Common Issues

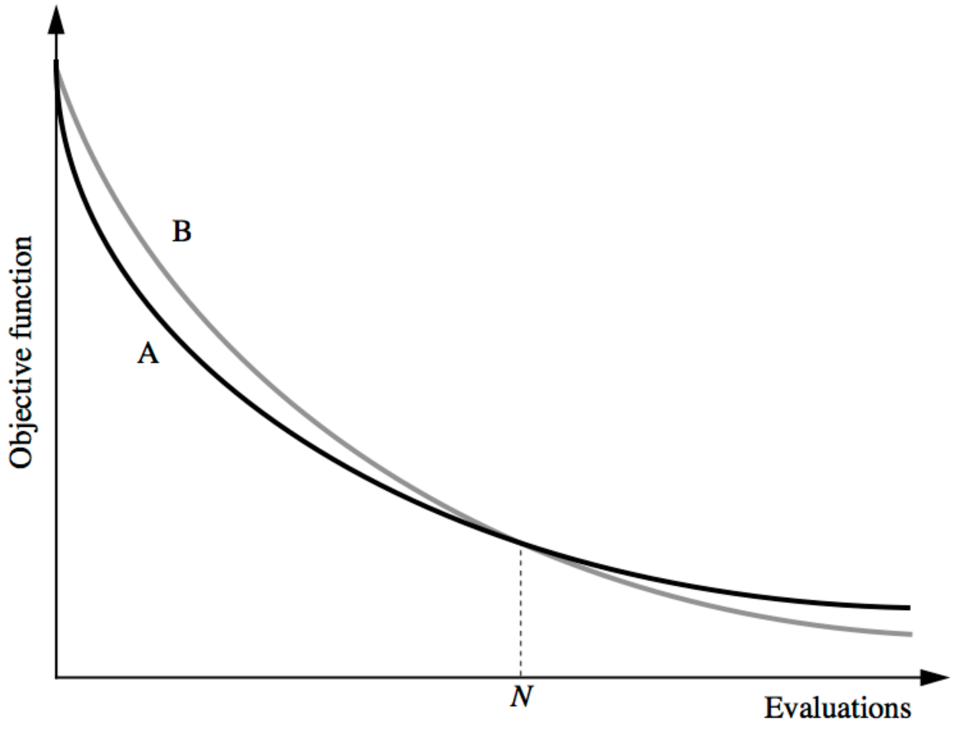
Performance Measures

It is not necessarily easy to measure the performance of a stochastic optimization method, because, unless exactly the same sequence of random numbers is used, the algorithm will not perform the same search on the same problem, even if given the same starting point. For this reason, before making any claims as to the performance of an algorithm on a given problem, several (at least 25, preferably 50 or more) runs should be made using different random number sequences (this is usually done by specifying different seeds to the random number generator used) and, if the starting point can affect the run, different initial solutions.

There are different ways in which the algorithm performance can be measured:

- If the optimal solution is known, then the length of (c.p.u.) time or the number of objective function evaluations required in order to locate the optimum can be used as a measure. As stochastic methods are not guaranteed to locate the global optimum, this is not a particularly helpful measure. Also, of course, in many (most) real-world problems, the optimal solution is not known *a priori*.
- The value of the objective function for the best solution found after a specified length of (c.p.u.) time or a specified number of objective function evaluations is a more generally useful performance measure. Because it is in general true that the longer a stochastic search

Figure 1: Hypothetical Performance Curves.



method is allowed to run the better the quality of the best solution found will be, it is important to stipulate sensible comparison points. This is best done by plotting objective function against number of evaluations (or c.p.u. time) graphs, in order to be able to compare not just the objective reductions achieved by different methods but also the rates of progress. For instance, Figure 1 shows that although method A makes better initial progress on the hypothetical problem than method B, after N evaluations method B overtakes method A and eventually does quite a lot better.

- The number of evaluations is only a fair basis for comparison if the computational cost of each evaluation is constant (or almost constant). The c.p.u. cost of evaluations can depend strongly on the optimization search method used. For instance, if an iterative method is being used as part of the evaluation method, then the time for the evaluation method to converge may well depend on the size of the change made to the solution, so that evaluations will be cheaper for a search method which makes small changes than for one which makes large changes. In such circumstances c.p.u. time is the only fair basis for comparison between search methods.

Once an appropriate time over which to compare performances has been identified, then, for stochastic search methods, two measures are of interest:

- The value of the best objective found averaged over multiple runs;
- The standard deviation in the value of the best objective found over multiple runs.

It may well be that a method which gives on average slightly worse objective values but does so very consistently (i.e. with a low standard deviation) will be preferred for some applications to a method which performs inconsistently (high standard deviation), potentially giving very good performance but also potentially quite poor performance.

Archiving

An important element of all heuristic search methods is the way in which information about the search is recorded. Obviously one is interested in the best solution found to the optimization problem under consideration (note that this is rarely the final solution visited in stochastic searches), but a stochastic optimization method will inevitably have explored the search space quite widely and additional information about the problem being solved may well be of value. On the other hand, many thousands, perhaps millions, of solutions will have been examined and one would certainly not want to be presented with all that information. Some form of discrimination must therefore be applied when storing information to be presented at the end of the optimization.

Best L Solutions

One obvious strategy is store the best L solutions located (their control variable, objective function and constraint values), where L may be of the order of 25. This is easily implemented. The one potential disadvantage is that the best 25 solutions may well be very similar and therefore this 'archive' may not give very much information about the rest of the search space explored.

Best L Dissimilar Solutions

This disadvantage may be overcome by storing the best L solutions with a minimum level of *dissimilarity*. An obvious requirement is therefore a measure of dissimilarity between solutions. This is most readily defined in terms of the control variables. For instance, for continuous control variables, a simple measure of dissimilarity between two solutions $\mathbf{x}_A$ and $\mathbf{x}_B$ is:

$$D_{AB} = \sqrt{(\mathbf{x}_A - \mathbf{x}_B)^T(\mathbf{x}_A - \mathbf{x}_B)}\ , \qquad (1)$$

i.e. the Euclidean distance between the solutions in control variable space. If the individual control variables vary over significantly different ranges (for instance, $0\text{ m} \le x_1 \le 10\text{ m}$ and $0\text{ m} \le x_2 \le 0.001\text{ m}$), then it may be appropriate to rescale them so that within the optimization routine they can vary over the same range, e.g. (0, 1) or (−1, 1).

In addition, one needs to define two dissimilarity thresholds $D_{\min}$ and D_{sim}, which are used as follows:

- Let the number of solutions in the archive be l, and let these be labelled $\mathbf{x}_K$.
  Let $\mathbf{x}_J$ be a new solution, a candidate for archiving.
  Let $\mathbf{x}_E$ be the archived solution $\mathbf{x}_J$ most closely resembles, i.e. $D_{EJ} \le D_{KJ}\ \forall\ K = 1, \ldots, l$.
  Let $\mathbf{x}_G$ be the worst archived solution, i.e. $f(\mathbf{x}_G) \ge f(\mathbf{x}_K)\ \forall\ K = 1, \ldots, l$.
- If fewer than L solutions have been archived (i.e. the archive is not yet full), archive $\mathbf{x}_J$ if it is sufficiently dissimilar to all the solutions archived:

$$\text{If } l < L, \text{ archive } \mathbf{x}_J \text{ if } D_{KJ} > D_{\min}\ \forall\ K = 1, \ldots, l\ . \qquad (2)$$

- If the archive is full, archive $\mathbf{x}_J$ if it is sufficiently dissimilar to all the solutions archived and better than the worst of these:

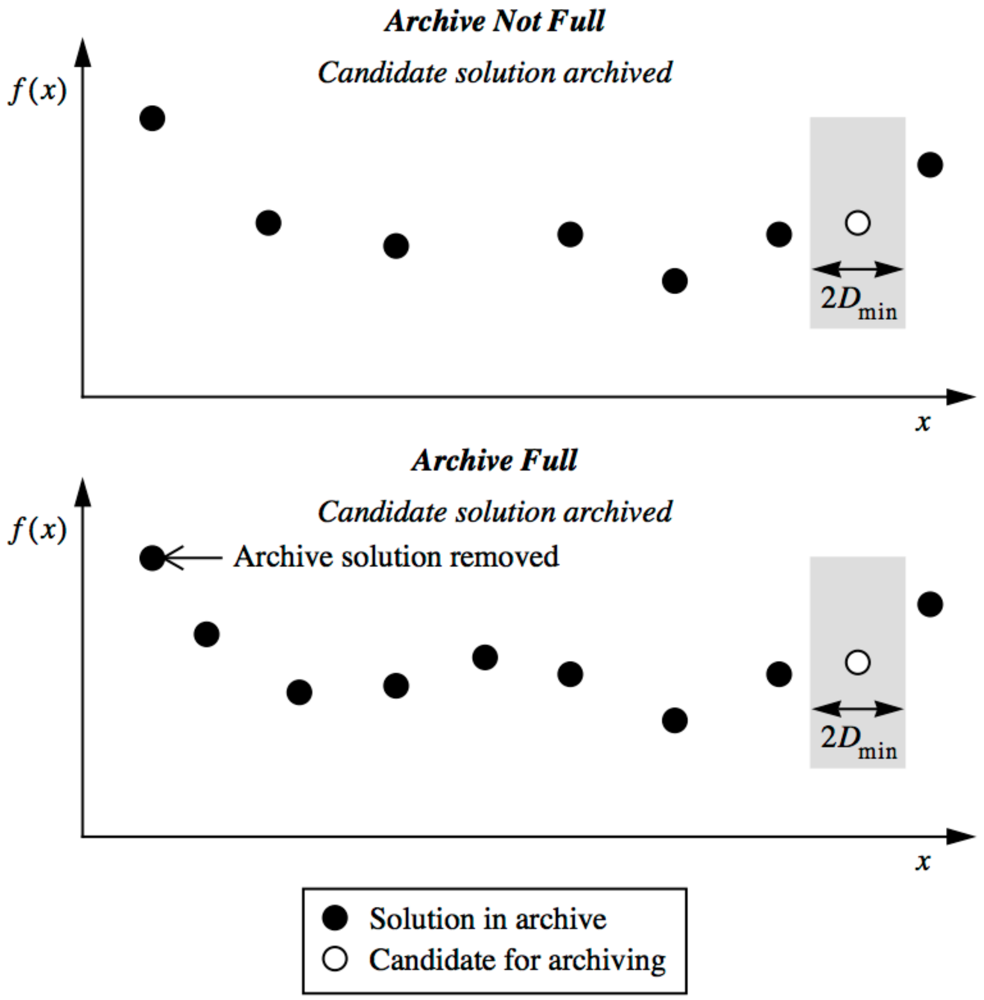
$$\text{If } l = L, \text{ archive } \mathbf{x}_J \text{ if } D_{KJ} > D_{\min}\ \forall\ K = 1, \ldots, L \text{ and } f(\mathbf{x}_J) < f(\mathbf{x}_G)\ . \qquad (3)$$

  (In this case $\mathbf{x}_J$ replaces $\mathbf{x}_G$ in the archive.)

- If $\mathbf{x}_J$ is not sufficiently dissimilar to the archived solutions, archive it if it is the best solution found so far:

$$\text{If } D_{KJ} < D_{\min} \text{ for some } K, \text{ archive } \mathbf{x}_J \text{ if } f(\mathbf{x}_J) < f(\mathbf{x}_K)\ \forall\ K = 1, \ldots, l\ , \qquad (4)$$

Figure 2: Dissimilarity Archiving — D_{min} Threshold Met.



or, if it is not the best solution found so far, archive it if it is sufficiently similar to and better than $\mathbf{x}_E$:

$$\text{If } D_{KJ} < D_{\text{min}} \text{ for some } K, \text{ archive } \mathbf{x}_J \text{ if } f(\mathbf{x}_J) < f(\mathbf{x}_E) \text{ and } D_{EJ} < D_{\text{sim}} . \qquad (5)$$

(In both cases $\mathbf{x}_J$ replaces $\mathbf{x}_E$ in the archive.)

Figures 2 and 3 show examples of this logic in action for the simple case of an optimization problem with just one control variable.

Using this logic a helpful picture of the search space can be built up. Obviously appropriate values of the dissimilarity thresholds D_{min} and D_{sim} will be problem dependent. It is clear that $D_{\text{sim}} < D_{\text{min}}$ and indeed D_{sim} should probably be at least an order of magnitude smaller than D_{min}.

Figures 4 and 5 show typical outputs from both types of archiving scheme for a stochastic optimisation algorithm applied to the two-dimensional version of Keane's Bump Function.

Figure 3: Dissimilarity Archiving — D_{min} Threshold Not Met.

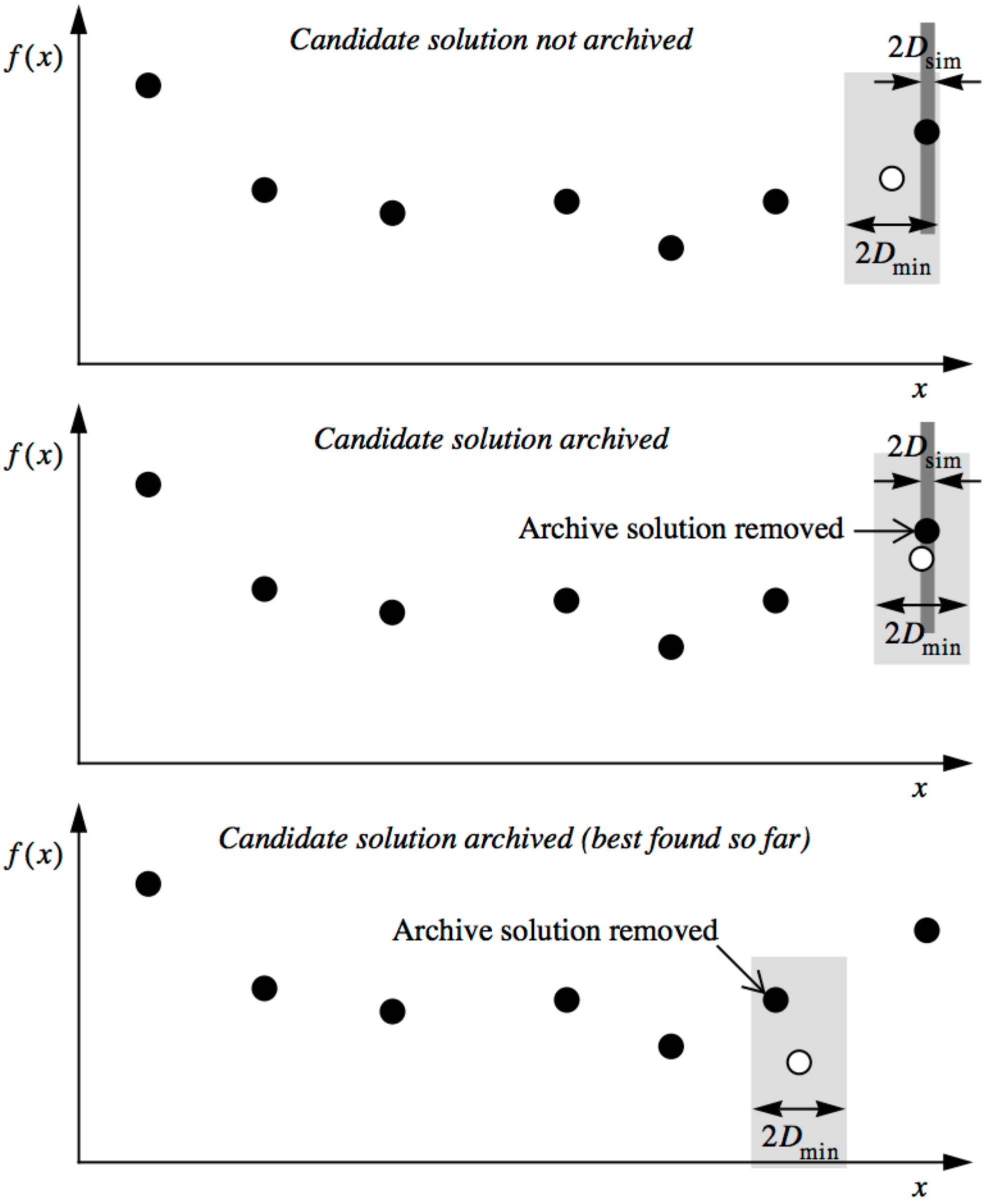


Figure 4: Archive Contents for Keane's 2D Bump Function.

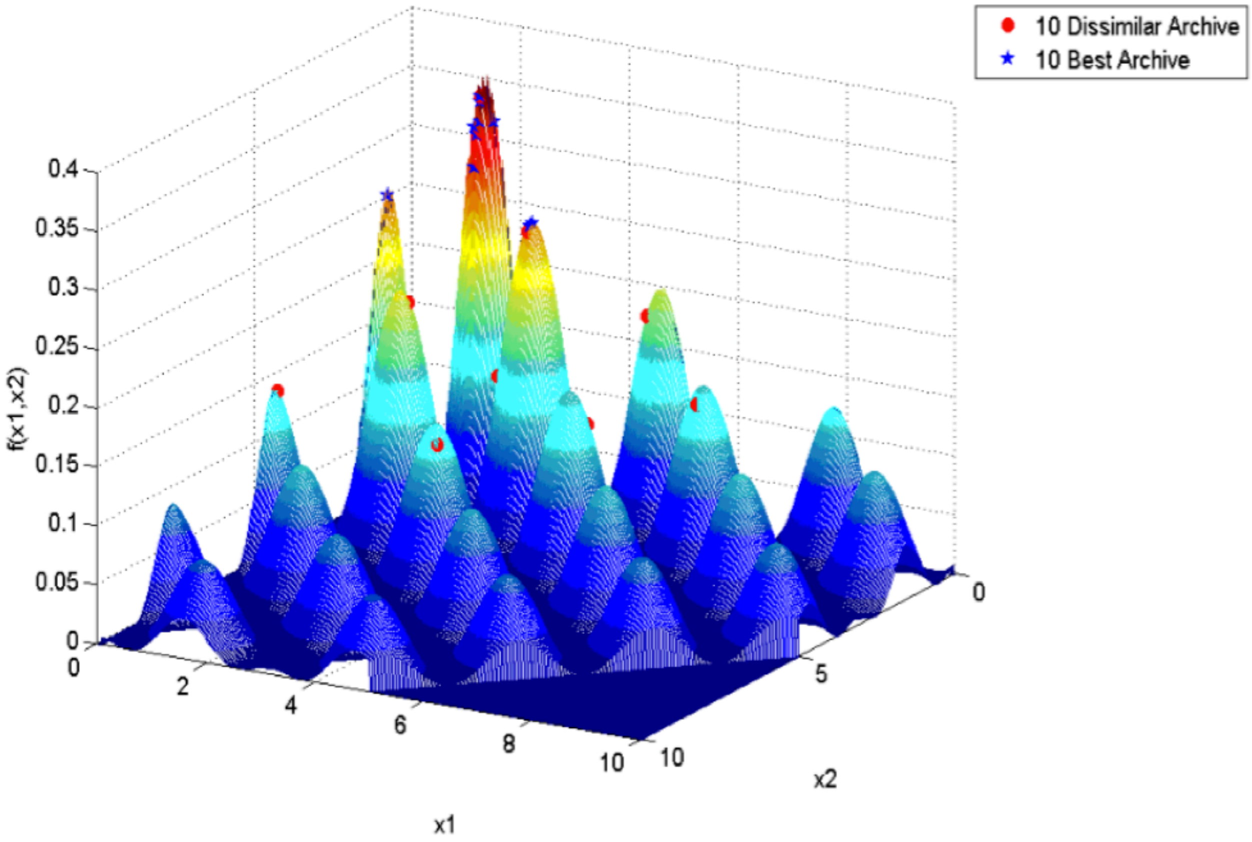
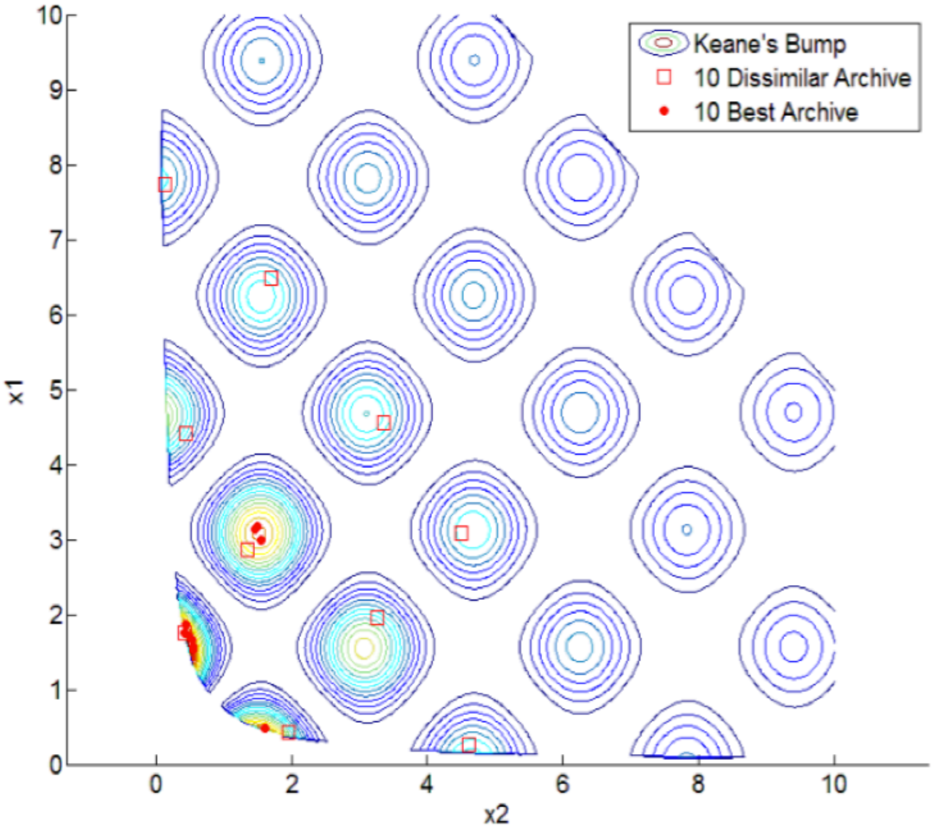


Figure 5: Archive Contents for Keane's 2D Bump Function (Plan View).


Keane's Bump
10 Dissimilar Archive
10 Best Archive
x1
x2