Risk and Multi-Attribute Utilities

When a problem involves multiple attributes and risk, we need to use ***multi-attribute utility functions***, which resemble the multi-attribute value functions described above for cases when risk and uncertainty are not present.

We shall give an example of a well-known approach due to Keeney and Raiffa.

For the Keeney and Raiffa method to be relatively straightforward we require that all the attributes be mutually utility independent, a stipulation that is stronger than mutual preference independence required when using the additive model for value functions.

If the certainty equivalent level of X for an outcome involving 50:50 chances of the worst and best values of attribute X is assessed for any fixed level of attribute Y, and is found to be the same, regardless of how Y is varied throughout the domain of Y, then X is ***utility independent*** of Y.

If Y is also utility independent of X, then the pair are said to be ***mutually utility independent***.

Given mutual utility independence between attributes, one can employ the ***multiplicative model*** to combine utility values of the individual attributes.

In the case of just two mutually utility independent attributes X and Y, this amounts to:

$$U(x, y) = k_1 * U_1(x) + k_2 * U_2(y) + (1 - k_1 - k_2) * U_1(x) * U_2(y)$$

where k_1 and k_2 are weights.

Example: Decanal Engineering Corporation Problem

The Decanal Engineering Corporation has recently signed a contract to carry out a major overhaul of a company's equipment.

Ideally, the customer would like the overhaul to be completed in 12 weeks and, if Decanal meet the target or do not exceed it by a significant amount of time, it is likely to gain substantial goodwill from the customer and an enhanced reputation throughout the industry.

However, to increase the chances of meeting the target, Decanal would have to hire extra labour and operate some 24-hour working, which would increase its costs.

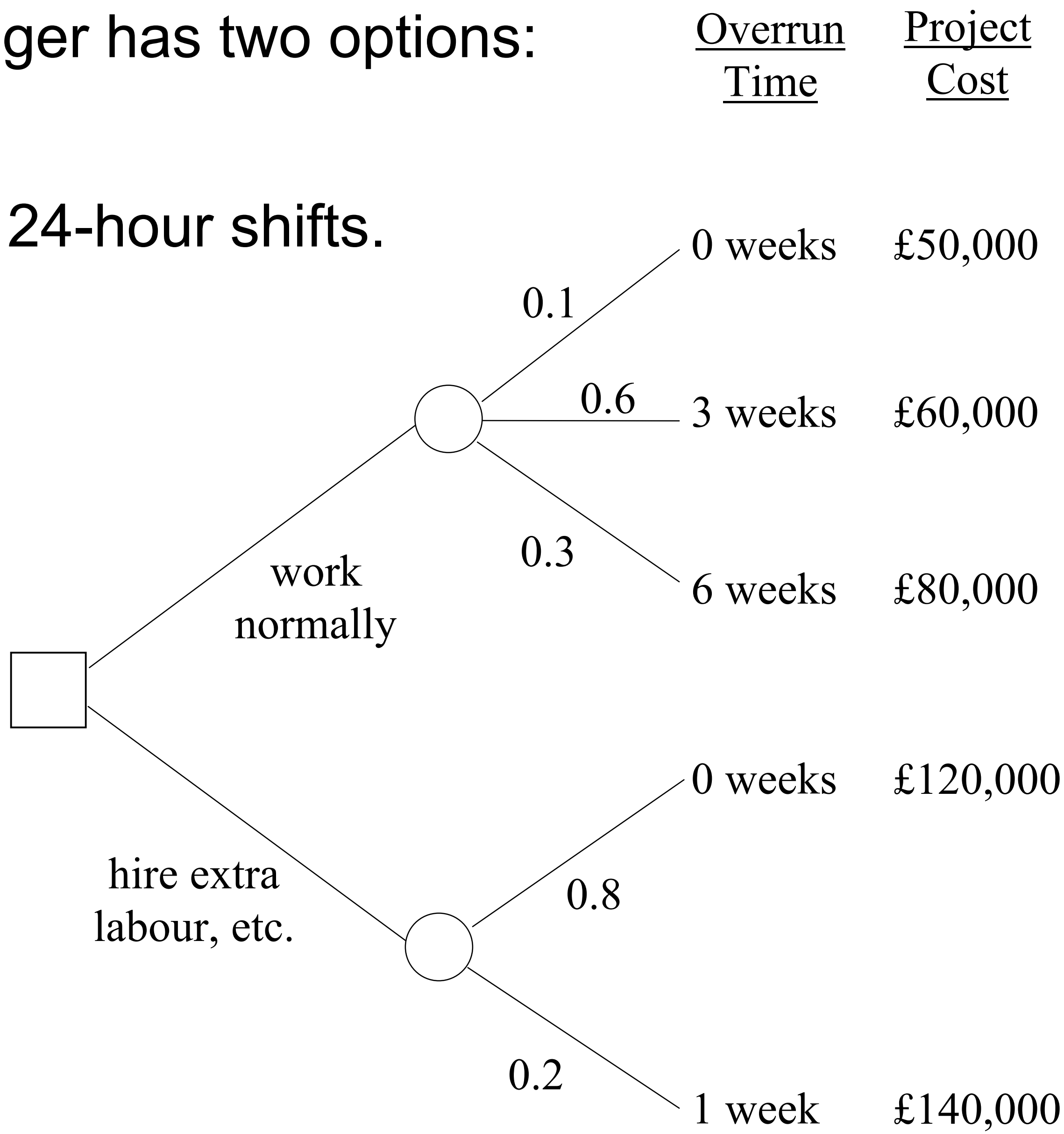
Thus the company has two conflicting objectives:

(A) minimise the time that the project overruns the target date and
(B) minimise the cost of the project.

Assume that the project manager has two options:

(1) work normally, or
(2) hire extra labour and work 24-hour shifts.

Note that once a given option is chosen, the longer the project takes to complete, the greater will be the costs, because labour, equipment, etc., will be employed on the project for a longer period.)



Deriving a Multi-Attribute Utility Function.

We shall assume that overrun time and project cost are mutually utility independent and will derive a multi-attribute utility function with which the project manager can compare the two options.

Step 1: Deriving single-attribute utility functions for overrun time and project cost.

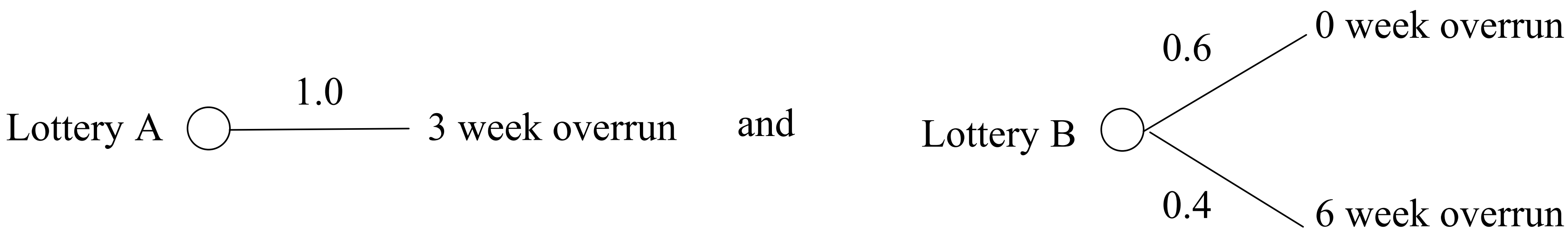
We derive a utility function for project overrun in much the same way as a utility function for money might be elicited.

Of the different outcomes, the best overrun is 0 weeks, which is given a utility of 1.0, and the worst overrun is 6 weeks, which is given a utility of 0.

In order to find some intermediate values on the utility curve we ask the decision maker to consider several lotteries and to come up with certainty equivalents.

These lotteries would either all have the same prizes: 0 weeks and 6 weeks, and different probabilities, or all have the same 50:50 probabilities, but different prizes (using the bisection method).

Adopting the former method, the decision maker indicates that he is indifferent between :



This implies that: U(3-weeks overrun) = 0.6 .

By a similar process, the manager indicates that:

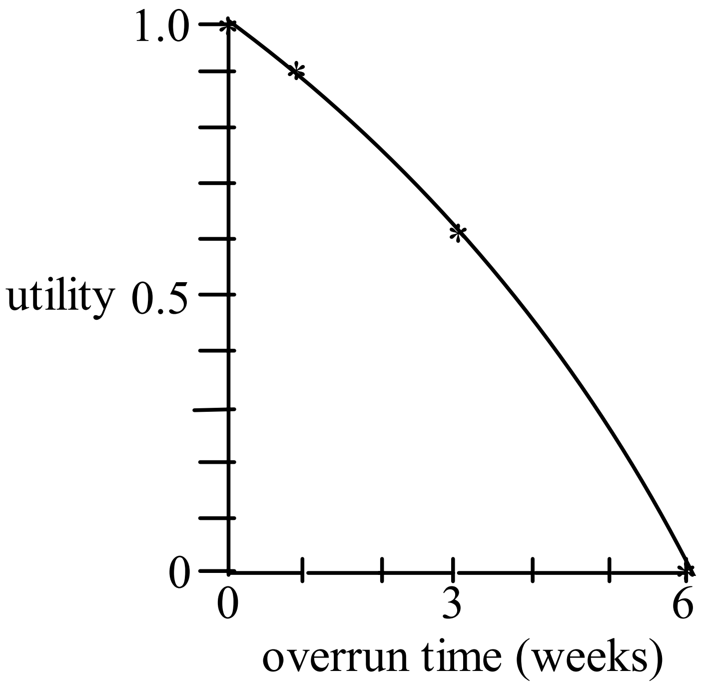
U(1-week overrun) = 0.9 .

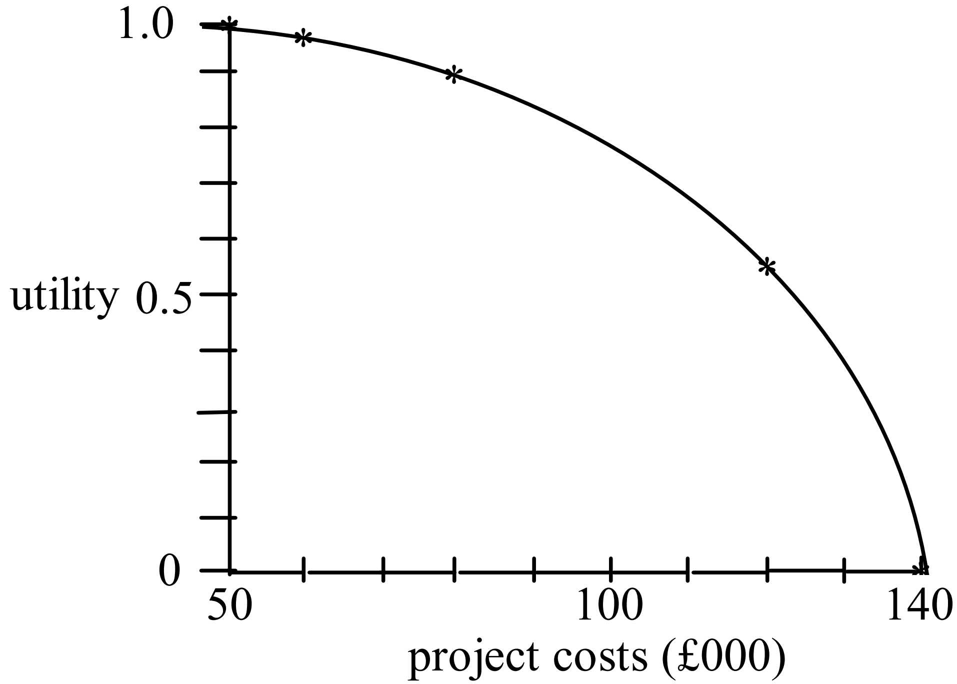
We then repeat the elicitation process to obtain a utility function for project cost.

Utility values for the four overrun times and five project costs that occur in the problem are:

U(0 weeks) = 1,
U(1 week) = 0.9,
U(3 weeks) = 0.6 and
U(6 weeks) = 0 .

U(£50,000) = 1,
U(£60,000) = 0.96,
U(£80,000) = 0.9,
U(£120,000) = 0.55 and
U(£140,000) = 0 .





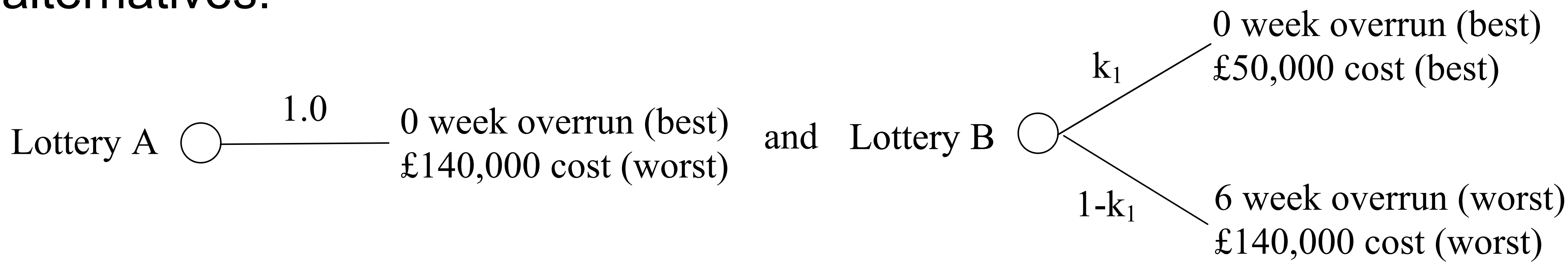
Step 2: Combining the single-attribute functions to obtain a multi-attribute utility function.

In order to produce a multi-attribute function U for comparing the two alternative courses of action in terms of their performance over both attributes, it remains to find weights k_1 and k_2, to slot in the equation:

$$U(\text{overrun, cost}) = k_1{*}U_1(\text{overrun}) + k_2{*}U_2(\text{cost}) + (1-k_1-k_2){*}U_1(\text{overrun}){*}U_2(\text{cost})$$

where U_1 and U_2 are the single-attribute utility functions.

To find k_1 we offer the project manager a choice between the alternatives:



Because we are finding k_1, it is attribute 1 (overrun) that appears at its best level in the certain outcome.

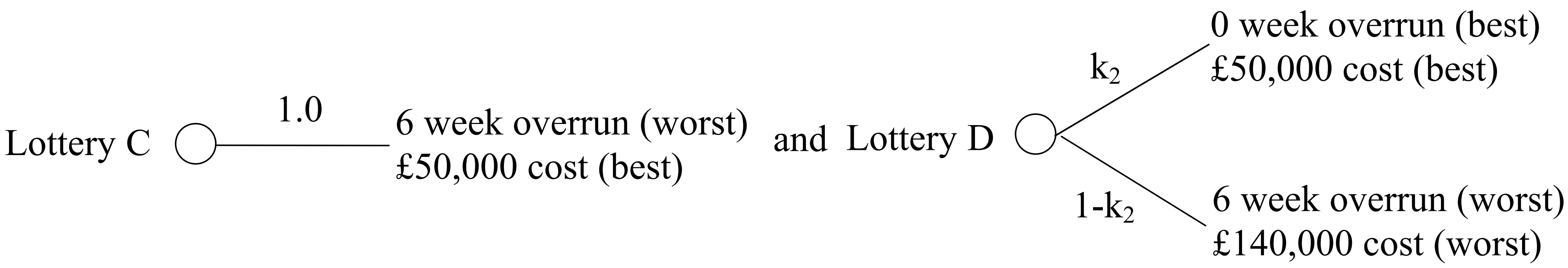
The decision maker is then asked what value the probability k_1 must have to make him indifferent between alternatives A (the certain outcome) and B (the lottery).

After some thought, he indicates that this probability k_1 is 0.8 .

This suggests that the “swing” from the worst to the best overrun is significant relative to project cost.

If he hardly cared whether the overrun was 0 or 6 weeks, it would have taken only a small value of k_1 to have made him indifferent to a gamble where overrun might turn out to be at its worst level.

To find k_2 we offer the project manager a choice between the alternatives:



The decision maker is asked what value the probability k_2 must have to make him indifferent between alternatives C (the certain outcome) and D (the lottery).

This time, he judges that the probability k_2 is 0.6 .

The fact that k_2 is less than k_1 suggests that the project manager sees the swing from the worst to best cost as being of less significance than the swing from worst to best overrun time.

Having been offered a project which is certain to incur the best cost, he requires a smaller probability of winning to tempt him to the lottery, where he might gain a project in which overrun is also at its best level, but where there is a risk of a project with costs at their worst level.

We can now use the multi-attribute utility function:

$$U(\text{overrun}, \text{cost}) = 0.8{*}U_1(\text{overrun}) + 0.6{*}U_2(\text{cost}) - 0.4{*}U_1(\text{overrun}){*}U_2(\text{cost})$$

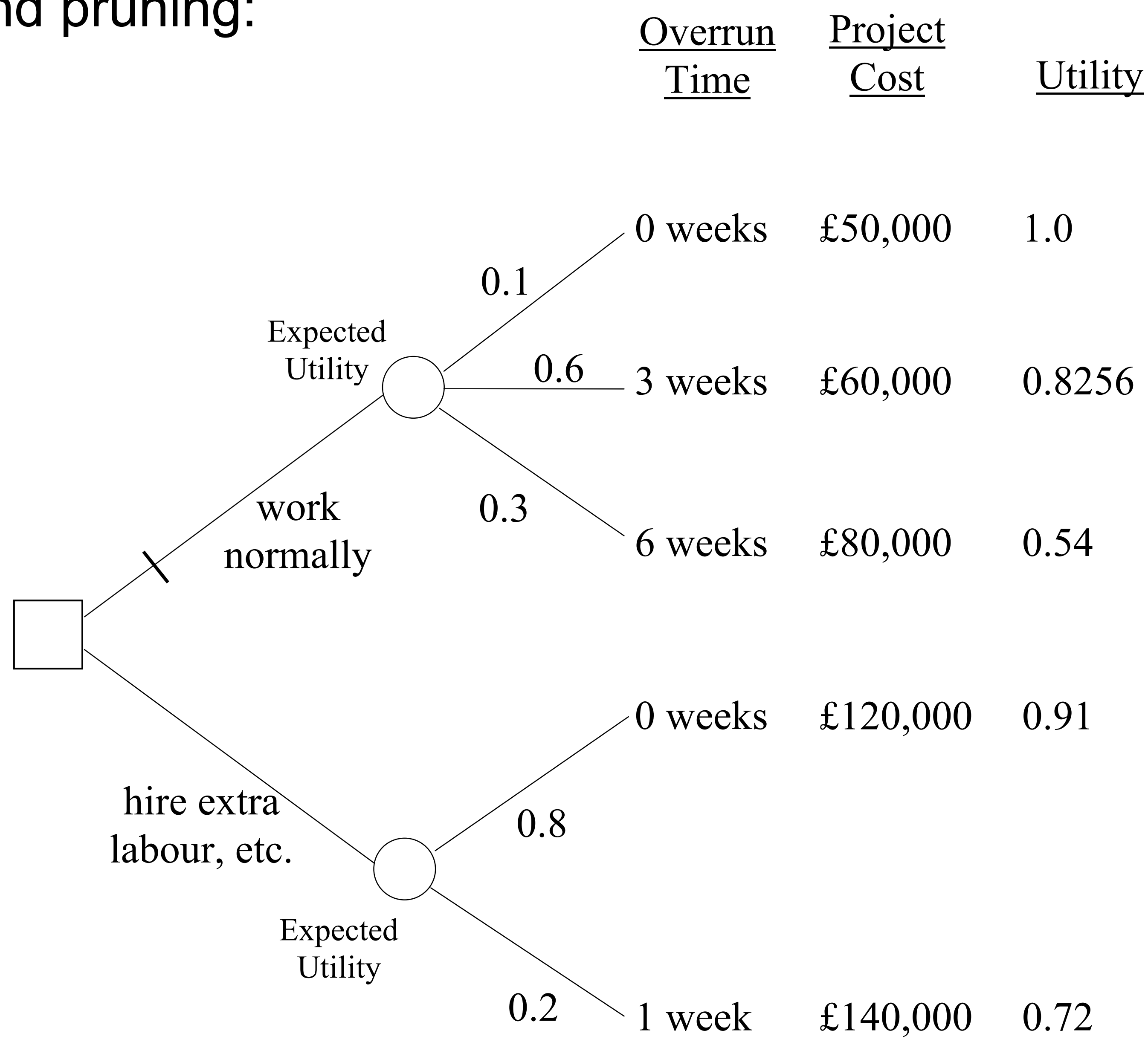
to determine the utilities of the different outcomes in the decision tree.

For example, the utility of a project that overruns by 3 weeks and costs £60,000 is:

U(3 weeks, £60,000) = 0.8*0.6 + 0.6*0.96 - 0.4*0.6*0.96 = **0.8256**

We use the usual procedure with decision trees of evaluating expected utilities, folding-back and pruning:

The results shown in the tree indicate that the project manager should hire extra labour and operate 24-hour working, as this yields the highest expected utility.



Step 3: Performing consistency checks and sensitivity analysis.

It is important to check that the results of the analysis have represented the project manager’s preferences faithfully.

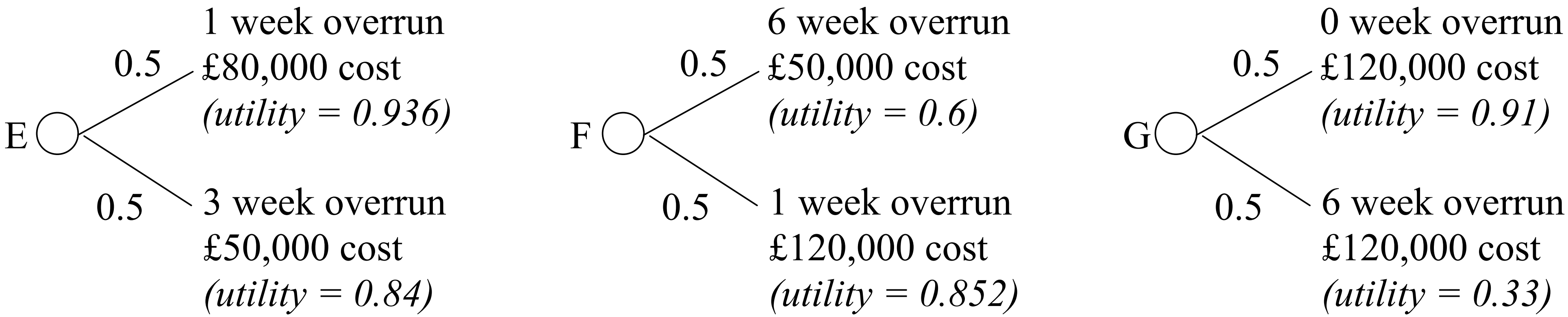
This can involve tracking back through the analysis and explaining why one option has performed well and another has performed badly.

If the decision maker does not feel that the explanations are consistent with his preferences then the analysis may need to be repeated.

In fact, it is likely that several iterations will be necessary before a consistent representation is achieved and, as the decision maker gains a greater understanding of his problem, he may wish to revise his earlier responses.

Another way of checking consistency is to offer the decision maker a new set of lotteries and ask him to rank them in order of preference.

For example, we could offer the project manager the three lotteries:



The expected utilities of these three lotteries are E: 0.888, F: 0.726 and G: 0.620, so if he is consistent he should rank them in order E, F, G.

We should also carry out sensitivity analysis on the probabilities and utilities by, for example, examining the effect of changes in the values of k_1 and k_2 on the recommended course of action.

Additivity Independence of Attributes:

We saw that if the attributes are mutually utility independent, then we can use the multiplicative model when calculating the multi-attribute utility function.

This was the model we used in the Decanal Engineering Corporation Problem.

When the additional condition of ***additivity independence*** holds between all pairs of attributes then we can use the even simpler additive form, in which the weights add up to 1.

In the case of just two attributes X and Y, we can express the ***additive model*** as:

$$U(x, y) = k_1 * U_1(x) + (1 - k_1) * U_2(y)$$

For this model to valid we need to ensure, in addition to mutual utility independence, that if attributes X and Y have best and worst outcomes X_L, X_S and Y_L, Y_S, then following pair of lotteries are equally preferable to the decision maker: