Multiobjective Satisfaction within an Interactive Evolutionary Design Environment

Ian C. Parmee
Plymouth Engineering Design Centre
University of Plymouth
Drake Circus
Plymouth PL4 8AA, UK
iparmee@plymouth.ac.uk

Dragan Cvetković
Plymouth Engineering Design Centre
University of Plymouth
Drake Circus
Plymouth PL4 8AA, UK
dcvetkovic@plymouth.ac.uk

Andrew H. Watson
Plymouth Engineering Design Centre
University of Plymouth
Drake Circus
Plymouth PL4 8AA, UK
awatson@plymouth.ac.uk

Christopher R. Bonham
Plymouth Engineering Design Centre
University of Plymouth
Drake Circus
Plymouth PL4 8AA, UK
cbonham@soc.plym.ac.uk

Abstract

The paper introduces the concept of an Interactive Evolutionary Design System (IEDS) that supports the engineering designer during the conceptual/preliminary stages of the design process. Requirement during these early stages relates primarily to design search and exploration across a poorly defined space as the designer's knowledge base concerning the problem area develops. Multiobjective satisfaction plays a major role, and objectives are likely to be ill-defined and their relative importance uncertain. Interactive evolutionary search and exploration provides information to the design team that contributes directly to their overall understanding of the problem domain in terms of relevant objectives, constraints, and variable ranges. This paper describes the development of certain elements within an interactive evolutionary conceptual design environment that allows off-line processing of such information leading to a redefinition of the design space. Such redefinition may refer to the inclusion or removal of objectives, changes concerning their relative importance, or the reduction of variable ranges as a better understanding of objective sensitivity is established. The emphasis, therefore, moves from a multiobjective optimization over a preset number of generations to a relatively continuous interactive evolutionary search that results in the optimal definition of both the variable and objective space relating to the design problem at hand. The paper describes those elements of the IEDS relating to such multiobjective information gathering and subsequent design space redefinition.

Keywords

Interactive evolutionary computing, multiobjective optimization, engineering design.

1 Introduction

Although application of evolutionary and adaptive computing technologies for design optimization is well-established, there is little recognition of their potential for design exploration through appropriate integration with conceptual design processes. Such integration supports search across predefined design spaces while allowing exploration outside of initial constraint and variable parameter bounds. Close designer interaction allows exploration involving off-line processing of initial results that leads to a redefinition of the design space. Further designer/evolutionary search of redefined space can lead to the discovery of innovative or even creative solutions. Strategies that support such exploration within the framework of an experimental interactive evolutionary design system (IEDS) are introduced.

It is evident from recent research (Parmee, 1999) that the following aspects must be considered if efficient interactive exploratory design processes are to be established:

- The ability to efficiently sample complex design spaces described by differing model representation (e.g., quantitative and qualitative);
- the addition, removal, and/or variation of constraints, objectives, and variable parameter bounds;
- the rapid identification of high-performance regions of complex spaces;
- the development of search/exploration systems that can capture design knowledge through extensive designer interaction;
- the interactive, on-line processing of information relating to multiple design, manufacturing, economic, and marketing criteria;
- the ability to define regions of design feasibility and to identify optimal solutions within them.

Entirely machine-based conceptual design is not suggested here nor currently considered viable. Best utility can be achieved from systems that enhance the designer's inherent capabilities. Appropriate integration can result in the development of prototype evolutionary design tools that provide powerful extensions to design team activity by supporting rapid, extensive exploration and stimulating innovative reasoning. A major conceptual design characteristic is the presence of many quantitative and qualitative criteria. Many techniques exist for combining objective vectors into a single scalar representation (Osyczka, 1984) or, alternatively, a Pareto frontier (Horn and Nafpliotis, 1993) containing solutions that best satisfy a range of objectives can be identified. Such techniques, however, assume that all criteria are quantifiable, well-defined, and that their relative importance remains constant. This is likely not the case during conceptual design. The proposed approach relates to designer support through the provision of succinct design information regarding multiple qualitative and quantitative objectives.

The current envisaged structure of the IEDS is shown in Figure 1. At its core are a number of coevolutionary processes concerning closely related but differing aspects of the design problem at hand. Scenarios A, B, and C, for instance, may relate to mechanical, aerodynamic, and thermal aspects of some element of preliminary gas turbine design or various operational scenarios relating to preliminary airframe design. In order to maintain their concurrent evolution, a communication protocol relating to conflicting constraint and objective satisfaction must be maintained. Current investigation of emerging intelligent agent technology is ascertaining their potential for providing such communication control. The design team interacts with the process either directly or through the revision of rule-based preference ratings. Information can also be archived in the on-line database and extracted when required. Cluster-oriented genetic algorithms (COGA) (Parmee, 1996) are an example of information gathering processes constantly extracting relevant information from the search processes and presenting this to the design team via machine-based agents.

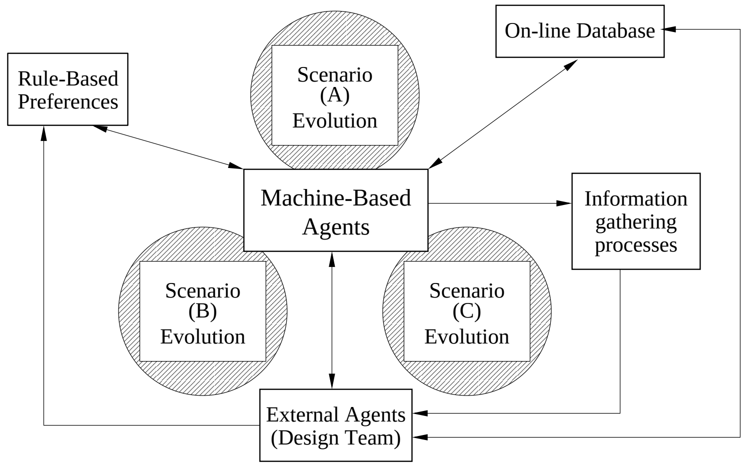


Figure 1: The proposed evolutionary design system.

The designers are regarded as external agents extracting information and processing this off-line. A dynamic redefinition of the problem based upon this processing is considered possible via reformed constraint, criteria, and objective functions. The design team rapidly assesses design concepts by utilizing the search and decision support capabilities of the evolutionary design system. Off-line discussion redefines the problem space while adding to the engineer's problem-specific knowledge base. It is proposed that an interactive cyclic process that allows a far broader search of design alternatives within an acceptable time frame can thus be achieved.

It is stressed that the perceived strengths of the proposed system lie within the interactive information gathering and processing activities rather than direct optimization of well-defined multiobjective problems. It is not, therefore, wholly appropriate to compare the performance of the developed techniques with that of established multiobjective genetic algorithm (MOGA) approaches, which, generally, address routine design problems.

Evaluation of the research should be assessed in terms of the following objectives:

- the establishment of novel techniques that not only identify high-performance solutions that provide best compromise in terms of a number of conflicting objectives but also concurrently gather additional information relating to both variable and objective space.

- the integration of such techniques within a framework that supports communication between them and interaction with the designer.

The satisfaction of such objectives should directly contribute to the overall goal—the development of a system that allows a far higher degree of conceptual design search and exploration within acceptable design lead times while providing an environment that supports design innovation and creativity.

1.1 The Airframe Preliminary Design Model

The above concepts have been applied within a military airframe preliminary design environment. This is a complex design domain characterized by uncertain requirements and fuzzy objectives relating to long gestation periods between initial design brief and realization of product. Changes in operational requirements and technological advances require a responsive, highly flexible strategy where design change and compromise are inherent features for much of the design period. Design exploration leading to innovative and creative activity must be supported. The introduction of rapid change to satisfy operational, engineering, and marketing considerations as they themselves change is essential.

A conceptual design model has been developed in collaboration with British Aerospace (BAe). The miniCAPS model is a version of Computer Aided Project Studies (CAPS) (BAe software used by designers during the earliest investigation stages of a new aircraft). MiniCAPS reproduces the general characteristics of CAPS but without the computational complexity. At present miniCAPS utilizes nine variable parameters producing a total of 13 outputs, each of which may be considered an objective. Variable parameters and outputs from the model are listed in Table 1.

Table 1: Model variables and outputs.

Input Parameter	
1	Climb Mach Number
2	Cruise Height
3	Cruise Mach Number
4	Gross Wing Plan Area
5	Aspect Ratio
6	Wing Taper Ratio
7	Wing LE Sweep
8	Wing Tip/Chord Ratio
9	Bypass Ratio

Output Parameter	
1	Take off Distance
2	Landing Speed
3	Specific Excess Power 1
4	Specific Excess Power 2
5	Sustained Turn Rate 1
6	Sustained Turn Rate 2
7	Attained Turn Rate 1
8	Attained Turn Rate 2
9	Ferry Range
10	Mass Take-off
11	Wing Span
12	Chord/Fuselage length
13	Ground attack mission

MiniCAPS models a variety of disciplines and consists of three modules: **Aerodynamics** (lift and drag coefficients, flight envelope, etc.), **Performance** (ferry range, sustained turn rate, take-off distance, cruise height, etc.), and **Configuration** (wing position, wing shape, canard position, number of engines, mass estimation, etc.).

A high degree of interaction is incorporated between these disciplines and many of the objectives are thus highly conflicting.

1.2 IEDS Components

Results relevant to multiobjective optimization are presented from some of the components of the IEDS system shown in Figure 1, and their complementary implementation is illustrated. These components are:

1. **Rule-based preferences:** A simple mathematical language facilitates direct preference manipulation by the engineering designer. Ranked preferences relating to operational requirements, multiobjectives and/or multidisciplinary constraint can be introduced and altered on-line. (Cvetković and Parmee, 1999).

2. **Multiobjective convergence:** There is a major requirement for satisfaction of large numbers of conflicting objectives (i.e., > 10). Coevolutionary strategies have been established that support the identification of high-performance regions of a multidimensional Pareto frontier. This eliminates problems associated with the generation and subsequent search of large data sets that describe entire high-dimensional Pareto surfaces (Parmee and Watson, 1999).

3. **Dynamic problem decomposition:** On-line techniques that, during a coevolutionary process, identify the sensitivity of several differing objectives to individual variable parameters have been developed. This allows the identification of variables (or specific ranges of variable values) that have little effect upon individual objective satisfaction and the subsequent conversion of such variables to fixed parameters. This reduces the variable space and facilitates the satisfaction of conflicting multidisciplinary objectives (Parmee and Watson, 1999).

4. **Identification of high-performance regions:** COGA strategies for the identification of high-performance design regions relating to single- or multiobjectives are now well established. Relevant design information can be extracted from such regions and presented to the engineer (Parmee and Bonham, 1998). These strategies can be included in the information gathering component of the IEDS. Those COGA capabilities directly relating to multiobjective satisfaction are described in Section 5.

Each of the current IEDS components is described in detail in the following sections and preliminary results of their application in the BAe airframe design domain are presented.

2 Multiobjective GA Search

Various methods have been employed for multiobjective optimization, including aggregating functions and Pareto approaches (Goldberg, 1989) that utilize GA search capabilities in addition to a small number of techniques that are entirely GA based. Some of these techniques provide single-objective optimal solutions while others define a nondominated solution frontier.

Aggregating functions include weighted sum methods, where the user assigns each objective, and the total fitness is the sum of all the weighted fitness values (Jakob et al., 1992). These methods will not produce a trade-off front unless many differing weight combinations are processed.

The Vector Evaluated Genetic Algorithm (VEGA) (Schaffer, 1984) uses subpopulations generated by performing proportional selection according to each objective in turn. A new generation is obtained by allowing crossover between these subpopulations. Other non-Pareto approaches include lexicographic ordering (Ben-Tal, 1979), evolutionary strategies (Kursawe, 1991), and weighted sum methods with sharing (Hajela and Lin, 1992).

Pareto-based approaches include Pareto-based fitness assignment using nondominated ranking and selection. The MOGA (Fonseca and Fleming, 1995) proposes a scheme in which the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. Other techniques include the Nondominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb, 1995) and the Niched Pareto GA (NPGA) (Horn and Nafpliotis, 1993). No single solution is given when using Pareto methods. The designer must choose an appropriate design point or region within the identified Pareto front.

Algorithms are, therefore, available that identify a single design solution satisfying a number of objectives or others that produce a Pareto front. It is suggested that an ideal for multiobjective optimization within preliminary design would be an algorithm that produces single-objective, high-performance solutions, the Pareto front, and, through designer interaction, an optimal solution to the problem at hand. This ideal is supported by the strategies of the following sections.

3 Use of Preferences in Multiobjective Optimization

3.1 Introduction

In multiobjective optimization we have a function to optimize:

DEFINITION 1: *Let* $n > 0, k > 0, D = X_1 \times X_2 \times \ldots \times X_n \subseteq \mathbf{R}^n$, *and* $R = Y_1 \times Y_2 \times \ldots \times Y_k \subseteq \mathbf{R}^k$. *Let further* $f_i : D \mapsto Y_i$ *for* $1 \leq i \leq k$, *and finally* $\mathbf{F} : D \mapsto R$ *so that* $\mathbf{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$.

The goal is to optimize function $\mathbf{F}(\boldsymbol{x})$ under constraints, i.e.,

$$\max_{\boldsymbol{x}} \mathbf{F}(\boldsymbol{x}) \quad \text{s.t.} \quad g_1(\boldsymbol{x}, \boldsymbol{p}) \leq 0, \ldots, g_l(\boldsymbol{x}, \boldsymbol{p}) \leq 0 \tag{1}$$

where $\boldsymbol{p} = (p_1, \ldots, p_u)$ are additional (real-valued) parameters. This problem is well-known, and a number of nongenetic (Hwang and Masud, 1979) and genetic algorithm (Van Veldhuizen, 1999; Deb, 1999) approaches exist. Another problem is that not all objectives are equally important, which necessitates the use of weights or preferences.

The designer cannot always completely objectively define the preferences regarding the objectives that have to be optimized (Nisbett and Wilson, 1977, 254). A common situation is a subjective statement, "objective A is much more important than objective B" but without any quantitative representation. One method for overcoming this problem is fuzzy multiple objective optimization (Lai and Hwang, 1996). In this section, this problem is addressed in a different manner, and developed methods are integrated with different GA-based optimization techniques. A weights-based Pareto optimization method is discussed, and a method for weights setting is described. Examples are provided.

3.2 Pareto Optimization with Weights-based Methods

When comparing Pareto-based multiobjective optimization with lexicographic order-based optimization (Ben-Tal, 1979), two extremes concerning objective importance are evident: in the case of Pareto optimization, all objectives are considered simultaneously (and equally important), whereas in the case of lexicographical order, the objectives are considered sequentially (the first objective is the most important and only if the same results are achieved for the first objective, is the second objective considered, etc.). In this section, an optimization method is developed that is based on the Pareto principle but where the relative importance of objectives can be specified.

As in the case of weighted sums-based methods, the relative importance of objectives in this modified Pareto method could be specified using weights (quantitatively), or they could be combined with a preference method that would translate qualitative specification into quantitative. Without this combination, the modified Pareto method described below would suffer from the same problem as the weighted sum method, i.e., how to specify weights in the case of 15–20 or more objectives.

3.3 Definition of the Modified Pareto Method

Lin (1976) defines the following orders on k-dimensional vectors:

$$\boldsymbol{x} \geq \boldsymbol{y} \quad \text{if and only if} \quad (\forall i \leq k)(x_i \geq y_i) \tag{2}$$

$$\boldsymbol{x} \geqslant \boldsymbol{y} \quad \text{if and only if} \quad (\boldsymbol{x} \geq \boldsymbol{y}) \wedge (\exists j \leq k)(x_j > y_j) \tag{3}$$

The orders (2) and (3) are definable in terms of each other (Lin, 1976, 46) and, thus, ordering a set in $\mathbf{R}^k$ by $\geqslant$ is equivalent to ordering a set by $\geq$. Usually, the definition of Pareto nondominance ($\succeq$) is given using (3), but according to the above remark, we might as well use (2).

We can write (2) in the following way:

$$\boldsymbol{x} \succeq \boldsymbol{y} \Leftrightarrow \frac{1}{k}\sum_{i=1}^{k} I_{\geq}(x_i, y_i) \geq 1, \qquad \text{where } I_{\geq}(x,y) = \begin{cases} 1, & x \geq y \\ 0, & x < y \end{cases} \tag{4}$$

To generalize (4), assuming $\sum_{i=1}^{k} w_i = 1$:

$$\boldsymbol{x} \succeq_w \boldsymbol{y} \text{ if and only if } \sum_{i=1}^{k} w_i \cdot I_{\geq}(x_i, y_i) \geq 1, \tag{5}$$

or some threshold $\tau \leq 1$ can be introduced:

$$\boldsymbol{x} \succeq_w^{\tau} \boldsymbol{y} \text{ if and only if } \sum_{i=1}^{k} w_i \cdot I_{\geq}(x_i, y_i) \geq \tau \tag{6}$$

DEFINITION 2: *Let relation* $\succeq_w$ *be defined by* (5) $\boldsymbol{w}$–nondominance *and the relation* $\succeq_w^{\tau}$ *be defined by* (6) $(\boldsymbol{w}, \tau)$–nondominance.

Note: The standard dominance relation is just a special case of (6) for $\boldsymbol{w} = (\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k})$ and $\tau = 1$.

Note: The relation $\succeq_{\boldsymbol{w}}^{\tau}$ is transitive as a product of transitive (component-wise) orders and has all the usual features of an order relation. Also, it is assumed that the weights do not change during the optimization process.

DEFINITION 3: *The* Pareto front *is defined as a maximal set of nondominated elements (according to a given order* $\succeq$*), and this definition is naturally extended to* $\mathbf{w}$*-Pareto fronts and to* $(\boldsymbol{w}, \tau)$*-Pareto fronts for a given vector of weights* $\mathbf{w}$ *and threshold* τ, *i.e., according to the order* $\succeq_{\mathbf{w}}$ *and* $\succeq_{\mathbf{w}}^{\tau}$ *given by* (5) *and* (6), *respectively. It is assumed that at least one of the inequalities is strict.*

Vector $\boldsymbol{w}$ could be either specified directly by the designer, or it can be calculated from personal preferences that help the designer to work in more qualitative terms. In the next section, a method for qualitative weights setting is presented (Cvetković and Parmee, 1999).

3.4 Fuzzy Preferences and Orders

Fuzzy preferences can support the estimation of the relative importance (weights) of objectives in a multiobjective optimization problem.

3.4.1 Preference Order

Given a fuzzy preference R (Fodor and Roubens, 1994) and its existing complete (fuzzy-) preference matrix, a definition of a complete order among the objects is possible in the following way:

Relation R defines the directed weighted graph $G = (A, R)$ and *entering score*, *leaving score*, *net flow*, and the corresponding orders can be defined (Fodor and Roubens, 1994, 151). In the case when $R(a,b) + R(b,a) = 1$ for all a, b (probabilistic relation), they all give the same order, so, as the relations will satisfy this property, we will concentrate on the leaving score and the induced order only:

$$S_L(a,R) \stackrel{\text{def}}{=} \sum_{c \in A \setminus \{a\}} R(a,c) \tag{7}$$

$$a \geq_L b \quad \text{if and only if} \quad S_L(a,R) \geq S_L(b,R) \tag{8}$$

EXAMPLE 1: Wine experts give their preferences on five Médoc wines a, b, c, d, and e using the following matrix (Fodor and Roubens, 1994, 150):

$$R = \begin{bmatrix} 0.50 & 0.57 & 0.57 & 0.29 & 0.67 \\ 0.43 & 0.50 & 0.70 & 0.52 & 0.28 \\ 0.43 & 0.30 & 0.50 & 0.72 & 0.48 \\ 0.71 & 0.48 & 0.28 & 0.50 & 0.48 \\ 0.33 & 0.72 & 0.52 & 0.52 & 0.50 \end{bmatrix}$$

We have the following leaving scores: $S_L(a) = 2.10$, $S_L(b) = 1.93$, $S_L(c) = 1.93$, $S_L(d) = 1.95$, $S_L(e) = 2.09$, giving the order

$$a \geq e \geq d \geq b \approx c$$

3.4.2 The Proposed Method

The proposed method is, in a way, similar to linguistic ranking methods (Chen et al., 1992, 265). For every two objectives, the designer is asked to specify one of the following characterizations:

- Less important ($\prec$)
- Much less important ($\ll$)
- Equally important ($\approx$)
- More important ($\succ$)
- Much more important ($\gg$)
- Don't care (#)

Don't care (#) is treated exactly as *equally important* ($\approx$). There are some psychological explanations for this, namely that if we don't care with respect to two objectives, then we also don't care which one provides better results. However, in future research it is intended to analyze this relationship more closely. Also, the number of degrees of importance (such as slightly more important, vastly more important, etc.) can be easily extended.

Since k objectives require, in the worst case, $k(k \Leftrightarrow 1)/2$ questions, the designer is first requested to identify the objectives of interest at this stage of the optimization process and to specify the relative importance of these only. However, for clarity and brevity, this step is omitted here.

3.5 Properties of Our Preference Relations

We define the following relations (Cvetković and Parmee, 1999):

relation	intended meaning
$\approx$	is equally important
$\prec$	is less important
$\ll$	is much less important
$\neg$	is not important
!	is important

The properties that we require are:

- Relation $\approx$ is an *equivalence relation* (reflexive, symmetric, and transitive).
- Relations $\prec$ and $\ll$ are *strict orders* (irreflexive and transitive).
- Relation $\approx$ is *congruent* with $\ll$ and $\prec$.
- Relation $\ll$ is a subrelation of $\prec$.
- Miscellaneous properties:

$$
\begin{array}{ll}
!x \vee \neg x & !y \wedge \neg x \Rightarrow x \ll y \\
\neg x \wedge \neg y \Rightarrow x \approx y & x \prec y \wedge y \ll z \Rightarrow x \ll z
\end{array}
$$

Predicates $\succ$ (is more important) and $\gg$ (is much more important) are defined in the following way:

$$x \succ y \overset{\text{def}}{\Leftrightarrow} y \prec x \qquad x \gg y \overset{\text{def}}{\Leftrightarrow} y \ll x \tag{9}$$

3.5.1 Description of the Algorithm

- Let the set of objectives be $O = \{o_1, \ldots, o_k\}$. Construct the equivalence classes $\{C_i \mid 1 \leq i \leq m\}$ of the equivalence relation $\approx$ so that $\cup_{i=1}^{m} C_i = O$, and $C_i \cap C_j = \emptyset$ for $i \neq j$. Choose one element x_i from each class C_i giving set $X = \{x_1, \ldots, x_m\}$, where $m \leq k$. In the sequel, we are going to work on set X.

- Use the following valuation v:

  - If $a \ll b$ then $v(a) = \alpha$ and $v(b) = \beta$
  - If $a \prec b$ then $v(a) = \gamma$ and $v(b) = \delta$
  - If $a \approx b$ then $v(a) = v(b) = \varepsilon$[1]

  Note: Taking into account the intended meaning of the relations, it can further be assumed that $\alpha < \gamma < \varepsilon = 1/2 < \delta < \beta$. It is also assumed that $\alpha + \beta = \gamma + \delta = 1$. Also note that we are not limited to constants α, β, γ, δ, and ε. Their order and property $\alpha + \beta = \gamma + \delta = 1$ is what matters.

[1] Since we work with equivalence classes, this is only possible if $a = b$.

- Initialize two matrices R and R_a of size $m \times m$ to the identity matrix $\mathbf{E}_m$. They will be used in the following way:

$$\begin{aligned} x_i \ll x_j \Leftrightarrow R(i,j) = \alpha, R(j,i) = \beta \Leftrightarrow R_a(i,j) = 0, R_a(j,i) = 2 \\ x_i \prec x_j \Leftrightarrow R(i,j) = \gamma, R(j,i) = \delta \Leftrightarrow R_a(i,j) = 0, R_a(j,i) = 1 \\ x_i \approx x_j \Leftrightarrow R(i,j) = \varepsilon, R(j,i) = \varepsilon \Leftrightarrow R_a(i,j) = 1, R_a(j,i) = 1 \end{aligned} \tag{10}$$

  Note: This valuation provides an idea how of to generalize preferences having s stages rather than only 5 (from "much less important" to "much more important"): if x_i is (say) s' times more important the x_j, simply assign $R_a(i,j) = s'$ and $R_a(j,i) = 0$, etc.

- Perform the following procedure:
  1. For all $i \leq m$ and for all $j \leq m$ such that $j \neq i$ do
     - If $R_a(i,j) + R_a(j,i) = 0$ then
       * Ask whether $x_i \ll x_j$, $x_i \prec x_j$, $x_j \ll x_i$ or $x_j \prec x_i$
       * Using Equations (10), set $R_a(i,j)$ and $R_a(j,i)$ accordingly;
     - Using Warshall's algorithm (Warshall, 1962), compute transitive closure of R_a (some straightforward modifications are necessary):
  2. Using (10), calculate matrix R from R_a:
  3. For each $x_i \in X$ compute weight as a normalized leaving score:

$$w(x_i) = \frac{S_L(x_i, R)}{\sum_{x_j \in X} S_L(x_j, R)}$$

     and for each $y \in C_i$, set $w(y) = w(x_i)$

EXAMPLE 2: Let $O = \{o_1, \ldots, o_6\}$, and $o_1 \approx o_2$ and $o_3 \approx o_4$. We have

$$C_1 = \{o_1, o_2\},\ C_2 = \{o_3, o_4\},\ C_3 = \{o_5\},\ C_4 = \{o_6\}$$

and $X = \{x_1, x_2, x_3, x_4\}$, where $x_i \in C_i$ for $1 \leq i \leq 4$. Let $R = R_a = \mathbf{E}_4$ — identity 4×4 matrix.

Suppose that the first question gives the answer $x_2 \ll x_1$, the second question gives the answer $x_3 \prec x_1$, and the third $x_1 \prec x_4$. The fourth question gives the answer $x_2 \ll x_3$. Since for each pair (i,j) we have $R_a(i,j) + R_a(j,i) \neq 0$, there is sufficient information (without computing transitive closure we would have to ask 6 questions and additionally handle nonconsistent answers), and the matrix R can be constructed. Suppose that $\alpha = 0.05$, $\beta = 0.95$, $\gamma = 0.35$, $\delta = 0.65$, and $\varepsilon = 0.5$. Then,

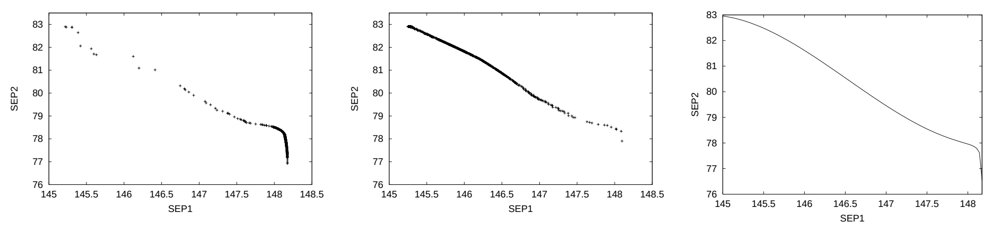
$$R = \begin{bmatrix} \varepsilon & \beta & \delta & \gamma \\ \alpha & \varepsilon & \alpha & \alpha \\ \gamma & \beta & \varepsilon & \gamma \\ \delta & \beta & \delta & \varepsilon \end{bmatrix} = \begin{bmatrix} 0.50 & 0.95 & 0.65 & 0.35 \\ 0.05 & 0.50 & 0.05 & 0.05 \\ 0.35 & 0.95 & 0.50 & 0.35 \\ 0.65 & 0.95 & 0.65 & 0.50 \end{bmatrix}$$

Further, $S_L(x_1, R) = 1.95$, $S_L(x_2, R) = 0.15$, $S_L(x_3, R) = 1.65$, and $S_L(x_4, R) = 2.25$, and the order of importance[2] is $x_2 \ll x_3 \prec x_1 \prec x_4$. Weights are further calculated and normalized so that $\sum_{i=1}^{6} w_i = 1$:

$$w_1 = w_2 = 0.2407, w_3 = w_4 = 0.0185, w_5 = 0.2037, w_6 = 0.2778$$

[2]Note that the order between objectives does not depend on the actual values of α, β, γ, δ, and ε, only their order matters.

This method integrated with weighted sums and the weights-based Pareto method has a significant advantage over the traditional optimization methods, since the user doesn't have to express the weights quantitatively but qualitatively (within a few categories). Figures 2(a) and (b) show two $\boldsymbol{w}$-Pareto fronts of subsonic Specific Excess Power (SEP1) versus supersonic Specific Excess Power (SEP2) of the BAe function, for different preferences, i.e., for different weights, whereas Figure 2(c) shows the shape of the complete Pareto front.



(a) (b) (c)

Figure 2: $\boldsymbol{w}$-Pareto front of SEP1 versus SEP2 of the BAe function for (a) SEP1 $\gg$ SEP2 and (b) SEP1 $\ll$ SEP2. (c) Shape of a complete Pareto front.

3.6 Example with Eight Objectives

Let us consider the case where we are interested in results for more than two objectives. Suppose that we are optimizing the following eight objectives of the BAe function (see Table 1 for the objective meaning)

$$\{y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{13}\}$$

and that we specify the following answers concerning preferences (note that there are only six questions for five distinctive objectives):

$$y_3 \approx y_4,\ y_5 \approx y_6,\ y_7 \approx y_8,$$
$$y_3 \succ y_5,\ y_3 \gg y_7,\ y_3 \prec y_9,\ y_3 \succ y_{13},\ y_5 \succ y_7,\ y_5 \prec y_{13}$$

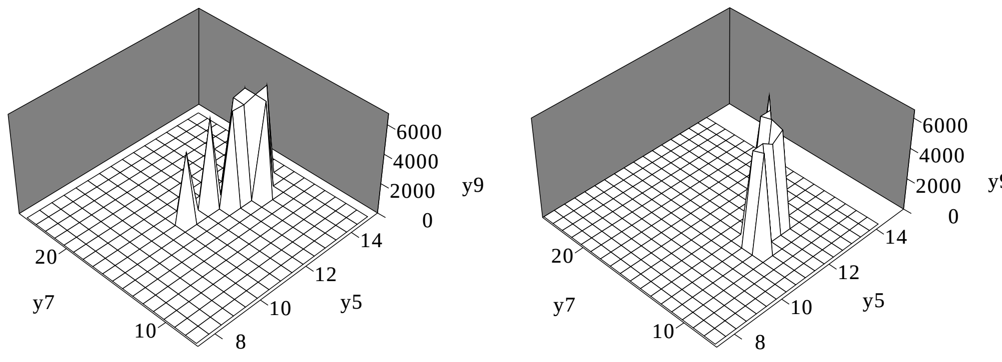
This gives us the following preference order:

$$y_9 \succ y_3 \approx y_4 \succ y_{13} \succ y_5 \approx y_6 \gg y_7 \approx y_8$$

and using the same valuation as before, we end up with the following weightings:

$$w_3 = w_4 = 0.1722,\ w_5 = w_6 = 0.1126,\ w_7 = w_8 = 0.053,\ w_9 = 0.1921,\ w_{13} = 0.1325$$

Performing a weighted Pareto GA optimization (with Pareto set size limited to 400), we end up with an 8-dimensional surface with some 3D slices presented in Figure 3(b). For the comparison, Figure 3(a) contains the same 3D slice without preferences (i.e., using the standard Pareto method). Please note that these are just 3-dimensional slices of an 8-dimensional surface. We are still facing the problem of n-dimensional surface presentation for $n > 4$. Although we are actually looking at the very small part of the 8D Pareto front, we can still notice the shift towards the smaller values of y_7 in Figure 3(b) as the objective y_7 is the least important of all. Similarly, the values of y_5 are less in Figure 3(b) than in Figure 3(a). This could be easily explained by noticing that the weight factor for y_9 (0.1921) is almost twice the weight of y_5 (0.1126) and almost four times the weight of y_7 (0.053).



(a) no preferences (b) $y_9 \succ y_5 \gg y_7$

Figure 3: 3D slices of 8D Pareto front: (a) (y_5, y_7, y_9) without preferences, (b) (y_5, y_7, y_9) with preferences.

There are several aspects of the preferences that are not described here such as the complexity of preference procedure (regarding number of questions needed to compute the complete preference matrix), sensitivity to the valuation and to initial values, questions regarding group preferences, etc. Some of them are discussed in Cvetković and Parmee (2000), and some are being discussed in forthcoming papers.

Recent work has successfully integrated the preference module with the coevolutionary MOGA approach of the following section.

4 Coevolutionary Multiobjective Genetic Algorithm

This represents the search and exploration core of the IEDS. A distributed coevolutionary method (Parmee and Watson, 1999) utilizes individual GAs for the optimization of each objective. Parallel virtual machine (PVM) software (Geist et al., 1994) controls the distributed architecture ensuring minimal clock time.

4.1 Fitness Calculation

The fitness for each objective is normalized relative to the maximum and minimum values found during each GA run with constant adjustment as new upper and lower limits are identified. For each generation, solutions relating to each objective are compared with the best individual from the other GA populations. If a variable is outside a range defined by a range constraint map, it is adjusted by a penalty function. For instance, suppose we are optimizing two objectives: subsonic specific excess power (SEP1) and ferry range (FR). Two GAs (S_0 and S_1) are initialized, S_0 optimizing SEP1 and S_1 optimizing the ferry range. Figure 4 shows the steps required to calculate the fitness of population S_0.

Note that the process is repeated for all individuals in population S_1, which are compared with the best individual in S_0.

1. Rank the fitness of population S_0 using SEP1.
2. Rank the fitness of population S_1 using the ferry range.
3. Starting with individual 1 (the fittest), variable 1, compare the value with the equivalent variable of the best individual in S_1. Return the difference between the two values divided by the total range defined for the variable being examined.
4. Compare the returned value against the value given by the range constraint map for the generation number.
5. If the returned value is greater than the constraint map value, apply a fitness penalty to individual 1.
6. Repeat steps 3–5 for all variables in individual 1.
7. Repeat steps 3–6 for all individuals in S_0.

Figure 4: Steps required for the fitness calculation.

4.2 The Range Constraint Map

The range constraint map must allow each GA to produce an optimal solution based on its own specified objective. This is achieved by setting the value of the map to 1.0, allowing each GA to use the whole range for each variable. As the run progresses, the map, through inflicted penalties, increasingly reduces variable diversity to draw all concurrent GA searches from their separate objectives towards a single optimal design region where all objectives are best satisfied. The constraint maps include a linear decrease in range constraint and a range constraint reduction based on a sine curve. The map must also allow some difference in variable values for each GA towards the end of a run to provide space within which the method can search for an overall optimal solution. This is achieved by setting a minimum value for the range constraint. The number of generations allocated to this final phase of exploration is investigated using 2 values (10% and 50%) of the maximum generations. This produces the maps presented in Figure 5. Note that the minimum value for the maps is set to 0.1 (10% of the variable range).

4.3 Sensitivity Analysis

All variable parameters are assigned equal importance when assessing constraint map penalties. In most situations, however, variables will have differing degrees of influence upon any given objective. It is necessary, therefore, to determine which variables have the greatest bearing on each objective. An on-line sensitivity analysis, which ranks the variables according to their influence upon each objective, is introduced. This design sensitivity ranking is then used to adjust the fitness of each solution to ensure that the values of the most influential variables are within the range defined by the constraint map. Solutions incur the highest penalty where their most influential variables lie outside the current constraint map range. This ensures that subsequent populations contain high levels of feasible solutions in terms of the most influential variables and relatively redundant variables have little or no effect on overall solution fitness.

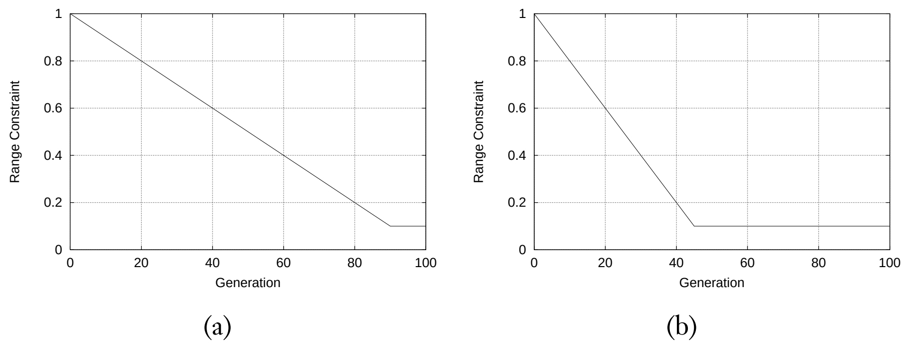


Figure 5: Various constraint maps used for the initial two-objective experiments, (a) linear ramp map, and (b) half linear ramp map.

The Taguchi method (Peace, 1993) has been selected from a number of available techniques to determine the sensitivity of each variable. Taguchi methods use a family of orthogonal fractional-factorial experiments (FFE) matrices (orthogonal arrays (OAs)) incorporating a process for generating data that utilizes a mathematically derived matrix to methodically gather and evaluate the effect of numerous parameters on a response variable. Once the number of design parameters and the number of settings per design parameter are determined, an already-constructed orthogonal array can be selected. As the model has nine variables, three levels are chosen for each input so the L_{27} OA is used for the sensitivity experiment. This OA requires 27 evaluations for each experiment. The fitness penalty is scaled between 0.5 for the most sensitive variable and 0.0 for the least sensitive.

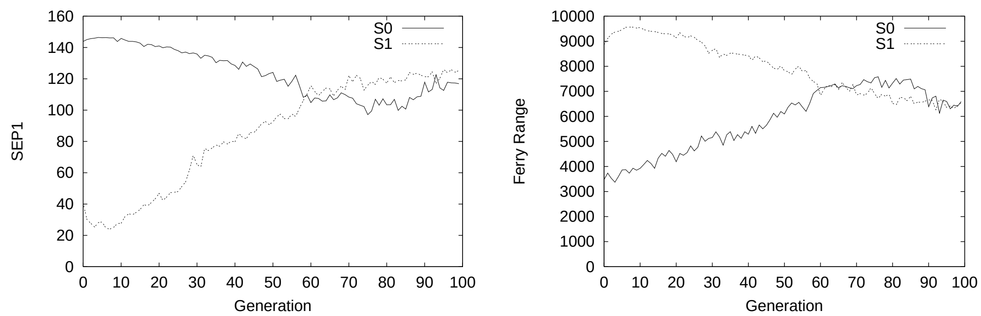
4.4 Results

In order to test the method, two objectives are initially chosen which are known to be highly conflicting: SEP1 and FR. Each GA process (labeled S_0 and S_1) initially has a population size of 100 using a 16-bit binary encoding for each variable. Crossover rate is 0.6, mutation rate is 0.01, and roulette wheel selection with one elite individual is utilized. A total of 100 generations is processed with the fitness penalty set to 0.5. Taguchi sensitivity analysis is not included in this experiment. Figures 6(a) and (b) show the average results obtained over 25 runs.

Figure 6(a) shows that S_0 (the GA optimizing SEP1) produces near optimal solutions at the start of the run, but as the run progresses, this decreases while the SEP1 value of S_1 (optimizing FR) increases. This effect is also shown in Figure 6(b) which shows the ferry range reducing for S_1 and increasing for S_0 to a common design region. This illustrates how both S_0 and S_1 converge upon a region of the design space where high-performance, best-compromise solutions are prevalent.

The best individuals from each generation are saved and the averaged results with the known Pareto front are shown in Figure 7.

The initial generations seek optimal solutions for their particular objective, and as the variables of each GA are restricted by the constraint map, the populations attempt to traverse the Pareto front before converging upon a common region. Further testing for other constraint maps, various population sizes, and inclusion of the Taguchi analysis



(a) (b)

Figure 6: (a) SEP1 vs. generation (b) ferry range vs. generation.

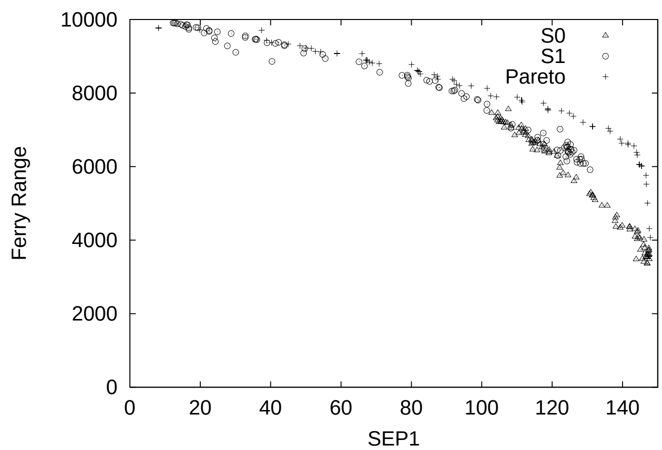


Figure 7: The Pareto front and GA results.

produced the 'end of run' results presented in Table 2. Results from initial runs are inconclusive as to the best type of constraint map to use.

The algorithm is to be used for conceptual design where identification of a high-performance design region is the primary objective. Analysis of the paths described by each map show that the linear map produced results closest to the known Pareto front. The results do show that, although the method is effective with smaller population sizes, the design paths tend to move further away from the Pareto front. The Taguchi analysis was tested with the linear constraint map, but for two objectives, the results show no significant advantage in using the additional fitness ranking based on sensitivity of the objectives to individual variables. However, the provision of information relative to the degree of influence of each variable upon each objective is available to the engineer through the Taguchi results.

Table 2: Two-objective results using various run parameters (at 10%).

Parameters			S_0 (optimizing SEP1)				S_1 (optimizing FR)			
Constr. Map	Tag-uchi	Pop size	SEP1	SD SEP1	FR	SD FR	SEP1	SD SEP1	FR	SD FR
Linear	no	100	116.8	21.9	6587.9	1453.8	125.2	26.0	6542.1	1079.0
Linear	yes	100	120.2	20.0	6344.6	1359.9	134.3	15.5	5647.2	926.7
Linear	no	50	108.4	22.5	6875.1	1307.4	119.4	23.0	6414.4	1191.6
Linear	yes	50	113.2	30.5	6210.7	1971.9	114.3	30.3	6649.5	1420.6
Linear	no	25	113.8	26.8	6442.1	1265.2	98.3	42.8	7065.1	1574.1
Linear	yes	25	104.8	23.6	6833.1	1153.8	110.9	27.0	6573.7	1442.5
Linear	no	10	106.1	27.1	6432.0	1583.9	91.4	27.0	6993.9	1379.2
Linear	yes	10	113.0	29.8	5890.4	1550.4	99.5	32.5	6609.4	1305.0
Lin. half	no	100	131.3	11.6	5946.2	1294.5	131.1	33.1	6062.3	1272.7
Lin. half	no	50	122.5	18.6	6272.4	1300.5	126.5	28.5	6155.7	1288.5
Lin. half	no	25	102.0	30.4	6202.5	2006.8	19.6	18.6	9489.6	389.0
Lin. half	no	10	107.6	23.7	6240.5	1412.0	98.6	28.6	6969.4	1091.3
Sine	no	100	117.8	22.2	6409.6	1496.9	128.9	24.9	6083.3	1242.9
Sine	no	50	109.9	29.2	6655.7	1633.3	117.7	35.4	6551.8	1389.7
Sine	no	25	120.7	21.9	6127.1	1373.1	99.5	41.5	6942.4	1481.6
Sine	no	10	93.9	30.7	6392.3	1721.1	25.4	19.9	9394.9	480.5

4.5 Optimizing Three Objectives

The complexity of the design problem is now increased by setting the number of design objectives to three: SEP1, FR, and subsonic Attained Turn Rate 1 (ATR1) with associated GAs S_0, S_1, and S_2. Figures 8(a)–(c) show the average results from 25 runs. GA settings remain the same. The constraint map used is the linear map to 10%, and, initially, the Taguchi analysis is not included. The results are averaged over 25 runs.

Figures 8(a), 8(b), and 8(c) show each GA converging to a single region. The best individual from each population can be plotted to show the evolution of the design from initial single-objective, high-performance regions to a single design region containing best compromise solutions for all objectives. The results for the three-objective problem are plotted in Figure 9.

Figure 9 also shows projected shadows in the three objective planes. Each GA initially optimizes its own objective at the start of a run and then converges to a single design region. The three-objective test is repeated with the Taguchi analysis included, and the results show that the standard deviations, calculated from the 25 runs, are reduced towards the end of the run for all objectives. This indicates that the Taguchi analysis improves the robustness of the method, ensuring that individual design runs produce consistent results within the design search space while also providing important variable/objective interaction information.

Further research has shown that convergence to a best-compromise region is attainable when optimizing all 13 of the miniCAPS objectives. The full implications of including the Taguchi analysis for an increased number of objectives is now under investigation.

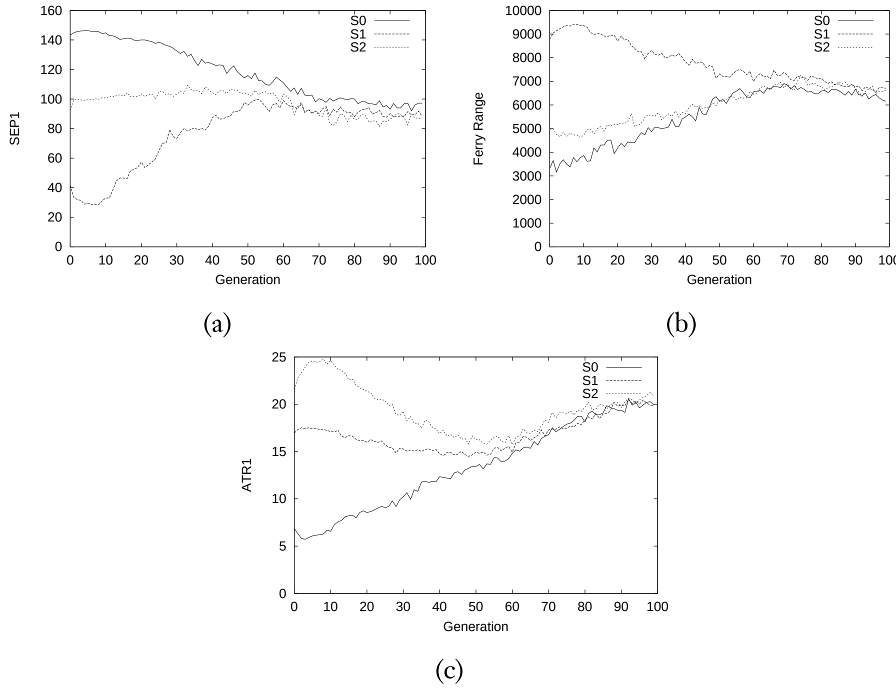


Figure 8: (a) SEP1 vs. generations (b) FR vs. generations (c) ATR1 vs. generations.

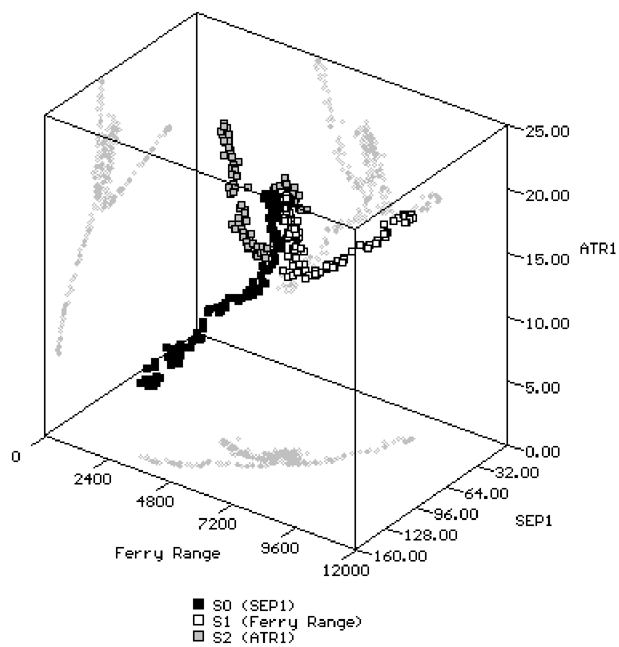


Figure 9: Three-objective problem design paths.

5 Information Gathering and Multiobjective Satisfaction Using Variable Mutation Cluster Oriented Genetic Algorithms (vmCOGAs)

The final individual component of the IEDS relates to information gathering processes. In this respect, the relatively well-established COGA strategies have been introduced.

The initial objective of the COGA evolutionary regional identification techniques has been the efficient decomposition of complex space. However, the longer term aim is their integration with the highly interactive designer/machine environment described in Section 1 to provide information that supports redefinition of the design space. The belief is that this will lead to the discovery of areas of high potential outside of an initial conceptual design brief.

The COGA technique, therefore, represents as an information gathering process that can be utilized in an interactive manner to support the building of design team knowledge relating to the system under design. It is best integrated with conceptual/preliminary design models that provide a degree of definition commensurate with the confidence in available data and the uncertainties relating to overall requirement. Such models should incur little computational expense. COGAs will reside within the information gathering module (IGM) of the IEDS. To extract relevant information, the following objectives must be achieved by the technique:

- good definition of high-potential regions
- good set cover of each identified region in terms of number and diversity of solutions within them
- a minimization of number of calls to the evaluation (fitness) function
- robust performance across differing problem domains
- succinct graphical presentation of extracted relevant, regional design information

Development of regional identification strategies and techniques that satisfy these lower level objectives is described. Those aspects of COGA utilization that directly relate to multiobjective satisfaction are prevalent. Graphical results illustrate the capabilities of the technique in the miniCAPS design domain.

Initial research (Parmee, 1996) concentrated upon the establishment of simple variable mutation regimes (vmCOGAs) that encourage diversity during the early stages of a GA search and promote the formation of clusters of high-performance solutions. A high mutation probability introduced at generation one is subsequently reduced at later selected generations (filtering stages, st_n). The populations from these selected generations are extracted and passed through an adaptive filter. Those solutions that survive the filtering process are stored in a final clustering set.

The fitness distribution of the population is normalized in terms of the number of standard deviations from the mean. A threshold R_f is then introduced, and solutions falling below that threshold are not allowed to enter the final clustering set. They do, however, remain within the GA population and contribute to the continuing evolutionary process. Solutions that exceed the threshold value of the previous stage in intermediate generations of the underlying GA are also passed into the final clustering set (if not already present).

This prevents loss of high-value information and improves set cover in the final clusters. The adaptive filtering avoids the requirement for a priori knowledge that is so often required for evolutionary multimodal optimizers. The value of the R_f threshold of the adaptive filter is relative to known solutions describing the surface topography at that generation from which the population is extracted. The filter system, therefore, adapts to the information available thereby eliminating a need for a priori knowledge relating to the design space. The variation of the R_f values in consecutive runs can be utilized in an investigative manner to assess the relative nature of differing regions of the design space. Relatively low R_f settings indicate general areas of the space containing solutions of above average fitness. Tightening of the filter threshold results in the decomposition of this general area into succinct regions of high performance. The designer, therefore, accumulates knowledge not only concerning individual solutions or particularly high-performance regions but also of the relative nature of the surrounding space. This investigative process based upon R_f variation is illustrated in following sections.

A more detailed description of COGA development is available within the referenced papers.

5.1 Exploration via R_f Filter Variation

VmCOGA has been applied to the BAe miniCAPS airframe problem in an investigative manner in order to build knowledge relating to the general nature of the design space before defining succinct high-performance regions. Initially, a low R_f filter setting is introduced that becomes more discriminatory in subsequent runs. The run times for the conceptual design models are small, allowing 250 generations of vmCOGA in approximately three minutes using a SUN Enterprise 4000 server with 167 MHz UltraSparc processors. This supports rapid investigation and designer interaction in terms of varying filter settings.

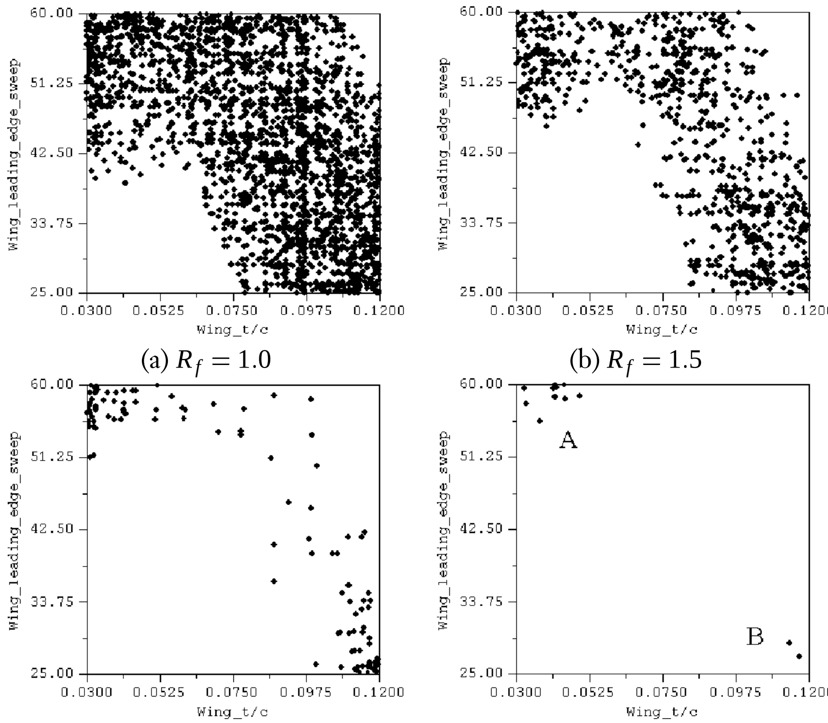
An experimental graphical user interface allows the selection of combinations of any two variables. This allows the vmCOGA generated results to be viewed across a range of two-dimensional hyperplanes of the overall design space. Each point in the figures indicates a design solution present in the final clustering set. Color coding normally indicates relative solution fitness. The results of each independent vmCOGA run are shown in two-dimensional hyperplanes of wing thickness-to-chord ratio against wing leading edge sweep angle. When low filtering thresholds are used, high levels of low fitness solutions pass into the final clustering set. This is illustrated by the ridge of solutions arcing from the top left to the bottom right of the two-dimensional hyperplane (see Figure 10).

As filtering is further increased, this region decomposes into a narrow ridge of solutions. At higher filtering thresholds, the ridge further decomposes into two regions of high performance, A and B (see Figure 10(d)).

This example clearly illustrates the effect of varying the filtering threshold. Low filtering provides the designer with maximum information relating to the general nature of the search space. Conversely, high filtering greatly reduces set cover and produces a limited number of near optimal solutions.

5.2 Multiobjective Continuous Search Space Exploration

The regional identification approach can be extended to further support multicriterion processing. A complex design space can be rapidly decomposed in terms of high-performance



(a) $R_f = 1.0$ (b) $R_f = 1.5$

(c) $R_f = 2.0$ (d) $R_f = 2.5$

Figure 10: Application of vmCOGA to preliminary airframe design.

solutions that satisfy individual objectives. Figures 11(a), (b), and (c) show the results of three independent vmCOGA runs using the miniCAPS model. In each run, a high-performance region is identified for an individual objective, i.e., FR, SEP1, and ATR1.

The objectives are, obviously, in conflict as the high-fitness regions for each objective are in differing regions of the wing aspect ratio/gross wing plan area hyperplane. To identify the region where high-performance solutions that best satisfy all objectives can be found, the mutually inclusive region or regions can be defined from the data of the three runs. Any solutions contained within these regions will be similarly high-fitness individuals in all objectives if all objectives are equally weighted.

The utilization of this multicriterion COGA strategy may identify a lack of common ground with respect to all objectives. Designer/machine interaction may then provide problem insight, which either leads to the identification of a mutually inclusive region or the development and implementation of a revised design. The identification of a common region is simply illustrated using the BAe model in Figures 12(a)–(c). With filter threshold settings of 1.0 for all objectives, a common region of high-performance solutions that satisfy FR and ATR1 can be identified, but solutions relating to SEP1 are not part of that region. However, relaxing the filter threshold relating to SEP1 allows lower performance solutions through to the final clustering set. The SEP1 region gradually expands until a common region for all three objectives is identified.

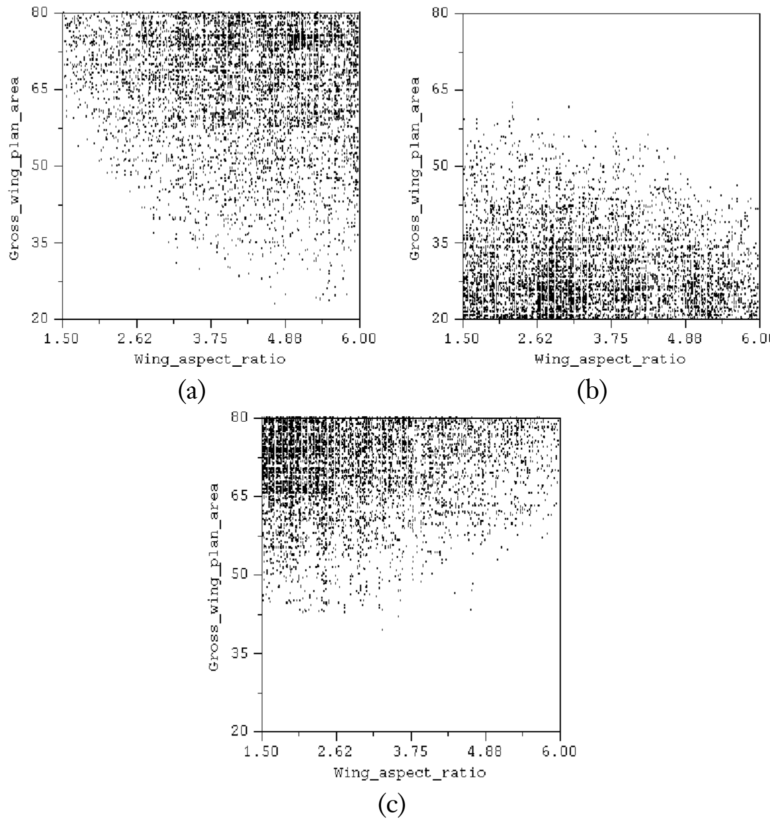


Figure 11: vmCOGA applied to individual objectives using miniCAPS (a) maximizing FR; (b) maximizing SEP1; (c) maximizing ATR1.

Relaxing the filter is equivalent to reducing the importance of the SEP1 objective through an acceptance of lower fitness solutions while maintaining the higher relative fitness of FR and ATR1. The size and shape of the regions can be tailored through differing variation of the filter threshold of each of the objectives. By varying filter settings for each objective, the engineer can explore the objective space relative to the variable space in terms of the two-dimensional hyperplanes. The flexibility of the graphical user interface allows objectives to be included or disregarded while also allowing variable ranges to be altered in order to support investigation of specific regions and objective/variable interaction. A facility, therefore, exists for concurrent search of both variable and objective space.

One aspect of variable/objective interaction is shown in Figure 13. Figure 13(a) shows a high-performance region relating to ATR1 plotted in the gross wing plan area/wing aspect ratio hyperplane and plainly indicates the settings for upper and lower bounds of the two variables for further search effort. Figure 13(b), however, shows the corresponding distribution of high-performance solutions in the climb Mach number/cruise height hyperplane. A uniform distribution of such solutions across this hyperplane is evident for the ATR1 objective. This is also the case for SEP1 and FR objectives. This immediately provides an indication of the sensitivity of the objective functions to variation of the parameter values in each hyperplane. Although high-performance regions of gross wing plan area and wing

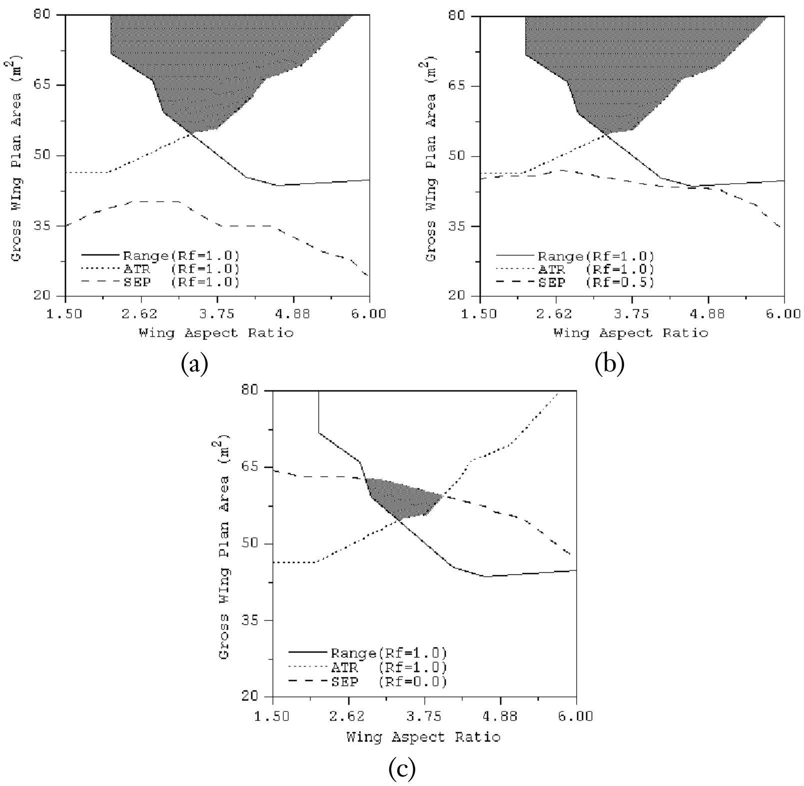


Figure 12: Identification of feasible high-performance regions relating to various objective combinations through filter threshold relaxation. (a) A common region containing high-performance solutions for FR and ATR1 has been identified, but SEP1 objectives cannot be satisfied. (b) Relaxing the filter threshold for SEP1 allows lower fitness SEP1 solutions through and boundary moves towards a compromise region. (c) Further relaxation results in the identification of a compromise region for all objectives.

aspect ratio values are well constrained, both climb Mach number and cruise height can take any value within existing upper and lower bounds. The engineer can thus choose values that are of seemingly best benefit before removing these variables from future search effort. This visual measure can rapidly provide an overall indication of the degree of interaction between those variables considered most significant to the engineer. Variables and objectives can be made temporarily redundant while others can be introduced during a largely explorative process.

6 Discussion and Conclusions

Each IEDS component can be utilized in a stand-alone manner and can produce meaningful results relating to multiobjective decision making. However, the research objective is to integrate these processes within the IEDS framework and to ascertain in what manner and to what extent the engineer may interact with them. A major research aspect relates to the development of agent structures to provide machine-based communication and control.

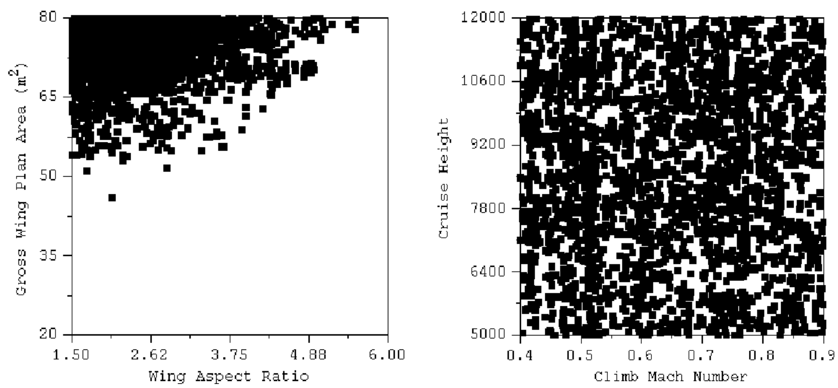


Figure 13: Objective sensitivity with reference to selected variable parameters.

Work in this area has recently commenced but is not sufficiently advanced to report upon at this time.

The research of Section 3, relating to interactive preference determination, has resulted in a method for easily and rapidly transforming qualitative characterization of objective/scenario relative importance into quantitative representation. The resulting algorithm implements such a transformation. On-line preference change, based upon increasing problem knowledge, will support extensive exploration of objective space through integration with the coevolutionary methodology of Section 4.

Results from the coevolutionary system of Section 4 can provide single solutions that satisfy a number of objectives while also defining pathways across the Pareto surface that directly relate to each objective. When integrated with the design preference component and the information gathering component, such capabilities will prove highly beneficial as iterative designer/evolutionary search processes become established. The main advantages of the method are that it produces within a single run of the algorithm: good solutions relating to local objectives within the first few generations; design paths that trace the trade-off surface to some extent; a compromise design region for the problem; and information about the relative importance of the variable parameters. Results approximately traverse the Pareto front and results using three objectives show the ability of the method to converge on an optimal solution by tracking across the Pareto surface from three different starting points. The identification of primary variables aids the search process by ensuring that the most important variables have the greatest influence upon search direction. The Taguchi analysis has little effect with two objectives but does improve results from the three-objective experiments. This is supported by recent work where all miniCAPS objectives have been included. Again, satisfactory convergence upon a common region has been achieved for all objectives.

The technique addresses convergence upon 'best compromise' regions through the monitoring of Euclidean distance in variable space between solutions relating to each objective and a subsequent variation of search pressure to draw independent evolutionary processes towards a common region. The objectives are to accumulate information relating to individual objective performance, objective sensitivities, and relevant areas of the Pareto surface in addition to the identification of an optimal multiobjective solution. It has not been considered appropriate to compare the results from this approach to other established

MOGA techniques as relative performance must be measured in terms of all of the above objectives.

The vmCOGA examples illustrate how adaptive filter variation can generate information and the manner in which the COGA concept can further support design exploration through investigative variation of the filter with respect to individual objectives. Other information gathering capabilities relate to identification of robust design regions, degree of constraint satisfaction, and the extraction of information relating to local fitness landscape topology. All of these aspects have been addressed in the referenced literature.

Appropriate use of the graphical user interface in both variable and objective space can provide the designer with much relevant information, which can be analyzed and discussed both subjectively and objectively by the design team. It should be possible through such discussion to capture and incorporate intuitive aspects and problem insight based upon previous experience during problem reformulation. It is, therefore, possible to argue that a degree of knowledge capture has taken place in that the application of engineering knowledge and intuition in relation to solution and information generated by the evolutionary system has resulted in a redefinition of the design space. This knowledge and intuition may, therefore, be considered implicit within further search and exploration of the redefined space.

With further developments related to minimization of evaluation calls, distribution of the underlying GA process, and increasing processor capability, it does not seem unreasonable to assume that results will be available in seconds rather than minutes in the relatively short term. The possibility of virtual real-time interaction within complex multiobjective conceptual design environments in the medium term seems likely.

The establishment of the IEDS is an ambitious objective. However, in order to best utilize future computing capabilities as they become available, such research and the associated introduction of experimental prototypes is seen as essential. There is already an under-utilization of current machine capability during conceptual design. It is suggested that this integration of evolutionary/adaptive search, exploration and optimization technologies, and complementary computational intelligence techniques could ensure a much improved take-up of such capability within complex conceptual design environments.

Acknowledgments

This work is partly supported by UK Engineering and Physical Sciences Research Council (EPSRC) and British Aerospace. We thank them for their continuing support.

References

Ben-Tal, A. (1979). Characterisation of Pareto and lexicographic optimal solutions. In Fandel, G. and Gal, T., editors, *Proceedings of the Third Conference on Multiple Criteria Decision Making Theory and Application*, pages 1–11, Springer-Verlag, Berlin, Germany.

Chen, S.-J., Hwang, C.-L. and Hwang, F. P. (1992). *Fuzzy Multiple Attribute Decision Making*. Springer-Verlag, Berlin, Germany.

Cvetković, D. and Parmee, I. C. (1999). Use of preferences for ga–based multi–objective optimisation. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M. and Smith, R. E., editors, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference*, pages 1504–1510, Morgan Kaufmann, San Francisco, California.

Cvetković, D. and Parmee, I. C. (2000). Designer's Preferences and Multi-Objective Preliminary Design Processes. In Parmee, I. C., editor, *Evolutionary Design and Manufacture: Selected Papers from the Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM'2000)*, pages 249–260, Springer-Verlag, London, England.

Deb, K. (1999). Evolutionary algorithms for multi–criterion optimization in engineering design. In Miettinen, K., Mäkelä, M., Neittaanmäki, P. and Périaux, editors, *Proceedings of Evolutionary Algorithms in Engineering and Computing Science (EUROGEN-99)*, pages 135–161, John Wiley and Sons, New York, New York.

Fodor, J. and Roubens, M. (1994). *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Fonseca, C. M. and Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16.

Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V. (1994). *PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing*. MIT Press, Cambridge, Massachusetts.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, Massachusetts.

Hajela, P. and Lin, C. Y. (1992). Genetic search strategies in multicriterion optimal design. *Structural Optimisation*, 4:99–107.

Horn, J. and Nafpliotis, N. (1993). Multiobjective optimization using the niched Pareto genetic algorithm. IlliGAL Technical Report 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Illinois.

Hwang, C.-L. and Masud, A. S. M. (1979). *Multiple Objective Decision Making – Methods and Applications*. Springer Verlag, Berlin, Germany.

Jakob, W., Gorges-Schleuter, M. and Blume, C. (1992). Application of genetic algorithms to task planning and learning. In Männer, R. and Manderick, B., editors, *Proceedings of Parallel Problem Solving from Nature II*, pages 291–300, North–Holland, Amsterdam, The Netherlands.

Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In Schwefel, H.-P. and Männer, R., editors, *Parallel Problem Solving from Nature — Proceedings of the First Workshop*, Volume 496 of *Lecture Notes in Computer Science*, pages 193–197, Springer-Verlag, Berlin, Germany.

Lai, Y.-J. and Hwang, C.-L. (1996). *Fuzzy Multiple Objective Decision Making*. Springer-Verlag, Berlin, Germany.

Lin, J. G. (1976). Maximal vectors and multi–objective optimization. *Journal of Optimization Theory and Application*, 18(1):41–64.

Nisbett, R. E. and Wilson, T. D. (1977). Telling more then we can know: Verbal reports on mental processes. *Psychological Review*, 84(3):231–259.

Osyczka, A. (1984). *Multicriterion Optimization in Engineering with FORTRAN Programs*. Ellis Horwood, Chichester, England.

Parmee, I. C. (1996). The maintenance of search diversity for effective design space decomposition using cluster oriented genetic algorithms (COGAs) and multi–agent strategies (GAANT). In Parmee, I. C., editor, *Proceedings of Adaptive Computing in Engineering Design and Control*, pages 128–138, University of Plymouth, Plymouth, England.

Parmee, I. C. (1999). Exploring the design potential of evolutionary search, exploration and optimisation. In Bentley, P. J., editor, *Evolutionary Design by Computers*. Morgan Kaufmann, San Francisco, California.

Parmee, I. C. and Bonham, C. R. (1998). Supporting innovative and creative design using interactive designer/evolutionary computing strategies. In Gero, J. and Maher, M., editors, *Computation Models of Creative Design Conference*, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, Australia.

Parmee, I. C. and Watson, A. H. (1999). Preliminary airframe design using co–evolutionary multiobjective genetic algorithms. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M. and Smith, R. E., editors, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference*, pages 1657–1665, Morgan Kaufmann, San Francisco, California.

Peace, G. S. (1993). *Taguchi Methods: A Hands–On Approach*. Addison-Wesley, Reading, Massachusetts.

Schaffer, J. D. (1984). *Some Experiments in Machine Learning using Vector Evaluated Genetic Algorithm*. Ph.D. thesis, Department of Electrical Engineering, Vanderbilt University, Nashville, Tennessee. TCGA file No. 00314.

Srinivas, N. and Deb, K. (1995). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248.

Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Ph.D. thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Warshall, S. (1962). A theorem on Boolean matrices. *Journal of the ACM*, 9(1):11–12.