Use of Preferences for GA–based Multi–objective Optimisation

Dragan Cvetković and Ian C. Parmee
PEDC, University of Plymouth
Drake Circus, Plymouth PL4 8AA, United Kingdom
e–mail: {dcvetkovic,iparmee@plymouth.ac.uk}

Abstract

In this paper we present a method based on preference relations for transforming non–crisp (qualitative) relationships between objectives in multi–objective optimisation into quantitative attributes (i.e. 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.

1 INTRODUCTION

When dealing with industrial design problems, it rapidly becomes apparent that there are significant differences between so called 'textbook optimisation problems' and 'real world applications'. In both cases, in multi–objective optimisation 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 finally $\mathbf{F} : \mathcal{D} \mapsto \mathcal{R}$, 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 (Veldhuizen & Lamont 1998) 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 multi–objective 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.

2 FUZZY PREFERENCES AND ORDERS

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

Preference ordering of the alternatives: an expert e_j provides his preferences on X as an individual preference ordering $O^j = \{o^j(1), \ldots, o^j(m_a)\}$.

Fuzzy preference relation: the expert's preferences on X is described by a fuzzy preference relation, $P^j \subseteq X \times X$ with membership function $\mu_{P^j} : X \times X \mapsto [0, 1]$.

Utility function: an expert e_j provides his preferences on X as a set of m_a utility values, $U^j = \{u_i^j \mid 1 \leq i \leq m_a\}$.

Unfortunately, this is not directly applicable to multi–objective optimisation problems as we are not choosing *one* 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 PREFERENCE ORDER

The *intensity of preference* or, shortly the *preference* of x over y is usually given by $R(x,y) = f(g(x), Ng(y))$, where f is a non–decreasing function of both arguments and N is a strong negation (strictly decreasing continuous involution) (Fodor & Roubens 1994, p. 177).

Strict preference relation P and *indifference* relation I are defined in the following way:

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

If we have a complete (fuzzy-)preference matrix, we would be able to define a complete order among the objects.

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:

$$S_L(a,R) \stackrel{\text{def}}{=} \sum_{c \in A \setminus \{a\}} R(a,c)$$
$$a \geq_L b \ \text{ iff } \ S_L(a,R) \geq S_L(b,R)$$

Example 1 Wine experts give their preferences on five Médoc wines a, b, c, d and e using the following matrix (Fodor & Roubens 1994, p. 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 OUR APPROACH

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* will be treated exactly as *equally important*. There are some psychological explanations for this, namely that if we don't care in respect of two objectives, then we also don't 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 requires in the worst case $k(k \Leftrightarrow 1)/2$ questions, we first ask the designer to identify the objectives of interest at this stage of the optimisation process and to 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).

3.1 PROPERTIES OF 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*
- Relations $\prec$ and $\ll$ are *strict orders*
- Relation $\approx$ is *congruent* with $\ll$ and $\prec$:
  $x \prec y \wedge y \approx z \Rightarrow x \prec z$
  $x \ll y \wedge y \approx z \Rightarrow x \ll z$
- Relation $\ll$ is sub–relation of $\prec$
- Miscellaneous properties:
  · $!x \vee \neg x$, · $x \prec y \wedge y \ll z \Rightarrow x \ll z$
  · $!y \wedge \neg x \Rightarrow x \ll y$, · $\neg x \wedge \neg y \Rightarrow x \approx y$

We can define predicates $\succ$ (is more important) and $\gg$ (is much more important) in the following way:

$$x \succ y \stackrel{\text{def}}{\Leftrightarrow} y \prec x, \qquad x \gg y \stackrel{\text{def}}{\Leftrightarrow} y \ll x$$

3.2 DESCRIPTION OF THE ALGORITHM

- Let the set of objectives be $O = \{o_1, \ldots, o_k\}$. Construct the equivalence classes $\{C_i \mid 1 \leq i \leq m\}$ according to $\approx$ and choose one element x_i from each class 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 can further assume that $\alpha < \gamma < \varepsilon = 1/2 < \delta < \beta$. We assume 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:

$$\begin{aligned} x_i \ll x_j &\Leftrightarrow R(i,j) = \alpha,\ R_a(i,j) = 0,\ R_a(j,i) = 2 \\ x_i \prec x_j &\Leftrightarrow R(i,j) = \gamma,\ R_a(i,j) = 0,\ R_a(j,i) = 1 \\ x_i \approx x_j &\Leftrightarrow R(i,j) = \varepsilon,\ R_a(i,j) = 1,\ R_a(j,i) = 1 \end{aligned} \quad (3)$$

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

  Note: This valuation gives already the idea how to generalise preferences to have s stages instead of only 5 (from "much less important" to "much more important"): if x_i is (say) s' times more important the x_j, we will simply assign $R_a(i,j) = s'$ and $R_a(j,i) = 0$ etc.

- Perform the following procedure:
  1. For all $i \le m$ and for all $j \le m$ such that $j \ne i$ 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 (3) set $R_a(i,j)$ and $R_a(j,i)$ accordingly.
     - Using Warshall's algorithm (Warshall 1962), compute transitive closure of R_a (some modifications are necessary but straightforward).
  2. Using (3), calculate matrix R from R_a.
  3. For each $x_i \in \mathcal{X}$ compute weight as a normalised leaving score:

$$w(x_i) = \frac{S_L(x_i, R)}{\sum_{x_j \in \mathcal{X}} S_L(x_j, R)}. \quad (4)$$

     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 $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$ where $x_i \in C_i$ for $1 \le i \le 4$. Assign initially R and R_a to $\mathbf{E}_4$ — 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 one $x_1 \prec x_4$. The fourth question gives the answer $x_2 \ll x_3$ and after performing transitive closure, we have

$$R_a = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 1 & 2 & 1 & 1 \end{bmatrix}$$

Since for each pair (i,j) we have $R_a(i,j) + R_a(j,i) \ne 0$, we have enough information (without computing transitive closure we would have to ask 6 questions and additionally handle non–consistent answers), and we can construct the matrix R. 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[1] is $x_2 \ll x_3 \prec x_1 \prec x_4$. Weights are further calculated using (4): $w(x_1) = 0.325$, $w(x_2) = 0.025$, $w(x_3) = 0.275$, $w(x_4) = 0.375$, and after normalisation $w(o_1) = w(o_2) = 0.2407$, $w(o_3) = w(o_4) = 0.0185$, $w(o_5) = 0.2037$, $w(o_6) = 0.2778$ and $w(o_7) = 0$.

4 APPLICATIONS OF PREFERENCES

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

1 weighted sum based optimisation, and
2 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.

4.1 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 3.2.

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 1 show results using different preferences on weighted sum optimisation of BAe function (explained in section 4.3).

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

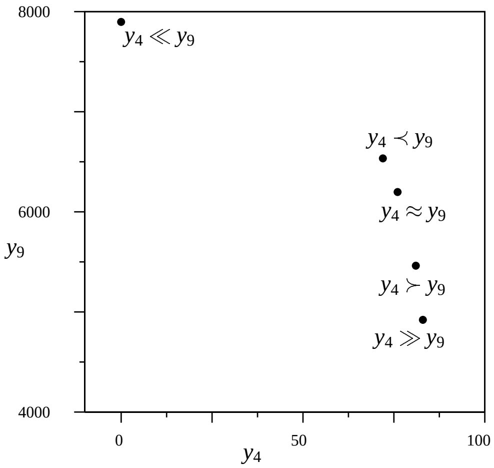

Figure 1: The Influence Of Preference Settings.

However, weighted sum based GA optimisation methods (although very useful for multi–objective optimisation in general) are generally not suitable for optimisation during the conceptual design. 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 re–calculation and normalisation of weights many times (Cvetković et al. 1998, p. 259).

4.2 PARETO OPTIMISATION

Comparing Pareto principle based multi-objective optimisation with lexicographic order based optimisation, we see two extremes concerning the objective importance: in the case of 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 would translate qualitative specification into quantitative. Without this combination, our modified Pareto method would suffer from the same problem as the weighted sum method: how to specify weights in the case of 15–20 or more objectives (it is estimated that 7 ± 2 is the maximal number of chunks a person can work on at the same time).

4.2.1 Definition Of The Modified Pareto method

In order to avoid any terminological confusion, we give the definition of *non–dominance*:

Definition 2 We say that (in object space) 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,

$$\boldsymbol{x} \succeq \boldsymbol{y} \Leftrightarrow \frac{1}{k} \sum_{i=1}^{k} I_{\geq}(x_i, y_i) \geq 1, \tag{5}$$

where

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

We can generalise (5) and say (assuming $\sum_{i=1}^{k} w_i = 1$):

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

or we can even put some threshold $\tau \leq 1$ and say

$$\boldsymbol{x} \succeq_w^\tau \boldsymbol{y} \text{ if and only if } \sum_{i=1}^{k} w_i \cdot I_{\geq}(x_i, y_i) \geq \tau. \tag{7}$$

Definition 3 We will call relation $\succeq_w$ defined by (6) $\boldsymbol{w}$–*non-dominance* and the relation $\succeq_w^\tau$ defined by (7) $(\boldsymbol{w}, \tau)$–*non-dominance*.

Note: The standard dominance relation is just a special case of (7) for $\boldsymbol{w} = (\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k})$ and $\tau = 1$.

Example 3 Let $\mathbf{F}(\boldsymbol{x}) = (x_1^2, x_2^2, x_3^2, x_4^2)$ and $\boldsymbol{w} = (1/2, 1/3, 1/6, 0)$. Using above defined orders, we have:

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

Note that 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.

Definition 4 The *Pareto front* is defined as a maximal set of non–dominated elements (according to a 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 (i.e. according to the order $\succeq_{\boldsymbol{w}}^\tau$) $\succeq_{\boldsymbol{w}}$ and $\succeq_{\boldsymbol{w}}^\tau$ given by (6) and (7) respectively. We assume that at least one of the inequalities is strict.

As mentioned before, vector $\boldsymbol{w}$ could be either specified directly by the designer or it can be calculated from his preferences which would help the designer to work in more qualitative terms without the burden to reason if the weight should be set of 0.5 or to 0.6 and how is it going to affect his results.

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

4.2.2 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 4.3, 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 ($n = k = 2$):

$$(f_1, f_2)(x_1, x_2) = (\sin(x_1^2 + x_2^2 \Leftrightarrow 1), \sin(x_1^2 + x_2^2 + 1))$$

for $x_1, x_2 \in [0, 3\pi/4]$. Obviously, f_1 is at maximum for $x_1^2 + x_2^2 = \pi/2 + 1 \approx 2.5708$ and f_2 is at maximum for $x_1^2 + x_2^2 = \pi/2 \Leftrightarrow 1 \approx 0.5708$ – so they don't have joint maximum. 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 2. Results were obtained running GA (described in more details in section 4.3) for 15 generations.

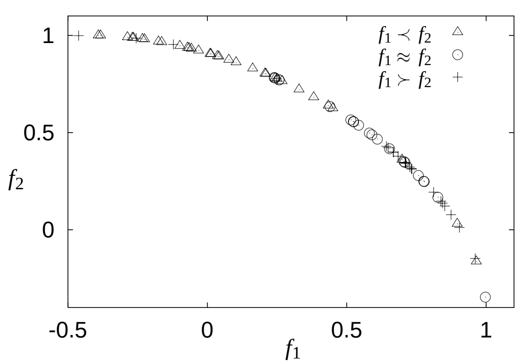

Figure 2: Different Preferences Produce Different Parts Of Pareto Front. Test Function Case.

4.3 BAE FUNCTION AND PARETO

The British Aerospace (BAe) design problem is presented in (Cvetković et al. 1998). Briefly (at the moment, as the complexity of the model is constantly increased), we have 9 input variables $x_1, \ldots, x_9$ and 13 output objectives $y_1, \ldots, y_{13}$ to optimise simultaneously. 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 (masked out).

However, BAe design problem is not only an optimisation problem, actually, optimisation is a rather small part of it. The problems of conceptual design relate to the fuzzy nature of initial design concepts and the many different variants that engineer wishes to try. Computers should be able to help the exploration of those variants whilst also suggesting some others as well (Cvetković et al. 1998, Parmee 1998*a*). Therefore, interaction with the designer (team) is very important. Our goal is to assist designer in the preliminary phase design process (more in the sense of MCDA (multiple criteria decision aid), then MCDM (multiple criteria decision making) (Carlsson 1996).

The core of the genetic algorithm we have used is based on the Breeder Genetic Algorithm (Mühlenbein & Schlierkamp-Voosen 1993). It utilises genetic operators suitable for real valued chromosomes (arithmetic crossover, exponential mutation etc.), and is adapted to use techniques for multi–objective optimisation (Cvetković et al. 1998, Cvetković & Parmee 1998). Considering the Pareto front obtained using Genetic Algorithm, Figure 3(a) shows the final size of the $(\boldsymbol{w}, \tau)$–Pareto front at the end of GA run for different weights $\boldsymbol{w}$ and different Pareto thresholds τ. Results were obtained running GA with population size 50 for 200 generations with τ from 0 to 1, step 0.1, and y_4–weight from 0 to 1 in steps of 0.05, and are averaged over 15 runs (3465 runs all together). Figure 3(b) show two $\boldsymbol{w}$–Pareto fronts of y_4 versus y_9 for different preferences i.e. for different weights together with the shape of the complete Pareto front. We can see immediately from Figure 3 that by varying Pareto threshold and the weights of each objective, we can obtain different Pareto fronts. Exact relationship and applicability will be investigated in further papers. Knowing the behaviour of Pareto front with respect to those parameters, 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.).

5 CONCLUSION

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/preferences developed. Some applications of preferences in the new Pareto based method are presented. In the future work we will try to fur-

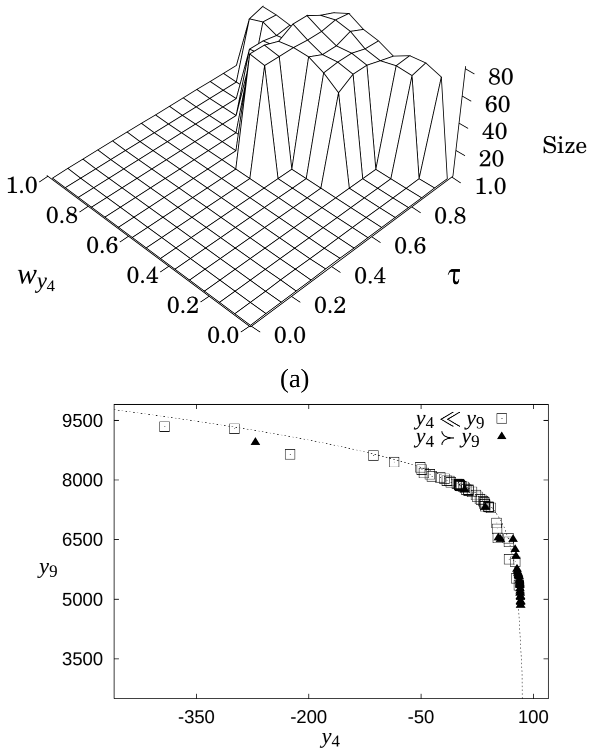

Figure 3: (a) Size of y_4 vs. y_9 $(\boldsymbol{w}, \tau)$–Pareto Front of the BAe Function as a Function of $\boldsymbol{w}$ and τ. (b) $\boldsymbol{w}$–Pareto Front of y_4 vs. y_9 of the BAe Function for $y_4 \succ y_9$ and $y_4 \ll y_9$ Together With the Shape of a Complete Pareto Front.

ther develop the preference model and to integrate it more tightly into the real world applications.

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

References

Carlsson, C. (1996), Active decision support systems, *in* '4th European Congress on Intelligent Techniques and Soft Computing EUFIT '96', ELITE–Foundation, Aachen, pp. 1279–1288.

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

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. & 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.

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

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.

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

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.

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.