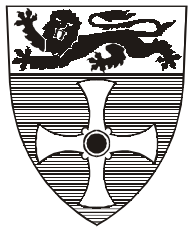
UNIVERSITY OF
NEWCASTLE



COMPUTING SCIENCE

Dynamic Selection of Software Components in the Face of Changing Requirements

Salah Merad, Rogério de Lemos and Tom Anderson

TECHNICAL REPORT SERIES

No. 664

March, 1999

Contact: S Merad
Salah.Merad@newcastle.ac.uk
http://www.cs.ncl.ac.uk/people/salah.merad/

.

Published by the University of Newcastle upon Tyne,
Department of Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, UK

Dynamic Selection of Software Components in the Face of Changing Requirements

Salah Merad, Rogério de Lemos and Tom Anderson

Centre for Software Reliability
Department of Computer Science
University of Newcastle
Newcastle upon Tyne, NE1 7RU, UK

ABSTRACT

Software components are characterised by their functional and non-functional properties. It is both difficult and expensive to build a software component that excels in all its non-functional properties and that can cope with a large input range for all environmental variables. We envisage to provide a computing system operating in a changing environment with a pool of software components, each having a specific profile, and a decision maker that selects the best component subject to the time varying environmental requirements. The resulting decision problem is a multi-attribute optimisation problem. In this paper, we propose several solution concepts for this problem and evaluate them qualitatively with respect to many criteria. Two of these concepts are treated in detail. The first is based on the combined utility function and the second is the compromise between the preference patterns of the various properties. The latter is game theoretic, and the solution is found by formulating the problem as a bargaining game. While the combined utility function can be expressed in a simple form when some independence assumptions between the properties hold, and it can incorporate the decision maker's value trade-offs, it involves evaluating subjectively a number of coefficients. But because the evaluations also need to be performed during run-time for new requirements, this becomes problematic. The game theoretic solution does not incorporate explicitly the decision maker's value trade-offs, but it is free of the subjective element and no evaluations are needed during run-time.

Key words: Non-functional requirements; Utility functions; Multi-attribute optimisation; Value trade-offs; Game theory.

1. Introduction

There is an increasing interest in component-based software engineering- see, for example, the special issue 'Working with Components' of IEEE Software (September/October 1998). Included are articles on testing by Weyuker, architecture by Ben-Shaul et al, and applications in telecommunications by Zave and Jackson. In this paper, we consider the decision problems which arise in a component-based software system operating in a changing environment.

Software components are characterised by their functional and non-functional properties or attributes (NF-attributes). It is both difficult and expensive to build a software component that excels in all its non-functional attributes and that can cope with a large input range for all environmental variables. Trade-offs between the attributes have to be made during the design of a software component. Moreover, the NF-requirements from the environment may vary during run-time. For example, the available memory space may be limited sometimes but large at other times. So, instead of having only a single component to cope with all eventualities, the system will be provided with several components that excel in different attributes. The system will then have a decision making algorithm that selects the best component for the situation. The objective of this paper is to describe the framework of the decision process and the issues that arise in its implementation.

The decision problem that we have is called a multi-attribute optimisation problem in the Decision Theory literature. The usual solution of this class of problems involves the construction of a utility function which amalgamates all the attributes, or a combined utility function (abbreviated CUF). During the construction process, the owner of the computing system, to whom we refer as the decision maker (abbreviated DM), is asked to compare alternatives or lotteries of alternatives. These comparisons are done using either the method of Von Neumann & Morgenstern (1947) or more recently, fuzzy decision making.

We describe the representation of the CUF introduced by Keeney, see Keeney & Raiffa (1976). This representation applies when some independence assumptions between the attributes are satisfied. It greatly simplifies the construction as it reduces the process to the evaluation of utility functions for each attribute in isolation or individual utility functions, and the evaluation of a small number of scaling coefficients. We discuss the problems that

may be encountered when the CUF approach is part of the decision support system of an autonomous computing system operating in a changing environment.

If the DM is not capable to form value trade-offs between the attributes because of the complexity of their interactions, or because the CUF is difficult to evaluate on-line for new requirements, we consider another solution which is a compromise between the preference patterns of all NF-attributes. We show that this is the solution of a bargaining game, where the players are agents, each one representing the interests of a single attribute, negotiating about which alternative is to be selected. We find the solution of this game by maximising the Nash product which represents a collective utility function - see Moulin (1988). This solution, called the Nash solution, achieves a compromise between all attributes, but it does not incorporate any interaction of preferences among the attributes.

In section 2, we introduce a model of a system of software components operating in an environment with changing NF-requirements. In section 3, we describe briefly the construction of the individual utility functions using the method of Von Neumann & Morgenstern (1947) and how to carry out the construction on-line for new requirements. In section 4, we consider two solution concepts for the selection problem. In the first part, we describe the representation of the CUF introduced by Keeney, and show how to use off-line data to make assessments on-line. In the second part, we describe the Nash solution for the bargaining game and discuss its use for our problem and explain its shortcomings. We illustrate this solution concept by solving an example. In the third part, we describe other solution concepts that are simpler but which use only a small amount of the available information. In section 5, we evaluate qualitatively the different solution concepts with respect to criteria such as the required information, the type of solution obtained, and the computational effort.

2. General Model of a System of Software Components

We have a computer system containing M software components or algorithms denoted by $A_1, A_2, ..., A_M$. $\mathrm{A} = \{A_1, A_2, ..., A_M\}$ is the set of available alternatives or the selection set. These components have a number of properties or attributes that can be classified into functional attributes and non-functional attributes (NF-attributes).

A functional attribute of a given component specifies the range of the environmental variable input that it can handle. Examples of NF-attributes are reliability, performance, maintainability, cost and capacity.

Let I_l, $l = 1,2,\ldots,L$, be the input environmental variables, and for $m = 1,2,\ldots,M$, let IR_{ml} denote the range of variable input I_l that component A_m can handle.

Let $X_1, X_2, \ldots, X_N$ be N NF-attributes.

Let a_{mn} be the value of attribute X_n in component A_m, where $m = 1,2,\ldots,M$, and $n = 1,2,\ldots,N$. Depending on the attribute, a_{mn} can be either a real, or an integer number. Without loss of generality, we will assume throughout this paper that the values a_{mn} are real numbers for all attributes.

The environment in which the system operates imposes both functional and non-functional requirements (NF-requirements).

Functional requirements:

Let EI_l be the range of input variable I_l that the system receives from the current environment. All functional requirements are strict in that the system can only use a component that can handle the whole input ranges for all input variables.

NF-requirements:

Let CV_n, be the range of values of attribute X_n that are required or desired by the environment at a certain time. For some attributes, CV_n is the set of values that are less than or equal to a critical value cv_n, and for others it is the set of values greater than or equal to a critical value cv_n. For some attributes, the environmental requirements are strict, but for others they are only desirable, that is, the critical values represent only levels of aspiration. When the requirements are strict, any component that does not satisfy them is excluded from the selection set.

The objective is to choose the best alternative for some specified environmental requirements. This is an optimisation problem with many attributes or a multi-attribute decision problem. This class of problems has been intensely investigated in the literature; see, for example, Keeney & Raiffa (1976) or Chankong & Haines (1983).

Without loss of generality, we assume that all the available software components are available for selection.

	NF-attributes				Functional attributes			
Attributes	X_1	X_2	…	X_N	I_1	I_2	…	I_L
Critical Values / Algorithms	CV_1	CV_2	…	CV_N	EI_1	EI_2	…	EI_L
A_1	a_{11}	a_{12}	…	a_{1N}	IR_{11}	IR_{12}	…	IR_{1L}
A_2	a_{21}	a_{22}	…	a_{2N}	IR_{21}	IR_{22}	…	IR_{2L}
⋮	⋮	⋮		⋮	⋮	⋮		⋮
A_M	a_{M1}	a_{M2}	…	a_{MN}	IR_{M1}	IR_{M2}	…	IR_{ML}

Let $\mathbf{a}_m = (a_{m1}, a_{m2}, ..., a_{mN})$ be the vector specifying component A_m, $m = 1,2,..., M$. Vector $\mathbf{a}_m$ will be referred to as the profile of alternative A_m. The problem is then to choose between M profile vectors corresponding to M software components with N attributes, subject to the environmental requirements. In fact, we want to optimise simultaneously all the attributes, and as a result we will have to make trade-offs or compromises between them.

3. Construction of individual utility functions

In this section, we will consider each attribute individually. The quantification of the relative preferences over the selection set is achieved by constructing empirically a utility function using the expected utility method of Von Neumann & Morgenstern (1947) or fuzzy decision making - see, for example, Seo (1995). The former is grounded in probability theory, whereas the latter is grounded in fuzzy logic.

The utility function is by definition subjective, in that different DMs will arrive at different functions and the same DM may arrive at different functions at different times. However, there are good heuristics that yield utility functions that reflect accurately the true preferences of the DM. Moreover, we deal with only one attribute at a time and the construction is done off-line, hence good results can be achieved. In this paper, we only use the method of Von Neumann & Morgenstern.

For $n = 1,2,..., N$, let $a_n^{(0)} = \min_{1 \le m \le M} \{a_{mn}\}$ and $a_n^* = \max_{1 \le m \le M} \{a_{mn}\}$, and let $RX_n = [a_n^{(0)}, a_n^*]$ be the range of values that attribute X_n can take over the available alternatives.

For each NF-attribute X_n, we want to construct a utility function $u_n : RX_n \to [0,1]$ scaled from 0 to 1. We say that u_n is normalised.

Note that $RX_n \supset \{a_{1n}, a_{2n}, ..., a_{Mn}\}$. We choose to define the utility function over the whole range RX_n instead of over the countable set $\{a_{1n}, a_{2n}, ..., a_{Mn}\}$ for convenience.

Let $x_n \in RX_n$. $u_n(x_n)$ is then the utility gained when a component has an amount x_n of attribute X_n.

In most of the remainder of this section, we drop the subscript n from the notation, so that $u_n(x_n)$ will be denoted by $u(x)$, for example. We will only introduce it when it is necessary.

For attribute X, let $x^{(W)}, x^{(B)} \in RX$ be, respectively, the worst and best values among the available alternatives. We assume the best (worst) alternative is the one with either the highest or lowest value (lowest or highest value). Then $x^{(W)}, x^{(B)} \in \{a^{(0)}, a^*\}$. As the utility function is scaled from 0 to 1, we have $u(x^{(B)}) = 0$ and $u(x^W) = 1$.

Let $x^{(1)}, x^{(2)}, x^{(3)} \in RX$, with $x^{(1)} < x^{(2)} < x^{(3)}$. Let $\langle x^{(1)}; p; x^{(3)} \rangle$ be the lottery in which outcome $x^{(1)}$ occurs with probability p and outcome $x^{(3)}$ occurs with probability $1 - p$. The DM is then asked to give an estimate to the value of p such that he will be indifferent between the certainty outcome $x^{(2)}$ and the lottery $\langle x^{(1)}; p; x^{(3)} \rangle$. This process of estimating p is called the indifference experiment.

Let $\hat{p}$ be the estimated value of p. Then

$$u(x^{(2)}) = \hat{p}u(x^{(1)}) + (1 - \hat{p})u(x^{(3)}) .$$

So, if the utilities at points $x^{(1)}$ and $x^{(3)}$ are known, we can find the utility at point $x^{(2)}$. We always start with the $x^{(B)}$ and $x^{(W)}$ as the end-points with known utilities.

The indifference experiment is repeated for different triplets, and checks for consistency are carried out periodically. Once the utility function is determined at a sufficient number of points, we find the equation of the best fit curve to these points. Let $h: XR \to [0,1]$ be its

functional form. The process of construction stops when no serious inconsistencies are found. Then

$$u(x) = h(x), \text{ for all } x \in XR .$$

The utility function u must reflect the DM's attitude to risk. We will consider the concept of attitude to risk using indifference experiments involving lotteries of only two outcomes that are real values. But the definitions can be generalised to all types of lotteries.

Definitions Let $x^{(1)}$, $x^{(2)} \in RX$, and $0 < p < 1$.

1. A DM is risk neutral if he is indifferent between the certainty outcome $px^{(1)} + (1-p)x^{(2)}$ and the lottery $\langle x^{(1)}; p; x^{(2)} \rangle$. That is, the utility function u is such that

$$u(px^{(1)} + (1-p)x^{(2)}) = pu(x^{(1)}) + (1-p)u(x^{(2)}) .$$

2. A DM is risk-averse if he prefers the certainty outcome $px^{(1)} + (1-p)x^{(2)}$ to the lottery $\langle x^{(1)}; p; x^{(2)} \rangle$. That is, the utility function u is such that

$$u(px^{(1)} + (1-p)x^{(2)}) > pu(x^{(1)}) + (1-p)u(x^{(2)}) .$$

3. A DM is risk-loving or risk-prone if he prefers the lottery $\langle x^{(1)}; p; x^{(2)} \rangle$ to the certainty outcome $px^{(1)} + (1-p)x^{(2)}$. That is, the utility function u is such that

$$u(px^{(1)} + (1-p)x^{(2)}) < pu(x^{(1)}) + (1-p)u(x^{(2)}) .$$

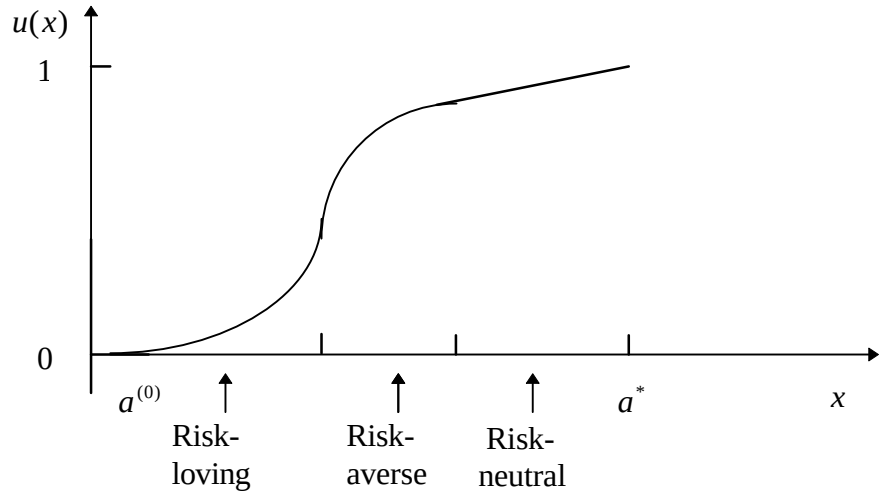
There is a link between the DM's attitude to risk and the form of his utility function. It is summarised it the following theorem.

Theorem 1

1. A DM is risk-averse if and only if his utility function is concave.
2. A DM is risk-loving if and only if his utility function is convex.
3. A DM is risk-neutral if and only if his utility function is linear.

***Proof*:** See Keeney and Raiffa (1976), pp 149-151.

The DM can be risk-averse within a certain range of values, risk-prone within another range, and risk-neutral within yet another range. The following graph illustrates this point. Suppose that high values are the most desired.



Changing requirements:

The critical values for the attributes will change over time, and it follows that the utilities for the alternatives will usually change too. Hence, the utility functions need to be constructed for all possible values of cv. But we expect to have a small number of functional forms to represent the utilities for all values of cv. For example:

$$u(x) = \begin{cases} h^{(1)}(x;cv) & \text{for} \quad cv \le a^{(0)}, \\ h^{(2)}(x;cv) & \text{for} \quad a^{(0)} < cv < a^{*} \text{ and } x \le cv, \\ h^{(3)}(x;cv) & \text{for} \quad a^{(0)} < cv < a^{*} \text{ and } x > cv, \\ h^{(4)}(x;cv) & \text{for} \quad cv \ge a,^{*} \end{cases}$$

where $h^{(i)}(.;cv), i = 1,\ldots,4$, are one-variable parametric functions.

This can be verified empirically by constructing the utility function for several values of cv. The utilities for the alternatives can then be computed on-line for all possible values of cv. For example, suppose that for attribute X_n the high values are the most desired. Suppose also that for r randomly selected values of cv_n lower than $a_n^{(0)}$, namely $cv_n^{(i)}, i = 1,\ldots,r$, we have verified that $u_n(x_n) = \lambda \log(x_n - cv_n^{(i)}) + \gamma$, where λ and γ are scaling coefficients. Then it may be safe to assume that

$$u_n(x_n) = \lambda \log(x_n - cv_n) + \gamma \ , \text{ for all } cv_n < a_n^{(0)}.$$

If for a given critical value, the DM is risk-neutral over the whole range, then the utility function is independent of the critical value as we show in the following theorem. This is

likely to occur for $cv < a^{(0)}$ if high values of the attribute are desired , and for $cv > a^{*}$ if low values of the attribute are desired.

Theorem 2 If the DM is risk-neutral over the whole interval RX for critical value cv, then his utility function is independent of cv, and it is given by

Eq. 1 $$u(x) = \frac{x - x^{(W)}}{x^{(B)} - x^{(W)}}.$$

Proof: From Theorem 1, the utility function of the DM is linear as he is risk-neutral. Hence we can write

Eq. 2 $$u(x) = f(cv)x + g(cv),$$

where $f(cv)$ and $g(cv)$ are two real functions of cv independent of x.

Now, we know that $u(x^{(W)}) = 0$ and $u(x^{(B)}) = 1$, which combined with Eq. 2 yields

Eq. 3 $$f(cv)x^{(W)} + g(cv) = 0,$$

and

Eq. 4 $$f(cv)x^{(B)} + g(cv) = 1.$$

Solving Eq. 3 and Eq. 4 simultaneously, we obtain

$$f(cv) = \frac{1}{x^{(B)} - x^{(W)}}$$

and

$$g(cv) = \frac{-x^{(W)}}{x^{(B)} - x^{(W)}}.$$

And Eq. 1 follows immediately. **QED.**

4. Solution Concepts

What do we mean by the best solution to the selection problem? Below, we will discuss several solution concepts.

4.1 Combined Utility

In multi-attribute decision making, we assume that the DM is able to choose between two alternatives on the basis of their profile vectors. That is, the DM can choose the best of two N-component vectors. This choice is not derived from more basic premises, it is subjective. When the number of alternatives M is small, the DM may be able to order them from the most preferred to the least preferred with relative ease. But with a large number of alternatives, it may be very tedious to do the ranking while being consistent. The problem will be greatly simplified if we can construct a utility function which amalgamates all the attributes, or a CUF with a simple expression. The optimal solution is then the alternative with the highest combined utility.

When some independence assumptions between the individual attributes are verified, the problem has a structure that makes it possible to express the combined utility in a simple form. In fact, it is shown that, even when the assumptions are not verified for all attributes, the utility function derived using these assumptions is still a good approximation.

Let $U_{\mathbf{cv}} : RX_1 \times RX_2 \times .... RX_N \to [0,1]$ denote the DM's CUF under environmental critical values **cv**. $U_{\mathbf{cv}}(x_1, x_2, ..., x_N)$ is then the utility gained if $x_1, x_2, ..., x_N$ are the values taken by attributes X_n, $n = 1, 2, ..., N$, respectively. The problem is to find the form of the $N-$variables utility function $U_{\mathbf{cv}}$. For ease of notation, we will drop the subscript **cv** from $U_{\mathbf{cv}}$.

There is an extensive literature on this problem- see, for example, Keeney & Raiffa (1976) or Chankong & Haines (1983). A large part of this section is based on the former reference. Therein, it is shown that, under some independence conditions that are satisfied in many cases, it is possible to write $U(x_1, x_2, ..., x_N)$ in an additive or multiplicative forms. But even after this simplification, the construction of the utility function is still very tedious, and it is very difficult to achieve consistency.

We will only consider the so called utility independence assumption because it is the one that is satisfied more often. We start by defining some terms that will be used subsequently.

Definitions

1. Let $X = \{X_1, X_2, ..., X_N\}$ be the set of attributes. The complement of attribute X_n in the set X is the subset $\{X_1, X_2, ..., X_{n-1}, X_{n+1}, ... X_N\}$ which we denote by $\overline{X}_n$.
2. Attribute X_n is utility independent of its complement $\overline{X}_n$ if the conditional preference order for lotteries involving only changes in the levels of attribute X_n does not depend on the levels at which the attributes in $\overline{X}_n$ are held fixed.

For example, consider the following vector profiles:

$\omega^{(1)^+} = (x_1^{(1)}, \overline{x}_1^+)$, $\omega^{(2)^+} = (x_1^{(2)}, \overline{x}_1^+)$, $\omega^{(3)^+} = (x_1^{(3)}, \overline{x}_1^+)$, $\omega^{(4)^+} = (x_1^{(4)}, \overline{x}_1^+)$.

$x_1^{(i)}$ is the level of attribute X_1 in profile $\omega^{(i)^+}$, $i = 1,..,4$. In all four outcomes, the attributes in the complement of X_1 are fixed at level $\overline{x}_1^+$.

Let $\langle \omega^{(1)^+}; p; \omega^{(2)^+} \rangle$, where $0 \le p \le 1$, be a lottery involving profiles $\omega^{(1)^+}$ and $\omega^{(2)^+}$.

Similarly, let $\langle \omega^{(3)^+}; p; \omega^{(4)^+} \rangle$, where $0 \le q \le 1$, be another lottery involving profiles $\omega^{(3)^+}$ and $\omega^{(4)^+}$.

Suppose that lottery $\langle \omega^{(1)^+}; p; \omega^{(2)^+} \rangle$ is preferred to lottery $\langle \omega^{(3)^+}; q; \omega^{(4)^+} \rangle$. We then write $\langle \omega^{(1)^+}; p; \omega^{(2)^+} \rangle \succ \langle \omega^{(3)^+}; q; \omega^{(4)^+} \rangle$.

Let $\omega^{(i)} = (x_1^{(i)}, \overline{x}_1)$, $i = 1,..4$, be four other profiles in which all the complement attributes of X_1 are at level $\overline{x}_1$. Let $\langle \omega^{(1)}; p; \omega^{(2)} \rangle$ and $\langle \omega^{(3)}; q; \omega^{(4)} \rangle$ be two other lotteries.

Attribute X_1 is utility independent of its complement attributes $\overline{X}_1$ if

$$\langle \omega^{(1)^+}; p; \omega^{(2)^+} \rangle \succ \langle \omega^{(3)^+}; q; \omega^{(4)^+} \rangle \Rightarrow \langle \omega^{(1)}; p; \omega^{(2)} \rangle \succ \langle \omega^{(3)}; q; \omega^{(4)} \rangle, \qquad \text{for all } \overline{x}_1.$$

Definition Attributes $X_1, X_2, ..., X_N$ are mutually utility independent if every subset of $\{X_1, X_2, ..., X_N\}$ is utility independent of its complement.

The next theorem gives the representation of the DM that involves only the individual one-dimensional utility functions u_n and $N+1$ scaling constants. That is, the CUF can be expressed in the form

$$U(x_1, x_2, ..., x_N) = H(u_1(x_1), u_2(x_2), ..., u_N(x_N), k_1, ..., k_{N+1}),$$

where H is a real valued function and $k_1, k_2 \ldots, k_{N+1}$ are constants.

Theorem 3 (Keeney & Raiffa 1976) If attributes $X_1, X_2, \ldots, X_N$ are mutually utility independent, then

Eq.5
$$\begin{aligned} U(x_1,x_2,\ldots,x_N) = & \sum_{n=1}^{N} k_n u_n(x_n) + K\sum_{\substack{n=1 \\ n<i}}^{N} k_n k_i u_n(x_n)u_i(x_i) \\ & + K^2 \sum_{\substack{n=1 \\ n<i \\ i<l}}^{N} k_n k_i k_l u_n(x_n)u_i(x_i)u_l(x_l) \\ & + \ldots\ldots + K^{N-1} k_1 k_2 \ldots\ldots k_N u_1(x_1)u_2(x_2)\ldots\ldots u_N(x_N) \ , \end{aligned}$$

where

$$k_n = U\left(x_1^{(W)}, x_2^{(W)}, \ldots, x_{n-1}^{(W)}, x_n^{(B)}, x_{n+1}^{(W)}, \ldots., x_N^{(W)}\right), \ n = 1,2,\ldots, N.,$$

and K are scaling constants related by equation

Eq. 6
$$1 + K = \prod_{n=1}^{N} (1 + Kk_n).$$

***Proof*:** See Keeney and Raiffa (1976), pp 289-291.

Scaling constant k_n is then the utility gained when attribute X_n takes the best value, whereas all the other attributes take the worst values.

The scaling constants are meant to insure consistency among the individual utilities and hence to incorporate the DM's value trade-offs between the attributes. Therefore, the constants cannot be evaluated independently from each other.

When $\sum_{n=1}^{N} k_n = 1$, then $K = 0$ and Eq.5 reduces to the additive utility function

$$U(x_1, x_2, \ldots., x_N) = \sum_{n=1}^{N} k_n u_n(x_n).$$

This representation was first introduced by Fishburn (1964), and in a series of papers he derived necessary and sufficient conditions in many situations, see Fishburn (1970) for a detailed account of this work. The additive representation holds under the independence assumption called preference independence.

When $\sum_{n=1}^{N} k_n \neq 1$, the representation given by Eq.5 is called multiplicative for the following reason. When $\sum_{n=1}^{N} k_n \neq 1$, $K \neq 0$, so multiplying each side of Eq.5 by K, adding 1 to each, and factoring yields

$$KU(\mathbf{x}) + 1 = \prod_{n=1}^{N} \left[k_n u_n(x_n) + 1 \right],$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.

For other independence concepts and corresponding representations, see Fishburn and Keeney (1974).

The values of constants $k_1, k_2, \ldots, k_N$ can be found if we can obtain N linearly independent equations in these constants. The equations can be obtained as a result of either a certainty scaling or a probabilistic scaling or both.

Using certainty scaling:

Let $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in RX_1 \times RX_2 \times \ldots RX_N$ be two profile vectors, such that the DM is indifferent between them. Then,

Eq. 7 $$U(\mathbf{x}^{(1)}) = U(\mathbf{x}^{(2)}).$$

Eq. 7 combined with the representation given in Eq.5 yields an equation relating the scaling constants.

Carrying the following type of indifference experiments yields simple linear equations.

Let $i, j \in \{1,2,\ldots,N\}$ and $\mathbf{x}^{(1)} = (x_1^{(W)}, x_2^{(W)}, \ldots, x_{i-1}^{(W)}, x_i^{'}, x_{i+1}^{(W)}, \ldots, x_N^{(W)})$, where $x_i^{'} \in XR_i$. The DM needs to find $x_j^{'} \in XR_j$ such that

$$\mathbf{x}^{(1)} = (x_1^{(W)}, x_2^{(W)}, \ldots, x_{i-1}^{(W)}, x_i^{'}, x_{n+1}^{(W)}, \ldots, x_N^{(W)}) \approx (x_1^{(W)}, x_2^{(W)}, \ldots, x_{j-1}^{(0)}, x_j^{'}, x_{j+1}^{(W)}, \ldots, x_N^{(W)}) = \mathbf{x}^{(2)}$$

It is easy to verify that $U(\mathbf{x}^{(1)}) = k_i u_i(x_i^{'})$ and $U(\mathbf{x}^{(2)}) = k_j u_k(x_j^{'})$ which yields equation

$$k_i u_i(x_i^{'}) = k_j u_j(x_j^{'}).$$

Using uncertainty scaling:

Let $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \in RX_1 \times RX_2 \times \ldots RX_N$ be three profile vectors. Suppose that the DM is indifferent between profile vector $\mathbf{x}^{(1)}$ and the lottery $\langle \mathbf{x}^{(2)}; p; \mathbf{x}^{(3)} \rangle$. Then,

Eq. 8 $$U(\mathbf{x}^{(1)}) = pU(\mathbf{x}^{(2)}) + (1-p)U(\mathbf{x}^{(3)}).$$

Similarly, Eq. 8 combined with the representation given in Eq.5 yields an equation relating the scaling constants.

Carrying out the following indifference experiments yield directly the values of the scaling constants.

Let $\mathbf{x}^{(1)} = \left(x_1^{(W)}, x_2^{(W)}, \ldots, x_{n-1}^{(W)}, x_n^{(B)}, x_{n+1}^{(W)}, \ldots, x_N^{(W)}\right)$, $\mathbf{x}^{(2)} = \left(x_1^{(B)}, x_2^{(B)}, \ldots, x_N^{(B)}\right)$ and $\mathbf{x}^{(3)} = \left(x_1^{(W)}, x_2^{(W)}, \ldots, x_N^{(W)}\right)$. Let p, $0 \le p \le 1$, be chosen by the DM so that $\mathbf{x}^{(1)} \approx \langle \mathbf{x}^{(2)}; p; \mathbf{x}^{(3)} \rangle$. Now, using Eq.5, it is easy to verify that $U\left(\mathbf{x}^{(1)}\right) = k_n$, $U\left(\mathbf{x}^{(2)}\right) = 1$, and $U\left(\mathbf{x}^{(3)}\right) = 0$. Hence, Eq. 8 yields

$$k_n = p.$$

We can see that in both methods of scaling , the DM has to choose vector profiles that are easy to compare and that yield simple linear equations; they do not have to be from those characterising the available alternatives. The value of constant K can then be found from Eq. 6.

In both the certainty and uncertainty scaling procedures, the DM has to decide whether two profiles or lotteries of profiles are equivalent. As we pointed out above, the decision is not derived from more basic premises; it is purely subjective. Usually, the DM relies on previous experience in such comparisons. One of the major problems is that this method yields inconsistency, and it is therefore necessary to carry out checks and then repeat all over again if serious inconsistencies are found. It can be very tedious to achieve a satisfactory utility function in which there are no major inconsistencies, especially with the existence of environmental requirements.

In addition to the problem of consistency, it is important to check that all the equations obtained are linearly independent. When there is a large number of attributes, this becomes operationally difficult.

Since the critical values from the environment will vary over time, new estimates of the scaling coefficients are needed. To estimate off-line the scaling coefficients for all possible values of the critical values cv_n may be very time consuming due to the large number of combinations that need to be covered, and hence not practical. Therefore, it may be necessary to find a way to evaluate the scaling coefficients on-line every time there are new NF-requirements using off-line estimates. This could be achieved in two ways.

1. If the coefficients are evaluated off-line for a "reasonable" number of combinations of the critical values and the results stored in the memory of the system, then, depending on the structure that exists in the data, it may be possible to obtain evaluations for new combinations from the stored data using, for example, numerical interpolation, statistical techniques or neural networks.

2. If there is a structure in the value trade-offs of the DM, then this can be exploited to deduce the value of the new coefficients from a previous evaluation. Here we consider an example of a structure characterised by the following two assumptions.

   Assumption 1 (ASP_1)

   Suppose that when the environmental critical values are given by $\mathbf{cv} = (cv_1, cv_2, \ldots, cv_N)$, profile vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are equivalent if and only if vectors $\mathbf{x}^{(1)} - \mathbf{cv}$ and $\mathbf{x}^{(2)} - \mathbf{cv}$ are equivalent under no requirements.

   Assumption 2 (ASP_2)

   $$\mathbf{x}^{(1)} - \mathbf{cv} \approx \mathbf{x}^{(2)} - \mathbf{cv} \quad \Rightarrow \mathbf{x}^{(1)} - \mathbf{cv}' \approx \mathbf{x}^{(2)} - \mathbf{cv}' \text{ for all } \mathbf{cv}'.$$

   In the indifference experiments done off-line, suppose that the environmental critical values are set equal to $cv_1^{(1)}, cv_2^{(1)}, \ldots, cv_N^{(1)}$. Assume that under these conditions the DM finds that profiles $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are equivalent. Then, from ASP_1, vectors $\mathbf{x}^{(1)} - \mathbf{cv}^{(1)}$ and $\mathbf{x}^{(2)} - \mathbf{cv}^{(1)}$ are equivalent.

   Let $\mathbf{cv}^{(2)} = (cv_1^{(2)}, cv_2^{(2)}, \ldots, cv_N^{(2)})$ describe the new environmental requirements.

   Then, from ASP_2, we deduce that vectors $\mathbf{x}^{(1)} - \mathbf{cv}^{(2)}$ and $\mathbf{x}^{(2)} - \mathbf{cv}^{(2)}$ are also equivalent.

   But this is equivalent to saying that vector profiles $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are equivalent under the requirements given by $\mathbf{cv}^{(2)}$.

Using the representation in Eq.5, we obtain an equation relating the scaling constants. We substitute the individual utility $u_n(x_n)$ with the parametric function $h_n\left(x_x; cv_n^{(2)}\right)$.

Hence, it is enough to carry out a large number of indifference experiments off-line for a single combination of critical values, because the resulting equations can be used to find on-line equations relating the scaling coefficients for all other combinations of critical values. We need to carry out more than N indifference experiments, because N experiments may not be enough to obtain the required N linearly independent equations for some combinations of critical values.

But in practice, it is doubtful that the required structures will exist in most cases. Thus, the method based on the CUF is not operationally applicable, and hence another method is needed.

4.2 The Nash Solution (Nash [1950])

We propose a solution concept which yields the selection of an alternative which is good in some well defined sense, without relying on the DM's value trade-offs. For every attribute there is a certain preference pattern over the available alternatives. The solution we seek is the one which achieves a compromise between all preference patterns.

This is the solution we obtain if the DM delegates his decision making to self-interested rational agents, each representing an attribute. The agent for attribute X_n has utility function u_n. The agents will then have to bargain with each other to reach an agreement on which alternative should be selected. Each agent will try to impose its most preferred alternative. This solution will yield the alternative that is as satisfactory as possible for each attribute.

Note that, in this formulation, the DM's value trade-offs between the attributes are not incorporated into the structure of the game.

Nash (1950) proposed a solution to the bargaining game. He reasoned that, to avoid the prospect of not reaching agreement, the players/agents are willing to submit their conflict to a "fair" arbitrar, an impartial outsider who will resolve the conflict by suggesting a solution. An arbitration scheme is defined by a function, i.e., a rule, which associates to each conflict, a unique payoff to the players. He then gave a number of axioms to formulate

mathematically our subjective intuition of fairness. We summarise the way Luce & Raiffa (1957) expressed these axioms verbally.

i. The arbitration solution should give each player at least as much as he could get under the worst case, and there should not be any other feasible payoff preferred by all players.

ii. The arbitration solution should not depend upon the particular utility units used by the players.

iii. The arbitration scheme should be egalitarian in the sense that it is independent of the names or labels attached to the players.

iv. The solution must be robust. Slight perturbations or errors of measurements should not alter drastically the arbitrated solution.

Definition (Nash Product) Let ω_n, $n = 1,2,..,N$, be a sequence of real numbers such that $0 \le \omega_n \le 1$ and $\sum_{n=1}^{N} \omega_n = 1$. Then, the function $F:[0,1]^N \to [0,1]$ such that $F(v_1, v_2, ..., v_N) = \prod_{n=1}^{N} v_n^{\omega_n}$ is the (generalised) Nash product (Binmore [1992]).

The coefficients ω_n are a measure of the strength or importance of the players. In our problem they correspond to importance of the attributes and they need to be evaluated by the DM. These coefficients are different from the scaling constants of the CUF.

Nash's original formulation considered only the case with two players with equal importance.

We choose to construct the individual utilities on the scale 0 to 1, but this choice is arbitrary. In fact we can assign any positive number to the most desired alternative. But we must assign 0 to the least desirable to conform with the Nash solution.

The Nash product not only satisfies these four axioms, it can be shown that it is the only function which does so (Nash [1950]). Hence, these fairness conditions implicitly define a unique arbitration scheme for bargaining games.

Before we describe the method to find the Nash solution, we will give some definitions.

Definitions

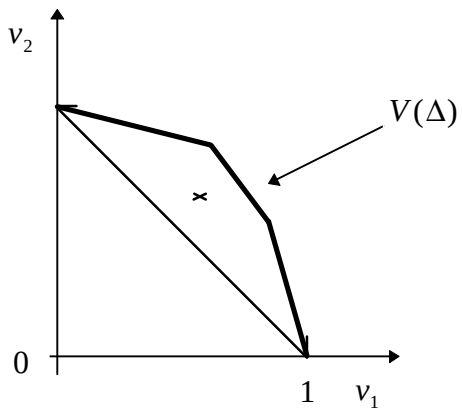
1. A randomised strategy δ is an M-tuple $(\delta_1, \delta_2, \ldots, \delta_M)$, where δ_m, $m = 1,2,\ldots,M$, is the probability of using alternative A_m.
2. Let $EU_n : [0,1]^M \to [0,1]$ be a real valued function. $EU_n(\delta)$ is the expected utility of attribute X_n under the randomised strategy δ. We have

Eq. 9 $$EU_n(\delta) = \sum_{m=1}^{M} \delta_m u_m(a_{mn}).$$

3. A strategy δ is said to be Pareto optimal if it is not dominated.
4. A strategy δ is dominated if there exists a strategy $\beta = (\beta_1, \beta_2, \ldots, \beta_M)$ such that $EU_n(\delta) \le EU_n(\beta)$, for all $n \in \{1,2,\ldots,N\}$ and at least one inequality is strict.
5. Let Δ be the set of randomised strategies that are Pareto optimal and such that the expected utility from every player is positive. Then

$$\Delta = \{\delta \mid \delta \text{ is Pareto-optimal and } EU_n(\delta) > 0\}.$$

Let $V(\Delta) = \{v = (v_1, v_2, \ldots, v_N) \mid v_n = EU_n(\delta), \delta \in \Delta\}$. $V(\Delta)$ is called the set of Pareto-optimal payoffs. It can be shown that $V(\Delta)$ is a portion of the boundary of the convex hull generated by the points $\mathbf{a}_m$, $m = 1,2,\ldots,M$, which form its extreme points or corners. The set $V(\Delta)$ will hence be generated by at most M points. For example, if there are two attributes, $V(\Delta)$ will be either one single point, or a segment, or a finite number of adjacent segments. The graph below illustrates the set of Pareto-optimal payoffs drawn in bold. It represents an example with five alternatives and two attributes. The extreme points of the polyhedron represents four alternatives, and the internal point represents the fifth alternative.



To find the Nash solution δ^*, we need to solve the non-linear constrained optimisation problem

$$\max_{\delta \in \Delta} \prod_{n=1}^{N} \left(EU_n(\delta) \right)^{\omega_n},$$

or, equivalently, maximise the Nash product, i.e.,

$$\max_{v \in V(\Delta)} \prod_{n=1}^{N} v_n{}^{\omega_n} .$$

$V(\Delta)$ is a bounded closed convex set, and the Nash product $\prod_{n=1}^{N} v_n{}^{\omega_n}$ is a continuous function. Hence the constrained optimisation problem has a unique solution.

Let $\left(v_1^*, v_2^*, ..., v_N^* \right)$ be the point at which the Nash product is maximal. Then, by solving the system of equations

$$\sum_{m=1}^{M} \delta_m u_m(a_{mn}) = v_n^*, \qquad n = 1,2.,.., N ,$$

we can find the optimal strategy δ^*.

Characteristics of the Nash solution:

1. There is no need for interpersonal comparison between the utilities of the various attributes;
2. it is the solution that a fair arbitrar would suggest to the bargaining players. The arbitrar will try to satisfy some consistency requirements and he will be able to defend his suggested solutions with some good rationalisation. That is, he should be able to formulate and to defend the basic principles which lie behind his suggested compromises-they should not be arbitrary (Luce & Raiffa (1957));
3. it can incorporate the importance or strength of the attributes;
4. this concept yields a definite solution even though the solution is usually randomised.

The following example illustrates the use of the Nash solution concept.

A. There are four algorithms available to select from, each having four attributes. Two of them are functional - the inputs for temperature and pressure, and the other two are non-functional - reliability and performance, both of equal importance. Reliability is measured in average number of failures per year, and the performance is measured in average time in seconds to process a specified computation. The following table gives the input ranges from

the environment for both temperature and pressure, the critical values required for the non-functional attributes, and the characteristics of all four algorithms with respect to functional and non-functional attributes. The second row contains the critical values for the non-functional attributes and the input ranges for temperature and pressure in the current environment.

	NF-attributes		Functional attributes	
	Reliability X_1	**Performance** X_2	**Temperature** I_1	**Pressure** I_2
Critical Values / **Algorithms**	**1**	**0.8**	**(25.0,50.0)**	**(3,8)**
A_1	0.85	0.80	(15.0,70.0)	(2,9)
A_2	0.80	0.82	(10.0,90.0)	(4,9)
A_3	0.90	0.75	(20.0,60.0)	(1,9)
A_4	0.95	0.70	(15.0,55.0)	(2,10)

We can see that algorithm A_2 cannot be used because its range of functionality for pressure does not cover the required range of input in the current environment. We then have the following table.

	Reliability X_1	**Performance** X_2
Critical Values / **Algorithms**	**1**	**0.8**
A_1	0.85 **1**	0.80 **0**
A_3	0.90 **0.5**	0.75 **0.5**
A_4	0.95 **0**	0.70 **1**

Construction of the utility function:

Let $u_n(x_n) = h_n\left(x_n; cv_n\right)$. Suppose that the DM is risk-neutral for both attributes over the whole ranges. So, using Theorem 2, we have

$$u_1(x_1) = \frac{x_1 - 0.95}{0.85 - 0.95} = -10x_1 + 9.5$$

since $x_1^{(0)} = 0.95$ and $x_2^* = 0.85$.

In the same way, we find the expressions of $u_2(x_2)$ which is

$$u_2(x_2) = -10x_2 + 8.$$

The resulting utilities for the 3 algorithms are given in the above table in bold. The negotiation set is the segment whose endpoints are $\mathbf{a}_1 = (1,0)$ and $\mathbf{a}_4 = (0,1)$. Note that the point $\mathbf{a}_3 = (0.5,0.5)$ lies on this segment. Hence the optimal strategy will be a mixture of algorithms A_1 and A_4 or algorithm A_3 if it is equivalent to the mixture.

The set of randomised strategies to choose from is $\Delta = \{\delta = (\lambda,0,0,1-\lambda), 0 \le \lambda \le 1\}$.

For $\delta \in \Delta$, we have $EU_1(\delta) = \lambda$ and $EU_2(\delta) = 1-\lambda$..

We want to maximise $EU_1(\delta)EU_2(\delta) = \lambda\ (1-\lambda)$ over $[0,1]$.

The product $\lambda(1-\lambda)$ is maximal for $\lambda = \frac{1}{2}$, so that algorithm A_1 should be chosen with probability $\frac{1}{2}$ and algorithm A_4 should be chosen with probability $\frac{1}{2}$.

Let δ^* denote the optimal strategy. Then $\delta^* = \left(\frac{1}{2},0,0,\frac{1}{2}\right)$. The expected utilities for the NF-attributes are:

$$EU_1(\delta^*) = \frac{1}{2} = u_1(a_{31}),$$

and

$$EU_2(\delta^*) = \frac{1}{2} = u_2(a_{32}).$$

Hence, it is also optimal to chose algorithm A_3, which is a deterministic strategy.

B. Everything is as in part A except that the critical value on performance is now 0.75. But this requirement is only desirable, not strict. The relevant data are in this table.

Construction of the utility function

To obtain its functional form, we would need to carry out several indifference tests.

Here we only have three alternatives, so we only need to find the utility for $x_2 = 0.75$.

We need to find the value of p such that $0.75 \approx \langle 0.7; p; 0.8 \rangle$. Hence

$$u_2(0.75) = pu_2(0.7) + (1-p)u_2(0.8) = p.$$

It is then easy to verify that the optimal strategy is

1. for $p \le 0.5$, select either alternative A_1 or A_4 with equal probability;
2. for $p \ge 0.5$, select A_3.

4.3 *Other Solution Concepts*

Without loss of generality, we assume throughout that the high values of an attribute are the most desirable, and that all the available components satisfy the functional requirements and the strict NF-requirements.

A. Methods that do not require individual utilities:

Because no utility functions are used in these methods, the decision maker's strength of preferences and his attitude to risk are either not incorporated or the DM is implicitly assumed risk-neutral as in the third method below.

1. Optimal with respect to the most important attribute.

Suppose that attribute X_{n^*} is considered to be crucial for a given application, and we want it to have the best possible value. The optimal component A_{m^*} is such that

$$a_{m^*n^*} = \max_{1\le m\le N} a_{mn^*}.$$

The obtained solution is clearly pareto-optimal.

2. Minimum weighted sum of ranks

Let r_{mn} be the rank of component A_m with respect to attribute X_n, and let

$$R_m = \sum_{n=1}^{N} \omega_n r_{mn}$$

be the weighted sum of ranks for component A_m. The optimal component A_{m^*} is such that

$$R_{m^*} = \min_{1\le m\le M} R_m.$$

Theorem 4 The minimum sum of weighted ranks criterion yields a Pareto-optimal solution.

***Proof*:** By contradiction. Let A_{m^*} be the component with the minimal sum of weighted ranks. Assume that its profile $\left(a_{m^*1}, a_{m^*2}, \ldots, a_{m^*N}\right)$ is not Pareto-optimal. Hence there exists a component $A_{\hat{m}} \in A$ such that

$$a_{\hat{m}n} \ge a_{m^*n} \qquad \text{for all } n = 1,2,\ldots,N$$

and

$$a_{\hat{m}\bar{n}} > a_{m^*\bar{n}} \qquad \text{for some } \bar{n} \in \{1,2,\ldots,N\}.$$

Now

$$a_{\hat{m}n} \geq a_{m^*n}, \forall n \in \{1,2,\ldots,N\} \Rightarrow r_{\hat{m}n} \leq r_{m^*n}, \forall n \in \{1,2,\ldots,N\},$$

and

$$a_{\hat{m}\bar{n}} > a_{m^*\bar{n}} \Rightarrow r_{\hat{m}\bar{n}} < r_{m^*\bar{n}}.$$

Because all the weight coefficients and the ranks are positive, we have

$$R_{\hat{m}} = \sum_{n=1}^{N} \omega_n r_{\hat{m}n} < \sum_{n=1}^{N} \omega_n r_{m^*n} = R_{m^*},$$

which contradicts the initial assumption. **QED**.

In the case where there are many options that achieve the minimum, the tie can be broken by selecting the best option for the most important attribute.

3. Maximum weighted product

This solution concept is similar to the Nash solution described above, but without the use of individual utility functions and it is restricted to deterministic strategies. In fact, it implicitly assumes risk-neutrality and hence the critical values do not appear in the performance index. It was first introduced by Bridgeman (1922) and used by Miller & Starr (1960) for goal programming problems. See also the discussion in Johnsen (1968) of this measure of utility and other measures.

Let

$$PI_m = a_{m1}^{\omega_1} a_{m2}^{\omega_2} \ldots \alpha_{mN}^{\omega_N}$$

be the performance index of component A_m. Then the optimal component A_{m^*} is such that

$$PI_{m^*} = \max_{1 \leq m \leq M} PI_m.$$

This performance index works well because it is independent of the units in which the attributes are expressed. It is straightforward to verify that the solution obtained is Pareto-optimal.

B. A method that requires inter-attribute utility comparisons: Minimax regret when the most crucial attribute is unknown

Suppose that for some given application a certain attribute will turn out to be crucial and hence it needs to take the highest possible value, but this crucial attribute is difficult to pinpoint in advance. We want to find a selection strategy so as to minimise a measure of utility loss from not choosing the best component for the attribute which turns out to be crucial

Let V_{mn} be the utility of component A_m when attribute X_n is the only one that matters for the application and the critical value is cv_n. The utilities $\{V_{mn}, m = 1,\ldots, M; n = 1,\ldots, N\}$ need to be comparable and thus they need to be evaluated by considering all attributes collectively, instead of constructing individual utility functions as was done above. This is a difficult task, and there is no general method for such constructions. We propose the following method:

1. For each attribute, evaluate $\tilde{a}_{mn} = \dfrac{a_{mn} - cv_n}{cv_n}$. Then

$$\tilde{a}_{mn} \geq \tilde{a}_{m'n'} \Leftrightarrow V_{mn} \geq V_{m'n'},$$

for all $m, m' \in \{1,2,\ldots, M\}$ and $n, n' \in \{1,2,\ldots, N\}, n \neq n'$.

The elements $\{\tilde{a}_{mn}, m = 1,\ldots, M; n = 1,\ldots, N\}$ are pure numbers whose values are between -1 and +1, and their comparison is meaningful. But to incorporate the DM's intensity of preferences and his attitude to risk, we need to associate a utility to each normalised value $\tilde{a}_{mn}$.

2. Let $y = \dfrac{x - cv}{cv}$. We need to find empirically the functional form of $V(y)$, where $V:[-1,1] \to [0,1]$, with $V(-1) = 0$ and $V(1) = 1$.

3. Determine the utilities $\{V_{mn}, m = 1,\ldots M; n = 1,\ldots, N\}$ using

$$V_{mn} = V\left(\frac{a_{mn} - cv_n}{cv_n}\right).$$

Note that the lowest utility value is greater than or equal to 0, not necessarily 0. Similarly, the highest utility value is less than or equal to 1, not necessarily equal to 1. That is the utilities are not scaled from 0 to 1.

Hence, once the functional form of $V(y)$ is determined, the utilities V_{mn} can be easily computed during run-time for all critical values.

Let $L(m,n)$ be the loss in utility when component A_m is used and attribute X_n turns out to be the most crucial. Then

$$L(m,n) = \max_{1\le m\le M}\{V_{mn}\} - V_{mn}.$$

Because the most crucial attribute is unknown, this problem can be formulated as a two-person zero-sum game between the controller and the environment. The environment chooses the most crucial attribute so as to maximise the DM's expected utility loss.

The payoff matrix of the game is $\left(L(m,n)\right)_{\substack{1\le m\le M\\ 1\le n\le N}}$ and it usually yields randomised strategies.

Let

$$\overline{L} = \min_{1\le m\le M}\max_{1\le n\le N} L(m,n)$$

and

$$\underline{L} = \max_{1\le n\le N}\min_{1\le m\le M} L(m,n).$$

We have

$$\underline{L} \le \overline{L}.$$

The DM can guarantee a maximum utility loss of $\overline{L}$ by selecting the component for which $\overline{L}$ is attained. This is a deterministic strategy.

The zero-sum game yields randomised strategies except when $\underline{L} = \overline{L}$.

Let $\delta = (\delta_1, \delta_2, \ldots \delta_M)$, where δ_m is the probability that the controller selects component A_m, denote the strategy of the DM. Similarly, let $\beta = (\beta_1, \beta_2, \ldots \beta_N)$, where β_n is the probability that the most crucial attribute is X_n.

Let (δ^*, β^*) be the Nash equilibrium (solution) of the game, and $L(\delta^*, \beta^*)$ the expected utility loss under the worst case scenario if the controller adopts the randomised strategy δ^*. Then

$$\underline{L} \le L(\delta^*, \beta^*) \le \overline{L}.$$

Hence, using the randomised strategy δ^*, the DM's maximum expected loss is less than the maximum loss under the deterministic strategy.

We also have the following equilibrium inequalities

$$L(\delta^*, \beta) \leq L(\delta^*, \beta^*) \leq L(\delta, \beta^*).$$

The first inequality implies that, whatever the attribute that will turn out to be crucial, the expected loss is at most $L(\delta^*, \beta^*)$ if the randomised strategy δ^* is employed.

5. Evaluation of methods

In the previous section, we suggested a number of solution concepts and the methods to compute the solution. The solution concepts that should be adopted depends on the decision maker's interests, the information he can provide, the practicality of the computation, especially during run-time, and most importantly the trade-offs between the computational effort and the gain in utility over simpler but cruder methods. But the best way of carrying out such an evaluation is a properly designed experiment in a real situation using real decision makers (Chankong and Haimes (1983)). Below, we will evaluate qualitatively the suggested methods with respect to the following criteria:

1. Information required and the manner and extent of the DM's participation;
2. the type of solution obtained;
3. the computational effort required.

5.1 Required Information

5.1.1 Components' profiles

In all the methods, the performance of the components need to be measured with respect to each attribute in some chosen scale. In the minimum sum of weighted ranks method, the components need only to be ranked from the worst to the best for each attribute, so the precision in the measurement is less important than in the other methods so long as the difference in performance between the components is obvious.

5.1.2 Individual utility functions

They are required in the combined utility and bargaining solution methods but not in the other methods. But the constructions are done off-line and involve only one attribute at a time. The utility functions incorporate information about the DM's strength of preferences and his attitude to risk. The minimum sum of ranks method incorporate the least information about the DM's structure of preferences.

5.1.3 Value trade-offs

Only the combined utility method incorporates the DM's value trade-offs explicitly through the scaling constants evaluated in indifference experiments that can only be carried out off-line. The off-line evaluations can be used on-line for changing requirements only when there is a structure in the value trade-offs. Moreover, the method of evaluation of the constants is subjective and can lead to inconsistencies which can be difficult to eliminate completely. Other methods such as the minimum weighted sum of ranks, the weighted product and the bargaining solution use the importance coefficients as partial measures of the value trade-offs. But their evaluation is subjective and there are no tests to insure consistency as for the scaling coefficients in the CUF. The Bargaining solution, the weighted product and the minimum sum of ranks seek the best compromise solution given the information available and the kind of strategies allowed. So they incorporate an element of value trade-off, but it is a value trade-off that is not specific to a particular DM, but one that all rational decision makers with the same weighting coefficients will make. The methods that base the selection on the most crucial attribute do obviously not incorporate any measure of value trade-offs.

5.2 Type of solution

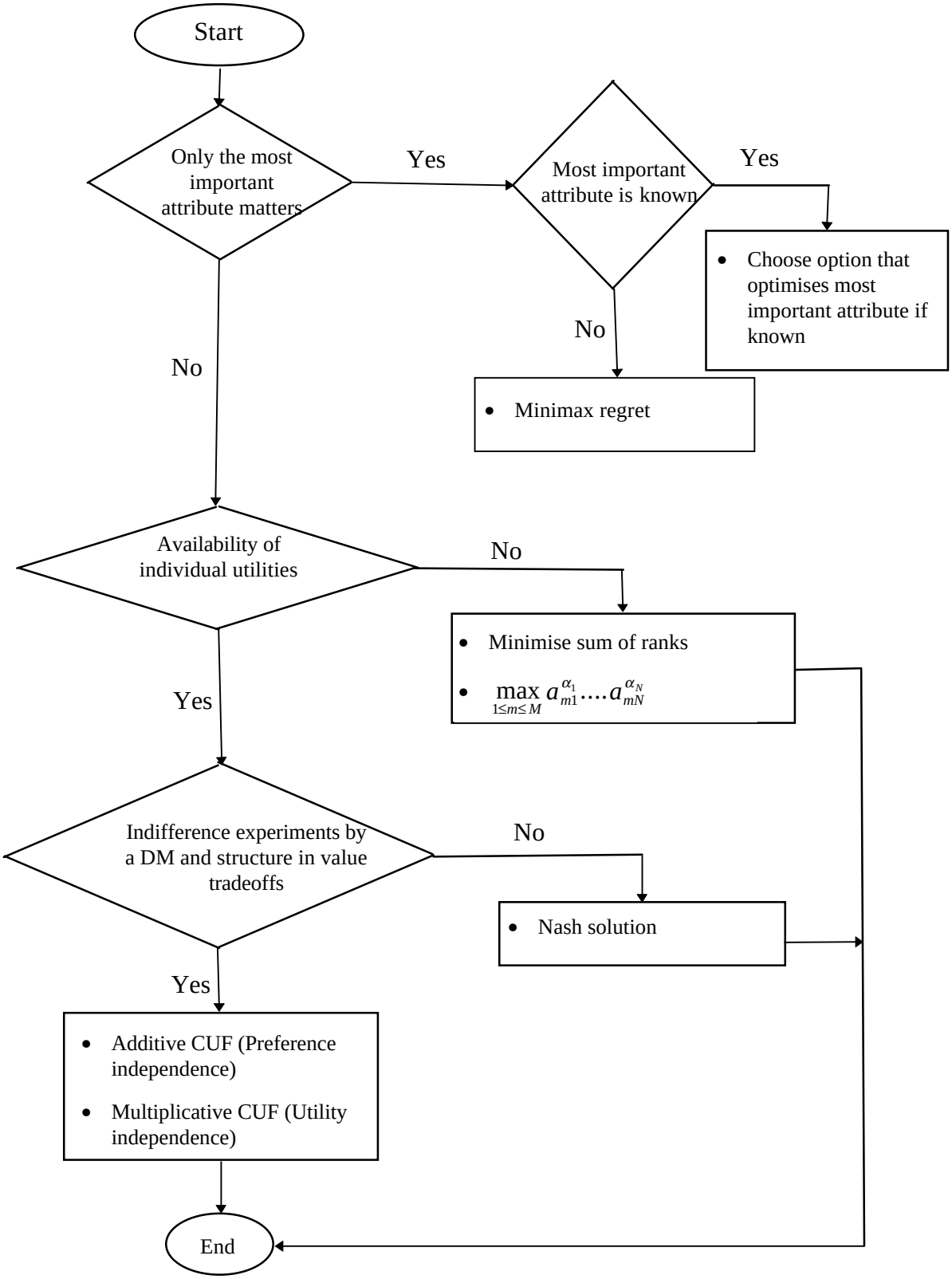
All the methods suggested yield Pareto-optimal solutions. This is a minimal requirement for any method. The two solutions based on game theory yield randomised strategies, and hence perform well on average in the long-term. For instance, the bargaining solution will yield better results than the weighted product in the long term, notwithstanding that the latter method incorporates less information on the DM's structure of preferences.

5.3 Computational effort

The amount of computations needed and the quantity of information that needs storing vary greatly between the methods. The methods based on the minimum sum of weighted ranks, the weighted product, or the best option for the most crucial attribute when it is known require very simple operations such as sorting, and multiplication, and little storage capacity. When practical, the combined utility method requires the storage of N scaled individual utility functions for all subsets (there can be up to 2^M subsets) of components that are admissible for various combinations of critical values, the scaling coefficients for many combinations of critical values, and the solution during run-time of linear systems of equations with N unknowns which represents the major computational operations. The bargaining solution requires the storage of the individual utilities, and the main computational is the optimisation of a separable function over the boundary of a convex hull for which efficient methods exist. In addition this solution is robust.

The minimax regret method requires the construction and the storage of the utility function $V(y)$. The main computation is the solution of a linear program with N constraints to find the equilibrium solution.

Chart of methods for optimal selection of software components under changing requirements



References

1. Ben-Shaul, I., Gish, J. W. and Robinson, W. : An integrated Network Component Architecture. IEEE Software (September/October 1998).
2. Binmore, K. : Fun and Games. A Text on Game Theory, D. C. Heath and Company, 1992.
3. Bridgeman, P. W. : Dimensional Analysis, Yale University Press, New Haven, 1922.
4. Chankong, V. and Haines, Y. Y. : Multiobjective Decision Making. Theory and Method, North-Holland, 1983.
5. Fishburn, P. C. : Decision and Value Theory, Wiley, New York, 1964.
6. Fishburn, P. C. : Utility Theory for Decision Making, Wiley, New York, 1970.
7. Fishburn, P. C. and Keeney, R. L. : Seven independence concepts and continuous multiattribute utility functions. Journal of Mathematical Psychology,11, pp 294-327, 1974.
8. Johnsen, E. : Studies in Multiobjective Decision Models, Studentlitteratur, Lund, 1968.
9. Keeney, R. L. and Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs, Wiley, New York, 1976.
10. Luce, R. D. and Raiffa, H.: Games and Decisions, Wiley, New York, 1957.
11. Miller, D. W. and Starr, M. K. : Executive Decisions and Operations Research, Prentice-Hall, Englewood Cliffs, 1960.
12. Moulin, H. : Axioms of Cooperative Decision Making, Cambridge University Press, 1988.
13. Nash, J. F. : The Bargaining Game, Econometrica, 18, pp155-162, 1950.
14. Seo, F. : Construction of Fuzzy Utility Functions in Group Decision Making. In: Kacprzyk, J., Nurmi, H., Fedrizzi, M.(eds.) : Consensus Under Fuziness, Kluwer Academic Publications, pp.211-230, 1995.
15. Von Neumann, J. and Morgenstern, O. : Theory of Games and Economic Behavior, Wiley, 2nd edition, 1947.
16. Weyuker, E. J. : Testing Component-Based Software. IEEE Software (September/October 1998).
17. Zave, P. and and Jackson, M. : A Component-Based Approach to Telecommunication Software. IEEE Software (September/October 1998).