

Integrating User Preferences and Decomposition Methods for Many-objective Optimization

Asad Mohammadi*, Mohammad Nabi Omidvar*, Xiaodong Li* and Kalyanmoy Deb†

* School of Computer Science and IT
RMIT University, Melbourne, Australia

Emails: {asad.mohammadi, mohammad.omidvar, xiaodong.li}@rmit.edu.au

† Computational Optimization and Innovation Laboratory (COIN)
Michigan State University, Michigan, USA
Email: kdeb@egr.msu.edu

Abstract—Evolutionary algorithms that rely on dominance ranking often suffer from a low selection pressure problem when dealing with many-objective problems. Decomposition and user-preference based methods can help to alleviate this problem to a great extent. In this paper, a user-preference based evolutionary multi-objective algorithm is proposed that uses decomposition methods for solving many-objective problems. Decomposition techniques that are widely used in multi-objective evolutionary optimization require a set of evenly distributed weight vectors to generate a diverse set of solutions on the Pareto-optimal front. The newly proposed algorithm, R-MEAD2, improves the scalability of its previous version, R-MEAD, which uses a simplex-lattice design method for generating weight vectors. This makes the population size is dependent on the dimension size of the objective space. R-MEAD2 uses a uniform random number generator to remove the coupling between dimension and the population size. This paper shows that a uniform random number generator is simple and able to generate evenly distributed points in a high dimensional space. Our comparative study shows that R-MEAD2 outperforms the dominance-based method R-NSGA-II on many-objective problems.

I. INTRODUCTION

Evolutionary multi-objective optimization (EMO) algorithms have been successfully applied to many real-world problems in the past decade [1]. It has been shown that evolutionary multi-objective approaches can find evenly distributed and well-converged solutions on two or three objective problems [2]. However, when dealing with many-objective optimization problems which involve four or more objectives, the performance of EMO algorithms degrade rapidly [3], [4]. Hence, there is a need for developing EMO methods that can efficiently solve many-objective problems.

A number of challenges exist in solving many-objective optimization problems [3], [5]. Firstly, in a high dimensional objective space, even in the initial random population most of the solutions are non-dominated to each other. Thus, there would not be an adequate selection pressure making the search process very slow or even completely stagnant when a dominance-based algorithm is used. Secondly, generating solutions to approximate the entire Pareto front becomes computationally expensive [3], [4], [6]. Thirdly, it is difficult to visualize the Pareto-optimal front in large dimensions, which makes it difficult for the decision makers to select their preferred solutions. Adopting a user-preference based

approach can mostly alleviate the second problem. In a user-preference based approach the aim is to find a set of solutions on a smaller region of the Pareto-optimal front which is preferred by the decision maker. This technique requires less computational resources by performing a more focused and guided search rather than approximating the entire Pareto-optimal front. This is of practical value when dealing with many-objective problems. One popular type of user-preference based EMO algorithm is the *a priori* method, which allows a decision maker to provide the preference information (e.g. a reference point) at the beginning of the search. Reference point based [7], [8], reference direction based [9], and light beam based [10] EMO approaches are a few attempts in this area.

Another approach for better handling many-objective optimization problems is using decomposition-based methods. They convert a multi-objective problem into a set of single-objective problems. This feature makes them less sensitive to the selection pressure issue which is the prime source of the degrading performance of dominance-based approaches in dealing with many-objective problems. In other words, in dominance-based algorithms individuals (with many objectives) become non-dominated to each other, hence the selection pressure diminishes rapidly, making the population difficult move towards the Pareto-optimal front [11]. Various techniques for decomposing a multi-objective problem have been developed, including boundary intersection [12], [13], Tchebycheff and weighted-sum [14]. Some popular evolutionary EMO algorithms that employ such decomposition methods are MOGLS [15], and MOEA/D [16].

As explained before, utilizing user preference information and a decomposition-based approach can improve the efficiency of an EMO algorithm in solving many-objective problems. Also motivated by the fact that in most real-world situations users are not usually interested in finding the entire Pareto-optimal front and the availability of some form of preference information makes preference-based EMO approaches more applicable in real-world settings. This makes a decomposition-based algorithm that takes into account the preference information a promising approach for tackling many-objective problems. R-MEAD [17] is the first attempt in incorporating the user-preference information into a decomposition-based EMO algorithm. R-MEAD uses weighted-sum and Tchebycheff as decomposition methods and a *a priori* approach to search for preferred regions. How-

ever, it has only been applied to two and three-objective problems [17]. The main difficulty in applying R-MEAD to many-objective optimization lies in initializing a set of weight vectors. As it inherited the simplex-lattice design method from MOEA/D, the number of sample points is governed by the dimensionality of the problem. In this paper, we aim to develop an algorithm called R-MEAD2 which resolves this issue of R-MEAD. In short, R-MEAD2 extends the R-MEAD in the following aspects:

- 1) A new method for initializing the weight vector is used which makes R-MEAD2 scalable and applicable to many-objective optimization problems. This method decouples the population size from the number of objectives.
- 2) A uniform random number generator (ie., RNG) is used which simplifies the update mechanism of weight vector and make it more efficient.
- 3) The performance of R-MEAD2 is compared with the state-of-the-art R-NSGA-II [7].

Recently, an algorithm called UMOEA/D [18] is proposed which detaches the population size from the number of objectives. It replaces the simplex-lattice design with the good lattice point (GLP) [19]. This study shows that UMOEA/D generates more uniformly distributed solutions than MOEA/D using the simplex-lattice design. In Section IV-A we will show that when the number of objectives goes beyond eight a simple uniform random number generator(RNG), which is a commonly used technique in various programming languages, produces more uniformly distributed solutions than GLP according to a measure called centered L_2 -discrepancy [20]. In case of less than eight objectives RNG and GLP have similar performance in most cases. R-MEAD2 replaces the simplex-lattice design with a uniform random number generator (RNG). Employing RNG is not only simpler and more efficient than GLP, but also generates more uniformly distributed solutions when dealing with problems with more than eight objectives. Another improvement of R-MEAD2 over R-MEAD is related to updating weight vectors. In R-MEAD once the weight vectors are initialized their relative distances to each other stay the same. This is because a single descent direction vector is applied to all weight vectors which is calculated based on the location of the best and worst weight vectors. In R-MEAD2 a simple hill climber is used to generate mutants based on a uniform distribution to maintain the uniformity of the weight vectors. In each iteration, a set of new weight vectors are generated based on a uniform distribution within a certain region around the best weight vector that is found so far and the process is repeated until the weights converge to a solution. Finally, R-MEAD2 with two well-known decomposition methods namely PBI and Tchebycheff are compared with R-NSGA-II [7] to show the advantage of decomposition-based approaches over dominance based approaches.

The remainder of this paper is organized as follows. Section II introduces some preliminaries and a brief survey of previous studies is given. The proposed algorithm is described in section III. Section IV contains the experimental results and comparison between R-NSGA-II and R-MEAD2. The paper is

concluded in Section V.

II. PRELIMINARIES AND RELATED WORKS

A. Basic Definitions

A minimization multi-objective optimization problem with m objectives is defined as:

$$\text{minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})), \quad (1)$$

where $\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m$ is the objective vector and $f_i(\mathbf{x})$ is the i th objective function. Each decision vector $\mathbf{x} \in \mathbb{R}^n$ is defined as (x_1, \dots, x_n) where n is the number of dimensions in decision space. A multi-objective problem with more than three objectives is commonly referred to as a many-objective problem.

Dominance concept can be introduced to multi-objective optimization for relative comparison of two solutions. In a minimization problem \mathbf{x}_1 dominates \mathbf{x}_2 which is shown as $\mathbf{x}_1 \prec \mathbf{x}_2$ if:

$$\forall i \exists j (f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \wedge f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)),$$

where $i, j \in \{1, \dots, m\}$. Pareto-optimal set (PS) is a set of non-dominated solutions (decision space) which dominate the other solutions. When PS is plotted in objective space the non-dominated vectors are collectively known as the Pareto-optimal front (PF).

B. Decomposition Approaches

There are several techniques to convert a multi-objective problem to a set of single-objective problems. Here, we discuss two most widely used approaches Tchebycheff (Te) [14], and penalty-based boundary intersection (PBI) [16].

1) *Tchebycheff Approach*: Tchebycheff method is formulated as follows:

$$\text{minimize } (g^{\text{tch}}(\mathbf{x}, \mathbf{w}, \mathbf{z}^*) = \max\{w_i | f_i(\mathbf{x}) - z_i^*|\}), \quad (2)$$

where $\mathbf{z}^* \in \mathbb{R}^m$ and is the ideal point, and $\mathbf{w} = (w_1, \dots, w_m)$ is a weigh vector, which is positive. For each Pareto optimal point x^* , there is at least a weight vector \mathbf{w} such that x^* is optimal solution of equation (2). The effect of the weight vector is depicted in Figure 1.

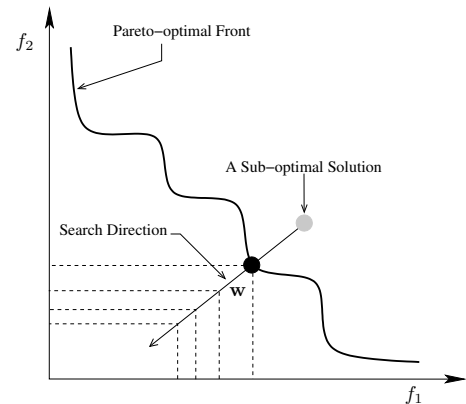


Fig. 1. Illustration of the Tchebycheff method.

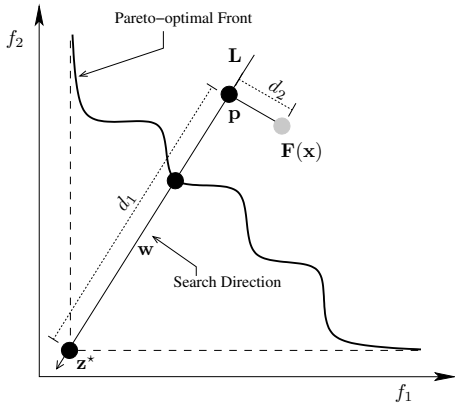


Fig. 2. Illustration of the PBI method.

2) Penalty-Based Boundary Intersection Approach (PBI):

This decomposition method is an improved version of normal boundary intersection (NBI) [12]. Unlike NBI that can only handle equality constraints, PBI [16] can handle both equality and inequality constraints. PBI is formulated as follows:

$$\text{minimize } g^{\text{pbi}}(\mathbf{x}, \mathbf{w}, \mathbf{z}^*) = d_1 + \theta d_2, \quad (3)$$

$$\text{where } d_1 = \frac{\|(\mathbf{F}(\mathbf{x}) - \mathbf{z}^*)^T \mathbf{w}\|}{\|\mathbf{w}\|}$$

$$\text{and } d_2 = \|\mathbf{F}(\mathbf{x}) - (\mathbf{z}^* + d_1 \mathbf{w})\|.$$

As shown in Figure 2, L is a line passing through \mathbf{z}^* with the direction of \mathbf{w} and \mathbf{p} is the projection of $\mathbf{F}(\mathbf{x})$ on L . The penalty parameter θ forces $\mathbf{F}(\mathbf{x})$ to be as close as possible to L (penalizing d_2). In this paper, the value of θ is set to 5 as suggested in [16].

C. User-preference Based EMO Algorithms

This section gives a brief survey of user-preference based EMO algorithms.

Guided multi-objective evolutionary algorithm (G-MOEA) [21], proposed by Branke *et al.*, allows the user to specify the linear trade-off between objectives. For example, in a two objective problem the decision maker has to specify how many units of the first objectives he/she is willing to trade for one unit of the second objective. G-MOEA then uses these trade-off information to guide the search towards the more desired regions of the Pareto-optimal front. Although G-MOEA is more flexible and intuitive than other approaches, it is not always an easy task for the decision maker to specify the trade-off between objectives, especially for many-objective problems. Deb [22] applied bias sharing technique on NSGA [23] where the biased Pareto-optimal solutions are generated on a desired region by changing the weights. An objective with a higher priority takes a higher weight value. The main disadvantage of this technique is that it cannot find solutions on a compromise region where all objectives are of similar importance to the decision maker. In another study, Branke and Deb [24] improved the idea of biased sharing and compared its performance with G-MOEA. They proposed biased crowding distance in NSGA-II which

has more flexibility than biased sharing in terms of finding solutions inside the region of interest.

Deb *et al.* [7] proposed a method that integrated user-preference information into NSGA-II [25]. The new method which was called R-NSGA-II requires the decision maker to provide one or more reference points at the beginning of the search process. In R-NSGA-II a modified version of crowding distance operator [25] *preference distance* was used to favor the solutions which are closer to the reference point(s). To compute the preference distance, the Euclidean distances of all solutions to the reference point(s) are calculated and the solutions with lower distance are ranked higher. To maintain the diversity of solutions in the desired regions an extra parameter ϵ was introduced.

Wickramasinghe and Li [26] integrated reference point and light beam with particle swarm optimization (PSO). The new approach, which is based on a distance metric, changes the position of particles based on the user supplied information to find the preferred regions. This distance-metric based method was compared with another user-preference based EMO (NSPSO) [27] which uses dominance-based comparison and it was shown that the distance metric approach performed better than NSPSO.

The above approaches use reference points as preference information. However, the preference information can also be obtained in terms of reference direction and light beam. Deb and Abhishek [9] applied reference direction on NSGA-II. In this new approach, RD-NSGA-II, user provided one or more reference directions. The procedure finds a set of Pareto-optimal solutions corresponding to the reference directions. In another study, Deb and Kumar [10] applied light beam search on NSGA-II. Decision maker provides aspiration, reservation and preference threshold for each objective. To control the density of solutions the parameter ϵ is used.

Most of the algorithms discussed so far are dominance-based and are not suitable for many-objective optimization. In this paper, we propose a scalable decomposition-based user-preference algorithm and we compare its performance with R-NSGA-II which is the state-of-the-art user-preference based EMO algorithm.

III. PROPOSED APPROACH

This section describes the R-MEAD2 algorithm which is a reference point based evolutionary algorithm that uses decomposition methods for solving many-objective optimization problems.

The population size of algorithms such as MOEA/D and R-MEAD that rely on a simplex-lattice design grows dramatically as the number of objectives increases. More precisely, the weight values w_1, w_2, \dots, w_m are chosen from the set $\{\frac{0}{H}, \frac{1}{H}, \dots, \frac{H}{H}\}$ where H is a parameter chosen by the user. Therefore, the number of these vectors and consequently the population size is calculated by the following formula: $\binom{H+m-1}{m-1}$. Table I shows how the population size grows with number of objectives when the simplex-lattice design is used to generate the weight vectors. As it can be seen, most of these population sizes are not commonly used in evolutionary algorithms.

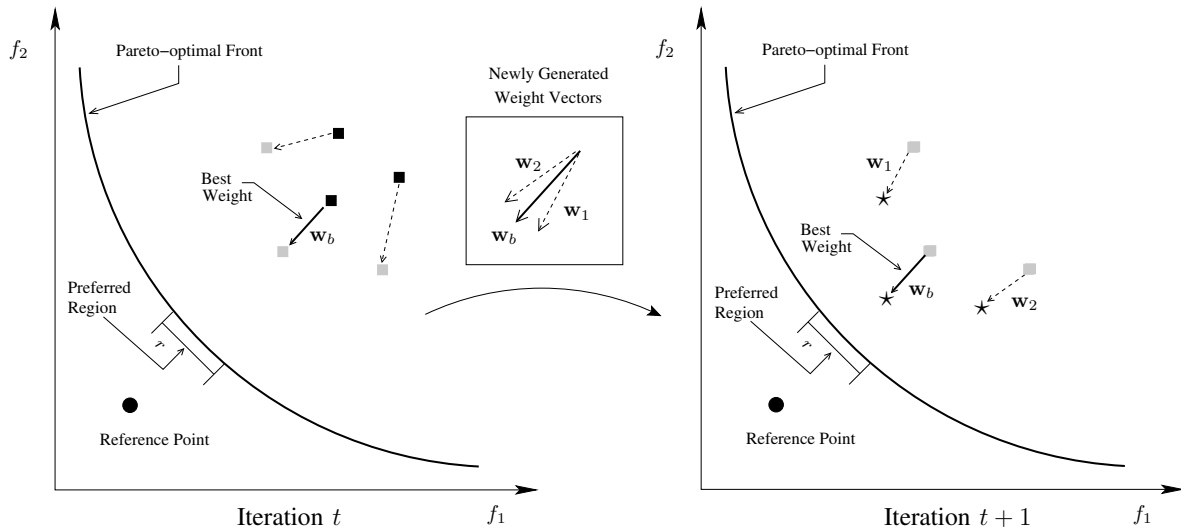


Fig. 3. Illustration how weight vectors are updated in RMEAD-2

To remove the coupling between the population size and the number of objectives, UMOEA/D [18] replaced the simplex-lattice design with the good lattice point (GLP) in MOEA/D [16] for generating the weight vectors. However, as shown in Section IV-A, GLP does not necessarily have a better uniformity as compared to a uniform random number generator (RNG) when the number of objectives grows beyond eight. Additionally, using RNG for generating the weight vectors also removes the coupling between the population size and the number of objectives. This allows the user to pick any suitable population size irrespective of the number of objectives.

A. The R-MEAD2 Algorithm

In the proposed decomposition-based user-preference approach in order to guide the solutions towards the desired region, the weight vectors should be dynamically updated so that the solutions can converge in the direction of the reference point, and ideally on the Pareto-optimal front. To achieve this effect each solution is assigned to a weight vector which is updated during the course of optimization.

Figure 3 shows how the weights are updated during the optimization. On the left the black squares denote the solutions at some iteration t . The gray squares represent the solutions after running an iteration of the evolutionary optimizer. The arrows show the update direction which is determined by the weight vectors associated with each individual. At this stage the weight vector associated with the closest solution to the reference point is marked as the best weight vector (w_b). Once the best weight vector is identified, a set of new weight vectors is generated using RNG within a region centered around the best weight vector. This forms a hypercube in an m dimensional space and the size of the region is determined by parameter r which is the edge size of hypercube. The weight

vectors w_1 and w_2 are generated with a uniform distribution around w_b as shown in the box at the center of Figure 3. Next, these weights are assigned back to the solutions that were obtained in the last iteration. Finally, the solutions are optimized with the newly assigned weights. This process is shown on the right-hand side of Figure 3. As it can be seen, the new solutions marked by ‘*’ gradually converge within a confined region in the direction of the reference point.

Algorithm 1 shows the details of the proposed method. The main steps are outlined below:

Step 1 - Initialization: The initial population is created, and the weight vectors are initialized using `rng` and `init_weights` functions respectively (lines 1-2). The initial weight vectors are generated using RNG over the entire space of weight vectors in order to increase the probability of finding a good initial weight vector. Also the weights are normalized so that the components of a weight vector add up to one.

Step 2 - Main Evolutionary Cycle: The population is evolved and the weight vectors are updated until a stopping criterion is met (lines 3-10).

Step 2.1 - Evolving the Population: The population is evolved to find better solutions. The `evolve` function first uses a decomposition method as specified by the `dm` variable to convert the multi-objective problem $F(\mathbf{x})$ to a set of single-objective problems similar to MOEA/D [16]. Then, it applies several genetic operations in order to evolve the population (lines 4). Finally, the new population is evaluated using the original objective function $F(\mathbf{x})$ to find a set of solutions (PF) in the objective space (line 5).

Step 2.2 - Updating weights: In order to update the weight vectors, the best weight vector which is associated to the solution with the shortest Euclidean distance to the reference point, has to be found. The `euclid_dist` function calculates the distance of all solutions to the reference point \mathbf{R} . Then the `min_ind` function finds the index of the closest solution to the reference point which matches the index of the best weight vector. The closest solution helps the convergence of solutions in the direction of reference points. Finally, the

TABLE I. POP-SIZE FOR R-MEAD FOR DIFFERENT OBJECTIVES ($H = 10$).

# Objs (m)	4	5	6	7	8	9	10
Pop-size	286	1001	3003	8008	19448	43758	92378

Algorithm 1: R-MEAD2

Inputs:

$popsize$: the initial population size.
 r : determines the size of the preferred region.
 m : number of objectives.
 n : number of decision variables.
 dm : the decomposition method (Tchebycheff or PBI).
 \mathbf{R} : the reference point.
 $\mathbf{F}(\mathbf{x})$: the objective function.

Variables:

\mathbf{P} : population of individuals ($popsize \times n$).
 \mathbf{W} : matrix of weight vectors ($popsize \times m$).
 \mathbf{PF} : solution set ($popsize \times m$).

1. $\mathbf{P} \leftarrow rng(lbounds, ubounds, popsize, n)$
2. $\mathbf{W} \leftarrow init_weights(popsize, m)$
3. **while** stop criteria not met **do**
4. $\mathbf{P} \leftarrow evolve(\mathbf{P}, \mathbf{F}(\mathbf{x}), \mathbf{W}, dm)$
5. $\mathbf{PF} \leftarrow evaluate(\mathbf{P}, \mathbf{F}(\mathbf{x}))$
6. $\mathbf{d} \leftarrow euclid_dist(\mathbf{PF}, \mathbf{R})$
7. $best \leftarrow min_ind(\mathbf{d})$
8. $\mathbf{w}_b \leftarrow \mathbf{W}[best, :]$
9. $\mathbf{W} \leftarrow uniform(popsize, r, \mathbf{w}_b)$
10. **end while**

uniform function uses RNG to generate a set of uniform weight vectors around the best weight vector \mathbf{w}_b (lines 7-9). The process repeats from Step 2.1.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This section includes three main experimental settings. Firstly, the uniformity of GLP and RNG is compared to measure which method is more suitable for many-objective problems. Secondly, the convergence behavior of weight vectors on different reference points are analyzed. Finally, based on the parameter setting in section IV-B the performance of R-MEAD2 using Tchebycheff and PBI on DTLZ1-DTLZ6 benchmark problems is compared with R-NSGA-II. All algorithms are tested on problems with 4 to 10 objectives.

A. RNG vs GLP

In order to compare the performance RNG and GLP, we use a discrepancy measure called centered L_2 -discrepancy (CD_2) [20]. Since we are interested in the uniformity of sample points, a lower value of CD_2 implies better uniformity. Let \mathbf{P} be a $n \times m$ matrix of sample points where the sample point $\mathbf{x}_i = (x_{i1}, \dots, x_{im})$ is the i th row vector in matrix \mathbf{P} , and m is the dimensionality of the sample points that, in this context, matches the number of objectives. Centered L_2 -discrepancy [20] is defined as follows:

$$CD_2(\mathbf{P}) = \left(\frac{13}{12}\right)^m - \frac{2}{n} \sum_{k=1}^n \prod_{j=1}^m \left(1 + \frac{1}{2}|x_{kj} - \frac{1}{2}| - \frac{1}{2}|x_{kj} - \frac{1}{2}|^2\right) + \frac{1}{n^2} \sum_{k=1}^n \sum_{j=1}^n \prod_{i=1}^m \left[1 + \frac{1}{2}|x_{ki} - \frac{1}{2}| + \frac{1}{2}|x_{ji} - \frac{1}{2}| + \frac{1}{2}|x_{ki} - x_{ji}|\right],$$

Table II contains CD_2 values for RNG and GLP using different sample sizes and dimensions (number of objectives m). As it can be seen, in case of all population sizes (50-5000) when the number of objectives grows beyond eight,

RNG is consistently better than GLP. For typical population sizes such as 100, 250, and 500 which are commonly used for solving many-objective problems, RNG is better when the number of objectives is more than seven. Table II shows that in case of lower number of objectives (less than eight) when the population size is relatively small, then GLP can be beneficial. GLP can be useful when solving problems with expensive objective function evaluation where a small population size is more practical. It is easy to see that when a small number of sample points are allowed, then a more systematic approach such as GLP might be advantageous. However, when the number of sample points grow, GLP and RNG may perform similarly. This behavior can be observed in Table II when the sample size is 1000 and 5000. It is clear that in majority of cases, both RNG and GLP have similar CD_2 values.

B. Parameter Settings and Performance Metrics

We used the population size of 200 for 4-7 objective problems, 300 for 8-objectives and 350 for 9 and 10-objective problems. A single reference point is used for all test problems ($f_i = 0.25, i \in \{1, \dots, m\}$). In R-MEAD2-Te and R-MEAD2-PBI the parameter r which determines the size of the preferred region is set to 2, and in R-NSGA-II the parameter ϵ is set to 0.002.

In order to compare the performance of R-MEAD2-PBI, R-MEAD2-Te and R-NSGA-II we have adopted inverted generational distance (IGD) [28] that measures both convergence and diversity of solutions. The calculation of IGD is based on the average closest distances between sample points on the Pareto-optimal front and the obtained solutions. We used 10^m sample points for four, five, six and seven objective problems and 5^m for eight, nine and ten objective problems, where m is the number of objectives. It should be noted that because of memory limitations, fewer number of samples points were generated to approximate the Pareto-optimal front for problems with more than seven objectives. IGD is calculated as follows:

$$IGD(P^*, Q) = \frac{\sum_{v \in P^*} d(v, Q)}{|P^*|}. \quad (4)$$

where Q is the set of obtained solutions and P^* is the set of sample points on the Pareto-optimal front, and function $d(v, Q)$ returns the Euclidean distance between a point v and the closest point to it in a set Q . Since all algorithms used in this study are user-preference based, we only consider the solutions that fall within a desired region. The desired region is a hypersphere with radius ρ around the sample point on the Pareto-optimal front which is closest to the reference point. For all experiments in this study the parameter ρ is set to 2.

C. Weight Vector Convergence

Figure 4 shows the convergence behavior of weight vectors for both PBI and Tchebycheff on DTLZ1 using two different reference points. As it can be seen the weight values fluctuate at the beginning of the search, but they gradually stabilize and converge to a fixed value towards the end of a run. Depending on the position of a reference point, weight vectors converge to different values. For example, when $\mathbf{R} = (0.25, 0.25)$ which is near the center of the Pareto-optimal front, both weight values converge to a value close to 0.5. However, when the reference point is biased towards a particular objective (e.g.

TABLE II. THE AVERAGE CD_2 VALUE OF 25 INDEPENDENT RUNS FOR TWO INITIALIZATION METHODS: A UNIFORM RANDOM NUMBER GENERATOR(RNG), AND GOOD LATTICE POINT (GLP) FOR DIFFERENT OBJECTIVES. THOSE LEAD TO STATISTICALLY SIGNIFICANTLY LOWER CD_2 VALUE (AT SIGNIFICANT LEVEL OF 5% ACCORDING TO MANN-WHITNEY-WILCOXON (MWW) TEST) OVER OTHERS ARE HIGHLIGHTED IN BOLD

# Objectives	Sample Sizes											
	50		100		250		500		1000		5000	
	RNG	GLP	RNG	GLP	RNG	GLP	RNG	GLP	RNG	GLP	RNG	GLP
4	2.74	2.71	2.73	2.73	2.75	2.74	2.75	2.75	2.75	2.75	2.75	2.75
5	2.99	2.94	2.97	2.96	2.98	2.97	2.98	2.98	2.98	2.98	2.98	2.98
6	3.21	3.19	3.23	3.22	3.24	3.22	3.23	3.23	3.22	3.23	3.23	3.22
7	3.48	3.47	3.51	3.50	3.50	3.49	3.49	3.50	3.50	3.50	3.50	3.50
8	3.82	3.76	3.78	3.81	3.78	3.81	3.78	3.81	3.80	3.79	3.79	3.79
9	4.12	4.14	4.12	4.15	4.11	4.14	4.11	4.14	4.10	4.14	4.11	4.11
10	4.48	4.48	4.47	4.54	4.46	4.50	4.46	4.50	4.45	4.50	4.45	4.45
11	4.91	4.92	4.86	4.94	4.83	4.89	4.82	4.90	4.83	4.89	4.82	4.87
12	5.33	5.43	5.27	5.41	5.24	5.32	5.23	5.32	5.24	5.31	5.22	5.30
13	5.83	6.01	5.73	5.94	5.68	5.69	5.69	5.79	5.67	5.78	5.66	5.75
14	6.46	6.70	6.30	6.58	6.16	6.17	3.16	6.23	6.15	6.29	6.14	6.23
15	6.97	7.50	6.80	7.33	6.73	6.86	6.69	6.71	6.66	6.76	6.64	6.76

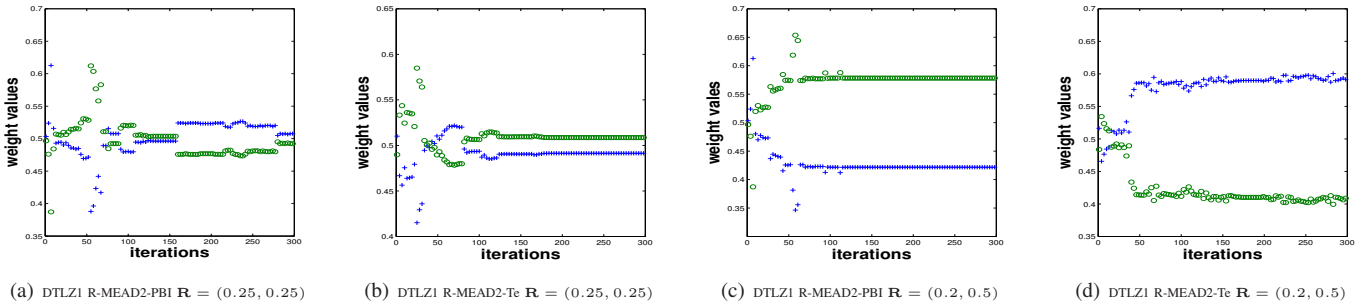


Fig. 4. Weight vector convergence behavior on 2-objective DTLZ1 using PBI and Tchebycheff. Objective-1 is shown by ‘+’ and objective-2 by ‘O’.

$\mathbf{R} = (0.2, 0.5)$) the weight values differ as it can be seen in Figures 4(c) and 4(d).

D. Benchmark Results

In this section we compare the performance of R-MEAD2 based on two decomposition methods Tchebycheff and PBI (abbreviated as R-MEAD2-Te and R-MEAD2-PBI respectively) with R-NSGA-II which is a dominance-based approach.

To test the significance of the obtained results we used the non-parametric Kruskal-Wallis one-way ANOVA [29] to detect if there is any significant difference between the performance of the three algorithms. The null and alternative hypotheses for Kruskal-Wallis are as follow:

H_0 : all samples come from the same distribution.

H_a : at least one sample comes from a different distribution.

In order to rank the algorithms, we used Mann-Whitney-Wilcoxon (MWW) test with Bonferroni correction only when the null hypothesis of Kruskal-Wallis was rejected under 95% confidence interval. Bonferroni correction is a simple technique for controlling the family-wise error rate [29]. Family-wise error rate is the accumulation of type I errors when more than one pair-wise comparison is used to rank a set of results. Under Bonferroni correction, in order to achieve an overall significance level of α , the pair-wise tests should be performed with a significance level of $\alpha' = \frac{\alpha}{h}$, where h is the number of pair-wise comparisons which is three in this study. For our experiments, the significance level of Kruskal-Wallis

was set to 5%, and for pair-wise MWW tests a significance level of 1.67% was used, which results in an overall significant level of approximately 5%. Table III contains the median for 25 independent runs for the three algorithms. The last three columns show the p -value for three MWW pair-wise tests and the column labeled ‘K-W’ contains the p -value for the Kruskal-Wallis ANOVA test.

Table III shows IGD results for all three algorithms. It can be observed that the difference between algorithms is significant for all functions and over all numbers of objectives. So in all cases MWW test is used and the best performing algorithm is shown in bold. If the performance of two algorithms are statistically similar then both entries are shown in bold. The table shows that in general a decomposition approach is superior to the dominance-based approach, R-NSGA-II. For the sake of clarity we have summarized the comparison between R-NSGA-II and both versions of R-MEAD2 in Table IV. From a total of 42 experiments, R-MEAD2-Te outperforms R-NSGA-II on 37 functions and ties on 2, and R-MEAD2-PBI outperforms R-NSGA-II on 35 functions and ties on 2. By looking back at Table III we can see that R-NSGA-II outperforms R-MEAD-PBI on DTLZ1 when the number of objectives is below 9. However, R-NSGA-II on DTLZ1 with 9 and 10 objectives, is outperformed by R-MEAD-PBI. Although R-NSGA-II in some objectives of DTLZ1 and DTLZ6 performs better than R-MEAD-PBI and R-MEAD-TE, in case of other test problems (DTLZ2-DTLZ5) R-NSGA-II is outperformed by both versions of R-MEAD. It should be noted that IGD takes both convergence and

TABLE III. IGD VALUES ON DTLZ1-DTLZ6 TEST PROBLEMS. THE MEDIAN OF 25 INDEPENDENT RUNS ARE REPORTED.

Func.	# Obj	R-MEAD2-PBI	R-MEAD2-Te	R-NSGA-II	K-W	R-MEAD2-PBI		
						vs R-NSGA-II	vs R-NSGA-II	vs R-NSGA-II
DTLZ1	4	2.7179e-02	2.0047e-02	2.6913e-02	2.37e-12	2.77e-12	4.43e-01	9.73e-11
	5	1.5969e-02	1.3061e-02	1.3489e-02	1.89e-12	2.77e-12	9.73e-11	1.99e-01
	6	1.9221e-03	1.4998e-03	1.5202e-03	8.51e-13	2.77e-12	9.73e-11	2.02e-02
	7	5.7668e-03	4.9279e-03	5.1213e-03	4.82e-13	2.77e-12	9.73e-11	4.50e-03
	8	1.0997e-02	8.7635e-03	1.0280e-02	1.10e-13	2.77e-12	1.05e-04	9.73e-11
	9	8.2376e-03	6.0180e-03	8.7701e-03	4.43e-14	2.77e-12	1.10e-05	9.73e-11
	10	1.3500e-02	1.0895e-02	1.5834e-02	3.74e-16	2.77e-12	9.73e-11	9.73e-11
DTLZ2	4	8.0403e-03	8.6697e-03	2.2126e-02	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	5	4.4710e-03	4.9080e-03	9.1054e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	6	4.8012e-04	5.0287e-04	7.2498e-04	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	7	1.1177e-03	1.4606e-03	1.6713e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	8	1.7828e-03	2.1404e-03	2.7116e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	9	9.8180e-04	1.0688e-03	1.4740e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	10	1.8515e-03	2.0637e-03	2.5305e-03	3.74e-16	2.77e-12	9.72e-11	9.73e-11
DTLZ3	4	7.1956e-03	9.1197e-03	2.2122e-02	1.04e-15	4.27e-11	7.54e-10	9.73e-11
	5	4.2700e-03	5.8662e-03	9.1130e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	6	1.1976e-03	4.6246e-04	7.2545e-04	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	7	1.2094e-03	1.3654e-03	1.6704e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	8	1.7592e-03	2.5105e-03	2.7037e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	9	9.9098e-04	1.1612e-03	1.4536e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	10	1.8656e-03	2.4228e-03	2.5326e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
DTLZ4	4	8.2896e-03	1.0301e-02	2.2212e-02	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	5	4.2842e-03	8.5091e-03	9.1201e-03	1.46e-15	2.77e-12	9.73e-11	2.64e-09
	6	4.1579e-04	7.0932e-04	8.3514e-04	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	7	1.0711e-03	1.6623e-03	2.0305e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	8	1.6788e-03	2.6767e-03	2.9353e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	9	1.1728e-03	1.3986e-03	1.5749e-03	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	10	1.7495e-03	2.4007e-03	2.6921e-03	4.40e-16	4.43e-12	1.38e-10	9.73e-11
DTLZ5	4	1.6671e-02	1.6204e-02	1.7121e-02	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	5	5.8838e-03	5.7738e-03	5.9900e-03	9.53e-13	2.77e-12	5.51e-08	5.51e-08
	6	1.8945e-03	1.7910e-03	1.9025e-03	1.46e-15	2.77e-12	2.64e-09	9.73e-11
	7	6.4055e-04	6.2722e-04	6.4189e-04	5.11e-15	2.77e-12	5.51e-08	9.73e-11
	8	9.6134e-04	9.1614e-04	9.6588e-04	3.74e-16	2.77e-12	9.73e-11	9.73e-11
	9	8.1450e-04	7.9596e-04	8.1874e-04	1.46e-15	2.77e-12	2.64e-09	9.73e-11
	10	7.2022e-04	6.7489e-04	7.2718e-04	1.46e-15	2.77e-12	2.64e-09	9.73e-11
DTLZ6	4	1.6829e-02	1.6831e-02	1.7129e-02	2.89e-11	2.77e-12	8.85e-07	8.85e-07
	5	5.8428e-03	5.7819e-03	5.9870e-03	3.42e-09	2.77e-12	4.50e-03	1.10e-05
	6	1.8867e-03	1.9213e-03	1.9077e-03	1.59e-14	2.77e-12	8.85e-07	9.73e-11
	7	6.3685e-04	6.6313e-04	6.4248e-04	4.43e-14	2.77e-12	1.10e-05	9.73e-11
	8	9.5918e-04	9.2020e-04	9.6915e-04	1.46e-15	2.77e-12	2.64e-09	9.73e-11
	9	8.1404e-04	8.2422e-04	8.2033e-04	5.11e-15	2.77e-12	5.51e-08	9.73e-11
	10	6.6082e-04	6.2611e-04	6.7947e-04	1.33e-12	4.43e-12	5.07e-02	9.73e-11

diversity of solutions into account. Therefore, we can conclude that decomposition methods generally perform better than R-NSGA-II in terms of both convergence and diversity on most many-objective problems. Finally, by comparing the results of R-MEAD2-PBI and R-MEAD2-Te we can see that both decomposition methods have very similar performance, but the PBI version performs slightly better than Tchebycheff on 24 functions and losing on 18 functions.

TABLE IV. R-NSGA-II'S NUMBER OF WINS, LOSSES AND TIES AGAINST R-MEAD2-Te AND R-MEAD2-PBI.

	R-MEAD2-Te			R-MEAD2-PBI		
	Wins	Losses	Ties	Wins	Losses	Ties
R-NSGA-II	3	37	2	5	35	2

V. CONCLUSIONS

In this paper, a user-preference based evolutionary algorithm for many-objective optimization problems has been proposed. The proposed algorithm R-MEAD2 has the following advantages: 1) Because of using a decomposition framework, R-MEAD2 is less susceptible to the selection pressure issue as

compared to dominance-based approaches such as R-NSGA-II. The experimental results using IGD metric showed that both versions of R-MEAD2 (Tchebycheff and PBI) outperform R-NSGA-II on a range of many-objective problems. 2) Unlike R-MEAD the population size and the number of objectives are decoupled and the increase of number of objectives does not cause the grow of population size. As a result, R-MEAD2 is better suited for solving many-objective problems. 3) R-MEAD2 uses a simple random number generator (RNG) to initialize the weight vectors. The results of centered L_2 -discrepancy showed that RNG can generate more uniform weights than GLP when the number of objectives grow beyond 7 with a typical sample size of 100, 250 and 500. In case of, other sample sizes (50, 1000 and 5000) RNG can produce points with lower discrepancy than GLP when the number of objectives goes beyond eight. It should be noted that a uniform set of weight vectors do not necessarily map to a uniform set of solution in the objective space, especially on highly non-linear and complex Pareto-optimal fronts. This requires a feedback mechanism to adjust the weights in order to obtain a set of uniform solutions which is the topic of our future works. In this paper, the performance of PBI and Tchebycheff appeared

to be very similar. However, a conclusive conclusion requires further investigations on functions with more complex Pareto-optimal fronts.

REFERENCES

- [1] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [2] R. C. Purshouse and P. J. Fleming, "Evolutionary many-objective optimisation: An exploratory analysis," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 3. IEEE, 2003, pp. 2066–2073.
- [3] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc of 2008 IEEE Congress on Evolutionary Computation*, 2008, pp. 2419–2426.
- [4] —, "Evolutionary many-objective optimization," in *Proc. of 3rd International Workshop on Genetic and Evolving Fuzzy Systems*, no. March, 2008, pp. 47–52.
- [5] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "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, 2009, pp. 1758–1763.
- [6] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithm*. Wiley, 2001.
- [7] K. Deb, J. Sundar, N. Udaya Bhaskara Rao, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273–286, 2006.
- [8] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evolutionary Computation*, vol. 17, no. 3, pp. 411–436, 2009.
- [9] K. Deb and A. Kumar, "Interactive evolutionary multi-objective optimization and decision-making using reference direction method," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, 2007, pp. 781–788.
- [10] —, "Light beam search based multi-objective optimization using evolutionary algorithms," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 2125–2132.
- [11] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "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, 2009, pp. 1758–1763.
- [12] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [13] A. Messac, A. Ismail-Yahaya, and C. A. Mattson, "The normalized normal constraint method for generating the pareto frontier," *Structural and multidisciplinary optimization*, vol. 25, no. 2, pp. 86–98, 2003.
- [14] K. Miettinen, *Nonlinear Multiobjective Optimization*. Norwell, MA: Kluwer, 1999.
- [15] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, pp. 392–403, 1998.
- [16] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.
- [17] A. Mohammadi, M. N. Omidvar, and X. Li, "Reference point based multi-objective optimization through decomposition," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.
- [18] Y. yan Tan, Y. chang Jiao, H. Li, and X. kuan Wang, "MOEA/D + uniform design: A new version of moea/d for optimization problems with many objectives," *Computers and Operations Research*, vol. 40, no. 6, pp. 1648 – 1660, 2013.
- [19] K. Fang and Y. Wang, *Number-theoretic methods in statistics*. Chapman and Hall/CRC, 1993.
- [20] K.-T. Fang and D. K. Lin, "Uniform experimental designs and their applications in industry," *Handbook of Statistics*, vol. 22, pp. 131–170, 2003.
- [21] J. Branke, T. Kaubler, and H. Schmeck, "Guidance in evolutionary multi-objective optimization," *Advances in Engineering Software*, vol. 32, no. 6, pp. 499–507, 2001.
- [22] K. Deb, "Multi-objective evolutionary algorithms: Introducing bias among pareto-optimal solutions," in *Advances in evolutionary computing*. Springer, 2003, pp. 263–292.
- [23] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [24] J. Branke and K. Deb, "Integrating user preferences into evolutionary multi-objective optimization," in *Knowledge incorporation in evolutionary computation*. Springer, 2005, pp. 461–477.
- [25] K. Deb, a. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, pp. 182–197, 2002.
- [26] U. K. Wickramasinghe and X. Li, "Using a distance metric to guide pso algorithms for many-objective optimization," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. ACM, 2009, pp. 667–674.
- [27] —, "Integrating user preferences with particle swarms for multi-objective optimization," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 2008, pp. 745–752.
- [28] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117–132, 2003.
- [29] D. J. Sheskin, *Handbook of parametric and nonparametric statistical procedures*. crc Press, 2003.