Multiattribute Utility Copulas

Ali E. Abbas

Department of Industrial and Enterprise Systems Engineering, College of Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, alieabbas@uiuc.edu

We introduce the notion of a multiattribute utility copula that expresses any (i) continuous; (ii) bounded multiattribute utility function that is (iii) nondecreasing with each of its arguments, and (iv) strictly increasing with each argument for at least one reference value of the complement attributes, in terms of single-attribute utility assessments. This formulation provides a wealth of new functional forms that can be used to model preferences over utility-dependent attributes and enables sensitivity analyses to some of the widely used functional forms of utility independence. We introduce a class of utility copulas, called Archimedean utility copulas, and discuss the conditions under which it yields the additive and multiplicative forms. We also discuss linear and composite transformations of utility copulas that construct utility functions with partial utility independence. We conclude with the risk aversion functions that are induced by utility copula formulations and work through several examples to illustrate the approach.

Subject classifications: utility elicitation; utility dependence; copula; utility copula; multiattribute utility.
Area of review: Decision Analysis.
History: Received September 2007; revisions received February 2008, April 2008; accepted May 2008. Published online in Articles in Advance August 12, 2009.

1. Introduction

The construction of a representative multiattribute utility function is a fundamental step in decision making under uncertainty and decisions with multiple attributes. Several methods have been proposed for its construction in the literature. One approach constructs a deterministic value function over the attributes, and then assigns a one-dimensional utility function over the value function (Matheson and Howard 1968). This approach does not require any assumptions of utility independence (for further details, see Dyer and Sarin 1982, Matheson and Abbas 2005).

A second approach constructs the utility function by making direct utility assessments over the attributes and establishing the presence of certain utility independence conditions to determine its functional form. In particular, if every attribute is utility independent of its complement, then the multiattribute utility function can be expressed as a multilinear combination of single-attribute utility assessments (Keeney and Raiffa 1976):

\[ U(x_1, \ldots, x_n) = \sum_{i=1}^{n} k_i U_i(x_i) + \sum_{i=1}^{n} \sum_{j>i}^{n} k_{ij} U_i(x_i) U_j(x_j) + \cdots + k_{123\ldots n} U_1(x_1) U_2(x_2) \cdots U_n(x_n). \] (1)

A stronger independence condition also holds if every subset of the attributes is utility independent of its complement and the utility function has either the multiplicative form

\[ 1 + k U(x_1, \ldots, x_n) = \prod_{i=1}^{n} [1 + k k_i U_i(x_i)], \] (2)

or the additive form

\[ U(x_1, \ldots, x_n) = \sum_{i=1}^{n} k_i U_i(x_i). \] (3)

The simplicity of constructing multiattribute utility functions using single-attribute assessments has popularized the use of the functional forms (1), (2), and (3) in many decision analysis applications. If these utility independence conditions do not hold, then we are faced with the need to incorporate utility dependence among the attributes into our analysis. Several authors have discussed this issue and have presented methods to incorporate utility dependence (see, for example, Farquhar 1975, Kirkwood 1976, Bell 1979, Keeney and Raiffa 1976, Keeney 1981, Abbas and Howard 2005). In particular, Farquhar (1975) proposes a general decomposition theorem to derive the functional form of a multiattribute utility function with certain utility independence structures known as fractional hypercubes. He shows that special cases of this formulation reduce to the multilinear form. Kirkwood (1976) incorporates utility dependence by defining “parametrically dependent” conditional utility functions. Bell (1979) incorporates utility dependence by interpolating conditional utility functions at the boundary values of the domain. Keeney and Raiffa (1976) construct a multiattribute utility function with utility dependence by assessing an isopreference contour and substituting it for one of the conditional utility assessments. Keeney (1981) takes an alternate approach and reformulates the attributes of a decision problem to derive a new set of attributes with a lesser degree of utility dependence.

Abbas and Howard (2005) introduce a class of multiattribute utility functions called attribute dominance utility...
functions. This class is continuous, bounded, nondecreasing with each of its arguments, and has a value equal to zero if any of the attributes is at a minimum value (this latter condition is known as the grounding condition for joint probability functions). They show that attribute dominance utility functions can be constructed using products of normalized marginal-conditional utility assessments or using single-attribute utility assessments and a copula structure. For more information on copula structures, see Sklar (1959) and Nelsen (1998). With this formulation, an attribute dominance utility function can be expressed as

\[ U(x_1, \ldots, x_n) = C(U_1(x_1), \ldots, U_n(x_n)), \tag{4} \]

where \( U(x_1, \ldots, x_n) \) is an attribute dominance utility function and \( C \) is a copula structure.

Abbas and Howard (2005) observe that using the conventional copula structures that were developed for joint probability functions results in attribute dominance utility functions that satisfy the \( n \)-increasing condition (for differentiable utility functions, the \( n \)-increasing condition implies that the mixed partial derivative \( \partial^n U(x_1, \ldots, x_n) / \partial x_1 \cdots \partial x_n \) is nonnegative). They observe that this is not a necessary condition for attribute dominance utility functions and pose the development of more general copula structures as an area for future research. Following this path, Abbas et al. (2007) conduct sensitivity analysis to the multiplicative form of utility independence, and in this paper we introduce a general framework for constructing multivariate utility functions using single-attribute utility assessments and a new multivariate function that we call the utility copula. This formulation relaxes the attribute dominance grounding condition mentioned above and also allows for utility functions with either positive or negative mixed partial derivatives.

The mixed partial derivative of a utility function is also known as its multivariate risk aversion (Richard 1975). A decision maker is said to be multivariate risk-averse (risk seeking) if the mixed partial derivative is negative (positive). A multivariate risk averse decision maker with a utility function, \( U(x, y) \), facing prospects \( x_0 < x_1, y_0 < y_1 \), prefers a lottery that provides a 50–50 chance of \( (x_0, y_1) \) and \( (x_1, y_0) \) to another that provides a 50–50 chance of \( (x_0, y_0) \) and \( (x_1, y_1) \)—i.e., he prefers getting one of the best and one of the worst prospects than a 50–50 chance of getting the best of both or the worst of both. In our work, we construct utility copulas that exhibit multivariate risk-averse (or risk-seeking) behavior on the domain of the utility function.

The use of utility copulas to construct multivariate utility functions also enables sensitivity analyses to the widely used forms of utility independence. We illustrate, using a Monte Carlo simulation, how the analyst can determine the percentage of incorrect decisions that could be made in a given problem by making a utility independence approximation if the attributes have utility dependence. We also present a class of utility copulas called Archimedean utility copulas, and discuss the conditions under which it yields the additive and multiplicative forms of utility functions. We then consider linear and composite transformations of utility copulas that incorporate partial utility independence among the attributes and enable sensitivity analyses to the multilinear form. We conclude with a derivation of the risk-aversion functions that are induced by using the utility copula approach.

In our search of the literature, we have found some related work on the use of copula structures to incorporate probability dependence in decision analysis (see, for example, Yi and Bier 1998 and Clemen and Reilly 1999). We have also found some work on the percentage of incorrect decisions that could be made by making a probability independence approximation when the variables have probability dependence (see, for example, Keefer 2004 and Abbas 2006). In the statistics literature, we have also found some work on quasi-copula structures that relax the \( n \)-increasing condition of probability copulas and replace it with a milder condition known as the Lipschitz condition (Alsin et al. 1993 and Genest et al. 1999). Finally, in the finance literature, we have found some work on Lévy copula structures that characterize the dependence in multidimensional Lévy processes using univariate Lévy processes (Kallsen and Tankov 2006).

The remainder of this paper is structured as follows. Section 2 presents the basic definitions and notation that will be used in the remaining sections of the paper. Section 3 defines a class of utility copulas called Archimedean utility copulas and conducts sensitivity to the additive and multiplicative forms of utility independence. Section 4 presents linear and composite transformations of utility copulas and methods to incorporate partial utility independence. Section 5 derives the risk-aversion functions induced by using utility copula formulations. Section 6 contains concluding remarks.

2. Multiattribute Utility Copulas

2.1. Basic Definitions

We assume in all of our analyses that the decision maker follows the axioms of normative utility theory (von Neumann and Morgenstern 1947) and has a multivariate utility function, \( U(x_1, \ldots, x_n) \), defined over \( n \) attributes, \( X_1, \ldots, X_n \). We use the lower case, \( x_i, i \in \{1, 2, \ldots, n\} \), to denote an instantiation of attribute \( X_i \), and use \( x_i^0 \) and \( x_i^* \) to denote the minimum and maximum values of \( X_i \), respectively. We use \( \bar{X}_i \) to denote the set of complement attributes to \( X_i \), and use \( \bar{x}_i \) to denote an instantiation of this complement. We also use the vector \((x_1, \ldots, x_n)\) to represent a consequence of the decision situation on the domain \([x_1^0, x_1^*] \times [x_2^0, x_2^*] \times \cdots \times [x_n^0, x_n^*]\). For simplicity, we also refer to this consequence as \((x_i, \bar{x}_i)\).

We assume that the multivariate utility function is (i) continuous, (ii) bounded, (iii) nondecreasing with each
of its arguments, and (iv) for each argument, \( x_i \), there exists at least one instantiation of the complement, \( \bar{x}_i \), such that \( U(x_i, \bar{x}_i) \) is strictly increasing with \( x_i \). This class of utility functions includes all continuous and bounded utility functions that are strictly increasing with each of their arguments, as well as other classes that are nondecreasing with their arguments at certain instantiations of the domain. Because the utility function is bounded, we assume hereafter (with no further loss of generality) that it is normalized to range from zero to one.

We seek the multivariate function \( C_\lambda \) that satisfies the equality

\[
U(x_1, \ldots, x_n) = C_\lambda(f_1(x_1, \bar{x}_1), \ldots, f_n(x_n, \bar{x}_n)),
\]

where \( f_i \) is a strictly increasing function of \( x_i \) at the instantiation \( \bar{x}_i \).

To provide a meaningful interpretation for \( f_i \) (and to simplify its assessment), we define it as a normalized conditional utility function of \( x_i \) at a particular instantiation, \( \bar{x}_i \), i.e.,

\[
f_i(x_i, \bar{x}_i) \triangleq U_i(x_i | \bar{x}_i) = \frac{U(x_i, \bar{x}_i) - U(x_0, \bar{x}_i)}{U(\bar{x}_i, \bar{x}_i) - U(x_0, \bar{x}_i)},
\]

Keeney and Raiffa (1976) provide a similar equation to (5) and derive the mathematical foundations of the additive, multiplicative, and multinumerical forms of multivariate utility functions. Our goal is to derive the function \( C_\lambda \) for all utility functions that satisfy (i)-(iv) above. With the regularity conditions on the utility function, we are guaranteed the existence of at least one instantiation \( \bar{x}_i \) for which \( U_i(x_i | \bar{x}_i) \) is strictly increasing. If the utility function is strictly increasing with each argument across the entire domain, then we can define a conditional utility function at any instantiation of the complement attributes. We can also define \( 2^n - 1 \) conditional utility functions for each attribute, \( X_i \), at all possible combinations of the boundary (minimum and maximum) values of its complement. In particular, we define the maximum marginal utility function as

\[
U_i(x_i | \bar{x}_i) = \frac{U(x_i, \bar{x}_i) - U(x_0, \bar{x}_i)}{1 - U(\bar{x}_i, \bar{x}_i)}, \quad U(x_0, \bar{x}_i) \neq 1,
\]

and the minimum marginal utility function as

\[
U_i(x_i | \bar{x}_i) = \frac{U(x_i, \bar{x}_i)}{U(\bar{x}_i, \bar{x}_i)}, \quad U(x_i, \bar{x}_i) \neq 0.
\]

**Definition 1 (Multiattribute Utility Copula).** A multiattribute utility copula, \( C_\lambda(v_1, \ldots, v_n) \), is a multivariate function of \( n \) variables that satisfies the following conditions:

1. **Normalized Range and Domain:** The function, \( C_\lambda \), is a continuous mapping from the \( n \)-dimensional hypercube \([0, 1]^n\) to the interval \([0, 1]\) and is normalized such that

\[
C_\lambda(0, \ldots, 0) = 0 \quad \text{and} \quad C_\lambda(1, \ldots, 1) = 1.
\]

2. **Nondecreasing:** The function, \( C_\lambda \), is nondecreasing with each of its arguments.

3. **Positive Linear Transformation at Reference Values:** For each argument \( v_i \), there exist some reference values, \( \lambda_{i,j}, i \neq j \), for which the utility copula satisfies

\[
C_\lambda(\lambda_{i,1}, \ldots, \lambda_{i,j-1}, v_i, \lambda_{i,j+1}, \ldots, \lambda_{i,n}) = a_i v_i + b_i,
\]

\[
i = 1, \ldots, n,
\]

where \( 0 \leq \lambda_{i,j} \leq 1 \), \( 0 < a_i \leq 1 \), and \( 0 \leq b_i < 1 \).

**Proposition 1.** Any multiattribute utility function that is (i) continuous, (ii) bounded, (iii) nondecreasing with each of its arguments, and (iv) strictly increasing with each argument for at least one reference value of the complement attributes, can be expressed in terms of normalized conditional utility functions, \( U_i(x_i | \bar{x}_i^m) \), \( i = 1, \ldots, n \), and a multiattribute utility copula, \( C_\lambda \), as

\[
U(x_1, \ldots, x_n) = C_\lambda(U_1(x_1 | \bar{x}_1^m), \ldots, U_n(x_n | \bar{x}_n^m)).
\]

Proposition 1 explains the importance of the utility copula definition as it enables the construction of a large class of multiattribute utility functions using single-attribute utility assessments. In the next sections, we present several examples of utility copulas and methods for their construction. First, we discuss the inverse problem of finding the utility copula for a given utility function and a given set of conditional utility assessments.

**Example 1.** (a) **Utility Function with Utility Dependence:** Suppose that we wish to derive the utility copula, \( C_\lambda(s, t) \), that satisfies \( C_\lambda(s, 1) = a_s s + b_s \), \( C_\lambda(1, t) = a_t t + b_t \), for the utility function

\[
U(x, y) = \frac{1 - e^{-\gamma s y}}{1 - e^{-\gamma}}, \quad 0 \leq x, y < 1.
\]

Because \( \lambda_{i2} = \lambda_{i1} = 1 \), the required utility copula provides a positive linear transformation at the maximum margins of its domain. We therefore determine the normalized conditional utility functions at these instantiations by direct substitution into (7) to get

\[
U_i(x | y^*) = \frac{U(x, y^*) - U(x_0, y^*)}{U(x^*, y^*) - U(x^0, y^*)} = \frac{1 - e^{-\gamma s}}{1 - e^{-\gamma}} \triangleq s \Rightarrow x = \frac{-1}{\gamma} \ln(1 - s(1 - e^{-\gamma})),
\]

\[
U_j(y | x^*) = \frac{1 - e^{-\gamma t}}{1 - e^{-\gamma}} \triangleq t \Rightarrow y = \left( \frac{-1}{\gamma} \ln(1 - t(1 - e^{-\gamma})) \right)^{1/\gamma}.
\]
Substituting for \( x, y \) from (13) and (14) into (12) gives the following utility copula:

\[
C_A(s, t) = \frac{1 - e^{-(1/\gamma)\ln(1-s(1-e^{-\gamma}))\ln(1-t(1-e^{-\gamma}))}}{1 - e^{-\gamma}},
\]

\[0 \leq s, t \leq 1. \tag{15}\]

For consistency, note that (15) is indeed normalized and satisfies the marginal condition of (10):

\[
C_A(0, 0) = \frac{1 - e^{-(1/\gamma)\ln(1\ln(1))}}{1 - e^{-\gamma}} = 0,
\]

\[
C_A(1, 1) = \frac{1 - e^{-(1/\gamma)\ln(e^{-\gamma})\ln(e^{-\gamma})}}{1 - e^{-\gamma}} = 1, \tag{16}\]

\[
C_A(s, 1) = \frac{1 - e^{\ln(1-t(1-e^{-\gamma}))}}{1 - e^{-\gamma}} = 1 - e^{\ln(1-s(1-e^{-\gamma}))} = s, \tag{17}\]

(b) Utility Function with Mutual Utility Independence: We now derive the utility copula, \( C_A(s, t) \), that satisfies \( C_A(s, 0) = a, s + b, C_A(0, t) = a, t + b, \) for the following utility function with mutual utility independence:

\[
U(x, y) = \frac{1 - e^{-\gamma(x+\beta y)}}{1 - e^{-\gamma(x + \beta y)}}, \quad 0 \leq x, y \leq 1. \tag{18}\]

Because \( \lambda_2 = \lambda_1 = 0 \), the utility copula provides a positive linear transformation at the minimum margins. We determine the normalized conditional utility functions at these instantiations by direct substitution into (8) to get

\[
U_i(x \mid y^0) = \frac{U(x, y^0)}{U(x^*, y^0)} = \frac{1 - e^{-\gamma y}}{1 - e^{-\gamma}} \triangleq s
\]

\[\Rightarrow x = -\frac{1}{\gamma} \ln(1 - s(1 - e^{-\gamma})). \tag{19}\]

\[
U_j(y \mid x^0) = \frac{U(x^0, y)}{U(x^0, y^*)} = \frac{1 - e^{-\gamma \beta x}}{1 - e^{-\gamma \beta}} \triangleq t
\]

\[\Rightarrow y = -\frac{1}{\gamma \beta} \ln(1 - t(1 - e^{-\gamma \beta})). \tag{20}\]

Substituting for \( x, y \) from (19) and (20) into (18) gives the multilinear utility copula

\[
C_A(s, t) = \frac{s(1 - e^{-\gamma}) + t(1 - e^{-\gamma \beta}) - st(1 - e^{-\gamma})(1 - e^{-\gamma \beta})}{1 - e^{-\gamma(1 + \beta)}},
\]

\[0 \leq s, t \leq 1. \tag{21}\]

Because the utility function (18) is strictly increasing with each of its arguments across the entire domain, we can define a utility copula at any instantiation of \( (\lambda_2, \lambda_1) \in [0, 1]^2 \). For example, we can derive the utility copula that satisfies \( C_A(s, 0.5) = a, s + b, C_A(0.2, t) = a, t + b, \) by deriving the conditional utility functions at these instantiations, i.e., \( U(x \mid y = 0.5) \) and \( U(y \mid x = 0.2) \), and substituting into (18) to get the corresponding utility copula.

Note that if an attribute is utility independent of its complement, then the normalized conditional utility function will not change with any instantiation of the complement attributes. Because (18) has mutual utility independence, both the normalized conditional utility functions, \( U(x \mid y), U(y \mid x) \), will not change across the entire domain of the utility function. This implies that (21) is the utility copula for any values of \( (\lambda_2, \lambda_1) \in [0, 1]^2 \). Consequently, we can construct the utility function (18) using (21) and normalized conditional utility assessments at any reference values of the complement attributes.

2.2. Classes 1 and 0 Utility Copulas

We now highlight two special classes of utility copulas: Class 1 utility copulas that satisfy

\[
C_1(1, \ldots, 1, v_i, 1, \ldots, 1) = a_i v_i + b_i, \quad i = 1, \ldots, n. \tag{22}\]

These utility copulas require conditional utility assessments at the maximum values of the complement attributes, \( U_i(x_i \mid x_i^*) \), \( i = 1, \ldots, n \). We also define Class 0 utility copulas that satisfy

\[
C_0(0, \ldots, 0, v_i, 0, \ldots, 0) = a_i v_i + b_i, \quad 1 \leq i \leq n, \tag{23}\]

and require conditional utility assessments at the minimum values of the complement.

To illustrate the need for utility copulas that are defined at different values of the margins, consider the multiattribute utility functions of Figure 1. Figure 1(a) shows an attribute dominance utility function that is nondecreasing with each of its arguments and strictly increasing with each argument at the maximum value of the complement. This utility function cannot be constructed using conditional utility assessments at the minimum margins because they are not strictly increasing (and not defined) at this instantiation. Refer back to (8), which requires \( U(x_i^\gamma, x_i^\gamma) \neq 0 \) for the definition of the minimum marginal utility function. We can, however, assess conditional utility functions at the maximum margins and construct this utility function using a Class 1 utility copula.

Figure 1(b), on the other hand, shows the extreme case of a multiattribute utility function that has a value equal to one if any of the attributes has a maximum value. This utility function cannot be constructed with conditional utility functions at the maximum margins (because they are not strictly increasing at this instantiation), but can be constructed using a Class 0 utility copula and utility assessments at the minimum margins. If the utility function is strictly increasing across the whole domain, then as we discussed, we can use conditional utility assessments at any reference values of the complement attributes and construct the utility function using a utility copula structure that provides a positive linear transformation at these reference values.

2.3. Relating Utility Copulas to Sklar Copulas

Sklar (1959) asserts that any continuous joint cumulative probability distribution can be expressed in terms of
Figure 1. Examples of multiattribute utility functions that are nondecreasing with each of their arguments, but strictly increasing with each argument for at least one reference value of the complement.

Note. Both these utility functions can be constructed using the utility copula formulation.

its marginal cumulative distributions and a multivariate function, \( C \), which we refer to as a Sklar copula. The continuous Sklar copula has the following properties:

(i) Sklar’s copula is a continuous mapping from \([0, 1]^n \rightarrow [0, 1]\) and is normalized such that \( C(0, \ldots, 0) = 0 \) and \( C(1, \ldots, 1) = 1 \).

(ii) Sklar’s copula satisfies the marginal property \( C(1, 1, \ldots, 1, v_i, 1, \ldots, 1) = v_i \forall i \).

(iii) Sklar’s copula is grounded, i.e., \( C(v_1, v_2, \ldots, v_{i-1}, 0, v_{i+1}, \ldots, v_n) = 0 \forall i \).

(iv) Sklar’s copula satisfies the \( n \)-increasing condition.

Property (i) is satisfied by all utility copulas. Property (ii) is a special case of the utility copula requirement, \( C(\lambda_{i,1}, \ldots, \lambda_{i,n}, v_i, \lambda_{i+1,1}, \ldots, \lambda_{i,n}) = a_i v_i + b_i \), when \( a_i = 1, b_i = 0, \lambda_{i,j} = 1 \forall i \neq j \). Property (iii) is a special instantiation of the utility copula definition because the utility copula does not need to (but can) be grounded. Property (iv) is also an instantiation of the utility copula definition because the utility copula may (or may not) satisfy the \( n \)-increasing condition. Every continuous Sklar copula is therefore a utility copula, but not every utility copula is a Sklar copula. Consequently, we can use the widely studied functional forms of Sklar copulas to generate utility copulas. We illustrate this result below.

**Proposition 2.** If \( C(v_1, \ldots, v_n) \) is a Sklar copula, then

\[
C_{\lambda}(v_1, \ldots, v_n) = a C(l_1 + (1 - l_1)v_1, \ldots, l_n + (1 - l_n)v_n) + b,
\]

where \( 0 \leq l_1 < 1, a = 1/(1 - C(l_1, \ldots, l_n)) \), \( b = 1 - a \), is a Class 1 utility copula that satisfies the \( n \)-increasing condition and

\[
C_1(1, \ldots, 1, 0, \ldots, 1, 1) = a(1 - l_i) v_i + (al_i + b),
\]

\[
i = 1, \ldots, n.
\]  

Equation (26) asserts that the utility copula obtained from (24) is not grounded unless \( l_i = 0, i = 1, \ldots, n \), in which case \( b \) is also equal to 0. We can therefore use (24) to generate ungrounded utility copulas from the widely used functional forms of Sklar (probability) copulas. With this approach, however, two properties still persist: (i) the utility copulas obtained using (24) satisfy the \( n \)-increasing condition, and (ii) they are Class 1 utility copulas requiring conditional utility assessments at the maximum values of the complement attributes. We relax these two conditions in the next section.

3. Archimedean Utility Copulas

In this section, we present some functional forms of utility copulas that relax both the grounding and \( n \)-increasing conditions, and generate either Class 1 or Class 0 utility copulas.

**3.1. Class 1. Archimedean Utility Copulas**

**Definition 2 (Extended Archimedean Functional Form).** We define the extended Archimedean functional form, \( E(v_1, \ldots, v_n) \), as

\[
E(v_1, \ldots, v_n) = a \psi^{-1} \left[ \prod_{i=1}^{n} \psi(l_i + (1 - l_i)v_i) \right] + b,
\]

where \( 0 \leq l_i < 1, a = 1/(1 - \psi^{-1} \left[ \prod_{i=1}^{n} \psi(l_i) \right]), b = 1 - a \), and the function \( \psi \) satisfies the following conditions: (i) \( \psi(v) \) is continuous on the domain \( v \in [0, 1] \); (ii) \( \psi(v) \) is strictly increasing on the domain \( v \in [0, 1] \); and (iii) \( \psi(0) = 0 \) and \( \psi(1) = 1 \).
The function $\psi$ has the same mathematical properties as either a strictly increasing cumulative probability distribution or a strictly increasing normalized utility function on the domain $[0, 1]$. With this observation, all well-known functional forms of cumulative distributions, such as Beta distributions, can be used as the function $\psi$ in (27). We now have the following proposition.

**Proposition 3.** The Archimedean functional form (27) is a Class 1 utility copula that satisfies

$$C_1(1, \ldots, 1, v_1, \ldots, v_n) = a_i v_i + b_i, \quad i = 1, \ldots, n, \tag{28}$$

where $a_i = a(1 - l_i)$ and $b_i = 1 - a_i = al_i + b$.

Proposition 3 asserts that (27) is a Class 1 utility copula. We therefore refer to the function $\psi$ as a proper Class 1 generating function. Direct substitution shows that (27) is strictly increasing with each of its arguments when $0 < l_i < 1$, $i = 1, \ldots, n$ and is grounded when $l_i = 0$, $i = 1, \ldots, n$. We now consider the sign of the mixed partial derivative of (27). To do so, we focus on differentiable generating functions.

**Corollary 1.** If the $n$th derivative of $\psi$ exists, then

$$\text{sgn} \left( \frac{\partial^n U(x_1, \ldots, x_n)}{\partial x_1 \ldots \partial x_n} \right) = \begin{cases} 0, & \text{if } \left( \prod_{i=1}^n \frac{\partial}{\partial v_i} \psi(l_i + (1 - l_i)v_i) \right) = 0, \\ \text{sgn} \left( \prod_{i=1}^n \frac{\partial}{\partial x_i} \psi(l_i + (1 - l_i)v_i) \right), & \text{otherwise,} \end{cases} \tag{29}$$

where $v_i = U_i(x_i | X_i^\lambda)$ and sgn(*) is the sign function.

Corollary 1 provides a general expression for the sign of the mixed partial derivative of an Archimedean Class 1 utility copula. Because $\psi$ can be concave, convex, or even have multiple inflection points, (27) allows for utility copulas with negative, positive, or zero mixed partial derivatives across the domain. We illustrate this result through the following example.

**Example 2 (Sign of the Mixed Partial Derivative of a Class 1 Archimedean Copula).** Consider the proper Class 1 generating function

$$\psi(t) = 1 - (1 - t)^\beta, \quad \beta > 0, \tag{30}$$

with $\psi^{-1}(t) = 1 - (1 - t)^{1/\beta}$. The two-attribute Archimedean utility copula generated by $\psi(t)$ is

$$U(x_1, x_2) = a \psi^{-1}\left( \prod_{i=1}^2 \psi(v_i') \right) + b = a(1 - (1 - v_1')^\beta + (1 - v_2')^\beta - (1 - v_1')^\beta(1 - v_2')^{1/\beta}) + b, \tag{31}$$

where $v_i' = l_i + (1 - l_i)U_i(x_i | X_i^\lambda)$. The mixed partial derivative of (31) can be calculated at any given values of $x_1, x_2$ by direct differentiation. To provide some insights into its sign, we use the results of Corollary 1. First, we consider the cases where

$$\prod_{i=1}^n \frac{\partial}{\partial v_i} \psi(l_i + (1 - l_i)v_i) = 0, \quad \frac{\partial}{\partial t} \psi(t) = \frac{e'}{\beta} (1 - e')^{1/\beta - 1}, \quad \frac{\partial^2}{\partial t^2} g(t) = \frac{e'(1 - e')^{1/\beta - 2}}{\beta} \left( 1 - \frac{e'}{\beta} \right). \tag{32}$$

We now distinguish two cases of (33): (i) when $\beta > 1$, the second partial derivative is nonnegative across the entire domain of the utility function, and (ii) when $0 < \beta < 1$, the second partial derivative changes sign from positive to negative at the inflection point $\prod_{i=1}^n \psi(v_i') = \beta$. Substituting for $\psi(t) = 1 - (1 - t)^\beta$ implies that when $0 < \beta < 1$, the inflection point occurs when $\prod_{i=1}^n \psi(v_i') = (1 - (1 - v_1')^\beta)(1 - (1 - v_2')^\beta) = \beta$, below which the sign of the second derivative is positive and above which it is negative. Figure 2 plots the second partial derivative of (33) versus $\prod_{i=1}^n \psi(v_i') = (1 - (1 - v_1')^\beta)(1 - (1 - v_2')^\beta)$ for different values of $\beta$. Note that the curves cross the horizontal axis at the value of $\beta$ (as expected) because this is the inflection point.

We now discuss how to assess the parameters $l_i$, $i = 1, \ldots, n$ of a Class 1 Archimedean copula.
Corollary 2 (Assessing the Parameters of the Class 1 Archimedean Utility Copula). The parameters \( l_i, i = 1, \ldots, n \) of the multiattribute utility function obtained using (27) satisfy
\[
a(1 - l_i) = 1 - U(\bar{x}_i, \bar{y}_i), \quad i = 1, \ldots, n. \tag{34}
\]

The following example illustrates how to construct a multiattribute utility function using (27).

Example 3 (Constructing an Extended Archimedean Utility Copula). Suppose that a decision maker faces two attributes, \( X \) and \( Y \), defined on the normalized domain \([0, 1] \times [0, 1]\). He states that his conditional utility functions, \( U(x \mid y^*) \) and \( U(y \mid x^*) \), are exponential with risk-aversion coefficients \( \gamma_x = 3 \) and \( \gamma_y = 2 \), respectively. He also provides the following utility values \( U(x^0, y^*) = 0.4 \), \( U(x^*, y^0) = 0.2 \) using indifference probability assessments. For example, the utility value \( U(x^0, y^*) = 0.4 \) asserts that he is indifferent between receiving \((x^0, y^*)\) for certain or receiving a binary deal that provides \((x^*, y^*)\) with probability 0.4 and \((x^0, y^0)\) with probability 0.6. We now consider the following proper Class 1 generating function
\[
\psi(v) = \frac{1 - e^{-dv}}{1 - e^{-\delta}}, \quad 0 \leq v \leq 1, \quad \delta \in \mathbb{R} \setminus \{0\}. \tag{35}
\]

Substituting from (35) into (27) gives the Class 1 utility copula
\[
C_A(v_x, v_y) = -\frac{a}{\delta} \ln\left(1 - \frac{(1 - e^{-\delta l_i + (1-l_i)v_y})(1 - e^{-\delta l_i + (1-l_i)v_y})}{1 - e^{-\delta}}\right) + b. \tag{36}
\]

By definition,
\[
a = \frac{1}{1 + (1/\delta) \ln(1 - (1 - e^{-\delta l_i})(1 - e^{-\delta l_i})/(1 - e^{-\delta}))}
\]
and, from Corollary 2, the parameters \( l_i \) and \( l_j \) must satisfy
\[
a(1 - l_i) = 0.6, \quad a(1 - l_j) = 0.8. \tag{37}
\]

If we use \( \delta = 1 \) in the generating function, then the solution to Equation (37) gives \( l_i = 0.54, \ l_j = 0.39 \), with \( a = 1.32 \) and \( b = -0.32 \). The multiattribute utility function and the isopreference curves obtained by this Archimedean functional form are shown in Figure 3.

Changing the value of the parameter \( \delta \) yields different multiattribute utility surfaces having the same normalized conditional utility functions at the maximum margins but provide different trade-off functions among the attributes. The value of \( \delta \) can be determined by changing its value to match the decision maker’s trade-offs. Alternatively, we can assess several points on the surface of the utility function and substitute into (36) to derive the value of \( \delta \).

If we are not able to match the two boundary values \( U(x^0, y^*), U(x^*, y^0) \) or the decision maker’s trade-off functions with a given single-parameter generating function, then we can (i) use a different generating function with more parameters, or (ii) use the same generating function and conduct a least squares estimate to determine the parameters \( l_i \) and the parameters of the generating function that best match the decision maker’s preferences.

We now consider a special case of the extended Archimedean functional form that yields the multiplicative forms of utility independence.

Corollary 3 (The Multiplicative Archimedean Class 1 Utility Copula). The multiattribute utility function obtained by the functional form (27) with generating function \( \psi(v) = v, \ 0 \leq v \leq 1 \) is the multiplicative form of mutual utility independence,
\[
U(x_1, \ldots, x_n) = \prod_{i=1}^{n} (l_i + (1-l_i)U_i(x_i \mid \bar{x}_i)) = \prod_{i=1}^{n} l_i, \tag{38}
\]
where the normalized conditional utility functions \( U_i(x_i \mid \bar{x}_i), \ i = 1, \ldots, n \) are assessed at any instantiations of the complement attributes.

Figure 3. Multiattribute utility function generated by the extended Archimedean copula.
Corollary 3 has an important implication: If we use a generating function \( \psi(v, \delta) \), where \( \delta \) is a transformation parameter, and if there exists \( \delta_0 \) such that \( \lim_{\delta \to \delta_0} \psi(v, \delta) = v \), then we can conduct a sensitivity analysis to the multiplicative utility independence assumption by changing the value of \( \delta \) and observing the change in the optimal decision. We illustrate this result through the following example.

**Example 4 (Sensitivity to Multiplicative Utility Independence).** Consider a decision situation with four attributes and a utility copula of the form (27) with generating function (35). Suppose that the analyst uses the multiplicative form of Equation (38) in his analysis instead of the utility copula form. What is the percentage of incorrect decisions that can result using this approximation?

To answer this question, we conduct a Monte Carlo simulation for a decision situation with two alternatives, each containing four attributes. We generate the probability distributions for each alternative uniformly from the space of all \( 3 \times 3 \times 3 \times 3 \) joint distributions (see the appendix for more details about this sampling approach). We then compare the expected utility using the copula form and the multiplicative form, and calculate the percentage of incorrect decisions made using the multiplicative approximation. Figure 4 shows the simulation results.

The percentage of incorrect decisions is small when \( \delta \to 0 \) (as expected because \( \lim_{\delta \to \delta_0}((1 - e^{-\delta_0})/(1 - e^{-\delta})) = v \) giving the multiplicative copula). As \( \delta \) increases, the percentage of incorrect decisions increases to about 16% for positive values \( \delta \). As \( \delta \) decreases to large negative values, the percentage of incorrect decisions is about 25%.

Figure 4 also shows the effects of the parameters \( l_i \) on the percentage of incorrect decisions when \( l_1 = l_2 = l_3 = l_4 = l \). In general, as \( l_i \) decreases (the utility function is closer to an attribute dominance utility function), the percentage error increases by making a utility independence assumption; however, this relation is not monotonic for larger values of \( \delta \), as we see by the crossover points. When \( l_i \) gets closer to one, the percentage of incorrect decisions decreases to only 8% for large positive values of \( \delta \) and to 2% for large negative values.

### 3.2. Class 0 Archimedean Utility Copulas

In some cases, (i) decision makers may be more comfortable conditioning their utility assessments on other instantiations besides the maximum values of the complement attributes. In other situations, as we discussed, (ii) the utility function may be constant at these maximum values. To provide additional flexibility in such situations, we now introduce an Archimedean functional form that requires conditional utility assessments at the minimum margins.

**Definition 4 (Scaled Archimedean Functional Form).** We define a scaled Archimedean functional form, \( S(v_1, \ldots, v_n) \), as

\[
S(v_1, \ldots, v_n) = a \eta^{-1}\left(\prod_{i=1}^{n} \eta(m_i v_i)\right), \quad 0 \leq v_i \leq 1, \quad (39)
\]

where \( 0 < m_i \leq 1; a = 1/(\eta^{-1}(\prod_{i=1}^{n} \eta(m_i))); \) and the function \( \eta(v) \) is (i) continuous, (ii) strictly decreasing on the domain \( v \in [0, 1] \) with (iii) \( \eta(0) = 1 \) and \( \eta(1) = 0 \).

Note that \( \eta(v) \) has the same mathematical properties as an excess cumulative probability distribution on the normalized domain and can be written as \( \eta(v) = 1 - \psi(v) \). We refer to the function \( \eta \) as a proper Class 0 generating function.

**Proposition 4.** The scaled Archimedean functional form (39) is a Class 0 utility copula with

\[
S(0, \ldots, 0, v_i, 0, \ldots, 0) = am_i v_i, \quad i = 1, \ldots, n, \quad (40)
\]

where the parameters \( m_i \) satisfy

\[
am_i = U(x_i^*, x_i^0), \quad i = 1, \ldots, n, \quad (41)
\]

**Figure 4.** Sensitivity to utility dependence for different values of \( l_i \).
and
\[ \frac{\partial^n U(x_1, \ldots, x_n)}{\partial x_1, \ldots, \partial x_n} \text{sgn} = \begin{cases} 0, \quad \left( \prod_{i=1}^{n} \frac{\partial}{\partial v_i} \eta(m_i v_i) \right) = 0, \\ \text{sgn} \frac{\partial^n}{\partial t^n} \eta^{-1}(e^{-t}) \bigg|_{t=-\ln(\prod_{i=1}^{n} \eta(m_i v_i))} 
\end{cases} \] (42)

Note that the assessments required for \( m_i \) are different than those needed for Class 1 utility copulas because \( X_1 \) is at its maximum value and the assessment occurs at the minimum margin.

**Example 5 (Class 0 Archimedean Utility Copulas Yielding the Multiplicative Form).** Consider the Class 0 generating function
\[ \eta(t) = (1 - t^\delta), \quad 0 \leq t \leq 1, \quad \delta > 0, \] (43)
with \( \eta^{-1}(t) = (1 - t)^{1/\delta} \). Substituting from (43) into (39) gives
\[ S(v_1, \ldots, v_n) = \eta^{-1}(\prod_{i=1}^{n} \eta(m_i v_i)) \eta^{-1}(\prod_{i=1}^{n} \eta(m_i)) = \left(1 - \prod_{i=1}^{n} (1 - m_i v_i)\right)^{1/\delta} \] (44)

Note that the functional form (44) is strictly increasing with each of its arguments when \( 0 < m_i < 1, \ i = 1, \ldots, n \), and when \( m_i = 1, \ i = 1, \ldots, n \),
\[ S(v_1, \ldots, v_n) = \left(1 - \prod_{i=1}^{n} (1 - v_i^\delta)\right)^{1/\delta} \] (45)

which is equal to one if any of the arguments is equal to one. Equation (44) also reduces to the multiplicative form when \( \delta = 1 \), giving
\[ S(v_1, \ldots, v_n) = \left(1 - \prod_{i=1}^{n} (1 - m_i v_i)\right) \] (46)

The generating function (43) thus enables sensitivity to multiplicative utility independence using conditional utility assessments at the minimum values of the complement attributes. To determine the sign of the mixed partial derivative, we first find the instantiations where \( \prod_{i=1}^{n} (\partial^2/\partial v_i \partial v_i) \eta(m_i v_i) = 0 \). Because \( \partial^2/\partial v_i \partial v_i \eta \) (for \( t = 0 \), the mixed partial derivative is zero if \( v_i = 0 \) (any of the attributes is at a minimum value). For other instantiations, the sign of the mixed partial derivative (for two attributes) is the sign of the second derivative of \( y(t) = \eta^{-1}(e^{-t}) \):
\[ y(t) = \eta^{-1}(e^{-t}) = (1 - e^{-t})^{1/\delta}, \]
\[ \frac{\partial}{\partial t} \eta^{-1}(e^{-t}) = \frac{e^{-t}}{\delta} (1 - e^{-t})^{1/\delta - 1}, \]
\[ \frac{\partial^2}{\partial t^2} \eta^{-1}(e^{-t}) = \frac{e^{-t}}{\delta} (1 - e^{-t})^{1/\delta - 2} \left[ \frac{e^{-t}}{\delta} - 1 \right]. \] (47)

Once again, we have two cases, only this time: (i) when \( \delta > 1 \), the mixed partial derivative is negative across the whole domain, and (ii) when \( 0 < \delta < 1 \), the mixed partial derivative changes sign from negative to positive at the inflection point \( \prod_{i=1}^{n} \eta(m_i v_i) = \delta \).

### 3.3. Archimedean Utility Copulas with Improper Generating Functions

We conclude our discussion of Archimedean utility copulas with the observation that other generating functions can also be used to construct utility copulas even if they are not normalized or restricted to the domain \([0, 1]\). These generating functions expand the domain of definition of the proper generating functions, and we refer to them as improper generating functions.

To illustrate an example of such functions, let \( \mu \) be a continuous and strictly decreasing function on the domain \([0, n]\), where \( n \) is the number of attributes, \( \mu(0) = 1, \mu(1) = a, 0 < a < 1 \), and \( (\mu(1))^n \geq \mu(n) \). Consider a special case of \( \mu \), where
\[ \mu(v) = e^{-v}, \quad v \in [0, n], \quad \text{and also define} \]
\[ \mu^{-1}(x) = \begin{cases} 1, \quad 0 < x < e^{-a}, \\ -\ln x, \quad e^{-a} \leq x < 1. \end{cases} \]

If we use the function, \( \mu \), with the scaled Archimedean form, we get
\[ S(v_1, \ldots, v_n) = \frac{\mu^{-1}(\prod_{i=1}^{n} \mu(m_i v_i))}{\mu^{-1}(\prod_{i=1}^{n} \mu(m_i))} = \frac{\sum_{i=1}^{n} m_i v_i}{\sum_{i=1}^{n} m_i} = \frac{w_i}{\sum_{i=1}^{n} m_i}, \]

an additive utility copula with \( w_i = m_i/\sum_{i=1}^{n} m_i, \ 0 \leq w_i \leq 1 \), and \( \sum_{i=1}^{n} w_i = 1 \).

**Example 6 (Constructing a Class 0 Utility Copula with an Improper Generating Function).** Consider the improper generating function, \( \mu(v) = e^{-v}, \delta > 0 \). Substituting into (39) gives
\[ S(v_1, \ldots, v_n) = \frac{\mu^{-1}(\prod_{i=1}^{n} \mu(m_i v_i))}{\mu^{-1}(\prod_{i=1}^{n} \mu(m_i))} = \frac{\left(\sum_{i=1}^{n} (m_i v_i)^{\delta}\right)^{1/\delta}}{\left(\sum_{i=1}^{n} (m_i)^{\delta}\right)^{1/\delta}}, \] (51)

which reduces to the additive utility copula of (50) when \( \delta = 1 \). For two attributes, \( X \) and \( Y \), the mixed partial derivative of (51) can be determined directly as
Equation (52) shows that the mixed partial derivative is zero when either \( v_i \) or \( v_j \) is zero. At other instantiations, it is positive when \( 0 < \delta < 1 \), and negative when \( \delta > 1 \). For example, when \( \delta = 2 \), (51) becomes

\[
S(v_i, v_j) = \frac{(m_x v_i^2 + m_y v_j^2)^{1/2}}{(m_x^2 + m_y^2)^{1/2}}.
\]  (53)

If the decision maker provides the assessments \( U(x^0, y^0) = 0.6 \), \( U(x^0, y^*) = 0.8 \), then from (41),

\[
\frac{m_x}{(m_x^2 + m_y^2)^{1/2}} = 0.6, \quad \frac{m_y}{(m_x^2 + m_y^2)^{1/2}} = 0.8.
\]  (54)

Solving for the values of \( m_x, m_y \) in (54) gives \( m_x = 0.3 \), \( m_y = 0.4 \). Figure 5 shows the multiattribute utility function constructed using (53) and exponential marginal utility functions at the minimum values of the complement attributes, with \( \gamma_x = 3 \), \( \gamma_y = 2 \). Note that the boundary values \( U(x, y^0) \) and \( U(x^0, y) \) are preserved in the figure.

Using the generating function \( \mu(v) = e^{-v^\gamma} \) instead of \( \mu(v) = e^{-v^\delta} \) also enables a sensitivity to additive utility independence to determine the percentage of incorrect decisions that could be made if the attributes have utility dependence. Figure 6 shows the simulation results for four attributes with \( m_i = 1, i = 1, \ldots, n \). The percentage of incorrect decisions is zero when \( \delta \to 1 \) because \( \lim_{\delta \to 1} e^{-v^\delta} = e^{-v} \). For larger values of \( \delta \), the percentage of incorrect decisions increases to about 25%. This result illustrates the need to verify additive independence before constructing an additive utility function, and also provides quantification on the percentage of incorrect decisions that could be made if additive independence is not satisfied, but is used as an approximation. Other generating functions that are monotonic, do not change sign, and have a value of unity at the origin can also be used as Class 0 generating functions and likewise for Class 1 generating functions if they pass by \((1, 1)\).

4. Linear and Composite Transformations of Utility Copulas

In some situations, it may be convenient to group the attributes of a decision problem into different categories before constructing the utility function. For example, Keeney (1979) analyzes a decision of evaluating pumped storage facilities. He starts with several attributes, including: economic and socioeconomic effects, public health, safety, and environmental effects. He then reformulates the attributes into four categories and observes that decision makers were comfortable asserting mutual utility independence among the different categories. In this section, we illustrate how linear and composite transformations of utility copulas can group certain subsets of the attributes into different categories and can provide mutual utility independence among them, or can incorporate utility independence for only a subset of the attributes and have utility dependence for the rest.

4.1. Partial Utility Independence with Utility Copula Formulations

Consider the functional form

\[
C_\lambda(v_1, \ldots, v_{k-1}, v_k, \ldots, v_n)
= a \sum_{i=1}^k w_i v_i + b C_\lambda(v_{k+1}, \ldots, v_n),
\]  (55)

where \( a, b, w_i > 0 \), \( a + b = 1 \), \( \sum_{i=1}^k w_i = 1 \).

The function \( C_\lambda(v_1, \ldots, v_{k-1}, v_k, \ldots, v_n) \) is a utility copula defined at (i) any reference values for attributes \( v_i \), \( i = 1, \ldots, k \), and (ii) the reference values of \( C_\lambda(v_{k+1}, \ldots, v_n) \) for attributes \( v_{k+1}, \ldots, v_n \). Equation (55) provides partial utility independence among the attributes: attributes \( v_i, i = 1, \ldots, k \) have additive utility independence.
and attributes $v_j$, $i = k + 1, \ldots, n$ have a utility dependence structure determined by $C_{\lambda_i}(v_{k+1}, \ldots, v_n)$. The term $C_{\lambda_i}(v_{k+1}, \ldots, v_n)$ groups the set of attributes $v_j$, $i = k + 1, \ldots, n$ into a single category and provides the flexibility of allowing more general trade-off functions among its attributes than can be modeled by the additive form.

Similar analysis also applies to the product form

$$C_{\lambda_i}(v_1, \ldots, v_{k-1}, v_k, \ldots, v_n) = C_{\lambda_{k+i}}(v_{k+1}, \ldots, v_n) \prod_{i=1}^{k} v_k,$$  \hspace{1cm} (56)

which provides multiplicative utility independence for attributes $v_j$, $i = 1, \ldots, k$ (they are also utility dominant attributes). $C_{\lambda_i}(v_1, \ldots, v_{k-1}, v_k, \ldots, v_n)$ is also a utility copula. The reference values for attributes $v_{k+1}, \ldots, v_n$ are those determined by $C_{\lambda_{k+i}}$, whereas any instantiations of the attributes $v_j$, $i = 1, \ldots, k$ can be used as reference values for the conditional utility assessments.

If we wish to group the attributes into more than one category, and have mutual utility independence among the different categories, then we can construct a utility copula using additive or multiplicative combinations of lower-order utility copulas, such as

$$C_{\lambda}(v_1, \ldots, v_{k-1}, v_k, \ldots, v_n) = w_1 C_{\lambda_1}(v_1, \ldots, v_k) + w_2 C_{\lambda_{k+i}}(v_{k+1}, \ldots, v_n),$$  \hspace{1cm} (57)

or

$$C_{\lambda}(v_1, \ldots, v_{k-1}, v_k, \ldots, v_n) = C_{\lambda_{k+i}}(v_k, \ldots, v_n) C_{\lambda_{k+i}}(v_{k+1}, \ldots, v_n).$$  \hspace{1cm} (58)

The utility copulas (57) and (58) provide mutual utility independence of the set $v_j$, $i = k + 1, \ldots, n$ from the set $v_j$, $i = 1, \ldots, k$, but allow for utility dependence among the attributes of each set. The additive and multiplicative combinations of (57) and (58) also lead to two natural extensions: (i) composite transformations of utility copulas to provide utility dependence among the different categories, and (ii) nonlinear combinations of general utility copulas.

### 4.2. Composite Transformations on Utility Copulas

Consider the composite transformation $C_{\lambda}(C_{\lambda_{12}}(v_1, v_2), v_3)$. This functional form groups the first two attributes, $v_1, v_2$, together and then combines them with a third attribute $v_3$. Special cases of this composite transformation lead to the product form $v_3 C_{\lambda_{12}}(v_1, v_2)$ or to the additive form $v_3 + C_{\lambda_{12}}(v_1, v_2)$ depending on the type of utility copula $C_{\lambda}$, but other cases also incorporate dependence among the attributes $v_1, v_2$ and the attribute $v_3$. This formulation enables a sensitivity analysis to the utility independence of attribute $v_3$ on the set of attributes $v_1, v_2$. Using composite transformations also enables more general trade-off functions between $v_3$ and the set $C_{\lambda_{12}}(v_1, v_2)$ than can be determined by the simpler additive and multiplicative combinations.

Similar analysis also applies to composite transformations of the different categories to provide utility copulas of the form $C_{\lambda}(C_{\lambda_{12}}(v_1, v_2), C_{\lambda_{34}}(v_3, v_4))$, which could lead to the product of utility copulas, $C_{\lambda}(C_{\lambda_{12}}(v_1, v_2), C_{\lambda_{34}}(v_3, v_4))$; to their summation, $C_{\lambda_{12}}(v_1, v_2) + C_{\lambda_{34}}(v_3, v_4)$, or to more general trade-off functions between the two subsets $C_{\lambda_{12}}(v_1, v_2)$ and $C_{\lambda_{34}}(v_3, v_4)$.

### 4.3. Linear Combinations of Utility Copulas: Yielding the Multilinear Form

We have discussed utility copulas whose independence conditions yield additive and multiplicative forms of utility functions. This section derives utility copulas whose independence condition yields the multilinear form. This formulation provides (i) a general framework for constructing utility functions with partial utility independence, and (ii) enables sensitivity analysis to the independence assumption that every attribute is utility independent of its complement. To illustrate, consider the linear combination shown below:

$$C_{\lambda}(v_1, \ldots, v_n) = \sum_{i=1}^{n} k_i v_i + \sum_{i=1}^{n} \sum_{j=1}^{i-1} k_{ij} C_{\lambda_{ij}}(v_i, v_j)$$

$$+ \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=1}^{j-1} k_{ijk} C_{\lambda_{ijk}}(v_i, v_j, v_k) + \cdots + k_{123\ldots n} C_{\lambda_{123\ldots n}}(v_1, v_2, \ldots, v_n), \hspace{1cm} (59)$$

where the scaling constants satisfy (Keeney and Raiffa 1976)

$$k_{xj} = U(x_i^x, x_j^x), \hspace{1cm} i = 1, \ldots, n,$$

$$k_{xj} = U(x_i^x, x_j^x) - k_{xj} - k_{xj}, \hspace{1cm} i = 1, \ldots, n, j > i,$$

and the utility copula functions in (59) all require conditional utility assessments at the same reference values, i.e., any argument $v_i$ corresponds to the same conditional utility assessment, $U_i(x_i, x_i^x)$.

Equation (59) yields the multilinear form of utility independence as a special case when each term on the right-hand side is the Class 1 Archimedean form $C(v_1, \ldots, v_n) = \psi^{-1}(\sum_{i=1}^{n} \psi(v_i))$. As we now show, other classes of utility copulas that are not necessarily Class 1 can also be used with (59) to derive more general functional forms. We also discuss two special cases: (i) convex combinations of utility copulas (with positive affine scaling constants), and (ii) linear combinations of utility copulas (with possibly negative scaling constants).
Proposition 5. The linear combination (59) of utility copulas requiring (i) conditional utility assessments $U_i(x_i | \tilde{x}_i^a)$ at the reference values, $\tilde{x}_i^a$, $i = 1, \ldots, n$; that (ii) are differentiable with each argument; and (iii) satisfy the set of inequalities,

$$k_i + \frac{\partial}{\partial v_j} \sum_{j \neq i} k_{j,i} C(v_i, v_j)$$

$$+ \cdots + \frac{\partial}{\partial v_j} k_{1,23,n} C(v_1, v_2, \ldots, v_n) > 0,$$

$$i = 1, \ldots, n,$$  \hspace{1cm} (61)

is a utility copula, $C_\lambda$, requiring conditional utility assessments at the same reference values.

The following proposition presents a special case of the above when the scaling constants are all positive and sum to one.

Proposition 6. The convex combination of utility copulas requiring conditional utility assessments at the reference values, $U_i(x_i | \tilde{x}_i^a)$, $i = 1, \ldots, n$ is a utility copula, $C_\lambda$, requiring conditional utility assessments at the same reference values.

Propositions 5 and 6 illustrate how we can derive a large family of utility copulas whose independence condition yields the multilinear form. The multilinear combination of utility copulas also provides a general framework for incorporating partial utility independence among the attributes by appropriate choice of the scaling constants. For example, the utility copula formed by a convex combination,

$$C_\lambda(v_1, v_2, v_3) = k_1 v_1 + k_2 v_2 + k_3 C_{\lambda,3}(v_2, v_3),$$  \hspace{1cm} (62)

is a special case of (59) and asserts that $v_1$ is utility independent of its complement, $v_2$ is utility independent of $v_1$ and utility dependent on $v_3$, whereas $v_3$ is utility independent on $v_1$ and utility dependent on $v_2$. If $C_{\lambda,3}$ is equal to a product copula, then (62) reduces to

$$C_\lambda(v_1, v_2, v_3) = k_1 v_1 + k_2 v_2 + k_3 v_2 v_3,$$  \hspace{1cm} (63)

which asserts mutual utility independence among the attributes.

5. Risk-Aversion Functions Induced by Utility Copulas

When constructing a multiattribute utility function, the decision analyst may also wish to consider other properties besides the utility independence relations or the multivariate risk aversion. In particular, he may be interested in the conditional risk-aversion function (Arrow 1965 and Pratt 1964) for each individual attribute and its variation across the domain of the utility function. In this section, we discuss the risk-aversion properties induced by utility copula formulations. To illustrate, suppose that we have assessed the conditional utility function for an attribute at some reference values, $\tilde{x}_i^a$, of the complement attributes. If this conditional utility function is twice continuously differentiable, then we can derive the conditional risk aversion function at this instantiation by

$$\gamma_i(x_i, \tilde{x}_i^a) = -\frac{\partial^2 U_i(x_i | \tilde{x}_i^a)}{\partial x_i^2} \frac{\partial U_i(x_i | \tilde{x}_i^a)}{\partial x_i}.$$  \hspace{1cm} (64)

If this attribute $X_i$ is utility independent of its complement, then

$$U(x_i, \tilde{x}_i^a) = l(\tilde{x}_i^a) U_i(x_i | \tilde{x}_i^a) + g(\tilde{x}_i^a)$$  \hspace{1cm} (65)

for some functions $l$ and $g$, and an arbitrary instantiation of the complement $\tilde{x}_i^a$.

Equation (65) implies that the risk-aversion function, $\gamma_i(x_i, \tilde{x}_i^a)$, does not depend on the instantiation $\tilde{x}_i^a$. If $X_i$ is not utility independent of its complement, however, then the risk-aversion function changes across the domain of the utility function. A natural question that poses itself now is the amount by which the conditional risk-aversion function $\gamma_i(x_i, \tilde{x}_i^a)$ changes at any instantiation $\tilde{x}_i$ relative to its value at the assessed reference point $\tilde{x}_i^a$. Before we answer this question, we make the following definition.

Definition (Copula Risk-Aversion Function). We define the copula risk-aversion function with respect to argument $v_i$ as

$$\gamma_C(v_1, \ldots, v_n) = -\frac{\partial^2 C_\lambda(v_1, \ldots, v_n)}{\partial v_i^2} \frac{\partial C_\lambda(v_1, \ldots, v_n)}{\partial v_i}.$$  \hspace{1cm} (66)

By the monotonicity condition, the first partial derivative $\partial C_\lambda(v_1, \ldots, v_n)/\partial v_i$ is nonnegative and so the sign of the copula risk aversion depends only on the sign of $\partial^2 C_\lambda(v_1, \ldots, v_n)/\partial v_i^2$. If the copula is a concave function of $v_j$, then the copula risk aversion for attribute $v_j$ is positive, and if it is convex, then the copula risk aversion is negative. If the utility copula is a linear function of $v_i$, then the copula risk aversion $\gamma_C$ is zero. Because $C_\lambda(v_1, \ldots, v_n)$ can also have multiple inflection points with $v_i$, the copula risk aversion may change sign across the domain.

We now observe that for each attribute, $X_i$, the utility copula function is in fact a transformation applied to the conditional utility function $U_i(x_i | \tilde{x}_i^a)$. With this observation, we can use the results of Dyer and Sarin (1982) and Matheson and Abbas (2005) to relate the risk aversion of an attribute at the assessed reference value of the complement, $\tilde{x}_i^a$, to the risk aversion at any other instantiation of the complement, $\bar{x}_i$, using the chain rule for risk aversion as

$$\gamma_i(x_i, \bar{x}_i) - \gamma_i(x_i, \tilde{x}_i^a) = \gamma_C(U_i(x_i | \tilde{x}_i^a), \ldots, U_n(x_n | \tilde{x}_n^a)) \frac{\partial}{\partial x_i} U_i(x_i | \tilde{x}_i^a).$$  \hspace{1cm} (67)
As a simple application of this result, suppose that we have assessed the conditional utility function of an attribute at the minimum values of its complement and determined the conditional risk-aversion function, \( \gamma(x_i, x_j^*) \). We now wish to determine the increment in this conditional risk aversion as we move to any other instantiation of its complement. Equation (67) shows that this increment is equal to the product of the copula risk aversion of \( x_i \) at the new values of the complement and the first partial derivative of the conditional utility function at the assessed reference values.

If we would like the assessed risk aversion to decrease at any other reference values of the complement, then the utility copula should be a convex function of that attribute (have negative copula risk aversion) at these other reference values. Moreover, if we would like the risk aversion to decrease at some reference values and increase at others, then the utility copula should be a concave function of \( x_i \) at some instantiations and be a concave function at others (for example, the copula function may be an S-shaped function of that attribute across the domain of the utility function). Finally, if we would like the risk-aversion function of that attribute to remain constant across any instantiations of the complement, then the copula risk aversion for that attribute must be zero (the utility copula must be a linear function of this argument). Examples of utility copulas satisfying this latter condition for attribute \( v_3 \) are the copulas \( v_3 C_{\lambda_3}(v_1, v_2), k v_3 + (1 - k) C_{\lambda_2}(v_1, v_2) \), discussed above, where \( C_{\lambda_2}(v_1, v_2) \) is a utility copula.

6. Conclusions

We introduced the notion of a multiattribute utility copula and highlighted several of its implications: (i) single-attribute conditional utility assessments that are commonly used in many of the utility independence formulations can be used to construct more general functional forms of utility functions that incorporate utility dependence. The use of a utility copula structure provides an additional degree of freedom over the independence forms and maintains the same normalized conditional utility functions at the assessed reference values. (ii) The utility copula formulation enables sensitivity to utility independence and characterizes the percentage of incorrect decisions that could be made in a given problem by making a utility independence approximation if the attributes have utility dependence. We illustrated that there are in fact situations where the assumptions of utility independence could provide reasonable approximations in a given problem, but there are others where the percentage of incorrect decisions could be large. (iii) The Archimedean functional forms provide a simple mechanism for generating a large family of utility copulas that can be used to incorporate utility dependence using a single generating function. (iv) Composite and linear transformations of utility copulas ensure the construction of an even larger family of utility copulas that incorporate partial utility independence among the attributes and enable sensitivity to the multilinear form.

The notion of multiattribute utility copulas leads to several new directions for future research—for example, the development of new families of utility copulas that match the conditional utility assessments at other reference values besides the minimum and maximum values of the complement attributes. Future work can also derive utility copulas that relax the nondecreasing condition from the utility copula definition and derive more general multiattribute utility functions using single-attribute assessments. Measures of utility dependence can also be developed to quantify the strength of utility dependence among the attributes. Future work can also focus on the synthesis of copula functions with specified risk-aversion functions using composite and linear transformations of utility copulas. Future work can also explore more general sensitivity analyses to investigate the effects of relaxing partial utility independence conditions on the recommended decision alternative. In this paper, we used a uniform sampling approach to generate the probability distributions for the simulation results. Other sampling mechanisms can also be used and tailored to the probability dependence structure of the given problem (see, for example, Keefer 2004 and Abbas 2006 for two other sampling approaches that incorporate problem)

A behavioral research perspective also leads to future work on comparisons of the conditional utility assessments at different reference values in terms of ease and comfort level of the decision maker. Future work can also focus on methods to assess utility dependence by direct assessment of the utility dependence parameters among the attributes. Finally, we look forward to future work on practical applications of multiattribute utility copulas to help with a wide variety of applications in decisions with multiple attributes.

Appendix

**Lemma 1.** The function \( \psi^{-1}[k\psi(v)], 0 < k \leq 1 \), where \( \psi \) is a strictly monotone function, is a strictly increasing function of \( v \).

**Proof.** First, we show that if \( \psi \) is strictly increasing, then \( \psi^{-1} \) is also a strictly increasing function. Consider two values \( v_1, v_2 \) with \( 0 < v_1 < v_2 < 1 \), and let \( \psi(v_1) = t_1, \psi(v_2) = t_2 \). If \( \psi \) is strictly increasing, then \( t_2 > t_1 > \psi(0) \). Because \( v_2 > v_1 \) and \( t_2 > t_1 \), and because \( v_2 = \psi^{-1}(t_2) \) and \( v_1 = \psi^{-1}(t_1) \), then \( \psi^{-1} \) is also a strictly increasing function. Finally, because \( \psi \) is strictly increasing, then \( \delta\psi(v) \), \( \delta > 0 \) is also strictly increasing. The composite transformation of strictly increasing functions \( \psi^{-1}[\delta\psi(v)] \) is, thus, strictly increasing. The proof for \( \psi \) being a strictly decreasing function follows identical steps. Q.E.D.

**Proof of Proposition 1.** Because the utility function \( U(x_1, \ldots, x_n) \) is bounded, it can be normalized into a strategically equivalent utility function that ranges from zero to one. Furthermore, because it is continuous and strictly increasing with each argument, \( x_i \), at an instantiation \( \bar{x}_i \), then the normalized conditional utility functions \( v_i = \hat{U}(x_i \mid \bar{x}_i^k), i = 1, \ldots, n \) must be continuous and
strictly increasing. This result also implies (see Lemma 1) that the inverse functions, \( x_i = U_i^{-1}(v_i \mid \tilde{x}_i) \), \( i = 1, \ldots, n \), are strictly increasing. Define the function
\[
C_a(v_1, \ldots, v_n) = U(U_i^{-1}(v_i \mid \tilde{x}_i), \ldots, U_n^{-1}(v_n \mid \tilde{x}_n)). \tag{68}
\]
Substitute for \( v_i = U_i(x_i \mid \tilde{x}_i) \) into the left-hand side and \( x_i = U_i^{-1}(v_i \mid \tilde{x}_i), i = 1, \ldots, n \) into the right-hand side of (68),
\[
C_a(U_i(x_i \mid \tilde{x}_i), \ldots, U_n(x_n \mid \tilde{x}_n)) = U(x_1, \ldots, x_n). \tag{69}
\]
The function \( C_a(v_1, \ldots, v_n) = U(U_i^{-1}(v_i \mid \tilde{x}_i), \ldots, U_n^{-1}(v_n \mid \tilde{x}_n)) \) satisfies the following properties:

(i) Normalized range and domain: Because \( U(x_i \mid \tilde{x}_i) \) and the conditional utility functions \( v_i = U_i(x_i \mid \tilde{x}_i) \), \( i = 1, \ldots, n \) are normalized to range from zero to one, then from (69), \( C_a : [0, 1]^n \rightarrow [0, 1] \). Furthermore, because \( U_i^{-1}(v_i = 0 \mid \tilde{x}_i) = \tilde{x}_i, U_i^{-1}(v_i = 1 \mid \tilde{x}_i) = x_i \), then
\[
C_a(0, \ldots, 0) = U(U_i^{-1}(0 \mid \tilde{x}_i), \ldots, U_n^{-1}(0 \mid \tilde{x}_n)) = U(x_1^0, \ldots, x_n^0) = 0. \tag{70}
\]
\[
C_a(1, \ldots, 1) = U(U_i^{-1}(1 \mid \tilde{x}_i), \ldots, U_n^{-1}(1 \mid \tilde{x}_n)) = U(x_1^1, \ldots, x_n^1) = 1. \tag{71}
\]

(ii) Monotonicity: Because \( U_i(x_i \mid \tilde{x}_i) \) is continuous and strictly increasing with \( x_i \), there is a one-to-one monotonic mapping from \( x_i \) to \( v_i \). Equation (69) asserts that \( C_a(v_1, \ldots, v_n) \) is strictly increasing (nondecreasing) with \( v_i \) whenever \( U(x_i, \ldots, x_n) \) is strictly increasing (nondecreasing) with \( x_i \). The regularity conditions on the utility function guarantee that it is nondecreasing, and for each attribute there exists at least one instantiation of the complement for which it is strictly increasing. The function \( C_a(v_1, \ldots, v_n) \) must exhibit this same property.

(iii) Positive linear transformation: Consider the instantiation \( U(x, \tilde{x}) \) for which the utility function is strictly increasing with \( x \). Let the value of each node in the complement be \( x^0 \) at this instantiation. At this instantiation, (68) asserts
\[
C(\lambda_{i,1}, \ldots, \lambda_{i,i-1}, U_i(x_i \mid \tilde{x}_i), \lambda_{i,i+1}, \ldots, \lambda_{i,n}) = U(x_i, \tilde{x}_i), \tag{72}
\]
where \( \lambda_{i,j} = U_i(x^0_j \mid \tilde{x}_i) \). Substitute for the definition of \( U_i(x_i \mid \tilde{x}_i) \) into the right-hand side of (72) to get
\[
C(\lambda_{i,1}, \ldots, \lambda_{i,i-1}, U_i(x_i \mid \tilde{x}_i), \lambda_{i,i+1}, \ldots, \lambda_{i,n}) = \left[ U(x_i^0, \tilde{x}_i) - U(x_i^0, \tilde{x}_i) + U(x_i^0, \tilde{x}_i) \right] U_i(x_i \mid \tilde{x}_i) = U(x_i, \tilde{x}_i) \tag{73}
\]
Define \( a_i = [U(x_i^0, \tilde{x}_i) - U(x_i^0, \tilde{x}_i)] = U(x_i^0, \tilde{x}_i), v_i = U_i(x_i \mid \tilde{x}_i), \) and substitute into (73)
\[
C(\lambda_{i,1}, \ldots, \lambda_{i,i-1}, v_i, \lambda_{i,i+1}, \ldots, \lambda_{i,n}) = a_i v_i + b_i. \tag{74}
\]
Because \( U(x_i, \tilde{x}_i) \) is strictly increasing with \( x_i \), then \( a_i = [U(x_i^0, \tilde{x}_i) - U(x_i^0, \tilde{x}_i)] > 0 \). Furthermore, because the utility function is normalized, both \( a_i, b_i \in [0, 1] \). The same applies to all attributes. From (i), (ii), and (iii), \( C_a(v_1, \ldots, v_n) \) is a utility copula. Q.E.D.

**Proof of Proposition 2** (Relating Sklar Copulas to Utility Copulas). By definition, every continuous Sklar copula \( C(v_1, \ldots, v_n) \) is nondecreasing with each of its arguments; is a Class 1 utility copula that satisfies \( C(1, \ldots, 1, v_i, 1, \ldots, 1) = v_i, i = 1, \ldots, n \) and the \( n \)-increasing condition. These conditions assert that:

(i) \( C_a \) is nondecreasing with each of its arguments and strictly increasing with each argument at the maximum values of the complement attributes.

(ii) Normalization: Because \( 0 \leq l_i < 1, 0 \leq v_i \leq 1 \), then \( 0 \leq (l_i + (1 - l_i))v_i \leq 1 \). The utility copula is defined on a normalized domain. Furthermore, \( C_a(1, \ldots, 1) = aC(1, \ldots, 1) + b = a + b = 1 \) and \( C_a(0, \ldots, 0) = aC(1, \ldots, 1) + b = (C(l_1, \ldots, l_n) - C(l_1, \ldots, l_n))/(1 - C(l_1, \ldots, l_n)) = 0 \). Thus, \( C_a : [0, 1]^n \rightarrow [0, 1] \).

(iii) Positive linear transformation: Because, \( C(1, \ldots, 1, v_i, 1, \ldots, 1) = v_i, i = 1, \ldots, n \), then by definition,
\[
C_a(1, \ldots, v_i, 1, \ldots, 1) = aC(1, \ldots, 1, l_i + (1 - l_i)v_i, 1, \ldots, 1) + b = a(l_i + (1 - l_i)v_i) + b = a_v_i + b_i, \tag{75}
\]
where \( a_i = a(1 - l_i)v_i, b_i = b + a_l_i \). From (75), \( C_a(1, \ldots, 1, 0, 1, \ldots, 1) = l_i = U(x_i^0, \ldots, x_i^0, x_{i+1}^0, \ldots, x_n^0), i \leq 1 \).

(iv) To show that it satisfies the \( n \)-increasing condition (we demonstrate below only the case of differentiability) and observe that the sign of the mixed partial derivative of \( C(v_1, \ldots, v_n) \) and \( C(l_1 + (1 - l_1)v_1, \ldots, l_n + (1 - l_n)v_n) \) is the same when \( 0 \leq l_i \leq 1, i = 1, \ldots, n \).

Proof of Proposition 3. First, we observe that the continuity of \( \psi \) guarantees the continuity of (27).

(i) Monotonicity: Let \( \delta = \prod_{i=1}^{n} \psi(l_i + (1 - l_i)v_i) \). From Lemma 1, if \( \delta > 0 \), then (27) is strictly increasing with \( v_i \) on the whole domain. If \( \delta = 0 \), then the copula is equal to zero, which is a nondecreasing function with \( v_i \) (this occurs when at least one of the attributes, \( X_j \), has \( l_j = 0 \) and we consider the value of the utility copula at \( x_j^0 \)). In this case, or even when \( l_j = 0, j = 1, \ldots, n \), Equation (27) becomes \( \psi^{-1}[[\prod_{i=1}^{n} \psi(v_i))] \), which is strictly increasing with each argument at the maximum value of the complement arguments. Therefore, for each argument, there exists an instantiation of the complement attributes for which the utility function is strictly increasing with this argument.

(ii) Normalization: Because \( 0 \leq v_i \leq 1 \), then (27) is defined on a normalized domain \([0, 1]^n \). Furthermore, direct
substitution shows that
\[
E(0, \ldots, 0) = \left\{ a\psi^{-1}\left[ \prod_{i=1}^{n} \psi(l_i) \right] + b \right\} = \frac{\psi^{-1}\left[ \prod_{i=1}^{n} \psi(l_i) \right] - \psi^{-1}\left[ \prod_{i=1}^{n} \psi(l_i) \right]}{1 - \psi^{-1}\left[ \prod_{i=1}^{n} \psi(l_i) \right]} = 0, \quad (76)
\]
\[
E(1, \ldots, 1) = \left\{ a\psi^{-1}\left[ \prod_{i=1}^{n} \psi(1) \right] + b \right\} = \frac{1 - \psi^{-1}\left[ \prod_{i=1}^{n} \psi(l_i) \right]}{1 - \psi^{-1}\left[ \prod_{i=1}^{n} \psi(l_i) \right]} = 1. \quad (77)
\]
From (i) and (ii), C: [0, 1]^n \rightarrow [0, 1].
(iii) A Class 1 utility copula requires a positive linear transformation at the reference values. Indeed,
\[
E(1, \ldots, 1, v_i, \ldots, 1) = a\psi^{-1}\left( \psi(l_i + (1 - l_i)v_i) \prod_{j \neq i} \psi(1) \right) + b = a(1 - l_i)v_i + b + al_i, \quad i = 1, \ldots, n. \quad \text{Q.E.D.}
\]

**Proof of Corollary 1.** It will be convenient to define \( \phi(t) = \ln \psi(t) \) and express the utility copula as
\[
U(x_1, \ldots, x_n) = a\phi^{-1}\left( \prod_{i=1}^{n} \phi(1 + (1 - l_i)x_i) \right) + b.
\]
Using the chain rule gives
\[
\frac{\partial^n}{\partial x_1 \ldots \partial x_n} U(x_1, \ldots, x_n)
= a\frac{\partial^n}{\partial t^n} \phi^{-1}(t) \left|_{t = \sum_{i=1}^{n} \phi(1 + (1 - l_i)x_i)} \right. 
= a \prod_{i=1}^{n} \frac{\partial}{\partial x_i} \phi(1 + (1 - l_i)x_i) \prod_{i=1}^{n} \frac{\partial}{\partial x_i} U(x_i | x_i^h). \quad (78)
\]

The term \( \prod_{i=1}^{n} \frac{\partial}{\partial x_i} U(x_i | x_i^h) \) on the right-hand side of (78) is positive by construct, whereas the term \( (\prod_{i=1}^{n} \frac{\partial}{\partial x_i} \phi(1 + (1 - l_i)x_i)) \) is either zero or positive. For example, if \( \phi(t) \) has an inflection point, then \( (\partial/\partial t) \phi(t) = 0 \) at that particular value. If \( (\prod_{i=1}^{n} \frac{\partial}{\partial x_i} \phi(1 + (1 - l_i)x_i)) = 0 \), then the sign of the mixed partial derivative is zero. Note that \( (\partial/\partial t) \phi(t) \) and \( (\partial/\partial t) \phi(t) \) have the same sign, and so this condition is also achieved when \( (\prod_{i=1}^{n} \frac{\partial}{\partial x_i} \phi(1 + (1 - l_i)x_i)) = 0 \). On the other hand, if \( (\prod_{i=1}^{n} \frac{\partial}{\partial x_i} \phi(1 + (1 - l_i)x_i)) > 0 \), then the sign of the mixed partial derivative is determined by
\[
\text{sgn} \frac{\partial^n}{\partial x_1 \ldots \partial x_n} U(x_1, \ldots, x_n)
= \text{sgn} \frac{\partial^n}{\partial t^n} \phi^{-1}(t) \left|_{t = \sum_{i=1}^{n} \phi(1 + (1 - l_i)x_i)} \right. 
= \text{sgn} \frac{\partial^n}{\partial t^n} \phi^{-1}(t) \left|_{t = \ln \prod_{i=1}^{n} \phi(1 + (1 - l_i)x_i)} \right. 
= \text{sgn} \frac{\partial^n}{\partial t^n} \phi^{-1}(t) \left|_{t = \ln \prod_{i=1}^{n} \phi(1 + (1 - l_i)x_i)} \right. 
= \text{sgn} \frac{\partial^n}{\partial t^n} \phi^{-1}(t) \left|_{t = \ln \prod_{i=1}^{n} \phi(1 + (1 - l_i)x_i)} \right. 
\]

**Proof of Corollary 2 (Assessing the Parameters).** From Proposition 1, setting \( v_i = 0, v_j = 1, j \neq i \), implies
\[
E(v_1 = 1, \ldots, v_{i-1} = 1, v_j = 0, v_{i+1} = 1, \ldots, v_n = 1)
= U(x_i^0 \hat{x}_i^*).
\]
Setting \( v_i = 0, v_j = 1, j \neq i \) into (27) gives
\[
E(v_1 = 1, \ldots, v_{i-1} = 1, v_j = 0, v_{i+1} = 1, \ldots, v_n = 1)
= al_i + b. \quad (81)
\]
Observing that \( b = 1 - a \), and equating (80) and (81) gives
\[
a(1 - l_i) = 1 - U(x_i^0, \hat{x}_i^*), \quad 1 \leq i \leq n. \quad \text{Q.E.D.} \quad (82)
\]

**Proof of Corollary 3.** By direct substitution of \( \psi(v) = v \) into (27), we get
\[
C_p(v_1, \ldots, v_n) = a\left[ \prod_{i=1}^{n} (1 + (1 - l_i)v_i) \right] + b,
\]
\[
a = 1 / \left( 1 - \prod_{i=1}^{n} l_i \right), \quad b = 1 - a. \quad (83)
\]

**Steps of Monte Carlo Simulation**

1. Uniform sampling from the space of \( 3 \times 3 \times 3 \times 3 \) probability distributions. Generate two \( 3 \times 3 \times 3 \times 3 \) joint probability distributions to represent two decision alternatives (each having four variables discretized to three outcomes), by uniform sampling as follows:
   - Generate \( 3^4 - 1 \) independent samples, \( x_1, x_2, \ldots, x_{3^4 - 1} \), from a uniform \([0, 1]\) distribution.
   - Sort the generated samples from highest to lowest to form an order statistic, \( u_3 \leq u_2 \leq \cdots \leq u_{3^4 - 1} \).
   - Take the difference between each two successive elements of the order statistic \( u_1 - u_2 = u_2 - u_1, \ldots, u_{3^4 - 1} - u_{3^4 - 2}, 1 - u_{3^4 - 1} \).
   - The increments form a \( 3^4 \)-outcome probability distribution that is uniformly sampled from the space of possible \( 3^4 \)-outcome probability distributions.

2. Generate four marginal utility values for each consequence in the tree from a uniform \([0, 1]\) distribution. Because each consequence is characterized by four attributes, we generate four normalized utility values for each consequence.

3. Evaluate the multiattribute utility function for each prospect using (i) the multiplicative form of multiattribute utility functions \( x_i = 0, i = 1, \ldots, 4 \), and (ii) the given utility copula form with \( l_i = 0, i = 1, \ldots, 4 \).

4. Calculate the expected multiattribute utility of both alternatives using (i) the product form, and (ii) the copula form.

5. Repeat 10,000 times.

6. Calculate the fraction of times a difference in the recommended decision alternatives occurs if we assume the product form when the actual utility function has a copula form.
(7) Change the value of the dependence parameter, $\delta$, in the copula form and repeat the simulation steps.

(8) Repeat for different values of $I_j$.

**Proof of Proposition 4.** The continuity of $\eta$ guarantees the continuity of the utility copula.

(i) Monotonicity: Using similar analysis as that conducted in Lemma 1, we can show that both $\eta$ and $\eta^{-1}$ are strictly decreasing and that $\eta^{-1}(\delta_i \eta(m_i))$ is a strictly increasing function of $m_i$ when $0 < \delta_i \leq 1$. Let $\delta_i = \prod_{j \neq i} \eta(m_j, v_j)$. When $0 < m_i < 1$, $i = 1, \ldots, n$, $0 < \delta_i \leq 1$, and (39) is strictly increasing with each of its arguments. If any $m_i = 1$, then the function is nondecreasing with any of its arguments at the instantiation $v_i = 1$ (because $\delta_i = 0$), but even in this case, or the case where $m_i = 1$, $j = 1, \ldots, n$, (39) reduces to $\eta^{-1}(\prod_{j \neq i} \eta(v_j))$, which is strictly increasing with any argument $v_i$ when the remaining attributes are at their minimum values. Therefore, the monotonicity condition is satisfied.

(ii) Normalization: $S(0, \ldots, 0) = a \eta^{-1}(\prod_{i=1}^n \eta(0)) = a \eta^{-1}(1) = 0$. $S(1, \ldots, 1) = a \eta^{-1}(\prod_{i=1}^n \eta(m_i)) = 1$. Furthermore, because $0 \leq v_i \leq 1$ and $0 \leq m_i \leq 1$, then $0 \leq m_i \leq 1$, and from (i) and (ii), $C_{\eta}: [0, 1]^n \to [0, 1]$.

(iii) When $v_j = 0, j \neq i$, we have $C_{\eta}(0, \ldots, 0, v_j, 0, \ldots, 0) = \eta^{-1}(\eta(m_i, v_j)) = am_i v_j$; hence, $a_i = am_i$, $b = 0$, and (39) is a Class 0 utility copula. Furthermore, $C_{\eta}(0, 0, \ldots, 1, 0, \ldots, 0) = \eta^{-1}(\eta(m_i)) = am_i$. However, from Proposition 1, $C_{\eta}(0, 0, \ldots, 1, 0, \ldots, 0) = U(x_i, \bar{x}_i)$. This implies that $am = U(x_i, \bar{x}_i)$. The proof of the sign of the mixed partial derivative is similar to that of Corollary 1, with $y(t) = \eta^{-1}(e^{-t})$, and hence it is omitted. Q.E.D.

**Proof of Proposition 5.** (i) Monotonicity: If the copula is differentiable, and its first partial derivative with any argument is positive, then it is strictly increasing with that argument. Equation (61) guarantees this condition for all arguments.

(ii) Normalized domain and range: Because each argument is defined on the $[0, 1]$, the linear combination with the given scaling constants is also defined on $[0, 1]$;

$$C(0, \ldots, 0) = \sum_{i=1}^n k_i, 0 + \sum_{i=1}^n \sum_{j>i} k_{ij} C(0, 0, 0) + \sum_{i=1}^n \sum_{j>i} k_{ij} C(0, 0, 0)$$

$$+ \cdots + k_{123-n} C(0, 0, 0) = 0,$$

$$C(1, \ldots, 1) = \sum_{i=1}^n k_i, 1 + \sum_{i=1}^n \sum_{j>i} k_{ij} C(1, 1, 1) + \sum_{i=1}^n \sum_{j>i} k_{ij} C(1, 1, 1)$$

$$+ \cdots + k_{123-n} C(1, 1, 1) = \sum_{i=1}^n k_i + \sum_{i=1}^n \sum_{j<i} k_{ij} + \sum_{i=1}^n \sum_{j<i} k_{ij} + \cdots + k_{123-n} = 1.$$

(iii) Because each of the individual utility copulas yields a positive linear transformation at some reference values of the complement arguments, the multilinear combination must yield a linear combination of this argument at the specified reference values,

$$C_{\eta}(\lambda_i, 1, \ldots, \lambda_{i-1}, v_j, \lambda_{i+1}, \ldots, \lambda_n) = a_i v_j + b.$$  (84)

Condition (61) guarantees that this linear combination must be a positive linear transformation ($a_i > 0$) because the partial derivative of $C_{\eta}$ must be positive. Q.E.D.

**Proof of Proposition 6.** (i) Monotonicity: because each individual utility copula is nondecreasing with each of its arguments and strictly increasing for at least reference value of the complement, the convex combination of utility copulas at the same reference values must satisfy the same conditions.

(ii) The proof of the normalized range and domain is identical to that of Proposition 5, observing that the affine scaling constants must sum to one.

(iii) To show the marginal property, observe that because each of the individual utility copulas yields a positive linear transformation of each argument at some reference values of the complement arguments, the convex combination of utility copulas must yield a positive linear combination of this argument at the same specified reference values. Q.E.D.

**Acknowledgments**

The author thanks Gordon Hazen for his comments at the INFORMS Midwest Conference that inspired the linear combinations of utility copulas in §4.3. The author also thanks the associate editor and three anonymous reviewers for their many helpful suggestions on both content and exposition. Finally, the author acknowledges the National Science Foundation CAREER award NSF-DRMS 0846417 for partial support.

**References**


