

# Multi-objective evolutionary algorithms for optimal scheduling

Wang, Y.

# Chapter 4

# Preference-based Multi-objective Evolutionary Algorithms

With the MOEAs, the entire Pareto front of a multi-objective optimization problem can be approximated. However, finding a well-distributed set of solutions on the Pareto front requests a large population size and computational effort. At the same time, the final goal of EMO is to help the DM to find solutions which match his/her preferences most, and the DM may only pay attention to a smaller set of Pareto optimal solutions. Therefore, integrating preferences in solving MOPs has become the subject of intensive studies of EMO. In other words, instead of spreading a limited size of individuals across the entire Pareto front, the search for solutions will be only guided towards the preference region.

The existing preference-based optimization methods can be classified into three categories according to the time when preference information is incorporated, i.e., a priori, interactive, and a posteriori methods. In a priori methods, the DM articulates preference information before the optimization process. In a posteriori methods, a set of Pareto optimal solutions is obtained first, and then the DM selects the most preferred ones among them. In interactive methods, the DM participates in the optimization process and directs the search according to his/her preferences. With the increasing understanding of the problem as the optimization proceeds, DMs are able to fine tune their preferences according to the obtained solutions in the optimization

process.

The preference based multi-objective evolutionary algorithms drive the population towards the region(s) of interest (ROI). The definition of the ROI depends on the way how the DM articulates his/her preference information. The preference information can be represented as the reference point(s) (e.g., [30, 43]); preference region(s) (e.g., [68, 79]), reference direction(s) (e.g., [28]); light beams (e.g., [27]), ranking (e.g., [44]), trade-offs (e.g., [95]), etc. However, the essence of preference information is to imply the ROI which allows a more focused search, thus to save computational resources.

This chapter is related to answer RQ3. First, a method that incorporates target region(s) into the core of the optimization process is proposed in Section 4.1. Then, in Section 4.2 to reduce the burden of the DM, an automatic preference based algorithm is proposed and integrated in DI-MOEA. In this algorithm, the preference region is generated automatically and narrowed down step by step to benefit its accuracy.

## 4.1 Target Region Based MOEAs

A target region based multi-objective evolutionary algorithm framework is proposed to find a more fine-grained resolution of a target region without exploring the whole set of Pareto optimal solutions. The idea of the target region is to first present a rough approximation of the Pareto front, then let the DM decide which region of interest to zoom in. Therefore, the target region is assumed to be the preference region according to the DM. The algorithm framework has been combined with SMS-EMOA, R2-EMOA, NSGA-II to form three target region based multi-objective evolutionary algorithms: T-SMS-EMOA, T-R2-EMOA and T-NSGA-II (where T stands for target).

NSGA-II is a frequently-used Pareto dominance-based MOEA; SMS-EMOA [9] and R2-EMOA [104] are indicator-based approaches which use performance measures (indicators) on the quality of the PF approximations to guide the search. The hypervolume is used in SMS-EMOA and the R2 indicator [55] is used in R2-EMOA. These two indicators measure both convergence and diversity of a PF approximation.

The R2 indicator of a solution set $A$ is defined as

$$R2(A, \Lambda, \mathbf{i}) = \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} \min_{a \in A} \{ \max_{j \in \{1,\dots,m\}} \{\lambda_j |\mathbf{i}_j - a_j|\}\}. \tag{4.1}$$

Here $\mathbf{i}$ is the ideal point and $m$ is the objective number, $\lambda = (\lambda_1, ..., \lambda_d) \in \Lambda$ is a given set of weight vectors. Usually, the weight vectors are chosen uniformly distributed over the weight space, for example, for m = 2 objectives, $\Lambda_k = (0, 1; \frac{1}{k-1}, 1 -$

$\frac{1}{k-1}; \frac{2}{k-1}, 1-\frac{2}{k-1}; ...; 1, 0)$ denotes $k$ uniformly distributed weights in the space $[0, 1]^2$.

The three proposed target region based algorithms have been tested with rectangular and spherical target regions on some benchmark problems, including continuous problems and discrete problems. Moreover, the proposed algorithms have been enhanced to support multiple target regions and preference information based on a target point or multiple target points.

The remainder of this section is organized as follows. In Section 4.1.1, the proposed algorithms are described. The experimental results are reported in Section 4.1.2. The details and graphical results of enhanced algorithms are presented in Section 4.1.3. Section 4.1.4 concludes the work with the summary and outlook.

### 4.1.1 Basic Algorithms

In the proposed algorithms, i.e., T-SMS-EMOA, T-R2-EMOA and T-NSGA-II, three ranking criteria: 1. non-dominated sorting; 2. performance indicator (Hypervolume in T-SMS-EMOA, R2 in T-R2-EMOA) and crowding distance in T-NSGA-II; 3. the Chebyshev distance to the target region, work together to achieve a well-converged and well-distributed set of Pareto optimal solutions in the target region using preference information provided by the DM. The Chebyshev distance speeds up evolution toward the target region and is computed as the distance to the center of the target region.

The second level ranking criterion: hypervolume, R2 indicator or crowding distance, is used as a diversity mechanism and measured based on coordinate transformations using desirability functions (DFs). The concept of desirability is introduced by Harrington [56] in the context of multi-objective industrial quality control and the approach of expressing the preferences of the DM using DFs is suggested by Wagner and Trautmann [112]. DFs map the objective values to desirabilities which are normalized values in the interval $[0, 1]$, where the larger the value, the more satisfying the quality of the objective value. The Harrington DF [56] and Derringer-Suich DF [100] are two most common types of DFs. By mapping the objective values to desirabilities according to preference information, both of these two DFs can result in biased distributions of the solutions on the PF. In the proposed algorithms, a simple DF 4.2 is used and it classifies the domain of the objective function into only two classes, "unacceptable" and "acceptable".

$$D(x) = \begin{cases} 1 & x \text{ is in the target region,} \\ 0 & x \text{ is not in the target region.} \end{cases} \tag{4.2}$$

The desirability here is for a solution. Solutions out of the target region are considered as unacceptable solutions and their desirabilities are assigned to be 0; at the same time, all solutions inside the target region are assumed acceptable and of equal importance, their desirabilities are assigned to be 1. For solutions with desirability 0, their second level ranking criterion is assigned to be 0 and for solutions with desirability 1, their second level ranking criterion needs to be calculated further. Since only solutions in the target region are retained, an approach is derived to simplify the calculation of the indicator values and realize a reference point free version of indicators [38], which is coordinate transformation. To be specific, the target region is treated as a new coordinate space of which the origin being the lower bound. For the maximization problem in T-SMS-EMOA or the minimization problem in T-R2-EMOA, a coordinate transformation is performed for the $i$-th objective as:

$$Ct_i(x) = f_i(x) - LB(f_i). \tag{4.3}$$

For minimization problem in T-SMS-EMOA or the maximization problem in T-R2-EMOA, coordinate transformation is performed for the $i$-th objective as:

$$Ct_i(x) = UB(f_i) - (f_i(x) - LB(f_i)) \tag{4.4}$$

where $LB(f_i)$ and $UB(f_i)$ are the lower bound and upper bound of the $i$-th objective in the target region predefined by the DM.

The reason for distinguishing the maximization and minimization problem in coordinate transformation is that the origin of the new coordinate space (i.e., the lower bound of the target region) is adopted as the reference point when calculating the indicator values. In T-SMS-EMOA, the worst point in the target region is chosen as the reference point when calculating hypervolume. On the contrary, the ideal point is chosen as the reference point when calculating R2 indicator in T-R2-EMOA. After completing coordinate transformation, the calculation of the second ranking criterion is implemented only in the target region instead of the whole coordinate system. It does make sense because the target region is the desired space to the DM. No reference point is needed in the calculation of crowding distance, therefore, any of the two formulas of coordinate transformation can be used in T-NSGA-II. Figure 4.1 shows an example of obtaining solutions in the target region by the proposed approach.

The target region is used to express the preference information from the DM, the shape of the target region does not necessarily need to be rectangular, it could as

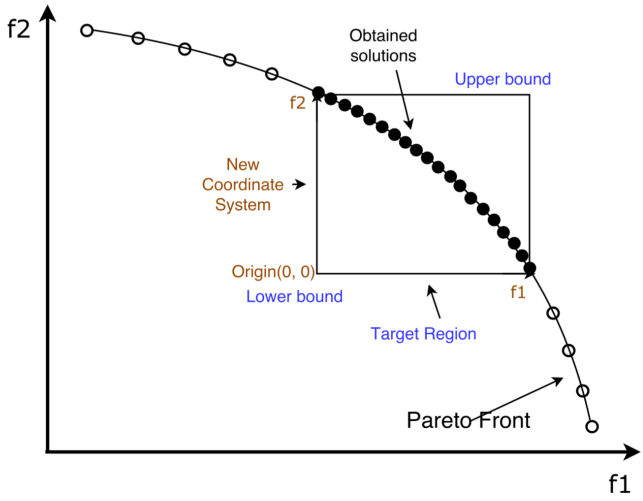



**Figure 4.1:** An example of obtaining solutions in the target region by proposed approach.

well be a circle, an ellipse or in other shapes as long as it can be confirmed efficiently whether or not a solution is in the target region. For instance, if the DM wants the solutions to be restricted into a sphere, s/he can specify the center point and radius of the sphere, the solution set can be achieved in the sphere.

### T-SMS-EMOA

The details of T-SMS-EMOA are given in Algorithm 3.

The framework of T-SMS-EMOA is based on the framework of SMS-EMOA. However, after the step of non-dominated sorting, all solutions in the worst ranked front are partitioned into two parts (i.e., acceptable and unacceptable) by the DF. Solutions in the first part have desirability 0 and their hypervolume contributions are assigned to be 0. Solutions in the second part have desirability 1 and coordinate transformation is conducted on each objective of each solution in this part; afterwards, their hypervolume contributions are calculated in the new coordinate system and the origin in the new coordinate system is adopted as the reference point. The other difference between T-SMS-EMOA and SMS-EMOA is the involvement of the Chebyshev distance. In the early iterations, the existence of individuals in the target region is low, the Chebyshev distance works on attracting solutions towards the target region.

**Algorithm 3** T-SMS-EMOA

```
1: P_0 ← init(); //Initialize random population
2: t ← 0;
3: while Stop criterion not satisfied() do
4:     q_{t+1} ← Gen(P_t); //Generate offspring by variation
5:     P_t ← P_t ∪ {q_{t+1}};
6:     {R_1, ··· , R_v} ← Nondominated-sort(P_t);
7:     ∀x ∈ R_v : compute D_Ch(x); //Chebyshev distance to center of target region
8:     R_v1 ∪ R_v2 ← R_v; //Solutions not in the target region → R_v1; others → R_v2
9:     ∀x ∈ R_v1 : HC(x) ← 0; //Hypervolume contribution
10:    R_v2 ← Coordinate Transformation(R_v2);
11:    ∀x ∈ R_v2 : HC(x) ← HV(R_v2) − HV(R_v2\x);
12:    if unique argmin{HC(x) : x ∈ R_v} exists then
13:        x* ← argmin{HC(x) : x ∈ R_v};
14:    else
15:        x* ← argmax{D_Ch(x) : x ∈ R_v}; //In case of tie, choose randomly
16:    end if
17:    P_{t+1} ← P\{x*};
18:    t ← t + 1;
19: end while
```

### T-R2-EMOA

The details of T-R2-EMOA are given in Algorithm 4.

R2-EMOA is extended to T-R2-EMOA in the same way SMS-EMOA is extended to T-SMS-EMOA. The formula of coordinate transformation used in T-R2-EMOA, however, is opposite to the formula used in T-SMS-EMOA for the same problem since the origin of the new coordinate system is used as the reference point in the measure of both hypervolume indicator in T-SMS-EMOA and R2 indicator in T-R2-EMOA.

### T-NSGA-II

The details of T-NSGA-II are given in Algorithm 5.

In T-NSGA-II, the size of the offspring population is the same as the size of the parent population, which is the specified population size. The next population is generated by choosing the best half solutions from the merged population: starting with points in the first non-domination front, continuing with points in the second non-domination front, and so on; if by adding all points in one front, the population size exceeds the specified population size, picking points in the descending order of crowding distance; if by adding all points with the same crowding distance, the population

**Algorithm 4** T-R2-EMOA

$P_0 \leftarrow init()$; //Initialize random population
$t \leftarrow 0$;
**while** Stop criterion not satisfied() **do**
  $q_{t+1} \leftarrow Gen(P_t)$; //Generate offspring by variation
  $P_t \leftarrow P_t \cup \{q_{t+1}\}$;
  $\{R_1, \cdots, R_v\} \leftarrow$ Nondominated-sort$(P_t)$;
  $\forall x \in R_v$ : compute $D_{Ch}(x)$; //Chebyshev distance to center of target region
  $R_{v1} \cup R_{v2} \leftarrow R_v$; //Solutions not in the target region $\rightarrow R_{v1}$; others $\rightarrow R_{v2}$
  $\forall x \in R_{v1} : r(x) \leftarrow 0$; //R2 indicator contribution
  $R_{v2} \leftarrow$ Coordinate Transformation$(R_{v2})$;
  $\forall x \in R_{v2} : r(x) \leftarrow R2(R_{v2} \backslash \{x\}; \Lambda; \mathbf{i})$; //$\mathbf{i}$ is ideal point
  **if** unique $argmin\{r(x) : x \in R_v\}$ exists **then**
    $x^* \leftarrow argmin\{r(x) : x \in R_v\}$;
  **else**
    $x^* \leftarrow argmax\{D_{Ch}(x) : x \in R_v\}$; //In case of tie, choose randomly
  **end if**
  $P_{t+1} \leftarrow P \backslash \{x^*\}$;
  $t \leftarrow t + 1$;
**end while**

**Algorithm 5** T-NSGA-II

$P_0 \leftarrow init()$; //Initialize random population
$t \leftarrow 0$;
**while** Stop criterion not satisfied() **do**
  $Q_t \leftarrow Gen(P_t)$; //Generate offsprings by variation
  $P_t \leftarrow P_t \cup Q_t$;
  $\forall x \in P_t$ : compute $D_{Ch}(x)$; //Chebyshev distance to center of target region
  $\{R_1, \cdots, R_v\} \leftarrow$ Nondominated-sort$(P_t)$;
  **for** $i = 1, \cdots, v$ **do**
    $R_{i1} \cup R_{i2} \leftarrow R_i$; //Solutions not in the target region $\rightarrow R_{i1}$; others $\rightarrow R_{i2}$
    $\forall x \in R_{i1} : D_c(x) \leftarrow 0$; //Crowding distance
    $R_{i2} \leftarrow$ Coordinate Transformation$(R_{i2})$;
    $\forall x \in R_{i2}$ : compute $D_c(x)$;
  **end for**
  $P_{t+1} \leftarrow$ half the size of $P_t$ based on rank, $D_c$ and then $D_{Ch}$;
  $t \leftarrow t + 1$;
**end while**

size still exceeds the specified population size, picking points in the ascending order of the Chebyshev distance. Unlike T-SMS-EMOA and T-R2-EMOA, no reference point is needed in T-NSGA-II.

### 4.1.2 Experimental Study

In this part, experiments are conducted on some benchmark problems, including ZDT, DTLZ and knapsack problems, to investigate performance of the proposed algorithms. In all experiments, the SBX operator with an index of 15 and polynomial mutation with an index 20 are used. The crossover and mutation probabilities are set to 1 and $1/L$, where $L$ is the number of variables. The population size and the number of evaluations are chosen to be dependent on the complexity of the test problem. Table 4.1 shows the population size and the number of evaluations used on different test problems.

**Table 4.1:** Population Size and Number of Evaluation

| Problems | Population Size | NE |
|---|---|---|
| ZDT1 | 100 | 10000 |
| ZDT2-3 | 100 | 20000 |
| DTLZ1-2 | 100 | 30000 |
| knapsack-250-2<br>knapsack-500-2 | 200 | 200000 |
| knapsack-250-3<br>knapsack-500-3 | 250 | 500000 |

#### Bi-objective ZDT Problems

Three bi-objective ZDT problems are considered. The 30-variable ZDT1 problem has a convex Pareto front which is a connected curve and can be determined by $f_2(x) = 1 - \sqrt{f_1(x)}$. The true PF spans continuously in $f_1 \in [0, 1]$. Four different target regions are chosen to observe the performance of T-SMS-EMOA, T-R2-EMOA and T-NSGA-II. The first target region covers the entire PF with the lower bound (0, 0) and the upper bound (1, 1). The second target region restricts preferred solutions to the central part of the PF and its lower bound is (0.1, 0.1), upper bound is (0.5, 0.5). The third and fourth target regions take two ends of the PF respectively and have their lower bounds to be (0, 0.6) and (0.6, 0), upper bounds to be (0.3, 1) and (1, 0.3).

Figure 4.2 $\sim$ Figure 4.4 show PF approximations obtained from the proposed algorithms on the four different target regions in a random single run. The target

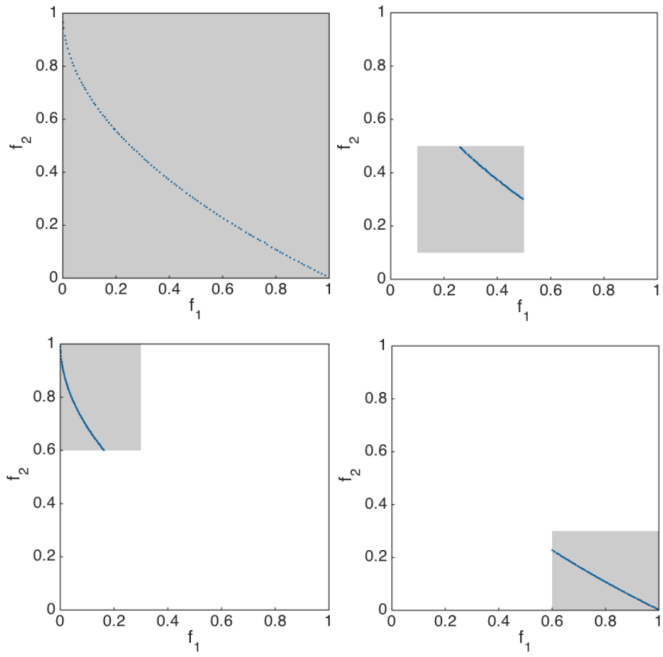



**Figure 4.2:** Representative PF approximations of T-SMS-EMOA on ZDT1.

regions are highlighted by gray boxes. It is observed that all three algorithms can find well-distributed and well-converged solutions on the PF in the target regions and no outliers exist. The solution set obtained by T-SMS-EMOA is more uniform than the solution sets obtained by the other two algorithms. It is also observable from the upper left graph in Figure 4.3 that the R2 indicator has a bias towards the center of the PF.

When examining the performance by the hypervolume metric, the hypervolume value of the obtained solution set is calculated within the target region by normalizing the values of each objective to the values between 0 and 1 and using the lower bound of the target region as the reference point for the maximization problem and the upper bound of target region as the reference point for the minimization problem. Table 4.2 shows the median and variance of hypervolume over 30 runs. The statistical results correspond to the observation that T-SMS-EMOA outperforms T-R2-EMOA and T-NSGA-II slightly. The original SMS-EMOA, R2-EMOA and NSGA-II are also involved in the comparison and the results of the original MOEAs are obtained by firstly presenting the target region as constraints in the description of the problem;

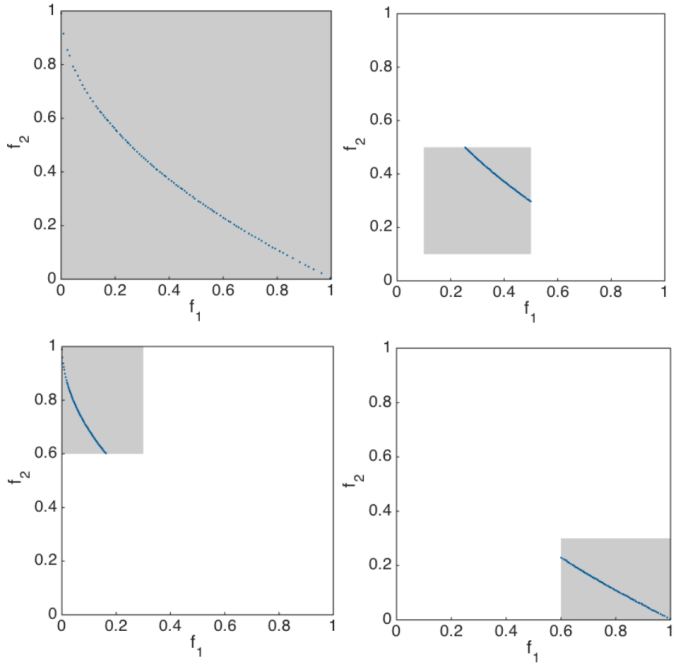



**Figure 4.3:** Representative PF approximations of T-R2-EMOA on ZDT1.

and secondly, making a change in the description of the problem. It is demonstrated that the new algorithms obtain higher hypervolume value than original MOEAs with no constraint descriptions in the problem. Although the results of the proposed algorithms are not better than original MOEAs with constraints on the range of objectives, experiments show that the proposed algorithms can reduce computation time dramatically on this problem.

In the table, the symbol of $*$ on the values for the same target region means the medians of these algorithms are significantly indifferent. The Mann-Whitney U test (also called the Mann-Whitney-Wilcoxon, Wilcoxon rank-sum test, or Wilcoxon-Mann-Whitney test) is used to determine if the medians of different algorithms for the same problem are significantly indifferent. The chances that the medians of T-SMS-EMOA and T-R2-EMOA are indifferent have been observed.

Next, circle target regions are adopted on the 30-variable ZDT2 and ZDT3 problems. ZDT2 has a concave Pareto front and ZDT3 has a disconnected set of Pareto front which consists of five non-contiguous convex parts. A circle with a center point (1, 0) and radius 0.5 intersects the whole PF of ZDT2 at its one end and a circle with

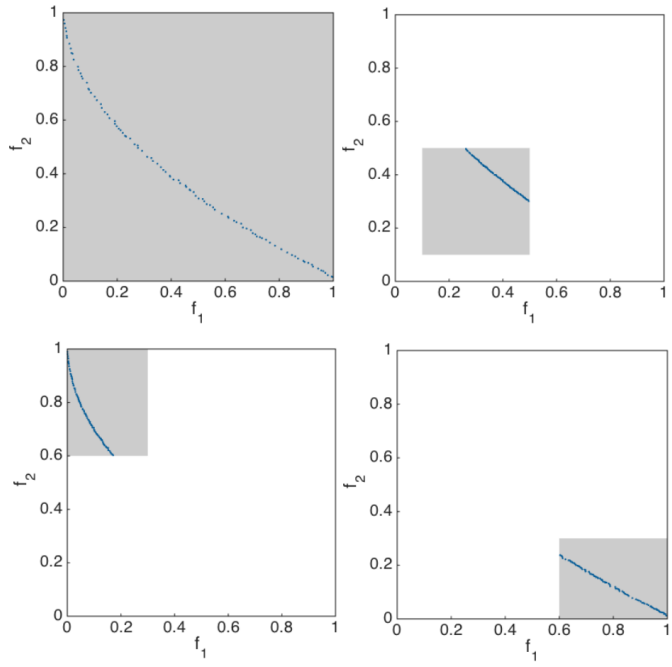



**Figure 4.4:** Representative PF approximations of T-NSGA-II on ZDT1.

a center point (0.6, 0.5) and radius 0.3 intersects the whole PF at its central part. The two different circles are chosen as examples for target regions on ZDT2 problem. Experiments for a circle with a center point (0.3, 0.1) and radius 0.3 as target region are conducted on ZDT3 problem.

Figure 4.5 shows PF approximations from T-SMS-EMOA in these target regions. Similar figures can also be achieved by T-R2-EMOA and T-NSGA-II. In the graph, the target regions are purple circles and center points are red points. Orange points denote the results obtained from T-SMS-EMOA with provided preference information. The blue points show the entire true PF of ZDT2 and ZDT3 problems. Statistical results of the median of hypervolume for three algorithms in 30 independent runs on each target region are shown in Table 4.3.

### Tri-objective DTLZ Problems

Next, tri-objective DTLZ1 and DTLZ2 problems are involved in the experiments. The 7-variable DTLZ1 problem has a linear Pareto optimal front which is a three-

**Table 4.2:** The median, variance of hypervolume and average computation time (Sec.) on ZDT1 with respect to different target regions and different algorithms

| New Algorithms | | T-SMS-EMOA | T-R2-EMOA | T-NSGA-II |
|---|---|---|---|---|
| Target Region | Metric | | | |
| (0,0)(1,1) | HV(m) | 0.6580 | 0.6566 | 0.6425 |
| | Variance | 6.4e-06 | 1.4e-06 | 1.0e-05 |
| | Time | 24.99 | 74.01 | 0.21 |
| (0.1,0.1)(0.5,0.5) | HV(m) | 0.1640* | 0.1638* | 0.1543 |
| | Variance | 1.4e-06 | 1.5e-06 | 8.9e-06 |
| | Time | 10.30 | 23.61 | 0.19 |
| (0,0.6)(0.3,1) | HV(m) | 0.8110 | 0.8103 | 0.7936 |
| | Variance | 5.9e-06 | 6.0e-06 | 4.4e-05 |
| | Time | 12.86 | 31.78 | 0.20 |
| (0.6,0)(1,0.3) | HV(m) | 0.6255* | 0.6233* | 0.6079 |
| | Variance | 8.9e-06 | 6.7e-06 | 4.5e-05 |
| | Time | 11.45 | 27.92 | 0.21 |
| Original Algorithms (Constraints) | | SMS-EMOA | R2-EMOA | NSGA-II |
| (0,0)(1,1) | HV(m) | 0.6621 | 0.6610 | 0.6609 |
| | Variance | 8.9e-11 | 1.2e-08 | 5.3e-08 |
| | Time | 108.57 | 314.99 | 0.25 |
| (0.1,0.1)(0.5,0.5) | HV(m) | 0.1694 | 0.1693 | 0.1690 |
| | Variance | 1.6e-11 | 1.1e-11 | 6.2e-09 |
| | Time | 106.32 | 274.05 | 0.23 |
| (0,0.6)(0.3,1) | HV(m) | 0.8197 | 0.8185 | 0.8191 |
| | Variance | 1.6e-08 | 4.6e-08 | 2.9e-08 |
| | Time | 105.73 | 271.00 | 0.21 |
| (0.6,0)(1,0.3) | HV(m) | 0.6364 | 0.6348 | 0.6356 |
| | Variance | 3.2e-09 | 2.2e-08 | 3.8e-08 |
| | Time | 101.82 | 283.3 | 0.22 |
| Original Algorithms | | SMS-EMOA | R2-EMOA | NSGA-II |
| (0,0)(1,1) | HV(m) | 0.6558 | 0.6566 | 0.6362 |
| | Variance | 1.6e-06 | 8.5e-07 | 3.5e-05 |
| | Time | 26.77 | 73.34 | 0.21 |
| (0.1,0.1)(0.5,0.5) | HV(m) | 0.1545 | 0.1585 | 0.1236 |
| | Variance | 4.7e-06 | 2.2e-06 | 4.4e-05 |
| | Time | 24.17 | 74.85 | 0.20 |
| (0,0.6)(0.3,1) | HV(m) | 0.8012 | 0.7972 | 0.7649 |
| | Variance | 6.3e-06 | 6.4e-06 | 0.00013 |
| | Time | 24.85 | 71.90 | 0.20 |
| (0.6,0)(1,0.3) | HV(m) | 0.6119* | 0.6110* | 0.5604 |
| | Variance | 2.3e-05 | 7.4e-06 | 0.00014 |
| | Time | 26.29 | 78.93 | 0.20 |

**Table 4.3:** The median hypervolume of ZDT2 and ZDT3 with a circular target region.

| Algorithm | T-SMS-EMOA | T-R2-EMOA | T-NSGA-II |
|---|---|---|---|
| Target Region | | | |
| ZDT2 (1,0) 0.5 | 0.3168 | 0.3167 | 0.3159 |
| ZDT2 (0.6,0.5) 0.3 | 0.3257 | 0.3256 | 0.3234 |
| ZDT3 (0.3,0.1) 0.3 | 0.3377 | 0.3375 | 0.3365 |

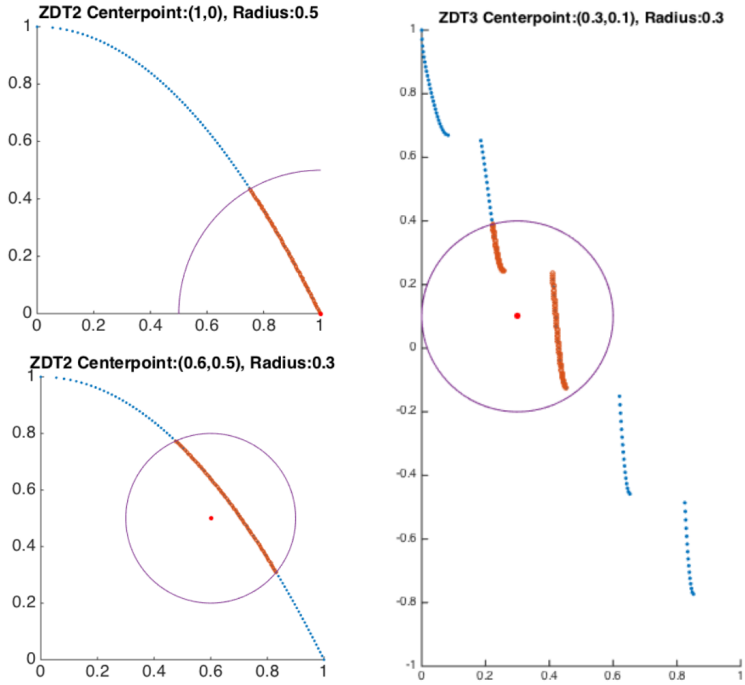



**Figure 4.5:** Representative PF approximations of T-SMS-EMOA on ZDT2 and ZDT3 with respect to different circular target regions.

dimensional, triangular hyperplane. A sphere with the center point (0.3, 0.3, 0.3) and radius 0.3 is defined as the target region for tri-objective DTLZ1 problem. The 11-variable DTLZ2 problem has a three-dimensional concave Pareto front. A box with the lower bound (0.4, 0.4, 0.2) and upper bound (0.8, 0.8, 0.8) is defined as the target region for DTLZ2 problem.

Figure 4.6 shows PF approximation of tri-objective DTLZ1 problem. The graphs in the upper row are solutions from T-SMS-EMOA, graphs in the middle row are solutions from T-R2-EMOA and graphs in the lower row are solutions from T-NSGA-II. The blue points show the true entire PFs. The transparent spheres depict target regions and red points are solutions obtained by T-SMS-EMOA, T-R2-EMOA and T-NSGA-II. It can be observed that T-SMS-EMOA behaves the best in three algorithms. Statistical results of the median of hypervolume in Table 4.4 are coincident with our observation.

Figure 4.7 shows PF approximations of tri-objective DTLZ2 problem. The trans-

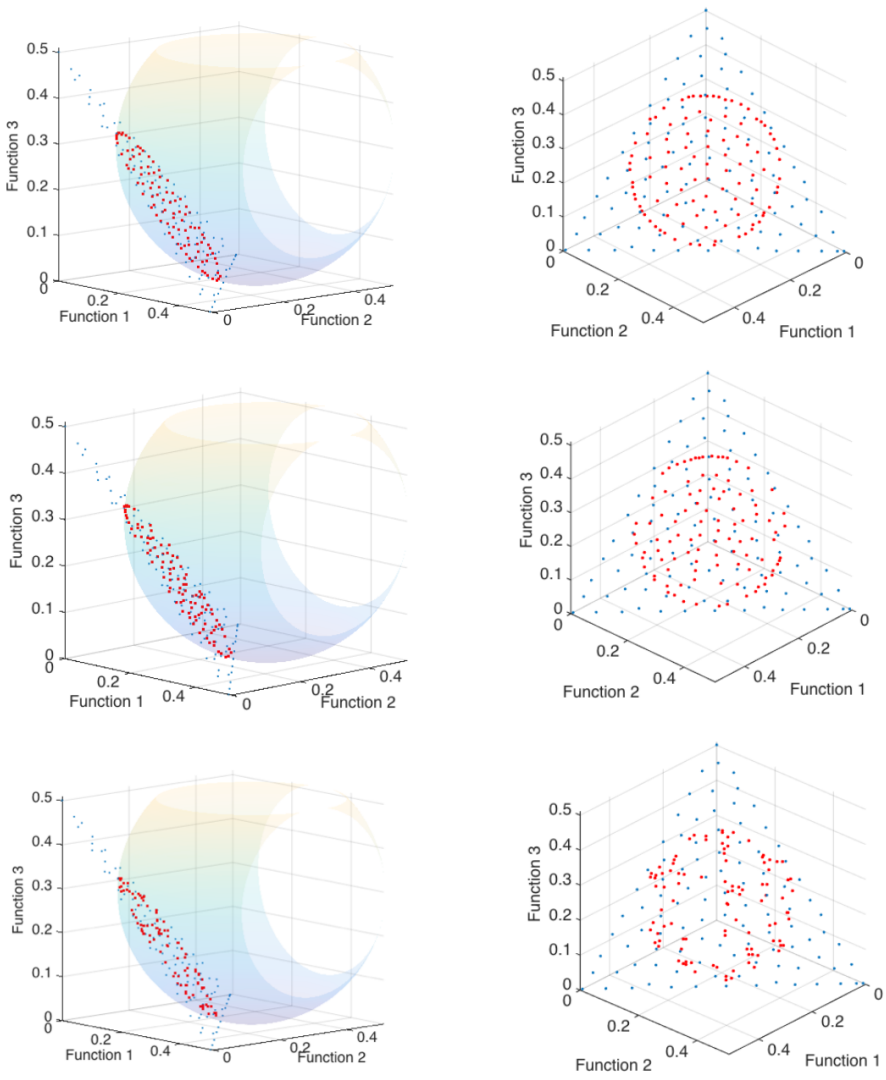



**Figure 4.6:** Representative PF approximations of T-SMS-EMOA(the upper row), T-R2-EMOA(the middle row) and T-NSGA-II(the lower row) with a spherical target region on tri-objective DTLZ1 problem.

**Table 4.4:** The median hypervolume of tri-objective DTLZ1 problem with a spherical target region.

| Algorithm / Target Region | T-SMS-EMOA | T-R2-EMOA | T-NSGA-II |
|---|---|---|---|
| (0.3,0.3,0.3) 0.3 | 0.8028 | 0.7992 | 0.7823 |

parent boxes depict target regions and red points are solutions obtained by T-SMS-EMOA. Statistical results of the median of hypervolume for three algorithms and original MOEAs with constraints description in 30 independent runs on cubic target regions are shown in Table 4.5. It is observable that the best result is achieved by T-SMS-EMOA and all new algorithms outperform original MOEAs except for R2-EMOA.

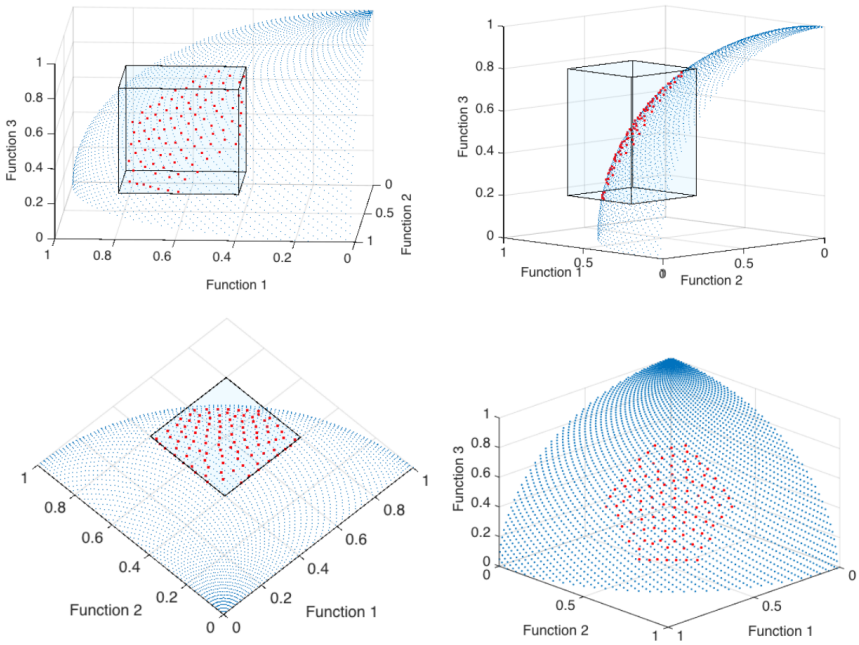



**Figure 4.7:** Representative PF approximations from T-SMS-EMOA in a cubic target region on tri-objective DTLZ2 problem.

**Table 4.5:** The median hypervolume on tri-objective DTLZ2 problem with a cubic target region.

| MOEA / Target Region | T-SMS -EMOA | T-R2 -EMOA | T-NSGA -II | SMS-EMOA | R2-EMOA | NSGA -II |
|---|---|---|---|---|---|---|
| (0.4,0.4,0.2) (0.8,0.8,0.8) | 0.4632 | 0.4303 | 0.4189 | 0.3369 | 0.4351 | 0.4185 |

### Knapsack Problems

Knapsack Problems have been studied first by Dantzig in the late 1950′s [19]. The problem is a general, understandable, and one of the most representative discrete optimization problems. At the same time, it is difficult to solve (NP-hard). The Multi-objective 0/1 Knapsack Problems from Zitzler and Thiele [142] are used as discrete test problems in this part. Formally, the multi-objective 0/1 knapsack problem can be formulated as the following maximization problem:

$$\begin{aligned}
&\text{maximize} && f(x) = (f_1(x), f_2(x), \cdots, f_m(x)) \\
&\text{subject to} && \sum_{j=1}^{n} w_{ij} x_j \leq c_i, && i = 1, \cdots, m \\
& && x_j \in \{0, 1\}, && j = 1, \cdots, n \\
&\text{where} && f_i(x) = \sum_{j=1}^{n} p_{ij} x_j, && i = 1, \cdots, m.
\end{aligned}$$

Here $n$ is the number of items and $m$ is the number of knapsacks, $w_{ij}$ is the weight of item $j$ according to knapsack $i$, $p_{ij}$ is the profit of item $j$ according to knapsack $i$ and $c_i$ is the capacity of knapsack $i$. The Multi-objective 0/1 Knapsack Problem is to find Pareto optimal vectors x $= (x_1, x_2, \cdots, x_n)$ and $x_j = 1$ when item $j$ is selected and $x_j = 0$ otherwise.

Figure 4.8 shows solutions obtained in a random single run when the number of knapsack is 2 and the number of items is 250. The results of SMS-EMOA, R2-EMOA and NSGA-II give the entire PF approximations. The target region for T-SMS-EMOA, T-R2-EMOA and T-NSGA-II is highlighted by the gray box. The lower bound is (9000, 9000), the upper bound is (9800, 9800). It can be observed that the proposed algorithms can find solutions in the target region except for several outliers from T-NSGA-II.

Statistical results of the median of hypervolume are presented in Table 4.6. No constraints of the target region are converted in the description of the problem for the results of original MOEAs in the table. In the experiments, two and three objectives are taken into consideration, in combination with 250 and 500 items. The test data sets are from [142]. The target region of knapsack-250-2 is from (9000, 9000) to (9800, 9800), of knapsack-250-3 is from (8500, 8500, 8500) to (10000, 10000, 10000), of knapsack-500-2 is from (18000, 18000) to (20000, 20000), of knapsack-500-3 is from

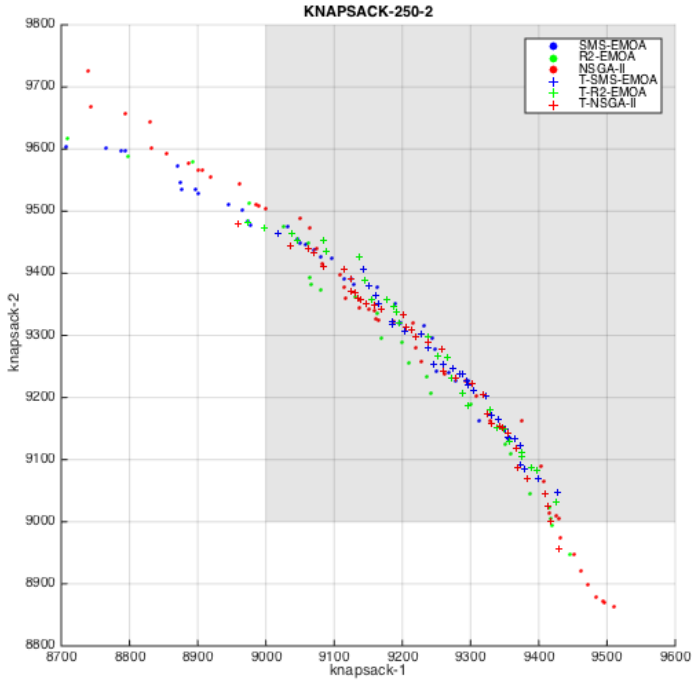



**Figure 4.8:** Representative PF approximations of knapsack-250-2 problem.

(17000, 17000, 17000) to (19000, 19000, 19000). The better results from the proposed algorithms than from the original algorithms can be seen.

### 4.1.3 Enhanced Algorithms and Experiments

As mentioned before, the second ranking criterion (Hypervolume, R2 indicator or crowding distance) in the proposed algorithms only works for solutions in the target region, which means if the intersection of the target region and true PF is empty, the second ranking criterion becomes useless. Under this condition, well-distributed solutions can not be obtained because solutions are guided only by non-dominated sorting and the Chebyshev distance. In addition, if multiple target regions are specified, sometimes the obtained solutions only converge to one target region. Inspired by some ideas from R-NSGA-II [30], two methods are adopted to overcome these limitations and extend the proposed algorithms.

**Table 4.6:** The median hypervolume of Knapsack problems with respect to different target regions.

| Algorithms / Problems | T-SMS -EMOA | T-R2 -EMOA | T-NSG A-II | SMS- EMOA | R2- EMOA | NSGA -II |
|---|---|---|---|---|---|---|
| Knapsack -250-2 | 0.2117 | 0.2160 | 0.2230 | 0.2102 | 0.2099 | 0.1821 |
| Knapsack -250-3 | 0.0295 | 0.0296 | 0.0311 | 0.0204 | 0.0293 | 0.0085 |
| Knapsack -500-2 | 0.3273 | 0.3318 | 0.3230 | 0.3225 | 0.3272 | 0.2857 |
| Knapsack -500-3 | 0.1936 | 0.1855 | 0.1713 | 0.1699 | 0.1747 | 0.0718 |

### Separate Population to Different Targets

The first method can attract the population to different targets and it is used in the calculation of both the second ranking criterion (Hypervolume, R2 indicator or crowding distance) and the third ranking criterion (the Chebyshev distance). The aim of this method is to support multiple targets.

Taking R2 indicator as an example, after coordinate transformation, for all target regions, R2 indicator values of all solutions on the worst ranked front are calculated and the solutions are sorted in descending order of R2 indicator. Thereafter, R2 indicator values are replaced by *R2 indicator ranks*: solutions with the largest R2 indicator values for all target regions are assigned the same largest *R2 indicator rank*, solutions having next-to-largest R2 indicator values for all target regions are assigned the same next-to-largest *R2 indicator rank*, and so on, until the number of solutions which have been assigned the *R2 indicator rank* for each target region reaches its proportion in population. For the solutions assigned more than one ranks, the largest rank number is kept as its final *R2 indicator rank*. If the even distribution of solutions in all target regions is expected, the proportion of each target region should be divided equally between all target regions. For example, when the number of target regions is two, the proportion for each target region should be $1/2$ and the number of solutions being assigned a *R2 indicator rank* should reach half of the size of the worst ranked front. Lastly, for solutions that haven't been assigned a *R2 indicator rank*, their R2 indicator values should be mapped to values smaller than the smallest *R2 indicator rank*. One way to do this is to calculate their R2 indicator values in the combined region of all target regions and normalize them to values lower than all *R2 indicator ranks*. By this way, solutions with larger R2 indicator values in each target region are emphasized more.

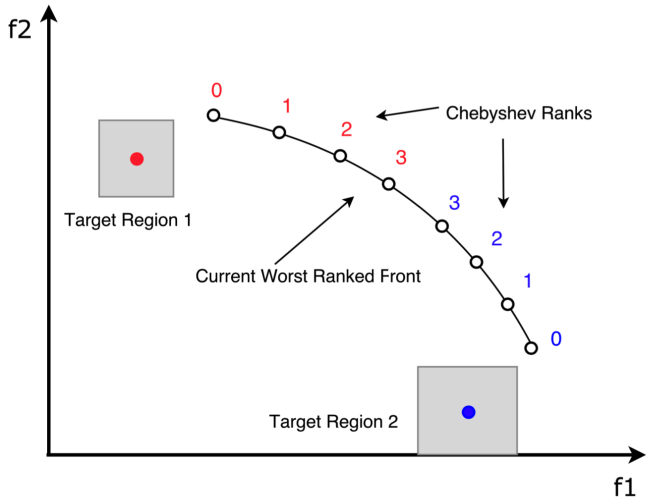



**Figure 4.9:** An example of assigning Chebyshev ranks to solutions on the current worst ranked front.

The *Chebyshev rank* is used in place of the Chebyshev distance when the method is added to work with the third ranking criterion. Firstly, the Chebyshev distances of each solution to all targets are calculated. The solutions are sorted in ascending order of distance. The *Chebyshev rank* of the solutions closest to all targets is assigned to be the same smallest rank of zero, the *Chebyshev rank* of the solutions having next-to-smallest Chebyshev distance is assigned to be the same next-to-smallest rank of one, and so on, until the number of solutions which have been assigned the *Chebyshev rank* for each target reaches its proportion in population. Similarly, the lowest rank of a solution is used as its Chebyshev distance if a solution is close enough to more than one target. Figure 4.9 shows an example of assigning *Chebyshev ranks* to solutions on the current worst ranked front. In the example, red Chebyshev ranks are assigned by sorting solutions in ascending order of the Chebyshev distance to Target Region 1 and blue Chebyshev ranks are assigned by sorting solutions in ascending order of the Chebyshev distance to Target Region 2, solutions with lower ranks are encouraged to remain in the population. The worst solution will be chosen from two solutions having the Chebyshev rank 3 randomly.

### Improve Diversity

Under the circumstance that the second ranking criterion (Hypervolume, R2 indicator or crowding distance) doesn't work well, the diversity is lost and the solution set would converge to one point when the number of evaluations is high enough. To solve the problem, a parameter $\epsilon$ is introduced in the proposed algorithms and works with the third ranking criterion (the Chebyshev distance) to improve the diversity. Firstly, the solution with the shortest Chebyshev distance to the target is picked out. Thereafter, all other solutions having a Chebyshev distance less than the sum of the current shortest Chebyshev distance and $\epsilon$ are assigned a relatively large distance to discourage them to remain in the population. Then, another solution (not already considered earlier) is picked and the above procedure is performed again. This way, only one solution within a $\epsilon$-neighborhood is emphasized and the diversity of the solution set is improved.

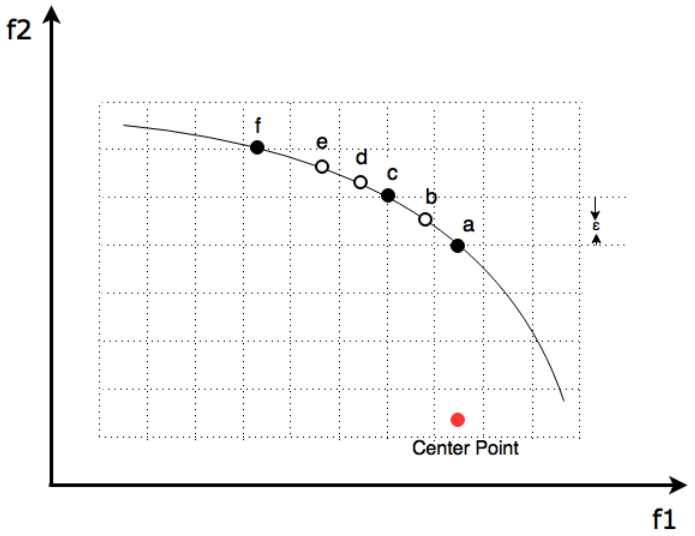



**Figure 4.10:** Illustration of $\epsilon$ parameter.

Figure 4.10 illustrates how the parameter $\epsilon$ takes effect when involved in the calculation of the Chebyshev distance. The Chebyshev distance between two vectors is the greatest of their differences along any coordinate dimension. Therefore, in the graph, the Chebyshev distances of solutions on the current font to the center point are distances along $f_2$. Apparently, point $a$ is the point with the shortest Chebyshev distance to the center point, point $b$ has a Chebyshev distance less than the sum of $a$'s

Chebyshev distance and $\epsilon$, thus, it will be assigned an relatively large distance value. So will point $d$ and $e$. Point $b$, $d$ and $e$ are discouraged to remain in the population.

### Algorithm Structure

When using above two methods in basic algorithms, Algorithm 6, 7, and 8 show the structures of enhanced algorithms.

### Experiments on Multiple Target Regions

If multiple target regions of interest can be found simultaneously, the DM can make a more effective selection towards finding the ultimate preferred solution(s). The enhanced algorithms can guide the search toward multiple target regions. Three pair of spherical target regions are used on tri-objective DTLZ1 problem separately to demonstrate differences of results between T-SMS-EMOA (Figure 4.11), T-R2-EMOA (Figure 4.12) and T-NSGA-II (Figure 4.13). The center point and radius of two target regions are shown above each graph. The first pair of target regions have the same radius and both center points are on the PF which is a three-dimensional, triangular hyperplane. The second pair of target regions have the same radius, but one center point is on the PF, the other is not. The third pair of target regions have different radius, and both center points are on the PF. Experimental results over consecutive 30 runs show that all three algorithms can obtain uniform solutions in two target regions, no outliers exist. When the assigned population size is 100, each target region obtains 50 solutions for all 30 runs. While when the number of runs increases to 50, the case of 49 and 51 solutions in two regions also appears once.

In the above experiments, it has been specified that solutions are equally distributed in multiple target regions. For the population size of 100, this means that there are 50 solutions in each target region. It is also possible that we hope for a different proportion of solutions for each target region. For example, the real intersection areas of the third pair of target regions and the PF are obviously different. Therefore, $1/4th$ of population size is specified as the number of obtained solutions in the small intersection area and $3/4th$ of population size is specified as the number of obtained solutions in the larger intersection area. Figure 4.14 shows the difference between equally distributed solutions (left graph) and solutions with newly-specified proportion for each target region (right graph) of T-SMS-EMOA.

Furthermore, when there is no intersection between the target region and the PF, the enhanced algorithms can still obtain solutions close to the target region. Fig-

**Algorithm 6** T-SMS-EMOA Enhanced version

$P_0 \leftarrow init()$; //Initialize random population
$t \leftarrow 0$;
**while** Stop criterion not satisfied() **do**
  $q_{t+1} \leftarrow Gen(P_t)$; //Generate offspring by variation
  $P_t \leftarrow P_t \cup \{q_{t+1}\}$;
  $\{R_1, \cdots, R_v\} \leftarrow$ Nondominated-sort$(P_t)$;
  **for** all targets **do**
    $\forall x \in R_v$ : compute $D_{Ch}(x)$; //Chebyshev distance to current target
    $R'_v \leftarrow R_v$; //Sorted in ascending order of $D_{Ch}$
    **for** unlabeled $x \in R'_v$ **do** //Start from the first solution in $R'_v$
      $Label(x)$;
      **for** $x' \in R'_v \setminus X$ **do** //$X$ is the set of labeled points
        **if** $D_{Ch}(x') < D_{Ch}(x) + \epsilon$ **then**
          $Label(x')$; $D_{Ch}(x') \leftarrow$ relatively large value;
        **end if**
      **end for**
    **end for**
    $R_v \leftarrow R'_v$; //Sorted in ascending order of $D_{Ch}$
    **for** pre-defined number of $x \in R_v$ **do**
      $D_{Ch}(x) \leftarrow D_{Ch}\_rank(x)$; //Start from the smallest; keep smaller rank
    **end for**
  **end for**
  **for** unranked $x \in R_v$ **do**
    $D_{Ch}(x) \leftarrow Normalized\_D_{Ch}(x)$; //$Normalized\_D_{Ch}$ >largest $D_{Ch}\_rank$
  **end for**
  **for** all targets **do**
    $R_{v1} \cup R_{v2} \leftarrow R_v$; //solutions not in the target region $\rightarrow R_{v1}$; others $\rightarrow R_{v2}$
    $\forall x \in R_{v1} : HC(x) \leftarrow 0$;
    $R_{v2} \leftarrow$ Coordinate Transformation$(R_{v2})$;
    $\forall x \in R_{v2} : HC(x) \leftarrow HV(R_{v2}) - HV(R_{v2} \backslash x)$;
    $R'_v \leftarrow R_v$; //Sorted in descending order of $HC$ values
    **for** pre-defined number of $x \in R'_v$ **do**
      $HC(x) \leftarrow HC\_rank(x)$; //Start from the largest; keep larger rank
    **end for**
  **end for**
  **for** unranked $x \in R_v$ **do**
    $HC(x) \leftarrow Normalized\_HC(x)$; //$Normalized\_HC$ <smallest $HC\_rank$
  **end for**
  **if** unique $argmin\{HC(x) : x \in R_v\}$ exists **then**
    $x^* \leftarrow argmin\{HC(x) : x \in R_v\}$;
  **else**
    $x^* \leftarrow argmax\{D_{Ch}(x) : x \in R_v\}$; //In case of tie, choose randomly
  **end if**
  $P_{t+1} \leftarrow P \backslash \{x^*\}$; $t \leftarrow t + 1$;
**end while**

**Algorithm 7** T-R2-EMOA Enhanced version

$P_0 \leftarrow init()$; //Initialize random population
$t \leftarrow 0$;
**while** Stop criterion not satisfied() **do**
  $q_{t+1} \leftarrow Gen(P_t)$; //Generate offspring by variation
  $P_t \leftarrow P_t \cup \{q_{t+1}\}$;
  $\{R_1, \cdots, R_v\} \leftarrow$ Nondominated-sort$(P_t)$;
  **for** all targets **do**
    $\forall x \in R_v$ : compute $D_{Ch}(x)$; //Chebyshev distance to current target
    $R'_v \leftarrow R_v$; //Sorted in ascending order of $D_{Ch}$
    **for** unlabeled $x \in R'_v$ **do** //Start from the first solution in $R'_v$
      $Label(x)$;
      **for** $x' \in R'_v \setminus X$ **do** //$X$ is the set of labeled points
        **if** $D_{Ch}(x') < D_{Ch}(x) + \epsilon$ **then**
          $Label(x')$; $D_{Ch}(x') \leftarrow$ relatively large value;
        **end if**
      **end for**
    **end for**
    $R_v \leftarrow R'_v$; //Sorted in ascending order of $D_{Ch}$
    **for** pre-defined number of $x \in R_v$ **do**
      $D_{Ch}(x) \leftarrow D_{Ch}\_rank(x)$; //Start from the smallest; keep smaller rank
    **end for**
  **end for**
  **for** unranked $x \in R_v$ **do**
    $D_{Ch}(x) \leftarrow Normalized\_D_{Ch}(x)$; //$Normalized\_D_{Ch}$ >largest $D_{Ch}\_rank$
  **end for**
  **for** all targets **do**
    $R_{v1} \cup R_{v2} \leftarrow R_v$; //solutions not in the target region $\rightarrow R_{v1}$; others $\rightarrow R_{v2}$
    $\forall x \in R_{v1} : r(x) \leftarrow 0$;
    $R_{v2} \leftarrow$ Coordinate Transformation$(R_{v2})$;
    $\forall x \in R_{v2} : r(x) \leftarrow R2(P \setminus \{x\}; \Lambda; \mathbf{i})$; //$\mathbf{i}$ is ideal point
    $R'_v \leftarrow R_v$; //Sorted in descending order of $R2$ values
    **for** pre-defined number of $x \in R'_v$ **do**
      $r(x) \leftarrow r\_rank(x)$; //Start from the largest, keep larger rank
    **end for**
  **end for**
  **for** unranked $x \in R_v$ **do**
    $r(x) \leftarrow Normalized\_r(x)$; //$Normalized\_r(x)$ <smallest $r\_rank$
  **end for**
  **if** unique $argmin\{r(x) : x \in R_v\}$ exists **then**
    $x^* = argmin\{r(x) : x \in R_v\}$;
  **else**
    $x^* = argmax\{D_{Ch}(x) : x \in R_v\}$; //In case of tie, choose randomly
  **end if**
  $P_{t+1} \leftarrow P \setminus \{x^*\}$; $t \leftarrow t + 1$;
**end while**

**Algorithm 8** T-NSGA-II Enhanced version

$P_0 \leftarrow init()$; //Initialize random population
$t \leftarrow 0$;
**while** Stop criterion not satisfied() **do**
  $Q_t \leftarrow Gen(P_t)$; //Generate offsprings by variation
  $P_t = P_t \cup Q_t$;
  **for** all targets **do**
    $\forall x \in P_t$ : compute $D_{Ch}(x)$; //Chebyshev distance to current target
    $P'_t \leftarrow P_t$; //Sorted in ascending order of $D_{Ch}$
    **for** unlabeled $x \in P'_t$ **do** //Start from the first in $P'_t$
      $Label(x)$;
      **for** $x' \in P'_t \setminus X$ **do** //$X$ is the set of points have been labelled
        **if** $D_{Ch}(x') < D_{Ch}(x) + \epsilon$ **then**
          $Label(x')$;
          $D_{Ch}(x') \leftarrow$ relatively large value;
        **end if**
      **end for**
    **end for**
    $P_t \leftarrow P'_t$; //Sorted in ascending order of $D_{Ch}$
    **for** pre-defined number of $x \in P_t$ **do**
      $D_{Ch}(x) \leftarrow D_{Ch}\_rank(x)$; //Start from the smallest; keep smaller rank
    **end for**
  **end for**
  **for** unranked $x \in P_t$ **do**
    $D_{Ch}(x) \leftarrow Normalized\_D_{Ch}(x)$; //$Normalized\_D_{Ch}$ >largest $D_{Ch}\_rank$
  **end for**
  $\{R_1, \cdots, R_v\} \leftarrow$ Nondominated-sort($P_t$);
  **for** $i = 1, \cdots, v$ **do**
    **for** all targets **do**
      $R_{i1} \cup R_{i2} \leftarrow R_i$; //Solutions not in target region $\rightarrow R_{i1}$; others $\rightarrow R_{i2}$
      $\forall x \in R_{i1} : D_c(x) \leftarrow 0$; //Crowding distance
      $R_{i2} \leftarrow$ Coordinate Transformation($R_{i2}$);
      $\forall x \in R_{i2}$ : compute $D_c(x)$;
      $R'_i \leftarrow R_i$; //Sorted in descending order of $D_c$
      **for** pre-defined number of $x \in R'_i$ **do**
        $D_c(x) \leftarrow D_c\_rank(x)$; //Start from the largest $D_c\_rank : |R'_i|$; keep larger rank if already assigned a rank.
      **end for**
    **end for**
    **for** unranked $x \in R'_i$ **do**
      $D_c(x) \leftarrow Normalized\_D_c(x)$; //$Normalized\_D_c(x)$ <smallest $D_c\_rank$
    **end for**
  **end for**
  $P_{t+1} \leftarrow$ half the size of $P_t$ based on rank, $D_c$ and then $D_{Ch}$;
  $t \leftarrow t + 1$;
**end while**

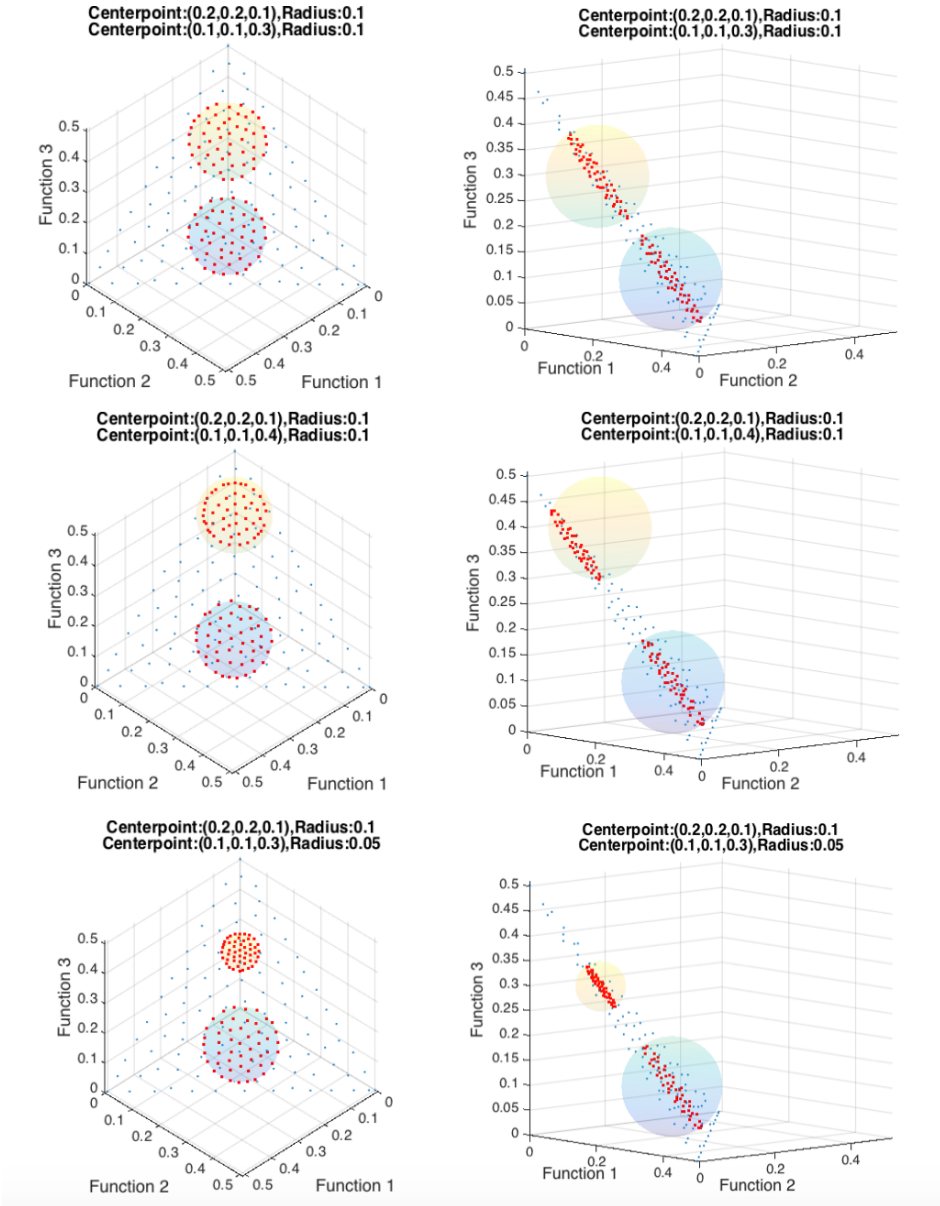



**Figure 4.11:** Representative PF approximations of T-SMS-EMOA with two spherical target regions on tri-objective DTLZ1.

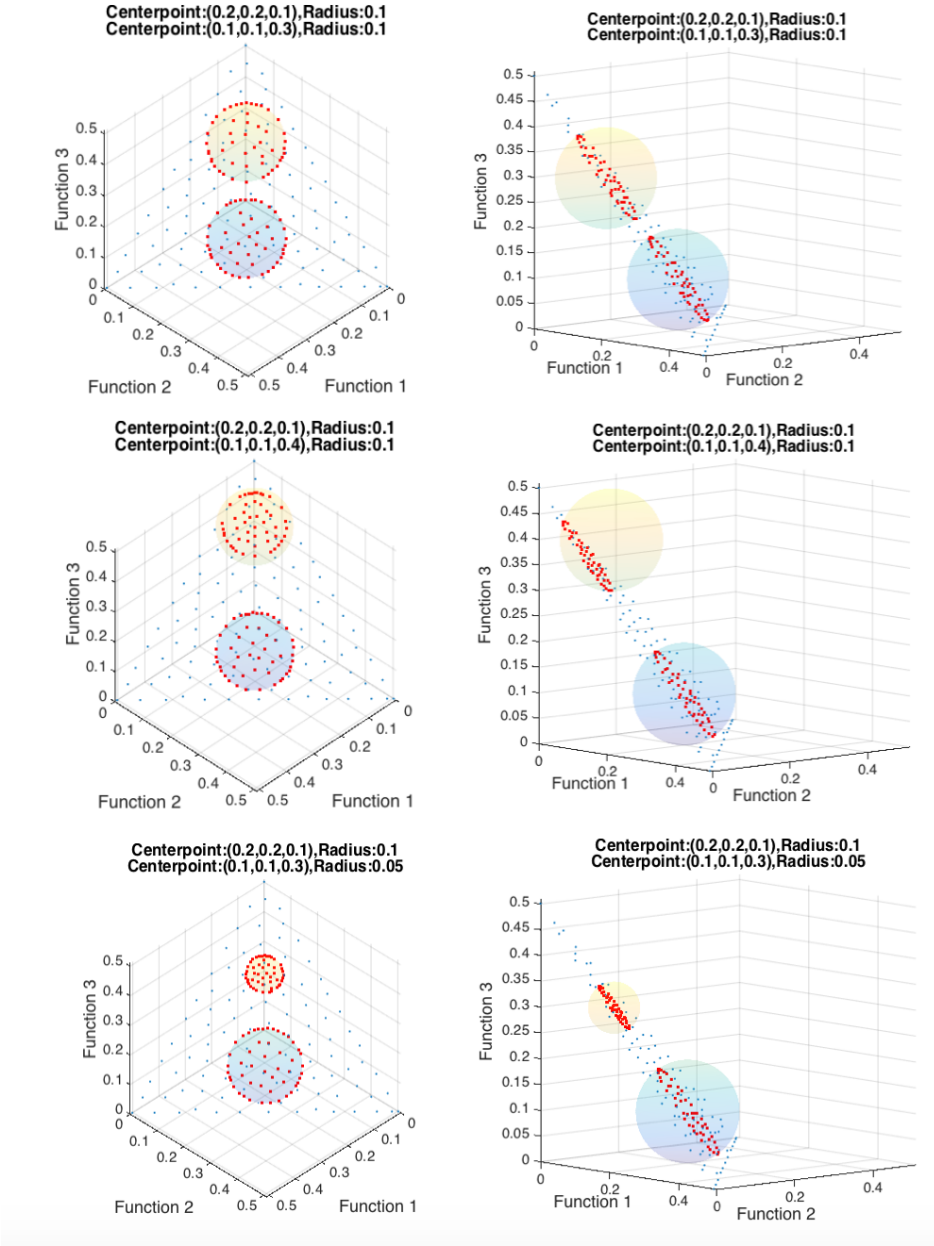



**Figure 4.12:** Representative PF approximations of T-R2-EMOA with two spherical target regions on tri-objective DTLZ1.

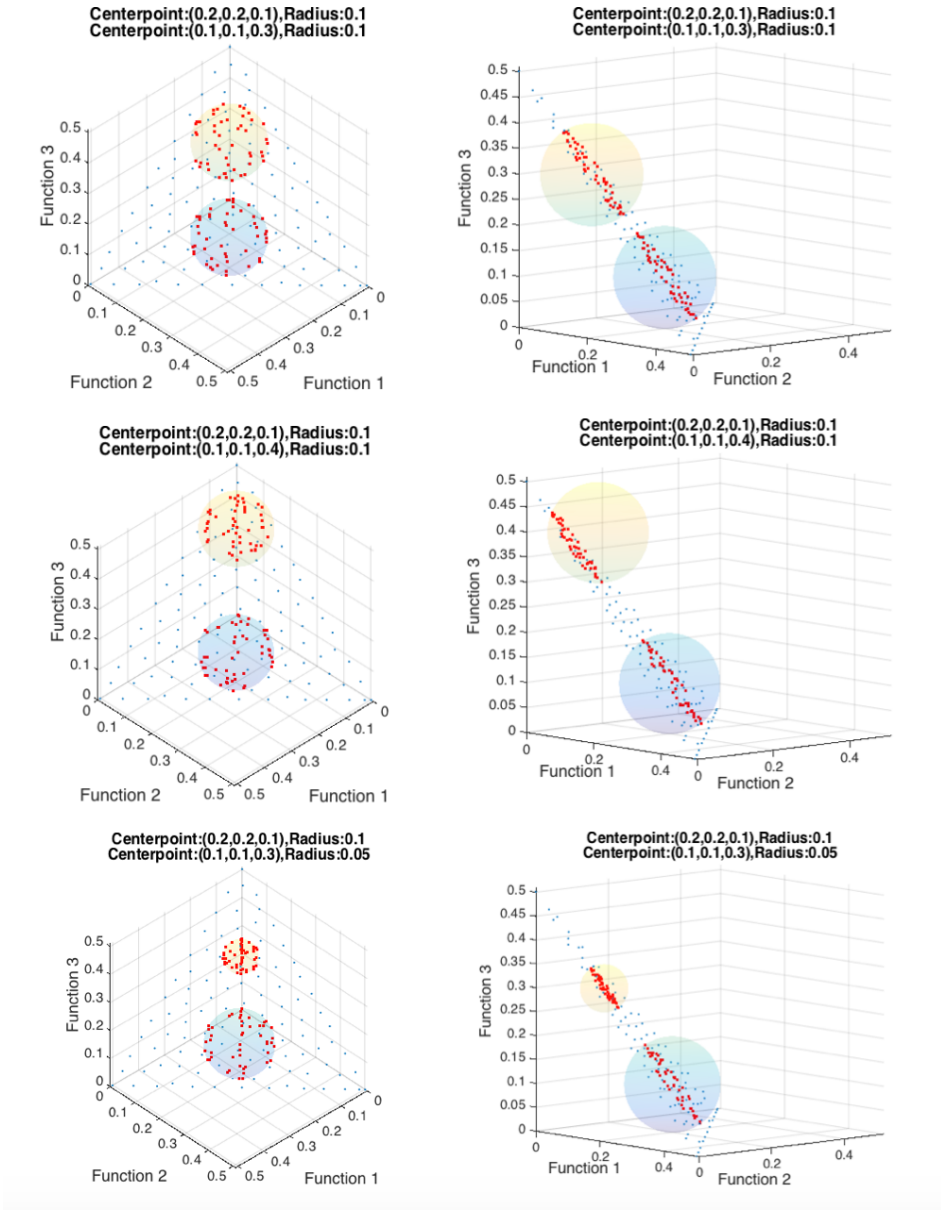



**Figure 4.13:** Representative PF approximations of T-NSGA-II with two spherical target regions on tri-objective DTLZ1.

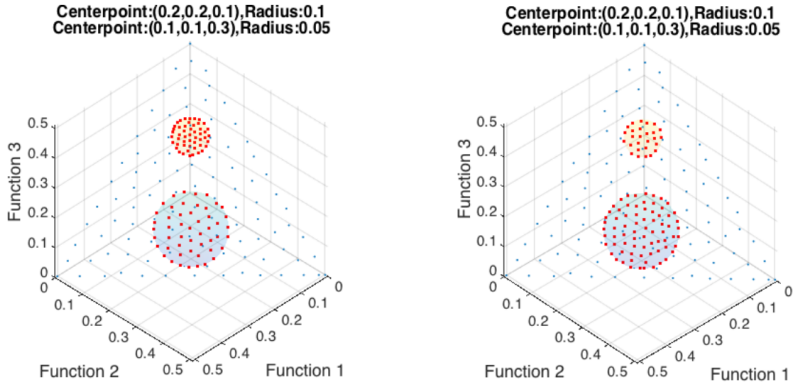



**Figure 4.14:** Representative PF approximations of T-SMS-EMOA with different solution distribution in two spherical target regions on tri-objective DTLZ1.

ure 4.15 shows PF approximations for different target regions which don't intersect with the PF.

### Experiments on Single Target Point

The enhanced algorithms are not only capable of obtaining solutions in the target region, they also belong to reference point-based approaches. When the lower bound and the upper bound of the target region specified in the algorithms are the same, the target region shrinks to a target point. In this section, only results of T-SMS-EMOA are presented, T-R2-EMOA and T-NSGA-II can obtain similar results. Figure 4.16 shows PF approximations of T-SMS-EMOA for different single target points: the point around the PF, near the border, in the feasible area and in the infeasible area.

For three objective problem, the parameter $\epsilon$ plays an essential role in balancing convergence and diversity of the solutions near the target point. Figure 4.17 shows PF approximations of T-SMS-EMOA for one target point when the values of parameter $\epsilon$ are different. The black point is the target point and red points are obtained solutions; blue points indicate the entire PF. It is observed that when the parameter $\epsilon$ is smaller, obtained solutions are denser and more concentrated.

### Experiments on Multiple Target Points

The enhanced algorithms can also work on multiple target points. Increasing the number of evaluations to 20000, Figure 4.18 shows PF approximations of T-SMS-

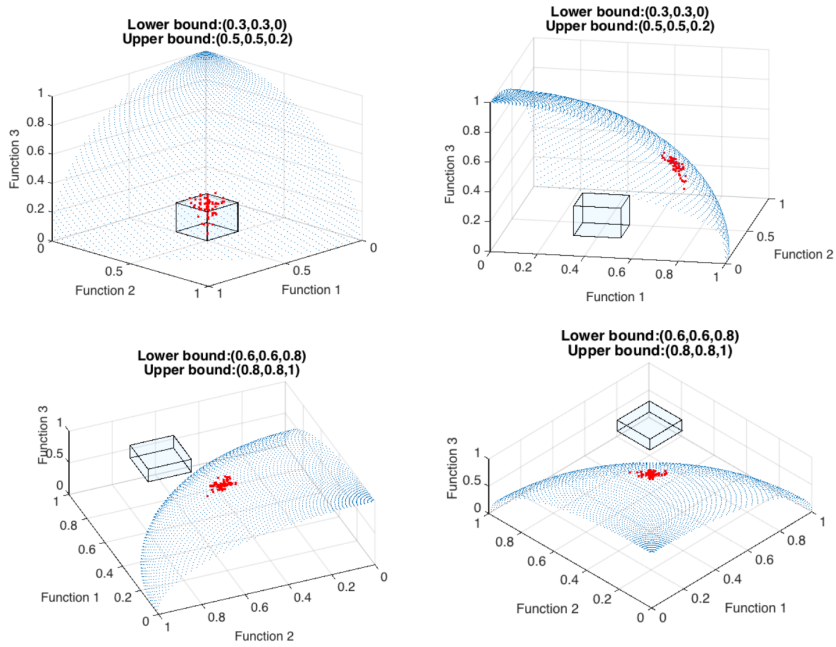



**Figure 4.15:** Representative PF approximations of T-SMS-EMOA on tri-objective DTLZ2; $\epsilon$=0.001.

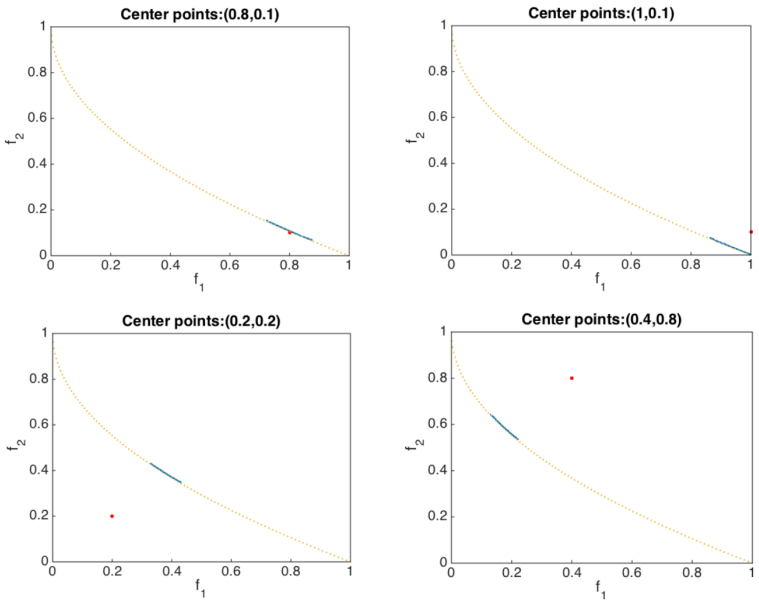



**Figure 4.16:** Representative PF approximations of T-SMS-EMOA on ZDT1; $\epsilon$=0.0001.

EMOA for two target points on ZDT1 problem when the values of parameter $\epsilon$ are different. The red points are the target points and blue points are obtained solutions; purple points indicate the entire PF.

Increasing the number of evaluations to 50000, Figure 4.19 shows PF approximations of T-SMS-EMOA for two target points on tri-objective DTLZ1 problem when the values of parameter $\epsilon$ are different. The black points are the target points and red points are obtained solutions; blue points indicate the entire PF.

### 4.1.4 Conclusion

In this part, a target region based multi-objective evolutionary approach has been proposed. Three algorithms named T-SMS-EMOA, T-R2-EMOA and T-NSGA-II have been instantiated when combining the proposed algorithm framework with original SMS-EMOA, R2-EMOA and NSGA-II. These new algorithms have been applied to a number of continuous and combinational benchmark problems with two or three objectives. Experimental results show that the proposed algorithms can guide the search toward the preferred region on the Pareto optimal front. Almost no outliers appear

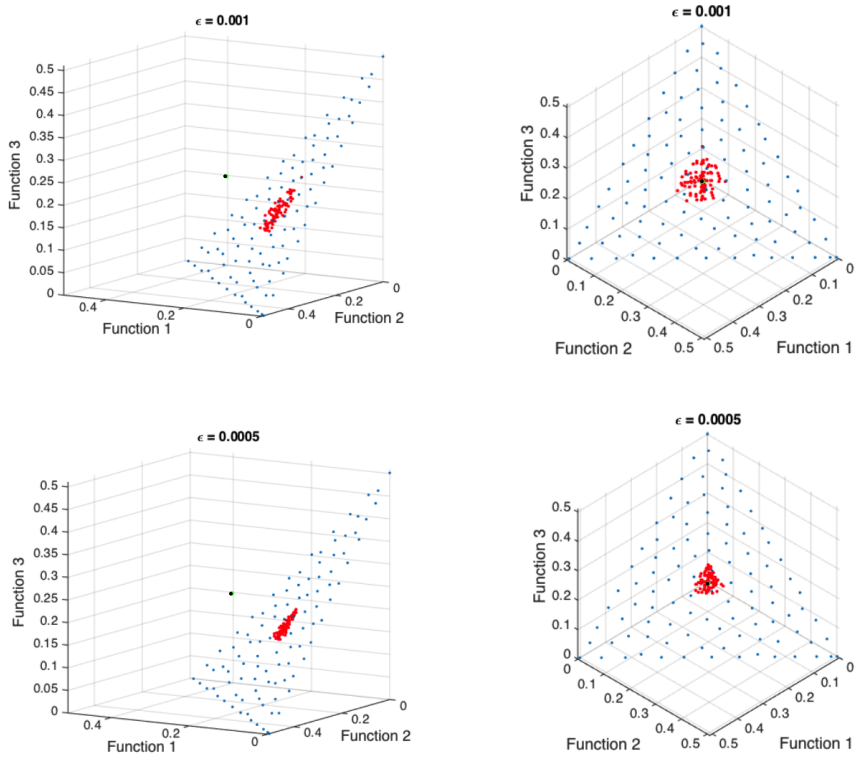



**Figure 4.17:** Representative PF approximations of T-SMS-EMOA on tri-objective DTLZ1 problem for one target point: (0.25, 0.25, 0.25).

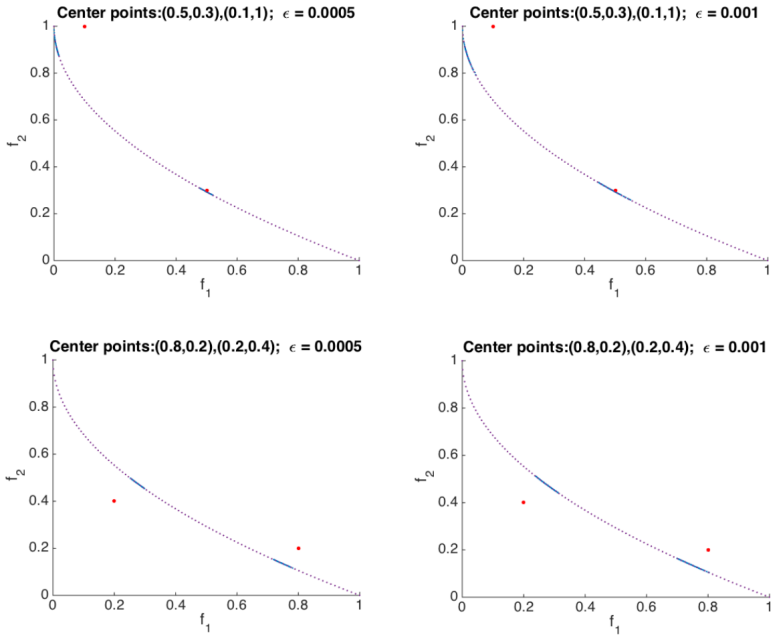



**Figure 4.18:** Representative PF approximations of T-SMS-EMOA on ZDT1 for two target points.

outside of the target region. In addition, these basic algorithms have been improved. The enhanced algorithms are more powerful and do not only support multiple target regions but also target point(s). It is worth noting that different numbers of solutions can be allocated to different targets by assigning the proportion of population size for each target.

On several instances, the proposed algorithms presented similar performance to the original MOEAs when converting the target region into constraints in the problem description. However, the proposed algorithms save a large amount of computational effort by guiding the search towards the preferred region without the calculation of the second ranking criterion in initial iterations. On the contrary, for original MOEAs, the increase in the number of constraints leads to the decrease of the search ability. Moreover, compared to the original MOEAs, the proposed algorithms exhibit the trend of behaving better with the increase in the number of objectives. More importantly, when there is no intersection between targets and the PF, the proposed algorithms can still find Pareto optimal solutions close to the targets. The future work would be

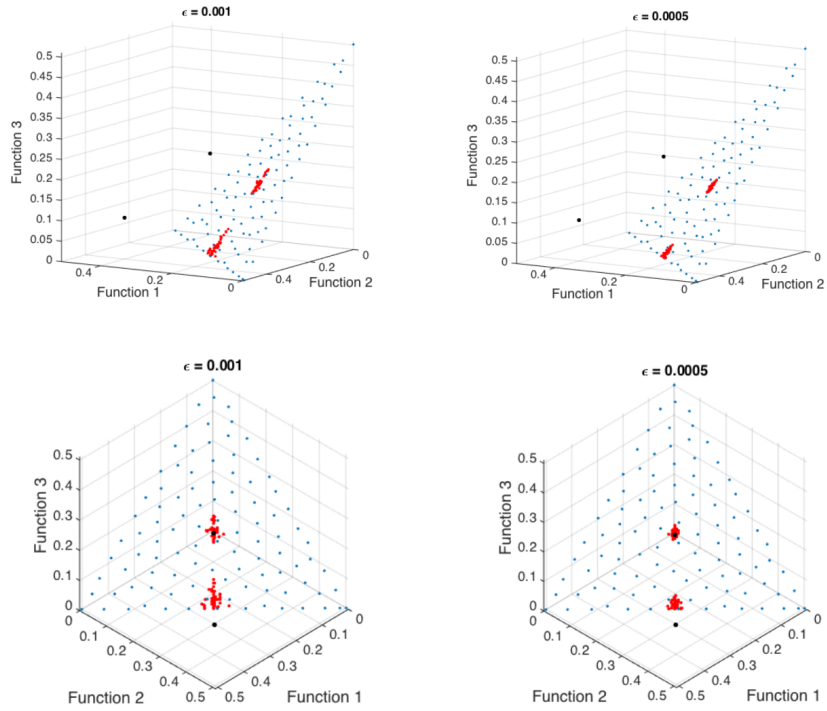



**Figure 4.19:** Representative PF approximations of T-SMS-EMOA on tri-objective DTLZ1 problem for two target points: (0.25, 0.25, 0.25) (0.4, 0.4, 0.1).

to compare the proposed algorithms with other preference-based MOEAs, especially multiple preferences based algorithms.

## 4.2 Automatic Preference Based MOEAs

When the algorithm aims at converging to the preferred solutions of the DMs, the DMs are asked for preference information. However, inspecting and choosing solutions from a large amount of solutions of a multi-objective optimization problem is not a trivial task for the DMs. The visualization of high-dimensional space further aggravates the difficulty. Sometimes, the DMs have no domain knowledge about the problems, they might set unreasonable goals which may mislead the search process. Therefore, an automatic preference based MOEA is proposed to avoid these difficulties and generate solutions in an automatically detected knee region. It is developed based on the

framework of DI-MOEA (see Section 3.1) and named automatic preference based DI-MOEA (AP-DI-MOEA).

AP-DI-MOEA can generate the preference region automatically, narrow down the feasible objective space progressively, and eventually obtain the preferred solutions. The preference region in AP-DI-MOEA is determined by the knee point. The knee point is a point for which a small improvement in any objective would lead to a large deterioration in at least one other objective. Several features of AP-DI-MOEA include: (1) no prior knowledge is used in identifying the knee point and knee region; (2) the preference region is generated automatically and narrowed down step by step to benefit its accuracy; (3) the proposed strategy can handle both multi-objective and many-objective optimization problems; (4) although AP-DI-MOEA is proposed based on DI-MOEA, the proposed strategy can be integrated with any standard MOEAs to form automatic preference based MOEAs; (5) the proposed algorithm is capable of finding preferred solutions for multi-objective optimization problems with linear, convex, concave Pareto fronts and discrete problems.

The remainder of this section is organized as follows. A literature review on knee based optimization is provided in Section 4.2.1. In Section 4.2.2, the proposed algorithm is described in detail. The experimental results are reported in Section 4.2.3 and Section 4.2.4 concludes the work with the summary and outlook.

### 4.2.1 Literature Review

In AP-DI-MOEA, the search for solutions is only guided towards the preference region which is determined by the knee point. It has been argued in the literature that knee points are most interesting solutions, naturally preferred solutions and most likely the optimal choice of the decision maker [12, 20, 23, 76].

In the last decade, several methods have been presented to identify knee points or knee regions. Das [20] refers the point where the Pareto surface "bulges" the most as the knee point, and this point corresponds to the farthest solution from the convex hull of individual minima which is the minima of the single objective functions. Zitzler [140] defines $\epsilon$-dominance: a solution $a$ is said to $\epsilon$-dominate a solution $b$ if and only if $f_i(a) + \epsilon \geq f_i(b)\ \forall i = 1, ..., m$ where $m$ is the number of objectives. A solution with a higher $\epsilon$-dominance value with respect to the other solutions in the Pareto front approximation, is a solution having higher trade-offs and in this definition corresponds to a knee point. The authors of [128] propose to calculate the density of solutions projected onto the hyperplane constructed by the extreme points of the non-

dominated solutions, then identify the knee regions based on the solution density.

Different algorithms of applying knee points in MOEAs have also been proposed. Branke [12] modifies the second criterion in NSGA-II, and replaces the crowding distance by either an angle-based measure or a utility-based measure. The angle-based method calculates the angle between an individual and its two neighbors in the objective space. The smaller the angle, the more clearly the individual can be classified as a knee point. However, this method can only be used for two objective problems. In the utility-based method, a marginal utility function is suggested to approximate the angle-based measure in the case of more than two objectives. The larger the external angle between a solution and its neighbors, the larger the gain in terms of linear utility obtained from substituting the neighbors with the solution of interest. However, the utility-based measure is not suited for finding knees in concave regions of the Pareto front.

Rachmawati [85, 86] proposes a knee-based MOEA which computes a transformation of original objective values based on a weighted sum niching approach. The extent and density of coverage of the knee regions are controllable by the parameters for the niche strength and pool size. The strategy is susceptible to the loss of less pronounced knee regions.

Schütze [93] investigates two strategies for the approximation of knees of bi-objective optimization problems with stochastic search algorithms. Several new definitions for identifying knee points and knee regions for bi-objective optimization problems has been suggested in [25] and the possibility of applying them has also been discussed.

Besides the knee points, the reference points, which are normally provided by the DM, have also been used to find a set of solutions near reference points. Deb [30] proposes an MOEA, called R-NSGA-II, by which a set of Pareto optimal solutions near a supplied set of reference points can be found. The dominance relation together with a modified crowding distance operator is used in this methodology. For all solutions of the population, the distances to all reference points are calculated and ranked. The lowest rank (over all reference points) of a solution is used as its crowding distance. Besides, a parameter $\epsilon$ is used to control the spread of obtained solutions. Recently, R-NSGA-II was extended and the reference point based NSGA-III (R-NSGA-III) is proposed for solving higher objective problems [111]. Bechikh proposes KR-NSGA-II [7] by extending R-NSGA-II. Instead of obtaining the reference points from the DM, in KR-NSGA-II, the knee points are used as mobile reference points and the search of the algorithm was guided towards these points. The number of knee points of the optimization problem is needed as prior information in KR-NSGA-II.

Gaudrie [49] uses the projection (intersection in case of a continuous front) of the closest non-dominated point on the line connecting the estimated ideal and nadir points as default preference. Conditional Gaussian process simulations are performed to create possible Pareto fronts, each of which defines a sample for the ideal and the nadir point, and the estimated ideal and nadir are the medians of the samples.

Rachmawati and Srinivasan [87] evaluate the worthiness of each non-dominated solution in terms of compromise between the objectives. The local maxima is then identified as potential knee solutions and the linear weighted-sums of the original objective functions are optimized to guide solutions toward the knee regions.

Another idea of incorporating preference information into multi-objective optimization is proposed in [103]. They combine the fitness function and an achievement scalarizing function containing the reference point. In this approach, the preference information is given in the form of a reference point and an indicator-based evolutionary algorithm IBEA [139] is modified by embedding the preference information into the indicator. Various further preference based MOEAs have been suggested, e.g., [13, 88, 117].

In our proposed algorithm, i.e., AP-DI-MOEA, we adopt the method from [20] to identify the knee point, design the preference region based on the knee point, and guide the search towards the preference region.

### 4.2.2 Algorithms

Two variants of DI-MOEA, DI-1 and DI-2 (see Section 3.1), exist. Analogously, two variants of AP-DI-MOEA, i.e., AP-DI-1 and AP-DI-2, are derived from the two variants of DI-MOEA. The workings of AP-DI-MOEA are outlined in Algorithm 9. In the algorithm, the variable *evals_update* is used to record the condition of generating or updating the preference region. Its initial value is assigned to *divide_size* (line 4 in Algorithm 9). Exceedance of *divide_size* is a predefined condition to divide the algorithm into two phases: learning phase and decision phase. In the learning phase, the algorithm explores the possible area of Pareto optimal solutions and finds the rough approximations of the Pareto front. In the decision phase, the algorithm identifies the preference region and finds preferred solutions. When the algorithm starts running and the number of evaluations reaches or exceeds *divide_size* at some moment, the first preference region will be generated and *evals_update* will be updated for determining a new future moment when the preference region needs to be updated (line 12 - 16 in Algorithm 9). The process of updating *evals_update* repeats until the end

to narrow down the preference region step by step. The first value of *evals_update*, i.e., *divide_size*, is a boundary line. Before it is satisfied, AP-DI-MOEA runs exactly like DI-MOEA to approximate the whole Pareto front; while, after it is satisfied, the preference region is generated automatically and AP-DI-MOEA finds solutions focusing on the preference region. The subsequent values of *evals_update* define the later moments to update the preference region; eventually, a precise ROI with a proper size can be achieved.

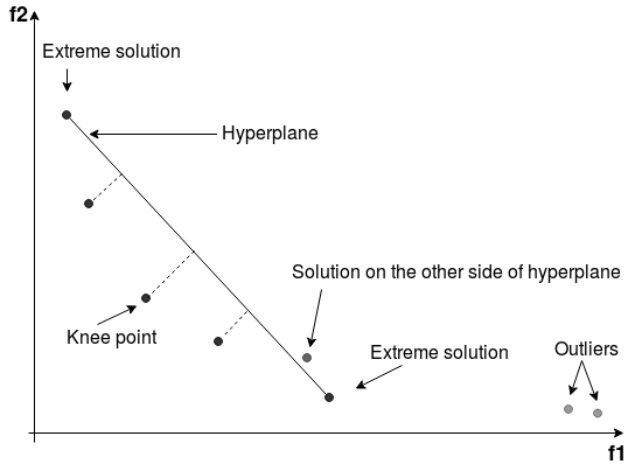



**Figure 4.20:** Finding the knee point in bi-dimensional space.

The first/new preference region is formed based on the population at the moment when the condition of *evals_update* is satisfied. To be specific, the preference region is determined by the knee point of the current Pareto front. Algorithm 10 gives the details of line 14 in Algorithm 9, it introduces the steps of finding the knee point of a non-dominated solution set and constituting a hypercube shaped preference region according to the knee point. Figure 4.20 also gives an illustration of finding the knee point in bi-dimensional space. Firstly, the upper quartile objective values (line 12 in Algorithm 10) in the solution set are used as a boundary to define outliers. To identify the knee point, solutions outside this boundary are removed from the solution set (line 15 - 19 in Algorithm 10). The extreme solutions (the solutions with the maximum value in one objective) are then found inside the boundary (line 22 in Algorithm 10) and a hyperplane is formed based on the extreme solutions (line 23 in Algorithm 10). In a bi-dimensional space (Figure 4.20), the hyperplane is only a line connecting two extreme solutions. According to the numbers of points below and above the hyperplane

**Algorithm 9** AP-DI-MOEA-2

**Inputs:**
*popsize*; // population size
*divide_size*; // number of evaluations before generating 1st preference region
*batch_size*; // number of evaluations between preference region updates

$P_0 \leftarrow init(popsize)$; // initialize random population
*existRegion* $\leftarrow$ false; // indicates whether a preference region was already computed
*evals* $\leftarrow 0$; // number of evaluations to far
*evals_update* $\leftarrow$ *divide_size*; // number of evaluations when 1st preference region is computed
$(R_1, ..., R_{\ell_0}) \leftarrow$ non_dominated_sorting($P_0$); // partition into fronts of increasing dominance ranks
**for each** $i \in \{1, \ldots, \ell_0\}$ **do**
  calculate diversity indicator for all solutions on $R_i$;
**end for**
$t \leftarrow 0$;
**while** Stop criterion not satisfied() **do**
  // update / computation of preference region
  **if** (*evals* > *evals_update* && $\ell_t == 1$) **then**
    *existRegion* $\leftarrow$ true;
    calculate *P_region*; //generate a (new) preference region, i.e., Algorithm 2
    *evals_update* $\leftarrow$ *evals_update* + *batch_size*;
  **end if**

  // offspring generation
  **if** ($\ell_t > 1$ || $t == 0$) **then**
    //($\mu + \mu$) generational scheme
    $Q_t \leftarrow Gen(P_t)$; // generate *popsize* offspring by recombination and mutation
    evaluate($Q_t$);
    *evals* $\leftarrow$ *evals* + *popsize*;
    $M_{t+1} = P_t \cup Q_t$; // combine offspring and parent population
  **else**
    // ($\mu + 1$) steady state generational scheme
    $q \leftarrow Gen(P_t)$; // generate only one offspring by recombination and mutation
    evaluate($q_t$);
    *evals* $\leftarrow$ *evals* + 1;
    $M_{t+1} = P_t \cup \{q_t\}$; // combine offspring and parent population
  **end if**

  // construction of new population based on non-dominated sorting
  $(R_1, ..., R_{\ell_{t+1}}) \leftarrow$ non_dominated_sorting($M_{t+1}$);
  $P_{t+1} \leftarrow \emptyset$;
  $i \leftarrow 0$;
  **while** $|P_{t+1}| <$ *popsize* **do**
    $i \leftarrow i + 1$;
    $P_{t+1} \leftarrow P_{t+1} \cup R_i$;
  **end while**

  // truncation of new population based on further ranking criterion(s)
  **if** ($|P_{t+1}| >$ *popsize*) **then**
    **if** (*existRegion* == false) **then**
      rank solutions in $R_i$ by diversity indicator contribution;
    **else**
      rank solutions in $R_i$ by diversity indicator contribution and Euclidean distance to knee point;
      assign the lowest possible rank to all solutions in $R_i$, which are outside *P_region*;
    **end if**
    $n \leftarrow |P_{t+1}| -$ *popsize*;
    remove the $n$ solutions in $R_i$ with lowest ranks from $P_{t+1}$;
  **end if**

  $t \leftarrow t + 1$;
  $l_t \leftarrow$the number of fronts of $P_t$;
**end while**

**Algorithm 10** Finding the knee point and defining the preference region.

**Inputs:**
$popsize$; // population size
$n$; // number of objectives
$P_t$; // current population
$\epsilon$; // parameter (>0) for distinguishing convex/concave shape

declare($Q[n]$); //upper quartile objective values of $P_t$
declare($L[n]$); //worst objective values of $P_t$
declare($knee[n]$); //knee point of $P_t$
declare($P\_region[n]$); //preference region of $P_t$
declare($Epoints[n][n]$); //extreme points (single-objectives)
$foundknee \leftarrow false$; // indicates whether the knee point was already found
$P'_t \leftarrow P_t$; //copy the current population for finding the knee

// remove outliers with lowest 25% of objective values
**for each** $i \in \{1, \ldots, n\}$ **do**
  sort($P'_t$) by the $i$th objective in ascending order;
  $Q[i] \leftarrow P'_t$.get_index($\frac{3}{4} \times popsize$).get_obj($i$); //upper quartile value of the $i$th objective
  $L[i] \leftarrow P'_t$.get_index($popsize$).get_obj($i$); //the largest (worst) value of the $i$th objective
**end for**
**for all** solution $s \in P'_t$ **do**
  **if** $s$.get_obj($i = 1, ..., n$) $> Q[i]$ **then**
    remove $s$ from $P'_t$; //remove outliers
  **end if**
**end for**

//find knee point by computing distance to hyperplane
$Epoints[\bullet][\bullet] \leftarrow$ extreme points in $P'_t$;
hyperplane($Epoints[\bullet][\bullet]$); //generate hyperplane by $Epoints[\bullet][\bullet]$
$num_a \leftarrow$ number of points in concave region of hyperplane;
$num_v \leftarrow |P'_t| - num_a$; // number of points in convex region
**if** $(num_v - num_a > \epsilon)$ **then**
  //roughly convex shape
  remove solutions in concave region from $P'_t$;
**else if** $(num_a - num_v > \epsilon)$ **then**
  //roughly concave shape
  remove solutions in convex region from $P'_t$;
**else**
  //roughly linear shape
  //find knee point by computing hypervolume
  **for all** solution $s \in P'_t$ **do**
    calculate hypervolume of $s$ with reference point $L[\bullet]$;
  **end for**
  $knee[\bullet] \leftarrow$ solution with the largest hypervolume value;
  $foundknee \leftarrow true$;
**end if**
**if** $(foundknee == false)$ **then**
  **for all** solution $s \in P'_t$ **do**
    calculate distance between $s$ and hyperplane;
  **end for**
  $knee[\bullet] \leftarrow$ solution with the largest distance;
**end if**

//determine current preference region by knee point.
**for each** $i \in \{1, \ldots, n\}$ **do**
  $P\_region[i] \leftarrow knee[i] + (L[i] - knee[i]) \times 85\%$
**end for**

(line 24 - 25 in Algorithm 10), the shape of the solution set can be roughly perceived. We will distinguish between "convex" and "concave" regions. Points in the *convex* (*concave*) *region* are dominating (dominated by) at least one point in the hyperplane spanned by the extreme points. However, when the number of the points in the convex region and the number of points in the concave region is close enough, it implies that the shape of the current solution set is almost linear. This occurs both when the true Pareto front is linear and when the solution set converges very well in a small area of the Pareto front. A parameter $\epsilon$ then is used to represent the closeness and it is a small number decided by the size of the solution set. In the case that the shape of the current solution set is (almost) linear, the solution with the largest hypervolume value with regards to the worst objective vector (i.e., $L[i]$ in line 13 in Algorithm 10) is adopted as the knee point (line 33 - 39 in Algorithm 10). While, under the condition that the shape of the current solution set is convex or concave, the knee point is identified by the method in [20]. The solution in the convex or concave region with the largest Euclidean distance to the hyperplane is chosen as the knee point (line 42 - 45 in Algorithm 10). After the knee point is found, the preference region can be determined based on the knee point by the following formula:

$$P\_region[i] = knee[i] + (L[i] - knee[i]) \times 85\%. \tag{4.5}$$

Let $i$ denotes the $i$th objective, as in Algorithm 10, $L[i]$ is the worst value of the $i$th objective in the population, $knee[i]$ is the $i$th objective value of the knee point and $P\_region[i]$ is the upper bound of the preference region. W.l.o.g. We assume the objectives are to be minimized and the lower bound of the preference region is the origin point. According to the formula, we can see that the first preference region is relatively large (roughly 85% of the entire Pareto front). With the increase in the number of iterations, the preference region will be updated and become smaller and smaller because every preference region picks 85% of the current Pareto front. Eventually, we want the preference region to reach a proper range, say, 15% of the initial Pareto front. The process of narrowing down the preference region step by step can benefit the accuracy of the preference region.

Algorithm 9 only shows the workings of AP-DI-2. The difference between AP-DI-1 and AP-DI-2 is the same as the difference between DI-1 and DI-2, i.e., the diversity criterion in the $(\mu + \mu)$ generational selection operator in AP-DI-1 is the crowding distance, in AP-DI-2 it is the diversity indicator. In the algorithm, the initialized population is sorted based on non-domination and the diversity value of each solution

is calculated to be used in later parent selection (line 5 - 8 in Algorithm 9). When evolution of the population takes place, according to different phases of optimization, the $(\mu + \mu)$ generational selection operator generates multiple offspring in one iteration to explore more decision space and push the population quickly towards the Pareto front (line 20 - 24 in Algorithm 9); the $(\mu + 1)$ steady state selection operator generates only one offspring in order to achieve a uniformly distributed set (line 26 - 30 in Algorithm 9). To achieve the next generation population, the combination of parents and offspring is classified into different layers according to the non-dominance relation. The points from the first non-domination front are preserved, continuing with points in the second non-domination front, until the number of points reaches the population size (line 34 -40 in Algorithm 9). Under the case that the number of points surpasses the population size, a truncation selection is carried out (line 43 - 52 in Algorithm 9). When there is no preference region, the population will be truncated based on the diversity indicator. While, if a preference region already exists, the population will be truncated based on first the diversity indicator, then Euclidean distance to the knee point. In this process, the diversity indicator contribution and Euclidean distance to the knee point are calculated only for the solutions in the preference region. Solutions outside of the region are given relatively larger values and they will be eliminated in the following optimization process.

There are different strategies to set and update the value of *evals_update*. In our algorithm, we divide the whole computing budget into two parts, the first half is used to find an initial entire Pareto front approximation, and the second half is used to update the preference region and find solutions in the preference region. Assume the total computing budget is *budget_size* (the number of evaluations), then the first value of *evals_update* is $\frac{1}{2} \times budget\_size$. Due to the reason that we expect a final preference region with a size of around 15% of the initial entire Pareto front and each new preference region takes 85% of the current Pareto front, according to the formula: $0.85^{12} \approx 0.14$, the value of *evals_update* can be updated by the following formula:

$$evals\_update = evals\_update + batch\_size = evals\_update + (budget\_size/2)/12. \tag{4.6}$$

Another half of the budget can be divided into 12 partial-budgets and a new preference region is constituted after each partial-budget. In the end, the final preference region is achieved and solutions focusing on this preference region are obtained.

### 4.2.3 Experimental Results

#### Experimental Design

For the two variants of AP-DI-MOEA: AP-DI-1 and AP-DI-2, their performances have been compared with DI-MOEA: DI-1, DI-2 and NSGA-III. NSGA-III is involved in the comparison because NSGA-III is a representative state-of-the-art evolutionary multi-objective algorithm and it is very powerful to handle problems with non-linear characteristics. For bi-objective benchmark problems, algorithms are tested on ZDT1 and ZDT2 with 30 variables. For tri-objective benchmark problems, DTLZ1 with 7 variables and DTLZ2 with 12 variables are tested. On every problem, each algorithm runs for 30 times with different seeds, while the same 30 different seeds are used for all algorithms. All the experiments are performed with a population size of 100. For bi-objective problems, experiments are run with a budget of 22000 (objective function) evaluations; for DTLZ tri-objective problems, the budget is 120000 evaluations.

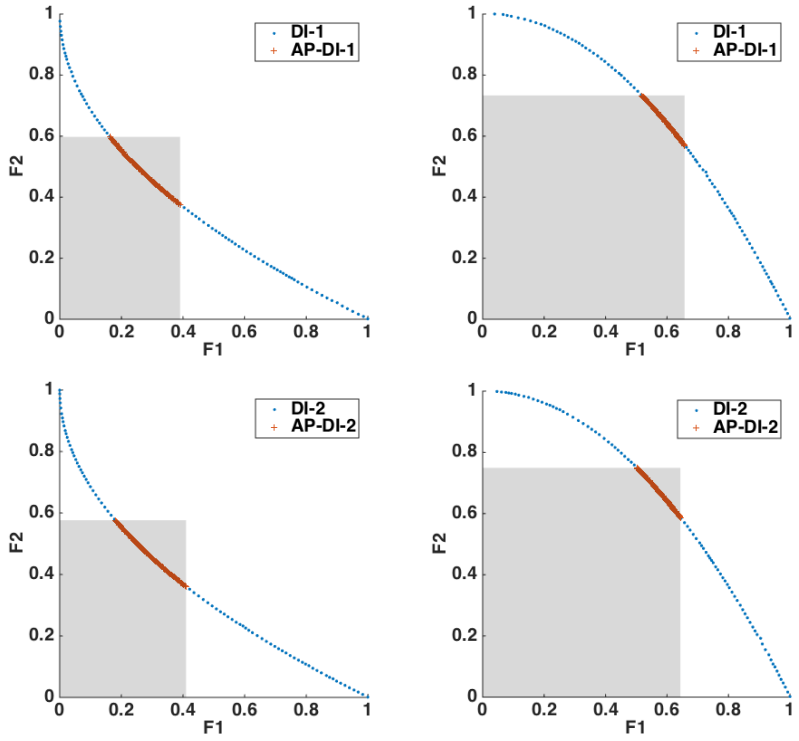
#### Experiments on Bi-objective Problems

Bi-objective problems are optimized with a total budget of 22000 evaluations, when the number of evaluations reaches 10000, the first preference region is generated, then after every 1200 evaluations, the preference region will be updated. Figure 4.21 shows Pareto front approximations from a typical run on ZDT1 (left column) and ZDT2 (right column). The graphs on the upper row are obtained from DI-1 and AP-DI-1, while the graphs on the lower row are from DI-2 and AP-DI-2. In each graph, the entire Pareto front approximations from DI-MOEA and the preferred solutions from AP-DI-MOEA (or *AP solutions*) are presented, at the same time, the final preference regions of AP-DI-MOEA are also shown by the gray areas.

Besides the visualization of the Pareto fronts, the knee point of the entire final Pareto front approximation from DI-MOEA is also computed via the strategy described in Algorithm 10. For each run of DI-MOEA and AP-DI-MOEA with the same seed, the following two issues have been checked:

- if the knee point from DI-MOEA is in the preference region achieved by its derived AP-DI-MOEA;
- if the knee point from DI-MOEA is dominated by or dominating AP solutions; or if it is a non-dominated solution (mutually non-dominated with all AP solutions).

Table 4.7 shows the results of 30 runs. For ZDT1 problem, all 30 knee points from DI-1 and DI-2 are in the preference regions from AP-DI-1 and AP-DI-2 respectively; in




**(a)** ZDT1. **(b)** ZDT2.

**Figure 4.21:** Pareto front approximations on ZDT1 and ZDT2.

all these knee points, 10 from DI-1 and 7 from DI-2 are dominated by AP solutions. For ZDT2 problem, most knee points are not in the corresponding preference regions, but for those in the preference regions, almost all of them are dominated by AP solutions. Please note that when a knee point from DI-MOEA is outside of the preference region from AP-DI-MOEA, it is not possible that it can dominate any AP solutions because all AP solutions are in the preference region and only solutions in the left side of the gray area can dominate AP solutions.

The same comparison is also performed between AP-DI-MOEA and NSGA-III, the results are shown in Table 4.8. For ZDT1 problem, all knee points from NSGA-III are in the preference regions from AP-DI-MOEA. Some of these knee points dominate AP solutions. For ZDT2 problem, most knee points from NSGA-III are not in the preference regions and these knee points are incomparable with AP solutions. For the knee points in the preference regions, all three dominating relations with AP solutions

**Table 4.7:** Space and dominance relation of knee point from DI-MOEA and AP solutions on ZDT problems.

| Problem | | ZDT1 | | ZDT2 | |
|---|---|---|---|---|---|
| Algorithm | | DI-1/ AP-DI-1 | DI-2/ AP-DI-2 | DI-1/ AP-DI-1 | DI-2/ AP-DI-2 |
| In preference region | Incomparable | 20 | 23 | 1 | 1 |
| | Dominated | 10 | 7 | 9 | 9 |
| | Dominating | 0 | 0 | 0 | 0 |
| Outside p-region | Incomparable | 0 | 0 | 20 | 20 |
| | Dominated | 0 | 0 | 0 | 0 |

appear. For both problems, when the knee point from NSGA-III is dominating AP solutions, it only dominates one AP solution.

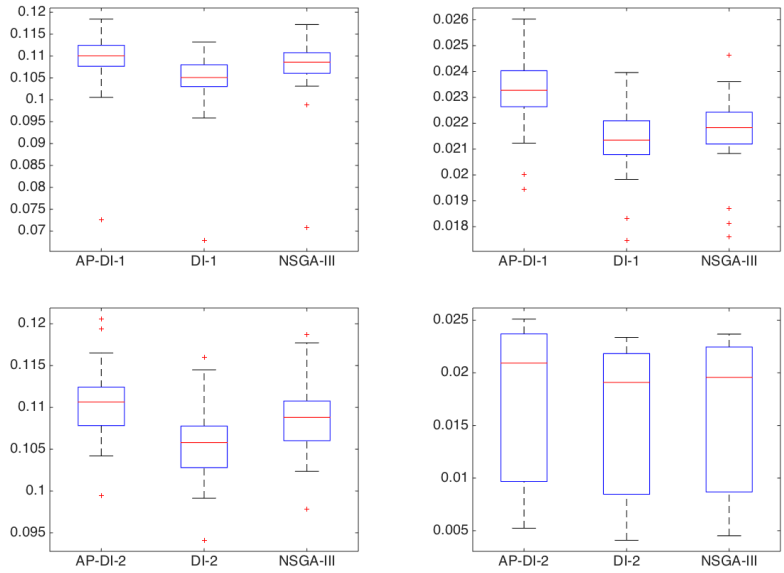
**Table 4.8:** Space and dominance relation of knee point from NSGA-III and AP solutions on ZDT problems.

| Problem | | ZDT1 | | ZDT2 | |
|---|---|---|---|---|---|
| Algorithm | | NSGA-III/ AP-DI-1 | NSGA-III/ AP-DI-2 | NSGA-III/ AP-DI-1 | NSGA-III/ AP-DI-2 |
| In preference region | Incomparable | 14 | 19 | 3 | 1 |
| | Dominated | 0 | 0 | 2 | 3 |
| | Dominating | 16 | 11 | 4 | 6 |
| Outside p-region | Incomparable | 0 | 0 | 21 | 20 |
| | Dominated | 0 | 0 | 0 | 0 |

Instead of spreading the population across the entire Pareto front, the optimization only focuses on the preference region. To ensure that the algorithm can guide the search towards the preference region and the achieved solution set is distributed across the preference region, the performance of AP-DI-MOEA, DI-MOEA and NSGA-III is compared in the preference region. For each Pareto front approximation from DI-MOEA and NSGA-III, the solutions in the corresponding preference region from AP-DI-MOEA are picked, and these solutions are compared with AP solutions through the hypervolume indicator. The point formed by the largest objective values over all solutions in the preference region is adopted as the reference point when calculating the hypervolume indicator. It has been found that all hypervolume values of new solution sets from DI-MOEA and NSGA-III in the preference region are worse than the hypervolume values of the solution sets from AP-DI-MOEA, which proves that the mechanism indeed works in practice. Figure 4.22 shows box plots of the distribution of hypervolume indicators over 30 runs.

It can be seen that on both ZDT1 (with a convex Pareto front) and ZDT2 (with a concave Pareto front), AP-DI-MOEA can find solutions in the preference region which is determined by the knee points. Here for ZDT2 problem, its knee region is still a

compromise of two objectives. If the Pareto front is severely concave, adopting the knee region as the preference region is probably not a good decision because solutions in the knee region have poor values on both objectives, and it is not clear which region is preferred by the decision maker. But for the common cases, especially most real-world application problems, they don't have a severely concave Pareto front, the knee region is still recommended as the preference region.



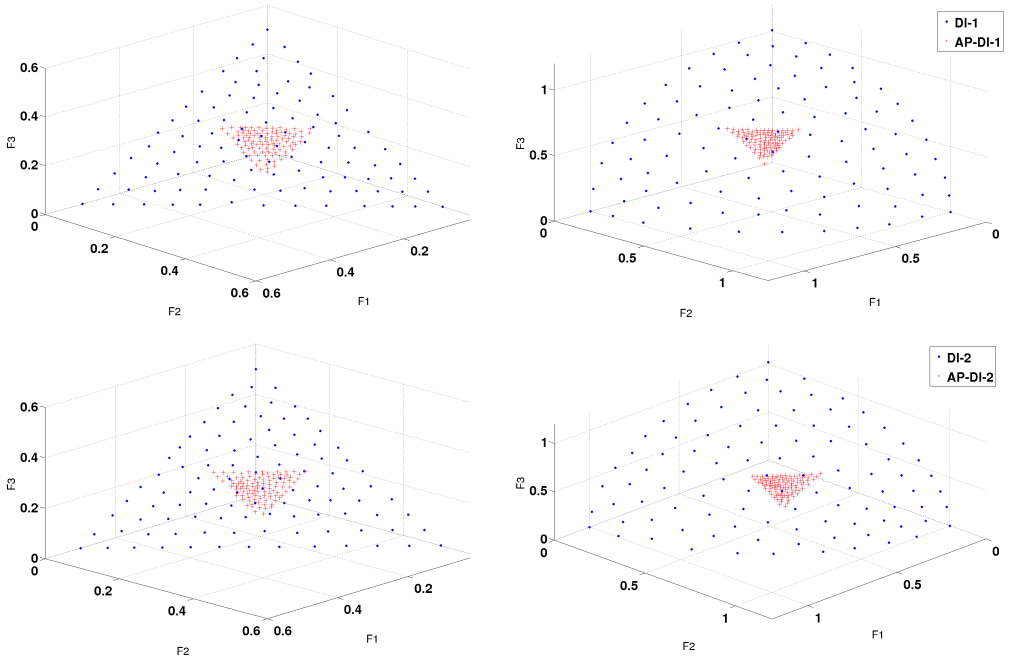

**(a)** ZDT1. **(b)** ZDT2.

**Figure 4.22:** Boxplots comparing the hypervolume values on ZDT1 and ZDT2.

### Experiments on Tri-objective Problems

DTLZ1 and DTLZ2 are chosen as tri-objective benchmark problems to investigate the algorithms. They are performed with a total budget of 120000 fitness evaluations, when the number of evaluations reaches 60000, the first preference region is formed, then after every 5000 evaluations, the preference region is updated. Figure 4.23 shows the Pareto front approximations from a typical run on DTLZ1 (left column) and DTLZ2 (right column). The upper graphs are obtained from DI-1 and AP-DI-1, while the lower graphs are from DI-2 and AP-DI-2. In each graph, the Pareto front approximations from DI-MOEA and corresponding AP-DI-MOEA are given. Since the target region is actually an axis aligned box, the obtained knee region (i.e., the intersection of the

axis aligned box with the Pareto front) has an inverted triangle shape for these two benchmark problems.




**(a)** DTLZ1 tri-objective problem. **(b)** DTLZ2 tri-objective problem.

**Figure 4.23:** Pareto front approximations on DTLZ1 and DTLZ2.

Table 4.9 shows the space and dominance relation of the knee point from DI-MOEA and the solution set from AP-DI-MOEA over 30 runs. For DTLZ1 problem, most knee points from DI-MOEA are in their respective preference regions and all knee points are mutually non-dominated with AP solutions. For DTLZ2 problem, it has been observed that more knee points are not in the corresponding preference regions. This is because too few solutions from DI-MOEA are in the preference region. For DTLZ1 problem, six solutions from DI-MOEA are in the corresponding preference region on average for each run, while, for DTLZ2 problem, only less than two solutions are in the corresponding preference region on average. Therefore, it can seen that on the one side, it is normal that many knee points from the entire Pareto fronts are not in their corresponding preference regions; on the other side, the aim of finding more fine-grained resolution in the preference region has been well achieved because only few solutions can be obtained in the preference region if the population is spread across

the entire Pareto front. At the same time, one knee point from DI-1 on DTLZ2 is dominated by solutions from the corresponding AP-DI-1, which proves that AP-DI-MOEA can converge better than DI-MOEA because AP-DI-MOEA focuses on the preference region.

**Table 4.9:** Space and dominance relation of knee point from DI-MOEA and AP solutions on DTLZ problems.

| Problem | | DTLZ1 | | DTLZ2 | |
|---|---|---|---|---|---|
| Algorithm | | DI-1/ AP-DI-1 | DI-2/ AP-DI-2 | DI-1/ AP-DI-1 | DI-2/ AP-DI-2 |
| In preference region | Incomparable | 29 | 27 | 10 | 13 |
| | Dominated | 0 | 0 | 1 | 0 |
| | Dominating | 0 | 0 | 0 | 0 |
| Outside p-region | Incomparable | 1 | 3 | 19 | 17 |
| | Dominated | 0 | 0 | 0 | 0 |

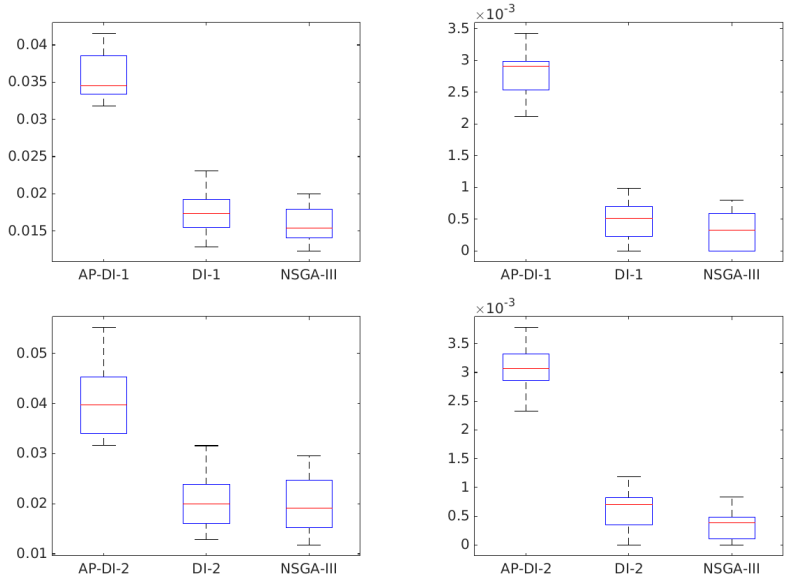
AP-DI-1 and AP-DI-2 have also been compared with NSGA-III in the same way. Table 4.10 shows the comparison result. For DTLZ1, the average number of solutions from NSGA-III in the corresponding preference regions from AP-DI-MOEA is six. Still, almost all knee solutions from NSGA-III are in the preference region. For DTLZ2, the average number of solutions from NSGA-III in the corresponding preference region from AP-DI-MOEA is less than one, while, in more than half of 30 runs, the knee points from NSGA-III are still in the preference region. To some extent, it can be concluded that the preference regions from AP-DI-MOEA are accurate. It can also be observed that AP-DI-1 behaves better than AP-DI-2 on DTLZ2, because two knee points from NSGA-III dominate the solutions from AP-DI-2.

**Table 4.10:** Space and dominance relation of knee point from NSGA-III and AP solutions on DTLZ problems.

| Problem | | DTLZ1 | | DTLZ2 | |
|---|---|---|---|---|---|
| Algorithm | | NSGA-III/ AP-DI-1 | NSGA-III/ AP-DI-2 | NSGA-III/ AP-DI-1 | NSGA-III/ AP-DI-2 |
| In preference region | Incomparable | 30 | 29 | 14 | 17 |
| | Dominated | 0 | 0 | 1 | 1 |
| | Dominating | 0 | 0 | 0 | 2 |
| Outside p-region | Incomparable | 0 | 1 | 15 | 10 |
| | Dominated | 0 | 0 | 0 | 0 |

Similarly, solutions which are in the corresponding preference region of AP-DI-MOEA are picked from DI-MOEA and NSGA-III, and the hypervolume indicator value is compared between these solutions and AP solutions. It has been found that all hypervolume values of solutions from AP-DI-MOEA are better than those of solutions from DI-MOEA and NSGA-III. The left column of Figure 4.24 shows box plots of

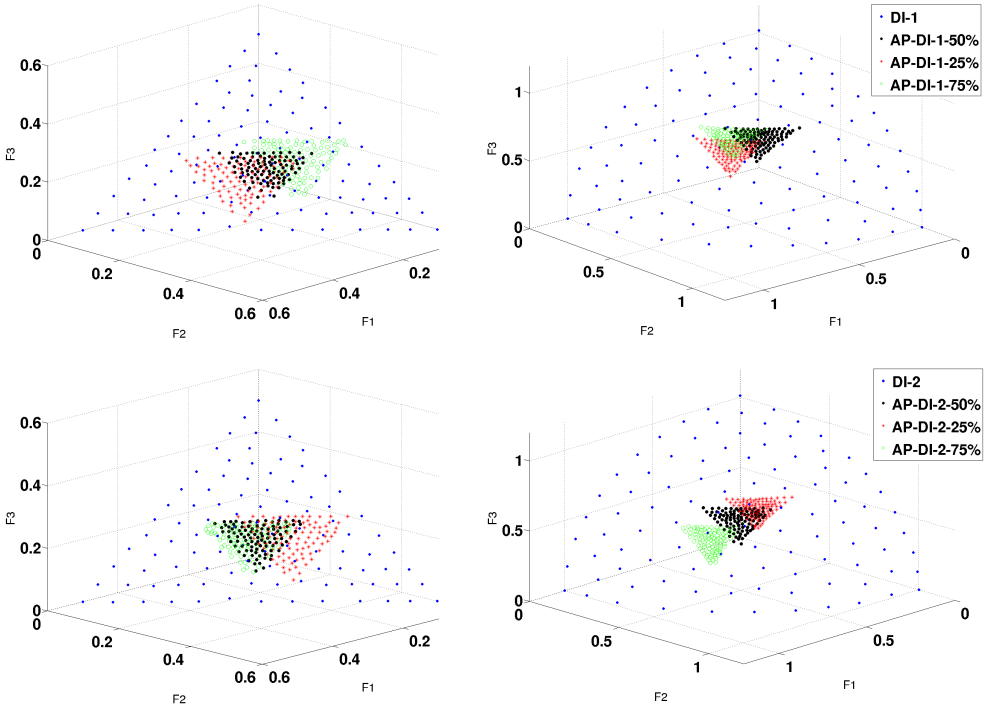
the distribution of hypervolume values over 30 runs on DTLZ1, and the right column shows the hypervolume comparison on DTLZ2.




**(a)** DTLZ1. **(b)** DTLZ2.

**Figure 4.24:** Boxplots comparing the hypervolume values on DTLZ1 and DTLZ2.

In the experiments, half of the total budget is used to find an initial Pareto front because it turned out to be a good compromise: half budget for the initial Pareto front and another half budget for the solutions focusing on the preference region. Experiments using 25% and 75% of the total budget for the initial Pareto front have also been conducted. Figure 4.25 presents the entire Pareto front from DI-MOEA and the Pareto front from AP-DI-MOEA with different budgets for the initial Pareto front. The left two images are on DTLZ1 and the right two images are on DTLZ2. The upper two images are from DI-1 and AP-DI-1; the lower two images are from DI-2 and AP-DI-2. In the legend labels, 50%, 25% and 75% indicate the budgets which are utilized to find the initial entire Pareto front. It can be observed that the preference region from AP-DI-MOEA with 50% of budget is located in a better position than with 25% and 75% of budget, and the position of the preference region from AP-DI-MOEA with 50% of budget is more stable. Therefore, in our algorithm, 50% of budget is used before the generation of the first preference region.




**(a)** DTLZ1 tri-objective problem. **(b)** DTLZ2 tri-objective problem.

**Figure 4.25:** Pareto front approximations by different budgets generating initial Pareto front.

### 4.2.4 Conclusion

A preference based multi-objective evolutionary algorithm, AP-DI-MOEA, is proposed. In the absence of explicitly provided preferences, the knee region is usually treated as the region of interest or the preference region. Given this, AP-DI-MOEA generates the knee region automatically and finds solutions with a more fine-grained resolution in the knee region. This has been demonstrated on the bi-objective ZDT1 and ZDT2 problems , and tri-objective DTLZ1 and DTLZ2 problems. In the benchmark, the proposed approach is also proven to perform better than NSGA-III which is included in the benchmark as a state-of-the-art reference algorithm.

It would be an interesting question how to adapt the algorithm to problems with multiple knee points and more irregular shapes. Besides, the proposed approach requires a definition of knee points. Future work will provide a more detailed comparison of different variants of methods to generate knee points.