Guidance in Evolutionary Multi-Objective Optimization

Jürgen Branke, Thomas Kaußler, Hartmut Schmeck
Institute AIFB, University of Karlsruhe
D-76128 Karlsruhe, Germany
Email:{branke|schmeck}@aifb.uni-karlsruhe.de

Abstract

Many real world design problems involve multiple, usually conflicting optimization criteria. Often, it is very difficult to weight the criteria exactly before alternatives are known. Multi-Objective Evolutionary Algorithms based on the principle of Pareto optimality are designed to explore the complete set of nondominated solutions, which then allows the user to choose among many alternatives. However, although it is very difficult to exactly define the weighting of different optimization criteria, usually the user has some notion as to what range of weightings might be reasonable. In this paper, we present a novel, simple, and intuitive way to integrate the user's preference into the evolutionary algorithm by allowing to define linear maximum and minimum trade-off functions. On a number of test problems we show that the proposed algorithm efficiently guides the population towards the interesting region, allowing a faster convergence and a better coverage of this area of the Pareto optimal front.

Keywords: evolutionary algorithm, multiple objectives, Pareto optimal, preferences, guidance

1 Introduction

Many real-world design problems involve multiple objectives that have to be optimized simultaneously. As these objectives are usually conflicting, no single solution may exist that is best regarding all criteria considered. Therefore, at some stage of the problem solving process, the decision maker (DM) has to articulate his/her preferences about the objectives. Following a classification by Veldhuizen [1], the articulation of preferences may be done either before (a priori), during (progressive) or after (a posteriori) the optimization process.

A priori approaches are usually not practicable, since they require the user to explicitly and exactly weight the different criteria and to effectively turn the multi-objective optimization problem into a single objective problem prior to knowing possible alternatives.

A posteriori approaches may use an evolutionary algorithm (EA) to produce as many different Pareto-optimal[1] solutions as possible, and then allow the DM to choose among them.

With progressive approaches, decision making and optimization are intertwined. An example for such an approach is given in [2], where an Artificial Neural Network is used to learn the user's preferences during the course of the optimization.

In general, a posteriori approaches are most convenient, since they do not require the user to interact with the process, and the DM can delay any choice until the alternatives are known.

Almost all such approaches employ two basic concepts:

1. The use of dominance/non-dominance for evaluating solutions
2. A diversity mechanism to ensure an equally-spaced coverage of the Pareto-optimal front.

For minimization problems, solution x is said to *dominate* solution y ($x \succ y$) when the following condition is satisfied:

[1] A solution is called Pareto optimal if, when compared to any other solution, it is superior in at least one criterion.

$$x \succ y \quad \Leftrightarrow \quad \begin{array}{l} f_i(x) \leq f_i(y) \text{ for all criteria } i \text{ and} \\ \exists j : f_j(x) < f_j(y) \end{array}$$

A solution is called *nondominated* with respect to a set of solutions when it is not dominated by any other solution in that set. If it is not dominated by any other feasible solution of the problem, it is also called *Pareto-optimal*. For example in Figure 1, solutions A and B are non-dominated, while C is dominated by both, A and B. As can be seen, for a two-dimensional minimization problem, each solution dominates the area to its upper right.

To maintain diversity in the population, often *sharing* is used. This mechanism aims at spreading out the population by decreasing an individual‘s fitness when it has many close neighbors. That way, solutions in less covered areas are relatively favored over solutions in crowded areas. For more details see e.g. [3].

Unfortunately, searching for all Pareto optimal solutions is a difficult and time-consuming process. Besides, generally the DM has some intuition about what trade-offs might be sensible, and this knowledge should be used during the search. For example, in the domain of project scheduling, the conflicting criteria for a schedule might be cost and completion time. Finishing the project one day earlier could be valued between \$500 and \$800 by the DM. Another solution, speeding up the project another day at the expense of several million dollars might be Pareto optimal, but totally unacceptable from the perspective of the DM. This shows that in reality, a small improvement in one criterion is worth only limited deterioration of the other criterion, and such limits can usually be specified easily.

Recently, two approaches have been published that aim at focusing the search towards interesting regions as defined by the DM. Deb [4] uses an analogy from goal programming and allows the DM to define a goal (a single desired combination of characteristics) towards which the search is directed. Cvetkovic and Parmee [5, 6] allow the DM to articulate fuzzy preferences, like “criterion 1 is much more important than criterion 2”. Both approaches will be discussed in more detail in Section 2.

With the Guided Multi-Objective Evolutionary Algorithm (G-MOEA) first proposed in [7] and examined more closely in this paper, we allow the DM to specify maximal and minimal acceptable weightings for one criterion over the other, and use these to efficiently guide the EA towards Pareto-optimal solutions within these boundaries.

As we will show, this not only speeds up convergence, but also leads to a better coverage of the interesting part of the Pareto optimal front. Also, the DM will know exactly what kind of solutions he/she can expect, the usage of the approach is very intuitive.

The paper's outline is as follows: first, we will review related work in Section 2. Then, in Section 3, we will explain our algorithm in detail. The approach is tested on a number of common benchmark problems and compared to other approaches in Section 4. The paper concludes with a summary and suggestions for future work.

2 Related Work

As already mentioned in the introduction, there have recently been published two other approaches to restrict the search to an “interesting” subset of the Pareto-optimal solutions.

The method by Cvetkovic and Parmee [5, 6] assigns each criterion a weight w_i, and a minimum level for dominance, τ. Then, the concept of dominance is defined as follows:

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

with inequality in at least one case. To facilitate specification of the required weights, in [6], a method is suggested to turn fuzzy preferences into specific quantitative weightings. 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 allows only a very coarse guidance and is difficult to control.

The approach by Deb [4] uses an analogy from goal programming. There, the DM can specify a goal in terms of a single desired combination of characteristics $\vec{t} = (t_1, \ldots, t_n)$, and the type of goal (e.g. $f(x) \leq t, f(x) \geq t, f(x) = t$). Deb [4] shows how these can be modified to suit MOEAs. Basically, the distances from that goal rather than the actual criteria are compared. If the goal for criterion i is to find a solution x with $f_i(x) \leq t_i$, then instead of considering the criterion $f_i(x)$, simply $f_i'(x) = \max\{0, f_i(x) - t_i\}$ is used. In other words, a solution is not rewarded if it is better than the goal. This is illustrated in Figure 2. As can be seen, the area dominated by an individual is not changed as long as that individual did not reach the goal in any criterion (solutions B or C). Individuals A or D on the other hand are now dominating a whole half-plane. Subsequently, the set of Pareto optimal solutions is basically restricted to the area dominated by the goal (dashed lines in Figure 2), with at most one Pareto-optimal solution lying outside this region on either side.

As can be seen, if the goal is set appropriately, this approach may indeed restrict the search space to an interesting region. The problem here is to set the goal a priori, i.e. before the Pareto optimal front is known. Therefore, the effect of a goal on the search process is difficult to assess. For example, if a goal is set that lies within the feasible region, it prevents the search for even better solutions, if it lies too far away from the feasible region, it has basically no effect at all. The Guided Multi-Objective Evolutionary Algorithm proposed in the next section will be much easier to handle since it does not require to know the location of the pareto optimal front in order to be used effectively.

Another interesting method has been published in [8]. There, not the selection step but the sharing method is altered such that the density of solutions produced by the EA is higher in "interesting" regions, and lower elsewhere. The interesting regions have to be specified by weighting the different criteria. Since with this method usually the whole Pareto-front is outlined, only with different density, it may give the decision maker some idea as to how the Pareto-optimal front looks like, even away from the region specified as "interesting". Unfortunately, the method requires to know whether the Pareto-optimal front is convex or not, and does not allow to focus on the area with similar weight on all criteria (for that setting, the whole Pareto front would be covered equally, just as in the standard approach).

3 Guided Multi-Objective Evolutionary Algorithm

The basic idea of our Guided Multi-Objective Evolutionary Algorithm (G-MOEA) is that although the DM is unable to define his/her preferences in the form of an exact utility function respectively weighting between the different criteria, he/she usually has some idea about reasonable trade-offs. This knowledge should be used in order to guide the search towards the interesting area.

Note that in this article we are only considering the case of two objectives and linear trade-offs specified by the DM. Possible extensions will be discussed at the end of this section.

So far, let us assume there are two optimization criteria $f_1(x)$ and $f_2(x)$. Then, the DM's intuition has to be specified in terms of maximal and minimal linear utility functions $w_{min}(f_1(x), f_2(x))$ and $w_{max}(f_1(x), f_2(x))$. Setting w_{min} or w_{max} constant defines the boundary for the points that might be equally valuable to the user.

The maximum and minimum utility functions can be easily constructed by asking how many units of f_2 the DM would be willing to trade-off against one unit of f_1 at most or at the least, respectively.

Assume the DM decides that one unit of f_1 is worth at most a units of f_2, i.e. $af_1(x) + f_2(x) =$ const. This corresponds to a straight line on the $(f_1(x), f_2(x))$-plane with slope of $\frac{f_2(x)}{f_1(x)} = -a$. We call this ratio also the maximum (or minimum, for w_{min}) weighting. The maximal utility function is just $w_{max}(f_1(x), f_2(x)) = af_1(x) + f_2(x)$.

The minimum utility function can be derived correspondingly

We believe that in most cases, the decision maker should be able specify these tradeoffs easily[2]. To give another example: if for the design of an engine, production cost and energy efficiency are the conflicting goals, the DM would only have to specify how much he or she would be willing to pay at most for an extra percent of efficiency, and how much a percent of efficiency would be worth at the least.

The core idea of our approach is to use this information to adapt the dominance scheme accordingly:

$$\begin{aligned} x \succ y \quad \Leftrightarrow \quad & w_{max}(f_1(x), f_2(x)) \leq w_{max}(f_1(y), f_2(y)) \\ \wedge \quad & w_{min}(f_1(x), f_2(x)) \leq w_{min}(f_1(y), f_2(y)) \end{aligned}$$

with inequality in at least one case.

An example is given in Figure 3. While with the ordinary dominance scheme, all depicted solutions would be non-dominated, when the dominance scheme has been adapted according to G-MOEA, solution A is dominated by solution B. The slope of the borders of the dominated area correspond to the user-specified minimum and maximum trade-offs.

By using this new dominance scheme, our G-MOEA will concentrate on the interesting subset of the Pareto-optimal front, as we will show.

The guided nondomination ranking applied is basically a topological sorting according to our new dominance scheme. Except for the dominance scheme, it corresponds to the ranking method proposed in [9] and proceeds as follows: First, all nondominated (w.r.t. our modified dominance scheme) individuals are determined and are given the best rank (starting from 1). Then, all thus ranked individuals are removed from the population, and the process starts anew with the remaining individuals, this time the next best rank is awarded. This procedure is repeated iteratively until each individual has been assigned a rank.

Since the new nondomination ranking can be achieved by a mere transformation of the objective space, it can be performed as efficiently as the "standard" nondomination ranking, namely in $O(n \log n)$ time for two dimensions, using e.g. the sweepline method [10].

Note that with $w_{min}(x) = f_2(x)$ respectively $\frac{f_2(x)}{f_1(x)} = 0$ and $w_{max}(x) = f_1(x)$ respectively $\frac{f_2(x)}{f_1(x)} = -\infty$ the case of no a priori preference articulation by the DM can be handled. The Pareto analysis then corresponds to the normal nondomination ranking without guidance.

The guided nondomination analysis leads the population towards the true Pareto-optimal front in the, for the decision maker, interesting part of the search space. Diversity along the Pareto optimal front is supported by fitness sharing which takes into account *all* individuals and works on the objective space. As in the standard sharing procedure, for each individual x the niche count m_x is calculated, a quantity proportional to the number of other individuals the neighborhood of individual x [9]. Then, if r_x is the rank assigned to individual x, this rank is modified according to

$$r'_x = \begin{cases} r_x + \frac{m_x - \min\limits_{k=1\ldots n} \{m_k\}}{\max\limits_{k=1\ldots n} \{m_k\} - \min\limits_{k=1\ldots n} \{m_k\}} & : \max\limits_{k=1\ldots n} \{m_k\} > \min\limits_{k=1\ldots n} \{m_k\} \\ r_x & : \text{otherwise} \end{cases}$$

with n denoting the population size.

That way, individuals with comparatively few neighbors are relatively favored over individuals in crowded areas, however no individual can get a worse modified rank or fitness than any individual with an inferior original rank. It is thus ensured that the current Pareto front converges quickly to the true Pareto front while the individuals are nevertheless well distributed. The sharing based on the whole population as opposed to the Pareto level (as e.g. used in [8]) was chosen since the goal is not to maintain diversity in each Pareto level, but in the overall population: it should be prevented that different levels converge towards the same area of the Pareto-optimal front, while neglecting other areas.

[2] To support that specification process, one might adapt a method to transform imprecise and fuzzy preferences into concrete numbers as suggested by Cvetkovic and Parmee [6]

As noted earlier, in this article we are only considering the case of two objectives $(f_1(x), f_2(x))$ and linear trade-offs specified by the DM. A generalization to $m > 2$ objectives would be straightforward, using hyperplanes instead of lines as boundaries. However then trade-offs for every pair of objectives, i.e. $m^2 - m$ trade-offs would have to be specified. The use of linear maximal and minimal trade-off boundaries might sometimes limit the applicability of G-MOEA, namely when the DM's trade-off is strongly dependent not only on the relative position of the two solutions to be compared, but also on the absolute positions (i.e. when the shape of the area dominated by a solution depends on this solution's location on the $f_1(x), f_2(x)$ plane). We are currently exploring ways to extend the approach to non-linear trade-off functions as well, however we will stick to linearity for the remainder of this paper.

For the experiments reported below, standard settings were generational replacement with elitism of all Pareto-optimal solutions, tournament selection with tournament size 2, and the use of Simulated Binary Crossover as introduced by Deb and Beyer [11]. Further problem dependent settings are mentioned in the next section.

4 Experimental Results

In this section we will first examine the effect of the G-MOEA and the goal-programming dominance scheme on the ranking of solutions. Then, G-MOEA will be tested on four test problems that were already examined or introduced in related work.

Dominance Levels

In this section, we examine the effect of the dominance scheme on the shape of the dominance levels, i.e. the set of individuals assigned the same rank. For that purpose, we generated 400,000 evenly distributed random points in the criterion space bounded by a simple convex function [3] and ranked them according to the different dominance schemes. Although in a real population, the number of individuals would be much smaller, and they would probably not be evenly distributed, this still gives us a pretty good idea about which individuals are favored depending on the dominance scheme, and how the EA's population is driven towards the Pareto-optimal front.

Figure 4 depicts several sets of solutions (layers) that have been assigned the same rank based on the standard dominance scheme. For reasons of readability, only a few layers have been displayed.

The same analysis has been performed on the G-MOEA dominance scheme for an arbitrarily chosen set of maximum and minimum trade-off (Figure 5 (a)), and on the goal-programming approach (Figure 5 (b)).

As can be seen, while for the standard approach, each layer covers the whole front, goal-programming and G-MOEA focus on solutions in the "interesting" region.

Of those two, G-MOEA seems to adapt the ranking more gradually, starting in the interesting area and expanding from there, broadening the covered area when moving away from the front, but maintaining its general shape.

That way, the selection pressure smoothly guides the search towards the interesting region of the Pareto-optimal front, and we would assume that in most cases, the convergence properties of G-MOEA are better than those of the goal-programming approach, particularly for difficult problems with a strong focus on a narrow region.

Test Problem 1

First we examine the algorithm's convergence properties using Schaffer's f2 Function. This is a very simple test function introduced by Schaffer in his dissertation [12]. Nevertheless nearly every multi-objective EA is applied to it to test whether the population becomes and remains well distributed along the Pareto optimal front. In the tests reported here, no guiding was used (i.e. no transformation).

[3] Any function could have served as boundary. We actually used here the Pareto optimal front from the Test Problem 4 discussed later.

Parameter settings used were population size of 100, $\sigma_{share} = 0.1$, crossover probability of 1.0 and no mutation. The mutation probability is set to zero to observe the effectiveness of G-MOEA alone.

As the results of a typical run show, the population quickly converges to the Pareto front and distributes the individuals fairly evenly over the Pareto-optimal front. G-MOEA is moreover able to maintain the diversity in the long run (not shown here). These results prove that our implemented fitness sharing as described above works fine.

Test Problem 2

The effect of guidance is tested on Deb's convex test problem [13] with $\alpha = 0.3$ (function specific parameter). The EA used a population size of 100, $\sigma_{share} = 0.5$, crossover probability of 0.8 and mutation probability of 0.01. As minimum and maximum weighting, $\frac{f_2}{f_1} = -0.8$ respectively $\frac{f_2}{f_1} = -1.25$ are used.

As can be seen in Figure 7, G-MOEA is not only capable of guiding its search towards the desired region of the Pareto optimal front (area between the two corner individuals), but moreover converges much faster to the true Pareto optimal front.

Figure 8 zooms into the interesting region. One can clearly see that this region of the Pareto optimal front is much better explored by G-MOEA than by the unguided variant, in terms of convergence as well as in terms of diversity or numbers of individuals.

Some additional tests show the impact of the weighting preferences on the G-MOEA. The plot in Figure 9 has been produced using the minimum and maximum weighting $\frac{f_2}{f_1} = -1.0$ respectively $\frac{f_2}{f_1} = -3.0$, while in Figure 10, the weightings $\frac{f_2}{f_1} = -0.5$ respectively $\frac{f_2}{f_1} = 0$ were used.

As can be seen, G-MOEA adapts its focus accordingly.

Test Problem 3

Cvetkovic and Parmee [5] introduced a simple test function to test their preference based MOEA. Note that this problem is a maximization problem.

In the following tests, G-MOEA was run with population size of 50, crossover probability of 0.8, mutation probability of 0.02, and $\sigma_{share} = 0.5$ for 15 generations. Weightings used were $\frac{f_2}{f_1} = (-0.4, -1.0); (-0.8, -1.25); (-1.5, -\infty)$.

Figure 11 again shows how the focus of G-MOEA adapts to the DM's preferences as defined by the maximum and minimum trade-off functions. The results match those reported in Cvetkovic and Parmee [5].

Test Problem 4

Finally, we compared our approach to the goal programming inspired MOEA presented by Deb [4] on his test problem P1.

Parameter settings were population size of 50, crossover probability of 0.8, mutation probability of 0.02, and $\sigma_{share} = 0.5$ run for 50 generations. Weighting used was $\frac{f_2}{f_1} = (-0.4, -2.5)$. As goal for the approach suggested in [4], the point (2.0;2.0) was used.

The results from Figure 12 show that the G-MOEA is able to provide solutions that are matching the ones obtained with goal programming.

The two approaches differ mainly in the way the preferences are expressed. The suitability of one approach over the other may depend on the application. But generally, the goal programming approach is sensitive to a good choice of the goal, while in our approach, the DM can explicitly define the interesting range of trade-offs without any knowledge of the Pareto optimal front. Therefore, for many problems it may be easier for the DM to define his/her preferences by means of minimum and maximum trade-offs than by means of a goal.

5 Conclusion

This paper addresses the issue of focusing evolutionary multi-criteria search towards "interesting" parts of the Pareto-optimal front. After briefly surveying some recent approaches to this problem, we have presented a new way to include a decision maker's vague preferences of one criterion over the other into the optimization process of Multi-Objective Evolutionary Algorithms.

The proposed Guided Multi-Objective Evolutionary Algorithm (G-MOEA) provides the user with a simple, intuitive and powerful possibility of guiding the search process towards a specified region of the Pareto optimal front. With guidance, the algorithm not only covers the interesting region much better than the unguided version, but it also converges much faster towards the optimal front, which may save significant computational efforts.

Compared to recently presented approaches with similar intentions, G-MOEA is able to at least match their reported performance. At the same time, G-MOEA seems to be more flexible and more intuitive to use than the other approaches, as it allows to explicitly set the minimum and maximum acceptable trade-offs.

To further facilitate the guiding, we see two interesting possibilities: it should be considered whether it is possible to use an approach similar to the one in [6] to support the specification of minimum and maximum weighting functions, and our approach could be combined with an approach similar to [2] that learns the user's preferences during the optimization run.

We are currently testing this approach also on combinatorial problems, so far with equally encouraging results.

Also, as has been noted above, so far the approach is restricted to linear trade-off functions. However, an extension to more complicated trade-off functions should be possible and is currently explored.

Acknowledgements: We would like to thank the anonymous referees for their helpful comments.

References

[1] D. A. Van Veldhuizen and G. B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of the-art. *Evolutionary Computation*, 8(2):125–148, 2000.

[2] D. S. Todd and P. Sen. Directed multiple objective search of design spaces using genetic algorithms and neural networks. In Wolfgang Banzhaf et al., editor, *Genetic and Evolutionary Computation Conference*, pages 1738–1743. Morgan Kaufmann, San Francisco, California, 1999.

[3] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Second International Conference on Genetic Algorithms*, pages 41–49, 1987.

[4] K. Deb. Solving goal programming problems using multi-objective genetic algorithms. In *Congress on Evolutionary Computation*, volume 1, pages 77–84. IEEE, 1999.

[5] D. Cvetkovic and I. C. Parmee. Use of preferences for GA-based multi-objective optimization. In W. Banzhaf et al., editor, *Genetic and Evolutionary Computation Conference*, volume 2, pages 1504–1509. Morgan Kaufmann, 1999.

[6] D. Cvetkovic and I. C. Parmee. Designer's preferences an multi-objective preliminary design processes. In I. Parmee, editor, *to appear in: Adaptive Computing and Manufacture*. Springer, 2000.

[7] J. Branke, T. Kaußler, and H. Schmeck. Guiding multi objective evolutionary algorithms towards interesting regions. Technical Report TR No. 399, Institute AIFB, University of Karlsruhe, Germany, 2000.

[8] K. Deb. Multi-objective evolutionary algorithms: Introducing bias among pareto-optimal solutions. Technical Report KanGAL Report No. 99002, Kanpur Genetic Algorithms Laboratory, 1999.

[9] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[10] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[11] K. Deb and H.-G. Beyer. Self-adaptive genetic algorithms with simulated binary crossover. Technical Report CI-61/99, Department of Computer Science/XI, University of Dortmund, Germany, 1999.

[12] J. D. Schaffer. *Some experiments in machine learning using vector evaluated genetic algorithms*. PhD thesis, Vanderbilt University, 1984.

[13] K. Deb. Evolutionary algorithms for multi-criterion optimization in engineering design. In *Evolutionary Algorithms in Engineering and Computer Science*, 1999.

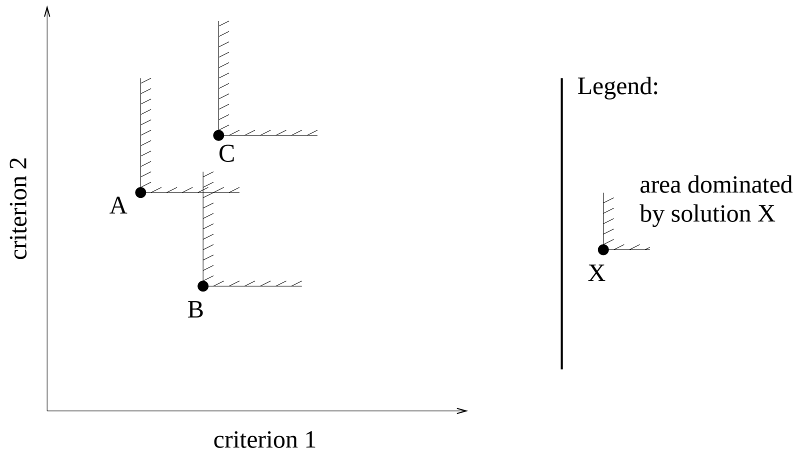


Figure 1: Illustration of the concept of dominance.

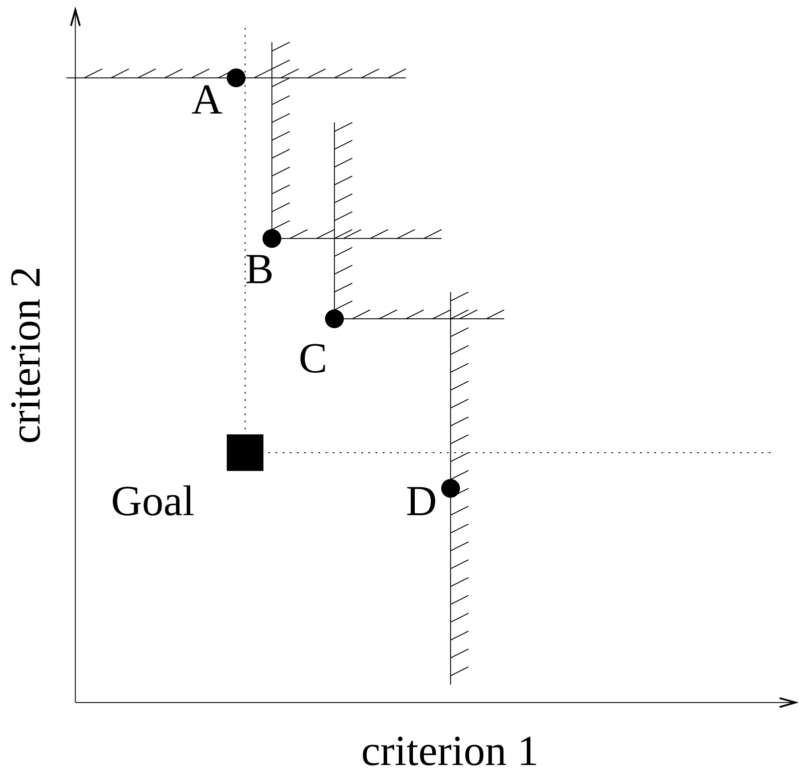


Figure 2: The dominance scheme as used in the goal-programming approach.

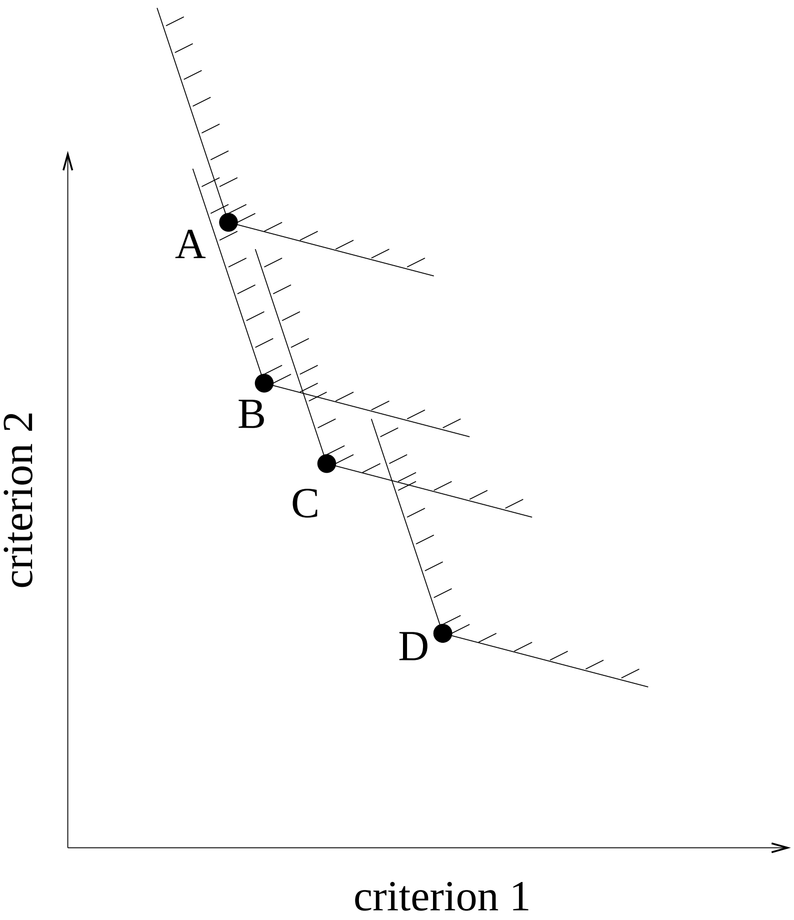


Figure 3: With G-MOEA, a different dominance scheme is used.

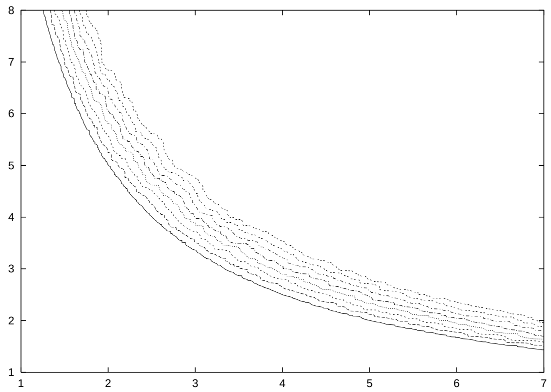
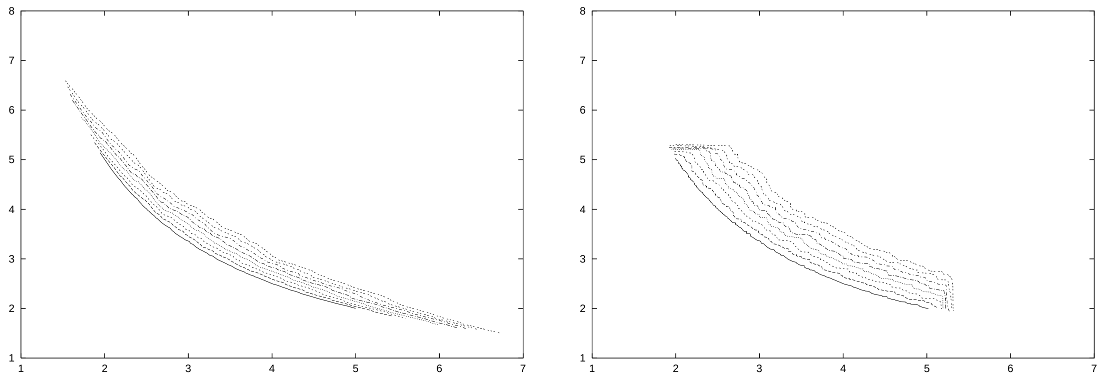
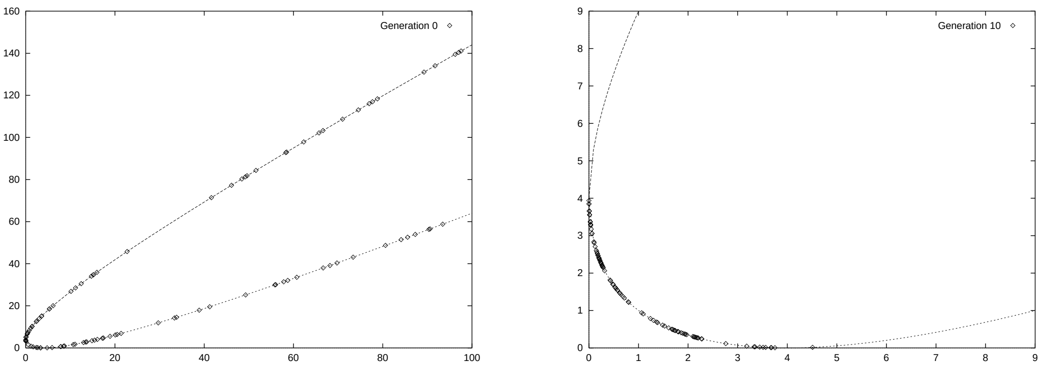


Figure 4: Ranking layers as produced by the standard dominance scheme



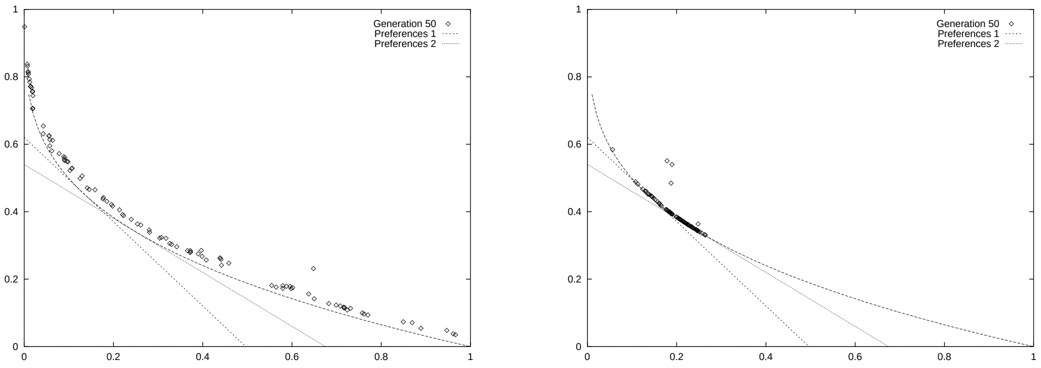
(a) (b)

Figure 5: Ranking layers as produced by the G-MOEA dominance scheme (a), and the goal-programming approach (b)



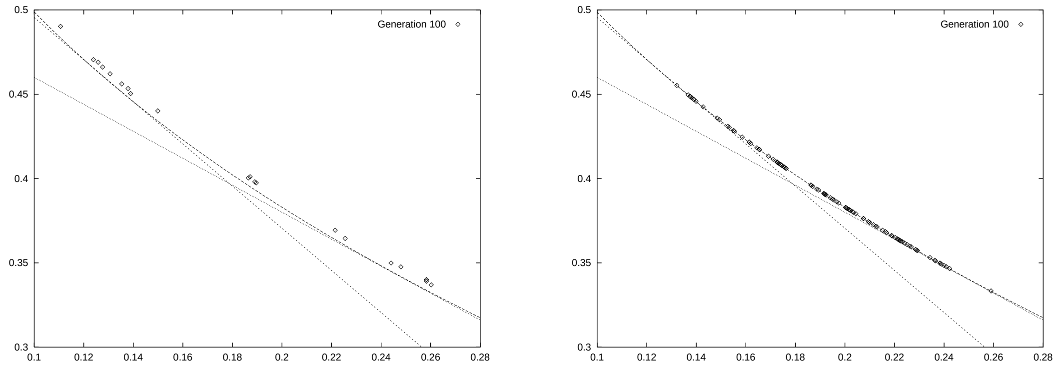
(a) (b)

Figure 6: Without any preferences, the G-MOEA converges quickly and evenly to the Pareto optimal front. (a) shows the initial population, (b) the situation after 10 generations.



(a) (b)

Figure 7: After 50 generations, G-MOEA (b) clearly outperforms the unguided version (a) in terms of convergence and coverage of the interesting area.



(a) (b)

Figure 8: When zooming into the interesting region, the better performance of G-MOEA (b) over the unguided version (a) is even more obvious. The plots show the population in generation 100.

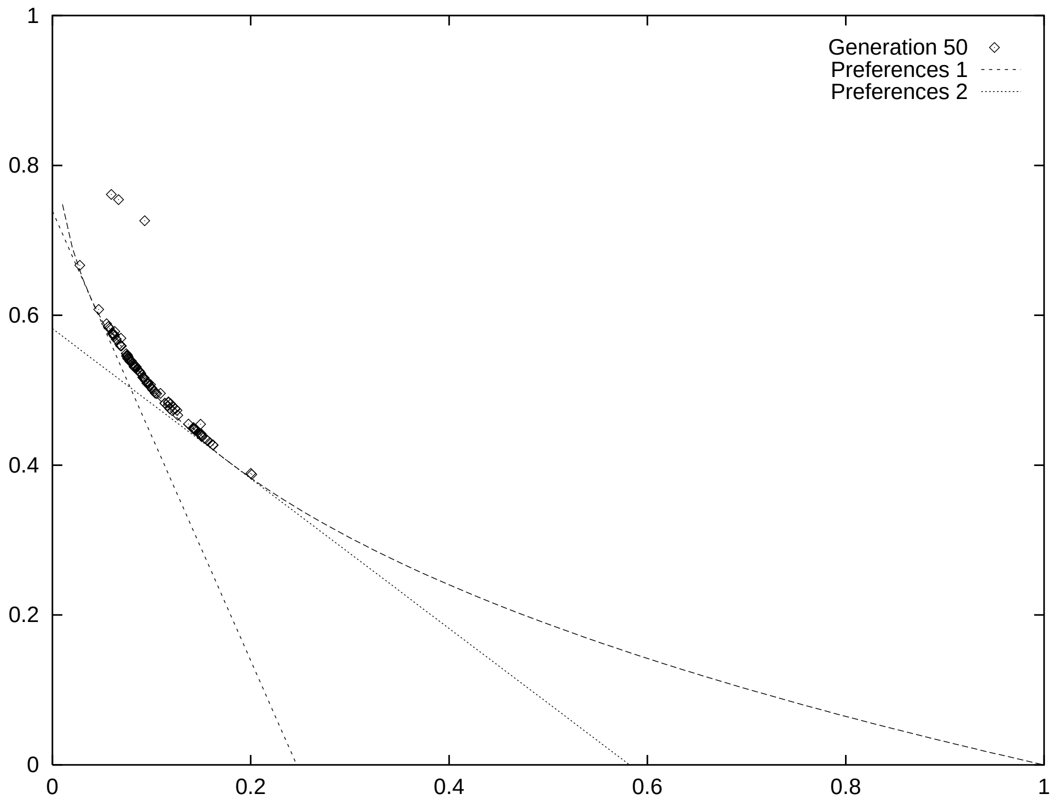


Figure 9: By stressing criterion 1 over criterion 2, the region of convergence is shifted accordingly.

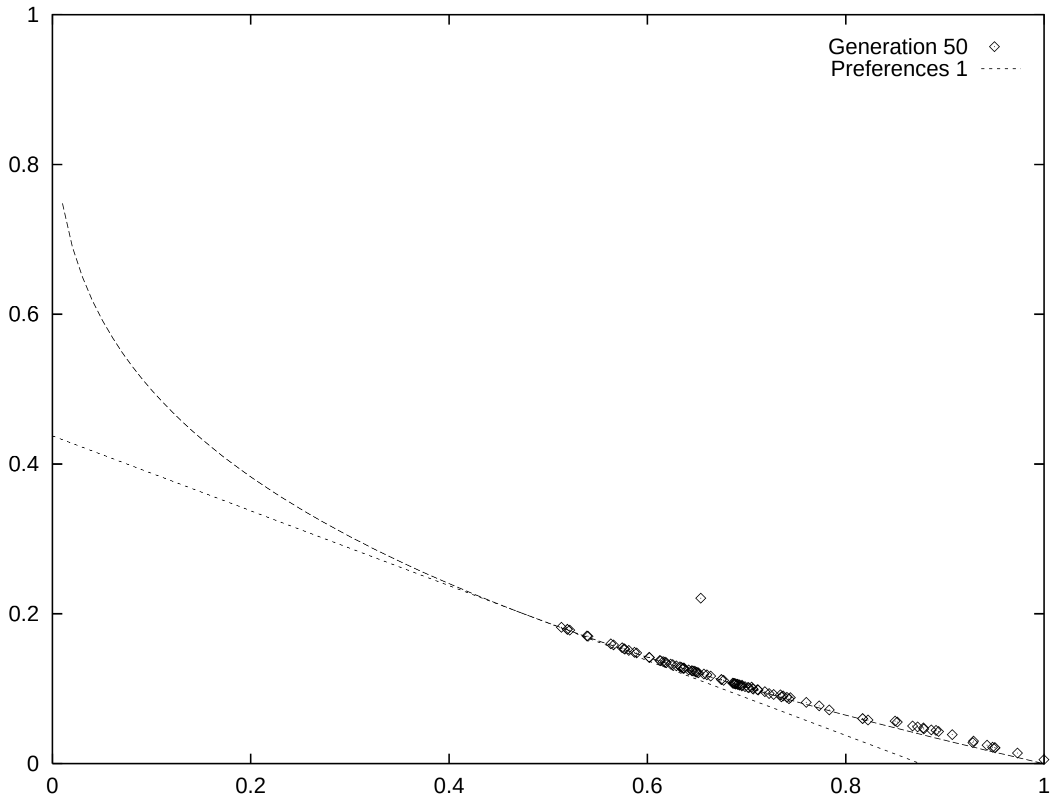
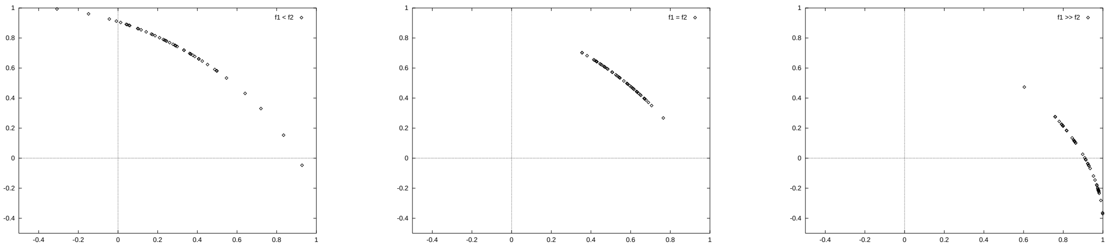
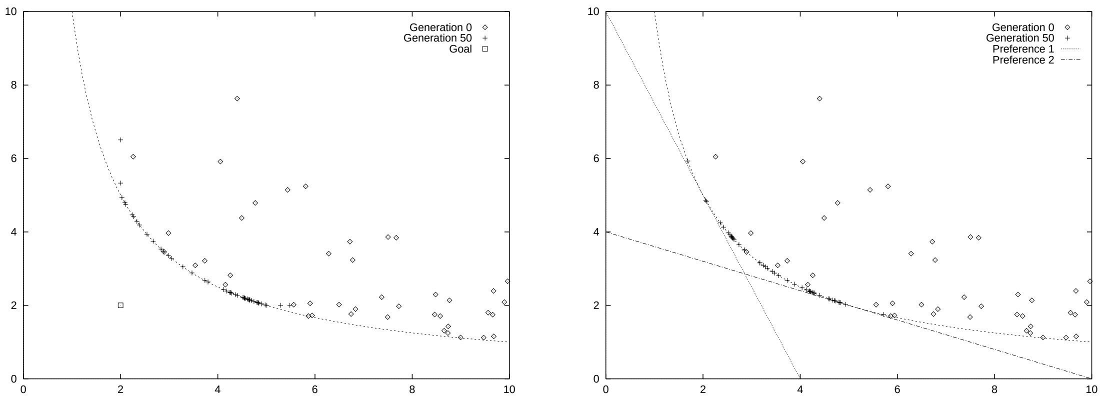


Figure 10: Strongly stressing criterion 2 over criterion 1 shifts the region of convergence accordingly.



(a) (b) (c)

Figure 11: Population after 15 generations with focus on criterion 2 (a), equal importance of both criteria (b), and strong focus on criterion 1 (c).



(a) (b)

Figure 12: G-MOEA (b) can lead to equivalent results when compared to the goal programming approach (a).