

# Multi-Objective Optimisation with Multiple Preferred Regions

Md. Shahriar Mahbub<sup>1,2</sup>, Markus Wagner<sup>3</sup>, Luigi Crema<sup>1</sup>

<sup>1</sup>Fondazione Bruno Kessler, Via Sommarive 18, 38123 Povo, Trento, Italy

<sup>2</sup>University of Trento, Via Sommarive 9, 38123 Povo, Trento, Italy

<sup>3</sup>University of Adelaide, Adelaide, SA 5005, Australia.

**Abstract.** The typical goal in multi-objective optimization is to find a set of good and well-distributed solutions. It has become popular to focus on specific regions of the objective space, e.g., due to market demands or personal preferences.

In the past, a range of different approaches has been proposed to consider preferences for regions, including reference points and weights. While the former technique requires knowledge over the true set of trade-offs (and a notion of “closeness”) in order to perform well, it is not trivial to encode a non-standard preference for the latter.

With this article, we contribute to the set of algorithms that consider preferences. In particular, we propose the easy-to-use concept of “preferred regions” that can be used by laypeople, we explain algorithmic modifications of NSGAI and AGE, and we validate their effectiveness on benchmark problems and on a real-world problem.

**Keywords:** Multi-objective optimization; preference; evolutionary algorithm

## 1 Introduction

For the last two decades, multi-objective evolutionary algorithms (MOEA) have been successfully used to solve multi-objective optimization (MOO) problems. Most real-world problems are MOO problems rather than single objective problems, where it is often found that multiple conflicting objectives exist. Typically, the goal of a MOEA is to find a set of trade-off solutions that are well-distributed over the objective space. Ideally, these solutions are on or at least close to the true set of trade-offs called Pareto front. Sometimes, however, the decision makers/users have little interest in exploring the entire objective space. It may be more interesting to them to explore some preferred regions of a front due to market demands, due to financial pressure, or simply due to curiosity.

Let us consider an energy system optimization problem, where the goal is to minimize  $CO_2$  emission and annual cost [14]. In this case, a reference system is analyzed (generally the current system), optimized systems are identified, and then compared with the reference system. Decision makers are often interested in exploring several regions defined by either  $CO_2$  emission or annual cost. For example, if a reference scenario has  $x$  amount of  $CO_2$  emissions, the interesting regions could be 10%-20% and 35%-40% reduction of  $x$ .

There are a few algorithmic advantages in exploring preferred regions over exploring the entire objective space. These include faster convergence speed and better approximation of Pareto front [15]. Based on the idea of incorporating user preference, a wide range of different concepts and algorithms has been proposed (see [1, 25] for comprehensive surveys, as it is impossible to list all relevant work here), such as (i) defining reference-point(s) [18, 19] and specifying weights in the objective space [9, 16]. However, the problem with these approaches typically is that it is difficult to set the corresponding parameters without knowing the shape of the true Pareto front. Therefore, we propose modifications of generic algorithms that require only very intuitive preference encoding in the form of intervals, as outlined above in the energy system example above.

The structure of this article is as follows. First, we introduce basic definitions in Section 2 including the idea of preferences for MOEAs. In Section 3 we show how we integrate preference information in two algorithms. Lastly, we present and discuss the results of our experimental studies in Section 4 and Section 5.

## 2 Definitions & Basic Principles

Without loss of generality, a multi-objective problem can be formulated as:

$$\min F(A) = (f_1(A), f_2(A), \dots, f_m(A))^T \quad A \in \mathbb{R}^n \quad (1)$$

where  $A = (a_1, \dots, a_n)$  is a vector of  $n$  decision variables and  $m$  is the number of objectives. In the context of multi-objective optimisation, the optimal solutions are also referred to as non-dominated solutions. In a minimization problem, a solution  $A$  is considered non-dominated in comparison to another solution  $A^*$  when no objective value of  $A^*$  is less than that of  $A$  and at least one objective value of  $A^*$  is greater than that of  $A$ . If necessary, the feasibility of solutions can be considered as well, however, this is not necessary in our study. For a more complete introduction to multi-objective optimization, we refer the interested reader to the overview articles [4, 12, 24].

There is a fundamental difference between how reference points and weights are defined, and how our preferred regions are defined. A reference point is defined by specifying values for all the dimensions for a point in the objective space. In contrast to this, a preferred region is defined by specifying an upper and lower bound of one particular dimension. For example, if a two-objective problem (objectives are plotted along  $x$  and  $y$ -axes) is considered, a user can define three preferred regions by setting three upper and lower bounds for intervals along either the  $x$  or  $y$  axis. Figure 1 illustrates an example of three preferred regions (bounded by three different color vertical lines). In this figure, we also show three reference points (gray crosses) which might have been set by a user. As the user lacks knowledge about the shape of the front, these points are not on the true Pareto front. Consequently, it is left up to the MOEA to follow its own interpretation of “closeness” in order to distribute the solutions around the reference. One outcome is that the solution density is high near the reference point and the density decreases with increasing distance (see Figure 1 for the

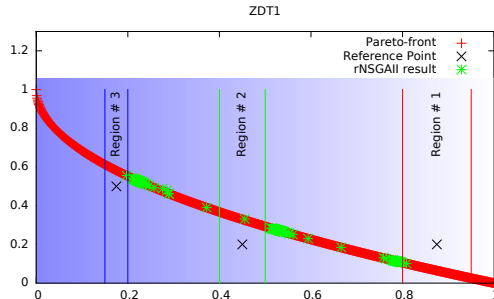


Fig. 1: Reference points, weights in the objective space, and preferred regions.

outcome of one run by r-NSGAI [7] using the shown reference points). As mentioned before, it is also possible to use weights in the objective space in order to encode preferences. In Figure 1 we indicate this using a color gradient along the  $x$ -axis, where a preference for smaller  $x$ -values is encoded. The preference formulation for a single objective using weights is relatively simple, however, the formulation becomes tricky when multiple objectives are preferred, and it becomes very complicated when reference points or preferred regions are to be encoded (using weights). The concept of our interval-based regions, on the other hand, is straightforward to use even for laypeople. To the best of our knowledge, even though there are lots of similar approaches, this is the first time this rather simple concept of intervals along axes is used in the context of MOO.

### 3 Preferred Regions for Different MOEAs

In the following, we present the ideas related to preferred regions and the adaptation of the ideas into different MOEAs.

#### 3.1 Ideas adopted in pNSGAI

We adopt several ideas in NSGAI [6] resulting in pNSGAI. Algorithm 1 presents the main loop of the proposed pNSGAI. There are couple of modifications with respect to the original NSGAI. Firstly, each solution is associated with a particular region (step # 4). Secondly, a parent selection procedure is used (step # 8); thirdly, the individuals of a merged population (containing solutions of the previous generation and the offspring) are associated with  $nR$  regions (step # 11) such that for each region,  $2 * \alpha_i$  individuals are associated. Based on the association, the merged population is divided into  $nR$  sub-populations (step # 12). Sub-populations are used to increase the likelihood of achieving the targeted  $\alpha_i$  well-distributed solutions per preferred region. Lastly, a modified ranking procedure (step # 14) is applied to rank the solutions of the sub-populations.<sup>1</sup>

<sup>1</sup> Consequently, our pNSGAI is somewhat equivalent to an island model approach for multi-objective optimization, with islands being responsible for preferred regions. In contrast to existing island model-MOO approaches (e.g. [2]), we are focussing on user-defined parts of the search space that are defined in an easy-to-use way.

---

**Algorithm 1** Main loop of pNSGAI
 

---

- 1:  $nR$  ▷ Number of given regions
  - 2:  $\alpha$  ▷ A set containing user-defined preferred number of solutions for each region;  
 $\alpha_i$  is the number of preferred solutions for  $i^{th}$  region
  - 3: Initialize population  $P$  with  $\sum_{i=1}^{nR} \alpha_i$  random individuals and  $O \leftarrow \emptyset$
  - 4: Associate  $\alpha$  number of solutions with  $nR$  regions
  - 5: **while** Stopping criteria not met **do**
  - 6:     **for**  $z \leftarrow 1$  to  $nR$  **do**
  - 7:         **for**  $j \leftarrow 1$  to  $\alpha_z/2$  **do**
  - 8:             Select two parents using modified parent selection procedure
  - 9:             Generate offspring and add to  $O$
  - 10:      $P \leftarrow P \cup O$
  - 11:     Associate  $2 * \alpha$  number of solutions with  $nR$  regions
  - 12:     Divide  $P$  into  $nR$  sub-populations ( $SP_i; i = 1, \dots, nR$ )
  - 13:     **for**  $i \leftarrow 1$  to  $nR$  **do**
  - 14:         Rank  $SP_i$  and select  $\alpha_i$  solutions based on ranking and crowding distance
  - 15:         Add these solutions to  $SP_i$
  - 16:      $P \leftarrow \bigcup_{i=1}^{nR} SP_i$
- 

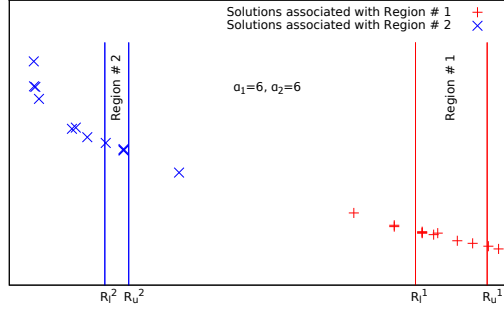


Fig. 2: Association of solutions with regions

The first important addition to pNSGAI is the association of solutions to regions. Before the optimization, the user provides the preferred number of solutions associated with each region. During the optimization, the solutions are assigned greedily to the regions based on the distance between solutions and regions (see Algorithm 2). The distance from a solution (that is outside of the region) to a region is calculated as  $\min(|f_i(A) - R_u|, |f_i(A) - R_l|)$ , where  $f_i(A)$  is the objective value (the objective dimension on which a user specifies the ranges) of a solution. The distance is 0 if a solution is inside a preferred region. Finally, Figure 2 illustrates an example of Algorithm 2 in step # 11 of Algorithm 1.

In our parent selection procedure, a parent is selected in either one of the following two ways: (i) from the same region that the procedure is currently working on ( $z^{th}$  region, step # 6 of Algorithm 1), (ii) from other regions. The selection of parents from other regions depends on a user-defined probability  $p_{ps}$ . This approach enables us to prevent an algorithm from getting trapped on a local multi-dimensional front, which we have observed in preliminary experiments.

---

**Algorithm 2** Associating regions to solutions

---

```
1:  $R_u, R_l$                                 ▷ Upper and lower bounds of given regions
2:  $S$                                          ▷ A set containing all the solutions
3:  $c\alpha \leftarrow 0$                        ▷ Current number of associated solutions with regions
4: for  $s \in S$  do
5:   for  $i \leftarrow 1$  to  $nR$  do
6:     if  $s$  is within  $R_u^i$  and  $R_l^i$  And  $c\alpha_i < \alpha_i$  then
7:       Associate  $s$  with  $i^{th}$  region;  $c\alpha_i \leftarrow c\alpha_i + 1$  and exit the loop
8:     if  $s$  is not yet associated then
9:       Depending on the status of  $c\alpha_j$ , associate  $s$  with the region  $j$  that has
          minimum distance to  $s$  (if  $c\alpha_j = \alpha_j$ , associate  $s$  with the region that has
          second least distance)
10:       $c\alpha_j \leftarrow c\alpha_j + 1$ 
```

---

When a parent is selected from the current working region, a tournament selection based approach is adopted. To select a parent, a given number of tournaments are played between randomly selected associated individuals (associated with the  $z^{th}$  region). The winner is decided based on the ordered criteria: (i) distance from a given region, (ii) overall constraints violation [5], (iii) dominance relation. The order of the criteria is strictly followed. Therefore, if a solution is closer than another solution with respect to a given region, then the subsequent two criteria are not considered. The overall constraint violation and dominance relations come into play when two solutions are within the given region.

To rank the individuals, we propose a ranking procedure based on dominance relations and closeness of an individual to the preferred regions (same ordered criteria as in parent selection). However, we do not apply the proposed ranking procedure in all generations, as narrowing down the search to some particular regions from the beginning may be problematic. Therefore, for a particular generation (Step # 14, Algorithm 1), only one of the two ranking procedures (i.e., default NSGAI ranking procedure and proposed ranking procedure) is applied with probability  $p_{rk} = \left(\frac{usedBudget}{totalBudget}\right)^n$ . The shape of this schedule can be controlled through the exponent  $n$ . For  $n = 1$ , the probability of applying the proposed ranking procedure is increased linearly over time. For larger values of  $n$ , the probability increases sharply in later stages of the algorithm's run.

### 3.2 pAGE

The algorithm Approximation-Guided Evolution (AGE) [3] in its original formulation uses an archive  $A$  in which it maintains a list of all non-dominated solutions seen. This archive can grow and thus slow down the algorithm. In its newer version, AGE maintains an archive  $A^\epsilon$  that is an  $\epsilon$ -approximation of all non-dominated solutions encountered [21, 23] and it uses more efficient parent selection [22]. In the following, we present two straightforward uses of the archive to guide the optimization towards preferred regions (see Algorithm 3). We name the two different uses *pAGEonline* and *pAGEoffline*.

---

**Algorithm 3**  $(\mu + \lambda)$ -Approximation Guided Evolution with preferences

---

```
1: Initialize population  $P$  with  $\mu$  random individuals, and set archive  $A \leftarrow P$ .
2: for each generation do
3:   Initialize offspring population  $O \leftarrow \emptyset$ 
4:   for  $j \leftarrow 1$  to  $\lambda$  do
5:     Select two random individuals from  $P$ , and apply crossover and mutation
6:     Add new individual to  $O$ , if it is not dominated by any individual from  $P$ 
7:   Insert each offspring in the archive  $A$  and in the population, i.e.,  $P \leftarrow P \cup O$ 
8:   [pAGEonline] remove each outlier from  $P$  with  $p_r$ 
9:   while  $|P| > \mu$  do
10:    Remove  $p$  from  $P$  for which the approximation of  $A$  by  $P$  is the smallest
        when  $p$  is left away
11: [pAGEoffline] 12.1: Remove all outliers from archive  $A$ 
                  12.2:  $P \leftarrow A$ 
                  12.3: if  $|P| > \mu$  then apply steps 9–10 to reduce the  $P$ 
```

---

*pAGEonline* largely corresponds to any of the above-mentioned AGE variants. After the generation of the offspring set  $O$  based on the population  $P$ , AGE would normally proceed to consider the union  $P \cup O$  and then reduce this set greedily to approximate the archive. At this point, we insert one action (step # 8): from the union  $P \cup O$  we remove each of the solutions that are outside the preferred regions with probability  $p_r$ .

In preliminary experiments we observed that a static choice of  $p_r = 1$  can be problematic, as this always removes all outliers. As an alternative to this we decided to increase  $p_r$  by reusing the exponential schedule for  $p_{rk}$  that we already use in pNSGAI. This way, the pressure remains low for a long time, which allows pAGEonline to find the front, and in the last generations pAGEonline can focus on spreading out the solutions within the preferred regions. Note that this schedule is by no means optimal.

*pAGEoffline* corresponds to the original AGE with post-processing added. First, pAGEoffline removes all *outliers* (solutions outside the preferred regions) from the archive  $A$ , and it assigns a copy of this reduced archive to the population  $P$ . Then, as  $P$  might be larger than the desired population size, we use AGE’s internal reduction mechanism from steps # 9–10 so that  $P$  approximates  $A$  well.

## 4 Experimental Study

We conduct a range of experiments to analyze the performance of our proposed algorithms pNSGAI, pAGEonline, and pAGEoffline. The benchmark problems include five two-dimensional benchmark problems from the ZDT family [11] and two three-dimensional problems from the DTLZ family [11]. To the best of our knowledge, no directly comparable algorithms for multiple preferred regions are available from the literature; algorithms that consider reference point(s) have a different goal, which puts them at a disadvantage by definition (see Section 2). Therefore, we compare our approaches with their original algorithms, and we vary the evaluation budgets and population sizes to investigate the effectiveness.

Problem	Algorithm	$\mu$	$FE$
	pMOEA	30	12000
ZDT	MOEA	30	12000
		100	12000
		100	24000
DTLZ	pMOEA	30	50000
	MOEA	150	49950

Table 1: Configurations in terms of population size  $\mu$  and evaluation budget  $FE$  to test the efficiency of the interval-based preferences.

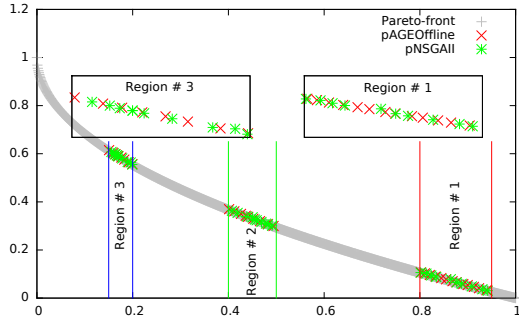


Fig. 3: Pareto fronts obtained by pAGEoffline and pNSGAI on ZDT1 problem.

#### 4.1 Experimental Setup

We developed all algorithms in the jMetal framework [8], which is a Java-based multi-objective meta-heuristic framework. Initially, each pMOEA variant is tested on the ZDT family with two configurations, based on population size  $\mu$  and maximum function evaluations ( $FE$ ). Table 1 presents the different configurations used in the experiments. The configurations are chosen in this way to demonstrate the efficiency of pMOEA in terms of convergence speed.

We consider short and long runs with an evaluation budget of  $FE = 12000$  and  $FE = 24000$  respectively. On the ZDT<sup>2</sup> family, we conduct only short runs of the pMOEAs, and we compare these with short and long runs of the original MOEAs to investigate the efficiency. For the DTLZ<sup>3</sup> functions, we only use a single configuration of the original algorithm since the solutions with an increased population size, are otherwise the solutions would be very thinly spread out over the three-dimensional front.<sup>4</sup>

In this study, the regions are defined in terms of  $R_l$  and  $R_u$  along the first objective. The regions are  $[0.80, 0.95]$ ,  $[0.40, 0.50]$  and  $[0.15, 0.20]$ . Moreover, we set  $\alpha = [10, 10, 10]$  for pNSGAI so that 10 solutions will be associated with each region - this was chosen arbitrarily, however, it is possible to have different numbers of solutions associated with different regions. Simulated binary crossover, polynomial mutation, and binary tournament selection [6] are used with their default values in the jMetal framework. In addition,  $n = 10$  is used for  $p_{rk}$  and  $p_r$  for pNSGAI and pAGEonline, respectively.  $p_{ps} = 0.20$  is used within pNSGAI's parent selection procedure. In pAGEonline,  $\epsilon_{grid} = 0.01$  is used for the approximating archive.

<sup>2</sup> The ZDT functions are used as provided by the jMetal framework. The number of decision variables is 30 for ZDT1/2/3 and 10 for ZDT4/6.

<sup>3</sup> Number of decision variables is 12 for DTLZ2/3, as set in the jMetal framework.

<sup>4</sup> If we use  $\mu = 30$  for typical MOEA (please see Table 1) then it is less probable to find adequate number of solutions in preferred regions, that makes it difficult to compare with pMOEA. In addition, compared in terms of  $FE$ , MOEA uses 50 less function evaluations than pMOEA only because the number is compatible with  $\mu$  (no extra function evaluations after completing last generation).

Over the years, a number of evaluation metrics for multi-objective optimization algorithms have been proposed. We use the popular ones available in the jMetal framework, i.e., the covered hypervolume (HV) [10], and additive  $\epsilon$ -approximation (EPS) [13] to measure the performance of the MOEAs<sup>5</sup>. We use them with a simple modification, i.e., separately for each preferred region. As the true Pareto front is required for the calculation of EPS values, we use the ones provided by the jMetal framework. From these, we extract the regional fronts from the original ones by discarding all outliers. To calculate HV values, we define the reference point for each region to be based on the extreme values in the preferred region. For example, in the introductory Figure 1, these reference points are (0.2, 0.62), (0.5, 0.38), and (0.95, 0.1). It is important to note that performance indicators for preference-incorporating algorithms exist (e.g. [10]), however, these are for reference point-based approaches and thus not applicable.

We run each algorithm independently 100 times and report the averaged indicator values in the following.<sup>6</sup>

## 4.2 Results and Discussion

Firstly, we present in Figure 3 an example of Pareto fronts obtained. We can observe that the solutions are concentrated in the user-defined regions.

Next, we report the results in terms of mean values and the corresponding standard deviations of HV and EPS for each region. Figure 4 shows a subset of the results. In different colours and by using markers of different shapes, we show how our pMOEAs perform compared to their original variants. The top two rows of plots show the results for the NSGAII variants (colour ●, different shapes indicating the different configurations), and in the third row we show the AGE results (colour ●). In the following, we summarize the results.<sup>6</sup>

Most of the time, pNSGAII outperforms NSGAII with the same evaluation budget ( $FE = 12000$ ) regardless of  $\mu$ . pNSGAII performs similarly to NSGAII ( $FE = 24000$ ) a number of times, i.e. on all regions for ZDT1/ZDT2 and regions #1/#2 for ZDT6. pNSGAII fails to converge on ZDT4 due to local optima.

The next row in Figure 4 demonstrates the comparison of pAGE variants (i.e., online and offline) and AGE with different configurations; we limit ourselves to approximation values due to space constraints. pAGEonline and offline perform consistently better in comparison to the generic AGE (with  $FE = 12000$  regardless of  $\mu$ ) for almost all the regions and all the problems. When comparing with AGE ( $FE = 24000$ ), sometimes pAGE performs better (ZDT1 and ZDT3), sometimes similar (ZDT2, ZDT6) and sometimes worse (ZDT4).

To briefly demonstrate that the approach also works on three-dimensional problems, we show a few results in Figure 5; all other algorithms and configurations performed worse and were left away due to space constraints. Although

<sup>5</sup> We do not report other indicator values, such as inverted generational distance (IGD) [20] or the Hausdorff distance [19] due to space constraints.

<sup>6</sup> We uploaded all code and results to <https://github.com/shaikatcse/pMOEAs>. This includes pSPEA2 as an algorithm and also IGD indicator values.



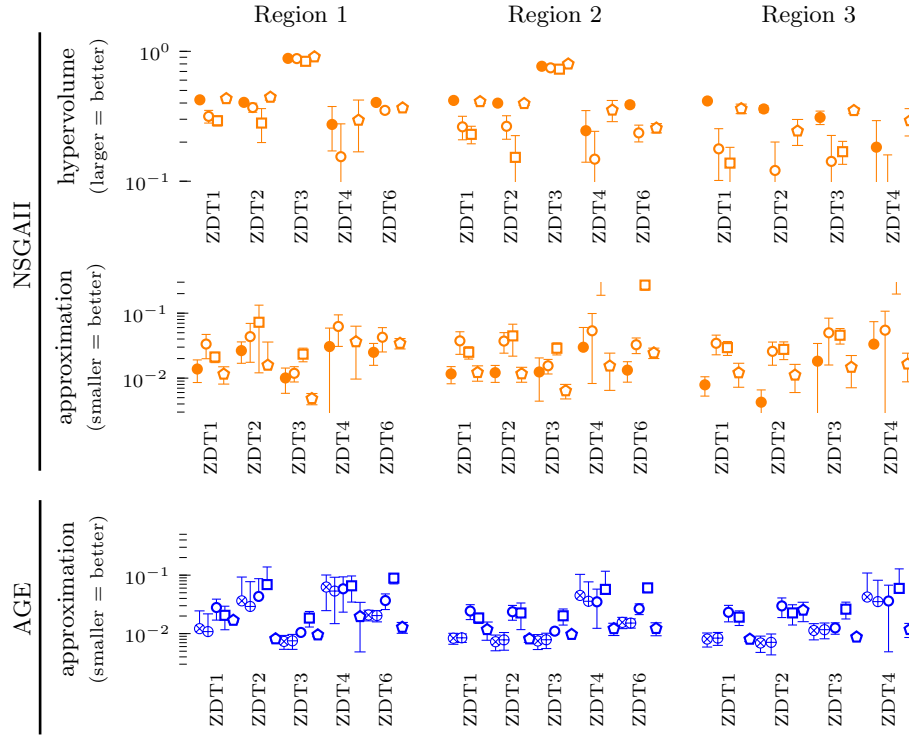


Fig. 4: Comparison of our pMOEAs with their original variants on the ZDT functions with  $m = 2$ . The Regions 1–3 are defined in Section 4.1. Shown are the means and standard deviations of 100 independent runs. From top to bottom, we show the results of NSGAI variants (colour ●), and AGE variants (colour ●). Within each block of four (respectively five) markers, we first show our pMOEA variant with ●, then with ○ and □ the original algorithm with two population sizes ( $\mu = 30$  and  $\mu = 100$ ,  $FE = 12,000$  each), and then with ◇ the original algorithm with twice the evaluation budget ( $\mu = 100$ ,  $FE = 24,000$ ). For pAGE, instead of solid circles, the crossed circles denote our pAGEoffline (×) and pAGEonline (+) variants. In short, the original algorithm with twice the evaluation budget typically performs similar to our pMOEAs.

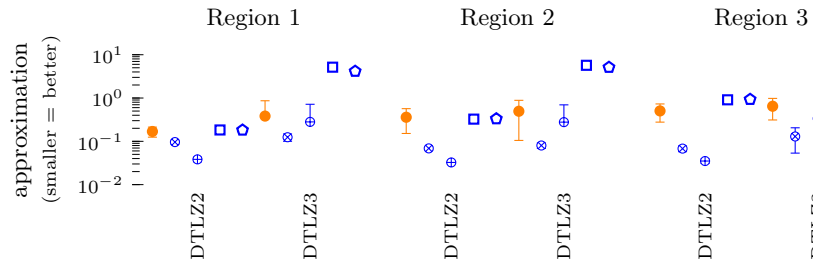


Fig. 5: Comparison of our pMOEAs ( $\mu = 30$ ,  $FE = 50,000$ ) on a subset of the DTLZ functions with  $m = 3$ . ● denotes pNSGAI and ● denotes our pAGEoffline (×) and pAGEonline (+) variants, and □ and ◇ denote the original AGE algorithm with  $\mu = 100$ ,  $FE = 50,000$  and  $\mu = 150$ ,  $FE = 49,950$  respectively. Typically, our pMOEAs achieve significantly better approximations of the preferred regions.

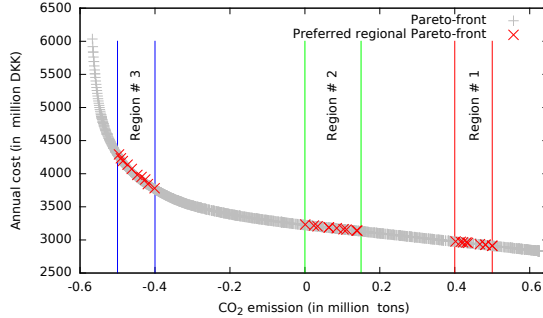


Fig. 6: Result of the energy system optimization. Shown is the final solution set computed by pNSGAI.

on the two-dimensional ZDT problems, all pMOEAs performed similarly (when comparing them with each other, see Figure 4), it is clear from Figure 5 that pAGE achieves better approximations than pNSGAI. The original AGE is not able to find the front given the computational budget (being 1–10 units away), whereas our pMOEAs achieve good approximations of the fronts with the same budget (being only 0.05–0.5 unit away). Between pAGEonline and pAGEoffline, there is no clear winner.

We conclude from these results that our pMOEAs are efficient and effective for considering user-defined regions.

## 5 pNSGAI on energy system optimization problem

To investigate the performance of our approach on a real-world problem, we have applied pNSGAI on an energy system optimization problem [14]. We are not considering pAGE here as it does not have any means of dealing with constraints.

The general goal of the problem is to identify multiple optimal systems in order to minimize  $CO_2$  emission and annual cost. Here, we want to identify multiple optimal systems for three specific regions of interest (i.e., 10 solutions for each region) for the Aalborg energy system [17]. The three regions are defined in terms of  $CO_2$  emission (i.e.,  $[0.40, 0.5]$ ,  $[0.0, 0.15]$  and  $[-0.40, -0.50]$ ). For example, we are interested in identifying 10 optimal solutions in a region within 0.40 to 0.5 million tons of  $CO_2$  emission. Details of energy system optimization framework and Aalborg energy system can be found in [14].

The result is illustrated in Figure 6;  $x$ -axis presents emission in million tons and  $y$ -axis presents annual cost in million Danish Krone. The gray points represent the true Pareto front, which is approximated by considering the outcomes of 240 independent runs of multi-objective algorithms. The red marker show the solutions found by our pNSGAI ( $\mu = 30$ ,  $EF = 6000$ , problem-specific constraint handling was added): 10 solutions per region are found, and they are very close to optimal solutions. As the experiment achieved these set goals, we conclude that our proposed approach can not only be successfully applied to test functions, but also to real-world optimization problems.

## 6 Conclusions

In this article, we proposed the concept of incorporating multiple user preferences into MOEAs via the use of intervals. The concept was designed with laypeople in mind who might not have detailed knowledge about the objective space.

We presented modifications for two MOEAs to handle multiple preferences, and we demonstrated the resulting capability on two- and three-dimensional test problems. On two-dimensional problems, our pMOEAs typically achieve the same hypervolume and additive approximation values as the original algorithms, where the latter had twice the evaluation budget. On three-dimensional problems, our online and offline variants of AGE with preferences perform best. Finally, the effectiveness of the algorithm is investigated on a real-world problem.

As solutions can be spread too diversely over the objective space for higher dimension problems (having more than 3 objectives), we think that preferences in general can be an interesting option for decision makers.

Our future work will include the adaption of the techniques to the higher dimensional problems. Technically, the extension is straightforward, as the intervals just have to be added to an internal array. Whether the approaches are effective in higher-dimensional objective spaces remains to be seen.

*Acknowledgements.* This work has been supported by the ARC Discovery Early Career Researcher Award DE160100850.

## Bibliography

- [1] J. Branke. *Multiobjective Optimization: Interactive and Evolutionary Approaches*, chapter Consideration of Partial User Preferences in Evolutionary Multiobjective Optimization, pages 157–178. Springer, 2008.
- [2] J. Branke, H. Schmeck, K. Deb, and M. R. S. Parallelizing multi-objective evolutionary algorithms: cone separation. In *Congress on Evolutionary Computation (CEC)*, volume 2, pages 1952–1957 Vol.2, 2004.
- [3] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner. Approximation-guided evolutionary multi-objective optimization. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1198–1203. AAAI, 2011.
- [4] S. Chand and M. Wagner. Evolutionary many-objective optimization: A quick-start guide. *Surveys in Operations Research and Management Science*, 20(2):35 – 42, 2015.
- [5] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4):311–338, 2000.
- [6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Trans. on Evolutionary Comp.*, 6(2):182–197, 2002.
- [7] K. Deb, J. Sundar, U. B. R. N, and S. Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithms. In *International Journal of Computational Intelligence Research*, pages 635–642. Springer-Verlag, 2006.
- [8] J. J. Durillo and A. J. Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42:760–771, 2011.
- [9] T. Friedrich, T. Kroeger, and F. Neumann. Weighted preferences in evolutionary multi-objective optimization. In *Advances in Artificial Intelligence (AI)*, pages 291–300. Springer, 2011.

- [10] Y. Guo, J. Zheng, and X. Li. An improved performance metric for multiobjective evolutionary algorithms with user preferences. In *Congress on Evolutionary Computation (CEC)*, pages 908–915, 2015.
- [11] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *Transactions on Evolutionary Computation*, 10(5):477–506, 2006.
- [12] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *Congress on Evolutionary Computation (CEC)*, pages 2419–2426, 2008.
- [13] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.
- [14] M. S. Mahbub, M. Cozzini, P. A. Østergaard, and F. Alberti. Combining multi-objective evolutionary algorithms and descriptive analytical modelling in energy scenario design. *Applied Energy*, 164:140–151, 2016.
- [15] A. Mohammadi, M. Omidvar, and X. Li. A new performance metric for user-preference based multi-objective evolutionary algorithms. In *Congress on Evolutionary Computation (CEC)*, pages 2825–2832, 2013.
- [16] A. Nguyen, M. Wagner, and F. Neumann. User preferences for approximation-guided multi-objective evolution. In *Simulated Evolution and Learning (SEAL)*, pages 251–262. Springer, 2014.
- [17] P. A. Østergaard, B. V. Mathiesen, and B. Möller. A renewable energy scenario for Aalborg Municipality based on low-temperature geothermal heat, wind power and biomass. *Energy*, 35(12):4892–4901, 2010.
- [18] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang. A review of hybrid evolutionary multiple criteria decision making methods. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 1147–1154. IEEE, 2014.
- [19] G. Rudolph, O. Schütze, C. Grimme, and H. Trautmann. A multiobjective evolutionary algorithm guided by averaged hausdorff distance to aspiration sets. pages 261–273. Springer, 2014.
- [20] H. Sato, H. Aguirre, and K. Tanaka. Local dominance using polar coordinates to enhance multiobjective evolutionary algorithms. In *Congress on Evolutionary Computation (CEC)*, pages 188–195, 2004.
- [21] M. Wagner, K. Bringmann, T. Friedrich, and F. Neumann. Efficient optimization of many objectives by approximation-guided evolution. *European Journal of Operational Research*, 243(2):465 – 479, 2015.
- [22] M. Wagner and T. Friedrich. Efficient parent selection for approximation-guided evolutionary multi-objective optimization. In *Congress on Evolutionary Computation (CEC)*, pages 1846–1853, 2013.
- [23] M. Wagner and F. Neumann. A fast approximation-guided evolutionary multi-objective algorithm. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 687–694. ACM, 2013.
- [24] T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Evolutionary Multi-Criterion Optimization*, volume 4403, pages 742–756. Springer, 2007.
- [25] A. P. Wierzbicki. *Reference Point Approaches*, pages 237–275. Springer US, Boston, MA, 1999.