A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers

Joshua Knowles
School of Chemistry, University of Manchester, Faraday Building,
Sackville Street, PO Box 88, Manchester M60 1QD, UK. Email:j.knowles@manchester.ac.uk

Abstract

When evaluating the performance of a stochastic optimizer it is sometimes desirable to express peformance in terms of the quality attained in a certain fraction of sample runs. For example, the sample median quality is the best estimator of what one would expect to achieve in 50% of runs, and similarly for other quantiles. In multiobjective optimization, the notion still applies but the outcome of a run is measured not as a scalar (i.e. the cost of the best solution), but as an attainment surface in k-dimensional space (where k is the number of objectives). In this paper we report an algorithm that can be conveniently used to plot summary *attainment surfaces in any number of dimensions (though it is particularly suited for three). A summary attainment surface is defined as the union of all tightest goals that have been attained (independently) in precisely s of the runs of a sample of n runs, for any $s \in 1..n$, and for any k. We also discuss the computational complexity of the algorithm and give some examples of its use. C code for the algorithm is available from the author.*

1 Introduction

Plots visualizing approximation sets obtained from runs of multiobjective optimizers are useful for a number of reasons:

1. Many performance indicators in multiobjective optimization are not always capable of discerning order relations between the approximation sets of even a single pair of optimization runs [4].

2. Decision makers may have preferences towards certain regions of or shapes of Pareto front, not generally (or easily) expressible before optimization, but that are ultimately used to judge the quality of the approximations sets.

3. Some performance indicators do not adequately express the amount by which one approximation set should be judged better than another.

4. Looking at approximation set shape can provide insight into the strengths and weaknesses of an optimizer, or provide information about how it is working.

5. Visualization methods can provide a 'sanity check' to validate any performance indicators being used.

6. When the true Pareto front is known, *seeing* the distance away from it and coverage along it achieved can provide a supplement to any performance indicators used.

For these reasons, it is still (and will probably remain) common practice to supplement the use of performance indicators with plots of approximation sets.

However, when running one or more algorithms several times, plots of approximation sets, showing all points, quickly become confusing and can even be misleading. In particular, it is difficult to separate out individual approximation sets (i.e. from different runs of the algorithm(s)) and to understand clearly the *distribution* of the location and extent of the different approximation sets over multiple runs. This is particularly so if performance differences are small and if, say, 50 runs of two algorithms are to be plotted.

In this paper we present a method that builds on the seminal work of Fonseca and Fleming [1] on *attainment surfaces*, a means of more clearly visualizing one or more approximation sets. In that article, *summary* or *quantile* attainment surfaces were conceptually defined as the union of all the 'tightest goals' attained in a fraction of at least s/n of the approximation sets (or algorithm runs). Our contribution is an algorithm that can be conveniently used to compute approximate attainment surfaces for plotting in any number of objectives. The algorithm follows very closely one given in [3], which was used to generate points weakly dominated by an approximation set. Our algorithm takes n approximation sets, and a parameter $s \in 1..n$ as input and

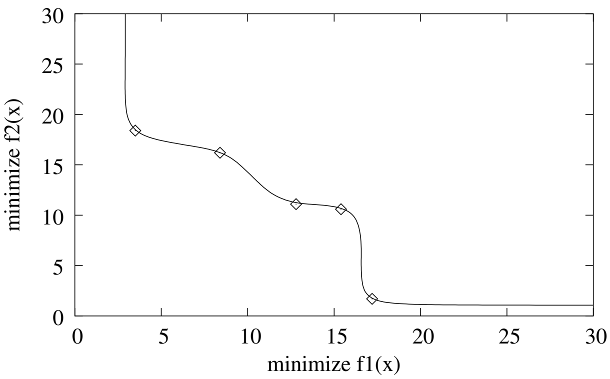

Figure 1. Interpolating an approximation set is easy to do but is 'dangerous' and incorrect

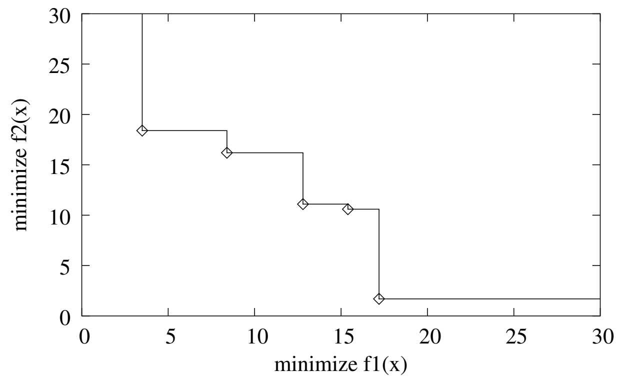

Figure 2. An attainment surface is the family of tightest goals that has been attained by the approximation set defining it

generates points (distributed on a grid), and lying on summary attainment surface s.

The rest of the paper is organized as follows. Section 2 revisits the concept of an attainment surface and demonstrates its advantages by comparing some 2-d plots of points from runs of an optimizer with plots of the associated attainment surfaces. Section 3 goes on to recall the definition of *summary* attainment surfaces and how these can be computed from sampling lines. Section 4 constitutes the original contribution of this paper. It gives a the new algorithm for computing grids of points on a summary attainment surface and derives the computational complexity of this algorithm. Section 5 gives more examples of the use of the method, including figures of 3-d attainment surfaces, and discusses the correct interpretation of these plots. Section 6 concludes.

2 Attainment surfaces

Much of the description given in the following two sections follows closely the presentation in the original attainment surfaces paper of [1], and is included here as a 'memory-refresher'.

The output of a single run of a multiobjective optimizer — an approximation set — on a two-objective minimization problem, is plotted in Fig. 1. It is tempting to interpolate the points obtained with a smooth curve, as shown in the plot, and conclude that this should be the shape of the underlying true Pareto front. Or, if not this, then to conclude that the curve is what has been attained in this particular run of the optimizer. Of course, everyone familiar with Pareto optimization should be clear that neither of these is a safe or correct interpretation of the approximation set. However, although it is not correct to interpolate the points with a smooth line, one *can* replace the points by a boundary, and usefully so; it is in fact possible to 'draw a boundary in the objective space separating those points that are dominated by or equal to at least one of the data points, from those that no data point dominates or equals' [1]. Such a boundary is called an *attainment surface*, and one is shown in Fig. 2. This boundary is 'the family of tightest goals known to be attainable as a result of the optimization run' [1].

Importantly, by exchanging the plot of the (approximation set) points only, with the plot of the attainment surface, it is much easier to identify 'gaps' in the distribution of points — thus the attainment surface emphasises the distribution achieved, as well as indicating the quality of the individual points.

Another advantage of attainment surfaces over simply plotting points comes when we want to display the outcome of multiple runs of one or more optimizers. Usually, plots showing several runs are confusing and misleading because, once again, the eye is tempted to interpolate between points, or worse, just finds it impossible to pick out the points of one run from those of another. For illustration, see Fig. 3. Plotting the same data using attainment surfaces again emphasizes gaps in the different runs of the optimizer(s) more easily, making it much easier to interpret results correctly, as shown in Fig. 4. (It is true that both Fig. 3 and Fig. 4 would benefit from colour, but the attainment surfaces plot would benefit the more).

Moreover, plots of multiple attainment surfaces yield two further important boundaries. There is a region above and to the right of all attainment surfaces (region 1 in Fig. 4); the boundary of this region (which is generally made up of different sections of the different attainment surfaces) is the set of goal vectors that has been attained in *every* single run of the optimizer(s). For a single optimizer, each point on this surface (independently) is therefore an estimate of a goal that is attained in the worst case (i.e. it visualizes worst-case performance). Similarly, there is a region below and to the left of all surfaces (region 2 in Fig. 4),

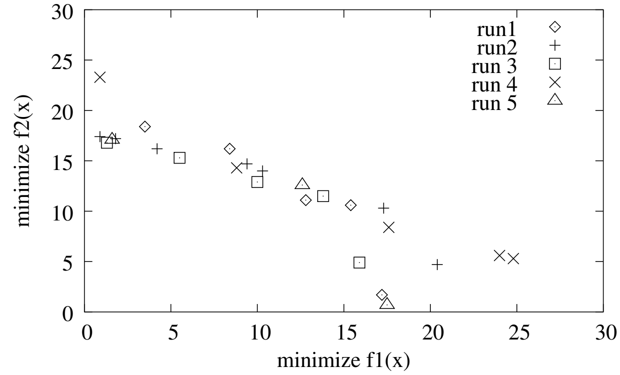

Figure 3. Plotting multiple approximation sets can be confusing, even when there are only a few points per set, and few sets

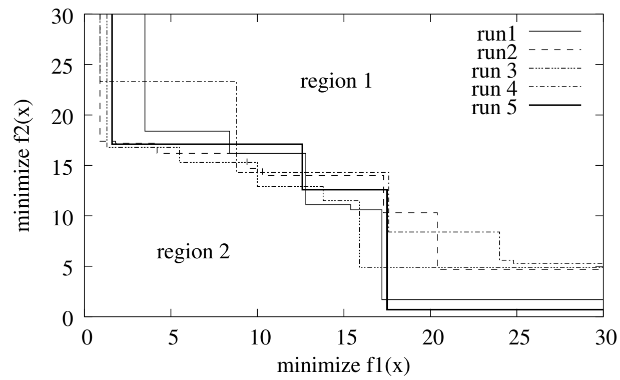

Figure 4. Plotting multiple attainment surfaces (rather than the approximation sets' points) makes it easier to interpret results properly

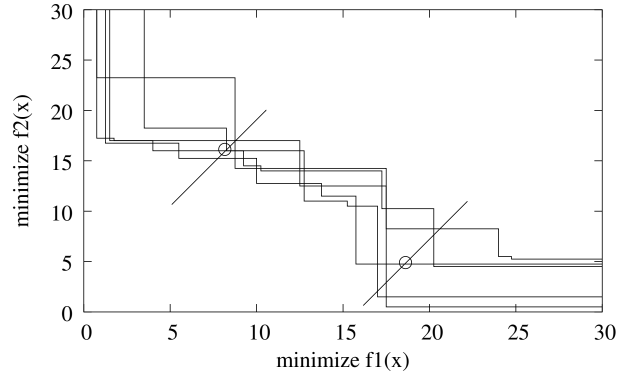

Figure 5. Diagonal lines used to sample the attainment surfaces locate the position of the jth summary attainment surface at different places. The circles indicate two points through which the third summary attainment surface passes

which is the region not attained in any run. The boundary thus represents what has been just attained by the *combination* of all the runs. In between these two boundaries is a region that represents what has been attained in some runs but not in others. The idea behind *summary* attainment surfaces (the concept was introduced in [1] but not referred to by this name) is to divide this region further into regions that were attained only in certain fractions of the runs. This is explained in the next section.

3 Summary attainment surfaces from diagonal sampling lines

Consider the attainment surfaces plot in Fig. 5. Two diagonal lines cutting through the attainment surfaces are shown. Consider the lower sampling line: it intersects the five attainment surfaces at five distinct points, and these points are obviously ordered along the sampling line. It is thus simple to pick out intersection one, two, ..., or s on this sampling line, counting from the lower left end. Intersection s on the sampling line is a goal (vector) that was achieved in precisely s of the runs. In the figure the third intersection on this line is circled. For the upper sampling line, two pairs of intersections coincide but the third intersection is, nevertheless, at the circled point. So, the two sample lines serve to define two points through which the third summary attainment surface passes.

By using very many diagonal sampling lines, we could build up a set of points that would approximately represent the union of all goals achieved (independently) in precisely s of the runs. We could join these goal vectors up using another attainment surface, and this would summarise some aspect of the distribution of approximation sets — it would be a summary attainment surface.[1] If we had, say 51 runs of an optimizer and we showed, say, the first, the twenty-sixth, and the fifty-first summary attainment surfaces, this would visualize the distribution of goals achieved more clearly than plotting all 51 of the result surfaces. An important reason why summary attainment surface plots are easier to interpret than plots of many result surfaces is that summary surfaces never cross each other. In other words, the summary attainment surface s weakly dominates summary attainment surfaces $s+1$, $s+2$, etc., for all s. Five summary

[1] At this point, we wish to distinguish a *summary* attainment surface, which summarises a number of approximation sets, from an attainment surface, which is just the surface defined by a single approximation set. To make this distinction clearer, we will use the term '*result* attainment surface' or the shorter 'result surface' for the latter. For the former we will also sometimes use a shorter form, i.e. 'summary surface'

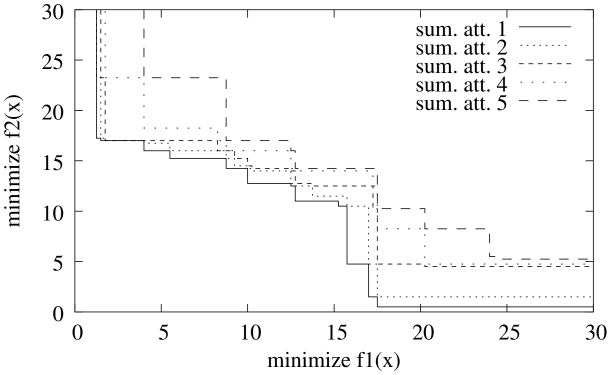

Figure 6. Five summary attainment surfaces. Notice how the surfaces never cross eachother

attainment surfaces for the five surfaces of Fig. 5 are show in Fig. 6.

Unfortunately, there is a problem with the diagonal sampling line method of computing approximate summary attainment surfaces: in three or more dimensions (objectives), it is difficult to arrange the sampling lines so that they intersect the surfaces in a 'nice' way that is informative to the eye. This is because although evenly spaced diagonal lines can be defined, the intersections along the diagonals will not be evenly spaced in any dimension. Thus the eye has no basis from which to reconstruct the three or more dimensions.

Recently, however, [3] proposed another method for finding points weakly dominated by an approximation set, which uses axis-aligned lines (in effect). This means that a uniform, grid-like sampling of the surface (in any number of dimensions) can be easily obtained. We make use of this method in our algorithm, described next.

Algorithm 1 Compute points on summary attainment surface s with resolution r. (Components of an objective vector are denoted using superscript)

```
procedure PLOT_ATTAINMENT(set_1..set_n, s, r)
    read in approximation sets 1..n of dimension k objectives
    for each objective j ∈ 1..k do
        for each approximation set m ∈ 1..n do
            sort points in ascending order on objective j
        end for
        for t in 1 to r^(k−1) do
            for each objective i ∈ 1..k, i ≠ j do
                q^i ← point_on_grid(t)^i
            end for
            weak_dom_count ← 0
            for each approximation set m ∈ 1..n do
                dominated ← false
                for next point p in sorted approximation set m do
                    q^j ← p^j
                    if p dominates q then
                        dominated ← true
                        break from for loop
                    end if
                end for
                if dominated = true then
                    increment weak_dom_count
                    intersection_m ← p^j
                end if
            end for
            sort intersections in ascending order
            if weak_dom_count ≥ s then
                q^j ← intersection_s
                print_vector(q)
            end if
        end for
    end for
end procedure
```

4 Summary attainment surfaces from a grid of points

4.1 The algorithm

The figure, Algorithm 1, gives the pseudocode of our algorithm for computing and plotting points on any of the n possible summary attainment surfaces, defined by an input of n approximation sets. Here we are assuming minimization of all objectives and the approximation sets $1..n$ are all internally nondominated.

The algorithm works by considering one objective j at a time. For each approximation set, independently, the points are sorted in ascending order according to their jth objective values. Lines on a grid of resolution r in the remaining $k-1$ dimensions are then projected onto each of the results surfaces. (This is visualized for a 3-objective case in Figure 7 with resolution $r = 10$, although intersections with only one surface are shown for the sake of clarity.) The intersections on each line are sorted and, if there are at least s of them, then intersection s in the sorted list is the required one on summary surface s. Lines in each of the k objective directions are used to build up the surfaces. Note: the specification of the grid's extent and resolution is done by the user; the point_on_grid(t) function then generates the tth point, $t \in 1..r^{k-1}$, for projection onto the attainment surface.

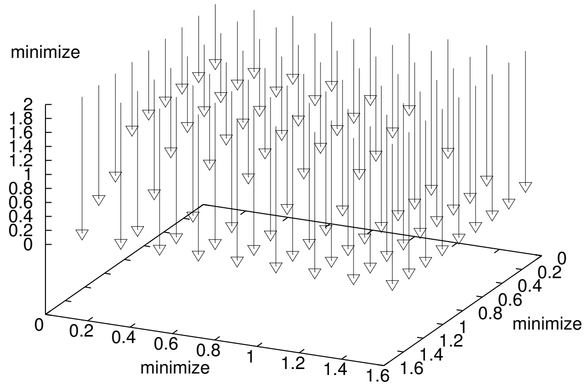

Figure 7. Axis-aligned lines on an even grid in $k-1$ dimensions being projected onto a single approximation set. The intersections are shown by the triangular points. Notice how the eye easily 'sees' the three dimensional shape of the underlying surface

4.2 Computational complexity

For fixed resolution r, which sets the number of sampling sites in each dimension, the above algorithm is clearly exponential in k, since the total number of sampling 'lines' grows as r^{k-1}. However, importantly, for a fixed total number of sampling lines (which is what determines how many points are on the attainment surface), the complexity of the algorithm is polynomial in n, in the size of the approximation sets, and in k. This polynomial time complexity means that the algorithm runs in just a few seconds for even 50 approximation sets of up to a 1000 points, and three objectives, even when the resolution is high (see figures 9 and 10). This contrasts with an exact computation of all the precise points where the attainment surface changes, a calculation that is exponential in the approximation set size when k is general, and which takes of the order of hours or days for similar inputs (the author has tested a simple implementation of such an algorithm).

5 Summary attainment surface plots in practice

In this section, we give two 3-d plots of summary attainment surfaces, derived from running different algorithms on a test function. We also show two futher higher resolution 3-d plots generated by our code on a mixed minimization/maximization problem.

5.1 Real cases

Much of the time, attainment surface plots will be used to visually compare the performance of two optimizers, after several runs of each have been performed. In this case, the best (1st) summary attainment surface for each algorithm is not especially informative, unless one is likely to run the optimizer multiple times in the final application. This is because the 1st summary attainment surface merely visualizes the goal vectors attained by the combination of all runs. More frequently, it is more useful to get some feel for the median and perhaps worst case performance of the optimizer, in terms of what goals are attained. Plotting the median and worst summary surface for two algorithms gives just four lines, and this is often easy to interpret on a single 2-d plot (for a 2-objective problem).

For three-objective problems, it is more difficult to show plots that are informative in a static, monochrome envrionment. However, plotting the same attainment surface from two different algorithms, side by side, on the same axes scales can do the trick, as shown in Figure 8.

5.2 Mixed minimization / maximization plots

The code we provide allows one to specify whether each objective is to be minimized or maximized. This allows plotting of mixed minimization / maximization attainment surfaces on their original axes. Two such attainment surface plots are shown in Figures 9 and 10. A higher resolution has been used than in Figure 8.

6 Conclusion

An algorithm for computing an even, grid-based sampling of a summary attainment surface, for general dimension k, was presented. The method is convenient for visualizing performance differences between algorithms because it is quite cheap to compute compared to exact methods. The fact that the intersections are derived from a grid means that the eye easily interprets the plots, as demonstrated.

Acknowledgments

JK is supported by a David Phillips fellowship from the Biotechnology and Biological Sciences Research Council (BBSRC), UK.

References

[1] C. M. Fonseca and P. J. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature—PPSN IV*, Lecture Notes in Computer Science, pages 584–593, Berlin, Germany, 1996. Springer-Verlag.

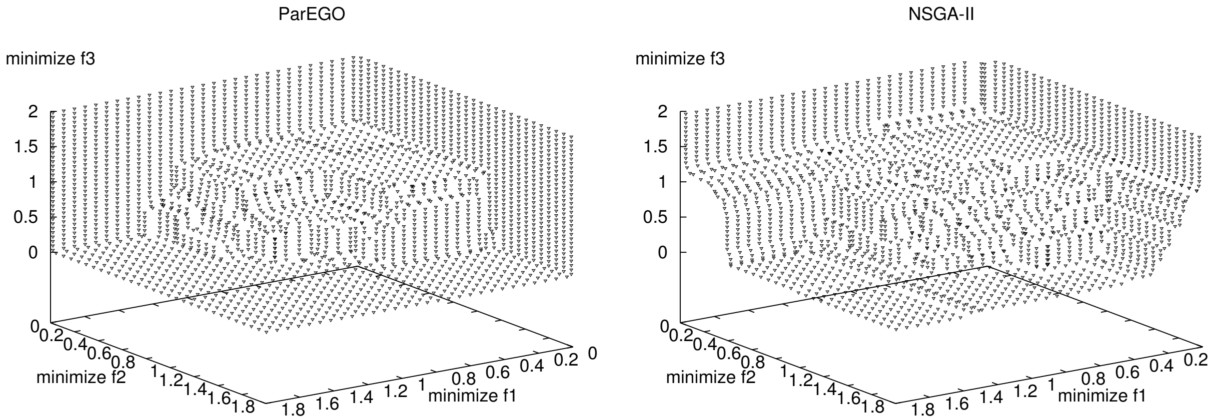

Figure 8. Comparing equivalent attainment surfaces from two different algorithms. Clearly, here, the surface for ParEGO is considerably better. The plot is taken from [2]

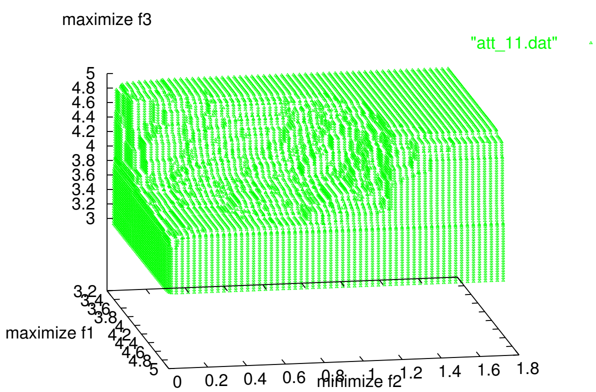

Figure 9. Median attainment surface on a mixed minimization/maximization problem from 21 approximation sets. Here, a resolution $r = 60$ was used and the plot was generated in approximately 5 seconds

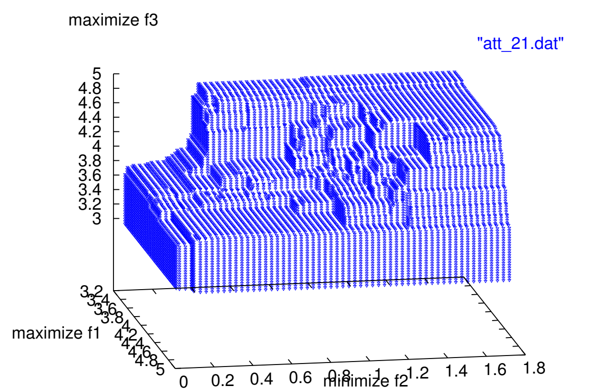

Figure 10. As for Figure 9, but here the worst attainment surface is shown

[2] J. Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 2005. (In press).

[3] K. I. Smith, R. M. Everson, and J. E. Fieldsend. Dominance measures for multi-objective simulated annealing. In *2004 Congress on Evolutionary Computation (CEC'2004)*, volume 1, pages 23–30, Portland, Oregon, USA, 2004. IEEE Service Center.

[4] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.