Many-Objective Optimization: An Engineering Design Perspective

Peter J. Fleming[1], Robin C. Purshouse[2], and Robert J. Lygoe[3]

[1] Department of Automatic Control and Systems Engineering,
University of Sheffield, Mappin Street,
Sheffield S1 3JD, UK
P.Fleming@sheffield.ac.uk
[2] PA Consulting Group,
123 Buckingham Palace Road,
London SW1W 9SR, UK
Robin.Purshouse@PACONSULTING.COM
[3] Powertrain Applications Product Development,
Ford Motor Company Ltd

Abstract. Evolutionary multicriteria optimization has traditionally concentrated on problems comprising 2 or 3 objectives. While engineering design problems can often be conveniently formulated as multiobjective optimization problems, these often comprise a relatively large number of objectives. Such problems pose new challenges for algorithm design, visualisation and implementation. Each of these three topics is addressed. Progressive articulation of design preferences is demonstrated to assist in reducing the region of interest for the search and, thereby, simplified the problem. Parallel coordinates have proved a useful tool for visualising many objectives in a two-dimensional graph and the computational grid and wireless Personal Digital Assistants offer technological solutions to implementation difficulties arising in complex system design.

1 Introduction

Real-world engineering design problems often involve the satisfaction of multiple performance measures, or objectives, which should be solved simultaneously. Automotive and aerospace examples provide illustrations of some typical design challenges and demonstrate that these problems often involve a large number of objectives. It is demonstrated how a typical set of engineering design specifications might be mapped onto a familiar formulation of an EMO problem. EMO research has, for the most part, focused on problems having 2 or 3 objectives; however, in recent years there has been growing interest in the area of *many-objective* optimization where the problem might consist of 4 – 20 objectives, for example.

Of the three key requirements for EMO solution set quality - proximity, diversity and pertinency - a case is made that pertinency, focussing on solutions in the designer's region of interest, has a special prominence in *many-objective* optimization studies. A method whereby the MOEA is operated in an interactive manner through progressive articulation of preferences is described and an example worked through to explore the potential of this approach.

C. A. Coello Coello et al. (Eds.): EMO 2005, LNCS 3410, pp. 14–32, 2005.

The means of using the method of parallel coordinates to reduce the study of a *many-dimensional* Pareto front to a 2-D representation reveals a number of strengths and limitations. *Many-objective* optimization in an engineering design context is inevitably very compute-intensive and there is an expectation that a design procedure will often be time-consuming. Two schemes are introduced to deal with these demands. In one scheme, the MOEA is parallelised to execute effectively in a computational grid environment in reduced time. For the second scheme, it is shown that wireless PDAs can be effective tools for designers in the interactive computational steering of the design process.

2 Design Approaches

In this section we provide examples of conflicting objectives in two areas of engineering design and then provide some background to solution approaches used in the past.

2.1 Typical Engineering Design Optimization Problems

Automotive Engineering Examples

Historically, in automotive engineering, the process of establishing trade-offs has been to conduct parametric studies. That is, evaluating the conflicting objective functions at different values of the decision variables (parameters), comparing the results in objective space and then finally selecting a single trade-off solution. An example of such a parametric study is shown in an enumeration plot (see Fig. 1), where two conflicting objective functions (empirical models of NOx and Brake Torque) are plotted against each other, evaluated as a function of their input (decision) variables.

Brake Torque is a surrogate variable for fuel economy and is easily measurable on an engine dynamometer test rig; maximising brake torque is equivalent to optimizing fuel economy. NOx or Oxides of Nitrogen are one of the three legislated exhaust emission pollutants and in this case is measured pre-catalytic converter. Minimising NOx minimises the precious metal (e.g. Platinum or Rhodium) coating in the catalytic converter and thus cost.

In Fig. 1 the decision variable is EGR (Exhaust Gas Recirculation) rate, which gives a benefit in brake torque and NOx. Brake Torque maximises at moderate EGR rate, but NOx minimises at maximum EGR rate. Thus, the objectives conflict.

Since the optimization problem is to maximise Brake Torque and minimise NOx, trade-off solutions in the lower RH corner of Fig. 1 are preferred. Using a parametric approach, many objective function evaluations are required, which may be expensive, particularly if there are a large number of objective functions. (In automotive engineering it is not uncommon to have problems with 4-10 objectives.) Also, it is possible that suitable Pareto-optimal solutions will not be discovered and that a sub-optimal trade-off solution will be selected.

Fig. 1 is obtained from empirical models of Brake Torque and NOx. The models are based on automated test data arising from a designed experiment on an engine on an engine dynamometer test rig. The resulting models are then validated against independent test data and against known physical trends.

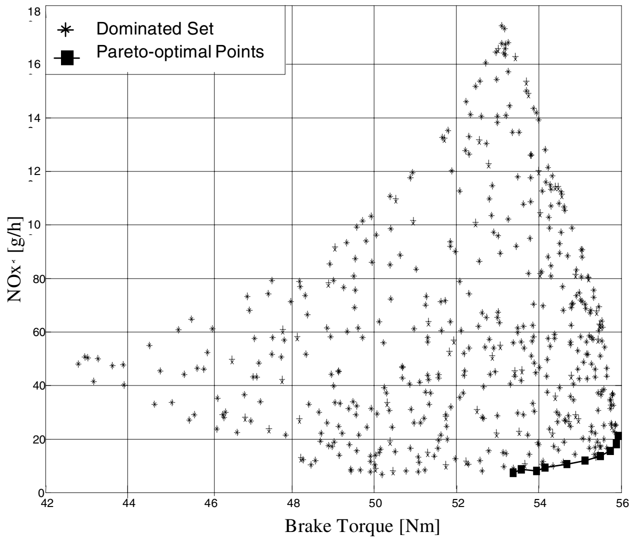

Fig. 1. Relationship between conflicting objective functions, NOx and Brake Torque

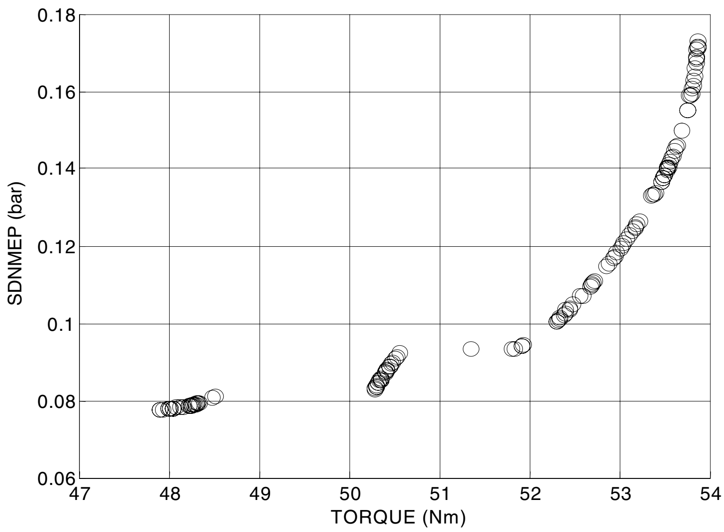

Fig. 2. Piece-wise continuous Pareto front obtained by comparing SDNMEP with torque

Such models offer practical advantages over an online approach (i.e. connecting an optimizer directly to the test rig) in terms of efficiency, re-use and a noise-free or repeatable objective function evaluation. Connecting the optimizer to these models provides an efficient and systematic search capability for Pareto-optimal solutions, which, by definition, represent optimal system capability that is of high engineering and business value.

Fig. 2 is an example of a piece-wise continuous Pareto front obtained by comparing SDNMEP (Standard Deviation of Net Mean Effective Pressure - a measure of combustion stability) with torque. Good combustion stability is necessary for engine smoothness, which is now a customer expectation of modern mass-produced engines. The decision variables, Intake Valve Opening (IVO) & Exhaust Valve Closing (EVC), are used to determine cam positions in a continuously variable Twin Independent Variable Cam Timing system. Torque maximises at moderate IVO + late EVC (medium overlap) or late IVO & EVC, whereas SDNMEP minimises at late IVO + early EVC (low overlap). Again, the objectives conflict.

Aerospace Engineering Examples

For different classes of gas turbine engines (GTEs) for aircraft propulsion, while the controller structure will often remain the same, there is a requirement for redesign of the control system for new engines and new performance requirements. (Besides aerospace, GTEs are also used for applications such as marine vehicle propulsion and power generation.) Each application requires the engine to operate in a specified manner and provide particular shaft velocities, efficiencies and thrust output. These depend on the purpose for which the engine is being used. One way of varying the performance of these engines is through the choice of a suitable control system that is able to regulate the fuel supplied to the combustion chamber of the engine and thus influences the performance of the engine.

Fig. 3 provides a typical set of design specifications (and, for later discussion, a mapping to a specific EMO treatment is provided). For example, objective (1) represents a measure of rapid acceleration for go-around/aborted landing, objective (2) represents a measure of rapid deceleration for stopping on runway for aborted takeoff, and objective (8) prevents flameout, a factor which is relevant on rapid decelerations, e.g. when decelerating at high altitude at the end of cruise or an aborted takeoff.

High-fidelity nonlinear dynamic models of engine performance exist and controller design is realized by implementing alternative strategies on these models for performance evaluation and comparison. Until recently based designs were limited to scalar objectives and, often, addressed the design specifications indirectly by employing measures that required weight manipulation to shape the response into the desired form. The availability of tools to directly address *multiple* objectives is leading to shorter design times and improved system performance.

Fig. 3 demonstrates how a conventional design specification can be mapped into a multiobjective optimization (MO) formulation. Subsequent sections will describe and illustrate how such a MO formulation, complete with goals, can be treated in a multiobjective optimization evolutionary algorithm (MOEA) framework and the designs executed using progressive preference articulation.

2.2 Requirements of a Multi-objective Optimizer for Engineering Design

The globally optimal trade-off surface of a multi-objective optimization problem can contain a potentially infinite number of Pareto-optimal solutions. The task of a multi-objective optimizer is to provide an accurate and useful representation of the trade- off surface to the decision-maker. The set of solutions generated by the optimizer is known as an approximation set [32]. Three aspects of solution set quality can be considered. These are listed below, and shown graphically in Fig. 4.

Proximity. The approximation set should contain solutions whose corresponding objective vectors are close to the true Pareto front.

Design Specifications

Obtain a controller such that

System is closed-loop stable

and

1) idle to 90% MTO < 6s
2) MTO to 15% < 6s
3) Thrust oscillations < 0.005 EPR
4) Bandwidth > 3 rad/s
5) Overshoot < 0.02 EPR
6) Maximum TET < 1960 K
7) Surge margin > 10%
8) Minimum FAR > 0.015

where
MTO = Max Take off Power.
EPR = Engine Pressure Ratio
TET = Turbine Entry Temperature
FAR - Fuel to Air Ratio

MOEA Formulation

Minimise the vector **f** , where

f = [idle to 90%MTO
MTO to 15%
Thrust oscillations
Bandwidth
Overshoot
Maximum TET
Surge margin
Minimum FAR]

with respect to goal vector g,
where
g = idle to 90%MTO < 6s
MTO to 15% < 6s
Thrust osc^{ns}< 0.005 EPR
- Bandwidth < -3 rad/s
Overshoot < 0.02 EPR
Max TET < 1960 K
- Surge margin < -10%
- Min FAR <-0.015

Fig. 3. Mapping of design specifications into a multiobjective evolutionary algorithm (MOEA) formulation

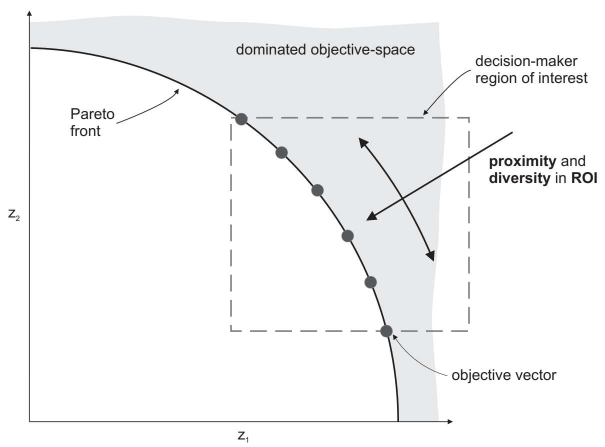

Fig. 4. The ideal solution to a multi-objective optimization problem

Diversity. The approximation set should contain a good distribution of solutions, in terms of both extent and uniformity. Good diversity is commonly of interest in objective-space, but may also be required in decision-space. In objective-space, the approximation set should extend across the entire range of the true Pareto front with a parametrically uniform distribution across the surface.

Pertinency. The approximation set should only contain solutions in the decision maker's (DM's) region of interest (ROI). In practice, and especially as the number of objectives increases, the DM is interested only in a sub-region of objective-space. Thus, there is little benefit in representing trade-off regions that lie outside the ROI. Focusing on pertinent areas of the search space helps to improve optimizer efficiency and reduces unnecessary information that the DM would otherwise have to consider.

2.3 A Brief Account of Multiobjective Optimization in Engineering Design

A typical control system design problem might be posed as follows. Given a system **x** = (**x**, **u**, *t*) where **x** and **u** are the system state and control vectors and is a vector non-linear function, find a controller **u** such that the design specifications,

$$f_i(\mathbf{x},\mathbf{u},t) \le g_i, i = 1,\ldots,m, \tag{1}$$

are satisfied, where g_i are the design goals. Such problems were most often addressed via optimization by aggregating objectives directly (or by indirect means) into a weighted sum, such as the following objective function used in linear quadratic regulator design [1],

$$J = \int_0^\infty \left\{\mathbf{x}^T Q\mathbf{x} + \mathbf{u}^T R\mathbf{u}\right\} dt, \tag{2}$$

where Q and R are user-selected weighting matrices to guide the design. The well-known drawbacks of the weighted sum approach are the difficulty in setting values for the weights, and the fact that the method has been proved to be incapable of generating solutions in non-convex regions of the trade-off surface [4]. It is also a single solution method, requiring multiple starts in attempts to build up trade-off information.

Zakian and Al-Naib proposed a method for obtaining a control vector, **u**, of pre-specified structure [31], which satisfied the design specifications/constraints eqn. (1). However, this was a constraint satisfaction approach that (i) did not attempt to *optimize* the solution once one was found that satisfied the constraints, and (ii) had no recovery strategy, should the solution space prove to be null.

In the goal attainment method [18], the designer is required to specify a set of goals, let us also call them **g,** for the objective function vector, **f.** The nonlinear programming problem to be solved is:

$$\text{Min } \lambda \text{, with } \lambda, \mathbf{p} \text{ subject to: } f_i - w_i\lambda \le g_i, i = 1,\ldots,m \tag{3}$$

where **p** is the decision variable vector (or controller parameters), $w_i \geq 0$ are weighting coefficients and $\geq$ is an unrestricted scalar variable. The quantity $w_i\,\lambda$ may thus be interpreted as the degree of under-attainment or over-attainment of the goal g_i. This method is not subject to the convexity limitations of the "weighted sum" approach and its use of weights and goal enables the designer to be more expressive and precise in directing the search. Goal expression has an affinity with the common form of engineering design specification (cf. Fig. 3). The goal attainment method is, though, irredeemably a scalar optimization method, capable of revealing only one solution on the Pareto front as a result of one pass of this algorithm. Nonetheless, it was to prove influential for Fonseca and Fleming as they refined their multiobjective genetic algorithm [14] for use in engineering design [15].

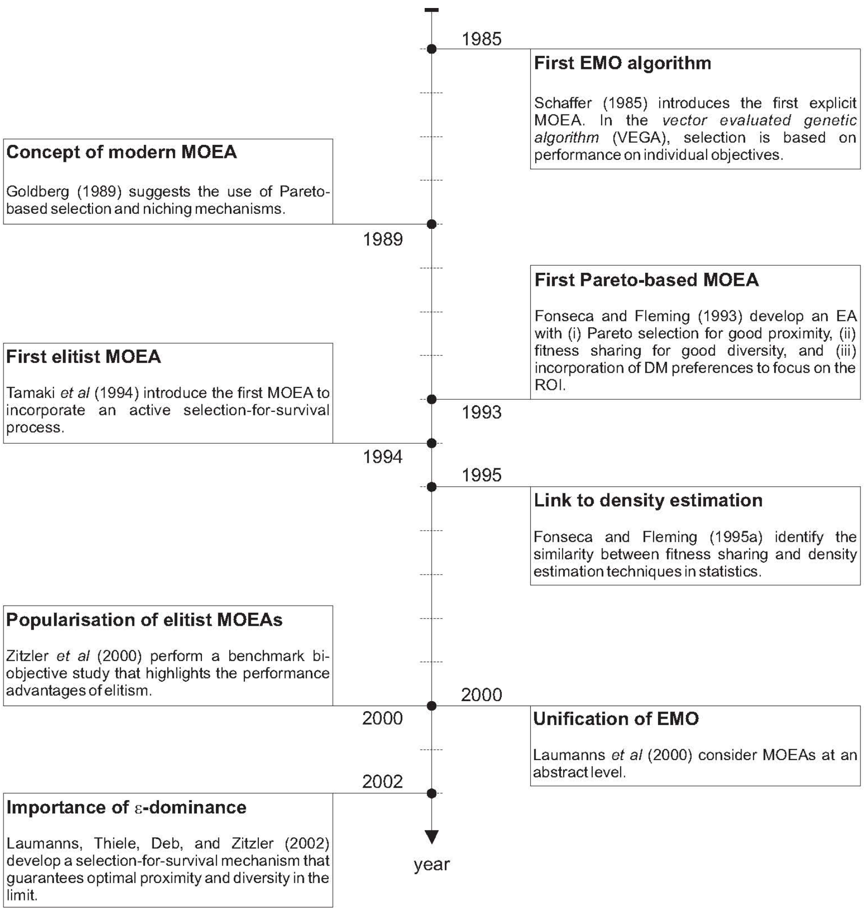

Fig. 5. Key developments in EMO history [24]

The population-based nature of evolutionary algorithms and their flexible selection mechanism have proved to be extremely successful for solving multi-objective optimization problems and for revealing a satisfactory approximation set to the desired globally optimal trade-off surface in a single execution of the algorithm. The fundamental benefit of this latter factor over multiple-start strategies is the potential for a cooperative search for ultimately different solutions, thus saving on the total number of solution evaluations required. Excellent descriptions of the history of EMO may be found, for example, in [8] and [6]. A timeline of key events is shown in Fig. 5.

3 *Many-Objective* Optimization

As we have seen in section 2, engineering design has a propensity to produce significant numbers of objectives, considerably in excess of 2 or 3 objectives commonly addressed by EMO researchers. This section looks at the new issues that arise and suggests way of tackling these.

3.1 Issues

Interactions often arise between objectives and these have been classified as *conflict* or *harmony* [23]. A relationship in which performance in one objective is seen to deteriorate as performance in another is improved is described as conflicting. A relationship in which enhancement of performance in an objective is witnessed as another objective is improved can be described as harmonious. The conflict that exists in a many-objective optimization task is a serious challenge for EMO researchers.

Given the typical numbers of requirements arising in engineering design, and elsewhere (see [6]), there is a very clear need to develop an understanding of the effects of increasing numbers of objectives on EMO. The phrase *many-objective* has been suggested in the OR community to refer to optimization problems with more than the standard two or three objectives [13].

For M conflicting objectives, an (M-1)-dimensional trade-off hypersurface exists in objective space. The number of samples required to achieve an adequate representation of the surface is exponential in M. In [9] it is shown that the proportion of locally non-dominated objective vectors in a finite randomly-generated sample becomes very large as the number of objectives increases. Since dominance is used to drive the search toward the true Pareto front, there may be insufficient selective pressure to make such progress. Of course, the use of a large population can help address this, but this is impractical for many engineering designs in which evaluation of objectives for a single candidate solution can be very compute-intensive.

Due to the ‘curse of dimensionality’ (the sparseness of data in high dimensions), the ability to fully explore surfaces in greater than five dimensions is regarded as highly limited [27]. Statisticians generally use dimensionality reduction techniques prior to application of the estimator. This assumes that the ‘true’ structure of the surface is of lower dimension, but the potential for reduction may be limited for a trade-off surface in which all objectives are in conflict with each other.

Possible measures that have been considered previously by the EMO community to address the issues arising from *many-objective* optimization include

- the use of preferences,
- aggregation,
- goals and priorities,
- dimension reduction, and
- visualisation.

Aspects of all of these measures are considered, both later in this section and in an example of use in the next section.

3.2 Preference-Based Methods

The exploitation of DM preferences, either *a priori*, *a posteriori*, or progressively, is arguably the current best technique for handling large numbers of conflicting objectives. In the *a priori* and progressive cases, the aim of EMO is to achieve a good representation of trade-off regions of interest to the DM (essentially limiting the ambition of the optimizer by requiring it to represent only a sub-space of the trade-off hypersurface).

In *a priori* schemes, DM preferences are incorporated before the search begins. In progressive methods, DM preferences are incorporated during the search. The key advantage of these techniques over *a priori* methods is that the DM may be unsure of his or her preferences at the beginning of the procedure and may be informed and influenced by information that becomes available during the search. The final class of methods is *a posteriori*, in which a solution is chosen from the approximation set returned by the optimizer.

Refer to [4] for a survey of the literature on preference methods for EMO. Progressive preference methods appear to hold much relevance for engineering design and are discussed below.

Fonseca and Fleming [15] possibly introduced the first truly progressive MOEA scheme as an extension to the Pareto-based ranking of their multiobjective genetic algorithm (MOGA) [14]. Using a preferability operator, the DM can set goal values and priority levels for any objective. This can be done at any time during the run of the MOEA and can be updated when required. The data feeds into a modified definition of dominance, which provides a unification of Pareto optimality, the lexicographic method, goal programming, constraint satisfaction, and constrained optimization. All these methods, plus hybrids, can be derived from the preference operator. A detailed explanation of the preferability operator is described in [21].

Deb *et al.* [10] developed a constrained-domination approach that is very similar to the preferability operator. The main distinction is that, in this new scheme, an overall quantity of goal violation is calculated. This enhances the amount of information available to the search, but requires the forced cohesion of objectives. Another similar scheme, known as favour, has been proposed by Drechsler *et al* [12]. Tan *et al.* [28] developed logical connectives to allow a DM to make alternative preference scenarios for a problem in the context of preferability-type schemes. Todd and Sen [29] proposed an alternative progressive scheme, which incorporates learning and automation of DM preferences. Rather than setting goals and priorities, the DM is asked to make judgements on a set of potential solutions at various intervals during the optimization process.

Cvetkovic and Parmee [7] have proposed an interactive scheme involving the combined use of a weighted Pareto front and a variable parameter representing the minimum level of dominance. Branke *et al.* [2] describe the Guided Multi-Objective Evolutionary Algorithm, where DM preferences are manifested through a modification of the dominance definition, which specifies the level of trade-offs acceptable among objectives, i.e. the maximum acceptable amount of degradation of an objective's performance is recompensed by a specified level of improvement of another conflicting objective's performance. In the biased crowding approach [3], Branke and Deb describe how the DM can control the ROI along a specific front and plane by specifying the value of a specific parameter, a "biased crowding measure" [11], which expresses the ratio of the true distances between neighbouring solutions on the true Pareto-optimal front and the projected distance of the same solutions on a plane with a user specified direction connoting a central linearly weighted utility function.

It is, however, the progressive articulation scheme [15], which is elaborated in Section 4.1.

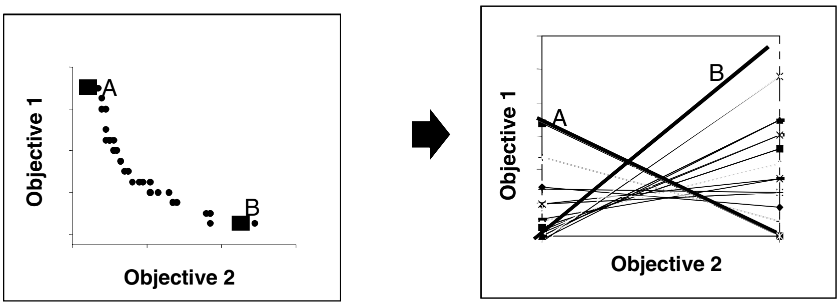

Fig. 6. Mapping between Cartesian system and corresponding parallel coordinate

3.3 Visualisation (Parallel Coordinates)

Fonseca and Fleming [15] used a user interface that featured the parallel coordinates method of visualising trade-offs between objectives [19]. The Cartesian system of having the axes orthogonal to each other has obvious limitations when trying to visualise geometry with higher than 3 dimensions or more than 3 variables in a set of data. Parallel coordinates give a systematic and rigorous way of representing the relationships between multiple variables – in our case, design objectives.

The approach of parallel coordinates places all the axes parallel to each other thus allowing any number of axes to be shown in a flat representation. Fig. 6 illustrates the mapping between the Cartesian system and the corresponding representation in parallel coordinates, where points A and B in the coordinate system are represented by lines in the parallel coordinates representation. Fig. 7 illustrates a representation that deals with more than two objectives (four objectives, in fact). Here, each line in the graph connects the performance objectives achieved by an individual member of the population and represents a potential solution to the design problem.

The order in which the axes are set out in parallel coordinates does not have any bearing on the translation of the data in to parallel coordinates, although it is not entirely neutral to the technique, a point that will be expanded upon later.

It is not sufficient just to be able to display multivariate data in a 2-dimensional representation. The key requirement is to be able to easily interpret the relationships between the variables. It can be shown that the geometrical features of a surface in n-dimensional space are preserved in the parallel coordinates system. This is important because it allows these features to be easily identifiable when represented in parallel coordinates and therefore the relationship between the variables that give rise to these features can be visualised. For example, in Fig. 7, "crossing lines" indicates conflict between the two adjacent objectives. The degree of conflict is demonstrated by the intensity, or degree to which, the lines cross. Conversely, lines that do not cross demonstrate objectives which are in relative harmony with one another.

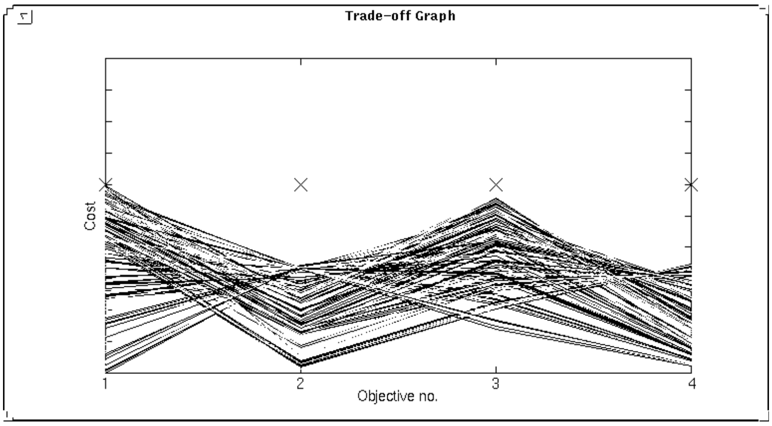

Fig. 7. Parallel coordinates for four objectives

Other requirements fulfilled by parallel coordinates are that there is no loss of data in the representation, which in turn ensures that there is a unique representation for each unique set of data. It also has a low representational complexity, O(n), where n is the number of variables modelled, allowing the technique to scale well to large numbers of variables. Weaknesses of this visualisation method, however, are (i) that it requires multiple views (different orderings of objectives) to see different trade-offs and (ii) that it can be hard to see what is going on when many vectors are represented. Wegman [30] describes some countermeasures to these problems.

4 Facilitating the Engineering Design Exercise

4.1 Use of Preference Articulation

We will now illustrate the way in which preference articulation, as described in [15], can be used in a *many-objective* problem to focus on a specific region of interest (ROI) on the Pareto front and, ultimately to isolate a desired design solution. An 8-

objective flight control system design problem is used to illustrate the process; for reasons of clarity and commercial sensitivity, the objectives are unspecified, having titles such as Objective 1, etc. The preference articulation sequence is illustrated in Fig. 8 (a-f). Selection throughout the progress of the optimization uses the preferability operator defined in [15].

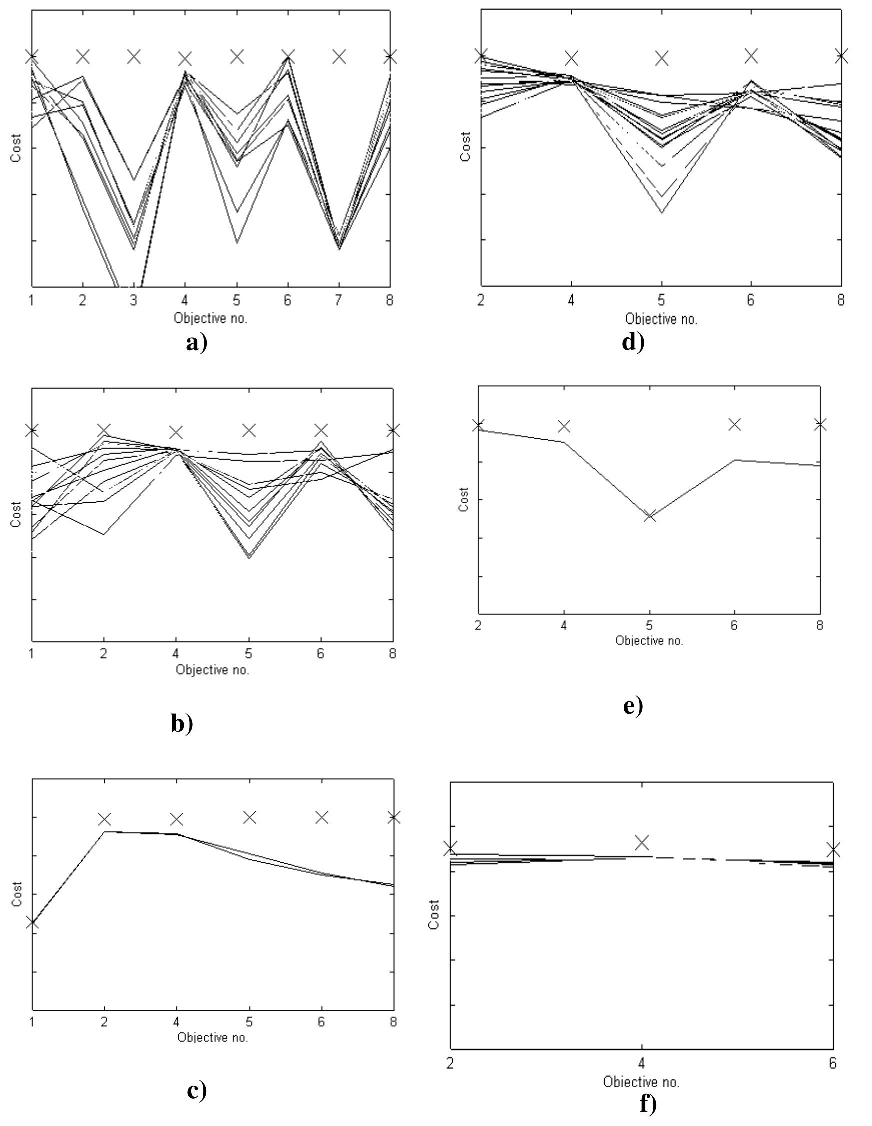

Fig. 8. Preference articulation sequence

Initially, the design is expressed as follows:

1. The designed controller seeks to simultaneously optimize *Objectives 1-8*
2. Each *Objective i* must satisfy *Goal i*, $i = 1,\ldots,8$
3. The controller has a prespecified structure with 7 variable parameters (*decision variables*)

Fig. 8(a) is a snapshot of the parallel coordinates representation of the eight objectives after a number of generations. Note that the initial design specification that each *Objective i* must satisfy *Goal i*, $i = 1\ldots8$ immediately imposes a strict ROI for the optimizer. Fig. 8(a) shows Pareto-optimal solutions obtained after running the optimizer for a number of generations. Goal points for each of the Objectives are marked with an “x” in the plots. Since this example will exercise progressive articulation of preferences, these goal values will be subject to change during the design process.

From the plot we can immediately see that Objectives 2 and 3 are in “harmony” and that the goal of Objective 7 is easily satisfied. There is also a suggestion that Objectives 4 and 5 might be in “harmony”.

The sequence that follows makes certain assumptions about the flight control system designer’s preferences but serves to provide an example of how the interactive preference articulation process can reduce the ROI, focus on key Objectives and, ultimately, identify an acceptable solution - in this case, a flight controller that satisfies initial design goals and is “optimal” with respect to the design objectives.

The representation in Fig. 8(a) leads to the first interactive design decision: to remove Objective 3 from further consideration since it will benefit from improvements in Objective 2 and the latter is the more important of these two objectives. Objective 7 is converted to be a constraint; its value is constrained to be at least as good as the “worst” solution for that objective shown in Fig. 8(a). The optimizer is now released to cycle through more generations. Fig. 8(b) is a snapshot of the parallel coordinates representation of the remaining six objectives under consideration, after further runs of the optimizer.

A decision is made now to isolate the best solution so far with respect to Objective 1, see Fig. 8(c). The designer knows that this objective has a strong cost impact on the final solution although modest improvements beyond the best case here are likely to have little further impact. A decision is now made to convert Objective 1 to be a constraint, where its value must be at least as good as the isolated solution.

Fig. 8(d) is a snapshot of the parallel coordinates representation of the remaining five objectives under consideration after further runs of the optimizer. Now is the time to reduce the ROI still further. Isolating the best solution so far with respect to

Objective 5, see Fig. 8(e), and converting this objective into a constraint in a similar way to that of Objectives 1 and 7 achieves this. Observing this isolated solution, a decision is also made that Objective 8 can similarly be converted to a constraint, provided that its value is at least as good as this solution.

To arrive at Fig. 8(f), the goals of Objectives 2, 4 and 6 are progressively tightened until a very small set of solutions are obtained with little to discriminate them. At this point, the designer selects the best of these solutions with respect to Objective 2 (there is a slight preference with regard to the importance of this objective) as the desired solution.

4.2 Grid-Enabled EMO

The Grid computing paradigm is a recent development that enables complex system designers seeking to accelerate EMO solutions. The computational grid is a "hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities." [17]. Originally motivated by "big science" with high-performance computing requirements, the Grid lends itself to EMO applications whose objective function evaluations are sufficiently compute-intensive.

MOGA-G [25] is a grid-enabled framework for EMO. For non-trivial objective functions, EMO is compute-intensive since designs normally require a relatively large number of evaluations of the objective function to produce a satisfactory result. Furthermore, objectives arising in engineering designs often require considerable computational effort, for example involving nonlinear dynamic simulations. The population-based nature of evolutionary algorithms means that they are well suited for parallelism using the master-worker paradigm (see Fig. 9). Here, EMO operations (ranking, crossover, mutation, fitness sharing, etc.) are performed by the Master Node, and the evaluations of the objective function are executed in parallel on the Worker Nodes.

MOGA-G implements the master-worker paradigm in a Service Orientated Architecture (SOA). This is the view of grid computing taken by the Globus Project (globus.org) and focuses on providing access to the resources of the grid via services. The main advantages of using this approach are:

- suitability to the proposed form of parallelism,
- flexibility of use,
- interoperability with current (and, hopefully, future) standards, and
- the modular nature of the Globus Toolkit.

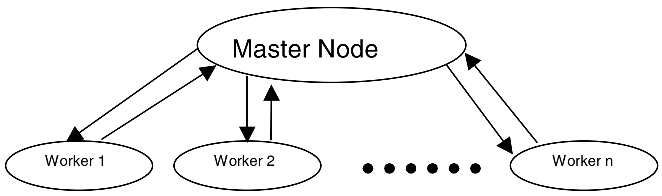

Fig. 9. Master-worker paradigm

Using the SOA approach, the client acts as the master node and the service acts as the worker. In the implementation of the MOGA-G framework (see Fig. 10) there are two different services. One service exposes the operations of the multi-objective evolutionary algorithm to the client, and the other provides operations for evaluating the objective function. Compute-intensive function evaluations can therefore be "farmed" out to a user-specified number of computing nodes on the Grid, often geographically distributed.

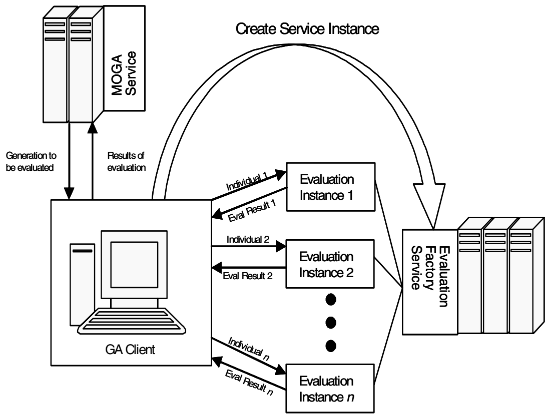

Fig. 10. MOGA-G implementation

The provision of these tools as services means that they can be accessed via the *http* protocol, and therefore via any device with a capable web browser such as a PDA Personal Digital Assistant (PDA). This flexibility stems from the use of easily accessible protocols like *http* and the loosely coupled nature of the SOA approach. For computationally trivial objective functions the communication overheads involved in executing the evaluations result in a decrease in performance. However, the framework shows significant performance advantages for more computationally complex objective functions such as nonlinear dynamic simulations.

4.3 Computational Steering of EMO-Based Engineering Designs Using a PDA

Large-scale, long-running, complex optimization routines, such as EMO, are usually run non-interactively. Typically, in an engineering design, the user will set the initial EMO parameters and then execute the algorithm. During this execution process, which can often take hours or days to complete, user interaction, if any, is limited to periodic interventions and the possible termination of the algorithm if it appears to have failed (for example, if the search process does not show convergence). When the execution is finished, the solutions produced by the algorithm are assessed and, if the design results are not satisfactory, the parameters of the algorithm are adjusted and it is run again. This process clearly leads to a very inefficient use of resources, and possibly, ultimately, to unsatisfactory solutions.

One solution to this problem is to allow the designer to interact with the optimization routine during execution (referred to as computational steering). This would allow the designer to influence the efficiency of the algorithm and the quality of the solutions that it produces. To enable interaction with the search process the user must be provided with an appropriate visualisation of the data so they can efficiently extract the relevant information [22].

A PDA-based client [26] has been developed to control this steering process (see Fig. 11). The client has to be stateless so that when there is no interaction from the user the optimization routine will run in batch mode. The PDA client provides an interface for observing the progress of the optimization routine (using a parallel coordinate plot, for example) and adjusting the parameters of the algorithm, if necessary. Due to issues related to scarcity of memory and computational power, plus small display size, a minimal interface has been developed, while still providing the desired functionality. It connects wirelessly to a web-service that allows the steering of the optimization algorithm. This steering web-service exposes methods for obtaining the current values of the candidate solutions and adjusting the parameters of the algorithm. Further development will allow the steering service to 'push' information to a client if one is connected.

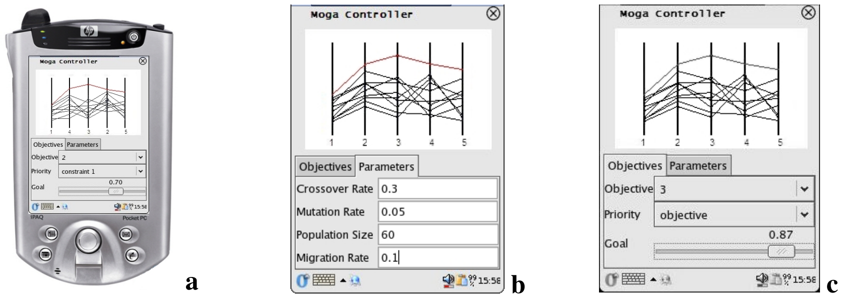

Fig. 11. a) PDA client for steering an EMO-based design; b) a parameter adjustment screen; c) a preference articulation screen

Steering of EMO can be performed in two main ways. Firstly, the internal parameters of the algorithm (such as crossover rate, mutation rate, etc.) can be adjusted from the steering client, see Fig. 11(b). Adjusting these parameters can alter the behaviour of the algorithm, such as speeding up or slowing down convergence. The second method of steering EMO is to alter the goal and priority information for the objectives, see Fig. 11(c). This information is used by the preferability operator [15]. Refining this preference information can help focus the algorithm on to a specific region of the non-dominated set. In this manner it is possible to guide the search and reduce the number of candidate solutions in the manner described in section 4.1.

5 Concluding Remarks

A particular application of EMO has been studied: design of engineering systems. These systems are often complex and, invariably, consist of many objectives. Large numbers of objectives present special problems for MOEAs and certain approaches have been advocated. The use of goals and a preferability operator in a progressive articulation of preferences setting have been demonstrated to be effective in selectively reducing the region of interest in a *many-objective* search, mitigating the prevailing lack of selective pressure. Through this approach, methods of reducing the dimensionality of the problem have been introduced. A special visualisation approach has assisted this development. The mix of complex systems and objectives and population-based search inevitably poses heavy computational demands on the design process and schemes have been described to address this. Grid computing affords a means of speeding up the search and remote computational steering is a valuable addition to the designer's toolset. Many other schemes could have been described that reduce computational load by introducing means of approximating objective functions. These range from response surface models, neural networks to a knowledge-based Kriging model (see [20] for an interesting comparison of methods). Inevitably, these and other approaches which assist the design engineer in *many-objective* optimization could not be included here but we trust that the sample of approaches described serve to stimulate interest in this fascinating and challenging area of EMO research.

Acknowledgements

We wish to acknowledge the contributions of past and present members of the Evolutionary Computation Group and Rolls-Royce University Technology Centre for Control and Systems Engineering at University of Sheffield, especially Carlos Fonseca, Andy Chipperfield, Haydn Thompson, Ian Griffin, Salem Adra, David Kumar, Alex Shenfield and Muhammad Alkarouri. We are also grateful for the financial support of Rolls-Royce plc and UK Engineering and Physical Science Research Council (latterly under Grant Number GR/R67668/01).

References

1. Athans M., Falb P.: *Optimal Control*, McGraw-Hill, New York (1966)
2. Branke, T., Kaußler, T., Schmeck, H.: Guidance in evolutionary multiobjective optimization. Advances in Engineering Software, (2001) 32: 499-507
3. Branke, J., Deb, K.: Integrating User Preferences into Evolutionary Multi-Objective Optimization. KanGal Report Number 2004004
4. Censor, Y.: Pareto optimality in multiobjective problems, Applied Mathematics and Optimization 4, 41–59 (1977)
5. Coello, C. A. C.: Handling preferences in evolutionary multiobjective optimization: a survey, in IEEE Neural Networks Council (ed.), Proceedings of the 2000 Congress on Evolutionary Computation (CEC 2000), Vol. 1, IEEE Service Center, Piscataway, New Jersey, pp. 30–37 (2000)
6. Coello, C. A. C., Veldhuizen, D. A. V., Lamont, G. B.: Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, New York (2002)
7. Cvetkovic, D., Parmee, I C.: Preferences and their application in evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation (2002) 6(1): 42-57

8. Deb, K.: Optimization for engineering design: Algorithms and examples. New Delhi: Prentice-Hall (1995)
9. Deb, K.: 2001a, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Chichester, UK (2000)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA–II, IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
11. Deb, K.: Multi-objective evolutionary algorithms: Introducing bias among Pareto-optimal solutions. In Ghosh, A and Tsutsui, S., editors, Advances in Evolutionary Computing: Theory and Applications. Springer-Verlag, London (2003) pages 263-292
12. Drechsler, N., Drechsler, R., Becker, B.: Multi-objective optimisation based on relation favour , in E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello and D. Corne (eds), Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001), Springer-Verlag, pp. 154–166 (2001)
13. Farina, M., Amato, P.: On the optimal solution definition for many-criteria optimization problems, in J. Keller and O. Nasraoui (eds), Proceedings of the 2002 NAFIPS-FLINT International Conference, IEEE Service Center, Piscataway, New Jersey, pp. 233–238 (2002)
14. Fonseca, C. M., Fleming, P. J.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, in S. Forrest (ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kau_man Publishers, San Mateo, California, pp. 416–423 (1993)
15. Fonseca, C. M., Fleming, P. J.: 1998a, Multiobjective optimization 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), 26–37 (1998).
16. Fonseca, C. M., Fleming, P. J.: 1998b, Multiobjective optimization and multiple constraint handling with evolutionary algorithms — Part II: An application example, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 28(1), 38–47 (1998)
17. Foster, I., Kesselman, C. (eds): The Grid: Blueprint for a Future Computing Infrastructure. Morgan Kaufmann (1998)
18. Gembicki FW. : Vector optimization for control with performance and parameter sensitivity indices, PhD Dissertation, Case Western Reserve Univ., Cleveland, Ohio, USA (1974)
19. Inselberg, A.: The plane with parallel coordinates, The Visual Computer 1, 69–91 (1985)
20. Leary, S.J., Keane, A. J.: Global approximation and optimisation using adjoint computational fluid dynamics codes, AIAA journal, 42 (3): 631-641 (2004)
21. Lygoe, R. J. and Fleming, P. J.: An Understanding of Fonseca & Fleming's Preferability Operator with respect to the Decision Making Process in Multi-Objective Optimisation, ACSE Research Report 880, University of Sheffield, Sheffield, UK.
22. Parker SG, Johnson CR, Beazley D: Computational steering software systems and strategies, IEEE Computational Science & Engineering 4 (4): 50-59 (1997)
23. Purshouse, R. C., Fleming, P. J.: 2003a An adaptive divide-and-conquer methodology for evolutionary multi-criterion optimisation, in C. M. Fonseca, P. J. Fleming, E. Zitzler (2003)
24. Purshouse, R. C.: On the Evolutionary Optimisation of Many Objectives, PhD thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK (2003)
25. Shenfield, A., Fleming, P. J.: A service oriented Architecture for Engineering Design Technical Report 877, Department of Automatic Control and Systems Engineering, University of Sheffield, UK
26. Shenfield, A., Alkarouri, M., Fleming, P. J.: Computational Steering of a Multi-Objective Genetic Algorithm using a PDA, 878, Department of Automatic Control and Systems Engineering, University of Sheffield, UK

27. Scott, D. W.: Multivariate Density Estimation: Theory, Practice, and Visualization, Wiley, New York (1992)
28. Tan, K. C., Khor, E. F., Lee, T. H., Sathikannan, R.: An evolutionary algorithm with advanced goal and priority specification for multi-objective optimization, Journal of Artificial Intelligence Research 18, 183–215 (2003)
29. Todd, D. S., Sen, P.: Directed multiple objective search of design spaces using genetic algorithms and neural networks, in W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela and R. E. Smith (eds), Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO 99), Vol. 2, Morgan Kaufmann Publishers, San Francisco, California, pp. 1738–1743 (1999)
30. Wegman, E. J.: Hyperdimensional data analysis using parallel coordinates, Journal of the American Statistical Association 85, 664–675 (1990)
31. Zakian V and Al-Naib U: Design of dynamical control systems by the method of inequalities, *Proc. IEE*, Vol 120, pp1421-1427 (1973)
32. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)