Human Preferences and their Applications in Evolutionary Multi–Objective Optimisation*

Dragan Cvetković[1] and Carlos A. Coello Coello[2]

[1] Soliton Inc.
44 Victoria Street 2100
Toronto ON M5C 1Y2
Canada

[2] CINVESTAV-IPN
Evolutionary Computation Group
Departamento de Ingeniería Eléctrica
Sección de Computación
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México D.F. 07300, México

Summary. This chapter talks about (human– and machine–generated) preferences, and their use in multi–objective optimisation. In the first part, preferences are introduced and discussed; utility function based preferences (American school) and outranking based preferences (French school) are presented and their properties explored and examined. Issues such as transitivity and group preferences are also discussed. In the second part, the integration of preferences with various evolutionary multi–objective optimisation methods is introduced and various applications thereof presented. Finally, a brief example of machine (agent) generated preferences is given.

1 Introduction

The Oxford dictionary [56], after a couple of circular definitions, defines preference as *the favouring of one person etc. before others.* The logic of preference, according to the Encyclopaedia Britannica [60], *seeks to systemize the formal rules that govern the conception "x is preferred to y"*. It is usually assumed that preferences are issued by humans (individuals or groups), but the word "human" is nevertheless used in the title to indicate that machine generated preferences are also used as it will be shown later on.

Apart from the logic of preferences with its various sets of axioms ([13, 37, 48, 80, 81], and many others), preferences are also used in Multiple Criteria

* RCS version 1.21, last checked in on February 23, 2004, TEX–ed 3rd March 2004.

Decision Methods (MCDM) and Multiple Criteria Decision Aid (MCDA) [5, 32, 41, 70, 76]. The distinction between these two are, in part, explained by Roy [62] and are also briefly discussed in Section 1.3. There is a vast literature in the field and the aforementioned references are just a very small part of it.

As mentioned by Marchant et al. [47, p. 4] in the context of election and by Vincke [78] in a more general context, preference modeling is a two stage process: *preference expression*, where each decision maker notes preferences among the candidates/alternatives and *the aggregation* or comparison – the process used to extract the best candidate/alternative according to expressed preferences. In this chapter, this two stage process will be followed.

This chapter has the following outline: First, we provide some basic concepts related to multi–objective optimization, which are intended to make the chapter self-contained. Then, a brief description of the two main schools in preference modeling is provided: the French school and the American school. After introducing the main characteristics of both methods, their differences, advantages and disadvantages will be discussed. The second part of the chapter introduces some applications of use of preference methods in multi–objective optimisation.

1.1 Basic Concepts

The emphasis of this chapter is the solution of multi–objective optimisation problems (MOPs) of the form:

$$\text{minimize} \;\; [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})] \tag{1}$$

subject to the m inequality constraints:

$$g_i(\boldsymbol{x}) \leq 0 \quad i = 1, 2, \ldots, m \tag{2}$$

and the p equality constraints:

$$h_i(\boldsymbol{x}) = 0 \quad i = 1, 2, \ldots, p \tag{3}$$

where k is the number of objective functions $f_i : \mathbf{R}^n \rightarrow \mathbf{R}$. We call $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ the vector of decision variables. We wish to determine from among the set $\mathcal{F}$ of all vectors which satisfy (2) and (3) the particular set of values $x_1^*, x_2^*, \ldots, x_n^*$ which yield the optimum values of all the objective functions.

1.2 Pareto optimality

It is rarely the case that there is a single point that simultaneously optimises all the objective functions of a multi–objective optimisation problem. Therefore, we normally look for "trade-offs", rather than single solutions when dealing with multi–objective optimisation problems. The notion of "optimality" is therefore, different in this case.

The most commonly adopted notion of optimality is that originally proposed by Francis Ysidro Edgeworth [31] and later generalised by Vilfredo Pareto [52]. Although some authors call this notion *Edgeworth-Pareto optimality* (see for example [72]), we will use the more commonly accepted term: *Pareto optimality.*

Definition 1. *A vector of decision variables $\boldsymbol{x}^* \in \mathcal{F}$ is* Pareto optimal *if there does not exist another $\boldsymbol{x} \in \mathcal{F}$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for all $i = 1, \ldots, k$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one j.*

In words, this definition says that $\boldsymbol{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables $\boldsymbol{x} \in \mathcal{F}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Unfortunately, this concept almost always gives not a single solution, but rather a set of solutions called the *Pareto optimal set.* The vectors $\boldsymbol{x}^*$ corresponding to the solutions included in the Pareto optimal set are called *nondominated.* The image of the Pareto optimal set under the objective functions is called *Pareto front.*

1.3 MCDM and MCDA

Much of the following description is based on [62]. Until the end of 1960s in operations research (OR), decision–making problems were formulated on the following three bases:

1. A well–defined set A of feasible alternatives a;
2. A real–valued function g defined on A precisely reflecting the preference of the decision maker (DM).
3. A well–formulated mathematical problem:

$$\text{Find } a^* \text{ in } A \text{ such that } g(a^*) \geq g(a) \; \forall a \in A.$$

Accordingly, the general framework of MCDM consists of:

1. A well–defined set A of feasible alternatives a;
2. A model of preferences, well shaped in DM's mind, rationally structured from a set of attributes. The preferences are defined using utility function U:

$$\begin{aligned} a'Pa &\text{ if and only if } U(a') > U(a) \\ a'Ia &\text{ if and only if } U(a') = U(a) \end{aligned}$$

3. A well–formulated mathematical problem: the discovery of an optimal alternative a^* in A such that $U(a^*) \geq U(a) \; \forall a \in A$.

The practice of operations research (OR) and MCDM has shed light on some fundamental limitations on objectivity. Five major aspects have to be taken into account (see also [24, 19]):

1. The frontier of A (of feasible alternatives) is often fuzzy. Because of this, the borderline between what is and what is not feasible has inevitably a certain amount of arbitrariness. A more crucial limitation on objectivity comes from the fact that this borderline is frequently modified in the light of what is found through the decision process itself [35].
2. In many real world problems, the decision maker (DM), as a person truly able to make the decision, does not really exist: usually, several people take part in the decision process and we tend to confuse the one who ratifies the decision with what is called the decision maker.
3. Even when the DM is not a mythical person, the DM's preferences very seldom seem well–stated: in and among the areas of firm convictions lie hazy zones of uncertainty, half–held belief or, indeed, conflicts and contradictions. We have to admit, therefore, that the study itself contributes to answering questioning, solving conflicts, transferring contradictions and destabilising certain convictions.
4. Data such as the numerical values of performances $g_k(a)$, the analytical forms of distributions such as $\delta_k^a(y_k)$ or $\delta^a(y_1, \ldots, y_n)$ and numerical values of the characteristics of those distributions are, in many cases, imprecise and/or defined in an arbitrary way.
5. In general, it is impossible to say that a decision is a good one or a bad one by referring only to a mathematical model: organisational, pedagogical, and cultural aspects of the whole decision process which leads to making a given decision also contribute to the quality and success of this decision.

Therefore, the general framework of multiple criteria decision aid (MCDA) consists of:

- A not necessarily stable set A of potential actions s;
- Comparison based on n criteria (or pseudo–criteria) g_k;
- An ill–defined mathematical problem.

Munda [50, p. 53] gives some requirements for an MCDA:

1. One should disregard those procedures which entail the weighting of criteria.
2. Interactive procedures are the only ones which actively involve a decision maker and as a result of this they represent the most desirable approach.
3. Since imprecision (quantitative and qualitative information) and uncertainty (stochastic and fuzzy) are the main features of social systems, there is a clear need for MCDA methods able to take into account the possible kinds of 'mixed information'.
4. The use of fuzzy sets is desirable for three reasons:
   a) It is possible to deal in a suitable manner with the ambiguity often present in the available information.
   b) It is possible to do more justice to the subjective or creative component of the individual decision maker.

   c) It is possible to interact with a DSS (decision support system) in natural language by employing linguistic variables.
5. When a DSS is constructed, it is necessary to remember that sophisticated and complex mathematical models are meaningless without a computer implementation, while computer models without strong underlying mathematical and philosophical models can become nice but 'empty' boxes; therefore there are two sides to the coin.
6. MCDA should mainly be devoted to the choice process.
7. According to MCDA philosophy, the subjective preferences of the decision maker must play a central role in the choice process. On the other hand, it is necessary to note that the results of a decision model depend on:
   a) available data,
   b) structured information,
   c) chosen method,
   d) decision maker's preferences.

2 Preference expression

As mentioned before, there are two main schools here:

- French school with outranking methods;
- American school with utility function;

Similarly to European and American contracts in stock/option trading [40], not all methods developed by French researchers are outranking based and also not all methods developed by American authors are based on utility functions.

We describe first outranking and then utility functions.

2.1 Outranking

The basics of outranking is given by the following (generalised) definition [77, p. 3]:

Definition 2. *A* $\{P, Q, I, J\}$ *– preference structure on a set* A *is a set of four binary relations* P *(strict preference),* Q *(weak preference),* I *(indifference) and* J *(incomparability), defined on* A *such that:*

- $\forall a, b \in A$, aPb *or* bPa *or* aQb *or* bQa *or* aIb *or* aJb *(exclusive or);*
- P *and* Q *are asymmetric;*
- I *is reflexive and symmetric;*
- J *is irreflexive and symmetric;*

Traditionally, assigned semantics to these relations is that a is strongly preferred to b if the difference in their values is larger than the first threshold, weakly preferred if the value is between the first and the second threshold and indifferent if the value is less than the second threshold. One generalised model is the M_1 model in [77]:

Definition 3. *A $\{P, Q, I, J\}$ – preference structure is representable by the model M_1 if and only if there exist a real–valued function g defined on A, and real–valued functions p and q defined on $\mathbf{R}$ such that $\forall a, b \in A$:*

$$(M_1) \begin{cases} aPb \Rightarrow g(a) - g(b) > p(g(b)); \\ aQb \Rightarrow p(g(b)) \geq g(a) - g(b) > q(g(b)); \\ aIb \;\; \Rightarrow \begin{cases} q(g(b)) \geq g(a) - g(b), \\ q(g(a)) \geq g(b) - g(a), \end{cases} \end{cases}$$

Due to the presence of the incompatibility relation J, the implications in (M_1) cannot be reversed.

In model (M_1), the only condition imposed on the threshold functions is that $\forall a \in A$,

$$0 \leq q(g(a)) \leq p(g(a))$$

The most popular outranking based methods are various variants of ELECTRE (I, II, III, IV, TRI, IS) [17, 63] and PROMETHEE (I and II) [10] and many others. However, it seems that a large number of applications actually use the PROMETHEE II method which gives a complete order among the alternatives and does not take into account the incomparability information that most outranking methods provide.

Bouyssou et al. [8] give a general model of preference aggregation. The authors analyse several more and more general models and end up with model M_4:

$$x \succeq y \text{ iff } F(\psi_i(u_i(x_i), u_i(y_i))_{i=1,\dots,n}) \geq 0$$

where $u_i : X_i \mapsto \mathbf{R}$, $F : \mathbf{R}^n \mapsto \mathbf{R}$ is a strictly increasing function and $\psi_i : \mathbf{R}^2 \mapsto \mathbf{R}$ is nondecreasing in the first and nonincreasing in the second argument for $1 \leq i \leq n$. X_i is the set of evaluations of alternatives with respect to criterion i.

They show how some popular outranking methods (ELECTRE, TACTIC, etc.) could be mapped onto that model.

2.2 Utility Functions

Axiomatic theory of utility functions to measure individual or group preferences was developed by John von Neumann and Oskar Morgenstern in 1947 [79]. Later on, it was further extended to Multi–attribute utility functions, that integrate objective functions into the preference structures [43, 59, 44, 6].

Whereas the outranking method does the pairwise ranking, the utility function $u(\cdot) : A \mapsto \mathbf{R}$ assigns a numerical value to each alternative and the preference modelling is simply provided by

$$aPb \text{ iff } u(a) > u(b), \quad aIb \text{ iff } u(a) = u(b)$$

Here $P \subseteq A^2$ is a strict preference relation and $I \subseteq A^2$ is an indifference relation. Due to the definition, and the properties of $<$ and $=$ on a real axis, P is a complete (strict) order whereas I is an equivalence relation.

The advantage of utility functions over outranking is that the former is much easier to handle afterwards since the order exhibits nice properties. The disadvantage is that the user has to specify the value of each alternative instead of only doing pairwise comparison like in the outranking methods. However, outranking methods are in general harder to apply and most of the applications using it employ PROMETHEE II where the preference relation is a complete order.

2.3 PEDC preference method

One method, by Cvetković et al [19, 23] used within the Plymouth Engineering Design Centre (PEDC) [54] in a way combines both schools: it establishes the preference by pairwise comparisons using linguistic concepts (or fuzzy preference relations) such as:

a is much less important than b	$(a \ll b)$;
a is less important than b	$(a \prec b)$;
a and b are equally important	$(a \approx b)$;
a is more important than b	$(a \succ b)$;
a is much more important than b	$(a \gg b)$;
don't care whether a or b	$(a\#b)$;
a is important	$(!a)$;
a is not important	$(\neg a)$;

with the following set of properties:

- Relation $\approx$ is an *equivalence relation* (reflexive, symmetric and transitive):

$$x \approx x \tag{4}$$

$$x \approx y \Rightarrow y \approx x \tag{5}$$

$$x \approx y \wedge y \approx z \Rightarrow x \approx z \tag{6}$$

- Relations $\prec$ and $\ll$ are *strict orders* (irreflexive and transitive):

$$x \not\prec x \tag{7}$$

$$x \not\ll x \tag{8}$$

$$x \prec y \wedge y \prec z \Rightarrow x \prec z \tag{9}$$

$$x \ll y \wedge y \ll z \Rightarrow x \ll z \tag{10}$$

- Relation $\approx$ is *congruent* with $\ll$ and $\prec$:

$$x \prec y \wedge y \approx z \Rightarrow x \prec z \tag{11}$$
$$x \ll y \wedge y \approx z \Rightarrow x \ll z \tag{12}$$

- Relation $\ll$ is a sub–relation of $\prec$:

$$x \ll y \Rightarrow x \prec y \tag{13}$$

- Miscellaneous properties:

$$!x \vee \neg x \tag{14}$$
$$!y \wedge \neg x \Rightarrow x \ll y \tag{15}$$
$$\neg x \wedge \neg y \Rightarrow x \approx y \tag{16}$$
$$x \prec y \wedge y \ll z \Rightarrow x \ll z \tag{17}$$

This set of properties is not a minimal one, as some of the properties could be inferred from the others. Since $\succ$ and $\gg$ can be (and are) defined in terms of $\prec$ and $\ll$, the above properties are expressed in terms of $\ll$ and $\prec$ only. Note that the $\ll$, $\prec$ and $\approx$ are all transitive relations.

Once preferences are established, a complete order can be constructed. The algorithms and the details are given in [19]. However, the general idea of the approach can be better understood through an example.

Example 1. (condensed version of one given in [54]) Suppose that there are seven objectives $y_1, \ldots y_7$, but the designer wishes to ignore the objectives y_3 and y_6 at this stage of (conceptual) design. Further, suppose that y_1 and y_2 are equally important and that y_1 is more important than y_5 and y_7 but less important than y_4 and that y_7 is much less important than y_5. This set of preferences can be expressed in the following way:

$$\neg y_3,\ \neg y_6 \tag{18}$$
$$y_1 \approx y_2 \tag{19}$$
$$y_1 \prec y_4,\ y_1 \succ y_5,\ y_1 \succ y_7 \tag{20}$$
$$y_5 \gg y_7 \tag{21}$$

Using the properties of relations, the following order is established:

$$y_4 \succ y_1 \approx y_2 \succ y_5 \gg y_7 \gg y_3 \approx y_6.$$

The above order can be deduced using the listed properties only, but another method, more computer friendly, is also available and is based on fuzzy preference matrix and on *graph leaving score* [33, 19]: for assigned numeric valuation to $\prec$, $\ll$, $\approx$, $\gg$ and $\succ$, construct a (fuzzy) matrix of preferences $R_{i,j}$, compute leaving score $S_L(a, R) = \sum_{c \in A \setminus \{a\}} R(a, c)$ for each alternative a and assign order

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

2.4 Other methods

The following two preference methods are historically important:

de Condorcet method [18, 47], originated in 1785: alternative A is favoured over B if the number of criteria where A is better than B is greater than the number of criteria where B is better than A. This method is e.g. also used by Drechsler et al. [29];

Borda method [7], dating back to 1781. It uses the following ranking method: If n candidates are voted for, a voter gives 1 point to his most favoured candidate, 2 points to the next one and so on until reaching the last one who is given n points. For each candidate, Borda score is computed i.e. the sum of points over all voters. The candidate with lowest Borda score is chosen.

3 Preference features

There are a couple of issues here that are considered controversial in preference design and establishment. These issues include:

- the issue of transitivity, indifference and noncomparability;
- the issue of group preferences.

We are going to discuss them in the following sections.

3.1 The issue of transitivity

Definition 4. *Relation $R \subseteq A^2$ is* transitive *if it satisfies the following property:*

$$(\forall a, b, c \in A) \quad aRb \wedge bRc \rightarrow aRc.$$

Whether preferences should be transitive is a matter of great controversy in the Logic of Preferences field [80]. Some authors argue that human preferences are not necessarily transitive and give very good examples of non–transitivity of preference relations [75].

The transitivity issue is a hot topic in the MCDA community as well. There are two main examples why some authors consider transitivity undesirable. The following examples are from [47, p. 16]:

- **Transitivity of indifference** To explain why transitivity of indifference is considered bad, Luce [46] gives a coffee example: since a decision maker (DM) can't make a difference between coffee with n grains of sugar and $n + 1$ grains of sugar, using transitivity we can infer that the DM can't make a difference between coffee with 1 grain of sugar and a coffee with 1000 grains of sugar (2 full spoons). Therefore, some systems drop the transitivity of indifference. The corresponding algebraic structure is called *semiorder* [47, 46].

- **Indifference and preferences** It is usually assumed that if the DM is indifferent between A and B and prefers A to C, then the DM prefers B to C. However, this shouldn't be taken for granted as the following example shows:

  Example 2. A child is asked to choose between two birthday presents: a pony and a blue bicycle. Since it likes both of them very much, it can't choose and is therefore indifferent between the pony and the bicycle. The child is further asked to choose between the blue bicycle and a red bicycle with a small bell and of course in that case, it chooses the red bicycle. However, we can't infer that the child would prefer the red bicycle to the pony, since after adding the bell the child could still be indifferent between the two.

The transitivity property is usually desirable since it simplifies the model but there are a lot of critics that argue that the human mind (or, rather, the human brain) doesn't really think in terms of transitivity, as illustrated above.

Furthermore, intransitivity of preferences could yield to cyclic preferences (e.g. $A \prec B \prec C \prec A$) and contradictions. In classical mathematical logic we try to avoid contradictions because by having contradictions, every possible conclusion is derivable. There are some branches of logic that try to bypass this situation i.e. *logic of relevance* [1] and *Paraconsistent logics* [58].

Therefore, a DM needs to be very careful in designing preference models. Note that preference models using indifference and preference thresholds (e.g. ELECTRE) would avoid the aforementioned paradoxes. However, preference thresholds methods pay their price in being more complex and harder to use.

3.2 The issue of group decision making

As mentioned before, the decision maker usually is not a single person so the issue of aggregating multiple preference relations into a single preference has always been an important one. One important group decision making example is the process of voting and elections. Marchant et al. [47] give a nice survey of voting methods employed in different countries.

A classical result in the field is a theorem by Arrow [2] stating that there is no group preference aggregation satisfying some basic principles. Arrow's problem is roughly as follows: Given the ranking of a set of alternatives by each individual in a decision making group, what should the grouping ranking for these alternatives be? He postulated some very reasonable assumptions concerning the aggregation of individuals' rankings, and then he investigated their composite implications. These assumptions are as follows [44, p. 523]:

- **Complete Domain**: The utility function should be able to define an ordering for the group, regardless of the individual members' ordering.

- **Positive Association of Social and Individual Values**: If the group ordering indicates that alternative x is preferred to alternative y for a certain set of individual rankings, and (1) if there are no changes on the ordering of each individual, and (2) each individual's paired comparison against x remains unchanged or is modified in x's favour, then the group ordering must imply that x is still preferred to y.
- **The Independence of Irrelevant Alternatives**: If an alternative is eliminated and the preference relations for the remaining alternatives remain unchanged for all the individuals, then the new group ordering should remain the same as before.
- **Individual's Sovereignty**: For each pair of alternatives x and y, there is some set of individual orderings which causes x to be preferred to y.
- **Nondictatorship**: It is impossible that the preferences of the group be always in agreement with the preferences of a single individual.

Arrow proved that there is no rule for combining the individual's rankings that is consistent with these seemingly innocuous assumptions.

Arrow's impossibility theorem also has some opponents, as the following quote from Tullock's *Towards a Mathematics of Politics* [74] indicates (quoted from [3]):

> A phantom has stalked the classrooms and seminars of economics and political sciences for nearly fifteen years. The phantom, Arrow's General Impossibility Theorem, has been generally interpreted as proving that no sensible method of aggregating preferences exists. The purpose of this essay is to exorcise the phantom, not by disproving the theorem in its strict mathematical form, but by showing that it is insubstantial. I shall show that when a rather simple and probable type of interdependence is assumed among the preference functions of the choosing individuals, the problem becomes trivial if the number of voters is large. Since most cases which require aggregation of preferences involve a large number of people, "Arrow problems" will seldom be of much importance.

Of course, it must not be forgotten that Arrow's theorem deals with independent individual preferences (such as in voting), whereas in the designer group there is always interactivity so the engineers can sort out their differences (although it might cause the possible danger of dictatorship). Scott et al. [69] conclude that "...engineering design decision–making occupies a middle ground between decision with an idealised decision maker and decision by groups of fully autonomous citizens, and on this middle ground Arrow's Theorem has no detrimental consequences".

Also of interest is a theorem relating to *non–manipulability*. Informally, a method is non–manipulable if, in no case, a voter can improve the results of the election in his view by not reporting his true preferences. The theorem by Gibbard and Satterwhaite is as follows [47, 64]: When the number of candidates is larger than two, there exists no aggregation method satisfying simultaneously the properties of universal domain, non–manipulability and non–dictatorship.

The method developed by Baron de Condorcet described in Section 2.4 above is also not without its problems as the following example shows [47, p. 4]:

Example 3. Let $\{a, \ldots, z\}$ be the set of 26 candidates for 100 voters election. Suppose that:

- 51 voters have preferences $aPbPcP \ldots PyPz$;
- 49 voters have preferences $zPbPcP \ldots PyPa$.

It is clear that 51 voters vote for a while 49 vote for z. Thus a will win although almost 50% of the voters consider him the worst one. In the above example b would be a much better compromise.

Its contemporary, the Borda method, is prone to manipulability as the following example shows:

Example 4. Suppose that

- First voter has preferences $aPcPb$;
- Second voter has preferences $bPaPc$

Then Borda scores are $B(a) = 3$, $B(b) = 4$, $B(c) = 5$ i.e. a is the winner. However, if the voters change their preferences to

- First voter has preferences $aPbPc$;
- Second voter has preferences $bPcPa$

then Borda scores are $B(a) = 4$, $B(b) = 3$, $B(c) = 5$ i.e. b is the winner. This means that although voters didn't change their preferences regarding a and b, by changing relative position of c they have changed the winner.

4 The use of preferences

Evolutionary multi–objective optimisation (EMOO) which started in the mid-1980s with the work of Schaffer [65, 66] and Fourman [34], gained quite in momentum in the last decade. Figure 1, plotting the number of EMOO publications per year, shows that 3 times as many papers on EMOO were published from 1994–2003 as in the first 10 years (1984–1993). This plot was produced using the statistics collected at the EMOO repository (`http://delta.cs.cinvestav.mx/~ccoello/EMOO`).

Coello [14, 16, 17] gives a very comprehensive survey listing various methods used in EMOO. EMOO applications using preferences are described in [15].

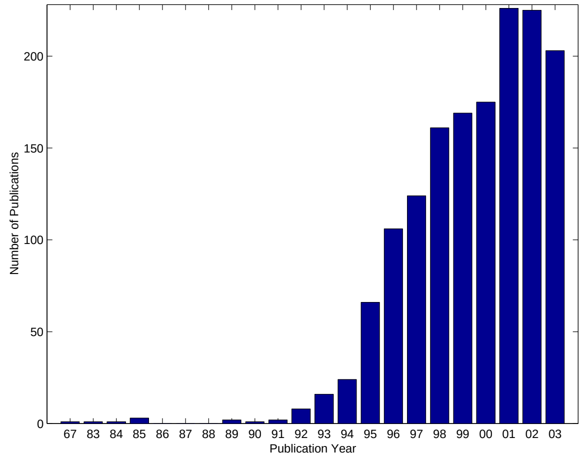


Fig. 1. Publications on evolutionary multi–objective optimisation (until 2003).

In the sequel, it will be shown how preferences can be integrated into some standard multi–objective optimisation methods on the example of PEDC preferences, described in Section 2.3 above. Afterwards, some alternative methods will be shown.

4.1 Integration of preferences into optimisation

Very common techniques used in EMOO include

- weighted sums;
- Pareto methods;
- coevolutionary methods;

In order to integrate preferences in the above methods, a numerical weight corresponding to each relation is given. The techniques used within PEDC [19, 23] rely on the valuation v of preference relations of the form:

$$\begin{array}{l} x \ll y \Rightarrow v(x) = \alpha, v(y) = \beta \\ x \prec y \ \Rightarrow v(x) = \gamma, v(y) = \delta \\ x \approx y \ \Rightarrow v(x) = v(y) = 1/2 \\ 0 \leq \alpha \leq \gamma \leq 1/2 \leq \delta \leq \beta \leq 1, \alpha + \beta = \gamma + \delta = 1 \end{array}$$

and the normalised leaving score of the corresponding preference matrix R, computed as:

$$w(x) = \frac{S_L(x, R)}{\sum_{y \neq x} S_L(y, R)}.$$

Previous work within PEDC used the following values for the parameters $\alpha = 0.05$, $\beta = 0.95$, $\gamma = 0.35$, $\delta = 0.65$, but empirical analysis [23] shows that the method is pretty robust to actual chosen values.

Integrating preferences in weighted sum based methods set is then quite straightforward.

4.2 Computation model and EA description

Examples presented use a computation model based on a miniCAPS conceptual airframe design project, an industrial project developed in cooperation with British Aerospace Systems (BAe). Not all details of the project are relevant here and it is not possible to describe the model in detail. The miniCAPS model [82] is a scaled down version of CAPS (computer aided project studies, BAe software used by designers during the earliest investigation stages of a new aircraft). MiniCAPS reproduces the general characteristics of CAPS but without its computational complexity. It is described in more details in [54], but the short description is provided here for the sake of completeness.

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.

MiniCAPS utilises nine variable parameters producing a total of thirteen outputs, each of which may be considered an objective. Variables (such as climb and cruise Mach numbers, cruise height, wing aspect ratio, etc.) are denoted by $x_1, \dots, x_9$ and objectives (such as landing speed, ferry range, take off mass, etc.) by $y_1, \dots, y_{13}$. Some of the objectives are highly conflicting (e.g., y_4 (specific excess power (SEP) for the supersonic case) and y_9 (ferry range)). All objectives are represented by their numerical values, as calculated by miniCAPS, and are normalised to $[0, 1]$ range. The graphs presented show the original, non–normalised objective values. The names miniCAPS and BAe function would be used as synonyms in the text.

In all cases, a genetic algorithm, described in more detail in [19], is based on Breeder Genetic Algorithm (BGA) [49] and uses the following parameter setting:

- Real–valued chromosomes [49, 68];
- 1–SBX as the crossover operator [25, 28];
- Crossover probability 1.0;
- 2–tournament as the selection method;
- Exponential step mutation with parameter $k = 0.01$;
- Mutation probability $1/n$ (n is the number of input variables) [49, 4];
- Population size 50;
- Number of generations 200;

A typical run takes about 10–15 seconds on a SUN 167MHz UltraSPARC workstation.

Testing has shown that the above algorithm gives the best compromise of quality of solutions versus time required to obtain solutions from the mini-CAPS function.

4.3 Preferences and Pareto method

Weighted Pareto method is based on the following definitions [21, 23]:

Definition 5. *(Weighted dominance relation) For a given weights–vector* $\boldsymbol{w} = (w_1, \ldots, w_k)$ *summing to 1 and a real number* $0 < \tau \leq 1$*, a real vector* $\boldsymbol{x} = (x_1, \ldots, x_k)$ $(\boldsymbol{w}, \tau)$–dominates *a real vector* $\boldsymbol{y} = (y_1, \ldots, y_k)$*, written* $\boldsymbol{x} \succeq_{\boldsymbol{w}}^{\tau} \boldsymbol{y}$*, if and only if*

$$\boldsymbol{x} \succeq_{\boldsymbol{w}}^{\tau} \boldsymbol{y} \;\Leftrightarrow \sum_{i=1}^{k} w_i \cdot I_{\geq}(x_i, y_i) \geq \tau, \tag{22}$$

where

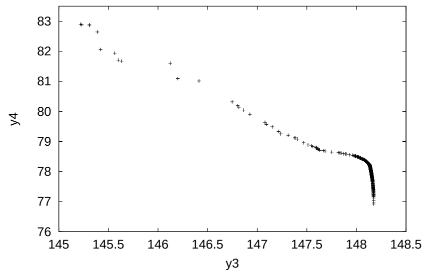
$$I_{\geq}(x, y) = \begin{cases} 1, \; x \geq y \\ 0, \; x < y \end{cases} .$$

The standard definition of Pareto dominance (definition 1 in Section 1.2) is a special case of $(\boldsymbol{w}, \tau)$–dominance for $\tau = 1$ and $w_1 = \ldots \; w_k = 1/k$. Note that the standard definition of dominance requires that at least one of the $x_i \geq y_i$ inequalities is strict. However this is not a problem since these two orders are definable in terms of each other [45].

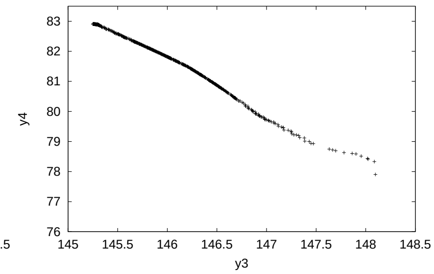
Definition 6. *(Weighted Pareto front) The* $(\boldsymbol{w}, \tau)$–Pareto front *is defined as a maximal (i.e. the largest according to* $\subseteq$ *order) set of nondominated elements according to the given order* $\succeq_{\boldsymbol{w}}^{\tau}$.

Figure 2 shows two $\boldsymbol{w}$–Pareto fronts of y_3 (specific excess power for subsonic case) versus y_4 (specific excess power for supersonic case) objectives for different preferences for the BAe function used in airframe design and described in [24] and in section 4.2 above. Both objectives are maximised (all other objectives are marked as non-important to minimise their influence on the optimisation process).

The method presented here is not the only one possible. There are also some other recent approaches related to the biasing of Pareto front generation. Deb [26, 27] uses weights in a sharing function to modify the density of identified Pareto solutions to generate more Pareto points in the regions of interest and less in other regions. Branke et al. [9] use an approach that changes the shape of the dominance region from a right angle to cover the wider region and to restrict the Pareto front accordingly. However, this research has been restricted to two–dimensional optimisation problems and its generalisation to higher–dimensional spaces is not straightforward. Hughes [38, 39] proposes the use of expressions that incorporate information about both feasibility (i.e., whether or not the constraints of a problem are satisfied) and priority satisfaction of a solution. Such expressions are then used to



(a) $y_3 \gg y_4$



(b) $y_3 \ll y_4$

Fig. 2. $\boldsymbol{w}$–Pareto front of (y_3, y_4) of the BAe function.

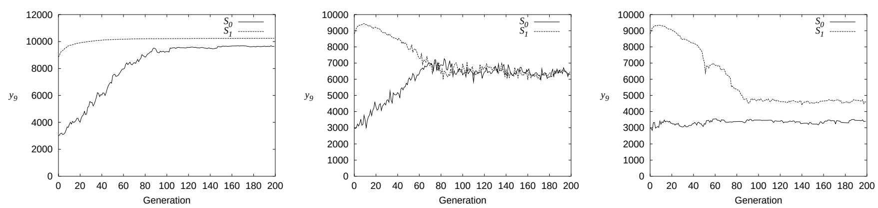
compute domination probabilities (i.e., this information is used to alter the ranking of the population). Drechsler et al. [30, 29] and Schmiedle et al. [67] propose the use of satisfiability classes to model the user's preferences. The approach consists of defining a relation "favour" (also briefly mentionied in Section 2.4), whose concept is similar but not equivalent to Pareto dominance (mathematically speaking, the relation "favour" is not a partial order, because it is not a transitive relation). Tan et al. [73] proposed an approach in which the user can define goals that are then used to modify the ranking assignment process of the evolutionary algorithm used for the search. The approach ranks separately to those individuals that satisfy the goals from those that do not satisfy them. This scheme allows the specification of "don't care" priorities, and non-attainable goals. Several other approaches have also been reported in the literature (see Chapter 8 in [17]).

4.4 Preferences and Coevolution

Coevolution can mean different things in different context (see for example [51, 61, 57]), but in this chapter it is used to describe parallel optimisation processes (genetic algorithms) that try to come to a common solution by means of compromise and penalty maps that penalise solutions being too far away (in variable space) from each other [55, 54]. A coevolutionary multiobjective genetic algorithm is introduced in [55]. In its original form, for k objectives optimisation, k parallel genetic algorithms $S_0, \ldots, S_{k-1}$ are used, each optimising a single objective, and as the run progresses, a penalty function is used progressively to guide solutions towards a common region. The penalty function is a function of the distance between variables, the penalty factor < 1, and of a generation number. Early generations allow results to be far apart, whilst later generations allow not more than 10% of the variable ranges. The penalty function is defined as a monotonically non–increasing real function. The following shapes have been used by Parmee and Watson [55]: constant, full linear, half linear, full cos–like and half cos–like constraint function.

The coevolutionary method described in [55] assumes the equal importance of all objectives, the same way as the Pareto method assumes the equal importance of the objectives.

Since coevolution applies penalties according to the distance in genotype space, it is not straightforward to integrate preferences as it is in e.g. the weighted sum method described in Section 4.1. The solution is to apply the penalties proportionally to the relative importance of the objectives: the more important objective is penalised less then the less important one. That ensures that the processes will converge closer to the better values for the more important objective, as illustrated in Figure 3. Figure 3(b) illustrates the original coevolution method whereas Figures 3(a)&(c) show results affected by the preference use. The objectives optimised are y_3 (specific excess power for subsonic case) and y_9 (ferry range). Again, the objectives are maximised (and all other objectives are marked as non–important). The results shown are averaged over 20 runs.



(a) $y_3 \ll y_9$ (b) $y_3 \approx y_9$ (c) $y_3 \gg y_9$

Fig. 3. Coevolution and different preferences: results for y_9 objective of BAe function. Process S_0 optimises y_3 and process S_1 optimises y_9.

It can be seen that using different preferences, the results are biased more towards regions where the preferred objective has better values. An interesting behaviour is noticeable in Figure 3(c) where the processes are diverging. One possible explanation is that the use of preferences decreases the severity of the penalty factors, so the preferred objective is not "encouraged" towards the compromise region as heavily as in the equal–preference case. More details and discussion are given in [19].

4.5 Machine based preferences in EMOO

Whereas most applications using preferences assume that the preferences are issued by a human DM (or a group of DMs), research on agents within PEDC [22] demonstrates the use of PEDC preferences as generated by machine based agents. The main use of agents and machine based preferences was in closing the interactive loop:

establish preferences, compute results, evaluate results, repeat which is quite common in conceptual engineering design.

The so called *incremental agent* [19, 22] changes the preferences according to the following algorithm (scenarios [20] are constrains that are added or removed interactively i.e. they are not built–in into the computational model and are subject to "more important/less important" ranking the same way the objectives are):

1. Use the original designer's preferences (both for objectives and for scenarios) and run the optimisation process;
2. If some of the scenarios are not fulfilled, suggest increasing importance of those scenarios that are not fulfilled and repeat the search process;
3. If some scenarios are still not fulfilled although they are ranked as the most important, suggest changing variable ranges (of those variables mentioned in the scenarios) and repeat the search with this new setting;
4. If some scenarios are still not fulfilled, give up and report the results to the designer.

Arguably, the intelligent agent used here is a very simple and not so intelligent one, but it nevertheless frees the designer (or DM) from mundane tasks.

5 Discussion and conclusion

In this chapter we have introduced and discussed preferences, both human and machine generated, and their application in multi–objective optimisation. Two major preference approaches have been introduced: outranking, and utility function — each method has its advantages and both methods continue to exist in parallel. Some problems facing preference methods (e.g. of transitivity and group decision making) are addressed. Two historically important preference (voting) methods have also been mentioned (de Condorcet and Borda) and discussed to some extent.

The second part of this chapter describes some EMOO methods incorporating preferences. In particular, the PEDC preference method (Section 2.3) is presented as an example of preference and optimisation method integration. A few examples have been shown that integrate preferences with weighted sum optimisation, Pareto optimisation and coevolutionary methods. Finally, machine based agents have entered the design loop.

There are several other aspects of preference use in EMOO not discussed here. They include restricting the set of potential candidates and the choice of final solution(s), user interface aspects of it (including both preference input and presenting the chosen solutions [53]) as well as the most suitable media for preference expressions. The use of minimal set of preference constraints (as used e.g. by Greenwood et al. [36]) and the uniqueness and unambiguity of preference representation in the optimisation context is also not addressed.

Does expressing preferences in two different ways yield to the same set of preferences and to the same set of solutions?

The use of preferences in multi–objective optimisation faces some challenges not apparent in its classical "use preferences to choose from a set of alternatives" framework. Even when preferences are used to restrict the set of possible solutions (or to direct search towards a certain subset of feasible solutions), particularly in continuous spaces, its sheer volume makes it mostly unmanageable and practically infeasible. There are methods developed in the literature addressing this problem, but nevertheless the issue remains as an active research area.

Most of the methods given here in some ways contravene some of Munda's requirements (Section 1.3 in this chapter), but the issues have been identified and decision support systems (DSS) [11, 12] research is growing more and more active. Finally, another "hot topic" addressed in this chapter, and not exclusively relevant to EMOO, is the use of intelligent agents in search and optimisation ([42, 71])

Acknowledgements

The first author acknowledges support of UK Engineering and Physical Sciences Research Council (EPSRC) and British Aerospace Systems and support from the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through project number 34201-A. The second author acknowledges support from the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through project number 34201-A.

References

1. Alan R. Anderson and Nuel D. Belnap, Jr. *Etailment. The Logic of Relevance and Necessity*, volume I. Princeton University Press, 1975.
2. Keneth J. Arrow. *Social Choice and Individual Values.* John Wiley & Sons, 1951. 2nd edition 1963, published by Yale University Press.
3. Keneth J. Arrow. Tullock and an existance theorem. *Public Choice*, 6:105–112, 1969.
4. Thomas Bäck. Optimal mutation rates in genetic search. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 2–8. Morgan Kaufmann, 1993.
5. Carlos A. Bana e Costa, editor. *Readings in Multiple Criteria Decision Aid.* Springer–Verlag, Berlin, 1990.
6. James O. Berger. *Statistical Decision Theory: Foundations, Concepts and Methods.* Springer–Verlag, New York, 1980.
7. Jean Charles de Borda. *Mémoire sur les Élections au Scrutin.* Mémoires de l'Académie des Sciences, 1781.

8. Denis Bouyssou, Marc Pirlot, and Philippe Vincke. A general model of preference aggregation. In M.H. Karwan, J. Spronk, and J. Wallenius, editors, *Essays in Decision Making. A Volume in Honour of Stanley Zionts*, pages 120–134. Springer, 1997. http://www.7kt.org/papers/ZIONTS.zip.
9. Jürgen Branke, Thomas Kaußler, and Hartmur Schmeck. Guidance in evolutionary multi–objective optimization. *Advances in Engineering Software*, 32(6):499–507, 2001.
10. Jean-Pierre Brans and Philippe Vincke. A Preference Ranking Organisation Method (The PROMETHEE Method for Multiple Criteria Decision–Making). *Management Science*, 31(6):647–656, 1985.
11. Christer Carlsson, editor. *Exploring the Limits of Support Systems*, TUCS General Publications No 3. Turku Centre for Computer Science, December 1996.
12. Christer Carlsson. Soft computing and decision support systems. In *6th European Congress on Intelligent Techniques and Soft Computing EUFIT '98*, Aachen, Germany, 1998. ELITE–Foundation. Invited lecture CB 3.
13. Roderick M. Chisholm and Ernest Sosa. On the logic of "intrinsically better". *American Philosophical Quarterly*, 3:244–249, 1966.
14. Carlos A. Coello Coello. A comprehensive survey of evolutionary–based multiobjective optimization techniques. *Knowledge and Information System. An International Journal*, 1(3):269–308, August 1999.
15. Carlos A. Coello Coello. Handling preferences in evolutionary multiobjective optimization: A survey. In *Congress on Evolutionary Computation (CEC'2000)*, volume 1, pages 30–37. IEEE Service Center, Piscataway, New Jersey, July 2000.
16. Carlos A. Coello Coello. An updated survey of GA–based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2):109–143, 2000.
17. Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi–Objective Problems.* Kluwer, Boston, 2002.
18. Marie Jean Antoine Nicolas de Caritat Condorcet. *Essai sur l'application de l'analyse à la probabilité des décision rendues à la pluralité des voix.* Imprimerie Royale, 1785.
19. Dragan Cvetković. *Evolutionary Multi–Objective Decision Support Systems for Conceptual Design.* PhD thesis, School of Computing, University of Plymouth, Plymouth, UK, 2000.
20. Dragan Cvetković and Ian C. Parmee. Genetic algorithm–based multi–objective optimisation and conceptual engineering design. In *Proceedings of the 1999 Congress on Evolutionary Computation – CEC99*, pages 29–36, Washington D.C., USA, July 1999. IEEE.
21. Dragan Cvetković and Ian C. Parmee. Designer's preferences and multi–objective preliminary design processes. In Ian C. Parmee, editor, *Evolutionary Design and Manufacture: Selected Papers from ACDM'00*, pages 249–260. Springer, London, 2000.
22. Dragan Cvetković and Ian C. Parmee. Agent–based Support within an Interactive Evolutionary Design System. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 16(5):331–342, November 2002.
23. Dragan Cvetković and Ian C. Parmee. Preferences and their application in evolutionary multiobjective optimisation. *IEEE Transactions on Evolutionary Computation*, 6(1):42–57, February 2002.

24. Dragan Cvetković, Ian C. Parmee, and Eric Webb. Multi–objective optimisation and preliminary airframe design. In Ian C. Parmee, editor, *Adaptive Computing in Design and Manufacture*, pages 255–267. Springer–Verlag, 1998.
25. Kalayanmoy Deb and Ram Bhusan Agrawal. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9(2):115–148, 1995.
26. Kalyanmoy Deb. Multi–objective evolutionary algorithms: Introducing bias among Pareto–optimal solutions. KanGAL report 99002, Indian Institute of Technology, Kanpur, India, 1999.
27. Kalyanmoy Deb. Multi-objective Evolutionary Algorithms: Introducing Bias Among Pareto-optimal Solutions. In Ashish Ghosh and Shigeyoshi Tsutsui, editors, *Advances in Evolutionary Computing. Theory and Applications*, pages 263–292. Springer, Berlin, 2003.
28. Kalyanmoy Deb and Hans-Georg Beyer. Self–adaptive genetic algorithms with simulated binary crossover. Technical Report CI-61/99, Department of Computer Science, University of Dortmund, Germany, March 1999.
29. Nicole Drechsler, Rolf Drechsler, and Bernd Becker. Multi–objective optimization in evolutionary algorithms using satisfiability classes. In Bernd Reusch, editor, *Computational Intelligence, Theory and Applications, International Conference, 6th Fuzzy Days, Dortmund, Germany, May 25-28, 1999, Proceedings*, volume 1625 of *Lecture Notes in Computer Science*, pages 108–117. Springer, 1999.
30. Nicole Drechsler, Rolf Drechsler, and Bernd Becker. Multi-objective optimisation based on relation *favour*. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 154–166. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
31. Francis Ysidro Edgeworth. *Mathematical Physics*. P. Keagan, London, England, 1881.
32. Günter Fandel and Tomas Gal, editors. *Proceedings of the Third Conference on Multiple Criteria Decision Making Theory and Application*, number 177 in Lecture Notes in Economics and Mathematical Systems, Hagen/Königswinter, West Germany, August 1979. Springer Verlag Berlin Heidelberg.
33. János Fodor and Marc Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.
34. Michael P. Fourman. Compaction of symbolic layout using genetic algorithms. In John J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms*, pages 141–153, 1985.
35. Ashok K. Goel. Design, analogy and creativity. *IEEE Expert, Intelligent Systems & Their Applications*, 12(3):62–70, 1997.
36. Garrison W. Greenwood, Xaobo (Sharon) Hu, and Joseph G. D'Ambrosio. Fitness functions for multiple objective optimization problems: Combining preferences with Pareto ranking. In Richard K. Belew and Michael D. Vose, editors, *Foundations of Genetic Algorithms 4 (FOGA'96)*, pages 437–455. Morgan Kaufmann, San Francisco, California, 1996.
37. Bengt Hannson. *Preference Logic: Philosophical Foundations and Applications in the Philosophy of Science*. Lund, 1970.
38. Evan J. Hughes. Multi-objective probabilistic selection evolutionary algorithm. Technical Report DAPS/EJH/56/2000, Department of Aerospace, Power, &

Sensors, Cranfield University, RMCS, Shrivenham, UK, SN6 8LA, September 2000.

39. Evan J. Hughes. Constraint handling with uncertain and noisy multi-objective evolution. In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 2, pages 963–970, Piscataway, New Jersey, May 2001. IEEE Service Center.
40. John C. Hull. *Options, Futures, and Other Derivatives.* Finance Series. Prentice Hall, Upper Saddle River, New Jersey, 5th edition, 2003.
41. Ching-Lai Hwang and Abu Syed Md. Masud. *Multiple Objective Decision Making – Methods and Applications.* Number 164 in Lecture Notes in Economics and Mathematical Systems. Springer Verlag, Berlin, 1979.
42. Nicholas R. Jennings and Michael J. Wooldridge, editors. *Agent Technology: Foundations, Applications and Markets.* Springer, 1998.
43. Ralph L. Keeney. Multidimensional utility functions: Theory, assessment and application. Technical Report TR-043, Operations Research Center, MIT, Cambridge, MA, USA, October 1969.
44. Ralph L. Keeney, Howard Raiffa, and Richard F. Meyer. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* John Wiley & Sons, 1976.
45. J. G. Lin. Maximal vectors and multi–objective optimization. *Journal of Optimization Theory and Application*, 18(1):41–64, 1976.
46. R. Duncan Luce. Semiorders and a theory of utility discrimination. *Econometrica*, 24:178–191, 1956.
47. Thierry Marchant, Denis Bouyssou, Patrice Perny, Marc Pirlot, Alexis Tsoukias, and Philippe Vincke. Choosing on the basis of several viewpoints: The example of voting. Technical Report 99/03, SMG, Université Libre de Bruxelles, Belgium, 1999. `http://smg.ulb.ac.be/Preprints/FullText/Marchant99_03.ps.gz`.
48. Richard M. Martin. *Intension and Decision. A Philosophical Study.* Prentice–Hall Inc., 1963.
49. Heinz Mühlenbein and Dirk Schlierkamp-Voosen. The science of breeding and its application to the breeder genetic algorithm (BGA). *Evolutionary Computation*, 1(4):335–360, 1993.
50. Giuseppe Munda. Multiple–criteria decision aid: Some epistemological considerations. *Journal of Multi–Criteria Decision Analysis*, 2:41–55, 1993.
51. Jan Paredis. Coevolutionary algorithms. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *The Handbook of Evolutionary Computation, 1st supplement*, pages 225–238. Institute of Physics Publishing and Oxford University Press, 1998.
52. Vilfredo Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.
53. Ian C. Parmee, Dragan Cvetković, Christopher R. Bonham, and Dominic Mitchell. Towards interactive evolutionary design systems for the satisfaction of multiple and changing objectives. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2000)*, Barcelona, Spain, September 2000.
54. Ian C. Parmee, Dragan Cvetković, Andrew H. Watson, and Christopher R. Bonham. Multi–objective satisfaction within an interactive evolutionary design environment. *Evolutionary Computation*, 8(2):197–222, 2000.
55. Ian C. Parmee and Andrew H. Watson. Preliminary airframe design using co–evolutionary multiobjective genetic algorithms. In W. Banzhaf and J. Daida

et al., editors, *GECCO–99: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1657–1665, Orlando, Florida, USA, July 1999. Morgan Kaufmann.

56. Judy Pearsall and Bill Trumble, editors. *The Oxford English Reference Dictionary.* Oxford University Press, Oxford, UK, 2nd edition, 1996.
57. Mitchell A. Potter and Kenneth A. De Jong. A cooperative coevolutionary approach to function optimization. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Proceedings from the Third Parallel Problem Solving from Nature—PPSN III*, Lecture Notes in Computer Science, Volume 866, pages 249–257, Jerusalem, Israel, October 1994. Springer-Verlag.
58. Graham Priest, Richard Routley, and Jean Norman, editors. *Paraconsistent Logic. Essays on the Inconsistent.* Philosophia Verlag, München, Germany, 1989.
59. Howard Raiffa. Preferences for multi–attributed alternatives. Technical Report RM-5868-DOT/RC, RAND Corporation, Santa Monica, CA, USA, 1969.
60. Nicholas Recher. Applied logic. In *Encyclopædia Britannica, Macropædia, Vol. 11*, pages 28–38. Encyclopædia Britannica Inc, 15th edition, 1981.
61. Christopher D. Rosin and Richard K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1996.
62. Bernard Roy. Decision–aid and decision–making. In Carlos A. Bana e Costa, editor, *Readings in Multiple Criteria Decision Aid*, pages 17–35. Springer–Verlag, Berlin, 1990.
63. Bernard Roy and Denis Bouyssou. *Aide Multicritère à la Décision: Méthodes et Cas.* Economica, Paris, France, 1993.
64. Tuomas W. Sandholm. Distributed rational decision making. In Gerhard Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. MIT press, 1999. `http://www-2.cs.cmu.edu/~sandholm/rational.ps`.
65. J. David Schaffer. *Some Experiments in Machine Learning using Vector Evaluated Genetic Algorithm.* PhD thesis, Deparment of Electrical Engineering, Vanderbilt University, Nashville, 1984. TCGA file No. 00314.
66. J. David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In John J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrance Erlbaum Associates, 1985.
67. Frank Schmiedle, Nicole Drechsler, Daniel Große, and Rolf Drechsler. Priorities in multi–objective optimization for genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 129–136. Morgan Kaufmann, 2001.
68. Hans-Paul Schwefel. *Evolution and Optimum Seeking.* John Wiley & Sons, New York, 1995.
69. Michael J. Scott and Erik K. Antonsson. Arrow's theorem and engineering design decision making. *Research in Engineering Design*, 11(4):218–228, 1999.
70. Pratyush Sen and Jian-Bo Yang. *Multiple Criteria Decision Support in Engineering Design.* Springer–Verlag, London, 1998.
71. Weiming Shen, Douglas H. Norrie, and Jean-Paul A. Barthès. *Multi–Agent Systems for Concurrent Intelligent Design and Manufacturing.* Taylor & Francis, London, UK, 2000.
72. Wolfram Stadler. Fundamentals of multicriteria optimization. In W. Stadler, editor, *Multicriteria Optimization in Engineering and the Sciences*, pages 1–25. Plenum Press, New York, NY, USA, 1988.

73. K. C. Tan, T. H. Lee, and E. F. Khor. Evolutionary algorithms with goal and priority information for multi-objective optimization. In *1999 Congress on Evolutionary Computation*, pages 106–113, Washington, D.C., July 1999. IEEE Service Center.
74. Gordon Tullock. *Towards a Mathematics of Politics*. University of Michigan Press, 1967.
75. Amos Tversky. Intransitivity of preferences. *Psychological Review*, 76(1):31–48, January 1969.
76. Philippe Vincke. *Multicriteria Decision–Aid*. John Wiley & Sons, Chichester, England, 1992.
77. Philippe Vincke. $\{P, Q, I, J\}$ – preference structures. Technical Report 2000/05, SMG, Université Libre de Bruxelles, Belgium, 2000. `http://smg.ulb.ac.be/Preprints/FullText/Vincke00_05.ps.gz`.
78. Philippe Vincke. Preferences and numbers. In A. Colorni, M. Paruccini, and B. Roy, editors, *A-MCD-A - Aide Multi Critère à la Décision - Multiple Criteria Decision Aiding*, pages 343–354. Joint Research Center, The European Commission, 2001.
79. John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press, Princeton, New Jersey, USA, 2nd edition, 1947.
80. Georg Henrik von Wright. The logic of preference. An essay. Edinburgh, 1963.
81. Georg Henrik von Wright. The logic of preference reconsidered. *Theory and Decision*, 3:140–167, 1972.
82. Eric Webb. MINICAPS – a simplified version of CAPS for use as a research tool. Unclassified Report BAe-WOA-RP-GEN-11313, British Aerospace, Warton, UK, July 1997.