Dynamic Adaptation of Decomposition Vector Set Size for MOEA/D

Yuta Kobayashi
University of Tsukuba
Ibaraki, Japan
kobayashi.yuta.xu@alumni.tsukuba.ac.jp

Claus Aranha
University of Tsukuba
Ibaraki, Japan
caranha@cs.tsukuba.ac.jp

Tetsuya Sakurai
University of Tsukuba
Ibaraki, Japan
sakurai@cs.tsukuba.ac.jp

ABSTRACT

The Multi Objective Evolutionary Algorithm based on Decomposition (MOEA/D) is a popular algorithm for solving Multi Objective Problems (MOP). The main characteristic of MOEA/D is to use a set of weight vectors to break the MOP into a set of single-objective sub problems. It is well known that the performance of MOEA/D varies greatly depending on the number of weight vectors. However, the appropriate value for this hyper-parameter is likely to vary depending on the problem, as well as the stage of the search. In this study, we propose a robust MOEA/D variant that evaluates the progress of the search, and deletes or creates weight vectors as necessary to improve the optimization or to avoid search stagnation. The performance of the proposed algorithm is evaluated on the DTLZ and ZDT benchmark. We observed that the proposed method without needing to explicitly choose the number of weight vectors is equivalent to MOEA/D with fine tuned vectors and superior than MOEA/D with poorly tuned vectors.

CCS CONCEPTS

• **Theory of computation** → **Continuous optimization**; **Evolutionary algorithms**; • **Applied computing** → **Multi-criterion optimization and decision-making**;

KEYWORDS

MOEA/D, Auto Adaptation, Weight Vectors, Multi Objective Optimization

ACM Reference Format:
Yuta Kobayashi, Claus Aranha, and Tetsuya Sakurai. 2021. Dynamic Adaptation of Decomposition Vector Set Size for MOEA/D. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459453

1 INTRODUCTION

The set of weight vectors of MOEA/D [5] can be considered a hyper parameter, and the appropriate number of vectors is not known beforehand for most problems. Using a very low number of vectors may lead to search stagnation, while a very high number may lead to waste of computational resources.

Because of this issue, research has been done to define the appropriate set of weight vectors in MOEA/D. However, this research usually focuses on adjusting the position of weight vectors, and not in changing the number of vectors itself. To address this issue, in this study we focus on automatically adapting the number of weight vectors in MOEA/D, adding or deleting vectors automatically based on the progress of the search.

Our proposed method has two main components: How to identify the timing to add or remove weight vectors, and how to decide which vectors to add or remove. To identify the timing to add or remove vectors, we use the "Consolidation Ratio", which was originally proposed as a stopping criteria. To decide which vectors to add or remove, we use two strategies, random and AWA, depending on the timing of the search.

The proposed method was tested on the DTLZ and ZDT benchmark, and compared with MOEA/D with different population settings, as well as MOEA/D-AWA [3], a method which adjusts the positions of the weight vectors during the search.

2 PROPOSED METHOD

We propose a method to enhance MOEA/D by automatically adding or removing weight vectors as the search progresses. The outline of the method is described in Algorithm 1.

At every generation the algorithm check stagnation using CR method [2]. If stagnation is detected, it deletes weight vectors. The number of vectors added or removed at each update is a fraction of the total number of weight vectors (ratio in algorithm 1).

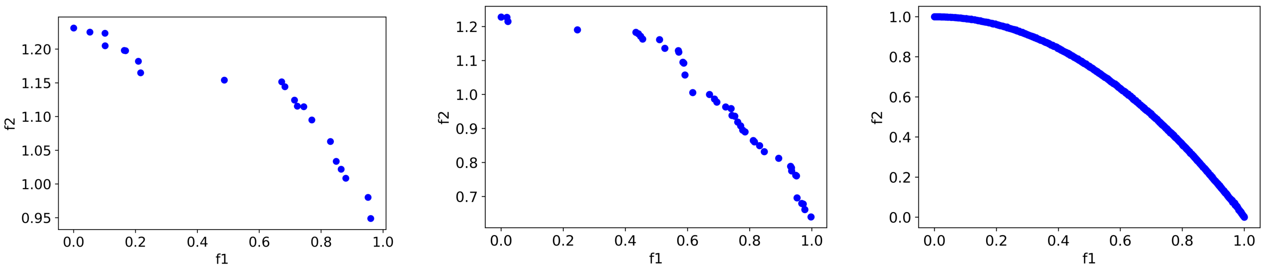
When adding new vectors, the method has a choice of adding vectors based on AWA, or adding random vectors. In our initial experiments, we noticed that using only AWA-based method to determine the position of new vectors led to early stagnation of the search. The solution that we found is that new vectors added early in the search have random positions. The choice of algorithm to calculate the vector controlled by this probability p which changes as the search progresses, calculated by

$$p = \frac{n\_fe}{n\_eval}, \tag{1}$$

where n_fe is the current number of function evaluations, and n_eval is the total evaluation budget. When creating a new weight vector, an AWA based vector is generated with probability p, otherwise a random weight vector is generated (probability $1 - p$). This means that, at the beginning of the search, a random weight vector is much more likely to be created, while at the end it is more likely to create a new weight vector using the AWA-based method.

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459453



(a) MOEA/D with 500 vectors (b) MOEA/D-AWA with 500 vectors (c) Proposed with 500 vectors

Figure 1: UEA of 3 methods starting from 500 vectors in ZDT2.

Besides adding weight vectors, the proposed method also delete weight vectors to avoid wasting computational resources when too many weight vectors exist. Currently, the weight vectors are deleted at random, excluding those weight vectors associated with the axis of each objective.

Algorithm 1 Proposed Adaptation method

Input: number of population **N**, number of adjustment ratio **ratio**, number of evaluation **n_eval**
Output: Unbounded External Archive **UEA**

if stagnation is detected by CR method **then**
  $p = \frac{n\_fe}{n\_eval}$
  nav = **ratio** $*$ **size**(**W**)
  if $p > random$ **then**
    $\mathbf{X'}^{(\mathbf{Gen+1})}$ = AWA-based Add(**UEA**, $\mathbf{X'}^{(\mathbf{Gen+1})}$, $\mathbf{z}^*$, **nav**)
  else
    $\mathbf{X'}^{(\mathbf{Gen+1})}$ = Random Add(**UEA**, $\mathbf{X'}^{(\mathbf{Gen+1})}$, $\mathbf{z}^*$, **nav**)
  end if
else
  $\mathbf{X'}^{(\mathbf{Gen+1})}$ = Delete Vector($\mathbf{X'}^{(\mathbf{Gen+1})}$, **nav**)
end if

3 EXPERIMENT

We perform an experiment to evaluate the robustness of the proposed method against the original MOEA/D, and the MOEA/D-AWA (which adjust the values of the weight vectors, but not their numbers). The three methods are compared with different number of initial weight vectors, to analyze their robustness to this parameter. We use the DTLZ benchmark set(3-objective, 10 dimensions)[1], and the ZDT set(2-objective, 30 dimensions)[6]. For the sake of evaluation fairness when comparing MOEAs with different population sizes, the algorithms were evaluated based on their Unbounded External Archive (UEA), and not their final population [4]. Our motivation was to use the parameters suggested in the original works of each method used.

4 DISCUSSION

As Fig 2, we show the change in the HV of the UEA that each method finally obtains against the number of weight vectors for ZDT2. Fig 2 show that the performance of the conventional methods deteriorates significantly when the number of weight vectors reaches around 500. Fig 1 shows the UEA of each method, and the UEA clearly shows that the performance of the proposed method is better. This is because the number of weight vectors in the conventional method is too large, and the computational resources allocated to each subproblem are not enough to approximate the Pareto front well. In the proposed method, the number of weight vectors is dynamically determined and deleted, which is considered to be an effective use of computational resources.

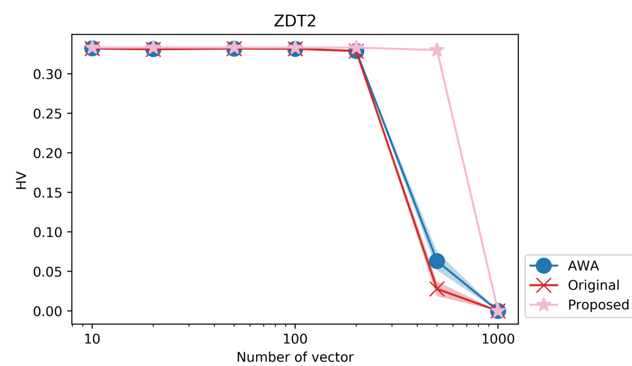


Figure 2: Mean HV value against initial number of weight vectors.

REFERENCES

[1] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2002. Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, Vol. 1. 825–830 vol.1. https://doi.org/10.1109/CEC.2002.1007032

[2] Tushar Goel and Nielen Stander. 2010. non-dominance-based online stopping criterion for multi-objective evolutionary algorithms. *Internat. J. Numer. Methods Engrg.* 88 (2010), 661–684.

[3] Yutao Qi, Xiaoliang Ma, Fang Liu, Licheng Jiao, Jianyong Sun, and Jianche We. 2014. MOEA/D with Adaptive Weight Adjustment. *Evolutionary Computation* 22, 2 (2014), 231–264.

[4] Ryoji Tanabe, Hisao Ishibuchi, and Akira Oyama. 2017. Benchmarking Multi- and Many-Objective Evolutionary Algorithms Under Two Optimization Scenarios. *IEEE Access* 5 (2017), 19597–19619.

[5] 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.

[6] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation* 8, 2 (2000), 173–195.