COMPUTATIONAL STEERING OF A MULTI-OBJECTIVE GENETIC ALGORITHM USING A PDA

A. SHENFIELD

M. ALKAROURI

P.J. FLEMING

Department of Automatic Control and Systems Engineering
University of Sheffield
Sheffield, S1 3JD
UK

Research Report No. 878
January 2005

Computational Steering of a Multi-Objective Genetic Algorithm using a PDA

A. Shenfield, M.Alkarouri and **P. J. Fleming**
Department of Automatic Control and Systems Engineering, University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK

Abstract

The execution process of a genetic algorithm typically involves some trial-and-error. This is due to the difficulty in setting the initial parameters of the algorithm – especially when little is known about the problem domain. The problem is magnified when applied to multi-objective optimisation, as care is needed to ensure that the final population of candidate solutions is representative of the trade-off surface. We propose a computational steering system that allows the engineer to interact with the optimisation routine during execution. This interaction can be as simple as monitoring the values of some parameters during the execution process, or could involve altering those parameters to influence the quality of the solutions produce by the optimisation process.

1. Introduction

Decision Making in Engineering Design can often be aided by using genetic algorithms to solve many-objective problems. Typically, these many-objective genetic algorithms (MOGAs) are run non-interactively. The engineer will set the initial parameters of the algorithm and then execute it. During this execution process, which can often take hours or days to complete, user interaction, if any, is limited to the occasional plotting of the intermediate solutions and the possible termination of the algorithm if it appears to have failed (for example, if the algorithm doesn't show convergence). When the execution has finished the solutions produced by the algorithm are assessed and, if the results are not satisfactory, the parameters of the algorithm are adjusted and it is run again. This process of repeated execution of the MOGA leads to an inefficient use of resources, and possibly also to inferior solutions.

As the process of setting the initial parameters of the algorithm can be difficult, especially if little is known about the problem domain, the re-execution of the MOGA with altered parameters is common. Unfortunately, the evolutionary computation community is still some way from possessing anything more useful than 'rules-of-thumb' when it comes to the setting of these initial parameters [1]. One potential solution to this problem is to allow the engineer to interact with the optimisation routine during execution. This is known as computational steering, and may be as simple as allowing the engineer to monitor the values of some parameters in the optimisation process and, if necessary, to adjust others. In this way, the engineer could influence the quality of the solutions produced by the optimisation process.

2. Computational Steering and PDAs

[2] defines computational steering as an approach that improves the integration of simulation and visualisation in the computational process, allowing the engineer or scientist to control the succession of steps required to solve engineering and computational science problems. The desire to interact with their simulations is nothing new for engineers and scientists however, as far back as 1987 the Visualization in Scientific Computing Workshop reported:

"Scientists not only want to analyze the data that results from super-computations; they also want to interpret what is happening to the data during super-computations. Researchers want to *steer* calculations in close to close-to-real time; they want to be able to change parameters, resolution, or representation, and see the effects. They want to drive the scientific discovery process; they want to *interact* with their data." [3]

Currently, the majority of computational steering systems are applied to large simulations of compute-intensive models, such as those used in the study of nano-indentation of iron [4 or the study of implantable defibrillator device designs [5].

The visualisation of the intermediate results of the computational process is extremely important. It must allow the engineer or scientist to efficiently extract the relevant information from the data [6], so as to be able to make an informed choice about which aspects of the process to adjust. The complexity of the visualisation should be able to be tailored to the hardware available to the user (for example, a lap-top, PDA, etc.) [4].

In our application domain of Engineering Design, it would be especially useful for an engineer working on a multi-objective optimisation problem to be able to check on the progress of the algorithm from the field. An ideal client for this computational steering system would provide low-cost, portable access to the system.

The implementation of this steering client can be effectively realised by using a PDA enabled with a wireless connection. The low cost involved in the use of a PDA based client would allow a wide uptake of this steering system by engineers in the field. The portability of a PDA with a wireless connection also results in a very flexible system.

3. Decision Making in Evolutionary Multi-Objective Optimisation

The main role of the decision maker (DM) in evolutionary multiobjective optimisation (EMO) is usually to select a single solution from the potentially infinite Pareto-optimal solution set, according to some criteria. In practice the DM is usually only interested in a sub-set of the trade-off surface, thus there is little or no benefit in representing parts of the trade-off surface that lie outside this region of interest (ROI). Allowing the DM to focus the search on relevant areas of the solution space increases the efficiency of the optimisation process and reduces the amount of irrelevant information that the DM has to consider [7].

DM preferences can be incorporated into the optimisation process in three ways; *a posteriori*, *a priori*, and progressively. *A posteriori* methods of preference articulation involve the DM selecting a compromise solution from the global set of Pareto-optimal solutions. *A priori* preference articulation and progressive preference articulation aim to achieve a good representation of the trade-off surface in the DM's ROI. They do this by concentrating the optimiser on a sub-set of the global trade-off surface. In *a priori* articulation of preferences the DM expresses their preferences before the start of the optimisation process. However, often the DM may not be sure of their preferences prior to optimisation, and by stating preferences *a priori* the DM may not investigate some areas of the search space that merit attention. A better method is progressive articulation of preferences, where the DM can express preferences during the search and thus incorporate information that becomes available during the search process.

The first scheme for progressive preference articulation was introduced by Fonseca and Fleming [8]. It extended the Pareto-based ranking scheme to allow preferences to be expressed during the run of the MOGA. These preferences are used in a modified version of dominance which combines Pareto-optimality, constraint satisfaction, goal programming, the lexicographic method, and constrained optimisation to rank the candidate solutions in a multi-objective genetic algorithm.

4. Visualisation in Computational Steering

As mentioned in section 2, visualisation is a key component of computational steering. The visualisation method of any computational steering system must be able to present the user with enough relevant information for the user to guide the process. Therefore, the visualisation method for the intermediate results of our multi-objective genetic algorithm must be able to display high-dimensional data sets, as we are dealing with many objectives.

The visualisation of high dimensional data sets in an intuitive manner is extremely difficult. While scatter diagrams provide a fundamental tool for visualisation of lower dimensional data – allowing the eye to see such features as clustering, outliers and linearity/nonlinearity – they do not generalise easily to more than three dimensions [9]. Methods that have been proposed for solving this visualisation problem include:

- *Scatter Plot Matrices* consist of an array of scatter diagrams formed into an *n* x *n* matrix. Each dimension of the data forms one column and one row of the matrix (see Figure 1.1).
- *Chernoff Faces* [10] are an iconic representation of multidimensional data, used to illustrate trends in that data. Each point in multidimensional space is represented by a cartoon face whose features, such as length of nose and curvature of mouth, correspond to a dimension of the data (see Figure 1.2).
- *Parallel Coordinate Plots* [11] allow the visualisation of high dimensional data in a simple two dimensional representation. Instead of having the axes orthogonal to each other, as in Cartesian geometry, the axes are placed in parallel. Thus a point in *n* dimensional space will be represented as a line that bisects *n* parallel axes (see Figure 1.3).

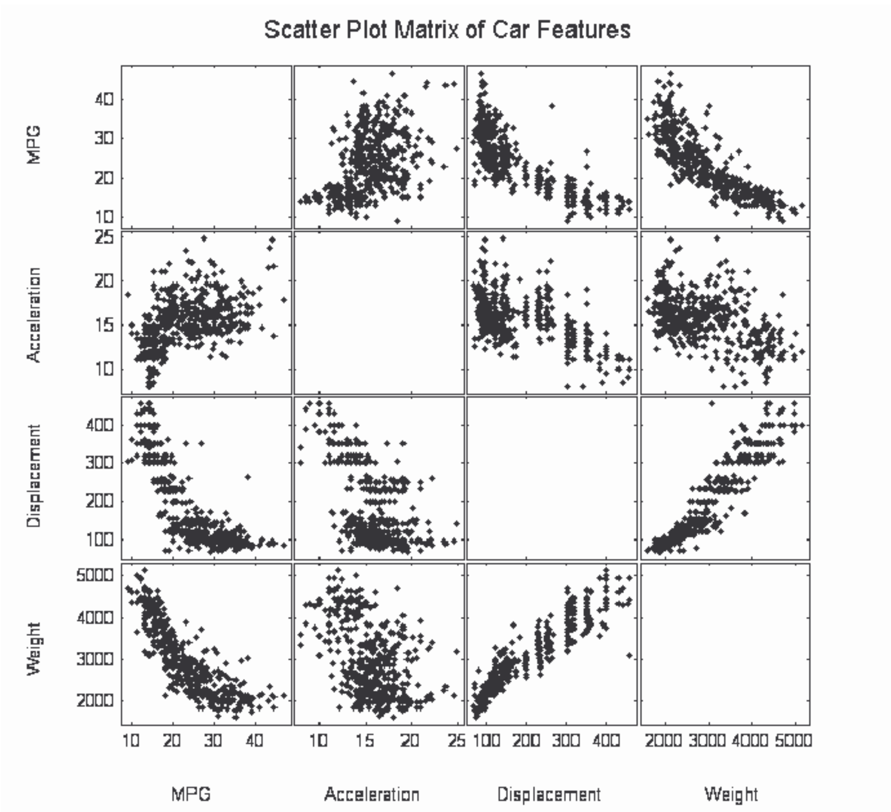

Fig. 1.1 – A Scatter Plot Matrix

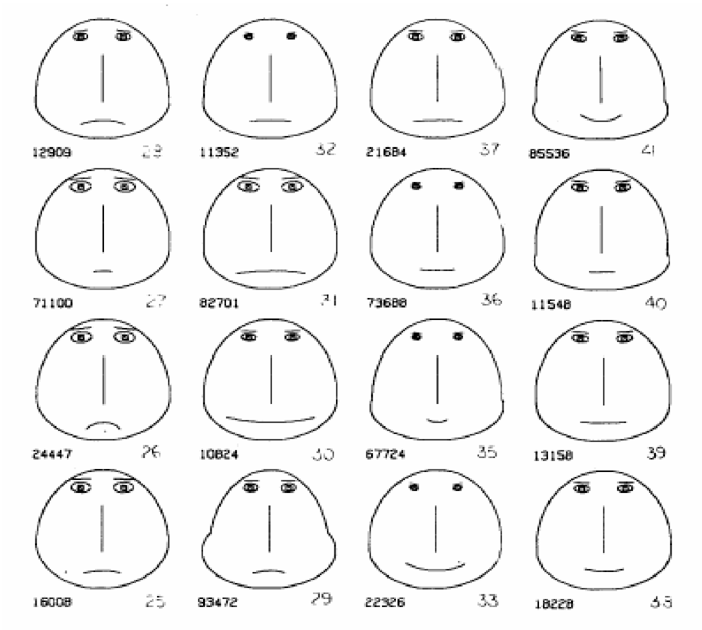

Fig. 1.2 – Chernoff Faces

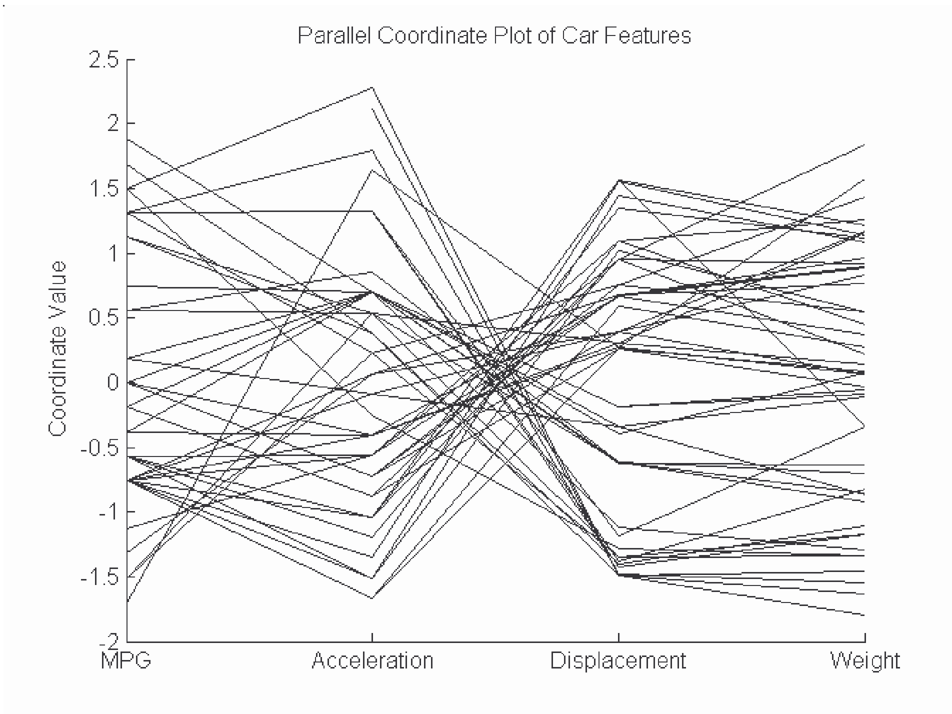

Fig. 1.3 – Parallel Coordinate Plot

5. Implementation

5.1. Visualisation

The visualisation technique used for our computational steering system must be capable of displaying the candidate solutions produced by our many-objective genetic algorithm in an intuitive manner. It is not sufficient just to be able to display this high dimensional data in a two dimensional representation, we must also be able to easily interpret the relationships between the data points. Many techniques have been proposed to solve this problem, and an overview of three commonly used methods is given in section 4. These methods are examined more closely below, and their suitability to our computational steering application is assessed.

Scatter plot matrices are one commonly used technique in the visualisation of high dimensional data sets. They provide a visualisation technique that facilitates rapid scanning of many dimensions; however discovery of high dimensional patterns can be complicated by the disconnected representation of multiple aspects of the same point in high dimensional space [12]. The representational complexity of these scatter plot matrices is high ($O(n^2)$), because they project n dimensions onto n x (n – 1) scatter plots. This means that this technique won't scale well to large numbers of variables. This high representational complexity also means that this technique will be unsuitable for use on a PDA due to the extremely limited size of the screen.

Another technique used to represent many-dimensional data is using Chernoff Faces. This visualisation technique has a lower representational complexity than scatter plot matrices, as each face represents a point of data in high dimensional space. This gives us a representational complexity of $O(n)$. The use of Chernoff Faces, however, amounts to drawing a two dimensional function surface in high dimensional space, and not to the representation of a genuinely high dimensional structure [13]. Other disadvantages to this method of visualisation are that the interpretation of the face is subjective and there is no quantitative information displayed. Chernoff proposed this visualisation technique simply as a way of highlighting which areas of the search space should be targeted for closer examination.

Parallel Coordinate plots also have a representational complexity of $O(n)$, as each line bisecting the axes of the plot represents a single point in high dimensional space. Parallel coordinates lose no data in the representation process; this in turn ensures that there is a unique representation for each unique set of data. Unlike Chernoff Faces, parallel coordinate plots treat each dimension of the data in the same way, resulting in easy plotting of data points. The main weaknesses of this method are that it requires multiple views to see different trade-offs and it can be difficult to distinguish individual points if many data points are represented.

Parallel Coordinate plots were chosen to perform the visualisation of the data in our computational steering system due to their ease of interpretation and the ability to display all the appropriate data on the screen of the PDA. To overcome the potential problem of having difficulty distinguishing individual points when the display is cluttered, we will only represent those candidate solutions that fit our modified definition of Pareto optimality (see section 3). This will prevent the plot from becoming cluttered without removing any of the useful data.

5.2. Steering of the Multi-Objective Genetic Algorithm

The computation steering of our optimisation process can be achieved in two ways. The first of these is by adjusting the parameters of the algorithm. These parameters control the behaviour of the algorithm and can affect both the rate of convergence and the quality of the solutions produced. For example, reducing the exploratory effects of mutation in the algorithm by lowering the mutation rate will reduce the amount of new genetic material coming in to the population in each new generation, and thus increase convergence. However this increase in the rate of convergence will come at the risk of converging to a local optimum.

Some other parameters that we can adjust in our steering system are the upper and lower bounds on the decision variables, the population size (either by increasing the number of immigrants or increasing the number of solutions produced by selection), and the fitness assignment method. These parameters all affect the behaviour of the algorithm in different ways. For instance, tightening or loosening the bounds on the decision variables allows the engineer to focus or widen the search in decision space, while increasing the number of immigrants in the population can force the algorithm out of local optima because it introduces new genetic material.

Changing how fitness is assigned in the algorithm can alter the probability of a solution being carried over to the next generation. If an exponential fitness assignment method is used, then the highest ranked solution will form a proportionally larger part of the next generation compared to a linear fitness assignment method (see Fig. 2).

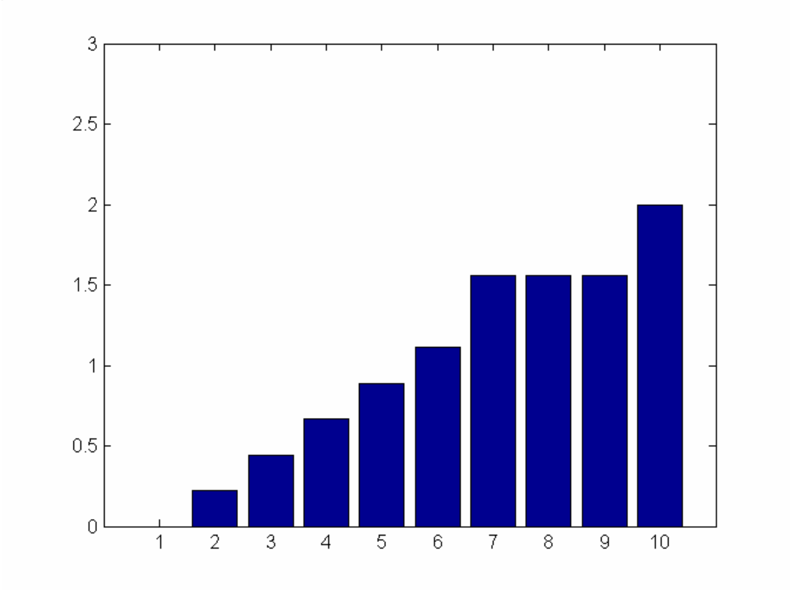

Fig. 2.1 – Graph showing linear fitness assignment to solutions

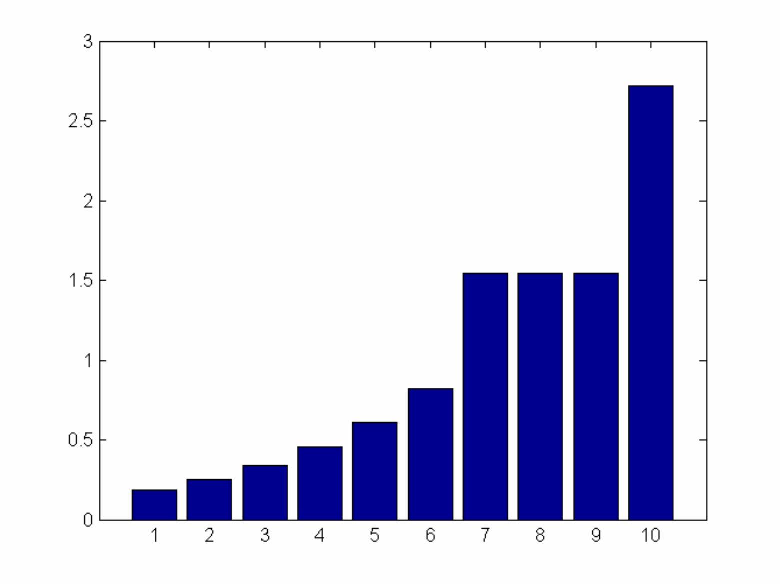

Fig. 2.2 – Graph showing exponential fitness assignment

The second way of steering the optimisation process is to use progressive preference articulation (see section 3) to alter the goals and priorities for the objectives, and thus affect the areas of the search space that the algorithm focuses on. The algorithm is focused on a region in the search space by assigning higher rank to those solutions that are in that region (which is defined by the preferences of the algorithm). Once a satisfactory value has been achieved for one of the objectives, the objective in question can be constrained to be at least as good as that value. All the potential solutions that do not meet this criterion are ranked worse than those that do, and therefore the algorithm is steered away from values that violate that constraint.

References

1. Bullock, S., Cartlidge, J., and Thompson, M., *"Prospects for Computational Steering of Evolutionary Computation"*, in Bilotta, E., et al. (eds.), Workshop Proceedings – Artificial Life VIII, pp. 131-137, MIT Press, 2002.
2. Johnson, C.R., Parker, S.G., et al., *"Interactive Simulation and Visualization"*, IEEE Computer, **32(12)**, pp. 59-65, 1999.
3. McCormick, B.H., DeFanti, T.A., and Brown, M.D. (eds.), *"Visualization in Scientific Computing"*, Computer Graphics, **21(6)**, 1987.
4. Brooke, J.M., Coveney, P.V., et al., *"Computational Steering in RealityGrid"*, in Cox, S. (ed.), Proceedings of the eScience All Hands Meeting 2003.
5. Sci-Run, http://software.sci.utah.edu/scirun.html, Sci-Run homepage.
6. Parker, S.G., Johnson, C.R., and Beazley, D., *"Computational Steering Software Systems and Strategies"*, IEEE Computational Science & Engineering, 4(4), pp. 50-59, 1997.
7. Fleming, P.J., Purshouse, R.C., and Lygoe, R.J., *"Many-Objective Optimization: An Engineering Design Perspective"*, in Coello Coello, C.A., et al. (eds.), Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), LNCS 3470, pp. 14-32, Springer-Verlag, Berlin, 2005.
8. Fonseca, C.M., and Fleming, P.J., *"Multi-Objective Optimisation and Multiple Constraint Handling with Evolutionary Algorithms – Part I : A Unified Formulation"*, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, **28(1)**, pp. 26-37, 1998.
9. Wegman, E.J., *"Hyperdimensional Data Analysis Using Parallel Coordinates"*, Journal of the American Statistical Association, **85(411)**, pp. 664-675, 1990.
10. Chernoff, H., *"The Use of Faces to Represent Points in K-Dimensional Space"*, Journal of the American Statistical Association, **68(342)**, pp. 361-368, 1973.
11. Inselberg, A., *"The Plane with Parallel Coordinates"*, The Visual Computer, **1**, pp. 69-91, 1985.
12. Carr, D.B., Nicholson, W.L., et al., *"Interactive Color Display Methods for Multivariate Data"*, in Wegman, E.J., and DePriest, D.J. (eds.), Statistical Image Processing and Graphics, pp. 215-250, Dekker, NY, 1986.
13. Donoho, D.L., Huber, P.J., et al., *"The Man-Machine Graphics Interface for Statistical Data Analysis"*, in Wegman, E.J., and DePriest, D.J. (eds.), Statistical Image Processing and Graphics, pp. 203-213, Dekker, NY, 1986.