Integrating User Preferences into Evolutionary Multi-Objective Optimization

Jürgen Branke[1] and Kalyanmoy Deb[2]

[1]Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany, *branke@aifb.uni-karlsruhe.de*
[2]Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, PIN 208 016, India, *deb@iitk.ac.in*

KanGAL Report Number 2004004

Summary. Many real-world optimization problems involve multiple, typically conflicting objectives. Often, it is very difficult to weigh the different criteria exactly before alternatives are known. Evolutionary multi-objective optimization usually solves this predicament by searching for the whole Pareto-optimal front of solutions. However, often the user has at least a vague idea about what kind of solutions might be preferred. In this chapter, we argue that such knowledge should be used to focus the search on the most interesting (from a user's perspective) areas of the Pareto-optimal front. To this end, we present and compare two methods which allow to integrate vague user preferences into evolutionary multi-objective algorithms. As we show, such methods may speed up the search and yield a more fine-grained selection of alternatives in the most relevant parts of the Pareto-optimal front.

1 Introduction

Many real-world optimization problems involve multiple objectives which need to be considered simultaneously. As these objectives are usually conflicting, it is not possible to find a single solution which is optimal with respect to all objectives. Instead, there exist a number of so called "Pareto-optimal" solutions which are characterized by the fact that an improvement in any one objective can only be obtained at the expense of degradation in at least one other objective. Therefore, in the absence of any additional preference information, none of the Pareto-optimal solutions can be said to be inferior when compared to any other solution, as it is superior in at least one criterion.

In order to come up with a single solution, at some point during the optimization process, a decision maker (DM) has to make a choice regarding the importance of different objectives. Following a classification by Veldhuizen [15], the articulation of preferences may be done either before (a priori), during (progressive), or after (a posteriori) the optimization process.

A priori approaches basically transform the multi-objective optimization problem into a single objective problem by specifying a utility function over all different criteria. However, they are usually not practicable, since they require the user to explicitly and exactly weigh the different objectives before any alternatives are known.

Most Evolutionary Multi-Objective Optimization (EMO) approaches can be classified as a posteriori. They attempt to discover the whole set of Pareto-optimal solutions or, if there are too many, at least a well distributed set of representatives. In this case, the DM can look at a large set of generated alternatives before making a decision and thereby revealing his or her preferences. While this may be very convenient, the search for all Pareto-optimal solutions poses high demands on the optimization algorithm, in particular if the number of objectives is large.

In this chapter, we consider an intermediate approach. Although we agree that it may be impractical for a DM to completely specify his or her preferences before any alternatives are known, we assume that the DM has at least a rough idea about reasonable trade-offs between the different objectives. The methods discussed here aim at integrating such imprecise knowledge into the EMO approach, biasing the search towards solutions that are considered as relevant by the DM. This may yield two important advantages:

1. Instead of a diverse set of solutions, many of them clearly irrelevant to the DM, a search bias towards the DM's preferences will yield a more fine-grained and more suitable selection of alternatives.
2. By focusing the search onto the relevant part of the search space, we expect the optimization algorithm to find these solutions more quickly.

In this chapter, we discuss and compare two ideas to incorporate imprecise user preferences into EMO: The guided multi-objective evolutionary algorithm introduced in [1], and a new biased crowding operator based on the idea of biased sharing as introduced in [8].

The outline of this chapter is as follows: in the next section, we will briefly survey some related work. Section 3 describes the guided dominance and biased crowding schemes. These two approaches are then tested and compared empirically in Section 4. The chapter concludes with a summary in Section 5.

2 Related Work

Evolutionary multi-objective optimization is a very active research area. For comprehensive books on the topic, the reader is referred to [7, 3].

Most work in this area is based on the concept of dominance. A solution x is said to dominate a solution y if and only if solution x is at least as good as y in all objectives, and strictly better in at least one objective. More formally, for a minimization problem (which we will assume throughout the paper without loss of generality), dominance can be specified as follows:

$$x \succ y \Leftrightarrow \forall i \in [1 \ldots m] : f_i(x) \leq f_i(y)$$
$$\wedge \exists j \in [1 \ldots m] : f_j(x) < f_j(y)$$

Figure 1 illustrates the concept for the case of two objectives. In that example, solution C is dominated by solution A and B, while solutions A and B are non-dominated and thus, without any additional information about the DM's preferences, have to be considered indifferent. If a solution is non-dominated with respect to any other solution in the search space it is called Pareto-optimal.

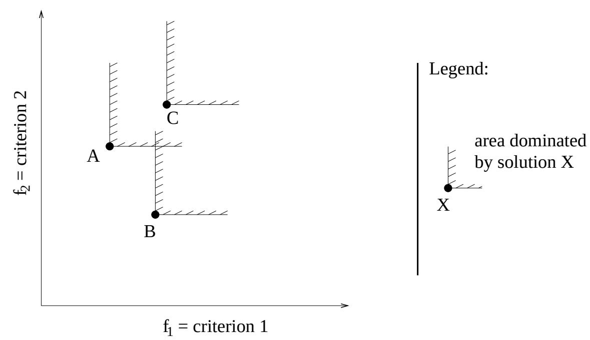

Fig. 1. Illustration of the standard dominance relation

There have already been a number of attempts to handle user preferences in EMO. A somewhat outdated survey is provided by Coello Coello in [2].

The approaches by Fonseca and Fleming [13] and Deb [6] are inspired by goal programming and allow the DM to specify a goal. Based on that information, Fonseca and Fleming give a higher importance to objectives not yet satisfying the goal. Deb uses the provided goal to modify the optimization criteria. Basically, the distances from the goal rather than the actual criteria are considered. If the provided goal is set appropriately, such approaches may indeed restrict the search space to the relevant region. However, the difficulty is to decide on the goal a priory, i.e. before any alternatives are known. If the goal is set too ambitious, it has basically no effect, as none of the solutions will reach the goal even for a single objective, and thus the search is not restricted at all. On the other hand, if the goal is easily attainable, search is actually hindered.

The method by Cvetkovic and Parmee [4] assigns each criterion a weight w_i, and additionally requires a minimum level for dominance τ. Then, the concept of dominance is defined as follows:

$$x \succ y \Leftrightarrow \sum_{i: f_i(x) \leq f_i(y)} w_i \geq \tau$$

with strict inequality for at least one objective. To facilitate specification of the required weights, they suggest a method to turn fuzzy preferences into specific quantitative weights. However, since for every criterion the dominance scheme only considers whether one solution is better than another solution, and not by how much it is better, this approach allows only a very coarse guidance and is difficult to control.

Jin and Sendhoff also propose a way to convert fuzzy preferences into weight intervals, and then use their dynamic weighted aggregation EA to obtain the corresponding solutions[14]. This approach converts the multi-objective optimization problem into a single objective optimization problem by weighted aggregation, but varies the weights dynamically during the optimization run within the relevant boundaries.

The guided dominance and biased sharing approaches will be discussed in more detail in the following section.

3 Guidance and Biased Crowding Distance in NSGA-II

In this section, we describe in more detail two approaches that make it possible to take into account vague user preferences. Both approaches have been integrated into NSGA-II [10] as basic evolutionary multi-objective optimizer. NSGA-II is one of today's most prominent and most successful EMO algorithms. It is based on two principles: convergence to the Pareto-optimal front is ensured by non-dominated sorting. This method ranks individuals by iteratively determining the non-dominated solutions in the population (non-dominated front), assigning those individuals the next best rank and removing them from the population. Diversity within one rank is maintained by favoring individuals with a large crowding distance, which is defined as the sum of distances between a solution's neighbors on either side in each dimension of the objective space.

Note that for the results reported in this chapter, we used a slightly different notion of crowding distance: for a particular solution i, we use the average over the k Euclidean distances from solution i to its k nearest neighbors as crowding distance. This method should lead to a more even distribution of solutions in higher dimensions, when the neighboring solutions in one dimension (as used for the original crowding distance) can actually be located quite far away from the individual under consideration (with respect to all dimensions).

In all simulation runs reported below, we have chosen $k = 5$ in the case of three objectives. Although the same distance measure can be used for two-objective problems, here we simply use d_i as the Euclidean distance between the left and right neighboring solutions in the objective space.

3.1 The Guided Dominance Principle

In the Guided Multi-Objective Evolutionary Algorithm (G-MOEA) proposed by Branke et al. [1], user preferences are taken into account by modifying the definition of dominance.

The approach allows the DM to specify, for each pair of objectives, maximally acceptable trade-offs. For example, in the case of two objectives, the DM could define that an improvement by one unit in objective f_2 is worth a degradation of objective f_1 by at most a_{12} units. Similarly, a gain in objective f_1 by one unit is worth at most a_{21} units of objective f_2.

This information is then used to modify the dominance scheme as follows:

$$\begin{aligned} x \succ y \Leftrightarrow & (f_1(x) + a_{12} f_2(x) \leq f_1(y) + a_{12} f_2(y)) \wedge \\ & (a_{21} f_1(x) + f_2(x) \leq a_{21} f_1(y) + f_2(y)) \end{aligned}$$

with inequality in at least one case.

Figure 2 visualizes the effect: when compared to the original dominance criterion, a particular solution now dominates a larger region. The slope of the borders of the dominated region correspond to the user-defined trade-offs. With this dominance scheme, only a part of the original Pareto-optimal front remains non-dominated. This region is bounded by the solutions where the trade-off functions are tangent to the Pareto-optimal front, see Figure 3. By choosing appropriate trade-off values, it is then possible to focus on any part of the convex Pareto-optimal front. However, since the approach implicitly assumes linear utility functions, it may not be possible to focus on all parts of a concave Pareto-optimal front. The original dominance criterion can be considered just as a special case of the guided dominance criterion by choosing $a_{12} = a_{21} = \text{inf}$.

Interestingly, in the case of two objectives, domination according to the guided dominance criterion corresponds just to the standard dominance principle together with a suitably transformed objective space, cf. Figure 4. It is sufficient to replace the original objectives with two auxilliary objectives Ω_1 and Ω_2 and use these together with the standard dominance principle, where

$$\begin{aligned} \Omega_1(x) &= f_1(x) + a_{12} f_2(x) \\ \Omega_2(x) &= a_{21} f_1(x) + f_2(x) \end{aligned}$$

Because the transformation is so simple, the guided dominance scheme can be easily incorporated into standard MOEAs based on dominance, and it does not change the complexity of the algorithm.

Although in principle, the guidance idea carries over to more than two objectives, the user will have to specify an increasing number of trade-offs ($\binom{2 \cdot n}{2}$) and the dominance calculations become more complex.

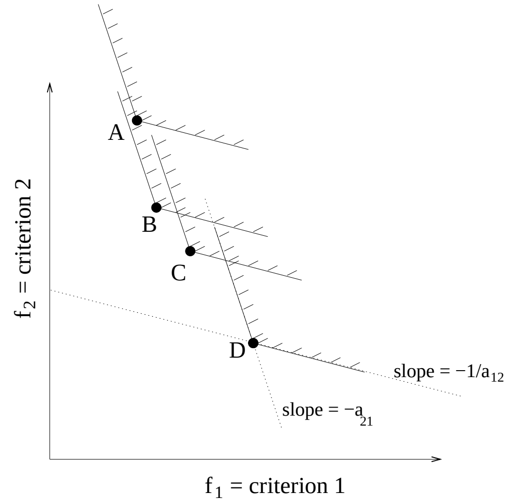

Fig. 2. Effect of the modified dominance scheme used by G-MOEA.

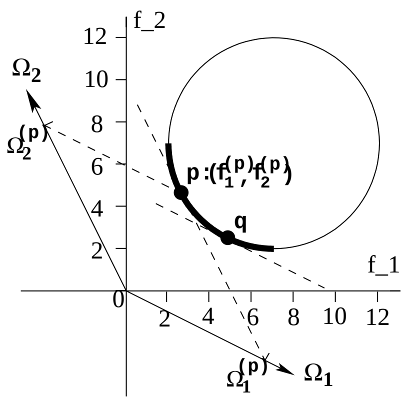

Fig. 3. When the guided dominance principle is used, non-dominated region of the Pareto-optimal front is bounded by the two solutions p and q where the trade-off functions are tangent.

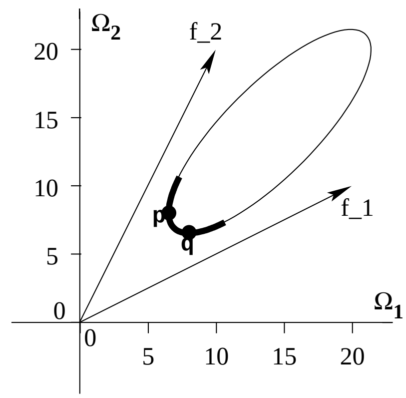

Fig. 4. The guided dominance principle is equivalent to the original dominance principle and appropriately transformed objective space.

3.2 Biased Crowding Distance in NSGA-II

In order to find a biased distribution anywhere on the Pareto-optimal front, a previous study by Deb [8] used a biased fitness sharing approach implemented on NSGA. In brief, the distance measure used for sharing was adapted to a weighted sum of distances in the different dimensions of the objective space.

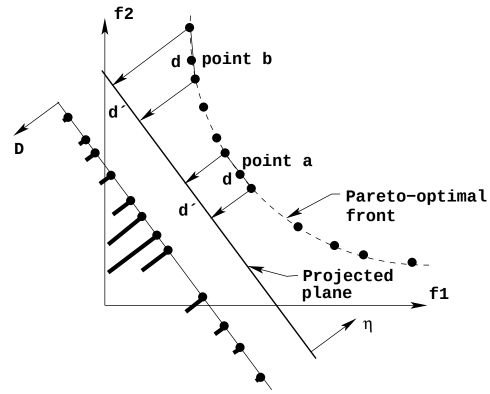

Fig. 5. The biased crowding approach is illustrated on a two-objective minimization problem.

By changing the weights, this allowed to produce a biased distribution of solutions along the Pareto-optimal front. However, while it was possible to focus on one objective or another, the approach did not allow to focus on a compromise region (for equal weighting of the objectives, the algorithm would produce no bias at all).

Here, we build on the idea of biased sharing and propose a biased crowding distance measure which is much more flexible and allows to control the region of interest and the expansion separately.

For a solution i on a particular front, we define a *biased crowding measure* D_i as follows: Let η be a user-specified direction vector indicating the most probable, or central linearly weighted utility function, and let α be a parameter controlling the bias intensity. Then,

$$D_i = d_i \left(\frac{d_i'}{d_i} \right)^{\alpha}, \tag{1}$$

where d_i and d_i' are the original crowding distance and the crowding distance calculated based on the locations of the individuals projected onto the (hyper)plane with direction vector η. Figure 5 illustrates the concept.

As a result, for a solution in a region of the Pareto-optimal front more or less parallel to the projected plane (such as solution 'a'), the original crowded distance d_a and projected crowding distance d_a' are more or less the same, thereby making the ratio d_a'/d_a close to one. Thus, according to equation 1, such a solution will have a biased crowding distance D_i almost the same as that in the original objective space (d_a). On the other hand, for a solution having a large difference in slope on the Pareto-optimal front from the chosen plane (such as solution 'b'), the projected crowding distance d_b' is much smaller than the original crowding distance d_b, thereby making the ratio d_b'/d_b a small number. For such a solution, the biased crowding distance value D_i will be a small quantity, meaning that such a solution is assumed to be artificially crowded by neighboring solutions. Figure 5 also shows the biased crowding distance values for all non-dominated solutions and how they would typically be distributed for a chosen plane and front. A preference of solutions having a larger biased crowding distance D_i will then enable solutions closer to the tangent point to be found. It is now clear that the exponent α will control the extent of obtained solutions in a simulation run. If a large α is chosen, solutions with a large deviation in slope from the chosen plane will have a small biased crowding distance and therefore the obtained extent of solutions would be less.

As we will show, the new biased crowding approach works well also on non-convex Pareto-optimal fronts and it easily scales in the number of objectives, two major advantages over the guidance scheme.

Note that biased crowding will focus on the area of the Pareto-optimal front which is parallel to the iso-utility function defined by the provided direction vector η. For a *convex* Pareto-optimal front, that is just the area around the optimal solution regarding a corresponding aggregate cost function. For a concave region, such an aggregate cost function would always prefer one of the edge points, while biased crowding may focus on the area in between.

4 Simulation Results

In this section, we empirically evaluate and compare the two methods described in the previous section. However, since the guided dominance scheme has already been explored in [1], our experiments will focus on the biased crowding distance scheme. First, we present results on two-objective ZDT problems. Thereafter, we shall present the obtained solutions using the biased crowding approach on three-objective problems.

For each problem, we use NSGA-II with a population size of 100. Since all problems involve real-parameter variables, we have used the simulated binary crossover (SBX) [9] and the polynomial mutation operator [11]. We have used a crossover probability of 1.0 and a variable-wise mutation probability of $1/n$, where n is the number of variables. The distribution index for SBX is 10 and that for mutation operator is 20. Each problem is run for 200 generations to investigate whether the methods are able to maintain the obtained distribution for a large enough number of generations.

4.1 ZDT1 Problem

The ZDT1 problem [16] has 30 real-parameter variables each varying in $[0,1]$. The resulting Pareto-optimal front is convex spanning in $f_1, f_2 \in [0,1]$. First, we use $\alpha = 100$ with a direction vector $\eta = (1,1)^T$. Figure 6 shows that the solutions are now biased around the point where the chosen plane is tangent

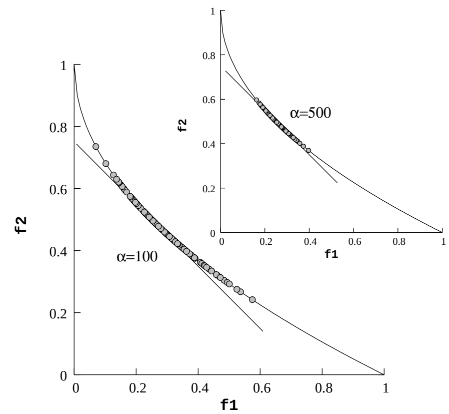

Fig. 6. NSGA-II populations with two exponent values and with $\eta = (1,1)^T$ for ZDT1.

to the Pareto-optimal front. It is interesting to note how the distribution of solutions gets sparse away from the tangent point. Since the solutions away from this tangent point produce a small ratio of d'/d, an exponent of $\alpha = 100$ to this ratio makes the distant solutions closer according to equation 1. Since they have a small crowding distance, NSGA-II does not motivate these distant solutions to remain in the population.

To investigate the effect of the exponent, α, we increase α to 500. Figure 6 also shows the corresponding distribution with otherwise identical parameter settings. Like before, only a portion of the Pareto-optimal front is found, but the extent of obtained solutions is smaller. The effect of increasing α is to intensify the effect of small (d'/d) ratios. Therefore, by simply changing the α value, the range of obtained solutions can be controlled.

Next, we use $\eta = (1,0)^T$ and $\eta = (0,1)^T$ in two different applications of NSGA-II (with all other parameters same as before) and the obtained distribution is shown in Figure 7. Since $\eta = (1,0)^T$ would capture Pareto-optimal solutions having an almost vertical slope on the front, solutions with small f_1 are found for ZDT1 in this case. An opposite effect is observed with $\eta = (0,1)^T$. Once again, the extent of obtained solutions can be controlled by changing the α value in these applications as well.

Finally, Figure 8 shows the obtained distribution with $\alpha = 0$. Since $\alpha = 0$ causes no biasing towards any particular region of the Pareto-optimal front (or, $D_i = d_i$), the figure shows that a good distribution of solutions on the entire Pareto-optimal front is obtained in this case.

We now use the guided dominance concept with NSGA-II to solve ZDT1 and demonstrate that this method is also capable to focus on a portion of the Pareto-optimal front, depending on user preferences. We keep the same NSGA-II parameter settings as above. First, we use the following two slopes:

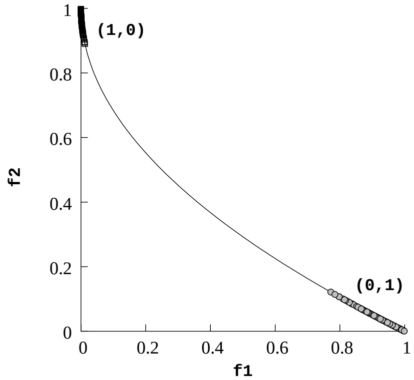

Fig. 7. Two applications of NSGA-II with $\eta = (1,0)$ and $\eta = (0,1)$. $\alpha = 100$ is used.

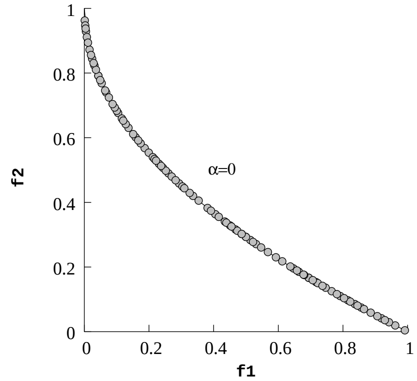

Fig. 8. NSGA-II population with $\alpha = 0$ finds the complete front.

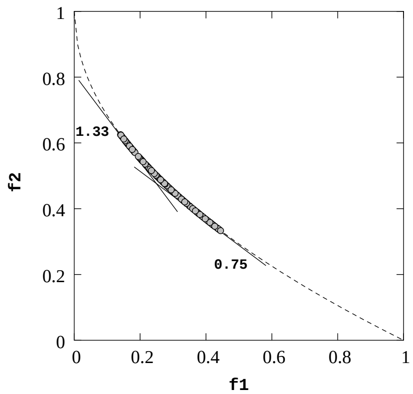

Fig. 9. NSGA-II with guided-dominance principle focuses on a part of the Pareto-optimal front on ZDT1 problem.

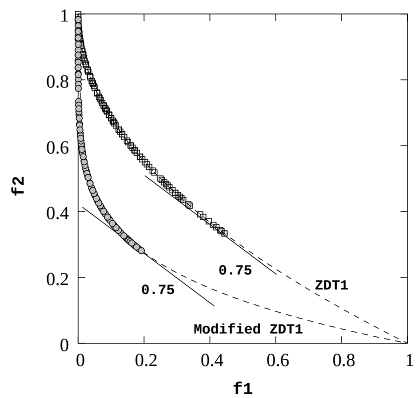

Fig. 10. NSGA-II population with another set of guided slopes applied to ZDT1 and a modified ZDT1 problem.

$a_{12} = 1.33$ and $a_{21} = 1.33$. Figure 9 shows that an intermediate portion of the Pareto-optimal front is found by the NSGA-II. As expected, NSGA-II with the guided dominance idea finds exactly the region bounded by the two points having a tangent to the Pareto-optimal front equal to $-1/a_{12}$ and $-a_{21}$, respectively. For clarity, these two tangents are also shown on the plot.

With $a_{12} = 1.33$ and $a_{21} = \infty$, we obtain the solutions shown in Figure 10. Once again, NSGA-II is able to find the portion of the complete Pareto-optimal front which is bounded by two tangents defined by $-1/a_{12}$ and $-a_{21}$.

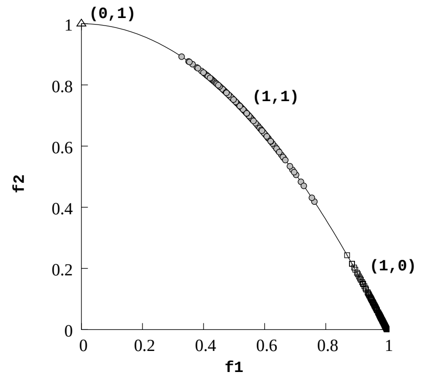

Fig. 11. NSGA-II with three different planes find corresponding portion of the Pareto-optimal front on ZDT2 problem. $\alpha = 100$ is used here.

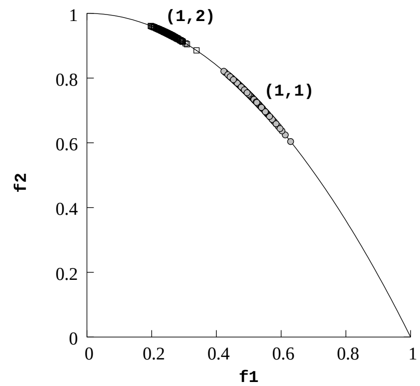

Fig. 12. NSGA-II populations for two planes with $\alpha = 500$ shows unequal extent of solutions.

This figure also shows another run of NSGA-II on a modified ZDT1 problem in which the Pareto-optimal front is simply changed by using a different h function [5]: $h(f_1, g) = 1 - (f_1/g)^{0.2}$. With identical guided slopes as used in the ZDT1 problem, NSGA-II finds a smaller extent of solutions (on f_1) here. Since the extent of solutions are determined by the chosen guided slopes (a_{12} and a_{21} here), it depends on the shape of the Pareto-optimal front.

4.2 ZDT2 Problem

The ZDT2 problem also has 30 variables, but the resulting Pareto-optimal front is non-convex. Figure 11 shows the obtained solutions in three different NSGA-II runs with $\eta = (1, 0)^T$, $(1, 1)^T$, and $(0, 1)^T$, respectively. In ZDT2, $\eta = (1, 0)^T$ causes the solutions with small f_2 to make larger d'/d values than solutions with small f_1 values. Thus, an opposite effect to that found in ZDT1 is observed here. This is due to the difference in shapes of the Pareto-optimal fronts in ZDT1 and ZDT2 problems. With $\eta = (1, 1)^T$, the intermediate solutions are preferred. However, for $\eta = (0, 1)^T$ only one solution (best f_1) is found. The change of slope around this solution is quite large, making the d'/d ratio to be small enough to not find any other neighboring solution in this case.

The simulation results, so far, demonstrate that although the extent of obtained solutions can be controlled with the α parameter, at different parts of the Pareto-optimal front the extent of solutions can be different for the same α value. The extent of solutions depend on the local curvature at different locations on the Pareto-optimal front. To continue to show the dependency of the extent on the location of the Pareto-optimal front, we rerun NSGA-II on

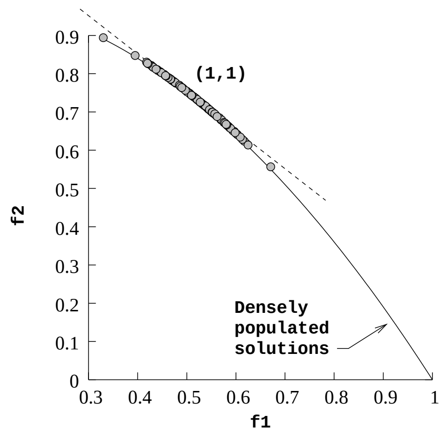

Fig. 13. NSGA-II population with biased crowding approach ($\alpha = 100$) on ZDT6 problem.

the same ZDT2 problem with $\alpha = 500$ and having $\eta = (1,1)^T$ and $\eta = (1,2)^T$ and present the results in Figure 12. Although crowding is established at different places on the Pareto-optimal front as they should, their expansions are different. Near the center of distribution with $\eta = (1,2)^T$, the ZDT2 has a sharper change in slopes, thereby making the extent of distribution smaller there.

4.3 ZDT6 Problem

The ZDT6 problem has 10 variables with a non-convex Pareto-optimal front. There is another difficulty in this problem. The density of solutions towards the Pareto-optimal front is thin and also differs across the front. The maximum density along the front occurs at the minimum f_2 value (at $f_2 = 0$) and reduces with increasing f_2. With $\eta = (1,1)^T$ and $\alpha = 100$, Figure 13 shows that although the corresponding focal point has a sparse distribution of solutions, NSGA-II with biased crowding can still find a biased distribution of solutions away from the area with densely crowded solutions. The center of the distribution is found to be near a point where the chosen plane is tangent to the Pareto-optimal front.

4.4 DTLZ2 Problem

The DTLZ2 problem [12] involves three objective functions defined with 12 variables. The resulting Pareto-optimal front is non-convex. Figure 14 shows the distribution of solutions obtained using $\alpha = 0$. Although not the best distribution, the figure clearly shows that all 100 solutions are spaced on

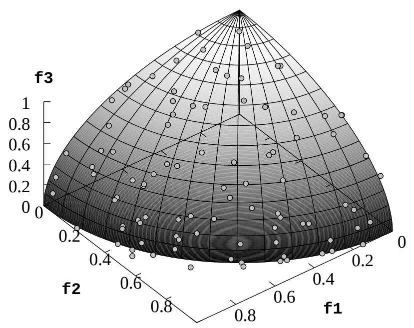

Fig. 14. NSGA-II population with $\alpha = 0$ is found to cover the entire Pareto-optimal front.

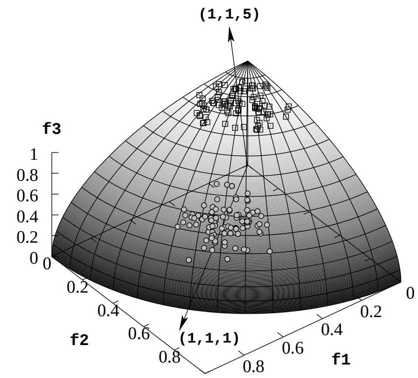

Fig. 15. NSGA-II populations with biased crowding approach ($\alpha = 500$) on DTLZ2 problem for two different projected planes.

the entire three-dimensional spherical (an octant of a sphere) Pareto-optimal front. However, when we use $\alpha = 500$ with $\eta = (1, 1, 1)^T$, the solutions are found only around the point where a plane with the above direction vector becomes tangent to the Pareto-optimal front (points marked with a circle in Figure 15). Once again, a set of solutions near individual best objective values produce a smaller projection on to the plane, thereby resulting in a small d'/d value. This causes the solutions to have a smaller crowding distance between each other, thereby making them unimportant solutions to be retained. Interestingly, this study shows that the biased crowding approach is able to make a biased distribution of solutions in a preferred region on the Pareto-optimal front even on a three-objective, non-convex problem.

When a different $\eta = (1, 1, 5)^T$ is chosen, a crowded set of solutions at a different location is obtained. The points marked with a box in Figure 15 show that they are located near a point which makes the chosen plane tangent to the true Pareto-optimal front.

4.5 DTLZ4 Problem

The DTLZ4 problem has an identical Pareto-optimal front, but the density of solutions is different from one location to another. In Figure 16, the solutions closer to the bold edges of the Pareto-optimal front are maximally densed with solutions. The density of solutions reduce away from both of these edges. To investigate the effect of the biased crowding, we apply NSGA-II (with $\alpha = 500$) with two different planes in two simulation runs: $\eta = (1, 1, 1)^T$ (solutions are marked with a box) and $\eta = (1, 2, 4)^T$ (solutions are marked with a circle). In each case, solutions are crowded near where these planes

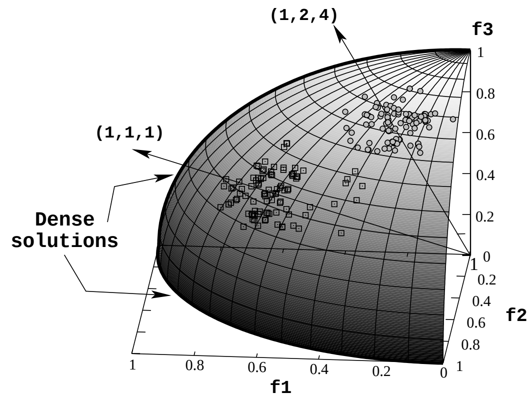

Fig. 16. NSGA-II populations with biased crowding approach ($\alpha = 500$) on DTLZ4 problem for two different projected planes.

make a tangent to the Pareto-optimal front. Despite the adverse density of solutions at these locations, NSGA-II with the biased crowding approach is able to find a biased population of solutions near the preferred regions.

4.6 Faster Convergence

In the previous sections, we have demonstrated that the proposed methods are capable of biasing the search according to the user's preferences. Now we look at the convergence speed of the algorithms. As has been noted already in [1], focusing the search on a part of the search region should not only result in a larger number of relevant solutions, but also allow a faster convergence.

To test this hypothesis, we run the guided dominance scheme as well as the biased crowding scheme again on ZDT1, comparing their convergence with the standard, non-biased scheme. Both approaches were set to focus on the same region of the Pareto-optimal front, with $a_{12} = a_{21} = 1.33$ for the guided dominance scheme and $\eta = (1, 1)$ and $\alpha = 500$ for the biased crowding distance scheme.

Convergence was measured by the individuals' distances to the Pareto optimal front. That distance was calculated as follows: first, we generated 50,000 equally spaced points on the true Pareto front (which is known for that problem). Then, for each non-dominated individual i in the population, we calculated the distance δ_i to the nearest of the one million points on the Pareto optimal front. The convergence measure at a certain generation is then the average distance of all non-dominated individuals.

Figure 17 shows the resulting convergence measure over time, averaged over 10 independent runs. As can be seen, the guided scheme indeed converges much faster than the non-biased NSGA-II, while the biased crowding approach seems only slightly faster.

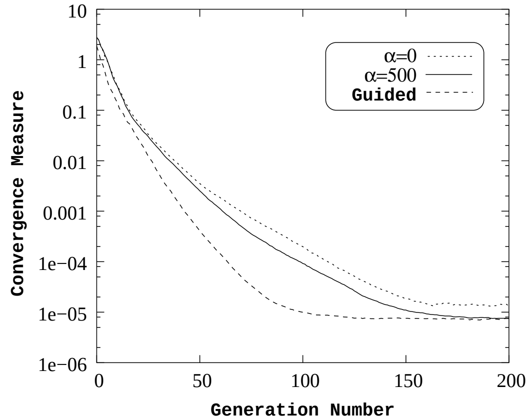

Fig. 17. Convergence of the different algorithms over time on ZDT1.

5 Summary and Conclusion

Many real world optimization problems have to consider several objectives simultaneously. So far, most literature on evolutionary multi-objective optimization is based on the principle of Pareto-optimality, which basically assumes that no knowledge about user preferences is available. As a consequence, it is attempted to find all Pareto-optimal solutions. Only after this large and diverse set of alternatives has been created, the decision maker (DM) specifies his/her preferences by choosing a particular solution.

Table 1. Comparison of the guided dominance and biased crowding schemes.

	Guided dominance	Biased crowding
DM preference specification	maximum and minimum trade-offs for each pair of objectives	Target tangent, focus strength parameter
Non-convex Pareto-optimal front	Focus on extreme solutions	Focus on inner region possible
Distribution of solutions	Even in relevant region, sharp boundaries	Biased towards target
Convergence compared to standard	Significantly faster	Slightly faster

We have assumed that in most applications, the DM has at least a vague idea about reasonable trade-offs between these objectives. We have presented two methods to extend standard EMO to take such vague user preferences into account: the guided dominance scheme and the biased crowding scheme. As has been demonstrated, both methods are simple and effective in focusing the search on relevant regions of the search space. We have shown that when the proposed methods are used, the EMO-algorithm is able to not only find a better selection of relevant solutions, but to find those also more quickly.

Which of the two methods is more suitable depends on the particular application. While the new biased crowding scheme is easier to use with a larger number of objectives, the guided dominance scheme allows a faster convergence and a more precise control over the focus region. Table 1 again summarizes the most important characteristics of the two methods compared in this chapter.

As future work, it seems particularly promising to integrate the two methods into an interactive framework, which allows to modify and increase the search focus during the optimization process.

References

1. J. Branke, T. Kaußler, and H. Schmeck. Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32:499–507, 2001.
2. C. A. Coello Coello. Handling preferences in evolutionary multiobjective optimization: A survey. In *Congress on Evolutionary Computation*, volume 1, pages 30–37. IEEE, 2000.
3. Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer, 2002.
4. Dragan Cvetkovic and Ian C. Parmee. Preferences and their application in evolutionary multiobjective optimisation. *IEEE Transactions on Evolutionary Computation*, 6(1):42–57, 2002.
5. K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation Journal*, 7(3):205–230, 1999.
6. K. Deb. Solving goal programming problems using multi-objective genetic algorithms. In *Proceedings of Congress on Evolutionary Computation*, pages 77–84, 1999.
7. K. Deb. *Multi-objective optimization using evolutionary algorithms.* Chichester, UK: Wiley, 2001.
8. K. Deb. Multi-objective evolutionary algorithms: Introducing bias among Pareto-optimal solutions. In A. Ghosh and S. Tsutsui, editors, *Advances in Evolutionary Computing: Theory and Applications*, pages 263–292. London: Springer-Verlag, 2003.
9. K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.
10. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
11. K. Deb and M. Goyal. A robust optimization procedure for mechanical component design based on genetic adaptive search. *Transactions of the ASME: Journal of Mechanical Design*, 120(2):162–164, 1998.
12. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC-2002)*, pages 825–830, 2002.

13. C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, 1993.
14. Yaochu Jin, Tatsuya Okabe, and Bernhard Sendhoff. Adapting weighted aggregation for multiobjective evolution strategies. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *Evolutionary Multi-criterion Optimization*, number 1993 in LNCS, pages 96–110. Springer, 2001.
15. D. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation Journal*, 8(2):125–148, 2000.
16. Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation Journal*, 8(2):125–148, 2000.