Part II – Decision Systems for Engineering Design
Lecture 5: Interactive Decision Support

Robin Purshouse
University of Sheffield

Spring Semester 2023/24

1 Lecture overview

This lecture looks at **interactive**, also known as **progressive**, decision systems for engineering design. In these approaches, decision-maker preferences are elicited and exploited either before or during the optimization process.

In this lecture, we will cover:

- Interactive Pareto-based decision systems—focusing in detail on a useful replacement for the dominance relation that is able to handle decision-makers who have priorities and goals for their performance criteria;
- Interactive decomposition-based and set-based decision systems—a very brief introduction to the incorporation of preferences into these two classes of optimizer.

2 Preferability

2.1 Definition of the preferability operator

Consider the situation where a decision-maker has *goals* for particular levels of performance for each objective, and is also able to *prioritise* which goals are the most important. For example, a Chief Engineer who is responsible for a new vehicle may have goals and priorities as shown in Table 1.

Consider now a *preference vector*, $\mathbf{g}$:

$$\begin{aligned}\mathbf{g} &= [\mathbf{g}_1, \ldots, \mathbf{g}_\rho] \\ &= [(g_{1,1}, \ldots, g_{1,M_1}), \ldots, (g_{\rho,1}, \ldots, g_{\rho,M_\rho})]\end{aligned} \tag{1}$$

Objective	Goal	Priority
3rd gear time-to-stabilised-acceleration	9.0 s	Low
CO_2 efficiency on the WLTP drive cycle	110 g/km	Low
NO_x emissions on the WLTP drive cycle	0.08 g/km	High
PM emissions on the WLTP drive cycle	0.005 g/km	High

Table 1: Illustrative goals and priorities for a four-objective engineering design problem

where ρ is a positive integer reflecting the highest level of priority and $\sum_{i=1}^{\rho} M_i = M$ (where M is the number of objectives). The preference vector essentially splits up the objective vector into groups of objectives of equal priority (assuming, without loss of generality, some convenient permutation of the original ordering of the objectives). Here $\mathbf{g}_i$ refers to the ith group of such objectives, and $g_{i,j}$ is the goal (or target) relating to the jth objective in that group.

Now let $\underset{\frown}{\mathbf{z}}$ be the objectives for which a particular solution's performance, $\mathbf{z}$, meets the associated goals $\mathbf{g}$, $\underset{\frown}{\mathbf{z}}$ be the objectives for which $\mathbf{z}$ does not meet the goals, and $\mathbf{z}_i$ indicate performance against the objectives defined at the ith priority level. We can now define the so-called **preferability operator**, $\preccurlyeq_{\mathbf{g}}$:

$$\mathbf{u} \underset{\mathbf{g}_{1,\ldots,\rho}}{\prec} \mathbf{v} \iff \begin{cases} (\mathbf{u}_{\rho}^{\underset{\frown}{\mathbf{u}}} \prec \mathbf{v}_{\rho}^{\underset{\frown}{\mathbf{u}}}) \vee \{(\mathbf{u}_{\rho}^{\underset{\frown}{\mathbf{u}}} = \mathbf{v}_{\rho}^{\underset{\frown}{\mathbf{u}}}) \wedge [(\mathbf{v}_{\rho}^{\underset{\frown}{\mathbf{u}}} \npreccurlyeq \mathbf{g}_{\rho}^{\underset{\frown}{\mathbf{u}}}) \vee (\mathbf{u}_{\rho}^{\underset{\frown}{\mathbf{u}}} \prec \mathbf{v}_{\rho}^{\underset{\frown}{\mathbf{u}}})]\} & \text{if } \rho = 1 \\ (\mathbf{u}_{\rho}^{\underset{\frown}{\mathbf{u}}} \prec \mathbf{v}_{\rho}^{\underset{\frown}{\mathbf{u}}}) \vee \{(\mathbf{u}_{\rho}^{\underset{\frown}{\mathbf{u}}} = \mathbf{v}_{\rho}^{\underset{\frown}{\mathbf{u}}}) \wedge [(\mathbf{v}_{\rho}^{\underset{\frown}{\mathbf{u}}} \npreccurlyeq \mathbf{g}_{\rho}^{\underset{\frown}{\mathbf{u}}}) \vee (\mathbf{u} \underset{\mathbf{g}_{1,\ldots,\rho-1}}{\prec} \mathbf{v})]\} & \text{otherwise.} \end{cases} \tag{2}$$

where $\mathbf{u}$ and $\mathbf{v}$ are two vectors in objective space (as generated by evaluations of two candidate designs), and $\prec$ is the dominance relation introduced in Equation 3 of Lecture 3. For compactness, we can also write $\mathbf{u} \underset{\mathbf{g}_{1,\ldots,\rho}}{\prec} \mathbf{v}$ as simply $\mathbf{u} \preccurlyeq_{\mathbf{g}} \mathbf{v}$.

Note that the preferability relation is transitive, i.e. given three objective vectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$, and a preference vector $\mathbf{g}$:

$$\mathbf{u} \preccurlyeq_{\mathbf{g}} \mathbf{v} \preccurlyeq_{\mathbf{g}} \mathbf{w} \implies \mathbf{u} \preccurlyeq_{\mathbf{g}} \mathbf{w} \tag{3}$$

2.2 Explanation

The preferability operator in Equation 2 looks quite daunting, but is actually reasonably straightforward to interpret. Consider first the case where there are only objectives of the lowest priority level, i.e. $\rho = 1$. Objective vector $\mathbf{u}$ is preferable to $\mathbf{v}$ if and only if:

- For objectives where $\mathbf{u}$ does not meet the respective goals, $\mathbf{u}$ dominates $\mathbf{v}$;
- Or, if for those objectives, $\mathbf{u}$ and $\mathbf{v}$ are equal in performance, then $\mathbf{u}$ is preferred if $\mathbf{v}$ does not meet all the goals for the remaining objectives (i.e. those that $\mathbf{u}$ is capable of meeting), or if $\mathbf{u}$ dominates $\mathbf{v}$ over these remaining objectives.

Note that in the case where $\mathbf{u}$ meets all the goals, $\mathbf{u}_{\rho}^{\underset{\frown}{\mathbf{u}}}$ and $\mathbf{v}_{\rho}^{\underset{\frown}{\mathbf{u}}}$ will both be null sets and the comparison will then be based solely on whether $\mathbf{v}$ fails to meet the goals or is otherwise dominated by $\mathbf{u}$.

Now consider the case where $\rho = 2$. The operator starts by considering the objectives and associated goals at this higher priority level. Objective vector $\mathbf{u}$ is preferable to $\mathbf{v}$ if and only if:

- For objectives of priority level 2, where $\mathbf{u}$ does not meet the goals, $\mathbf{u}$ dominates $\mathbf{v}$;
- Or, if for those objectives, $\mathbf{u}$ and $\mathbf{v}$ are equal in performance, then $\mathbf{u}$ is preferred if $\mathbf{v}$ does not meet all the goals for the remaining objectives at priority 2. In the situation where $\mathbf{v}$ *does* meet these goals, rather than making a decision purely at this higher priority level based on dominance, a new preferability comparison is undertaken for the lower priority objectives, if they exist.

2.3 Special cases of preferability

Preferability is a very flexible operator which encapsulates many other popular schemes for interactive optimization as special cases. These schemes are briefly introduced below.

2.3.1 Lexicographic ordering

Formulation:

$$\mathbf{g} = [g_1, \ldots, g_M] = [-\infty, \ldots, -\infty] \tag{4}$$

In the *lexicographic* approach, the decision-maker indicates an order in which they would like the performance criteria to be optimized (i.e. a full prioritisation of objectives). This approach is captured by a preferability operator with a different priority level for each objective and goals that are impossible to achieve (i.e. $g_i = -\infty \ \forall i$). In this situation, the operator will prefer solutions offering better performance in the highest objective only. For solutions that offer equal performance according to this objective, the operator will then prefer the solution offering better performance at the next priority level down, and so on, until performance is based on only the lowest-priority objective.

2.3.2 Constrained optimization

Formulation:

$$\mathbf{g} = [\mathbf{g}_1, \mathbf{g}_2] = [(-\infty, \ldots, -\infty), (g_{2,1}, \ldots, g_{2,M_c})] \tag{5}$$

where M_c is the number of constraints in the problem formulation.

In the *constrained optimization* approach, the decision-maker has a set of criteria for which a certain level of performance must be achieved (but where over-achievement is not rewarded)—i.e. the conventional definition of hard constraints. The decision-maker also has other criteria which must be optimized but for which there is no specific goal—i.e. the conventional definition of objectives. This type of preference

information can be incorporated into a preferability operator with two priority levels (1 and 2). Finite goals are provided only for the criteria at level 2. The operator will prefer solutions that can meet the goals on the constraints compared to solutions which cannot; however when both solutions meet the goals, the operator recursively calls itself at level 1 and makes a comparison based purely on Pareto dominance on the level 1 objectives.

2.3.3 Constraint satisfaction

Formulation:

$$\mathbf{g} = [\mathbf{g}_2] = [g_{2,1}, \ldots, g_{2,M}] \tag{6}$$

In many cases, decision-makers are equally happy with any solutions that can meet their goals—this is so-called *satisficing* behaviour, and is handled using a *constraint satisfaction* approach. To implement constraint satisfaction with the preferability operator, only objectives at priority level 2 are specified. The implication is that the Pareto dominance comparison at priority level 1 is never encountered; rather, the comparison is based purely on violation (or not) of the goals at level 2.

2.3.4 Goal programming

Formulation:

$$\mathbf{g} = [\mathbf{g}_1] = [g_{1,1}, \ldots, g_{1,M}] \tag{7}$$

In the *goal programming* approach it is assumed that, whilst the decision-maker is principally interested in satisfying the goals, they are also interested in achieving further improvements where possible. The preferability operator achieves this effect by specifying goals at priority level 1. In this case, the comparison will be based on dominance if both solutions satisfy the goals for all objectives.

2.3.5 Pareto dominance

Formulation:

$$\mathbf{g} = [\mathbf{g}_1] = [-\infty, \ldots, -\infty] \tag{8}$$

Pareto dominance is, of course, used in *a posteriori* methods rather than interactive methods. However it is also a particular case of preferability, where all objectives have equal priority and goals are all set to $-\infty$ (i.e. are unachievable). In this case, the goals are never met and comparisons are based on dominance.

2.4 Example

To illustrate the workings of the preferability operator, we consider how it can be used to compare three separate vehicle designs for the vehicle design problem introduced in Table 1. The designs are given in Table 2.

Objective	Priority	Goal	Design $\mathbf{a}$	Design $\mathbf{b}$	Design $\mathbf{c}$
3rd gear accel.	1	$g_{1,1}$ = 9.0 s	9.5	8.5	9.5
WLTP CO_2	1	$g_{1,2}$ = 110 g/km	115	105	116
WLTP NO_x	2	$g_{2,1}$ = 0.08 g/km	0.08	0.09	0.07
WLTP PM	2	$g_{2,2}$ = 0.005 g/km	0.004	0.004	0.003

Table 2: Three alternative solutions to the vehicle design problem

2.4.1 Comparing design $\mathbf{a}$ and design $\mathbf{b}$

Consider the situation from the perspective of design $\mathbf{a}$ (so $\mathbf{u} = \mathbf{z}(\mathbf{a})$ and $\mathbf{v} = \mathbf{z}(\mathbf{b})$ in Equation 2). At priority level 2, design $\mathbf{a}$ meets both goals, so $\mathbf{u}_2^{\widehat{\mathbf{u}}}$ is the null set. The comparison is now based on whether design $\mathbf{b}$ can meet the goals that $\mathbf{a}$ meets at level 2; and since $\mathbf{b}$ fails to achieve the NO_x target ($0.09 \nleqslant 0.08$), we can say that $\mathbf{a} \preccurlyeq_{\mathbf{g}} \mathbf{b}$.

2.4.2 Comparing design $\mathbf{a}$ and design $\mathbf{c}$

As in the previous case, the comparison is first based on whether design $\mathbf{c}$ can meet the level 2 priorities that design $\mathbf{a}$ satisfies. We see that it can (and, in fact, it outperforms $\mathbf{a}$). The comparison now falls to the situation at priority level 1. Here, design $\mathbf{a}$ fails to meet the goals for both objectives. However, $\mathbf{a}$ also dominates $\mathbf{c}$ on these objectives, since it achieves equal performance for third gear acceleration and better performance in CO_2; so we can also now say that $\mathbf{a} \preccurlyeq_{\mathbf{g}} \mathbf{c}$, despite $\mathbf{c}$ performing better on the higher priority objectives. If we wanted to make these objectives also the subject of dominance comparisons then they would need to be replicated at priority level 1.

2.4.3 Comparing design $\mathbf{b}$ and design $\mathbf{c}$

Looking at the priority 2 objectives, design $\mathbf{b}$ can meet the PM goal but not the NO_x goal, so we focus comparisons first on the latter. For this objective, $\mathbf{b}$ does not dominate $\mathbf{c}$—in fact it performs worse, so we now know that $\mathbf{b}$ is not preferred to $\mathbf{c}$.

To check whether $\mathbf{c}$ is preferred to $\mathbf{b}$, we first note that $\mathbf{c}$ satisfies both of the priority 2 goals. Since $\mathbf{b}$ cannot satisfy one of these goals, this is sufficient to establish that $\mathbf{c}$ is preferred to $\mathbf{b}$, i.e. $\mathbf{c} \prec_{\mathbf{g}} \mathbf{b}$.

2.4.4 Outcome of pair-wise comparisons

By comparing all three designs on a pair-wise basis, we have identified that design $\mathbf{a}$ is the most preferred design, with design $\mathbf{b}$ being the least preferred design, i.e. $\mathbf{a} \preccurlyeq_{\mathbf{g}} \mathbf{c} \preccurlyeq_{\mathbf{g}} \mathbf{b}$.

2.5 Interactive use of preferability

The preferability operator can be used as a direct replacement for the dominance relation in any Pareto-based optimizer. For example, in NSGA-II (covered in Lecture 3), the non-dominated sorting procedure can be modified to establish ranks on the basis of preferability rather than dominance.

If goals and priorities can be elicited from the decision-maker prior to performing optimization, and remain fixed during optimization, then the use of preferability will lead to an approximation set that acknowledges these preferences. Note that this approximation set is still likely to remain a family of solutions rather than a single solution—so *a posteriori* decision-making approaches can still be applied.

However, it is more often the case that decision-makers cannot fully specify a set of goals and priorities prior to beginning the optimization process; further, the preferences of the decision-maker may be *influenced* by the nature of the trade-offs that exist in the problem. For these reasons, preference-based approaches like preferability are more likely to be used in an interactive way, where the goals and priorities are updated periodically as the search progresses.

In an interactive approach, the general idea is to begin with a fairly weak set of preferences (e.g. by specifying priority level 1 goals that are perhaps considered easy to satisfy, or by not specifying any goals at all and just making comparisons on the basis of dominance). Then, after an initial set of solutions have been presented to the decision-maker (e.g. the results of a space-filling design, or an initial iteration of a population-based optimizer), the DM is invited to tighten or relax the current set of goals. The optimizer is then allowed to progress for a few further iterations, and the interaction stage of the process is repeated. An example of such an iterative process using the preferability operator is shown in Figure 1.

3 Other approaches to incorporating preferences

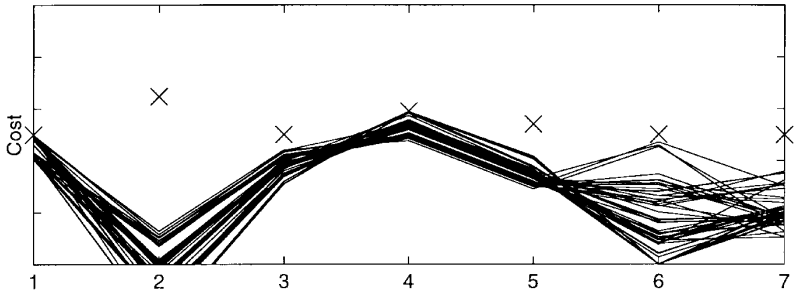
3.1 Decomposition-based decision systems

Decomposition-based algorithms are implicitly preference-based, in that each of the sub-problems focuses on a different direction in performance space. These sub-problems can be designed to relate to directions of particular interest to the DM (for example, a direction that passes through a goal vector), and these directions can potentially be modified as the search progresses.

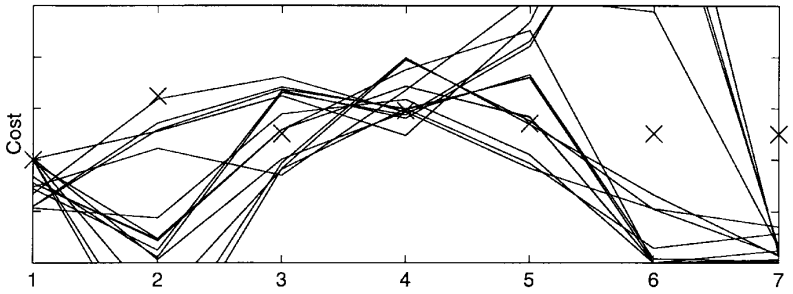
When working with normalised objectives, care is needed to ensure that the scalarising function is configured with a weight vector that is still 'pointing' in the direction of the goal vector in the normalised performance space.

3.2 Set-based decision systems

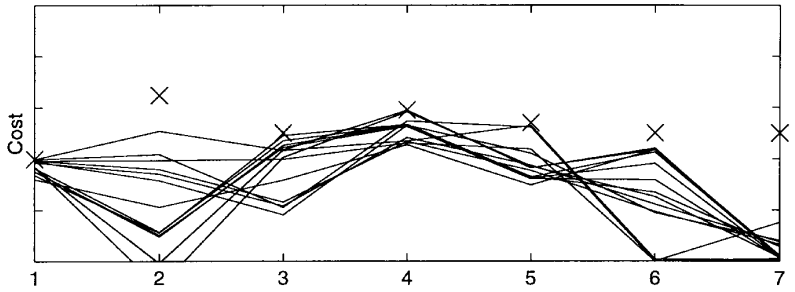
Recall from the previous lecture that the hypervolume is the volume of performance space that is dominated by an approximation set, when bounded by a reference point. The basic idea of incorporating preferences into hypervolume-based schemes is to



(a) Interim findings after 40 generations of the optimizer



(b) Effect of tightening the goal for objective 1



(c) Results after a further 20 iterations of the optimizer

Figure 1: Interactive decision system for a gas turbine engine controller gain tuning problem: parallel coordinates plots for seven objectives, with goals shown as crosses. Adapted from Fonseca & Fleming's *Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms – Part II: Application Example*, 1998

.

weight the volume calculation based on the relative importance of different parts of the space. Such a weighting can be achieved by defining a monotonically non-decreasing *value function* or *desirability function* for each objective, or combinations of objectives, that captures the importance to the DM of certain levels of performance against those objectives.

Incorporating preferences into hypervolume-based optimizers is a relatively recent innovation where further research is still needed. The following are important considerations:

- Since set-based schemes work on a hypervolume-contribution basis, the value function will need to be superlinear in regions of interest in order to focus in detail on designs in these areas. Typically, exponential value functions are used to promote enhanced representation of important areas.
- Calculating hypervolume is computationally expensive and the weighted hypervolume is even more so, although efficient numerical integration approaches have been proposed.

4 Further reading

- Fonseca, C.M., Fleming, P.J. Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part I: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* 1998;28:26–37.
- Xin, B., Chen, L., Chen, J., Ishibuchi, H, Hirota, K., Liu, B. Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access* 2018;6:41256–41279.