# User preferences for Approximation-Guided Multi-Objective Evolution

Anh Quang Nguyen, Markus Wagner, and Frank Neumann

Optimisation and Logistics
School of Computer Science
The University of Adelaide, Australia

**Abstract.** Incorporating user preferences into evolutionary multi-objective evolutionary algorithms has been an important topic in recent research in the area of evolutionary multi-objective optimization. We present a very simple and yet very effective modification to the Approximation-Guided Evolution (AGE) algorithm to incorporate user preferences. Over a wide range of test functions, we observed that the resulting algorithm called iAGE is just as good at finding evenly distributed solutions as similarly modified NSGA-II and SPEA2 variants. However, in particular for "difficult" two-objective problems and for all three-objective problems we see more evenly distributed solutions in the user preferred region when using iAGE.

**Keywords:** multi-objective optimisation, approximation, user preference

## 1 Introduction

Many real-world optimization problems consist of multiple objectives that conflict with each other. Solving a multi-objective optimization (MOO) problem usually means finding a set of trade-offs regarding the given objective functions. The set of all trade-offs according to the given objective functions is called the Pareto front of the underlying MOO problem. Since the size of the Pareto front can grow exponentially for discrete problems and can even be infinite for continuous problems, evolutionary algorithms on MOO problems have to restrict themselves to a smaller set of solutions which should be a good approximation of the Pareto front. There are different algorithms such as NSGA-II [4], SPEA2 [21], or IBEA [19] which try to solve two main goals of a MOO problem: find the Pareto front or a good approximation thereof by preferring a diversity of non-dominated solutions.

Motivated by the studies of multiplicative and additive approximations for multi-objective problems [3, 7, 16], the algorithm Approximation-Guided Evolution (AGE) has been introduced in [2]. AGE works with a formal notion of approximation and improves the approximation quality during its runtime without having a full knowledge about the true Pareto front. The results in [2, 17] show that, given a fixed number of evaluation, AGE outperforms state-of-the-art algorithms in terms of additive approximation and covered hypervolume. AGE has later been improved in [18] to overcome the problem of over growth archive size in high dimensional objective spaces by adapting the $\epsilon$-dominance approach and a non-random selection of parents used for next generation of population.

Recently, great efforts have been made in order to incorporate user preferences into evolutionary multi-objective optimization (EMO) where specific regions in the objective space have higher priority than others. For NSGA-II, a reference point approach has been presented in [6]. Later on, the crowding distance assignment function has been changed in order to meet the requirement of non-even distribution of solutions along the Pareto front [10]. Zitzler et al. [22] have shown that the weighted hypervolume indicator is a good method to integrate user preferences and showed that their results are superior than the ones obtained by NSGA-II and SPEA2, where no user preference information is considered. However, all of these hypervolume-based approaches have a negative effect on the runtime of the algorithm because they require exponential runtime in the number of dimensions [1]. To overcome that problem, Friedrich et al. [9] proposed a simple approach to integrate the weight function into a wide range of EMO algorithms, including NSGA-II and SPEA2, and showed that their results now match the ones in [22] without changing the performance of the algorithms.

In relation to our series of works of integrating preferences into existing algorithms is that presented in [13–15]. There, the authors focus on reference points and on a performance metric for comparing algorithms with reference points. The preference functions that are considered in our article here, however, go beyond reference points.

We propose a new variant of AGE [2, 18] called iAGE which incorporates user preferences into the algorithm. iAGE widens the range of preference functions by using not only reference points but also preferred regions and spaces. Furthermore, we change the selection process of AGE by considering the preference functions as a factor to keep or discard solutions from the population while still keep the complexity remaining unchanged. Our experimental results show that iAGE is fast and works just as well as integrated NSGA-II and SPEA2. Furthermore, iAGE provides more evenly distributed solutions in the preferred region of the objective space.

The outline of this paper is structured as follows. In Section 2, we introduce some basic definitions of multi-objective optimization and the AGE algorithm. Section 3 shows how user preferences are incorporated into AGE and how input parameters can affect the distribution of solutions. In Section 4, we report on our experimental results, and compare them with the ones from NSGA-II and SPEA2. Finally, we finish with some conclusions.

## 2 Preliminaries

In this section, we give a basic introduction into the setting for multi-objective optimization, the approach of using weight function to incorporate user preferences, and the approximation-guided evolution approach.

### 2.1 Multi-objective optimization

In multi-objective optimization the task is to optimize a function $f = (f_1, \ldots, f_d) : S \rightarrow \mathbb{R}_+^d$ with $d \geq 2$, which assigns to each element $s \in S$ a $d$-dimensional objective vector. Each objective function $f_i \colon S \mapsto \mathbb{R}$, $1 \leq i \leq d$, maps from the considered search space $S$ into the positive real values. Elements from $S$ are called search points and the corresponding elements $f(s)$ with $s \in S$ are called objective vectors.

Throughout this paper, we consider the minimization problems of $d$ objectives. In multi-objective optimization the given objective functions $f_i$ are usually conflicting, which implies that there is no single optimal objective vector. Instead of this the Pareto dominance relation is defined, which is a partial order. In order to simplify the presentation we only work with the Pareto dominance relation on the objective space and mention that this relation transfers to the corresponding elements of $S$.

The Pareto dominance relation $\preceq$ between two objective vectors $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$, with $x, y \in \mathbb{R}^d$ is defined as

$$x \preceq y \;:\Leftrightarrow\; x_i \leq y_i \text{ for all } 1 \leq i \leq d.$$

We say that $x$ dominates $y$ iff $x \preceq y$. If

$$x \prec y \;:\Leftrightarrow\; x \preceq y \text{ and } x \neq y$$

holds, we say that $x$ strictly dominates $y$ as $x$ is not worse than $y$ with respect to any objective, and at least better with respect to one of the $d$ objectives.

The objective vectors $x$ and $y$ are called incomparable if

$$x \parallel y \;:\Leftrightarrow\; \neg(x \preceq y \vee y \preceq x)$$

holds. Two objective vectors are therefore incomparable if there are at least two (out of the $d$) objectives where they mutually beat each other. An objective vector $x$ is called Pareto optimal if there is no $y = f(s)$ with $s \in S$ for which $y \prec x$ holds. The set of all Pareto optimal objective vectors is called the Pareto front of the problem given by $f$. Note that the Pareto front is a set of incomparable objective vectors.

Even for two objectives the Pareto front might grow exponentially with respect to the problem size. Therefore, algorithms for multi-objective optimization usually have to restrict themselves to a smaller set of solutions. This smaller set is then the output of the algorithm.

We make the notion of approximation precise by considering a weaker relation on the objective vectors called additive $\epsilon$-dominance. It is defined as

$$x \preceq_{\epsilon^+} y \;:\Leftrightarrow\; x_i + \epsilon \leq y_i \text{ for all } 1 \leq i \leq d.$$

Furthermore, we also define additive approximation of a set of objective vectors $T$ with respect to another set of objective vectors $S$.

**Definition 1.** *For finite sets $S, T \subset \mathbb{R}^d$, the additive approximation of $T$ with respect to $S$ is defined as*

$$\alpha(S,T) := \max_{s \in S} \min_{t \in T} \max_{1 \le i \le d} (s_i - t_i).$$

We will use Definition 1 in order to judge the quality of a population $P$ with respect to a given archive $A$ that contains all non-dominated solutions seen so far (or an approximation thereof)—effectively, the value of $\alpha(S,T)$ is the approximation value achieved for the worst-approximated solution. In this way, we can measure how good the current population is with respect to the search points seen during the run of the algorithm.

Although, we are only using the notion of additive approximation, we would like to mention that our approaches can be easily adapted to multiplicative approximation. This can be done by adjusting the definitions accordingly.

### 2.2 User preferences as weight functions in the objective space

User preferences provide information that guides the search process of the algorithm and tells the differences among incomparable solutions. In this article, we denote a weight function $w : \mathbb{R}^d \to \mathbb{R}$ which represents user preferences. In general, $w$ can be an arbitrary function that specifies preferences to certain regions or points in the objective space.

In this article, we will use different *weight* functions for both 2- and 3-dimensional problems, which calculate the weight of a solution based on a given *scheme* value. Given a solution $x = \{x_1, x_2, \cdots, x_d\}$, the weight functions for 2-objectives problems, originally introduced in [22] and investigated in [9, 22], are defined as follows:

- *scheme* = 1: Both objectives are treated equally and the weight of a solution $x$ is given by
$$w(x) = (e^{20x_1} + e^{20x_2})/(2 \cdot e^{20})$$
- *scheme* = 2: The user preference is based on only the second objective and the weight of a solution $x$ is given by
$$w(x) = (e^{20x_2})(e^{20})$$
- *scheme* = 3: Given a reference point $r = \{r_1, r_2\}$, solutions closer to this point have higher user preference than the further ones. The weight of a solution $x$ is given by
$$w(x) = \begin{cases} 10^{-5} + \frac{(3-((x_1-r_1)^2+(x_2-r_2)^2))}{(0.001+(2(x_1-r_1)-2(x_2-r_2))^2)} \\ 10^{-5} \text{ otherwise} \end{cases}$$

For 3-objective problems, no *weight* functions had previously been defined. We extended the above-defined schemes 1 and 3:

**Algorithm 1:** Outline of AGE [18]

Initialize population $P$ with $\mu$ random individuals;
Set $\epsilon_{grid}$ the resolution of the *approximative archive* $A_{\epsilon_{grid}}$;
**foreach** $p \in P$ **do**
  Insert offspring *floor*$(p)$ in the *approximative archive* $A_{\epsilon_{grid}}$ such that only non-dominated solutions remain;
**foreach** *generation* **do**
  Initialize offspring population $O \leftarrow \emptyset$;
  **for** $j \leftarrow 1$ **to** $\lambda$ **do**
    Select two individuals from $P$ (see Section 3.2 in [18]);
    Apply crossover and mutation;
    Add new individual to $O$;
  **foreach** $p \in O$ **do**
    Insert offspring *floor*$(p)$ in the *approximative archive* $A_{\epsilon_{grid}}$ such that only non-dominated solutions remain;
    Discard offspring $p$ if it is dominated by any point *increment*$(a)$, $a \in A$;
  Add offsprings to population, i.e., $P \leftarrow P \cup O$;
  **while** $|P| > \mu$ **do**
    Remove $p$ from $P$ that is of least importance to the approximation (for details on this step see [2]);

– *scheme* = 1:
$$w(x) = (e^{20x_1} + e^{20x_2} + e^{20x_3})/(3 \cdot e^{20})$$
– *scheme* = 3: Given a reference point $r = (r_1, r_2, r_3)$, the weight of a solution $x$ is given by

$$w(x) = \begin{cases} 10^{-5} + \frac{(3-((x_1-r_1)^2+(x_2-r_2)^2+(x_3-r_3)^2))}{0.001+(\frac{(x_1-r_1)+(x_2-r2)+(x_3-r_3)}{3})^2} \\ 10^{-5} \text{ otherwise} \end{cases}$$

For *scheme* = 3 we selected a reference point of $(0.5, 0.6)$ for the two-dimensional problems, and $(0.5, 0.6, 07)$ for the three-dimensional ones.

How these *weight* functions will be incorporated into AGE will be shown in Section 3

### 2.3 Approximation-Guided Evolution

Definition 1 allows us to measure the quality of the population of an evolutionary algorithm with respect to a given set of objective vectors. AGE [2] is an evolutionary multi-objective algorithm that works with this formal notion of approximation. It stores an archive $A$ consisting of the non-dominated objectives vectors found so far. Its aim is to minimize the additive approximation $\alpha(A, P)$ of the population $P$ with respect to the archive $A$.

**Algorithm 2:** Function *floor* [18]

**input** : $d$-dimensional objective vector $x$, archive parameter $\epsilon_{grid}$
**output**: Corresponding vector $v$ on the $\epsilon$-grid

**for** $i = 1$ **to** $d$ **do** $v[i] \leftarrow \left\lfloor \frac{x[i]}{\epsilon_{grid}} \right\rfloor$ ;

**Algorithm 3:** Function *increment* [18]

**input** : $d$-dimensional vector $x$, archive parameter $\epsilon_{grid}$
**output**: Corresponding vector $v$ that has each of its components increased by 1

**for** $i = 1$ **to** $d$ **do** $v[i] \leftarrow o[i] + 1$ ;

We consider the further developed version of AGE (called AGE-II in [18]). This algorithm is parametrized by the desired approximation quality $\epsilon_{grid} \geq 0$ of the archive with respect to the seen objective vectors. The algorithm is shown in Algorithm 1, and it uses the helper functions given in Algorithms 2 and 3. The latter is used to perform a relaxed dominance check on the offspring $p$ in Line 13. A strict dominance check here would require an offspring to be not dominated by any point in the entire archive. However, as the archive approximates all the solutions seen so far (via the flooring), it might be very unlikely, or even impossible, to find solutions that pass the strict dominance test.

## 3 Adding User Preferences

Interactive AGE (iAGE) is a variant of AGE that considers user preferences as one of its parameters called *scheme* along with using the corresponding *weight* functions, which is mentioned in Section 3.2. The selection process of iAGE follows the same structure as the original AGE [2]. Let $P$ be the current population where we need to remove an individual and $A$ be the current archive. For each solution $a \in A$, we denote the best and second best approximation $\alpha_1(a)$, $\alpha_2(a)$ accordingly while $p_1(a)$ and $p_2(a)$ are solutions $p \in P$ that approximates $a$ best and second best. In case of AGE, $p \in P$ with minimum $\beta(p)$ is removed from the population where $\beta(p)$ is known as the importance of solution $p$ and defined as

$$\beta(p) := max_{a \in A}\{\alpha_2(a) | p_1(a) = p\}.$$

iAGE integrates the *weight* function into the selection process of the algorithm to ensure that user preference is one of the factors that decides whether a solution is removed or accepted to the next generation. We use a combination between the weight, $w(p)$, and the approximation to determine the importance of a given solution $p$ given by expression

$$\beta(p) := max_{a \in A}\{w(p) \cdot \alpha_2(a) | p_1(a) = p\}.$$

Let $\beta_{min} := \min_{p \in P} \beta(p)$ be the minimum $\beta$-value among all individuals of the population. The selection process removes a $p$ from $P$ for which $\beta(p) = \beta_{min}$

**Algorithm 4:** Outline of iAGE selection process.

See lines 12-16 in [2], Algorithm 4
**foreach** *solution* $p \in P$ **do**
    $\beta(p) := max_{a \in A}\{w(p) \cdot \alpha_2(a) | p_1(a) = p\}$
**while** $|P| > \mu$ **do**
    Remove an individual $p^* = \arg\min_{p \in P}(\beta(p), w(p))$ chosen uniformly at random from $P$.
    See lines 21-23 in [2], Algorithm 4

holds. If there are multiple solutions $p$ of value $\beta_{min}$, one with the smallest weight $w$ value is discarded. If there are multiple solutions $p$ with the same $\beta_{min}$ and $w$ value, then the removed solution is chosen at random. Therefore, the selection process removes an individual $p$ from $P$ that has the smallest vector $(\beta(p), w(p))$ according lexicographic order. The detail of the changes from Algorithm 4 in [2] is shown in Algorithm 4 where min is taken according to lexicographic order of the vector $(\beta(p), w(p))$.

The choice of iAGE's parameter $\epsilon_{\text{grid}}$ influences how well the set of solutions seen so far is approximated. Interestingly, this parameter also has a small but noticeable impact on the distribution of solutions. Some results are shown in Figure 1. As we can see, the solutions are packed more densely with decreasing grid size. The explanation is that the number of potential points in the archive increases, and consequently solutions in the population are more likely to be "responsible" for the approximation of an archive point. This, in combination with the increasing preference, results in a higher density of solutions.

We also investigate the impact of user preference on the distribution of solutions in the final population by providing different adjustments to the *weight* function. In particular, given the calculated $\beta$ value for each $p \in P$, we want to study how different adjusted weight functions overwhelm the approximation and hence affect the selection process of the algorithm. In the following example, three adjustment strategies $d$ are used:

- weight strategy 1 : $w(x) = w(x)$
- weight strategy 2 : $w(x) = sqrt(w(x))$
- weight strategy 3: $w(x) = ln(1 + w(x))$

We show some results in Figure 2. It can be seen that the choice of the adjustment strategy has hardly any impact on the distribution of points. The reason is that the weight remains its high impact even after a logarithmic scale-down. In addition, the approximation part of the adjustment only ensures that the archive points are better and better approximated; the slight change in the relative positioning of "the best population point for an archive point" (after considering the weight and the adjustment strategy) is barely noticeable in the final population and within the typical variations of results of randomized algorithms.

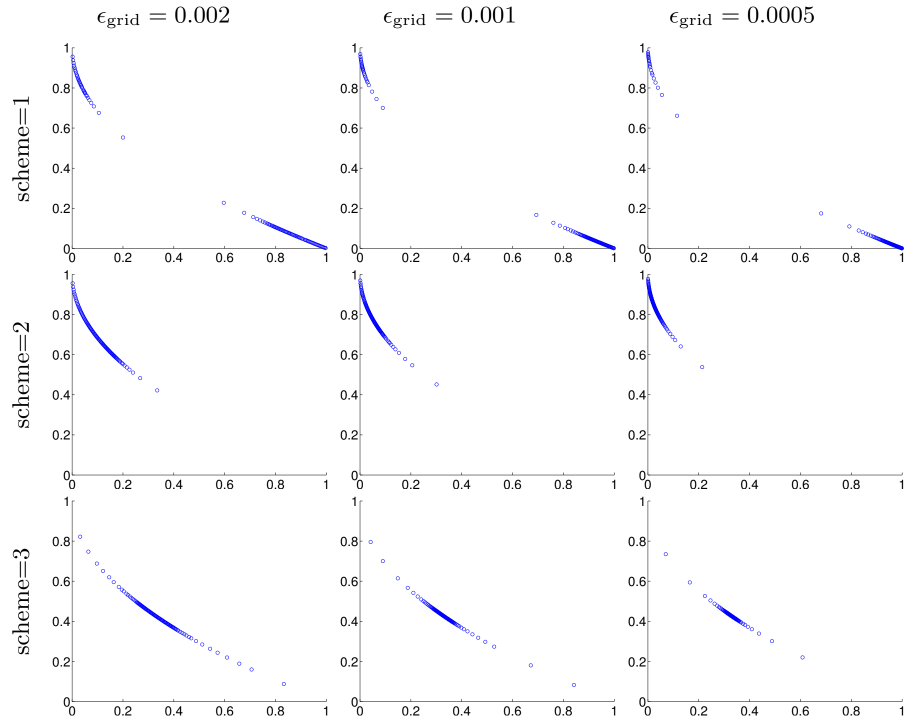


**Fig. 1.** Influence of $\epsilon_{\text{grid}}$ on the distributions of the solutions. The underlying problem is ZDT 1.

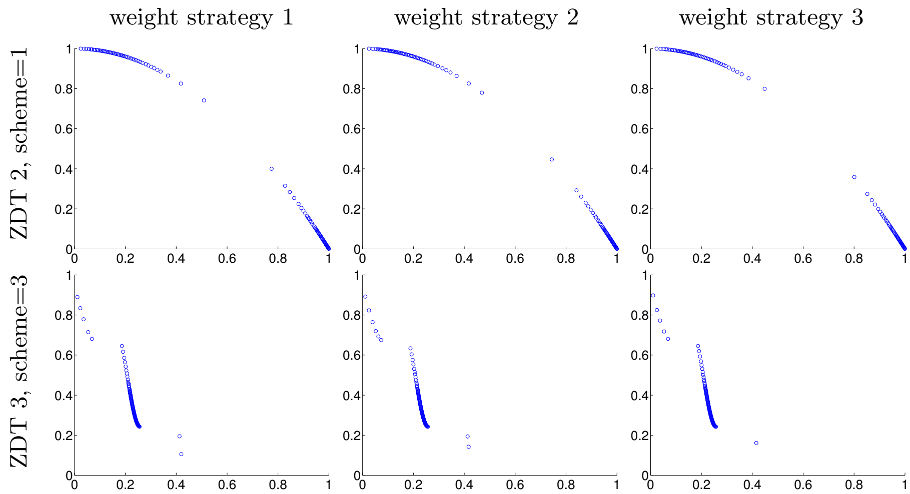


**Fig. 2.** Influence of different adjustment strategies to the weight function of iAGE with $\epsilon_{\text{grid}} = 0.0005$.

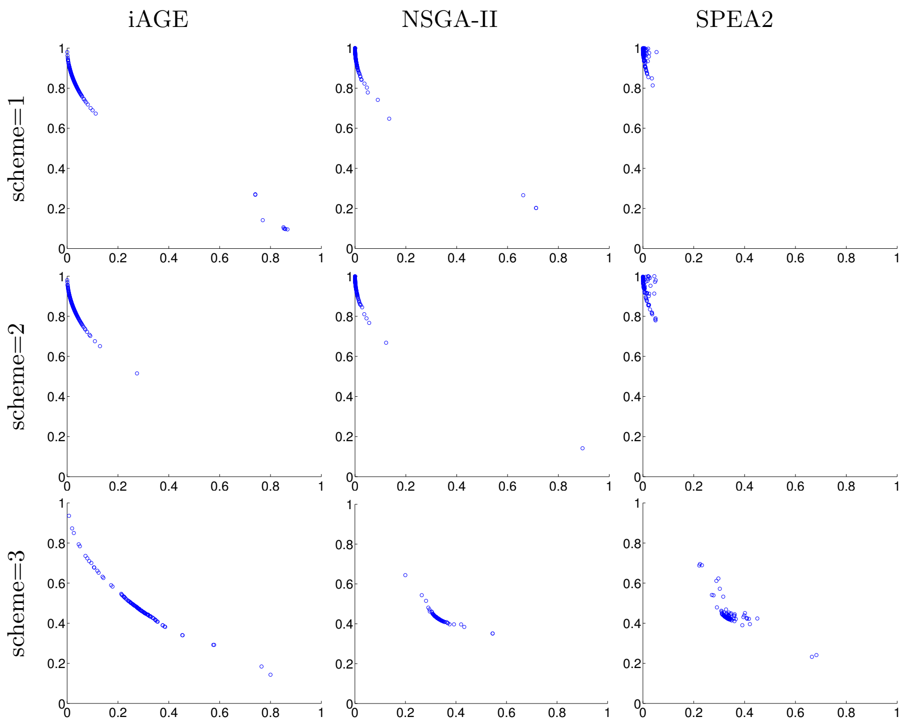


**Fig. 3.** LZ F5, 100.000 evaluations.

## 4 Comparison With Other Algorithms

In our study, we investigate the performance of iAGE (using weight strategy 2) on problems with two and three objectives. We use the jMetal framework [8] to compare iAGE with the established algorithms NSGA-II [4], and SPEA2 [21]. Both algorithms are used as described in [9]: the weight functions are used multiplicatively to adjust either the crowding distance (NSGA-II) or the density (SPEA2). As benchmarks, we use the benchmark families WFG [11] and LZ [12], DTLZ [5], and ZDT [20].[1]

Note that we compare the final populations only visually. For the computation of indicator values, we would need reference sets: these are available for the true Pareto fronts in the "preference free" case, but not when non-linear preferences schemes are considered.

For many problems (mostly for the ZDT family and for DTLZ 1/2/3) we noticed very few differences between the final distributions of the three algorithms. In contrast to this, we noticed for several other problems that all algorithms would have immense problems to achieve good approximations of the true Pareto front when a preference function was used. Some results are shown in Figure 3. The top row shows that all three algorithms have problems to cover the lower

[1] The code is available on our project page
`http://cs.adelaide.edu.au/~optlog/research/foundations.php`

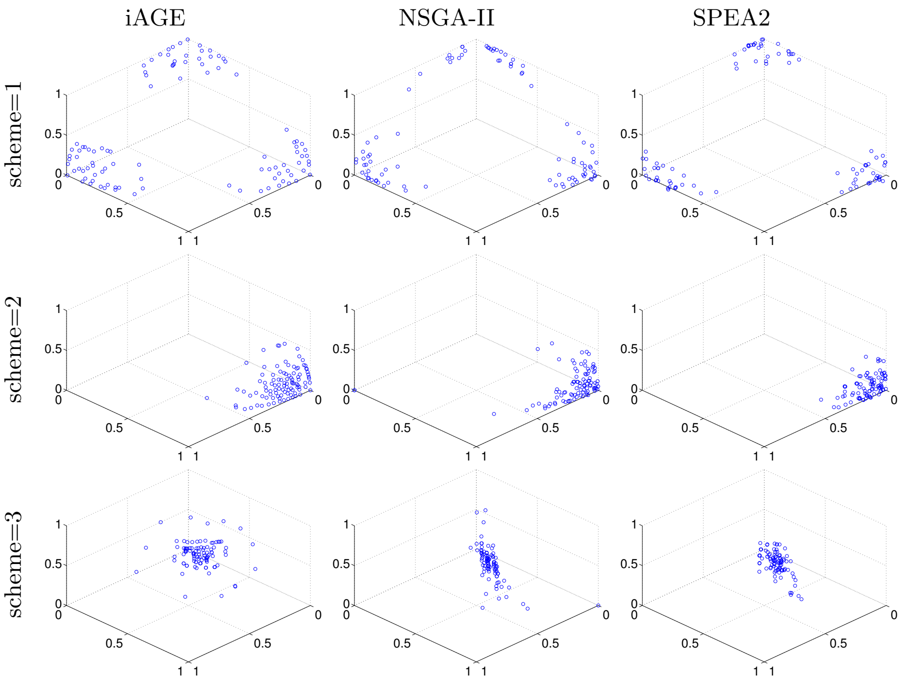


**Fig. 4.** DTLZ 2, d=3, 100.000 evaluations.

right sections of the Pareto front, even though this was a preferred region just as the top left section was. In the bottom row, we can observe that all algorithms find solutions close to the reference point. However, NSGA-II's solutions are often dominated by iAGE's, and SPEA2 itself maintains many dominated solutions.

We conjecture that the use of a preference function can restrict the diversity so much that it is not possible towards the end of the optimisation process to "rediscover" certain parts of the objective space anymore. We observed such difficulties for many functions, including DTLZ 4, many of the LZs and many of the WFGs.

As an example that preferences in objective spaces with more than two dimensions are possible, and as another extension to existing work, Figure 4 shows the results of the different algorithms on DTLZ 2, d=3. Because it is difficult to compare the outcomes using indicator values, we compare them visually. All three algorithms produce solution sets that follow the preference scheme. For iAGE, we notice "ray-like" patterns for the second scheme, and circular patterns around the reference point for the third scheme. NSGA-II and SPEA2, without their sense of an approximated archive, produce sets without any obvious visual structure. Consequently, we argue that iAGE produces the most evenly distributed solutions, even though this is in the eye of the beholder.

## 5 Conclusions

Evolutionary multi-objective methods are often considered in the unbiased case, where no particular area of the objective space is favored. This is in contrast to the actual decision making processes in the real world, where the decision maker typically has a preference for a particular range of non-dominated solutions.

In this article, we presented a simple and yet very effective modification to the algorithm AGE. The resulting algorithm iAGE differs from the original AGE only of the consideration of the weight function in a single step—the overall low computational complexity of the algorithm remains unchanged. Over a wide range of test functions, we observed that iAGE is just as good at finding evenly distributed solutions as similarly modified NSGA-II and SPEA2 variants. However, in particular for "difficult" two-objective problems and for all three-objective problems we have seen more evenly distributed solutions in the preferred regions of the objective space.

## Bibliography

[1] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. In *Algorithms and Computation*, pp. 436–447. Springer, 2008.

[2] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner. Approximation-guided evolutionary multi-objective optimization. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pp. 1198–1203. AAAI Press, 2011.

[3] C. Daskalakis, I. Diakonikolas, and M. Yannakakis. How good is the chord algorithm? In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 978–991. Society for Industrial and Applied Mathematics, 2010.

[4] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 6:182–197, 2002.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, pp. 105–145. 2005.

[6] K. Deb, J. Sundar, N. Udaya Bhaskara Rao, and S. Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithms. *Journal of Computational Intelligence Research*, 2:273–286, 2006.

[7] I. Diakonikolas and M. Yannakakis. Small approximate pareto sets for bi-objective shortest paths and other problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 74–88. Springer, 2007.

[8] J. J. Durillo, A. J. Nebro, and E. Alba. The jMetal framework for multi-objective optimization: Design and architecture. In *IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 4138–4325, 2010.

[9] T. Friedrich, T. Kroeger, and F. Neumann. Weighted preferences in evolutionary multi-objective optimization. In D. Wang and M. Reynolds, editors, *AI 2011: Advances in Artificial Intelligence*, Vol. 7106 of *Lecture Notes in Computer Science*, pp. 291–300. Springer Berlin Heidelberg, 2011.

[10] Q. Hu, L. Xu, and E. D. Goodman. Non-even spread nsga-ii and its application to conflicting multi-objective compatible control. In *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, pp. 223–230. ACM, 2009.

[11] S. Huband, L. Barone, R. L. While, and P. Hingston. A scalable multi-objective test problem toolkit. In *3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO '05)*, Vol. 3410 of *LNCS*, pp. 280–295. Springer, 2005.

[12] H. Li and Q. Zhang. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Trans. on Evolutionary Computation*, 13:284–302, 2009.

[13] A. Mohammadi, M. N. Omidvar, and X. Li. Reference point based multi-objective optimization through decomposition. In *Congress on Evolutionary Computation (CEC)*, pp. 1150–1157. IEEE, 2012.

[14] A. Mohammadi, M. N. Omidvar, and X. Li. A new performance metric for user-preference based multi-objective evolutionary algorithms. In *Congress on Evolutionary Computation (CEC)*, pp. 2825–2832. IEEE, 2013.

[15] A. Mohammadi, L. X. Omidvar, Mohammad Nabi, and D. Kalyanmoy. Integrating user preferences and decomposition methods for many-objective optimization. In *Congress on Evolutionary Computation (CEC)*, pp. 421–428. IEEE, 2014.

[16] S. Vassilvitskii and M. Yannakakis. Efficiently computing succinct trade-off curves. In *Automata, Languages and Programming*, pp. 1201–1213. Springer, 2004.

[17] M. Wagner and T. Friedrich. Efficient parent selection for approximation-guided evolutionary multi-objective optimization. In *Congress on Evolutionary Computation (CEC)*, pp. 1846–1853, 2013.

[18] M. Wagner and F. Neumann. A fast approximation-guided evolutionary multi-objective algorithm. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pp. 687–694, New York, NY, USA, 2013. ACM.

[19] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature-PPSN VIII*, pp. 832–842. Springer, 2004.

[20] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8:173–195, 2000.

[21] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pp. 95–100, 2002.

[22] E. Zitzler, D. Brockhoff, and L. Thiele. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary multi-criterion optimization*, pp. 862–876. Springer, 2007.