Genetic Algorithm–based Multi–objective Optimisation and Conceptual Engineering Design

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

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

Abstract- In this paper we present a genetic algorithm based system for conceptual engineering design. First, we present a method based on preference relations for transforming non–crisp (qualitative) relationships between objectives in multi–objective optimisation into quantitative attributes (numbers). This is integrated with two multi–objective Genetic Algorithms: weighted sums GA and a method for combining the Pareto method with weights. Examples of preference relations application together with traditional Genetic Algorithms are also presented. A further method for dynamical inclusion and modification of extra constraints (not included in the mathematical model of the system) via scenarios is presented. Its use is discussed and potential applications indicated. Finally, some future work paths are mentioned.

1 Introduction

There are significant differences between so called 'textbook optimisation problems' and 'real world applications'. In both cases we have a function to optimise:

Definition 1 Let $n > 0$, $k > 0$, $\mathcal{D} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n \subseteq \mathbf{R}^n$, and $\mathcal{R} = \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_k \subseteq \mathbf{R}^k$. Let further $f_i : \mathcal{D} \mapsto \mathcal{Y}_i$ for $1 \leq i \leq k$ and $\mathbf{F} : \mathcal{D} \mapsto \mathcal{R}$, so that $\mathbf{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$. □

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

$$\max_{\boldsymbol{x}} \mathbf{F}(\boldsymbol{x}) \tag{1}$$

$$g_1(\boldsymbol{x},\boldsymbol{p}) \leq 0, \ldots, g_l(\boldsymbol{x},\boldsymbol{p}) \leq 0 \tag{2}$$

where $\boldsymbol{p} = (p_1, \ldots, p_u)$ are additional (real–valued) parameters. This problem is well known and a number of non–genetic (Hwang & Masud 1979, Osyczka 1984) and genetic algorithm (Schaffer 1985, Horn & Nafpliotis 1993, Fonseca & Fleming 1995, Veldhuizen & Lamont 1998, Coello 1999) approaches exist. An additional problem is that not all objectives are equally important which necessitates the use of weights or preferences. We have applied Genetic Algorithms adapted to solve multiobjective optimisation problems and described some of the design problems in (Cvetković, Parmee & Webb 1998).

The designer cannot always completely objectively define the preferences regarding the objectives which have to be optimised (cf. Nisbett & Wilson 1977, p. 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 optimisation (Lai & Hwang 1996). In this paper we address this problem in a different manner and integrate our developed methods to different GA–based optimisation techniques. Further details are given in (Cvetković & Parmee 1999).

Also, among the solutions that a given multi–objective system delivers, there are more preferable solutions and less preferable solutions. The designer can help the system by specifying additional constraints and conditionals such as: "I would like to have $y_5 \in [0,4]$ or, if not possible (or feasible), then I would like to have $y_1 + y_3 > 100$ etc." This leads the system to a more concentrated search in a given direction.

Basically, the process of Engineering Design has the components presented in Figure 1.

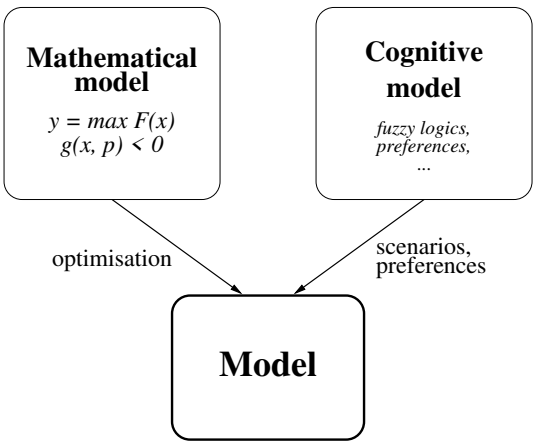


Figure 1: The initial phase of whole system design

As we can see, there is a mathematical model component for which we are using genetic algorithms (based on the Breeder GA (Mühlenbein & Schlierkamp-Voosen 1993*a*) utilising genetic operators suitable for real valued chromosomes)

adapted to use techniques for multi–objective optimisation (Cvetković et al. 1998, Cvetković & Parmee 1998). There is also a component that is more subjective, that cannot always be expressed in strict mathematical model terms and where different designers may have different ideas and opinions.

2 Preferences

The most common situation found in the literature is the following: if we have a finite set X of $m_a \geq 2$ alternatives and a set of $m_e \geq 2$ experts E, experts can represent their preferences in three different ways:

1. Using preference ordering of the alternatives;
2. Using fuzzy preference relations;
3. Using utility functions.

Unfortunately, all this is not directly applicable to multi–objective optimisation problems as we are not choosing *single* alternative from a finite set, but *all* of them — only their order of importance matter. However, fuzzy preferences could support the estimation of relative importance (weights) of objectives in multi–objective optimisation problems.

2.1 Fuzzy preferences and preference order

The *preference* of x over y is usually expressed as $R(x,y) = f(g(x), Ng(y))$, where f is a non–decreasing function of both arguments and N is a strong negation (Fodor & Roubens 1994, p. 177). *Strict preference* relation P and *indifference* relation I are further defined in the following way:

$$P(x,y) = 1 - R(y,x), \quad I(x,y) = \min(R(x,y), R(y,x))$$

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

$$S_L(a,R) \stackrel{\text{def}}{=} \sum_{c \in A \setminus \{a\}} R(a,c) \tag{3}$$

$$a \geq_L b \text{ if and only if } S_L(a,R) \geq S_L(b,R) \tag{4}$$

2.2 The proposed method

Our approach is in a way similar to linguistic ranking methods (Chen, Hwang & Hwang 1992, p. 265). For every two objectives we ask the designer to specify one of the following characterisations:

- Less important
- Much less important
- Equally important
- More important
- Much more important
- Don't care

In the further text, *don't care* is treated exactly as *equally important*. There are some psychological explanations for this, namely that if we do not care in respect of two objectives, then we also do not care which one provides better results. However, in future research it is intended that this relation will be analysed more closely. Also, we can easily and quite straightforwardly extend the number of degrees of importance (such as slightly more important, vastly more important etc.).

Since k objectives require in the worst case $k(k-1)/2$ questions, we first ask the designer to identify the objectives of interest at this stage of the optimisation process and specify the relative importance of these only. However, for clarity and brevity, this step will be skipped in the rest of this paper.

Concerning objective importance escalation (from much less important to much more important), it is worth mentioning the work of Lootsma (1996, 1997).

2.3 Properties of our preference relations

We define the following relations:

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 sub–relation of $\prec$;
- Miscellaneous properties:

$$!x \vee \neg x \tag{5}$$

$$!y \wedge \neg x \Rightarrow x \ll y \tag{6}$$

$$\neg x \wedge \neg y \Rightarrow x \approx y \tag{7}$$

$$x \prec y \wedge y \ll z \Rightarrow x \ll z \tag{8}$$

Further we define $\succ$ (more important) and $\gg$ (much more important) in terms of $\prec$ and $\ll$:

$$x \succ y \stackrel{\text{def}}{\leftrightarrow} y \prec x \tag{9}$$

$$x \gg y \stackrel{\text{def}}{\leftrightarrow} y \ll x \tag{10}$$

2.4 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 $\mathcal{C}_i$ giving set $\mathcal{X} = \{x_1, \ldots, x_m\}$ where $m \leq k$. In the sequel we are going to work on set $\mathcal{X}$.

- 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 $v(a) = v(b) = \varepsilon$

  Note: Taking into account the intended meaning of the relations, we assume that $\alpha < \gamma < \varepsilon = 1/2 < \delta < \beta$, and that $\alpha + \beta = \gamma + \delta = 1$.

- Initialise 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:

$$\begin{aligned} x_i \ll x_j &\leftrightarrow R(i,j) = \alpha \leftrightarrow R_a(i,j) = 0,\ R_a(j,i) = 2 \\ x_i \prec x_j &\leftrightarrow R(i,j) = \gamma \leftrightarrow R_a(i,j) = 0,\ R_a(j,i) = 1 \\ x_i \approx x_j &\leftrightarrow R(i,j) = \varepsilon \leftrightarrow R_a(i,j) = 1,\ R_a(j,i) = 1 \end{aligned} \quad (11)$$

  together with $R(j,i) = 1 - R(i,j)$.

- Perform the following procedure:
  1. For all $i \leq m$ and for all $j \leq m$ such that $j \neq i$ do
     - If $R_a(i,j) = 0$ and $R_a(j,i) = 0$ then
       - Ask whether $x_i \ll x_j$, $x_i \prec x_j$, $x_j \ll x_i$ or $x_j \prec x_i$. Note that since we work with class–represents, we can exclude the case $x_i \approx x_j$.
       - Using (11) set $R_a(i,j)$ and $R_a(j,i)$ accordingly.
     - Using Warshall's algorithm (Warshall 1962) (a standard algorithm for computing transitive closure which for every two points in relation graph check if there is a (direct or indirect) path between them), compute transitive closure of R_a.
  2. Using equations (11) calculate matrix R from R_a.
  3. For each $x_i \in \mathcal{X}$ compute weight as a normalised leaving score:

$$w(x_i) = \frac{S_L(x_i, R)}{\sum_{x_j \in \mathcal{X}} S_L(x_j, R)}. \quad (12)$$

     and for each $y \in \mathcal{C}_i$ set $w(y) = w(x_i)$.

3 Applications of preferences

The concept of relative importance of the objectives can be integrated with Genetic Algorithms in at least two different situations:

- weighted sum based optimisation, and
- the Pareto optimisation.

In both cases the preference method will be used to calculate the weights as required. More details are described in the following sections.

3.1 Weighted sum based optimisation

Weighted sum based optimisation uses a genetic algorithm that instead of the vector function (1) optimises the scalar function

$$F'_{\boldsymbol{w}}(\boldsymbol{x}) = \sum_{i=1}^{k} w_i \cdot f'_i(\boldsymbol{x})$$

where f'_i is objective f_i normalised to $[0,1]$ and $\boldsymbol{w} = (w_i)_{i=1,k}$ is the weights vector computed using the algorithm presented in section 2.4.

Our method integrated with weighted sums has a significant advantage over the traditional weighted sum based optimisation methods since the user doesn't have to express the weights quantitatively but qualitatively (within a few categories) which is much easier. Figure 2 show results using different preferences on weighted sum optimisation of BAe function (explained in section 4.2). Results were obtained optimising y_4 (SEP2) vs. y_9 (FR) only.

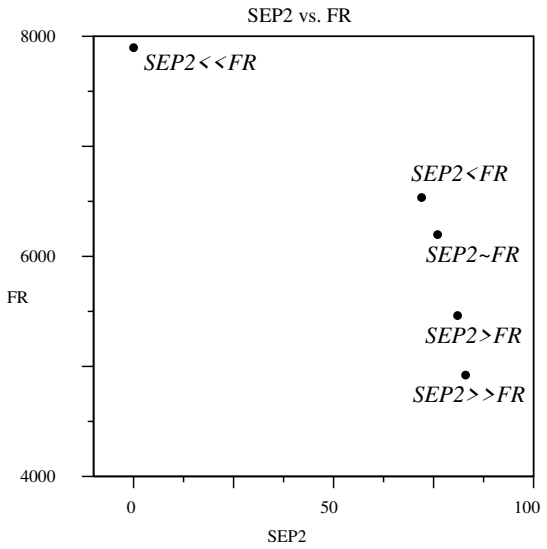


Figure 2: The influence of preference settings on BAe function.

However, weighted sum based GA optimisation methods, although very useful for multi–objective optimisation in general, may not be suitable for optimisation during the conceptual design phase. The main reason is that in the conceptual design phase the likelihood of objective and constraint variation is high. Thus the fitness landscape will change, therefore necessitating the constant re–calculation and normalisation of weights (Cvetković et al. 1998, p. 259).

3.2 Pareto optimisation

Comparing the Pareto principle based multi-objective optimisation with lexicographic order based optimisation (Ben-Tal 1979), we see two extremes concerning the objective importance: in the case of the Pareto optimisation all objectives are considered equally important whereas in the case of lexicographical order the first objective is the most important one and only if we get the same results for the first objective, do we then consider the second objective etc. In this section we try to develop an optimisation method that is based on the Pareto principle but where we can specify the relative importance of objectives.

As in the case of weighted sums based methods, relative importance of objectives in this modified Pareto method could be specified using weights (quantitatively) or they could be combined with the above developed preference method that translates qualitative specification into quantitative. Without this combination, the modified Pareto method would suffer from the same problem as the weighted sum method: how to specify weights when there are many objectives.

Definition 2 We say that the vector $\boldsymbol{x} = (x_1, \ldots, x_k)$ is *non–dominated* by vector $\boldsymbol{y} = (y_1, \ldots, y_k)$, denoted $\boldsymbol{x} \succeq \boldsymbol{y}$, if $x_i \geq y_i$ for all $1 \leq i \leq k$. In other words,

$$\boldsymbol{x} \succeq \boldsymbol{y} \;\leftrightarrow \frac{1}{k}\sum_{i=1}^{k} I_{\geq}(x_i, y_i) \geq 1, \tag{13}$$

where

$$I_{\geq}(x,y) = \begin{cases} 1, & x \geq y \\ 0, & x < y \end{cases}$$

□

We can generalise (13) and say (assuming $\sum_{i=1}^{k} w_i = 1$):

$$\boldsymbol{x} \succeq_w \boldsymbol{y} \;\text{ if and only if }\; \sum_{i=1}^{k} w_i \cdot I_{\geq}(x_i, y_i) \geq 1, \tag{14}$$

or we can even put some threshold $\tau \leq 1$ in (14) and say

$$\boldsymbol{x} \succeq_w^{\tau} \boldsymbol{y} \;\text{ if and only if }\; \sum_{i=1}^{k} w_i \cdot I_{\geq}(x_i, y_i) \geq \tau. \tag{15}$$

Definition 3 We will call relation $\succeq_w$ defined by (14) $\boldsymbol{w}$*–non-dominance* and the relation $\succeq_w^{\tau}$ defined by (15) $(\boldsymbol{w}, \tau)$*–non-dominance*. □

Note: The standard non-dominance relation is just a special case of (15) for $\boldsymbol{w} = (\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k})$ and $\tau = 1$.

Example 1 Let $\mathbf{F}(\boldsymbol{x}) = (x_1^2, x_2^2, x_3^2, x_4^2)$ and let the weights vector $\boldsymbol{w} = (1/2, 1/3, 1/6, 0)$. We have:

$$\mathbf{F}(1,2,3,-4) \succeq \mathbf{F}(1,2,3,1)$$
$$\mathbf{F}(1,2,3,5) \succeq_w \mathbf{F}(1,2,3,7)$$
$$\mathbf{F}(1,2,3,7) \succeq \mathbf{F}(1,2,3,5)$$
$$\mathbf{F}(1,3,7,4) \succeq_w^{0.6} \mathbf{F}(0,4,5,9)$$

If we set all weights to $1/k$ (for k objectives) and threshold $\tau < 1$, we get the case of so called 'weak dominance' where for the dominance we require at least $[k \cdot \tau]$ components to perform better.

Note: The relation $\succeq_w^{\tau}$ is transitive as a product of transitive (component–wise) orders and has all the usual features of an order relation. Also, we assume that the weights do not change during the optimisation process.

Definition 4 The *Pareto front* is defined as a maximal set of non–dominated elements (according to the given order $\succeq$) and this definition is naturally extended to $\boldsymbol{w}$–Pareto front and to $(\boldsymbol{w}, \tau)$–Pareto front for a given vector of weights $\boldsymbol{w}$ and threshold τ according to the order $\succeq_w$ and $\succeq_w^{\tau}$ given by (14) and (15) respectively. □

As mentioned before, vector $\boldsymbol{w}$ could be either specified directly by the designer or can be calculated from his preferences which would help the designer to work in more qualitative terms without determining if the weight should be set of 0.1 or to 0.09 etc.

The Pareto front method combined with genetic algorithms is a very powerful optimisation method since it maintains the diversity of population. However, it can be computationally very expensive.

4 Genetic Algorithms and Pareto Optimisation

This section describes applications of above method. First we give a fairly simple example of a test function, and after that, in section 4.2, we present a real world problem in cooperation with British Aerospace (BAe).

As a simple test function that can immediately show some features of $\boldsymbol{w}$–Pareto front, let us try to maximise the following functions ($n = k = 2$):

$$\begin{aligned} f_1(x_1, x_2) &= \sin(x_1^2 + x_2^2 - 1) \\ f_2(x_1, x_2) &= \sin(x_1^2 + x_2^2 + 1) \end{aligned} \tag{16}$$

for $x_1, x_2 \in [0, 3\pi/4]$. Obviously they don't have a joint maximum since f_1 is at maximum for $x_1^2 + x_2^2 = \pi/2 + 1$ and f_2 is at maximum for $x_1^2 + x_2^2 = \pi/2 - 1$.

Using different preferences, i.e $f_1 \prec f_2$, $f_1 \approx f_2$ and $f_1 \succ f_2$, we obtained three different vectors $\boldsymbol{w}$ and three different Pareto fronts showed in Figure 3(a)–(c). Results were obtained running GA (described in more details in section 4.2) for 15 generations.

4.1 Genetic Algorithm Used

The core of the genetic algorithm we have used is based on the Breeder Genetic Algorithm (Mühlenbein & Schlierkamp-Voosen 1993*a*). It works on real numbers so there is no need for binary encoding/decoding process. It utilises genetic operators suitable for real valued chromosomes (i.e. arithmetic crossover, intermediate crossover, exponential mutation, Gauß mutation etc.), and is adapted for multi–objective optimisation (Cvetković et al. 1998, Cvetković & Parmee 1998). For

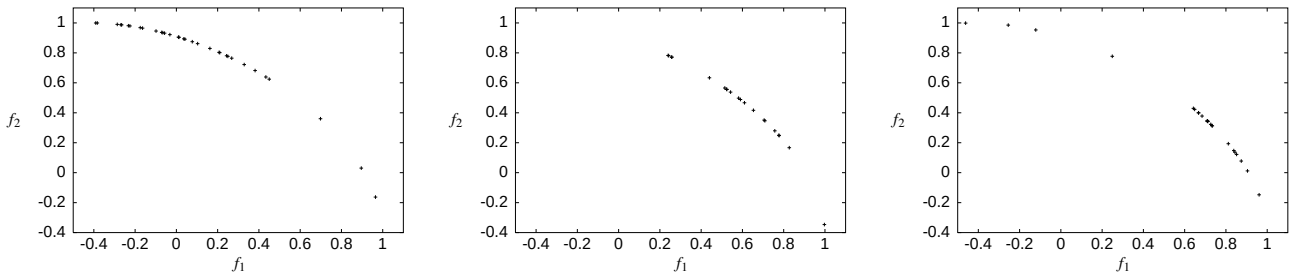


Figure 3: Different parts of Pareto front of function (16) for different preferences: (a) $f_1 \prec f_2$, (b) $f_1 \approx f_2$, (c) $f_1 \succ f_2$.

the selection method, there are standard selection methods such as tournament and 'roulette wheel' (Goldberg & Deb 1990), and truncation (Mühlenbein & Schlierkamp-Voosen 1993*b*), but the methods used also include VEGA–like selection and selection based on Pareto ranking of individuals (Srinivas & Deb 1995, Cvetković et al. 1998) rather then the fitness.

In all subsequent experiments, we have used population size of 50, uniform crossover with the probability 1.0, EXP mutation with the probability $1/k$ (k is the number of objectives) and Pareto–based tournament selection. Every GA was run for 200 generations.

4.2 BAe Function and GA/Pareto Optimisation

The British Aerospace (BAe) design problem is presented in (Cvetković et al. 1998). The project is based on a collaboration with British Aerospace (BAe) and utilises their CAPS system. CAPS (Computer aided project studies) is an integrated computer software suite, developed at BAe for use by engineers and designers during the earliest investigation stages of a new aircraft project. The scope of disciplines covered by CAPS is wide: preliminary geometric definition, aerodynamic analysis, mass estimation, performance analysis, cost estimation etc. In a typical job the user programs CAPS to search for design solutions that meet performance requirements whilst satisfying a number of constraints. BAe has developed a MINICAPS model (Webb 1997) based upon the full CAPS representation. Input (variables) and output (objectives) of the BAe function are presented in Figure 4.

At the moment we have 9 input variables $x_1, \ldots, x_9$ and 13 output objectives $y_1, \ldots, y_{13}$ to optimise simultaneously. The constraints are mainly range–based. The interaction between y_4 (specific excess power) and y_9 (ferry range) is specially interesting as they strongly conflict. All results we discuss in this section will be based on simultaneous optimisation of y_4 and y_9 objectives. All other objectives are being ignored.

Figure 5(a)&(b) show two $\boldsymbol{w}$–Pareto fronts of y_4 versus y_9 for different preferences i.e. different weights. For comparison, Figure 5(c) shows the shape of the complete Pareto front obtained using classical methods and then fitted using

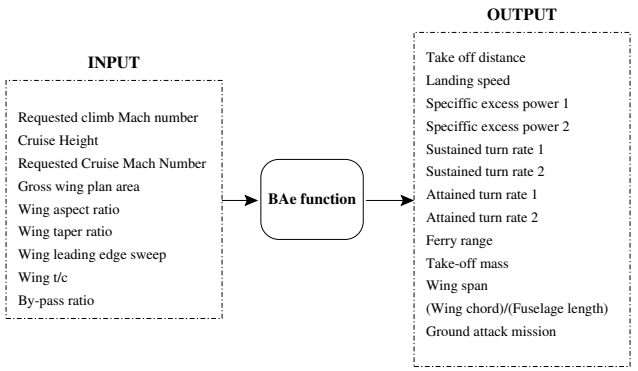


Figure 4: BAe function

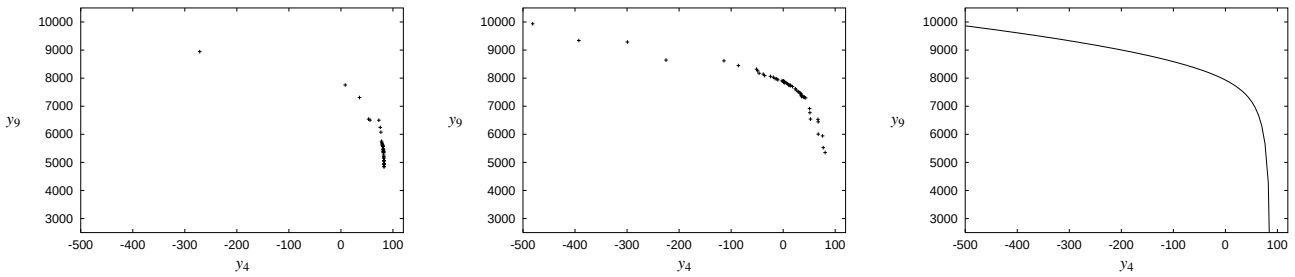
function $f(x) = 7216 - 28925/(83 - 0.9x) + 117\sqrt{83 - 0.9x}$. Knowing the behaviour of the Pareto front with respect to $\boldsymbol{w}$ and τ would enable us to vary the parameters during the genetic algorithm run to identify those parts of the Pareto front that are of special interest (considering the density of points in given regions etc.). Combination with preference methods give us a very powerful tool for region exploration.

5 Scenarios

Another tool that we want to integrate into our genetic algorithm for multi–objective optimisation is scenario handling.

In a typical design situation, the designer has one or several preferential situation (scenarios) that he adds to the fitness function such as: "I would like to have $y_5 \in [0,4]$ or, if not possible (or feasible), then I would like to have $y_1 + y_3 > 100$ etc." So, together with our model with its set of objectives and constraints, we have a dynamical, interactive set of additional constraints or goals. The difference to the model–embedded constraints is that those scenarios are interpreted, so there is no need for program re–compilation.

Definition 5 *Scenario* is formally represented as a conjunction of relations (constraints) in a fairly rich mathematical language (similar to the concept of clause in mathematical logic, Prolog or resolution based automated theorem proving methods (Chang & Lee 1971, Loveland 1978)). Each scen-



(a) (b) (c)

Figure 5: ***w***–Pareto front of y_4 versus y_9 of BAe function for (a) $y_4 \succ y_9$ and (b) $y_4 \ll y_9$. (c) shape of the complete Pareto front.

ario is a function of inputs, objectives and (eventually) some additional parameters and returns a value between 0 and 1 as a percentage of the relations satisfied for given values of (its) inputs. □

Example 2 Suppose we have a set of 3 scenarios:

S_1: $(y_3 > 100) \wedge (y_2 > 7000)$

S_2: $(y_4 \cdot e^{-x_1} \leq 2x_2 + \sqrt{7y_5}) \wedge (y_1 < 12x_4) \wedge (x_1 + x_2 > 20)$

S_3: $\sin(x_2/100) < 0.54$

We specify those scenarios in the following (C–like) notation:

```
y3 > 100 & y2 > 7000
y4*exp(-x1) <= 2*x2 + sqrt(7*y5) & \
      y1 < 12*x4 & x1 + x2 > 20
sin(x2/100) < 0.54
```

If $x_1 = 1.0$, $x_2 = 17.9$, $x_4 = 5.12$, $y_1 = 59.4$, $y_2 = 7122$, $y_3 = 95$, $y_4 = 360$ and $y_5 = 1400$, then

$$v(S_1) = \frac{1}{2}, \quad v(S_2) = \frac{2}{3}, \quad v(S_3) = 1$$

5.1 Combinations of scenarios

Since the evaluation of scenarios S_i (for $1 \leq i \leq M_s$) gives the set of real values $\{v(S_i) \mid 1 \leq i \leq M_s\}$ the next question is the interpretation the set of scenarios $S = \{S_i \mid 1 \leq i \leq M_s\}$ and its value $v(S)$. This is a problem in itself and there are several possible approaches:

1. Let the designer specify the relative importance of each scenario, using weights w_i or using our preference based method for specifying importance. Then

$$v(S) = \sum_i w_i \cdot v(S_i).$$

2. Apply the following geometric progression: importance factor of scenario S_1 is α ($\alpha < 1$), scenario S_2 is a factor α less important than S_1 (i.e. its importance factor is α^2), scenario S_3 is a factor α less important than S_2 etc., where α is a solution of the system

$$\sum_{i=1}^{M_s} x^i = 1, \quad 0 < x \leq 1.$$

In that case we can compute $v(S)$ as

$$v(S) = \sum_{i=1}^{M_s} \alpha^i \cdot v(S_i) \tag{17}$$

To continue Example 2 using this method we have $M_s = 3$, $\alpha = 0.5437$ and $v(S) = 0.6296$, which means that the set of scenarios is accomplished by approximately 63%.

This method has the advantage that the designer doesn't have to worry about specifying the scenario importance, everything is being calculated automatically. However, this model is rather limited, and obviously depends heavily on the scenario order, so that the designer should at least know what is the most preferable scenario, what is the second one etc.

3. Third method is to consider each scenario independently of the others and to perform parallel, distributed, island–based optimisation where each island is the original problem plus one scenario or one set of scenarios. In this case we can use agents or other methods that can monitor the fulfilment of the scenarios. The advantage of this method is that we do not have to fulfil all scenarios (which might be contradictory to each other) and that the system (through agents) can signal back possibility or impossibility of fulfilments of scenarios. Obviously, this method has the broadest potential and our future work will mainly be in that area.

4. Fourth method is the classical one: use penalty functions (Michalewicz 1995) and penalise the solutions that violate some of those scenarios. However, '*there is no general guideline on designing penalty functions, and constructing an efficient penalty function is quite problem–dependent*' (Gen & Cheng 1997, p. 52).

5.2 System for conceptual design

A system that applies scenarios together with designer interaction and genetic algorithms is simply represented as in Figure 6 and the final fitness function for our engineering design system can be schematically presented as in Figure 7.

For the sake of simplicity, the intermediate fitness function is illustrated for weighted sum based transformation, but we can equally well use any other multi–objective optimisation method. We see, taking into account Figure 1, that the model of our system (reflected, as required for GAs by fitness function), goes through several transformations: first we transform the set of our objectives into a (usually but not necessary one–dimensional) fitness function, which is then, by means of scenarios further transformed into another fitness function which, hopefully, better reflect and refines the designer's needs and ideas. This transformation is at any time interactively changeable by the user (designer).

We have applied our system to the BAe design problem (as described in section 4.2), which is not only an optimisation problem but also a general conceptual design problem (Parmee 1998*a*), and the first results are encouraging.

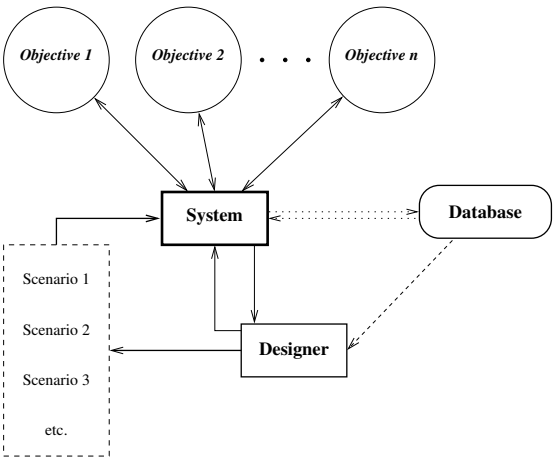


Figure 6: Schema of the computer/human design system.

6 Conclusion and Further Work

In this paper we have presented one method for transforming qualitative characterisation of objective relative importance into quantitative characterisation. One algorithm is given that implements the transformation. Integration with traditional and GA based multi–objective optimisation method is discussed and a novel Pareto optimisation method combination with weights developed. Further, some work on dynamical insertion and deletion of constraints and scenarios is presented and some interaction with agent based systems indicated. Future work on this system will include more tight integration with real world problems and further refinements to both scenario and preference models. Development of fuzzy arithmetic (Kaufmann 1975) scenarios is also planned.

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

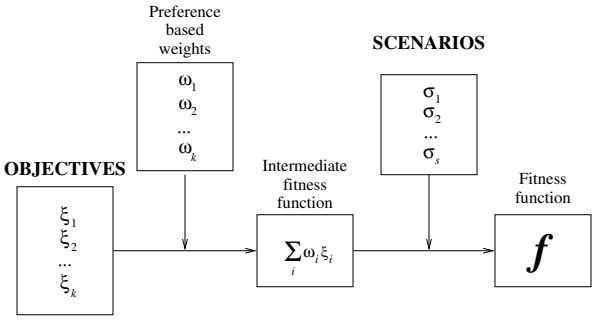


Figure 7: Transformation of fitness function

References

Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M. & Smith, R. E., eds (1999), *GECCO–99: Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, Orlando, Florida, USA.

Ben-Tal, A. (1979), Characterisation of Pareto and lexicographic optimal solutions, *in* G. Fandel & T. Gal, eds, 'Proceedings of the Third Conference on Multiple Criteria Decision Making Theory and Application', number 177 *in* 'Lecture Notes in Economics and Mathematical Systems', Springer Verlag, pp. 1–11.

Chang, C.-L. & Lee, R. C.-T. (1971), *Symbolic Logic and Mechanical Theorem Proving*, Academic Press.

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

Coello, C. A. C. (1999), An updated survey of ga–based multiobjective optimization techniques, Technical Report LANIA RD–98–08, LANIA, Xalapa, Veracruz, México.

Cvetković, D. & Parmee, I. C. (1998), Evolutionary design and multi–objective optimisation, *in* '6th European Congress on Intelligent Techniques and Soft Computing EUFIT'98', Aachen, Germany, pp. 397–401.

Cvetković, D. & Parmee, I. C. (1999), Use of preferences for ga–based multi–objective optimisation, *in* Banzhaf, Daida, Eiben, Garzon, Honavar, Jakiela & Smith (1999).

Cvetković, D., Parmee, I. C. & Webb, E. (1998), Multi–objective optimisation and preliminary airframe design, *in* Parmee (1998*b*), pp. 255–267.

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

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

Gen, M. & Cheng, R. (1997), *Genetic Algorithms & Engineering Design*, Wiley Series in Engineering Design and Automation, J. Wiley & Sons.

Goldberg, D. E. & Deb, K. (1990), A comparative analysis of selection schemes used in genetic algorithms, *in* Rawlins (1990), pp. 69–93.

Horn, J. & Nafpliotis, N. (1993), Multiobjective optimization using the niched Pareto genetic algorithm, Technical Report IlliGAL Report No. 93005, Illinois Genetic Algorithm Laboratory.

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

Kaufmann, A. (1975), *Introduction to the Theory of Fuzzy Subsets*, Vol. I, Academic Press.

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

Lootsma, F. A. (1996), 'A model for the relative importance of the criteria in the multiplicative AHP and SMART', *Eur. J. Operational Research* **94**(3), 467–476.

Lootsma, F. A. (1997), *Fuzzy Logic for Planning and Decision Making*, Delft University of Technology, The Netherlands. Lecture Notes a197.

Loveland, D. W. (1978), *Automated Theorem Proving: A Logical Basis*, North-Holland, Amsterdam.

Michalewicz, Z. (1995), A survey of constraint handling techniques in evolutionary computation methods, *in* 'Proceedings of the 4th Annual Conference on Evolutionary Programming', MIT Press, pp. 135–155.

Mühlenbein, H. & Schlierkamp-Voosen, D. (1993*a*), 'Predictive models for the breeder genetic algorithm I. Continuous parameter optimization', *Evolutionary Computations* **1**(1), 25–49.

Mühlenbein, H. & Schlierkamp-Voosen, D. (1993*b*), 'The science of breeding and its application to the breeder genetic algorithm (BGA)', *Evolutionary Computation* **1**(4), 335–360.

Nisbett, R. E. & 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.

Parmee, I. C. (1998*a*), Exploring the design potential of evolutionary/adaptive search and other computational intelligence technologies, in *ACDM'98* (Parmee 1998*b*), pp. 27–42.

Parmee, I. C., ed. (1998*b*), *Adaptive Computing in Design and Manufacture: The Integration of Evolutionary and Adaptive Computing Technologies with Product/System Design and Realisation. Proceedings of the 3rd Conference on Adaptive Computing in Design and Manufacture (ACDM'98)*, Springer–Verlag.

Rawlins, G. J. E., ed. (1990), *Foundations of Genetic Algorithms 1*, Morgan Kaufmann.

Schaffer, J. D. (1985), Multiple objective optimization with vector evaluated genetic algorithms, *in* J. J. Grefenstette, ed., 'Proceedings of the First International Conference on Genetic Algorithms', Lawrance Erlbaum Associates, pp. 93–100.

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

Veldhuizen, D. A. V. & Lamont, G. B. (1998), Multiobjective evolutionary algorithm research: A history and analysis, Technical Report TR-98-03, Air Force Institute of Technology, Wright–Paterson AFB.

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

Webb, E. (1997), MINICAPS – a simplified version of CAPS for use as a research tool, Unclassified Report BAe-WOA-RP-GEN-11313, British Aerospace.