Simulation algorithms for hierarchical Archimedean copulas beyond the completely monotone case

Abstract: Two simulation algorithms for hierarchical Archimedean copulas in the case when intra-group generators are not necessarily completely monotone are presented. Both generalize existing algorithms for the completely monotone case. The underlying stochastic models for both algorithms arise as a particular instance of a more general probability space studied recently in Ressel, P. (2018): A multivariate version of Williamson's theorem, $\ell^1$-symmetric survival functions, and generalized Archimedean copulas. Depend. Model. 6, 356–368. On this probability space the inter-group dependence need not be Archimedean, however, we highlight two particular circumstances that guarantee that a hierarchical Archimedean copula is obtained.

Keywords: copula, hierarchical Archimedean copula, simulation, $d$-monotone

MSC: 60E10, 60E05, 62H99, 65C10

1 Introduction

Copulas are a popular tool to model dependence outside the Gaussian world. Archimedean copulas form one of the best-studied (semi-) parametric family of copulas, due to their convenient analytical expression and the availability of efficient and exact simulation algorithms. The concept of hierarchical Archimedean copulas as a generalization of Archimedean copulas was initiated in [14, p. 88]. The idea is to model exchangeable subgroups via well-understood Archimedean copulas and introduce dependence between them again by Archimedean copulas. A full understanding of the (non-exchangeable) dependence structure can then be decomposed into accessible (exchangeable) Archimedean building blocks. As a result, hierarchical Archimedean copulas have been found useful in mathematical finance, e.g., [7, 13, 23, 27], and insurance risk, e.g., [1]. Furthermore, there is extensive literature on the design and estimation of hierarchical Archimedean copulas, see [2, 4, 5, 19, 24, 25].

The construction of hierarchical Archimedean copulas requires the parameterizing functions of the involved Archimedean copulas to satisfy non-trivial compatibility conditions. However, [17] presented such sufficient and useful compatibility conditions in arbitrary dimension, which were investigated further in [6, 8, 10, 11], who in particular constructed many examples of compatible parameterizing functions. More recently, [1] present a second, different compatibility condition. Both of these compatibility conditions are based on parameterizing functions that are all completely monotone. It is well-known that this assumption in the sufficient condition of [17] is unnecessarily strict. For instance, [9, Remark 7.3.1, p. 153] and [12] indicate that the completely monotone conditions on the parameterizing functions can be weakened. Indeed, [22, Corollary 2.3] derives sufficient compatibility conditions which are significantly weaker than those of [17].

*Corresponding Author: Jan-Frederik Mai: XAIA Investment GmbH, Sonnenstr. 19, 80331 München, Germany, E-mail: jan-frederik.mai@xaia.com
Simulation algorithms for hierarchical Archimedean copulas beyond the completely monotone case

Simulationalgorithms for hierarchical Archimedean copulas beyond the well-known completely monotone case are derived based on the presented probabilistic model. This paves the way to making use of these copulas in Monte Carlo studies. Until now, it was not possible to simulate random vectors from these copulas.

By weakening the conditions of [1], we construct hierarchical Archimedean copulas which are outside the realm of the conditions in [22], see Example 3.4 below.

Within the exchangeable groups, the presented probabilistic model agrees with the well-known \( \ell_1 \)-norm symmetric model of [18] for exchangeable Archimedean copulas, in which dependence is caused by random radial variables. Archimedean dependence between groups relies on a tricky, non-arbitrary, dependence model of these radial variables. This sheds light on how hierarchical Archimedean copulas arise as dependence structures of a natural and easy-to-grasp probabilistic model.

The remainder of the article is organized as follows. Section 2 repeats known simulation algorithms and required mathematical background. Section 3 presents simulation algorithms based on weaker assumptions than previous literature. Section 4 finally concludes.

### 2 Mathematical preliminaries and existing algorithms

Throughout, \( \overset{d}{=} \) denotes equality in distribution, the symbol \( \sim \) means “is distributed according to”, and \( f^{(k)} \) denotes the \( k \)-th derivative of a real-valued function \( f \) in one variable, provided it exists. We further denote by \( \mathbb{I}([0, 1]) \) the uniform distribution on the unit interval \([0, 1]\) and by \( \text{Bernoulli}(p) \) the Bernoulli distribution with success probability \( p \in [0, 1] \). Finally, all random variables to be introduced in the sequel are formally defined on some probability space \((\Omega, \mathcal{F}, P)\), and we denote the expected value with respect to the probability measure \( P \) by \( \mathbb{E}[.] \).

#### 2.1 Mathematical background

We recall that a function \( \varphi : [a, b] \to \mathbb{R} \) is \( n \)-monotone for \( n \in \{2, 3, \ldots\} \) if it has \( n-2 \) derivatives on \((a, b)\) and, for all \( j \in \{0, \ldots, n-2\} \), \((-1)^j \varphi^{(j)} \geq 0\) with \((-1)^{n-2} \varphi^{(n-2)}\) being non-increasing and convex on \((a, b)\). For further reference we introduce the notation

\[
\Phi_n := \{ \varphi : [0, 1] \to [0, 1] : \varphi \text{ \( n \)-monotone, } \lim_{x \to \pm \infty} \varphi(x) = 0, \varphi(0) = 1 \},
\]

as well as \( \Phi_\infty := \cap_{n \geq 2} \Phi_n \). Functions which are \( n \)-monotone for arbitrary \( n \in \mathbb{N} \) are said to be completely monotone. An Archimedean copula is a distribution function with uniform one-dimensional margins given by

\[
C_{\varphi,n}(u_1, \ldots, u_n) := \varphi^{-1}(u_1) + \cdots + \varphi^{-1}(u_n), \quad u_1, \ldots, u_n \in [0, 1],
\]
for $\varphi \in \Phi_n$, where $\varphi^{-1}$ denotes the generalized inverse of $\varphi$. We recall from [18] that for each $\varphi \in \Phi_n$ there is a random variable $R$ taking values in $(0, \infty)$, uniquely determined in law, such that

$$\varphi(x) = \mathbb{E}\left[\left(1 - \frac{x^{n-1}}{R}\right)_+\right], \quad x \geq 0,$$

and $\varphi$ is called the Williamson-$n$-transform of $R$, named after [26]. The Williamson inversion formula states the distribution function of $R$ in terms of $\varphi$ as

$$\mathbb{P}(R \leq x) = 1 - \sum_{k=0}^{n-1} \frac{(-1)^k x^k \varphi^{(k)}(x)}{k!},$$

where $\varphi^{(n-1)}$ denotes the right-continuous version of the almost everywhere existing derivative of $\varphi^{(n-2)}$. If $\tilde{S} = (S_1, \ldots, S_n)$ is independent of $R$ and uniformly distributed on the $n$-dimensional unit simplex, then the random vector

$$\bar{U} := (\varphi(R S_1), \ldots, \varphi(R S_n))$$

has distribution function $C_{\varphi,n}$, which implies a simulation algorithm for Archimedean copulas. Furthermore, we have $\varphi \in \Phi_\infty$ if and only if $R \stackrel{d}{=} (\epsilon_1 + \cdots + \epsilon_n)/M$ for a positive random variable $M$ and $n$ iid unit exponential random variables $\epsilon_1, \ldots, \epsilon_n$ independent of $M$. In this special case one has

$$\bar{U} \stackrel{d}{=} (\varphi(\epsilon_1/M), \ldots, \varphi(\epsilon_n/M)),$$

$\varphi$ is the Laplace transform of $M$, and the components of $\bar{U}$ are iid conditioned on $M$. It is worth noting for the sake of completeness that a simulation of $\tilde{S}$, which is uniformly distributed on the $n$-dimensional unit simplex, can be accomplished by

$$\tilde{S} \stackrel{d}{=} \left(\frac{\epsilon_1}{\sum_{i=1}^n \epsilon_i}, \ldots, \frac{\epsilon_n}{\sum_{i=1}^n \epsilon_i}\right),$$

with $\epsilon_1, \ldots, \epsilon_n$ iid unit exponential random variates, see [3, Theorem 5.2(2)].

We let $d_1, \ldots, d_j \in \mathbb{N}$, $d := d_1 + \cdots + d_j$, and represent $\bar{u} \in \mathbb{R}^d$ as $\bar{u} = (\bar{u}_1, \ldots, \bar{u}_j)$ accordingly. We will later denote other partitioned random vectors using the same notation, which will be made clear from the context. As a convenience, we will sometimes denote the group of all indices corresponding to the sub-vector $\bar{u}_j$ by $G_j$, so that $G_1 \cup \cdots \cup G_j$ is a partition of $\{1, \ldots, d\}$. The $d$-dimensional function

$$C(\bar{u}) := C_{\varphi_{0,j}}(C_{\varphi_{1,d_1}}(\bar{u}_1), \ldots, C_{\varphi_{j,d_j}}(\bar{u}_j))$$

is called a hierarchical Archimedean copula, provided it is a proper distribution function, which depends on properties of the functions $\varphi_{0,j}, \varphi_{1,d_1}, \ldots, \varphi_{j,d_j}$. The function $\varphi_{0,j}$ is called an outer generator, since it specifies the inter-group dependence, and the functions $\varphi_{j,d_j}$ are called inner generators, since they specify the intra-group dependence. Known simulation algorithms for $C$ rely on the assumption that all involved generators are in $\Phi_\infty$, hence completely monotone. The present article relaxes this assumption by presenting simulation algorithms for the case when the inner generators are allowed to be proper $d_j$-monotone.

For later reference, we recall that a function $g : [0, \infty) \to [0, \infty)$ is called $n$-alternating if $-g$ is $n$-monotone, and $n$-absolutely monotone if $x \mapsto g(-x)$ is $n$-monotone on $(-\infty, 0]$, see [21]. Furthermore, if $g$ is $n$-alternating for all $n \geq 2$ and $g(0) = 0$, then $g$ is called a Bernstein function. Recall that a function $g$ is a Bernstein function if and only if $\exp(-mg)$ is a Laplace transform for each $m > 0$, i.e. the probability law on $[0, \infty]$ specified by the Laplace transform $\exp(-g)$ is infinitely divisible.

### 2.2 Overview of known and new results

The following list contains analytical conditions on inner and outer generators which are relevant with respect to compatibility for nesting Archimedean copulas:
• Conditions on the outer generators:

(O₀) \( \varphi_{0,J} \in \Phi_d \), where \( d = d_1 + \ldots + d_J \).

(O₁) \( \varphi_{0,J} \in \Phi_\infty \), i.e., \( \varphi_{0,J} \) is the Laplace transform of some random variable \( M > 0 \).

(O₂) \( \varphi_{0,J} \in \Phi_\infty \), and a random variable \( M \) with Laplace transform \( \varphi_{0,J} \) is assumed to take values in the set of natural numbers \( \mathbb{N} \).

• Conditions on the inner generators (given some outer generator \( \varphi_{0,J} \)):

(I₀) \( \varphi_{j,d_j} = \varphi_{0,J} \circ (-\log) \circ h_j \) with \( h_j \in \Phi_{d_j} \) and \( h_j \) strictly positive, for each \( j = 1, \ldots, J \).

(I₁) \( \varphi_{j,d_j} = \varphi_{0,J} \circ g_j \) with \( g_j : [0, \infty) \to [0, \infty) \) bijective and \( d_j \)-alternating, for each \( j = 1, \ldots, J \).

(I₂) \( \varphi_{j,d_j} = \varphi_{0,J} \circ (-\log) \circ h_j \) with \( h_j \in \Phi_\infty \), for each \( j = 1, \ldots, J \), i.e., \( h_j \) is the Laplace transform of some positive random variable \( B_j \).

(I₃) \( \varphi_{j,d_j} = \varphi_{0,J} \circ g_j \) with \( g_j : [0, \infty) \to [0, \infty) \) an unbounded Bernstein function that is continuous¹, for each \( j = 1, \ldots, J \).

Figure 1 illustrates the different levels of generality of these conditions. In particular, we have \( O₂ \subset O₁ \subset O₀ \) and \( I₃ \subset I₂ \subset I₁ \), as well as \( I₃ \subset I₁ \). The following list provides an overview of known results (○) and contributions of the present article (*) with regards to the nesting and simulation of hierarchical Archimedean copulas:

○ [17] shows that \( O₁ \cap I₃ \) is sufficient, and provides a simulation algorithm under these conditions.

○ [22] shows that \( O₀ \cap I₃ \) is sufficient, but no simulation algorithm under these conditions is known to date.

* Lemma 3.1 below provides a simulation algorithm under the sufficient condition \( O₁ \cap I₁ \), which lies between \( O₁ \cap I₃ \) and \( O₀ \cap I₃ \).

○ [1] show that \( O₂ \cap I₂ \) is sufficient, and provide a simulation algorithm under these conditions.

* Lemma 3.3 below shows that \( O₂ \cap I₀ \) is sufficient, and provides a simulation algorithm under these conditions.

---

¹ A Bernstein function is always continuous on \((0, \infty)\) but might potentially have a discontinuity at zero, so continuous here essentially means continuous at zero.
2.3 Survey of existing algorithms and idea for relaxation

Previous literature establishes simulation algorithms under either the conditions $O_1 \cap I_3$ or the conditions $O_2 \cap I_2$ on the generators of $C$ in (4). On the one hand, the conditions $O_1 \cap I_3$ are introduced in [17] and re-interpreted and explored further in [6], leading to the simulation of a random vector $\tilde{U} \sim C$ as follows:

(i) Simulate $M$ with Laplace transform $\varphi_{0,J}$, and independently, a list of $d$ iid unit exponentials $\tilde{e} = (\tilde{e}_1, \ldots, \tilde{e}_d)$.

(ii) For $j = 1, \ldots, J$, simulate a positive random variable $M_j$ with Laplace transform $\exp(-M g_j)$. (Notice that $M_j$ is positive and finite, since $g_j$ is assumed to be unbounded and continuous.)

(iii) Return the random vector $\tilde{U}$ where for $i \in G_j$ we set $U_i := \varphi_{0,J} \circ g_j(\tilde{e}_i / M_j)$.

On the other hand, [1] consider the conditions $O_2 \cap I_2$. Under these, a random vector $\tilde{U} \sim C$ can be simulated as follows:

(i) Simulate $M$ with Laplace transform $\varphi_{0,J}$, and independently, a list of $d$ iid unit exponentials $\tilde{e} = (\tilde{e}_1, \ldots, \tilde{e}_d)$.

(ii) For $j = 1, \ldots, J$, simulate a positive random variable $M_j$ with Laplace transform $h_j^M$. (In probabilistic terms, $M_j \overset{d}{=} B_j^{(1)} + \ldots + B_j^{(M)}$, where the $B_j^{(k)}$ are iid, each with Laplace transform $h_j$.)

(iii) Return the random vector $\tilde{U}$ where for $i \in G_j$ we set $U_i := \varphi_{0,J} \circ (-\log) \circ h_j(\tilde{e}_i / M_j)$.

Remark 2.1 (Conditions $O_1 \cap I_3$ and $O_2 \cap I_2$ are different). It is already remarked in [1] that the conditions $O_1 \cap I_3$ and $O_2 \cap I_2$ are truly different, but do have an overlap. It is educational to explain this again, since it sheds some light on the probabilistic argument on which the simulation algorithms are based. Analytically, $O_2 \subset O_1$, while $I_3 \subset I_2$. Both algorithms generalize the well-known factor model algorithm for (exchangeable) Archimedean copulas based on Laplace transforms, i.e. completely monotone generators, see [16] and (3). On an analytical level, the essential ingredient for both algorithms is a Laplace transform $\varphi$ with the property that $\varphi^M$ is again a Laplace transform for all possible outcomes of the inter-group dependence generating variable $M$. The key to the simulation based on $O_1 \cap I_3$ is infinite divisibility of $\varphi = \exp(-g_j)$, which implies that $M$ can be an arbitrary positive random variable, hence $O_1$ is quite general. The essential trick in the simulation based on $O_2 \cap I_2$ is that $M$ is only allowed to take values in $\mathbb{N}$. This allows $h_j$ in $I_2$ to be an arbitrary element of $\Phi_\infty$, and with $\varphi = h_j$ the function $\varphi^M$ is a Laplace transform for all potential values of $M$. If $h_j$ in $I_2$ is infinitely divisible (i.e. $h_j = \exp(-g_j)$ with a Bernstein function $g_j$), and if $M$ in $O_1$ takes values in $\mathbb{N}$ only, then simulation based on conditions $O_1 \cap I_3$ is equivalent to simulation based on $O_2 \cap I_2$. In general, however, both algorithms are different.

For both conditions $O_1 \cap I_3$ and $O_2 \cap I_2$, all involved generators are completely monotone. Consequently, the respective probabilistic models must be seen as hierarchical generalizations of the conditionally iid model (3) rather than the more general model (2). In the simulation algorithms derived below, we stick with the completely monotone assumptions $O_1$ and $O_2$ on the outer generators, but weaken the conditions $I_3$ and $I_2$ on the inner generators. Both of our resulting simulation algorithms arise as two distinct special cases of a more general stochastic model considered in [21]. In general this model only leads to something “similar” to a hierarchical Archimedean copula. More precisely, Let $\tilde{R} = (R_1, \ldots, R_J)$ be an arbitrary random vector on $(0, \infty)^J$ and define

$$f(\tilde{X}) := \mathbb{E}\left[ \prod_{j=1}^J \left( 1 - \frac{X_j}{R_j^*} \right)^{d_j - 1} \right], \quad f_j(x) := \mathbb{E}\left[ \left( 1 - \frac{X_j}{R_j^*} \right)^{d_j - 1} \right], \quad j = 1, \ldots, J.$$  

Then the survival copula of

$$\bar{X} := (R_1, \bar{S}_1, \ldots, R_J, \bar{S}_J),$$

(5)
with \( \tilde{S}_j, j = 1, \ldots, J \), independent and each \( \tilde{S}_j \) uniformly distributed on the \( d_j \)-dimensional unit simplex (independent of \( \tilde{R} \)) is given by

\[
C_f(\tilde{u}) = f \left( \sum_{i \in G_1} f_i^{-1}(u_i), \ldots, \sum_{i \in G_I} f_i^{-1}(u_i) \right).
\]

In general, copulas of the form \( C_f \) do not coincide with hierarchical Archimedean copulas, but there is an intersection. Whereas the survival copulas of the sub-vectors \( \tilde{X}_j = R_j \tilde{S}_j \) are always Archimedean, the dependence between components from distinct groups needs not be, but can be, Archimedean, relying on a clever intersection. However, we purposely present both generalizations separately in order to highlight an open problem. The derivations in sections 3.1 and 3.2 are very similar, and could also be formulated under a hierarchical generalization of (2) rather than the conditionally iid model (3).

### 3 Simulation algorithms beyond the completely monotone case

In this section, we present two different simulation algorithms. In section 3.1 we generalize the algorithm based on conditions \( O_1 \cap I_3 \) by relaxing the condition \( I_3 \) to \( I_1 \), and include the possibility for inner generators \( \varphi_{j, d_j} \) that are not completely monotone. In section 3.2 we generalize the algorithm based on conditions \( O_2 \cap I_2 \) through relaxing the condition \( I_2 \) to \( I_0 \), and allow for the possibility of inner generators \( \varphi_{j, d_j} \) that are not completely monotone. The conditions \( O_1 \) and \( O_2 \) on the outer generators are not relaxed, as this remains an open problem. The derivations in sections 3.1 and 3.2 are very similar, and could also be formulated under one common umbrella. However, we purposely present both generalizations separately in order to highlight that the compatibility conditions are truly different in general, see Remark 2.1.

#### 3.1 Relaxing the condition \( O_1 \cap I_3 \)

Already [22, Corollary 2.3] shows that the compatibility conditions \( O_1 \cap I_3 \) of [17] can be relaxed to \( O_0 \cap I_1 \), but no simulation algorithm under these general conditions is presented. In the sequel, we present a simulation algorithm based on \( O_1 \cap I_1 \). Thus, compared to [17], we weaken the condition on the inner generators from \( I_3 \) to \( I_1 \), while the relaxation of \( O_1 \) to \( O_0 \) remains an open problem. As already indicated, the hierarchical Archimedean copula (4) under the conditions \( O_1 \cap I_1 \) will arise as a special case of the survival copula of model (5), with \( \tilde{R} \) defined appropriately.

---

2 Recall that a Lévy subordinator \( \{A_t\}_{t \geq 0} \) is a non-decreasing stochastic process with independent increments and the Laplace transform of \( A_t \) equals \( \exp(-t \phi) \) for some Bernstein function \( g \), which fully determines the probability law of the process.
Lemma 3.1 (Simulation under $O_1 \cap I_1$). Assume $O_1 \cap I_1$ holds, and consider the following list of independent variables:

- A list $e_{i,j}$, $j = 1, \ldots, J$, $i = 1, \ldots, d$, of iid exponential variables with unit mean.
- A random variable $M$ with Laplace transform $\varphi_{0,M}$.
- For arbitrary $m > 0$ independent random variables $R_{j,m}$, $j = 1, \ldots, J$, such that the Williamson-$d_j$-transform of $R_{j,m}$ equals $x \mapsto \exp(-m g_j(x))$, $x \geq 0$.

Then the partitioned random vector $\vec{U} = (\vec{U}_1, \ldots, \vec{U}_J)$, where for $j = 1, \ldots, J$ we let

$$\vec{U}_j := \left( \varphi_{0,J} \circ g_j \left( R_{j,M} \frac{e_{1,j}}{\sum_{i=1}^{d_j} e_{i,j}} \right), \ldots, \varphi_{0,J} \circ g_j \left( R_{j,M} \frac{e_{d_j,j}}{\sum_{i=1}^{d_j} e_{i,j}} \right) \right),$$

has as distribution function the hierarchical Archimedean copula $C$ in (4).

Proof. We first observe that the function

$$c_{j,m}(x) := e^{-m g_j(x)}, \quad x \geq 0,$$

lies in $\Phi_{d_j}$ for each $j = 1, \ldots, J$ and each $m > 0$. To see this, observe that the functions $\exp(m \cdot)$ and $-g_j(-\cdot)$ are $d_j$-absolutely monotone, so is their composition by [20, Theorem 12]. This implies that $c_{j,m}(\cdot)$ is $d_j$-absolutely monotone, which means that $c_{j,m}$ is $d_j$-monotone. Since further $g_j$ is assumed to be a bijection, in particular is unbounded, we realize $c_{j,m} \in \Phi_{d_j}$. Consequently, there exists a unique random variable $R_{j,m}$ whose Williamson-$d_j$-transform equals $c_{j,m}$. We consider $\vec{X} = (\vec{X}_1, \ldots, \vec{X}_J)$, where

$$\vec{X}_j := \left( R_{j,M} \frac{e_{1,j}}{\sum_{i=1}^{d_j} e_{i,j}}, \ldots, R_{j,M} \frac{e_{d_j,j}}{\sum_{i=1}^{d_j} e_{i,j}} \right), \quad j = 1, \ldots, J.$$

We compute the survival function of $\vec{X}$ as follows, where (*) follows from the general result on (exchangeable) Archimedean copulas in [18]:

$$\mathbb{P}(\vec{X} > \vec{x}) = \mathbb{E}[\mathbb{P}(\vec{X} > \vec{x} \mid M)] \supseteq \prod_{j=1}^{J} \mathbb{E} \left[ \left( 1 - \frac{\sum_{i \in G_j} x_i}{R_{j,M}} \right)^{d_j-1} \mid M \right]$$

$$= \mathbb{E} \left[ \prod_{j=1}^{J} c_{j,M} \left( \sum_{i \in G_j} x_i \right) \right] = \mathbb{E} \left[ e^{-M \sum_{j=1}^{J} g_j \left( \sum_{i \in G_j} x_i \right)} \right]$$

$$= \varphi_{0,J} \left( \sum_{j=1}^{J} g_j \left( \sum_{i \in G_j} x_i \right) \right) = \varphi_{0,J} \left( \sum_{j=1}^{J} \varphi_{0,J}^{-1} \circ g_j \left( \sum_{i \in G_j} x_i \right) \right),$$

where $\varphi_{j,d_j} = \varphi_{0,J} \circ g_j(X_j)$ is used in the last equality. This implies for the one-dimensional margin $i \in G_j$ that

$$\mathbb{P}(X_i > x) = \varphi_{j,d_j}(x) = \varphi_{0,J} \circ g_j(x).$$

Observing that $U_i = \varphi_{0,J} \circ g_j(X_i)$ for each $i = 1, \ldots, d$ proves the claim. 

\[ \square \]

Example 3.2. To illustrate we consider the case $J = d_1 = d_2 = 2$. We specify $g_j(x) := \min \{ x, a_j + b_j x \}$ for constants $a_j > 0$ and $1 > b_j > 0$, then $g_j$ is clearly not a Bernstein function, but is concave, and hence $2$-alternating. Therefore, it satisfies $I_1$ but not $I_3$. Further, we observe from (1) that $R_{j,m}$ with Williamson-$2$-transform $\exp(-m g_j)$ satisfies

$$\mathbb{P}(R_{j,m} \leq x) = \begin{cases} 1 - e^{-m x} (1 + x m), & \text{if } x < \frac{a_j}{1 - b_j}, \\ 1 - e^{-m (a_j + b_j x)} (1 + b_j m x), & \text{if } x \geq \frac{a_j}{1 - b_j}. \end{cases}$$

Apparently,

$$\mathbb{P}(R_{j,m} = \frac{a_j}{1 - b_j}) = e^{-\frac{a_j}{1 - b_j} m},$$
and so \( R_{j,m} \) has an atom at \( a_j/(1 - b_j) \). Since the position of this atom \( a_j/(1 - b_j) \) does not depend on \( m \), the resulting inner Archimedean copula with generator \( \varphi_{j,d_j} \) will exhibit a singular component, see Figure 2. If \( F_{n,\lambda} \) denotes the Erlang distribution function with parameters \( \lambda > 0 \) and \( n \in \mathbb{N} \), that is the law of the sum of \( n \) iid exponential variates with rate \( \lambda \), then we observe that

\[
\mathbb{P}(R_{j,m} \leq x) = \begin{cases} 
F_{2,m}(x) & \text{if } x < \frac{a_j}{1 - b_j} \\
1 - e^{-m a_j} + e^{-m a_j} F_{2,m,b_j}(x) & \text{if } x \geq \frac{a_j}{1 - b_j}.
\end{cases}
\]

Thus, an exact simulation algorithm for \( R_{j,m} \) is given as follows:

1. Draw \( U \sim \mathbb{U}[0, 1] \) and distinguish three cases:

   **Case 1:** \( U < F_{2,m}\left(\frac{a_j}{1 - b_j}\right) \)
   - Draw \( X \sim F_{2,m} \)
   - WHILE \( X \geq \frac{a_j}{1 - b_j} \)
   - Draw \( X \sim F_{2,m} \)
   - END WHILE
   - RETURN \( R_{j,m} = X \)

2. **Case 2:** \( U \geq 1 - e^{-m a_j} + e^{-m a_j} F_{2,b_j m}\left(\frac{a_j}{1 - b_j}\right) \)
   - Draw \( X \sim F_{2,b_j m}, B \sim \text{Bernoulli}(e^{-a_j m}) \)
   - WHILE \( BX < \frac{a_j}{1 - b_j} \)
   - Draw \( X \sim F_{2,b_j m}, B \sim \text{Bernoulli}(e^{-a_j m}) \)
   - END WHILE
   - RETURN \( R_{j,m} = X \)

3. **Case 3:** otherwise
   - RETURN \( R_{j,m} = \frac{a_j}{1 - b_j} \).

Now assume that \( \varphi_{0,2}(x) = \lambda/((\lambda + x)) \), so that the outer generator is of Clayton type with parameter 1. Notice that the choice of \( \lambda \) has no influence on the outer Archimedean copula \( C_{\varphi_{0,2}} \), but it does have an effect on the inner Archimedean copulas \( C_{\varphi_{1,2}} \) and \( C_{\varphi_{2,2}} \). Figure 2 visualizes bivariate scatter plots of some two-dimensional margins from the constructed hierarchical Archimedean copula.
Then the partitioned random vector

\[ \mathbf{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_k \end{pmatrix} \]

archimedean copula \( \psi \) under the conditions of Lemma 3.3.

variables:

model \( \psi \), with

hierarchical archimedean copula that does not even satisfy

dependence is a Clayton copula with parameter one, as can be seen from the scatter plot for

that the conditions

the preceding paragraph. As in Remark 2.1 for the completely monotone case, it is educational to point out

Figure 2: Bivariate scatter plots (5000 samples) to illustrate the hierarchical Archimedean copula of Example 3.2. Inter-group dependence is a Clayton copula with parameter one, as can be observed from the singular component in the scatter plots for \( \bar{U}_1 = (U_1, U_2) \) and \( \bar{U}_2 = (U_3, U_4) \).

3.2 Relaxing the condition \( O_2 \cap I_2 \)

In the following, we construct a simulation algorithm based on the conditions \( O_2 \cap I_0 \). Condition \( I_0 \) significantly weakens the condition \( I_3 \) in [1], while the relaxation of \( O_2 \) remains an open problem, analogous to the preceding paragraph. As in Remark 2.1 for the completely monotone case, it is educational to point out that the conditions \( O_2 \cap I_0 \) are truly different from the conditions \( O_1 \cap I_1 \), and in Example 3.4 we present a hierarchical Archimedean copula that does not even satisfy \( O_0 \cap I_1 \). As already indicated, the hierarchical Archimedean copula \( \psi \) under the conditions \( O_2 \cap I_0 \) will arise as a special case of the survival copula of model \( \psi \), with \( \bar{R} \) defined appropriately.

Lemma 3.3 (Simulation under \( O_2 \cap I_0 \)). Assume \( O_2 \cap I_0 \) holds, and consider the following list of independent variables:

- A list \( \varepsilon_{i,j} \), \( j = 1, \ldots, J \), \( i = 1, \ldots, d_j \), of iid exponential variables with unit mean.
- A random variable \( M \) with Laplace transform \( \varphi_{0,M} \).

For arbitrary \( m \in \mathbb{N} \) independent random variables \( R_{j,m} \), \( j = 1, \ldots, J \), such that the Williamson-\( d_j \) transform of \( R_{j,m} \) equals \( x \mapsto h_j(x)^m \), \( x \geq 0 \).

Then the partitioned random vector \( \bar{U} = (\bar{U}_1, \ldots, \bar{U}_J) \), where for \( j = 1, \ldots, J \) we let

\[
\bar{U}_j := \left( \varphi_{0,j} \circ (- \log) \circ h_j \left( R_{j,M} \frac{\varepsilon_{1,j}}{\sum_{i=1}^{d_j} \varepsilon_{i,j}} \right), \ldots, \varphi_{0,j} \circ (- \log) \circ h_j \left( R_{j,M} \frac{\varepsilon_{d_j,j}}{\sum_{i=1}^{d_j} \varepsilon_{i,j}} \right) \right),
\]
has as distribution function the hierarchical Archimedean copula $C$ in (4).

**Proof.** We observe that the function

$$c_{j,m}(x) := h_j(x)^m, \quad x \geq 0,$$

lies in $\Phi_{d_j}$ for each $j = 1, \ldots, J$ and each $m \in \mathbb{N}$. To see this, observe that the functions $x \mapsto x^m$ for $m \in \mathbb{N}$ and $h_j(-)$ are $d_j$-absolutely monotone, so is their composition by [20, Theorem 12]. This implies that $c_{j,m}(-)$ is $d_j$-absolutely monotone, which means that $c_{j,m}$ is $d_j$-monotone, and therefore $c_{j,m} \in \Phi_{d_j}$. Consequently, there exists a unique random variable $R_{j,m}$ whose Williamson-$d_j$-transform equals $c_{j,m}$. Considering $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_J)$, where

$$\tilde{X}_j := \left( R_{j,M} \frac{\epsilon_{1,j}}{d_j}, \ldots, R_{j,M} \frac{\epsilon_{d_j,j}}{d_j} \right), \quad j = 1, \ldots, J,$$

we compute the survival function of $\tilde{X}$ as follows, where (*) follows from the general result on (exchangeable) Archimedean copulas in [18]:

$$P(\tilde{X} > x) = E[P(\tilde{X} > \tilde{x} | M)^C] = E \left[ \prod_{j=1}^J E \left[ \left( \frac{1 - \sum_{i \in G_j} x_i}{R_{j,M}} \right)^{d_j-1} \right] \left| M \right. \right]$$

$$= E \left[ \prod_{j=1}^J c_{j,M} \left( \sum_{i \in G_j} x_i \right) \right] = E \left[ \left\{ \prod_{j=1}^J h_j \left( \sum_{i \in G_j} x_i \right) \right\}^M \right]$$

$$= E \left[ \exp \left\{ -M \sum_{j=1}^J (-\log) \circ h_j \left( \sum_{i \in G_j} x_i \right) \right\} \right] = \varphi_{0,J} \left\{ \sum_{j=1}^J (-\log) \circ h_j \left( \sum_{i \in G_j} x_i \right) \right\}$$

$$= \varphi_{0,J} \left\{ \sum_{j=1}^J \varphi_{0,j} \circ \varphi_{j,d_j} \left( \sum_{i \in G_j} x_i \right) \right\},$$

where $\varphi_{j,d_j} = \varphi_{0,J} \circ (-\log) \circ h_j$ is used in the last equality. This implies for the one-dimensional margin $i \in G_j$ that

$$P(X_i > x) = \varphi_{j,d_i}(x) = \varphi_{0,J} \circ (-\log) \circ h_j(x).$$

Observing that $U_i = \varphi_{0,J} \circ (-\log) \circ h_j(X_i)$ for each $i = 1, \ldots, d$ justifies the claim. \hfill $\square$

**Example 3.4.** For illustrative purposes we consider the case $d_1 = d_2 = 2$. We specify

$$h_j(x) := \frac{1}{\max\{1, x\}^{a_j}} - \frac{xa_j}{a_j + 1} \frac{1}{\max\{1, x\}^{a_j+1}}$$

for a constant $a_j > 0$. Then $h_j \in \Phi_2$, since it equals the Williamson-2-transform of a random variable $R$ with density $f_R(x) = a_j x^{-a_j-1} 1_{\{x > 1\}}$. Furthermore, we notice that $h_j \notin \Phi_\infty$, since $R$ has support bounded away from zero and thus cannot be represented in distribution as the quotient of an Erlang-distributed random variable and some other independent positive random variable. Also, Figure 3 shows that $(-\log) \circ h_j$ is not concave for arbitrary $a_j$, and thus not 2-alternating, implying that this example does not arise as a special case of the conditions $O_1 \cap I_1$ in the previous paragraph.

With the help of the Williamson inversion formula (1), a tedious but straightforward computation gives

$$P(R_{j,m} > x) = \left( \frac{1}{\max\{1, x\}^{a_j}} - \frac{xa_j}{a_j + 1} \frac{1}{\max\{1, x\}^{a_j+1}} \right)^{m-1} \times \left( \frac{1}{\max\{1, x\}^{a_j}} - \frac{xa_j}{a_j + 1} \frac{1 - m}{\max\{1, x\}^{a_j+1}} \right).$$
Figure 3: The function \((- \log) \circ h_j\) for \(a_j = 5\) in Example 3.4, obviously not concave.

For \(m = 1, 2\) we obtain

\[
P(R_{j,1} \leq x) = 1 - \frac{1}{\max(1, x)^{a_j}}, \quad P(R_{j,2} \leq x) = \begin{cases} 
\frac{a_j^j}{(a_j+1)^j} x, & \text{if } x < 1 \\
1 - \frac{2 a_j+1}{(a_j+1)^j} \frac{1}{x^{a_j}}, & \text{if } x \geq 1
\end{cases},
\]

which are both distribution functions that can be simulated exactly and quickly via the inversion method, in particular

\[
R_{j,2} \overset{d}{=} \begin{cases} 
\frac{(a_j+1)^j}{a_j} U, & \text{if } U < \frac{a_j^j}{(a_j+1)^j}, \\
\left(\frac{2 a_j+1}{(a_j+1)^j} \frac{1}{x^{a_j}}\right)^{1/a_j}, & \text{else}
\end{cases}, \quad U \sim \mathcal{U}[0,1].
\]

For \(m \geq 3\) we can also use the inversion method together with a bisection routine to simulate \(R_{j,m}\). To this end, denote the survival function of \(R_{j,m}\) by \(\bar{F}(x) := P(R_{j,m} > x)\) and simulate \(R_{j,m} \overset{d}{=} \bar{F}^{-1}(U)\) with \(U \sim \mathcal{U}[0,1]\). To this end, we compute the constant \(z := \frac{1}{(1+a_j)^{m-1}} \left(1 + \frac{(m-1) a_j}{a_j+1}\right)\). Then for \(U < z\) we have

\[
\bar{F}^{-1}(U) = \left(1 + \frac{(m-1) a_j}{a_j+1}\right)^{1/(a_j m)} \frac{U}{(a_j + 1)^{m-1}}.
\]

For \(U \geq z\) we use a bijection algorithm to determine \(\bar{F}^{-1}(U)\) as the unique value \(x \in [0,1]\) such that

\[
\left(1 - \frac{x a_j}{a_j + 1}\right)^{m-1} \left(1 + \frac{x a_j (m-1)}{a_j + 1}\right) = U.
\]
Now assume that $\varphi_{0,2}(x) = 1 - (1 - \exp(-x))^{1/\theta}$ (Joe's family), which means that $M \sim \varphi_{0,2}$ can be simulated according to the algorithm presented in [10], see also [15, Algorithm 2.2, p. 71]. Figure 4 visualizes bivariate scatter plots of some two-dimensional margins from the constructed hierarchical Archimedean copula.

**Figure 4**: Bivariate scatter plots (5000 samples) to illustrate the hierarchical Archimedean copula of Example 3.4. Inter-group dependence is visualized in the scatter plot for $(U_1, U_3)$, specified by $\varphi_{0,2}(x) = 1 - (1 - \exp(-x))^{1/\theta}$, corresponding to Joe’s family. Intra-group dependence is based on generators that are not completely monotone, and is visualized by the scatter plots for $\vec{U}_1 = (U_1, U_2)$ and $\vec{U}_2 = (U_3, U_4)$.

### 4 Conclusion

Simulation algorithms have been presented for two-level hierarchical Archimedean copulas, for which the intra-group dependence can be parameterized by generators that are not completely monotone, thus generalizing existing simulation algorithms. The algorithms have been illustrated along with two explicit examples for hierarchical Archimedean copulas. The outer generators, however, were still assumed to be completely monotone. The simulation from hierarchical Archimedean copulas with outer generators that are not completely monotone remains an interesting open problem, relevant also for the introduction of further levels of nesting.
Acknowledgements Helpful comments on earlier versions of the manuscript by D. Struble as well as three anonymous referees are gratefully acknowledged.

References

[1] Cossette, H., S.-P. Gadoury, É. Marceau, and I. Mtalai (2017). Hierarchical Archimedean copulas through multivariate compound distributions. *Insurance Math. Econom.* 76, 1–13.

[2] Matsypura, D., E. Neo, and A. Prokhorov (2016). Estimation of hierarchical Archimedean copulas as a shortest path problem. *Econ. Lett.* 149, 131–134.

[3] Fang, K.-T., S. Kotz, and K.-W. Ng (1990). *Symmetric Multivariate and Related Distributions.* Chapman & Hall, London.

[4] Górecki, J., M. Hofert, and M. Holeňa (2016). An approach to structure determination and estimation of hierarchical Archimedean copulas and its application to Bayesian classification. *J. Intell. Inf. Syst.* 46(1), 21–59.

[5] Górecki, J., M. Hofert, and M. Holeňa (2017). On structure, family and parameter estimation of hierarchical Archimedean copulas. *J. Stat. Comput. Simul.* 87(17), 3261–3324.

[6] Hering, C., M. Hofert, J.-F. Mai, and M. Scherer (2010). Constructing hierarchical Archimedean copulas with Lévy subordinators. *J. Multivariate Anal.* 101(6), 1428–1433.

[7] Höcht, S. and R. Zagst (2010). Pricing distressed CDOs with stochastic recovery. *Rev. Deriv. Res.* 13(3), 219–244.

[8] Hofert, M. (2008). Sampling Archimedean copulas. *Comput. Statist. Data Anal.* 52(12), 5163–5174.

[9] Hofert, M. (2010). Construction and sampling of nested Archimedean copulas. In P. Jaworski, F. Durante, W. Härdle, and T. Rychlik (Eds.), *Copula Theory and its Applications,* pp. 147–160. Springer, Heidelberg.

[10] Hofert, M. (2011). Efficiently sampling Archimedean copulas. *Comput. Statist. Data Anal.* 55(1), 57–70.

[11] Hofert, M. (2012). A stochastic representation and sampling algorithm for nested Archimedean copulas. *J. Stat. Comput. Simul.* 82(9), 1239–1255.

[12] Hofert, M. and D. Pham (2013). Densities of nested Archimedean copulas. *J. Multivariate Anal.* 118, 37–52.

[13] Hofert, M. and M. Scherer (2011). CDO pricing with nested Archimedean copulas. *Quant. Finance* 11(5), 775–787.

[14] Joe, H. (1997). *Multivariate Models and Dependence Concepts.* Chapman & Hall, London.

[15] Mai, J.-F. and M. Scherer (2017). *Simulating Copulas.* Second edition. World Scientific Publishing, Singapore.

[16] Marshall, A. W. and I. Olkin (1988). Families of multivariate distributions. *J. Amer. Statist. Assoc.* 83(403), 834–841.

[17] McNeill, A. J. (2008). Sampling nested Archimedean copulas. *J. Stat. Comput. Simul.* 78(5-6), 567–581.

[18] McNeill, A. J. and J. Nešlehová (2009). Multivariate Archimedean copulas, $d$-monotone functions and $\ell_1$-symmetric distributions. *Ann. Statist.* 37(5B), 3059–3097.

[19] Okhrin, O., Y. Okhrin and W. Schmid (2013). On the structure and estimation of hierarchical Archimedean copulas. *J. Econom.* 173(2), 189–204.

[20] Ressel, P. (2014). Higher order monotonic functions of several variables. *Positivity* 18(2), 257–285.

[21] Ressel, P. (2018). A multivariate version of Williamson’s theorem, $\ell^1$-symmetric survival functions, and generalized Archimedean copulas. *Depend. Model.* 6, 356–368.

[22] Rezapour, M. (2015). On the construction of nested Archimedean copulas for $d$-monotone generators. *Statist. Probab. Lett.* 101, 21–32.

[23] Savu, C. and M. Trede (2010). Hierarchies of Archimedean copulas. *Quant. Finance* 10(3), 295–304.

[24] Segers, J. and N. Uyttendaele (2014). Nonparametric estimation of the tree structure of a nested Archimedean copula. *Comput. Statist. Data Anal.* 72, 190–204.

[25] Uyttendaele, N. (2018). On the estimation of nested Archimedean copulas: a theoretical and an experimental comparison. *Comput. Statist.* 33(2), 1047–1070.

[26] Williamson, R. E. (1956). Multiply monotone functions and their Laplace transforms. *Duke Math. J.* 23, 189–207.

[27] Zhu, W., C.-W. Wang, and K. S. Tan (2016). Structure and estimation of Lévy subordinated hierarchical Archimedean copulas (LSHAC): Theory and empirical tests. *J. Bank. Finance* 69, 20–36.