IJMSIT

International Journal of Multidisciplinary Studies and Innovative Technologies

e-ISSN: 2602-4888
dergipark.org.tr/en/pub/ijmsit
Research Article

2022, 6 (1), 11-18
DOI: DOI: 10.36287/ijmsit.6.1.11
Received: May16, 2022; Accepted: June12, 2022

Comparison of MOEA/D Variants on Benchmark Problems

Tolga Altinoz[1*]

[1*]*Department of Electrical and Electronics Engineering, Ankara University, Ankara, Turkey*
(taltinoz@ankara.edu.tr) (ORCID 0000-0003-1236-7961)

Abstract – Given that the definition of the multi-objective optimization problem is raised when number of objectives is increased in number at the optimization problem, where not only the number of objectives but also the computational resources which are needed to solve the problem, is also more desired. Therefore, novel approaches had required to solve multi-objective optimization problem in a reasonable time. One of this novel approach is utilization of the decomposition method with the evolutionary algorithm/operator. This algorithm was called multi-objective evolutionary algorithm based on decomposition (MOEA/D). Later on, variants have been proposed to improve the performance of the MOEA/D algorithm. However, a general comparison between these variants has needed for demonstrate the performance of these algorithm. For this reason, in this research the variants of MOEA/D algorithms have implemented on benchmark problems (DTLZ and MaF) and the performances has compared with each other. Two metrics had selected to evaluate/compare the performances of the variants. The metrics are IGD and Spread metrics. The results at the end of the implementations suggest that adaptive weighting idea is the most promising idea to increase the performance of the MOEA/D algorithm.

Keywords – *multi-objective optimization, evolutionary algorithm, decomposition, evolutionary algorithms*

Citation: Altinoz, O.T. (2022). Comparison of MOEA/D Variants on Benchmark Problems. International Journal of Multidisciplinary Studies and Innovative Technologies, 6(1): 11-18.

I. Introduction

The number of objectives in the optimization problem decides the complexity and the title of the problem set. İf the number of objectives is more than one, it is called multiobjective optimization problem. The definition of the multi-objective optimization problem is given as

$$\min \quad F(x) = \left(f_1(x) \ldots f_M(x)\right) \quad (1)$$
$$subject\ to\ x \in \Omega$$
$$g(x) \leq 0$$
$$h(x) = 0$$

where g(x) and h(x) are constraints with the decision vector $x \epsilon \Omega$ is the decision space and $F{:}\Omega \rightarrow R^M$ is the real valued objective space, where F is the objective function vector of real valued *f*. As the number of objectives is increase in number conventional methods for sorting the solutions with respect to the dominance become computationally costly and since the distance between dimensions on the objective space increase, it is hard to obtain a more diverse solutions on the objective space. Therefore, modern multiobjective optimization algorithms prefer alternative methods like decomposition.

Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [1] is an evolutionary multi objective optimization algorithm which is proposed by Zhang et al. in 2007. The algorithm is based on decomposition (Aggregation function). Decomposition stands for converting the multiobjective problem into many single objective sub-problems. The sub-problems composed from the solutions on the objective space and the pre-defined weight vectors. The diversity of the population is detected explicitly from these weight vectors. MOEA/D is an evolutionary multiobjective optimization algorithm so that the algorithm is built from Genetic operators; crossover, mutation and selection. Instead of the selection operator; other two operators are almost same at MOEA/D. SBX method is selected as the crossover operator at the algorithm. However, parents are selected from neighbouring members of the population. For this reason, two set of vectors are recorded for neighbouring data and weights for the decomposition. Polynomial mutation is selected as the mutation operator. Selection operator differs from evolutionary algorithms. Decomposition is applied to the solutions on the objective space by using a set of weight vectors. Then solutions are converted to the number of sub-problems. The sub-problem values of the offspring and parents are compared in number with respect to the minimization or maximization. Finally, the best members are survived to the next generation.

Even MOEA/D proves itself on many problems, still the performance of the algorithm needs to be improved to get a better distribution of the solutions with a better result with a faster or by using lower computational resources. Therefore, many variants are proposed to increase the performance of the algorithm. To discuss the performance of these variances, in this paper the performance of these variants on 22 benchmark problems with five objectives are compared with each other.

This paper is organized as four sections. After the introduction materials and methods used in this research is given. These are optimization algorithms, benchmark problems and metrics. Then implementation results are given numerically and finally the conclusion of the paper is given as the final section.

Table 1. DTLZ Benchmark Problems

	Mathematical Formulation
DTLZ1	$f_1 = \frac{1}{2}x_1x_2 \ldots x_{M-1}(1+g(x_M)) \ldots (1-x_{M-1})(1+g(x_M)) \ldots f_M = \frac{1}{2}(1-x_1)(1+g(x_M))$ $g(x_M) = 100\left[\lvert x_M \rvert + \sum_{i=1}^{M}\left(\left(x_i - \frac{1}{2}\right)^2 + cos\left(20\pi\left(x_i - \frac{1}{2}\right)\right)\right)\right]$
DTLZ2	$f_1 = (1+g(x_M))cos\left(x_1\frac{\pi}{2}\right) \ldots cos\left(x_{M-2}\frac{\pi}{2}\right)cos\left(x_{M-1}\frac{\pi}{2}\right) \ldots cos\left(x_{M-2}\frac{\pi}{2}\right)sin\left(x_{M-1}\frac{\pi}{2}\right) \ldots$ $f_M = (1+g(x_M))sin\left(x_1\frac{\pi}{2}\right) g(x_M) = \sum_{i=1}^{M}\left(\left(x_i - \frac{1}{2}\right)^2\right)$
DTLZ3	$f_1 = \left(1+g(x_M)\right)cos\left(x_1\frac{\pi}{2}\right) \ldots cos\left(x_{M-2}\frac{\pi}{2}\right)cos\left(x_{M-1}\frac{\pi}{2}\right) \ldots cos\left(x_{M-2}\frac{\pi}{2}\right)sin\left(x_{M-1}\frac{\pi}{2}\right) \ldots f_M = \left(1+g(x_M)\right)sin\left(x_1\frac{\pi}{2}\right)$ $g(x_M) = 100\left[\lvert x_M \rvert + \sum_{i=1}^{M}\left(\left(x_i - \frac{1}{2}\right)^2 + cos\left(20\pi\left(x_i - \frac{1}{2}\right)\right)\right)\right]$
DTLZ4	$f_1 = (1+g(x_M))cos\left(x_1^{100}\frac{\pi}{2}\right) \ldots cos\left({x^{100}}_{M-2}\frac{\pi}{2}\right)cos\left({x^{100}}_{M-1}\frac{\pi}{2}\right) \ldots f_M = \left(1+g(x_M)\right)sin\left({x^{100}}_1\frac{\pi}{2}\right), g(x_M) = \sum_{i=1}^{M}\left(\left(x_i - \frac{1}{2}\right)^2\right)$
DTLZ5	$f_1 = \left(1+g(x_M)\right)cos\left(\theta_1\frac{\pi}{2}\right) \ldots cos\left(\theta_{M-2}\frac{\pi}{2}\right)cos\left(\theta_{M-1}\frac{\pi}{2}\right) \ldots cos\left(\theta_{M-2}\frac{\pi}{2}\right)sin\left(\theta_{M-1}\frac{\pi}{2}\right) \ldots f_M = \left(1+g(x_M)\right)sin\left(\theta_1\frac{\pi}{2}\right)$ $\theta_i = \frac{\pi}{4\left(1+g(x_M)\right)}(1+2g(x_M)x_i), g(x_M) = \sum_{i=1}^{M}\left(\left(x_i - \frac{1}{2}\right)^2\right)$
DTLZ6	$f_1 = (1+g(x_M))cos\left(\theta_1\frac{\pi}{2}\right) \ldots cos\left(\theta_{M-2}\frac{\pi}{2}\right)cos\left(\theta_{M-1}\frac{\pi}{2}\right)$ $\ldots f_M = (1+g(x_M))sin\left(\theta_1\frac{\pi}{2}\right) \theta_i = \frac{\pi}{4\left(1+g(x_M)\right)}(1+2g(x_M)x_i), g(x_M) = \sum_{i=1}^{M}(x_i^{0.1})$
DTLZ7	$f_1 = x_1, f_2 = x_2 \ldots f_M = \left(1+g(x_M)\right)h g(x_M) = 1 + \frac{9}{\lvert x_M \rvert}\sum x_i, h = M - \sum_{i=1}^{M-1}\left(\frac{f_i}{1+g}(1+sin(3\pi f_i))\right)$

II. Materials and Method

In this section the variants (16) of MOEA/D are briefly given. Then benchmark problems are defined with the metrics that used to compare the performances of the algorithms.

A. Optimization Algorithms

MOEA/D Adaptive Weight Vector Adjustment (MOEA/D-AWA) [2]:

This paper aims at developing a method for weights of the decomposition process. It is stated on the paper that uniformly distributed weights for the decomposition cannot work well on the relatively complex Pareto front. At the problems with complex Pareto front, several sub-problems produce same optimal solution; causes wase of computational sources. Therefore, in this paper, a weight initialization method based on geometric relationship between weight vectors is proposed. For this reason, this variant is a special case solution to address the complex Pareto front problems. The idea behind the method is to find the crowding and sparse area of the objective space. Overpopulated subproblems are deleted and new subproblems are added to the sparse regions.

Covariance Matrix Adaptation-based MOEA/D (MOEA/D-CMA) [3]:

Covariance Matrix Adaptation is a classical method for solving single objective optimization algorithms. In conventional MOEA/D algorithm the operators from Genetic Algorithm like SBX and Polynomial Operator. Generally, instead of SBX, DE is selected as the optimizer however in this variant Covariance Matrix Adaptation method is selected and applied to the MOEA/D algorithm. The results showed that instead of DE, CMA presents better performance than DE.

Multi-objective Evolutionary Algorithm based on Dominance and Decomposition (MOEA/D-DD) [4]:

In this variant both decomposition and dominance ideas are joint to balance the convergence and diversity properties. The idea behind the paper is expending the MOEA/D performance for especially large number of objectives. In this method, widely spread weight vectors for both subproblem and subregion defines. The subregion is considered as niche of the population and the density of the population on this subregion is estimated. Therefore, most of the parents are selected from the neighbouring sub-regions (like in MOEAD-M2M). Instead of all offspring, only one offspring is considered for updating the population.

MOEA/D with Detect-and-Escape Strategy (MOEA/D-DAE) [5]:

This is an improved version of the MOEA/D algorithm especially for the constrained problems. It is stated that the conventional constraint handling method like ϵ-constraint method, is built based on to drag the solution from infeasible region to the feasible region. However, if the population map trap on this new region. Therefore, a new method called detect-and-escape strategy is proposed. Since the proposal is based on only for the constraint problems (even boundaries), it is not expected a better performance than MOEA/D under un-constrained (or boundary) problems.

MOEA/D with Differential Evolution (MOEA/D-DE) [6]:

The conventional MOEA/D algorithm is proposed to use SBX crossover, polynomial mutation, and mostly PBI aggregation function. In this variant instead of SBX crossover operator, Differential Evolution rules are applied to obtain offspring through the algorithm.

MOEA/D with Dynamical Resource Allocation (MOEA/D-DRA) [7]:

In this variant of MOEA/D, tournament selection operator is preferred. To assign different computational effort, the idea of Dynamic Resource Allocation is defined so that for ever fifty generation the index for determining the number of individuals for selection operator is updated.

Table 2. MaF Benchmark Problems

	Mathematical Formulation
MaF1	$f_1 = (1 - x_1 \dots x_{M-1})\dots(1 + g(x_M))\dots f_M = (x_1)(1 + g(x_M)), g(x_M) = \sum_{i=1}^{M}\left(\left(x_i - \frac{1}{2}\right)^2\right)$
MaF2	$f_1 = (1 + g(x_M))cos\left(\frac{\pi}{2}\left(\frac{x_1}{2} + \frac{1}{4}\right)\right)\dots cos\left(\frac{\pi}{2}\left(\frac{x_{M-1}}{2} + \frac{1}{4}\right)\right)\dots f_M = (1 + g(x_M))x_1 sin\left(\frac{\pi}{2}\left(\frac{x_M}{2} + \frac{1}{4}\right)\right)\ g(x_M) = \sum_{i=1}^{M}\left(\left(\frac{\pi}{2}\left(\frac{x_i}{2} + \frac{1}{4}\right) - \frac{1}{2}\right)^2\right)$
MaF3	$f_1 = \left[(1 + g(x_M))cos\left(x_1\frac{\pi}{2}\right)\dots cos\left(x_{M-2}\frac{\pi}{2}\right)cos\left(x_{M-1}\frac{\pi}{2}\right)\dots cos\left(x_{M-2}\frac{\pi}{2}\right)sin\left(x_{M-1}\frac{\pi}{2}\right)\right]^4\dots f_M = \left[(1 + g(x_M))sin\left(x_1\frac{\pi}{2}\right)\right]^2$ $g(x_M) = \left[100\lvert x_M\rvert + \sum_{i=1}^{M}\left(\left(x_i - \frac{1}{2}\right)^2 + cos\left(20\pi\left(x_i - \frac{1}{2}\right)\right)\right)\right]$
MaF4	$f_1 = a(1 + g(x_M))\left(1 - cos\left(x_1\frac{\pi}{2}\right)\dots cos\left(x_{M-2}\frac{\pi}{2}\right)cos\left(x_{M-1}\frac{\pi}{2}\right)\dots cos\left(x_{M-2}\frac{\pi}{2}\right)sin\left(x_{M-1}\frac{\pi}{2}\right)\right)\dots f_M = a(1 + g(x_M))\left(1 - sin\left(x_1\frac{\pi}{2}\right)\right)$
MaF5	$f_1 = a^m\left[(1 + g(x_M))cos\left({x^a}_1\frac{\pi}{2}\right)\dots cos\left({x^a}_{M-2}\frac{\pi}{2}\right)cos\left({x^a}_{M-1}\frac{\pi}{2}\right)\dots cos\left({x^a}_{M-2}\frac{\pi}{2}\right)sin\left({x^a}_{M-1}\frac{\pi}{2}\right)\right]^4\dots f_M = a\left[(1 + g(x_M))sin\left({x^a}_1\frac{\pi}{2}\right)\right]^4$
MaF6	$f_1 = (1 + g(x_M))cos(\theta_1)\dots cos(\theta_{M-2})cos(\theta_{M-1})\dots cos(\theta_{M-2})sin(\theta_{M-1})\dots f_M = (1 + g(x_M))sin(\theta_1)g(x_M) = \sum_{i=1}^{M}\left(\left(x_i - \frac{1}{2}\right)^2\right)$ $\theta_i = \begin{cases} \frac{\pi}{2}x_i, i = 1,2,\dots,l-1 \\ \frac{\pi}{4(1 + g(x_M))}(1 + 2g(x_M)x_i), i = l,\dots,M-1 \end{cases}$
MaF7	$f_1 = x_1, f_2 = x_2\dots f_M = (1 + g(x_M))h g(x_M) = 1 + \frac{9}{\lvert x_M\rvert}\sum x_i, h = M - \sum_{i=1}^{M-1}\left(\frac{f_i}{1+g}(1 + sin(3\pi f_i))\right)$
MaF8	Multi-Point Distance Minimization Problem $f_1 = d(x, A_1), f_2 = d(x, A_2), \dots f_M = d(x, A_M)$
MaF9	Multi-Line Distance Minimization Problem $f_1 = d(x, A_1A_2), f_2 = d(x, A_2A_3), \dots f_M = d(x, A_1A_M)$
MaF10	$f_1 = y_M + 2\left(1 - cos\left(y_1\frac{\pi}{2}\right)\right)\dots\left(1 - cos\left(y_{M-1}\frac{\pi}{2}\right)\right)f_M = y_M + 2M\left(1 - y_1 - \frac{cos\left(10\pi y_1 + \frac{\pi}{2}\right)}{10\pi}\right), z_i = \frac{x_i}{2i} for\ i = 1,\dots,D$ ${t^1}_i = \begin{cases} z_i\ if\ i = 1,\dots,K \\ \frac{\lvert z_i - 0.35\rvert}{\lvert 0.35 - z_i\rvert + 0.35} if\ i = K+1,\dots,D \end{cases}, {t^2}_i = \begin{cases} {t^1}_i \\ 0.8 + \frac{0.8(0.75 - t_i^1)min(0, [t_i^1 - 0.75])}{0.75} - \frac{0.2(t_i^1 - 0.85)min(0, [0.85 - t_i^1])}{0.15} \end{cases}$ ${t^3}_i = {{t^2}_i}^{0.02}, {t^4}_i = \begin{cases} \frac{\sum 2j{t^3}_i}{\sum 2j} \\ \frac{\sum 2j{t^3}_i}{\sum 2j} \end{cases}, y_i = \begin{cases} ({t^4}_i - 0.5)max(1, {t^4}_i) + 0.5 \\ {t^4}_M \end{cases}$
MaF11	$f_1 = y_M + 2\left(1 - cos\left(y_1\frac{\pi}{2}\right)\right)\dots\left(1 - cos\left(y_{M-1}\frac{\pi}{2}\right)\right)f_M = y_M + 2M(1 - y_1 cos^2(5\pi y_1)), z_i = \frac{x_i}{2i} for\ i = 1,\dots,D$ ${t^1}_i = \begin{cases} z_i\ if\ i = 1,\dots,K \\ \frac{\lvert z_i - 0.35\rvert}{\lvert 0.35 - z_i\rvert + 0.35} if\ i = K+1,\dots,D \end{cases}, {t^2}_i = \begin{cases} {t^1}_i \\ {t^1}_{K+2(i-K)-1} + {t^1}_{K+2(i-K)} + 2{t^1}_{K+2(i-K)-1} - {t^1}_{K+2(i-K)} \end{cases}$ ${t^3}_i = \begin{cases} \frac{\sum {t^2}_i}{K/M - 1} \\ \frac{\sum {t^2}_i}{D - K/2} \end{cases}, y_i = \begin{cases} ({t^3}_i - 0.5)max(1, {t^3}_i) + 0.5 \\ {t^3}_M \end{cases}$
MaF12	$f_1 = y_M + 2\left(1 - cos\left(y_1\frac{\pi}{2}\right)\right)\dots\left(1 - cos\left(y_{M-1}\frac{\pi}{2}\right)\right)f_M = y_M + 2M\left(1 - y_1 cos(\frac{\pi}{2}y_1)\right), z_i = \frac{x_i}{2i} for\ i = 1,\dots,D$ $y_i = \begin{cases} ({t^3}_i - 0.5)max(1, {t^3}_i) + 0.5 \\ {t^3}_M \end{cases}$
MaF13	$f_1 = sin\left(\frac{\pi}{2}x_1\right) + \frac{2}{J_1}\sum y_j^2, \dots, f_M = f_1^2 + f_2^{10} + f_3^{10} + \frac{2}{J_4}\sum y_j^2$
MaF14	$f_1 = x_1^f \dots x_{M-1}^f\left(1 + \sum c_{1,j}g_1\right), \dots, f_M = (1 - x_1^f)\left(1 + \sum c_{1,j}g_1\right)$
MaF15	$f_1 = \left(1 - cos\left(\frac{\pi}{2}x_1^f\right)\dots cos\left(\frac{\pi}{2}x_{M-1}^f\right)\right)\left(1 + \sum c_{1,j}g_1\right), \dots, f_M = 1 - sin\left(\frac{\pi}{2}x_1^f\right)\left(1 + \sum c_{1,j}g_1\right)$

MOEA/D with Distance-based Updating Strategy (MOEA/D-DU) [8]:

The motivation of the proposed method is to maintain the desired diversity of the population. For this purpose, the perpendicular distance between solution on the objective space and the weight vector calculate. For the multiobjective problems (three or less objective) the propose idea will fall behind the conventional decomposition algorithm. However, as indicated in the paper, the distance calculation (as a metric) is well suited for many objective optimization problems. Only the updating scheme differs from MOEA/D algorithm. First a new solution is produced (offspring), then perpendicular distance to all weight vectors is calculated. Then, some of these minimum distance solutions are selected. They are compared with neighbourhood solutions and replaced if objective value is smaller.

Dynamic Thompson Sampling for MOEA/D (MOEA/D-DYTS) [9]:

This paper is another variant of the MOEA/D-FRRMAB so that multi-armed bandit problem where the dynamic Thompson sampling (DYTS) is applied to adapt the bandit model. Therefore, to solve Multiarmed Bandit problem an alternative solution algorithm called Thompson sampling is integrated into MOEA/D algorithm.

Table 3. IGD Metric Value for DTLZ1-DTLZ7

Problem	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
MOEAD	6.8068e-2 (6.54e-5) =	2.1216e-1 (1.36e-4) =	2.1261e-1 (2.97e-4) =	4.0601e-1 (1.23e-1) -	2.6947e-2 (6.83e-4) +	2.6433e-2 (2.56e-3) +	1.0009e+0 (1.96e-1) +
MOEADDU	6.8935e-2 (3.83e-4) -	2.1578e-1 (8.37e-4) -	2.2094e-1 (1.53e-3) -	2.1772e-1 (1.30e-3) +	1.5516e-1 (1.28e-2) -	1.5026e-1 (2.21e-2) -	9.9328e+0 (3.17e+0) -
MOEADUR	6.9644e-2 (1.25e-3) -	2.1228e-1 (1.42e-3) =	2.3047e-1 (2.02e-3) -	3.2507e-1 (1.75e-1) -	2.8850e-2 (4.84e-3) +	3.1476e-2 (7.68e-3) +	4.8472e-1 (2.62e-2) +
MOEADURAW	6.7862e-2 (9.25e-4) =	2.1213e-1 (1.28e-3) =	2.3258e-1 (6.62e-3) -	3.8249e-1 (1.94e-1) -	6.4273e-2 (1.12e-2) +	7.6124e-2 (2.47e-2) +	3.0985e-1 (1.21e-2) +
MOEADD	6.8082e-2 (1.90e-5)	2.1221e-1 (2.29e-6)	2.1272e-1 (2.08e-4)	2.5670e-1 (9.38e-2)	8.5350e-2 (1.23e-2)	1.0025e-1 (1.49e-2)	3.0005e+0 (1.03e-6)

Table 4. IGD Metric Value for MaF1-MaF8

Problem	MaF1	MaF2	MaF3	MaF4	MaF5	MaF6	MaF7	MaF8
MOEAD	2.2409e-1 (2.62e-3) +	1.3617e-1 (1.40e-3) +	1.2579e-1 (1.84e-3) -	1.0244e+1 (4.70e-1) -	1.0849e+1 (4.26e+0) -	2.0358e-1 (2.93e-1) =	1.0947e+0 (8.50e-2) +	2.9575e-1 (1.15e-2) +
MOEADDU	2.5433e-1 (1.37e-2) =	1.3504e-1 (2.30e-3) +	9.8807e-2 (5.77e-4) +	4.3000e+0 (3.55e-1) +	2.6880e+0 (3.09e-2) +	5.0453e-2 (8.02e-4) +	1.0986e+1 (3.28e+0) -	8.0931e+2 (1.18e+3) =
MOEADUR	1.8549e-1 (4.58e-3) +	1.3334e-1 (2.80e-3) +	9.7018e-2 (1.11e-2) +	2.7576e+0 (6.68e-2) +	3.5005e+0 (1.33e+0) +	9.6127e-3 (6.54e-4) +	4.8641e-1 (3.43e-2) +	1.5651e-1 (7.76e-3) +
MOEADURAW	1.4938e-1 (8.90e-4) +	1.1949e-1 (2.00e-3) +	9.7627e-2 (1.01e-2) +	2.1795e+0 (1.79e-2) +	2.8346e+0 (1.30e+0) +	5.0413e-3 (4.99e-5) +	3.3438e-1 (4.94e-2) +	1.2899e-1 (2.28e-3) +
MOEADD	2.5944e-1 (1.61e-2)	1.5669e-1 (2.78e-3)	1.1565e-1 (3.26e-3)	7.6115e+0 (2.40e-1)	6.6025e+0 (5.30e-1)	8.2438e-2 (5.79e-3)	2.8126e+0 (5.94e-1)	3.7499e-1 (1.23e-2)

Table 5. IGD Metric Value for MaF9-MaF15

Problem	MaF9	MaF10	MaF11	MaF12	MaF13	MaF14	MaF15	+/-/=
MOEAD	1.4732e-1 (1.54e-3) +	7.8067e-1 (7.28e-3) -	7.6132e-1 (6.26e-3) -	1.8090e+0 (1.47e-1) -	1.8624e-1 (3.27e-2) +	6.5899e-1 (2.52e-1) =	6.8952e-1 (7.14e-2) -	9/8/5
MOEADDU	3.5747e-1 (1.36e-2) -	4.9382e-1 (5.72e-3) +	5.2448e-1 (9.48e-3) +	1.4959e+0 (3.57e-2) -	2.0459e-1 (1.30e-2) +	7.6645e-1 (1.04e-1) =	2.8804e+0 (7.90e-1) -	9/10/3
MOEADUR	1.8975e-1 (7.83e-3) +	5.7551e-1 (2.84e-2) =	5.0500e-1 (1.08e-2) +	1.2627e+0 (2.12e-2) +	1.8528e-1 (2.68e-2) +	7.5975e-1 (1.72e-1) =	5.5488e-1 (8.74e-2) =	15/3/4
MOEADURAW	2.0152e-1 (1.36e-2) +	5.0565e-1 (3.29e-2) +	5.5437e-1 (1.11e-2) +	1.1953e+0 (1.59e-2) +	1.5749e-1 (2.30e-2) +	9.0937e-1 (1.10e-1) -	5.8964e-1 (9.34e-2) =	16/3/3
MOEADD	2.9661e-1 (2.86e-3)	5.8604e-1 (3.35e-2)	5.8168e-1 (1.45e-2)	1.3438e+0 (1.31e-2)	2.8767e-1 (5.31e-2)	6.8320e-1 (1.52e-1)	5.1391e-1 (6.54e-2)	

MOEA/D Efficient Global Optimization (MOEA/D-EGO) [10]:

This algorithm is proposed especially solving the expensive optimization problems. This algorithm is based on efficient global optimization algorithm which has been widely accepted as one of the most popular methods using Gaussian Process Model. Through this algorithm, Gaussian model for each subproblem is built based on obtained data from previous generation. In this variation, sampled points select among the decision space. The function value for this sampled data calculates. Decomposed sub-problems and their objective values are collected and predictive model are generated. The idea behind this paper aims at developing a cost-efficient method, hence it is expected to present better performance for expensive optimization problems.

MOEA/D Fitness-rate-rank-based Multiarmed Bandit (MOEA/D-FRRMAB) [11]:

Adaptive operator selection is a method for deciding which operator should be employed in the MOEA/D algorithm. Two stage operator selection is to give reward to the operator, and at the second stage based on the reward operator is selected. The reward mechanism is related to the exploration vs. exploitation dilemma in multiobjective optimization problem similar to a common problem called multiarmed bandit problem. Therefore, upper confidence bound algorithm is employed to address this problem. The overall algorithm is named as Fitness-rate-rank-based Multiarmed Bandit MOEA/D algorithm. As the MOEA/D algorithm, instead of GA operators like SBX crossover and polynomial mutation, in this algorithm Differential Evolution (DE) operators (DE/rand/1, DE/rand/2, DE/current-to-rand/1, and DE/current-to-rand/2) are selected from this proposed method.

Number of Simple Multiobjective Subproblems-based MOEA/D (MOEA/D-M2M) [12]:

The decomposition method at the MOEA/D algorithm aims at developing such a method for obtaining many single-objective optimization problems/sub-problems. In this variant, instead of a many single objective problems (aggregated) number of simple multiobjective problems are considered and algorithm solves these problems in a collaborative way in a single run. In other words, the objective space divide into subregions and considered as multiobjective problems.

MOEA/D with Maximum Relative Diversity Loss (MOEA/D-MRDL) [13]:

At each generation of the MOEA/D algorithm the populations slowly (relatively) converge to the Pareto front. At each generation the direction of the convergency is differs and it causes loss of the diversity. Therefore, in the current algorithm the notion of maximum relative diversity loss. The idea is to compare the parent solution o any other parent solutions since they lead to poor convergence. Similarly, offspring are compared for similarity with respect to the estimated convergence direction.

Table 6. Spread Metric Value for DTLZ1-DTLZ7

Problem	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
MOEAD	3.5257e-2 (6.95e-4) -	1.7181e-1 (2.03e-3) +	1.6727e-1 (2.07e-3) +	7.0562e-1 (3.31e-1) =	1.6592e+0 (1.53e-2) =	1.7142e+0 (5.19e-2) -	1.0006e+0 (1.19e-1) -
MOEADDU	6.7552e-2 (1.32e-2) -	2.0470e-1 (5.23e-3) -	2.4238e-1 (1.43e-2) -	2.1815e-1 (8.52e-3) +	5.8727e-1 (4.29e-2) +	9.5654e-1 (7.61e-2) -	1.0005e+0 (9.35e-4) -
MOEADUR	3.1791e-1 (7.88e-2) -	1.8340e-1 (3.02e-2) =	5.4401e-1 (3.54e-2) -	4.8640e-1 (1.44e-1) -	1.3690e+0 (6.60e-2) +	1.4321e+0 (6.69e-2) -	8.3171e-1 (6.39e-2) +
MOEADURAW	2.7893e-1 (5.02e-2) -	1.6652e-1 (2.02e-2) =	6.3180e-1 (8.46e-2) -	2.3389e-1 (1.44e-1) =	3.7914e-1 (5.80e-2) +	6.3333e-1 (1.56e-1) +	3.7629e-1 (6.32e-2) +
MOEADD	3.3831e-2 (8.79e-5)	1.7482e-1 (1.38e-4)	1.7385e-1 (1.39e-3)	3.9945e-1 (4.74e-1)	1.5753e+0 (1.31e-1)	8.0050e-1 (1.16e-1)	1.0000e+0 (2.05e-11)

Table 7. Spread Metric Value for MaF1-MaF8

Problem	MaF1	MaF2	MaF3	MaF4	MaF5	MaF6	MaF7	MaF8
MOEAD	1.7038e+0 (5.77e-2) =	4.0448e-1 (5.82e-3) +	5.0758e-1 (1.81e-2) -	1.0782e+0 (1.20e-1) =	1.0244e+0 (1.92e-1) =	1.4441e+0 (3.09e-1) +	9.4827e-1 (1.67e-2) +	7.9903e-1 (3.85e-2) +
MOEADDU	9.4372e-1 (1.50e-1) +	3.9310e-1 (2.90e-2) +	4.1267e-1 (1.42e-3) +	1.2122e+0 (7.66e-1) =	3.6488e-1 (1.11e-2) +	1.6278e+0 (2.11e-1) +	1.0002e+0 (6.29e-4) +	9.1045e-1 (1.36e-1) +
MOEADUR	6.1989e-1 (4.08e-2) +	5.4407e-1 (4.34e-2) +	5.7743e-1 (8.75e-2) -	6.6258e-1 (7.46e-2) +	6.9546e-1 (6.20e-2) =	1.0101e+0 (6.17e-2) +	8.1426e-1 (5.85e-2) +	6.8274e-1 (6.15e-2) +
MOEADURAW	1.4181e-1 (1.83e-2) +	1.6477e-1 (1.13e-2) +	5.3471e-1 (7.11e-2) -	2.0803e-1 (5.31e-2) +	2.2215e-1 (1.35e-1) +	2.4994e-1 (2.95e-2) +	3.6599e-1 (6.53e-2) +	2.8329e-1 (5.16e-2) +
MOEADD	1.7955e+0 (1.46e-1)	1.2944e+0 (1.03e-1)	4.3287e-1 (1.09e-2)	1.1162e+0 (8.61e-2)	1.0100e+0 (3.44e-1)	2.2488e+0 (4.66e-1)	1.0233e+0 (7.38e-2)	1.0554e+0 (8.10e-2)

Table 8. Spread Metric Value for MaF9-MaF15

Problem	MaF9	MaF10	MaF11	MaF12	MaF13	MaF14	MaF15	+/-/=
MOEAD	5.4185e-1 (1.08e-2) +	6.5211e-1 (1.95e-2) -	5.9811e-1 (1.36e-2) -	5.0025e-1 (4.38e-2) -	1.2917e+0 (6.01e-2) +	1.0217e+0 (2.18e-1) =	8.5327e-1 (5.97e-2) +	9/7/6
MOEADDU	1.6584e+0 (2.05e-1) =	4.6757e-1 (5.27e-3) +	4.8288e-1 (5.41e-2) =	8.9825e-1 (1.09e-1) -	1.4527e+0 (4.53e-1) =	1.3956e+0 (3.46e-1) =	2.1147e+0 (1.64e-1) -	10/7/5
MOEADUR	9.5542e-1 (1.19e-1) +	4.9890e-1 (7.46e-2) =	5.0098e-1 (4.40e-2) =	3.6429e-1 (5.24e-2) =	1.0210e+0 (1.66e-1) +	2.0320e+0 (9.06e-2) -	1.2889e+0 (2.59e-1) =	10/6/6
MOEADURAW	1.5266e+0 (5.60e-1) =	6.6860e-1 (3.46e-2) -	4.3264e-1 (2.83e-2) +	1.4906e-1 (2.51e-2) +	1.3248e+0 (5.89e-1) =	1.5382e+0 (2.78e-1) -	6.9317e-1 (3.19e-1) +	13/5/4
MOEADD	1.6938e+0 (2.28e-1)	5.4218e-1 (4.93e-2)	4.8975e-1 (1.02e-2)	3.8770e-1 (1.32e-2)	1.5086e+0 (1.74e-1)	1.1115e+0 (5.10e-1)	1.1895e+0 (1.35e-1)	

Pareto Adaptive Scalarizing MOEA/D (MOEA/D-PaS) [14]:

In the paper initially all L_p methods for scalarization are evaluated and from the results it is observed that the parameter *p* is crucial for the performance of the algorithm for different Pareto geometries. The L_p aggregation function is calculated by *1/p* power of the sum of weighted difference between objective value and ideal points. For four different parameters set of p, the maximum value is selected as the aggregation function (PaS) of the algorithm. From the information at the paper, it is claimed that Pareto adaptive scalarization is a cost-efficient method that avoids the estimation of the Pareto front shape.

Stable Matching Method-based MOEA/D (MOEA/D-STM) [15]:

In this algorithm, the MOEA/D selection operator is considered as an operator for matching the subproblems with the solutions. Therefore, stable marriage problem is selected as a model problem. This stable matching model employ as a selection operator. Each subproblem has one solution in the current population. Subproblems are built based on aggregation functions, and the ranks elated to the objective value and the weight vector is obtained (similar to MOEA/D-DU). The matching algorithm (STM) is applied to these values to assign solutions to each subproblem.

MOEA/D with Updating when Required (MOEA/D-UR) [16]:

Similar to MOEA/D-AWA algorithm, MOEA/D-UR is a method for changing the weights. It is stated that uniformly distributed weights may be failed under complex (geometry of the problem) Pareto front. In the proposed method, a metric is defined and calculated to detect convergency, based on the value of this improvement metric/threshold (rate of offspring and parent vectors obtained from Tchebycheff decompostion), the objective space is divided adaptively to increase diversity.

MOEA/D with Uniformly Randomly Adaptive Weights (MOEA/D-URAW) [17]:

Similarly, also indicated in this variant, the shape and geometry of the Pareto front is the weakness of the MOEA/D algorithm due to the weight vector selection. In the proposed method, based on the sparsity of the population weight vectors are adapted by combining uniform random sampling with the adaptive weight vector selection. The flexible population size allows in this method. Also, the performance of the SBX over DE is indicated and SBX is suggested. Initially, the population sparsity level is calculated. Then external population stores non-dominated solutions during the search. Based on the highest sparsity level of this external population the new weight vector set is constructed.

Table 9. Runtime (sec) for DTLZ1-DTLZ7

Problem	DTLZ1	DTLZ2	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7
MOEAD	1.2361e+1 (1.10e-1) +	1.2337e+1 (1.33e-1) +	1.2598e+1 (1.48e-1) +	1.2693e+1 (9.64e-2) +	1.2839e+1 (9.73e-2) +	1.3047e+1 (9.69e-2) +	1.2639e+1 (7.64e-2) +
MOEADDU	1.6550e+1 (3.09e-1) +	1.6541e+1 (1.86e-1) +	1.6457e+1 (1.19e-1) +	1.6807e+1 (8.46e-2) +	1.7224e+1 (2.29e-1) +	1.7547e+1 (1.85e-1) +	1.6883e+1 (1.76e-1) +
MOEADUR	2.6046e+1 (6.46e-1) +	2.6015e+1 (1.73e-1) +	2.4563e+1 (2.00e-1) +	2.8987e+1 (6.07e-1) +	2.7793e+1 (1.71e-1) +	2.8689e+1 (5.25e-1) +	2.8769e+1 (2.87e-1) +
MOEADURAW	2.3460e+1 (2.40e-1) +	3.7011e+1 (8.03e-2) +	1.9754e+1 (2.38e-1) +	3.5192e+1 (1.24e+0) +	2.9633e+1 (5.52e-1) +	2.9077e+1 (5.26e-1) +	3.3074e+1 (1.18e-1) +
MOEADD	4.1776e+1 (3.69e-1)	5.5178e+1 (3.07e+0)	4.9959e+1 (5.22e-1)	5.7491e+1 (3.58e+0)	5.5168e+1 (8.00e-1)	5.7953e+1 (4.83e-1)	5.3916e+1 (4.49e-1)

Table 10. Runtime (sec)for MaF1-MaF8

Problem	MaF1	MaF2	MaF3	MaF4	MaF5	MaF6	MaF7	MaF8
MOEAD	1.2277e+1 (1.34e-1) +	1.2866e+1 (1.18e-1) +	1.2808e+1 (1.54e-1) +	1.3085e+1 (8.25e-2) +	1.4560e+1 (2.43e-1) +	1.4101e+1 (9.21e-2) +	1.7218e+1 (1.57e-1) +	1.9812e+1 (1.35e-1) +
MOEADDU	1.6660e+1 (1.42e-1) +	1.6930e+1 (8.42e-2) +	1.6893e+1 (1.66e-1) +	1.7442e+1 (1.99e-1) +	1.8572e+1 (2.38e-1) +	1.8824e+1 (1.74e-1) +	1.8632e+1 (1.61e-1) +	2.0431e+1 (1.66e-1) +
MOEADUR	2.9304e+1 (2.47e-1) +	3.0985e+1 (4.33e-1) +	2.8182e+1 (4.42e-1) +	2.8489e+1 (4.42e-1) +	3.2310e+1 (2.32e-1) +	3.1842e+1 (3.07e-1) +	3.3762e+1 (4.05e-1) +	3.6927e+1 (2.58e-1) +
MOEADURAW	3.2609e+1 (1.08e-1) +	3.5920e+1 (1.50e-1) +	2.0622e+1 (3.26e-1) +	2.3121e+1 (2.98e-1) +	3.9667e+1 (9.98e-1) +	2.4730e+1 (1.92e-1) +	4.0329e+1 (1.58e-1) +	2.5470e+1 (3.69e-1) +
MOEADD	5.1169e+1 (6.12e-1)	5.8239e+1 (6.23e-1)	4.8092e+1 (2.04e+0)	4.4284e+1 (6.76e-1)	5.8443e+1 (3.09e+0)	5.5572e+1 (2.15e+0)	5.6649e+1 (7.95e-1)	4.7841e+1 (6.17e-1)

Table 11. Runtime (sec)for MaF9-MaF15

Problem	MaF9	MaF10	MaF11	MaF12	MaF13	MaF14	MaF15	+/-/=
MOEAD	1.9057e+1 (1.78e-1) +	1.9914e+1 (1.43e-1) +	2.0716e+1 (1.67e-1) +	2.1227e+1 (1.25e-1) +	1.9136e+1 (1.83e-1) +	2.2750e+1 (1.40e-1) +	2.3080e+1 (2.35e-1) +	22/0/0
MOEADDU	1.9233e+1 (2.06e-1) +	2.3111e+1 (1.33e-1) +	2.4341e+1 (2.40e-1) +	2.4469e+1 (1.78e-1) +	2.3090e+1 (1.13e-1) +	2.6213e+1 (2.67e-1) +	2.6991e+1 (1.62e-1) +	22/0/0
MOEADUR	3.1236e+1 (2.57e-1) +	3.3810e+1 (3.19e-1) +	3.5421e+1 (2.84e-1) +	3.6056e+1 (2.82e-1) +	3.3453e+1 (2.65e-1) +	3.6629e+1 (3.15e-1) +	3.8039e+1 (4.37e-1) +	22/0/0
MOEADURAW	2.1531e+1 (3.28e-1) +	3.3416e+1 (3.58e-1) +	4.0231e+1 (2.55e-1) +	4.3848e+1 (2.43e+0) +	2.5660e+1 (7.51e-1) +	2.8468e+1 (3.12e+0) +	3.8587e+1 (1.21e+0) +	22/0/0
MOEADD	4.8611e+1 (6.66e-1)	6.0058e+1 (5.16e-1)	6.3404e+1 (4.25e-1)	6.3566e+1 (4.18e-1)	5.6062e+1 (9.73e-1)	6.4232e+1 (1.92e+0)	6.4383e+1 (7.30e-1)	

B. Benchmark Problems

In this research in total 22 benchmark problems are considered to implement the MOEA/D variants. These benchmark problems are defined in DTLZ [18] (seven benchmark problems) and MaF [19] (15 benchmark problems). Table 1 and Table 2 gives the test problems respectively.

C. Metrics and Statistical Tests

Unlike single objective optimization problems, a set of solutions are reported from optimization algorithm. The shape of the solutions on the objective space is called Prato approximated solutions. Therefore, some functions needed to extract the feature from this set. These functions are named as metrics. Two important properties are needed to observe for comparison of the algorithms. These properties are accuracy and distribution of the solutions on objective space.

For the accuracy, inverted generalized distance (IGD) metric is proposed in [20] and mathematical description of this metric is given as

$$f_{IGD} = \frac{\sum ds(a,P)}{|P|} \quad (2)$$

The IGD metric is based on computing the average distance between obtained solution candidates and the Pareto Front where $ds(a,P) = \sqrt{\sum(a_i - p_i)^2}$. The second metric is related to the distribution of the solution on the objective space. The spread metric is defined in [21], given as

$$f_{Spread} = \sqrt{\frac{1}{M}\sum\left(\frac{\max(a,PF)-\min(a,PF)}{PF_{max}-PF_{min}}\right)^2} \quad (3)$$

The metric is based on calculation of the normalized squared sum of the distance between maximum and minimum difference between produced solutions and PF.

III. IMPLEMENTATION AND RESULTS

In this research the variants of the MOEA/D algorithm are compared with respect to the accuracy and diversity of the solutions. For this purpose, two metrics are selected as IGD and Spread, respectively. Each variant is implemented independently 10 times with the same number of population size (100) and maximum number of function evolution (10^5) for 5 objective benchmark problems.

The variants of MOEA/D [1] algorithms are MOEA/D-AWA [2], MOEA/D-CMA [3], MOEA/D-DD [4], MOEA/D-DAE [5], MOEA/D-DE [6], MOEA/D-DRA [7], MOEA/D-DU [8], MOEA/D-DYTS [9], MOEA/D-EGO [10], MOEA/D-FRRMAB [11], MOEA/D-M2M [12], MOEA/D-MRDL [13], MOEA/D-PaS [14], MOEA/D-STM [15], MOEA/D-UR [16], and MOEA/D-URAW [17]. These algorithms had applied into benchmark problems. Among all these variants only MOEA/D [1], MOEA/D-DD [4], MOEA/D-DU [8], MOEA/D-UR [16], and MOEA/D-URAW [17] had given the comparative results. For this reason, only the results belonging to these five algorithms has reported on the paper. The statistical results for the mean and standard deviation of these independent runs are reported in Tables. Tables 3-5 are given for IGD metric, Tables 6-8 are for Spread metric and Tables 9-11 is presented for runtime of these algorithm.

IGD (Convergence): When all of the algorithms are compared with each other with respect to the number of the benchmark problems; URAW variant presents best performance for 22 benchmark problems. Original MOEA/D algorithm present best performance for 5 benchmark problems, similarly DU, DUR and DD variants present best performance for 3,3, and 1 benchmark problems, respectively. The results support the superior results of the URAW variant.

Spread (Distribution): Spread metric gives the distribution of the solutions on the objective space. Well distributed solutions are desired from the algorithms. Therefore, for the comparison on the spread metric, URAW presents best result of 14 of 22 benchmark problems. MOEAD, DU, DUR and DD variants only present best results from 3, 3, 1 and 1 benchmark problems, respectively.

Runtime: Since it is the main algorithm for the variants, MOEA/D gives the fastest results among all variants. However, if MOEA/D is removed from the results, DU gives the fastest results, meaning that it can be considered to use the lowest computational resources.

IV. CONCLUSION

The aim of this research is to compare the MOEA/D variants under 22 benchmark problems with five objectives. The results are evaluated on two metrics IGD and Spread. From the results, it is clearly demonstrated that URAW variant gives the best results almost all benchmark problems in both IGD and Spread metrics. In addition, the URAW variant uses relatively less computational resources when it is compared with other variants. The reason behind that is not only the adaptive weight vectors but also flexible population size. Therefore, both convergency and distribution property of the algorithm improves. It is suggested with respect to the results obtained in this paper, weights of the decomposition method and sparce detecting methods will increase the performance of the algorithm. Also, it is suggested to compare the performance of a novel algorithm with URAW variant.

Authors' Contributions

The authors' contributions to the paper are equal.

Statement of Conflicts of Interest

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The authors declare that this study complies with Research and Publication Ethics

REFERENCES

[1] 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.

[2] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," Evolutionary Computation, vol. 22, no. 2, pp. 231-264, 2014.

[3] H. Li, Q. Zhang, and J. Deng, "Biased multiobjective optimization and decomposition algorithm," IEEE Transactions on Cybernetics, vol. 47, no. 1, pp. 52-66, 2017.

[4] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," IEEE Transactions Evolutionary Computation, vol. 19, no. 5, pp. 694-716, 2015.

[5] Q. Zhu, Q. Zhang, and Q. Lin, "A constrained multi-objective evolutionary algorithm with detect-and-escape strategy," IEEE Transactions on Evolutionary Computation, vol. 24, no. 5, pp. 938-947, 2020.

[6] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 13, no. 2, pp. 284-302, 2009.

[7] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," Proceedings of the IEEE Congress on Evolutionary Computation, pp. 203-208, 2009.

[8] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," IEEE Transactions on Evolutionary Computation, vol. 20, no. 2, pp. 180-198, 2016.

[9] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," IEEE Transactions on Evolutionary Computation, vol. 18, no. 1, pp. 114-130, 2014.

[10] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," IEEE Transactions on Evolutionary Computation, vol. 14, no. 3, pp. 456-474, 2010.

[11] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," IEEE Transactions on Evolutionary Computation, vol. 18, no. 1, pp. 114-130, 2010.

[12] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," IEEE Transactions on Evolutionary Computation, vol. 18, no. 3, pp. 450-455, 2014.

[13] S. B. Gee, K. C. Tan, V. A. Shim, and N. R. Pal, "Online diversity assessment in evolutionary multiobjective optimization: A geometrical perspective," IEEE Transactions on Evolutionary Computation, vol. 19, no. 4, pp. 542-559, 2015.

[14] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods," IEEE Transactions on Evolutionary Computation, vol. 20, no. 6, pp. 821-837, 2016.

[15] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," IEEE Transactions on Evolutionary Computation, vol. 18, no. 6, pp. 909-923, 2014.

[16] L. R. de Farias, A. F. Araujo, "A decomposition-based many-objective evolutionary algorithm updating weights when required," Swarm and Evolutionary Computation, 2021.

[17] L. R. C. Farias and A. F. R. Araujo, "Many-objective evolutionary algorithm based on decomposition with random and adaptive weights," In Proceedings of the IEEE International Conference on Systems, Mans and Cybernetics, 2019.
[18] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," Evolutionary Multiobjective Optimization Theoretical Advances and Applications, pp. 105-145, 2005.
[19] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A Benchmark Test Suit for Evolutionary Many-objective Optimization," Complex Intell. Syst. Vol. 3, pp. 67–81, 2017.
[20] H. Ishibuchi H. Masuda Y. Tanigaki and Y. Nojima "Modified distance calculation in generational distance and inverted generational distance," in International Conference on Evolutionary Multi-Criterion Optimization. Springer, 2015, pp. 110–125.
[21] M. Ehrgott, "Approximation algorithms for combinatorial multicriteria optimization problems," International Transactions in Operational Research, vol. 7, no. 531, 2000