Test Algorithms

To assess the performance of MOEA/D-URAW, the Wilcoxon's rank sum test is used with the significant level of 5%. Three ways of initializing unadjusted weights in the MOEA/D framework presented in section Section 3.3 is considered for comparison: Das and Dennis [1], Uniformly Randomly [11] and TCH Scalarizing Function [3, 6]. These algorithms used the same general settings as shown in Table 2.

4.3 Performance Metric

For WFG41 to WFG48 test problems, we use the Hypervolume metric (HV), since the PFs are unknown. Given a reference point $z^r = (z_1^r, \ldots, z_n^r)^T$ dominated by all Pareto-optimal solutions, the HV of a solution set P is defined as the volume of the objective space dominated by all solutions in P, bounded by z^r:

$$HV(P) = VOL(\bigcup_{z \in P} [z_1, z_1^r] \times \ldots \times [z_m, z_m^r]), \tag{7}$$

Algorithm 1: MOEA/D-URAW

```
1  EP ← ∅;
2  Initialize the population P and a set of weight vectors W;
3  Apply the WS-transformation on the weight vectors W;
4  Determine the neighbors of each weight vector of W;
5  Calculate the reference point according to P;
6  Gen ← 0;
7  while Gen < Gen_max do
8      for each i ∈ {1, ..., N} do
9          if uniform(0, 1) < δ then
10             E ← B(i);
11         else
12             E ← {1, ..., N};
13         Randomly select mating solutions from E to generate
             an offspring x̄, Evaluate F(x̄);
14         update ← 0;
15         while update < nr and E ≠ ∅ do
16             j ← Randomly select an index from E;
17             E ← E \ j;
18             if g^TCH(x̄|w^j, z*) ≤ g^TCH(x^j|w^j, z*) then
19                 x^j ← x̄;
20                 update ← update + 1;
21         if ∄q ∈ EP, q ≺ x̄ then
22             EP ← EP ∪ x̄;
23             EP ← EP \ {q ∈ EP | x̄ ≺ q};
24     if |EP| > 2|P| then
25         Remove from EP the individual with the highest
             sparsity level;
26     if Gen = Gen_max × 5% and < Gen_max × 90% then
27         adjust ← 0
28         while adjust < nus do
29             Calculate the sparsity level of each individual in
                 population P among P by Equation (5);
30             Remove the individual with the minimum sparsity
                 level;
31             adjust ← adjust + 1;
32         while adjust > 0 do
33             Calculate the sparsity level of each individual in
                 EP among the population P by Equation (5);
34             Generate a new subproblem using the individual
                 which has the largest sparsity level by Equation
                 (6);
35             Add the newly constructed subproblem associated
                 with the individual which has the largest sparsity
                 level to the current population P;
36             adjust ← adjust − 1;
37         Update the neighbors of each weight vector of W;
38     Gen ← Gen + 1;
39 return P;
```

Table 1: Settings used for WFG41 to WFG48.

Parameters	Values
runs	100
maxGen	400
N (2-4 objectives)	120
N (5 and 6 objectives)	126
T	24
nus (0.05N)	6
k (2 objectives)	2
k (otherwise)	m-1
l	10

Table 2: General Parameters.

Parameters	Values
Crossover	Simulated Binary
P_c	1.00
η_c	20
Mutation	Polynomial
P_m	1/n
η_m	20
δ	0.9
nr	2

where VOL indicates the Lebesgue measure. The objective vectors of the final solution set are normalized according to min-max normalization concerning all experiments before calculating the HV with $z^r = (1.2, \ldots, 1.2)^T$.

5 EXPERIMENTAL RESULTS

As shown in Section 4, the Wilcoxon's rank sum test is used to indicate the MOEA/D-URAW performance when compared to the DD, UR and TSF approaches in MOOP and MaOP. According to Table 3 and Table 5, MOEA/D-URAW presents results with a significance level of 5%, better in 72.5% of cases, draws at 5% and not so good at 22.5%. Therefore, the proposed method was better or equal to in 77.5% of the problems. The results in the context of MOOP and MaOP are detailed below.

Firstly, Table 3 and Table 4 shows the HV results for 2 and 3 objectives. In this context of MOOP, the MOEA/D-URAW performs the best on most of the test instances except for WFG47 for 2 objectives, and WFG43 for 3 objectives. For these cases, Tchebycheff Scalarizing Function, and Das and Dennis obtained the best HV results, respectively.

The final solution set with the median HV metric values for 2 objectives is shown in Figures 1 and 2. It can be seen that the PFs of MOEA/D-URAW are mostly more evenly distributed. The adaptation of the weight vectors in the UR initialization method allows that it happens.

Secondly, Table 5 and Table 6 shows the HV results for MaOPs with 4, 5, and 6 objectives. Here, MOEA/D-URAW continues to present better result in most of the test instances. The problem with WFG43 persists for all the cases, where Das and Dennis, and Uniformly Randomly outperforms it.

Moreover, for 5 and 6 objectives, WFG44 appears as a problem for MOEA/D-URAW. For the WFG47, the MOEA/D-URAW is not the best in the 4 and 5 objectives, in these cases, its version without adaptation of weights, UR, presented better performance.

These problems presented by MOEA/D-URAW to the WFG43 and WFG44 are interesting to analyze. The fact is that both of them have strong shapes, concave and convex, respectively. It may impact the effectiveness of MOEA/D-URAW. The WFG47 result problems for 2 objectives was unexpected, and it is a key point of future investigation.

6 CONCLUSIONS

This work introduces the MOEA/D-URAW, a model that uses UR initialization method combined with weights adaptation in order to obtain both flexible population size and better adapted final solution sets. Its performance was evaluated using MOOPs and MaOPs with different PF shapes.

The results indicate that the MOEA/D-URAW presents a better performance, specifically in WFG problems with concave, convex, mixed, linear, convex and disconnected hyperplane PF shapes. The proposed model is better than the other methodologies evaluated in 72.5% of cases, considering all problems and objectives tested. However, it is important to note that the proposed adaptation did not significantly improve the results when the PF shape is disconnected and convex. In 22.5% of the results, there were statistically significant differences. In these cases, the PFs have strong concave or strong convex formats. These results indicate that the use of sparsity level may not be appropriate to adapt the weights with these PFs formats. There is still a case in which the TSF approach performed better than the others, i.e., the WFG47 with 2 objectives. This case has a PF with disconnected and concave shape, and the reason why this happens will be investigated in more detail in a future work.

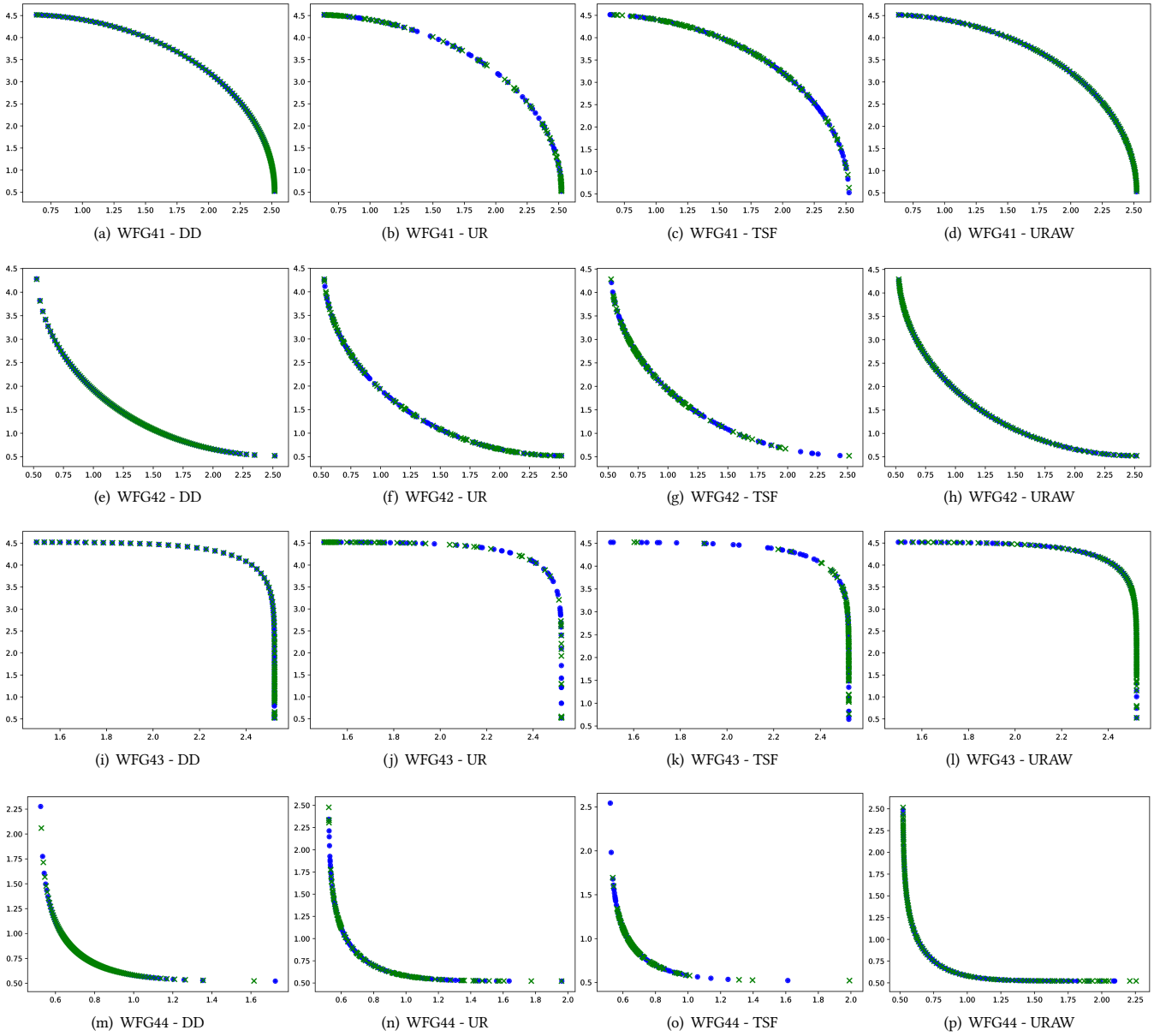
Another relevant issue is that, in the literature, there is no consensus on the best frequency of weight adaptation. Different frequencies affect the outcome of the approaches that use weight adaptation. As future work, it is worth investigating the impact of this parameter in approaches that use weights adaptation.

ACKNOWLEDGMENTS

The authors would like to thank the Brazilian National Council for Technological and Scientific Development (CNPq) for partially financing this research study.

REFERENCES

[1] Indraneel Das and John E Dennis. 1998. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization* 8, 3 (1998), 631–657.
[2] Kalyanmoy Deb. 2001. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons.



(a) WFG41 - DD (b) WFG41 - UR (c) WFG41 - TSF (d) WFG41 - URAW

(e) WFG42 - DD (f) WFG42 - UR (g) WFG42 - TSF (h) WFG42 - URAW

(i) WFG43 - DD (j) WFG43 - UR (k) WFG43 - TSF (l) WFG43 - URAW

(m) WFG44 - DD (n) WFG44 - UR (o) WFG44 - TSF (p) WFG44 - URAW

Figure 1: Final solutions set with the best (blue circle) and median (green cross sign) HV metric values obtained by the algorithms on WFG41 to WFG44.

[3] Fangqing Gu, Hai-Lin Liu, and Kay Chen Tan. 2012. A multiobjective evolutionary algorithm using dynamic weight design method. *International Journal of innovative Computing, Information and Control* 8, 5 (B) (2012), 3677–3688.

[4] Saku Kukkonen and Kalyanmoy Deb. 2006. A fast and effective method for pruning of non-dominated solutions in many-objective problems. In *PPSN*, Vol. 4193. 553–562.

[5] 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.

[6] Miqing Li and Xin Yao. 2017. What Weights Work for You? Adapting Weights for Any Pareto Front Shape in Decomposition-based Evolutionary Multi-Objective Optimisation. *arXiv preprint arXiv:1709.02679* (2017).

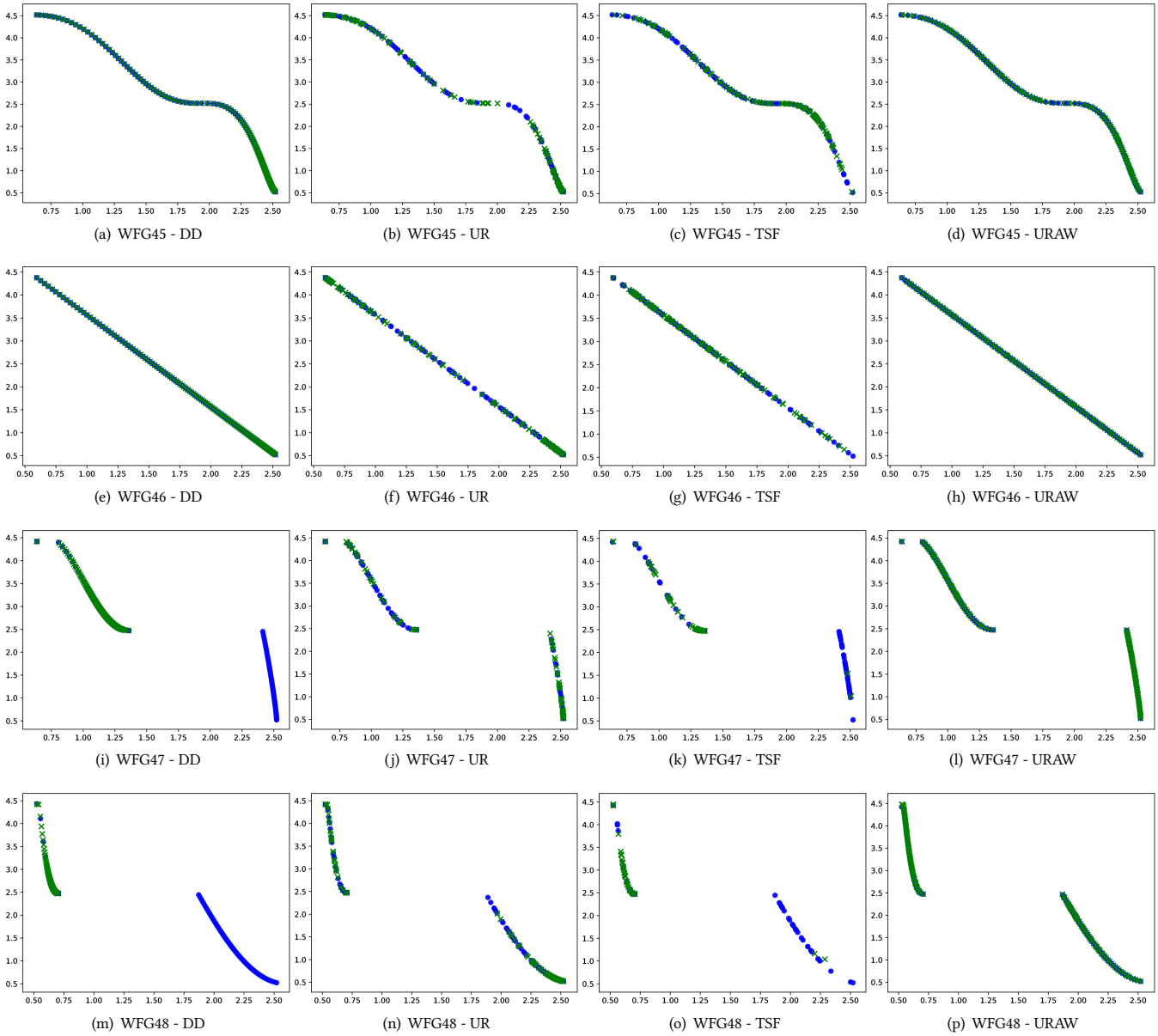
[7] Yutao Qi, Xiaoliang Ma, Fang Liu, Licheng Jiao, Jianyong Sun, and Jianshe Wu. 2014. MOEA/D with adaptive weight adjustment. *Evolutionary computation* 22, 2 (2014), 231–264.

[8] Rui Wang, Robin C Purshouse, and Peter J Fleming. 2015. Preference-inspired coevolutionary algorithms using weight vectors. *European Journal of Operational Research* 243, 2 (2015), 423–441.

[9] Mengyuan Wu, Sam Kwong, Yuheng Jia, Ke Li, and Qingfu Zhang. 2017. Adaptive weights generation for decomposition-based multi-objective optimization using Gaussian process regression. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 641–648.

[10] 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.

[11] Qingfu Zhang, Wudong Liu, and Hui Li. 2009. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 203–208.



(a) WFG45 - DD (b) WFG45 - UR (c) WFG45 - TSF (d) WFG45 - URAW

(e) WFG46 - DD (f) WFG46 - UR (g) WFG46 - TSF (h) WFG46 - URAW

(i) WFG47 - DD (j) WFG47 - UR (k) WFG47 - TSF (l) WFG47 - URAW

(m) WFG48 - DD (n) WFG48 - UR (o) WFG48 - TSF (p) WFG48 - URAW

Figure 2: Final solutions set with the best (blue circle) and median (green cross sign) HV metric values obtained by the algorithms on WFG45 to WFG48.

Table 3: Average and standard deviation of HV results obtained in 100 independent runs on WFG41 to WFG48 on MOOP. The highest average is highlighted with gray background. ⋆ indicates whether the MOEA/D-URAW is significantly better than all other models or which algorithm presented significantly better results than it.

Problems	2 Objectives				3 Objectives			
	DD	UR	TSF	URAW	DD	UR	TSF	URAW
WFG41	0.6622 1.50E-03	0.6567 4.80E-03	0.6620 1.30E-02	0.6669 ⋆ 1.60E-03	1.0736 2.70E-03	1.0672 8.10E-03	0.9665 2.90E-02	1.1145 ⋆ 3.30E-03
WFG42	1.2078 9.00E-04	1.206 1.20E-03	1.1994 4.00E-03	1.2095 ⋆ 6.00E-04	1.652 6.50E-03	1.6829 2.70E-03	1.6408 9.00E-03	1.6906 ⋆ 1.70E-03
WFG43	0.4806 5.10E-03	0.4798 2.10E-03	0.4831 5.60E-02	0.5015 3.70E-03	0.6533 ⋆ 1.70E-03	0.6401 4.10E-03	0.5088 4.70E-02	0.6416 8.60E-03
WFG44	1.3865 1.20E-03	1.3859 8.00E-04	1.3802 3.70E-03	1.3909 ⋆ 5.00E-04	1.7225 3.50E-03	1.7243 1.20E-03	1.6887 7.70E-03	1.7260 ⋆ 1.40E-03
WFG45	0.8091 1.60E-03	0.8074 2.90E-03	0.8098 5.70E-03	0.8109 1.40E-03	1.1995 5.20E-03	1.2181 7.10E-03	1.1167 2.70E-02	1.2524 ⋆ 3.50E-03
WFG46	0.9354 1.60E-03	0.9278 3.90E-03	0.9268 1.10E-02	0.9369 ⋆ 1.20E-03	1.4210 5.20E-03	1.4546 6.80E-03	1.3606 2.00E-02	1.4866 ⋆ 2.80E-03
WFG47	0.8012 5.50E-02	0.8014 5.30E-02	0.8474 ⋆ 5.30E-02	0.8179 5.30E-02	1.2521 1.40E-02	1.2648 5.70E-03	1.1487 2.80E-02	1.2773 ⋆ 3.00E-02
WFG48	0.9317 9.80E-02	0.9509 9.50E-02	0.9357 6.70E-02	0.9526 ⋆ 9.70E-02	1.6108 9.00E-03	1.6212 3.00E-03	1.4753 4.30E-02	1.6308 ⋆ 2.20E-03

Table 4: The best and median HV metric results obtained in 100 independent runs on WFG41 to WFG48 on MOOP. The highest values in the results are highlighted in bold.

Problems	2 Objectives				3 Objectives			
	DD	UR	TSF	URAW	DD	UR	TSF	URAW
WFG41	0.6635 0.6625	0.6628 0.6580	**0.6768** 0.6647	0.6687 **0.6675**	1.0788 1.0744	1.0811 1.0684	1.0238 0.9676	**1.1196** **1.1155**
WFG42	1.2088 1.2079	1.2080 1.2061	1.2049 1.2004	**1.2102** **1.2093**	1.6620 1.6530	1.6871 1.6834	1.6653 1.6404	**1.6924** **1.6908**
WFG43	0.4829 0.4815	0.4828 0.4802	**0.5417** **0.5096**	0.5053 0.5030	**0.6561** **0.6535**	0.6467 0.6410	0.6373 0.5110	0.6508 0.6434
WFG44	1.3880 1.3866	1.3871 1.3861	1.3870 1.3810	**1.3914** **1.3912**	1.7252 1.7228	1.7264 1.7241	1.7015 1.6907	**1.7272** **1.7268**
WFG45	0.8104 0.8095	0.8116 0.8081	**0.8173** **0.8115**	0.8126 0.8111	1.2063 1.2006	1.2298 1.2196	1.1732 1.1194	**1.2586** **1.2529**
WFG46	0.9368 0.9358	0.9317 0.9288	**0.9388** 0.9302	0.9382 **0.9371**	1.4308 1.4227	1.4665 1.4559	1.4006 1.3616	**1.4908** **1.4871**
WFG47	0.8621 0.7511	0.8440 0.8414	**0.9104** **0.8643**	0.8557 0.8545	1.2689 1.2550	1.2759 1.2660	1.2186 1.1515	**1.2856** **1.2807**
WFG48	1.0320 0.8377	1.0327 1.0290	1.0246 0.9647	**1.0337** **1.0324**	1.6225 1.6137	1.6267 1.6218	1.5558 1.4806	**1.6344** **1.6314**

Table 5: Average and standard deviation of HV results obtained in 100 independent runs on WFG41 to WFG48 on MaOP. The highest average is highlighted with gray background. ⋆ indicates whether the MOEA/D-URAW is significantly better than all other models or which algorithm presented significantly better results than it.

Problems	4 Objectives				5 Objectives				6 Objectives			
	DD	UR	TSF	URAW	DD	UR	TSF	URAW	DD	UR	TSF	URAW
WFG41	1.340 2.6E-02	1.464 1.1E-02	1.145 4.1E-02	1.494 ⋆ 8.4E-03	1.563 1.6E-02	1.840 1.5E-02	1.312 4.6E-02	1.867 ⋆ 1.3E-02	1.812 4.2E-02	2.215 1.9E-02	1.513 5.1E-02	2.261 ⋆ 2.4E-02
WFG42	2.011 1.5E-02	2.046 5.9E-03	1.942 1.5E-02	2.056 ⋆ 4.1E-03	2.451 5.3E-03	2.460 1.1E-02	2.300 3.0E-02	2.471 ⋆ 7.0E-03	2.939 2.1E-02	2.939 1.9E-02	2.701 5.3E-02	2.961 ⋆ 1.3E-02
WFG43	0.821 4.9E-03	0.832 ⋆ 5.4E-03	0.521 5.1E-02	0.819 1.7E-02	1.009 1.0E-02	1.032 ⋆ 8.9E-03	0.537 5.1E-02	0.998 2.6E-02	1.248 ⋆ 1.4E-02	1.242 1.9E-02	0.582 6.1E-02	1.144 5.8E-02
WFG44	2.067 5.3E-03	2.060 9.8E-03	1.901 3.6E-02	2.068 ⋆ 5.3E-03	2.481 ⋆ 9.2E-03	2.458 3.2E-02	2.070 1.4E-01	2.472 1.4E-02	2.964 ⋆ 3.1E-02	2.861 1.6E-01	2.165 4.8E-01	2.871 1.5E-01
WFG45	1.428 2.2E-02	1.602 1.1E-02	1.331 3.4E-02	1.628 ⋆ 8.7E-03	1.617 2.7E-02	1.984 1.5E-02	1.553 3.7E-02	2.002 ⋆ 1.2E-02	1.860 3.3E-02	2.375 2.5E-02	1.818 5.0E-02	2.400 ⋆ 2.3E-02
WFG46	1.717 1.7E-02	1.877 7.0E-03	1.598 3.0E-02	1.904 ⋆ 6.5E-03	2.011 1.6E-02	2.30 1.2E-02	1.901 3.6E-02	2.337 ⋆ 8.9E-03	2.378 4.6E-02	2.762 2.3E-02	2.210 4.1E-02	2.820 ⋆ 1.6E-02
WFG47	1.506 7.3E-02	1.650 ⋆ 8.1E-03	1.366 3.9E-02	1.643 6.6E-02	1.646 1.1E-01	2.029 ⋆ 8.9E-02	1.610 3.7E-02	2.025 6.1E-02	1.864 8.0E-02	2.429 1.3E-01	1.874 5.8E-02	2.435 ⋆ 7.5E-02
WFG48	1.977 1.7E-02	2.032 6.4E-03	1.759 6.3E-02	2.036 ⋆ 8.1E-02	2.410 1.3E-02	2.445 1.2E-02	2.091 1.0E-01	2.457 ⋆ 1.2E-02	2.903 2.9E-02	2.925 2.1E-02	2.526 1.1E-01	2.944 ⋆ 1.8E-02

Table 6: The best and median HV metric results obtained in 100 independent runs on WFG41 to WFG48 on MaOP. The highest values in the results are highlighted in bold.

Problems	4 Objectives				5 Objectives				6 Objectives			
	DD	UR	TSF	URAW	DD	UR	TSF	URAW	DD	UR	TSF	URAW
WFG41	1.3632 1.3480	1.4878 1.4657	1.2339 1.1504	**1.5110** **1.4948**	1.5854 1.5656	1.8730 1.8395	1.4196 1.3206	**1.8906** **1.8693**	1.8822 1.8264	2.2670 2.2158	1.6321 1.5104	**2.3094** **2.2619**
WFG42	2.0306 2.0102	2.0557 2.0466	1.9716 1.9441	**2.0619** **2.0576**	2.4627 2.4519	2.4717 2.4621	2.3623 2.3021	**2.4812** **2.4739**	2.9570 2.9451	2.9657 2.9449	2.7988 2.7096	**2.9769** **2.9668**
WFG43	0.8327 0.8212	0.8405 **0.8326**	0.6353 0.5283	**0.8432** 0.8247	1.0303 1.0105	**1.0513** **1.0328**	0.6875 0.5343	1.0383 1.0006	**1.2750** **1.2498**	1.2731 1.2463	0.7043 0.5926	1.2245 1.1640
WFG44	2.0731 2.0678	2.0718 2.0620	1.9819 1.9032	**2.0734** **2.0685**	**2.4882** **2.4786**	2.4869 2.4642	2.3018 2.0749	2.4879 2.4731	**2.9860** **2.9694**	2.9833 2.9079	2.7231 2.2491	2.9841 2.9008
WFG45	1.4753 1.4314	1.6228 1.6031	1.4270 1.3342	**1.6428** **1.6284**	1.6613 1.6235	2.0136 1.9857	1.6324 1.5530	**2.0289** **2.0032**	1.9064 1.8678	2.4430 2.3753	1.9181 1.8242	**2.4475** **2.4010**
WFG46	1.7642 1.7148	1.8910 1.8784	1.6657 1.5997	**1.9155** **1.9056**	2.0698 2.0135	2.3211 2.3015	1.9831 1.9044	**2.3536** **2.3377**	2.4870 2.3920	2.8109 2.7661	2.3071 2.2160	**2.8461** **2.8225**
WFG47	1.6036 1.5072	1.6656 1.6500	1.4428 1.3639	**1.6698** **1.6535**	1.8537 1.6232	**2.0642** **2.0420**	1.6999 1.6145	2.0552 2.0328	2.0029 1.8890	**2.4939** **2.4536**	2.0005 1.8813	2.4918 2.4453
WFG48	2.0005 1.9786	2.0406 2.0337	1.8416 1.7771	**2.0427** **2.0367**	2.4268 2.4157	2.4640 2.4489	2.2423 2.1112	**2.4679** **2.4604**	2.9349 2.9120	2.9608 2.9309	2.6785 2.5407	**2.9668** **2.9518**