MULTICRITERIA DECISION AIDS

Basic Concepts of Multicriteria Decision-Making

- We have a set of actions, A, to be explored during the decision procedure.
- Example:
  - Which combination of development methods is most appropriate to develop a safe system in the given environment?
  - A consists of all combinations of mutually compatible methods selected from some original set
  - An enumeration of A might consist of combinations such as
    - (Alloy specification, correctness proofs)
    - (Alloy specification, UML design)
    - (OCL specification, Z formal verification, proof)
    - (OOD)
    - (Specification from Python rapid prototyping, agile development)

Basic Concepts of Multicriteria Decision-Making

- Once we have a set of actions, we define a criterion, g, to be a function from the set of actions to a totally ordered set, T
- T satisfies four properties:
  - R is reflexive: for each element x in T, the pair (x, x) is in R.
  - R is transitive: if (x, y) and (y, z) are in R, then (x, z) must be in R.
  - R is antisymmetric: if (x, y) and (y, x) are in R, then x must equal y.
  - R is strongly complete: for any x and y in T, either (x, y) is in R or (y, x) is in R.

Basic Concepts of Multicriteria Decision-Making

Suppose A is a set of verification and validation techniques used on a software project. We can define a criterion that maps A to the set of real numbers by defining g as follows: For each element a of A,

$$g(a) = \text{the total effort in person-months devoted to using technique } a$$

This measure gives us information about experience with each element of A. Alternatively, we can define another criterion, g', to be a mapping to the set of nonnegative integers, so that we can compare the effectiveness of two techniques:

$$g'(a) = \text{the total number of faults discovered when using technique } a$$

We can also have criteria that map to sets that are not numbers. For example, define the set T to be {poor, moderate, good, excellent}, representing categories of ease of use. Then we can define a criterion that maps each element of A to an element of T that represents the ease of use for a particular technique, as rated subjectively by an expert. In this example, T is totally ordered, since

$$\text{poor} < \text{moderate} < \text{good} < \text{excellent}$$

Basic Concepts of Multicriteria Decision-Making

- A decision-maker must compare each pair actions, a and b, in one of the three ways:
  - Strict preference for one of the actions
  - Indifference to either action
  - Refusal or inability to compare the actions
- We can also define a preference relation S using only conditions 1 and 2, where a is related to b if and only if either a is strictly preferred to b, or there is indifference between the two
- Define a function, f, from the set A to some number system, N satisfying -
  - $f(a) > f(b)$ if and only if a is strictly preferred to b
  - $f(a) = f(b)$ if and only if there is indifference between a and b
- If there are no ties, then the preference structure is a complete order.

Dominance

- Suppose a and b are possible actions, and we have a set of n criteria $\{g_i\}$.
- We say that a dominates b if $g_i(a) \geq g_i(b)$ for each i from 1 to n.
- An action is said to be efficient if it is strictly dominated by no other action

Dominance

We are considering eight software packages, P1 through P8, to select one for use in a safety-critical application. All of the packages have the same functionality. There are four criteria that govern the selection: cost (measured in dollars), speed (measured in number of calculations per minute on a standardized set of test data), accuracy, and ease of use. The latter two criteria are measured on an ordinal scale of integers from 0 to 3, where 0 represents *poor*, 1 represents *fair*, 2 represents *good*, and 3 represents *very good*. Table 6.8 contains the results of the ratings for each criterion. To ensure that the dominance relation holds, we have changed the sign of the values for cost. In this example, we see that

P2 dominates P1
P5 dominates both P4 and P6
P3, P7, and P8 dominate no other package
P2, P5, P3, P7, and P8 are efficient

Example

Software Package	g_1: Cost	g_2: Speed	g_3: Accuracy	g_4: Ease of Use
P1	–1300	3000	3	1
P2	–1200	3000	3	2
P3	–1150	3000	2	2
P4	–1000	2000	2	0
P5	–950	2000	2	1
P6	–950	1000	2	0
P7	–900	2000	1	0
P8	–900	1000	1	1

Example

- The ideal point is the point (z1 , z2 , ..., zn), where zi = gi(ai*) . For instance, the ideal point in this example is (−900, 3000, 3, 2)
- Payoff Matrix

$$M = \begin{bmatrix} a_1^* = P8 & -900 & 1000 & 1 & 1 \\ a_2^* = P2 & -1200 & 3000 & 3 & 2 \\ a_3^* = P2 & -1200 & 3000 & 3 & 2 \\ a_4^* = P3 & -1150 & 3000 & 2 & 2 \end{bmatrix}$$

$$M' = \begin{bmatrix} a_1^* = P7 & -900 & 2000 & 1 & 0 \\ a_2^* = P3 & -1150 & 3000 & 2 & 2 \\ a_3^* = P1 & -1300 & 3000 & 3 & 1 \\ a_4^* = P2 & -1200 & 3000 & 3 & 2 \end{bmatrix}$$

Example

- The nadir is the point whose ith coordinate is the minimum of the values in the ith column
- The nadir for matrix M is (−1200, 1000, 1, 1); for M′, the nadir is (−1300, 2000, 1, 0)
- Choose efficient action by maximizing distance from the nadir point and minimizing distance from the ideal point

Multiattribute utility theory (MAUT)

- Takes into consideration the subjectivity issue

Software Package	g_1: Cost	g_2: Speed	g_3: Accuracy	g_4: Ease of Use
P1	–1300	3000	3	1
P2	–1200	3000	3	2
P3	–1150	3000	2	2
P4	–1000	2000	2	0
P5	–950	2000	2	1
P6	–950	1000	2	0
P7	–900	2000	1	0
P8	–900	1000	1	1

Multiattribute utility theory (MAUT)

Criterion	Measurement Scale
g_1: Effort required	{Little, moderate, considerable, excessive}
g_2: Coverage	{Bad, reasonable, good, excellent}
g_3: Tool support	{No, yes}
g_4: Ranking of usefulness by expert	$\{1, 2, \ldots, n\}$ where n is number of techniques

Multiattribute utility theory (MAUT)

A decision-maker rates four techniques, and the results are shown in Table 6.10. We have several choices for defining a utility function, U, which is a sum of functions U_i on each g_i. One possibility is to define each U_i to be a transformation onto the unit interval [0, 1]. For instance, we may define:

$$U_1(\text{little}) = 0.8 \quad U_1(\text{moderate}) = 0.5 \quad U_1(\text{considerable}) = 0.2 \quad U_1(\text{excessive}) = 0$$

$$U_2(\text{bad}) = 0 \quad U_2(\text{reasonable}) = 0.1 \quad U_2(\text{good}) = 0.3 \quad U_2(\text{excellent}) = 0.6$$

$$U_3(\text{no}) = 0.2 \quad U_3(\text{yes}) = 0.7$$

$$U_4(x) = 1/x$$

Multiattribute utility theory (MAUT)

$$U(\text{inspections}) = 0.2 + 0.6 + 0.2 + 1 = 2$$
$$U(\text{proof}) = 0 + 0.3 + 0.2 + 0.5 = 1$$
$$U(\text{static analysis}) = 0.5 + 0 + 0.7 + 0.25 = 1.45$$
$$U(\text{black box}) = 0.8 + 0.3 + 0.2 + 0.33$$

Outranking Methods (Electre I)

Action	g_1: Effort Required (Weight = 5)	g_2: Potential for Detecting Critical Faults (Weight = 4)	g_3: Coverage Achieved (Weight = 3)	g_4: Tool Support (Weight = 3)
1	Excessive	Excellent	Good	Yes
2	Considerable	Excellent	Average	Yes
3	Considerable	Good	Good	Yes
4	Moderate	Good	Good	No
5	Moderate	Good	Average	Yes
6	Moderate	Reasonable	Good	Yes
7	Little	Reasonable	Average	No

Outranking Methods (Electre I)

Action	g_1: Effort Required (Weight = 5)	g_2: Potential for Detecting Critical Faults (Weight = 4)	g_3: Coverage Achieved (Weight = 3)	g_4: Tool Support (Weight = 3)
1	Excessive	Excellent	Good	Yes
2	Considerable	Excellent	Average	Yes
3	Considerable	Good	Good	Yes
4	Moderate	Good	Good	No
5	Moderate	Good	Average	Yes
6	Moderate	Reasonable	Good	Yes
7	Little	Reasonable	Average	No

Concordance Indices

	1	2	3	4	5	6	7
1	–	10	10	10	10	10	10
2	12	–	12	7	10	7	10
3	11	11	–	11	10	10	10
4	8	8	12	–	12	12	10
5	8	11	12	12	–	12	10
6	11	11	11	11	11	–	10
7	5	8	5	8	8	9	—

Outranking Methods (Electre I)

Concordance Indices

	1	2	3	4	5	6	7
1	–	10	10	10	10	10	10
2	12	–	12	7	10	7	10
3	11	11	–	11	10	10	10
4	8	8	12	–	12	12	10
5	8	11	12	12	–	12	10
6	11	11	11	11	11	–	10
7	5	8	5	8	8	9	—

Next, we restrict the structure by defining a *concordance threshold*, t, so that a is preferred to b only if the concordance index for (a, b) is at least as large as t. Suppose t is 12. Then, according to Table 6.12, action 2 is still preferable to action 1, but we no longer have preference of action 1 to action 7 because the concordance index for (1, 7) is not large enough.

Outranking Methods (Electre I)

Concordance Indices

	1	2	3	4	5	6	7
1	–	10	10	10	10	10	10
2	12	–	12	7	10	7	10
3	11	11	–	11	10	10	10
4	8	8	12	–	12	12	10
5	8	11	12	12	–	12	10
6	11	11	11	11	11	–	10
7	5	8	5	8	8	9	—

We further refine the preference structure to include other constraints. For instance, suppose that for criterion g_1 (effort required) we never allow action a to outrank action b if $g_1(a)$ is excessive and $g_1(b)$ is little. In other words, regardless of the values of the other criteria, b is so superior to a with respect to g_1 that we veto it being outranked by a. In general, we handle this situation by defining a *discordance set* for each criterion, containing the ordered pairs of values for the criterion for which the outranking is refused.

For example, let us define

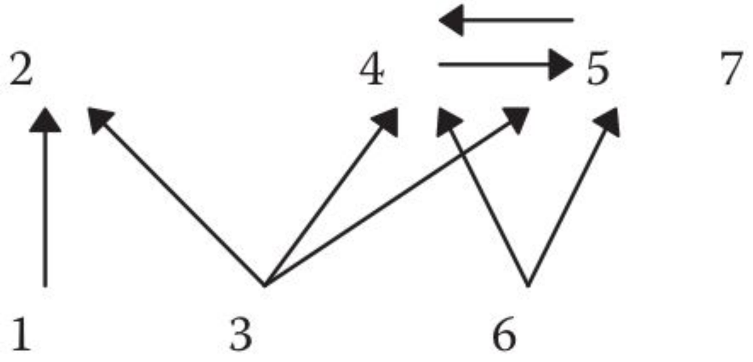
$$D_1 = \{(\text{excessive, little}), (\text{considerable, little})\}$$
$$D_2 = \{(\text{moderate, excellent})\}$$
$$D_3 = D_4 = \{\}$$

Outranking Methods (Electre I)

Concordance Indices

	1	2	3	4	5	6	7
1	–	10	10	10	10	10	10
2	12	–	12	7	10	7	10
3	11	11	–	11	10	10	10
4	8	8	12	–	12	12	10
5	8	11	12	12	–	12	10
6	11	11	11	11	11	–	10
7	5	8	5	8	8	9	—

Graph of outranking relation



Optimal Subsets of Actions/ Kernels?