Use of Reference Point Sets in a Decomposition-based Multi-Objective Evolutionary Algorithm

Edgar Manoatl Lopez and Carlos A. Coello Coello*

CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Computación
México D.F. 07300, MÉXICO
emanoatl@computacion.cs.cinvestav.mx, ccoello@cs.cinvestav.mx

Abstract. In recent years, decomposition-based multi-objective evolutionary algorithms (MOEAs) have gained increasing popularity. However, these MOEAs depend on the consistency between the Pareto front shape and the distribution of the reference weight vectors. In this paper, we propose a decomposition-based MOEA, which uses the modified Euclidean distance ($d+$) as a scalar aggregation function. The proposed approach adopts a novel method for approximating the reference set, based on an hypercube-based method, in order to adapt the reference set for leading the evolutionary process. Our preliminary results indicate that our proposed approach is able to obtain solutions of a similar quality to those obtained by state-of-the-art MOEAs such as MOMBI-II, NSGA-III, RVEA and MOEA/DD in several MOPs, and is able to outperform them in problems with complicated Pareto fronts.

1 Introduction

Many real-world problems have several (often conflicting) objectives which need to be optimized at the same time. They are known as Multi-objective Optimization Problems (MOPs) and their solution gives rise to a set of solutions that represent the best possible trade-offs among the objectives. These solutions constitute the so-called *Pareto optimal set* and their image is called the *Pareto Optimal Front* (POF). Over the years, Multi-Objective Evolutionary Algorithms (MOEAs) have become an increasingly common approach for solving MOPs, mainly because of their conceptual simplicity, ease of use and efficiency.

Decomposition-based MOEAs transform a MOP into a group of sub-problems, in such a way that each sub-subproblem is defined by a reference weight point. Then, all these sub-problems are simultaneously solved using a single-objective optimizer [16]. Because of their effectiveness (e.g., with respect to Pareto-based

* The first author acknowledges support from CONACyT and CINVESTAV-IPN to pursue graduate studies in Computer Science. The second author gratefully acknowledges support from CONACyT project no. 221551.

MOEAs[1]) and efficiency,[2] decomposition-based MOEAs have become quite popular in recent years both in traditional MOPs and in many-objective problems (i.e., MOPs having four or more objectives).

However, the main disadvantage of decomposition-based MOEAs is that the diversity of its selection mechanism is led explicitly by the reference weight vectors (normally the weight vectors are distributed in a unit simplex). This makes them unable to properly solve MOPs with complicated Pareto fronts (i.e., Pareto fronts with irregular shapes).

Decomposition-based MOEAs are appropriate for solving MOPS with regular Pareto front (i.e., those sharing the same shape of a unit simplex). There is experimental evidence that indicates that decomposition-based MOEAs are not able to generate good approximations to MOPs having disconnected, degenerated, badly-scaled or other irregular Pareto front shapes [5, 2].

Here, we propose a decomposition-based MOEA, which adopts the modified Euclidean distance ($d+$) as a scalar aggregation function. This approach is able to switch between a PBI scalar aggregation function and the $d+$ distance in order to lead the optimization process. In order to adopt the $d+$ distance, we also incorporate an adaptive method for building the reference set. This method is based on the creation of hypercubes, which uses an archive for preserving good candidate solutions. We show that the resulting decomposition-based MOEA has a competitive performance with respect to state-of-the-art MOEAs, and that is able to properly deal with MOPs having complicated Pareto fronts.

The remainder of this paper is organized as follows. Section 2 provides some basic concepts related to multi-objective optimization. Our decomposition-based MOEA is described in Section 3. In Section 4, we present our methodology and a short discussion of our preliminary results. Finally, our conclusions and some possible paths for future research are provided in Section 5.

2 Basic Concepts

Formally a MOP in terms of minimization is defined as:

$$\text{minimize} \quad \boldsymbol{f}(\boldsymbol{x}) := [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})]^T \tag{1}$$

subject to:

$$g_i(\boldsymbol{x}) \leq 0, \quad i = 1, 2, \ldots, p \tag{2}$$

$$h_j(\boldsymbol{x}) = 0, \quad j = 1, 2, \ldots, q \tag{3}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ is the vector of decision variables, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \ldots, m$ are the objective functions and $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 0, \ldots, p$, $j = 1, \ldots, q$ are the constraint functions of the problem.

[1] It is well-known that Pareto-based MOEAs cannot properly solve many-objective problems [12].

[2] The running time of decomposition-based MOEAs is lower than that of indicator-based MOEAs [1, 9] and reference-based MOEAs [14].

We also need to provide more details about the IGD+ indicator, which uses the modified Euclidean distance that we adopt in our proposal. According to [11], the IGD+ indicator can be described as follows:

$$\text{IGD}^+(\mathcal{A}, \mathcal{Z}) = \frac{1}{|\mathcal{Z}|} \left(\sum_{j=1}^{|\mathcal{Z}|} d_j^+(\boldsymbol{z}, \boldsymbol{a})^p \right)^{1/p} \tag{4}$$

where $\boldsymbol{a} \in \mathcal{A} \subset \mathbb{R}^m$, $\boldsymbol{z} \in \mathcal{Z} \subset \mathbb{R}^m$, $\mathcal{A}$ is the Pareto front set approximation and $\mathcal{Z}$ is the reference set. $d^+(\boldsymbol{a}, \boldsymbol{z})$ is defined as:

$$d^+(\boldsymbol{z}, \boldsymbol{a}) = \sqrt{(\max\{a_1 - z_1, 0\})^2, \ldots, (\max\{a_m - z_m, 0\})^2}. \tag{5}$$

Therefore, we can see that the set $\mathcal{A}$ represents a better approximation to the real $\mathcal{PF}$ when we obtain a lower IGD^+ value, if we consider the reference set as $\mathcal{PF}_{True}$. IGD^+ was shown to be weakly Pareto complaint, and this indicator presents some advantages with respect to the original Inverted Generational Distance (for more details about IGD and IGD^+, see [4] and [11] respectively).

3 Our Proposed Approach

3.1 General Framework

Our approach adopts the same structure of the original MOEA/D [16], but we include some improvements in order to solve MOPs with complicated Pareto fronts. Our approach has the following features: (1) An archiving process for preserving candidate solutions which will form the reference set; (2) a method for adapting the reference set in order to sample uniformly the Pareto front; and (3) a rule for updating the reference set. Algorithm 1 shows the details of our proposed approach. Our proposed MOEA decomposes the MOP into scalar optimization subproblems, where each subproblem is solved simultaneously by an evolutionary algorithm (same as the original MOEA/D). The population, at each generation, is composed by the best solution found so far for each subproblem. Each subproblem is solved by using information only from its neighborhood, where each neighborhood is defined by the n candidate solutions which have the nearest distance based on the scalar aggregation function. The reference update process is launched when certain percentage of the evolutionary process (defined by "UpdatePercent") is reached. The reference update process starts to store the non-dominated solutions in order to sample the shape of the Pareto front. When the cardinality of the set $|\mathcal{A}|$ is equal to "ArchiveSize", the reference method is launched for selecting the best candidate solutions, which will form the *new reference set*. Once this is done, the scalar aggregation function is updated by choosing the modified Euclidean distance ($d+$) (see equation (4)), and the set $\mathcal{A}$ is cleaned up. The number of allowable updates is controlled by the variable "maxUpdates".

Algorithm 2: General Framework

Input: A MOP, a stopping criterion, N subproblems, a uniform spread of N reference vectors: $\lambda^1 \dots \lambda^N$, number of solutions in the neighborhood and a scalar aggregation function (g).

Output: Approximation of the MOP

Create each neighborhood for every reference vector: $B(i)$;
Generate an initial population randomly $(x_i, \dots, x_N) \in X$;
$t \leftarrow 0$;
$\mathcal{A} \leftarrow \{\}$;
while $t < gen_{max}$ **do**
  for each $B(i) \in \mathcal{B}$ **do**
    Apply evolutionary operators: Randomly select two parents from $B(i)$ and create an individual y;
    Improvement: Apply a problem-specific repair/improvement heuristic on y to produce y';
    for each $j \in B(i)$ **do**
      if $g(F(y'), \lambda^j) < g(F(x_j), \lambda^j)$ **then**
        $x_j \leftarrow y'$;
      end if
    end for
    Update of Neighboring Solutions: For each index in $B(i)$;
    if $t > UpdatePercent$ **then**
      if $|A| < ArchiveSize$ **then**
        $\mathcal{A} \leftarrow nonDominated(\mathcal{A} \cup y')$;
        $y_{ref} \leftarrow getNadirPoint(\mathcal{A})$;
      end if
      if $|A| == ArchiveSize$ and $Updates < maxUpdates$ **then**
        $\lambda^1 \dots \lambda^N \leftarrow$ ComputeReferenceSet$(\mathcal{A}, y_{ref}, z_{size})$;
        $g(.) \leftarrow d+$;
        $\mathcal{A} \leftarrow \{\}$;
        $Updates \leftarrow Updates + 1$;
      end if
    end if
  end for
  $t \leftarrow t + 1$;
end while
$\mathcal{Q} \leftarrow$ non-Dominated $(F(X))$;
return $\mathcal{Q}, X$;

3.2 Archiving Process

As mentioned before, the archive stores non-dominated solutions, up to a maximum number of solutions defined by the "ArchiveSize" value. When the archive reaches its maximum capacity, the approximation reference algorithm is executed for selecting candidate solutions (these candidate solutions will form the so-called *candidate reference set*). After that, the archive is cleaned and the archiving process continues until reaching a maximum number of updates. The archiving process is applied after a 60% of the total number of generations. It is worth mentioning that the *candidate reference set* is not compatible with the weight relation rule[3], which implies that it is not possible to use the Tchebycheff scalar aggregation function for leading the search. However, the PBI function works because it only requires directions (for more details see [16]).

3.3 Reference Set

In our approach, we aim to select the best candidate points whose directions are promising (these candidate solutions will sample the Pareto front as uniformly as possible). The main idea is to apply a density estimator. For this reason, we propose to use an algorithm based on the hypercube contributions to select a certain number of reference points from the archive. Algorithm 3 provides the pseudo-code of an approach that is invoked with a set of non-dominated candidate points (called $\mathcal{A}$ set) and the maximum number of reference points that we aim to find. The algorithm is organized in two main parts. In the first loop, we create a set of initial candidate solutions to form the so-called $\mathcal{Q}$ set. Thus, the solutions from $\mathcal{A}$ that form part of $\mathcal{Q}$ will be removed from $\mathcal{A}$. After that, the greedy algorithm starts to find the best candidate solutions which will form the reference set $\mathcal{Z}$. In order to find the candidate reference points, the selection mechanism computes the hypercube contributions of the current reference set $\mathcal{Q}$. Once this is done, we remove the i^{th} solution that minimizes the hypercube value and we add a new candidate solution from $\mathcal{A}$ to $\mathcal{Q}$. This process is executed until the cardinality of $\mathcal{A}$ is equal to zero. In the line 21 of Algorithm 3, we apply the *expand* and *translate* operations. A hypercube is generated by the union of all the maximum volumes covered by a reference point. The i^{th} maximum volume is described as "the maximum volume generated by a set of candidate points" (these candidate points are obtained from the archive using a reference point y_{ref}). The hypercube is computed using Algorithm 4. The main idea of this algorithm is to add all the maximum volumes, which are defined by the maximum point and the reference point (y_{ref}). When a certain point is considered to be the maximum point, the objective space is split between m parts. The maximum point is removed from the set $\mathcal{Q}$. This process is repeated until $\mathcal{Q}$ is empty.

In the first part of Algorithm 4, we validate if $\mathcal{Q}$ contains one element. If that is the case, we compute the volume generated by y_{ref} and $\boldsymbol{q} \in \mathcal{Q}$. Otherwise,

[3] The weights of the reference point problem should be $\sum_{i=0}^{m} \lambda_i = 1$.

we compute the union of all the maximum hypercubes. In order to apply this procedure, we find the vector $\boldsymbol{q_{max}}$ that maximizes the hypercube. Once this is done, we create m reference points which will form the so-called $\mathcal{Y}$ set. For each reference point from $\mathcal{Y}$, we reduce the set $\mathcal{Q}$ into a small subset in order to form the set $\mathcal{Q}_{new}$. Once this is done, we proceed to compute recursively the hypercube value of the new set formed by the subset $\mathcal{Q}_{new}$ and the new reference point y_{new}. It is worth noting that this value allows to measure the relationship among each element of a non-dominated set.

Algorithm 3: ComputeReferenceSet($\mathcal{A}, z_{size}$)

Input: A current non-dominated set $\mathcal{A} \subset \mathbb{R}^m$ and maximum number of reference points z_{size}.
Output: Reference point set $\mathcal{Z} \subset \mathbb{R}^m$ with $|\mathcal{Z}| = z_{size}$

$y_{ref} \leftarrow FindMaxValue(\mathcal{A}) + \epsilon$;
$\mathcal{Q} \leftarrow \{\}$;
while $|\mathcal{Q}| < (z_{size} + 1)$ **do**
  $\boldsymbol{a} \leftarrow pop(\mathcal{A})$;
  $\mathcal{Q} \bigcup \{\boldsymbol{a}\}$;
end while
while $\mathcal{A}! = \{\}$ **do**
  $i \leftarrow 0$;
  $maxHypercube \leftarrow HCB(\mathcal{Q}, y_{ref})$;
  for each $\boldsymbol{q} \in \mathcal{Q}$ **do**
    $ContHyperCube[i] \leftarrow maxHypercube - HCB(\mathcal{Q} \backslash \{\boldsymbol{q}\}, y_{ref})$;
    $i \leftarrow i + 1$;
  end for
  $i_{min} \leftarrow \text{argmin}\, ContHyperCube$;
  $\mathcal{Q} \backslash \{q_{i_{min}}\}$;
  $\boldsymbol{a} \leftarrow pop(\mathcal{A})$;
  $\mathcal{Q} \bigcup \{\boldsymbol{a}\}$;
end while
$\mathcal{Z} \leftarrow \{\}$;
for each $\boldsymbol{q} \in \mathcal{Q}$ **do**
  $\mathcal{Z} \bigcup \{\boldsymbol{q} * \epsilon - \boldsymbol{l}\}$;
end for
return $\mathcal{Z}$;

4 Experimental Results

We compare the performance of our approach with respect to that of four state-of-the-art MOEAs: MOEA/DD [13], NSGA-III [5], RVEA [2], and MOMBI-II [9]. These MOEAs had been found to be competitive in MOPs with a variety of Pareto front shapes. MOEA/DD [13] is an extension of MOEA/D which includes the Pareto dominance relation to select candidate solutions and is able

Algorithm 4: HCB($\mathcal{Q}, y_{ref}$)

Input: A current set $\mathcal{Q} \subset \mathbb{R}^m$ and a reference point y_{ref}
Output: Hypercube value

if $|\mathcal{Q}| = 1$ **then**
  return vol($\mathcal{Q}, y_{ref}$);
end if
$VolList \leftarrow \{\}$;
for each $\boldsymbol{p} \in \mathcal{Q}'$ **do**
  $VolList \bigcup \{vol(\boldsymbol{p}, y_{ref})\}$;
end for
$i_{max} \leftarrow \text{argmax}\, VolList$;
$\boldsymbol{q}_{max} \leftarrow \mathcal{Q}[i_{max}]$;
$\mathcal{Y} \leftarrow$ SplitReferencePoint ($\boldsymbol{q}_{max}, y_{ref}$);
$\mathcal{Q} \backslash \{\boldsymbol{q}_{max}\}$;
$hypercube \leftarrow 0$;
for each $\boldsymbol{y}_{new} \in \mathcal{Y}$ **do**
  $\mathcal{Q}_{new} \leftarrow$ CoverPoints ($\mathcal{Q}, \boldsymbol{y}_{new}$);
  $hypercube \leftarrow hypercube + HCB(\mathcal{Q}_{new}, \boldsymbol{y}_{new})$;
end for
return $hypercube + max(VolList)$;

to outperform the original MOEA/D, particularly in many-objective problems having up to 15 objectives. NSGA-III [5] uses a distributed set of reference points to manage the diversity of the candidate solutions, with the aim of improving convergence. The *Reference Vector Guided Evolutionary Algorithm* (RVEA) [2] provides very competitive results in MOPs with complicated Pareto fronts. *Many Objective Meta-heuristic Based on the R2 indicator* (MOMBI) [8] adopts the use of weight vectors and the $R2$ indicator, and both mechanisms lead the optimization process. MOMBI is very competitive but it tends to lose diversity in high dimensionality. This study includes an improved version of this approach, called MOMBI-II [9].

4.1 Methodology

For our comparative study, we decided to adopt the Hypervolume indicator, due to this indicator is able to assess both convergence and maximum spread along the Pareto front. The reference points used in our preliminary study are shown in Table 1.

We aimed to study the performance of our proposed approach when solving MOPs with complicated Pareto front shapes. For this reason, we selected 18 test problems with a variety of representative Pareto front shapes from some well-known and recently proposed test suites: the DTLZ [7], the WFG [10], the MAF [3] and the VNT test suites [15].

Problem	Reference point	Problem	Reference point
DTLZ1	(1,1,1)	VNT1	(5, 6, 5)
DTLZ2-6	(2,2,2)	VNT2	(5, -15, -11)
DTLZ7	(2, 2, 7)	VNT3	(9, 18, 5)
MAF1-3	(2,2,2)	WFG1	(3, 5, 7)
MAF4	(3,5, 9)	WFG2	(2, 4, 7)
MAF5	(9, 5, 3)	WFG3	(2, 3, 7)

Table 1: Reference points used for the hypervolume indicator

4.2 Parameterization

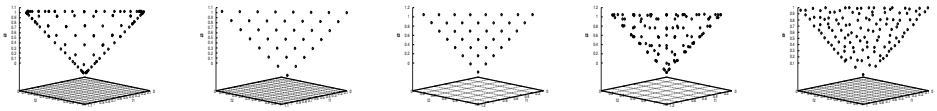
In the MAF and DTLZ test suites, the total number of decision variables is given by $n = m + k - 1$, where m is the number of objectives and k was set to 5 for DTLZ1 and MAF1, and to 10 for DTLZ2-6, and MAF2-5. The number of decision variables in the WFG test suite was set to 24, and the position-related parameter was set to $m-1$. The distribution indexes for the Simulated Binary crossver and the polynomial-based mutation operators [6] adopted by all algorithms, were set to: $\eta_c = 20$ and $\eta_m = 20$, respectively. The crossover probability was set to $p_c = 0.9$ and the mutation probability was set to $p_m = 1/L$, where L is the number of decision variables. The total number of function evaluations was set in such a way that it did not exceed 60,000. In MOEA/DD, MOMBI-II and NSGA-III, the number of weight vectors was set to the same value as the population size. The population size N is dependent on H. For this reason, for all test problems, the population size was set to 120 for each MOEA. In RVEA, the rate of change of the penalty function and the frequency to conduct the reference vector adaptation were set to 2 and 0.1, respectively. Our approach was tested using a PBI scalar aggregation function and the modified Euclidean distance ($d+$). The maximum number of elements allowed in the archive was set to 500 and the maximum number of reference updates was set to 5.

4.3 Discussion of Results

Table 2 shows the average hypervolume values of 30 independent executions of each MOEA for each instance of the DTLZ, VNT, MAF and WFG test suites, where the best results are shown in **boldface** and grey-colored cells contain the second best results. The values in parentheses show the variance for each problem. We adopted the Wilcoxon rank sum test in order to compare the results obtained by our proposed MOEA and its competitors at a significance level of 0.05, where the symbol "+" indicates that the compared algorithm is significantly outperformed by our approach. On the other hand, the symbol "-" means that MOEA/DR is significantly outperformed by its competitor. Finally, "$\approx$" indicates that there is no statistically significant difference between the results obtained by our approach and its competitor.

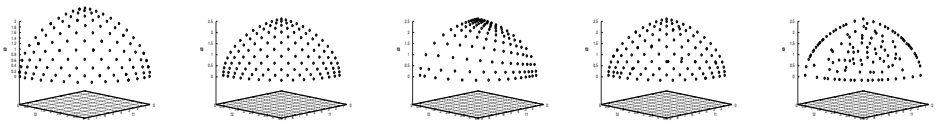
	MOMBI-II	RVEA	MOEA/DD	NSGA-III	MOEA/DR
DTLZ1	0.96622 (0.000001) +	0.66911 (0.000152) +	**0.97379 (0.000000)** ≈	0.96256 (0.001064) +	0.97265 (0.000007)
DTLZ2	7.36755 (0.000028) +	7.42224 (0.000000) ≈	7.42234 (0.000000) ≈	7.41893 (0.000000) +	**7.42684 (0.000143)**
DTLZ3	7.38843 (0.000084) -	7.40582 (0.000084) -	**7.4118 (0.000047)** -	7.38048 (0.000258) -	7.26131 (0.000248)
DTLZ4	7.3593 (0.036144) -	**7.42226 (0.000000)** -	7.42224 (0.000000) -	7.10506 (0.227356) ≈	7.10433 (1.093691)
DTLZ5	6.00978 (0.000000) +	5.9632 (0.000369) +	6.02456 (0.000062) +	5.84002 (0.05518) +	**6.10349 (0.000002)**
DTLZ6	5.79608 (0.00523) +	5.13815 (0.016264) +	5.6037 (0.006442) +	5.49135 (0.023354) +	**5.84857 (0.003765)**
DTLZ7	**13.37473 (0.000091)** -	13.0605 (1.283746) -	12.99409 (0.015542) ≈	13.32733 (0.002554) -	12.37989 (0.181549)
VNT1	61.44939 (0.000533) +	60.51323 (0.011862) +	60.55111 (0.021176) +	61.19214 (0.011932) +	**61.88114 (0.512056)**
VNT2	7.79702 (0.000001) +	7.7712 (0.000368) +	7.80468 (0.000037)	7.77446 (0.000935) +	**7.84291 (0.000554)**
VNT3	15.11767 (0.000262) +	15.03082 (0.000422) +	15.06016 (0.000114) +	15.12629 (0.000502) +	**15.15149 (6.685422)**
MAF1	**5.44926 (0.000019)** -	5.37408 (0.000659) +	5.37139 (0.00009) +	5.4129 (0.000875) -	5.3986 (0.013358)
MAF2	5.08952 (0.000056)	**5.1583 (0.000058)** ≈	5.11373 (0.000003) +	5.09758 (0.000043) +	5.14115 (0.000105)
MAF3	7.90637 (0.000043) -	**7.91154 (0.004847)** -	7.64261 (1.915744) +	7.89441 (0.00452) -	7.82731 (0.000558)
MAF4	**84.87316 (0.151259)** -	83.53436 (29.511151) -	51.80943 (1120.296924) +	83.73257 (1.377427) -	75.81219 (4.084039)
MAF5	95.97704 (52.294491) +	96.66782 (53.122845) +	96.95207 (0.017991) +	88.72762 (237.475764) +	**98.26977 (44.804422)**
WFG1	50.38691 (7.353216) -	**51.68413 (5.001739)** -	41.77398 (7.334821) +	44.95726 (10.36034) ≈	43.02462 (6.595565)
WFG2	48.72516 (12.06217) -	**51.14414 (0.045119)** -	44.23925 (3.146579) +	48.14747 (12.622738) -	46.87356 (1.171321)
WFG3	**24.28138 (0.007298)** -	22.12339 (0.086504) -	21.04349 (0.178677) -	23.54542 (0.037132) -	16.85662 (0.76122)

Table 2: Performance comparison among several MOEAs using the average hypervolume values obtained from 30 independent executions solving 18 benchmark problems for 3 objectives.



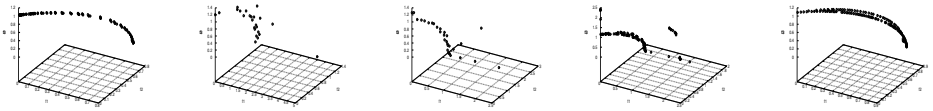
(a) MOMBI-II (b) RVEA (c) MOEA/DD (d) NSGA-III (e) MOEA/DR

Fig. 1: Graphical representation of the final set of solutions obtained by each MOEA on MAF1 with 3 objectives



(a) MOMBI-II (b) RVEA (c) MOEA/DD (d) NSGA-III (e) MOEA/DR

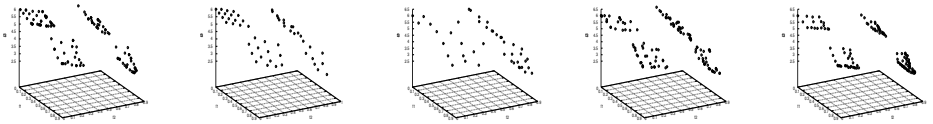
Fig. 2: Graphical representation of the final set of solutions obtained by each MOEA on MAF5 with 3 objectives



(a) MOMBI-II (b) RVEA (c) MOEA/DD (d) NSGA-III (e) MOEA/DR

Fig. 3: Graphical representation of the final set of solutions obtained by each MOEA on DTLZ6 with 3 objectives

As can be seen in Table 2, our MOEA was able to outperform MOMBI-II, RVEA, MOEA/DD, and NSGA-III in seven instances and in several other cases, it obtained very similar results to those of the best performer. We can see that our approach outperformed its competitors in MOPs with degenerate Pareto fronts (DTLZ5-6 and VNT2-3). In this study, MOMBI-II is ranked as the second best overall performer, because it was able to outperform its competitors in four cases. It is worth mentioning that all the adopted MOEAs are very competitive because the final set of solutions obtained by them has similar quality in terms of the hypervolume indicator.



(a) MOMBI-II (b) RVEA (c) MOEA/DD (d) NSGA-III (e) MOEA/DR

Fig. 4: Graphical representation of the final set of solutions obtained by each MOEA on DTLZ7 with 3 objectives

Figures 1, 2, 3, 4 show a graphical representation of the final set of solutions obtained by each MOEA. On the MOPs with inverted Simplex-like Pareto fronts, our algorithm had a good performance (see Figure 1). Figures 1.a to 1.e show that the solutions produced by all the MOEAs adopted have a good coverage of the corresponding Pareto fronts. However, the solutions of MOMBI-II and NSGA-III are not distributed very uniformly, while the solutions of RVEA and MOEA/DD are distributed uniformly but their number is apparently less than

their population size. On MOPs with badly-scaled Pareto fronts, our approach was able to obtain the best approximation (see Figure 2). Figures 2.a to 2.e show that the solutions produced by all the MOEAs adopted are distributed very uniformly. On MOPs with degenerate Pareto fronts, it is clear that the winner in this category is our algorithm since the solutions of NSGA-III, RVEA and MOEA/DD are not distributed very uniformly, and they were not able to converge (see Figure 3). On MOPs with disconnected Pareto fronts, our approach did not perform better than the other MOEAs. The reason is probably that the evolutionary operators were not able to generate solutions in the whole objective space, which makes the approximations produced by our approach to converge to a single region. Figure 4 shows that RVEA was able to obtain the best approximation in DTLZ7 since its approximation is distributed uniformly along the Pareto front.

5 Conclusions and Future Work

We have proposed a decomposition-based MOEA for solving MOPs with different Pareto front shapes (i.e. those having complicated Pareto front shapes). The core idea of our proposed approach is to adopt the modified Euclidean distance ($d+$) as a scalar aggregation function. Additionally, our proposal introduces a novel method for approximating the reference set, based on an hypercube-based method, in order to adapt the reference set to address the evolutionary process. Our results show that our method for adapting the reference point set improves the performance of the original MOEA/D. As can be observed, the reference set is of utmost importance since our approach leads its search process using a set of reference points. Our preliminary results indicate that our approach is very competitive with respect to MOMBI-II, RVEA, MOEA/DD and NSGA-III, being able to outperform them in seven benchmark problems. Based on such results, we claim that our proposed approach is a competitive alternative to deal with MOPs having complicated Pareto front shapes. As part of our future work, we are interested in studying the sensitivity of our proposed approach to its parameters. We also intend to improve its performance in those cases in which it was not the best performer.

References

1. N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 16 September 2007.
2. R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791, October 2016.
3. R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao. A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3(1):67–81, Mar 2017.

4. C. A. Coello Coello and M. Reyes Sierra. A Study of the Parallelization of a Coevolutionary Multi-Objective Evolutionary Algorithm. In R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, and H. Sossa, editors, *Proceedings of the Third Mexican International Conference on Artificial Intelligence (MICAI'2004)*, pages 688–697. Springer Verlag. Lecture Notes in Artificial Intelligence Vol. 2972, April 2004.
5. K. Deb and H. Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, August 2014.
6. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
7. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.
8. R. Hernández Gómez and C. A. Coello Coello. MOMBI: A New Metaheuristic for Many-Objective Optimization Based on the R2 Indicator. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 2488–2495, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.
9. R. Hernández Gómez and C. A. Coello Coello. Improved Metaheuristic Based on the R2 Indicator for Many-Objective Optimization. In *2015 Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 679–686, Madrid, Spain, July 11-15 2015. ACM Press. ISBN 978-1-4503-3472-3.
10. S. Huband, P. Hingston, L. Barone, and L. While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, October 2006.
11. H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. In A. Gaspar-Cunha, C. H. Antunes, and C. Coello Coello, editors, *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*, pages 110–125. Springer. Lecture Notes in Computer Science Vol. 9019, Guimarães, Portugal, March 29 - April 1 2015.
12. H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 2424–2431, Hong Kong, June 2008. IEEE Service Center.
13. K. Li, K. Deb, Q. Zhang, and S. Kwong. An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716, October 2015.
14. E. Manoatl Lopez and C. A. Coello Coello. IGD$^+$-EMOA: A Multi-Objective Evolutionary Algorithm based on IGD$^+$. In *2016 IEEE Congress on Evolutionary Computation (CEC'2016)*, pages 999–1006, Vancouver, Canada, 24-29 July 2016. IEEE Press. ISBN 978-1-5090-0623-9.
15. D. A. V. Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations.* PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA, May 1999.
16. Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.