J. Appl. Prob. **52,** 490–507 (2015)
Printed in England

ALGORITHMS AND FORMULAE FOR CONVERSION BETWEEN SYSTEM SIGNATURES AND RELIABILITY FUNCTIONS

JEAN-LUC MARICHAL,* *University of Luxembourg*

Abstract

The concept of a signature is a useful tool in the analysis of semicoherent systems with continuous, and independent and identically distributed component lifetimes, especially for the comparison of different system designs and the computation of the system reliability. For such systems, we provide conversion formulae between the signature and the reliability function through the corresponding vector of dominations and we derive efficient algorithms for the computation of any of these concepts from any other. We also show how the signature can be easily computed from the reliability function via basic manipulations such as differentiation, coefficient extraction, and integration.

Keywords: Semicoherent system; system signature; reliability function; domination vector

2010 Mathematics Subject Classification: Primary 62N05; 90B25
Secondary 94C10

1. Introduction

Consider an n-component system (C, ϕ), where C is the set $[n] = \{1, \ldots, n\}$ of its components and $\phi\colon \{0,1\}^n \to \{0,1\}$ is its structure function which expresses the state of the system in terms of the states of its components. We assume that the system is *semicoherent*, which means that the structure function ϕ is nondecreasing in each variable and satisfies the conditions $\phi(0, \ldots, 0) = 0$ and $\phi(1, \ldots, 1) = 1$. We also assume that the components have continuous, and independent and identically distributed (i.i.d.) lifetimes $T_1, \ldots, T_n$.

Samaniego [10] introduced the *signature* of such a system as the n-vector $\boldsymbol{s} = (s_1, \ldots, s_n)$ whose kth coordinate s_k is the probability that the kth component failure causes the system to fail. That is,

$$s_k = \mathbb{P}\{T_{\mathrm{S}} = T_{\{k:\, n\}}\} \quad \text{for } k = 1, \ldots, n,$$

where T_{S} denotes the system lifetime and $T_{\{k:\, n\}}$ denotes the kth smallest lifetime. From this definition we can immediately derive the identity $\sum_{k=1}^{n} s_k = 1$.

It is often very convenient to express the signature vector $\boldsymbol{s}$ in terms of the *tail signature* of the system, a concept introduced by Boland [3] and named so by Gertsbakh *et al.* [5]. The tail signature of the system is the $(n+1)$-vector $\overline{\boldsymbol{S}} = (\overline{S}_0, \ldots, \overline{S}_n)$ defined from $\boldsymbol{s}$ by

$$\overline{S}_k = \sum_{i=k+1}^{n} s_i \quad \text{for } k = 0, \ldots, n. \tag{1}$$

Received 13 February 2014; revision received 7 May 2014.
* Postal address: Mathematics Research Unit, FSTC, University of Luxembourg, 6, rue Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg. Email address: jean-luc.marichal@uni.lu

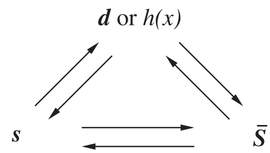

FIGURE 1: Bijective linear transformations.

In particular, we have $\overline{S}_0 = 1$ and $\overline{S}_n = 0$. Moreover, it is clear that the signature $\boldsymbol{s}$ can be retrieved from the tail signature $\overline{\boldsymbol{S}}$ through the following equation:

$$s_k = \overline{S}_{k-1} - \overline{S}_k \quad \text{for } k = 1, \dots, n. \tag{2}$$

Recall that the *reliability function* associated with the structure function ϕ is the unique multilinear polynomial function $h\colon [0, 1]^n \to \mathbb{R}$ whose restriction to $\{0, 1\}^n$ is precisely the structure function ϕ. Since the component lifetimes are independent, this function expresses the reliability of the system in terms of the component reliabilities (for general background; see [2, Chapter 2] and for a more recent reference; see [9, Section 3.2]).

By identifying the variables of the reliability function, we obtain a real polynomial function $h(x)$ of degree n at most. The n-vector $\boldsymbol{d} = (d_1, \dots, d_n)$ whose kth coordinate d_k is the coefficient of x^k in $h(x)$ is called the *vector of domination* of the system (see, e.g. [11, Section 6.2]).

The computation of the signature of a large system by means of the usual methods may be cumbersome and tedious since it requires the evaluation of the structure function ϕ at every element of $\{0, 1\}^n$. However, Boland *et al.* [4] observed that the n-vectors $\boldsymbol{s}$ and $\boldsymbol{d}$ can always be computed from each other through simple bijective linear transformations (see also [11, Section 6.3]). Although these linear transformations were not given explicitly, they show that the signature vector $\boldsymbol{s}$ can be efficiently computed from the domination vector $\boldsymbol{d}$, or, equivalently, from the polynomial function $h(x)$. Since (1) and (2) provide linear conversion formulae between vectors $\boldsymbol{s}$ and $\overline{\boldsymbol{S}}$, we observe that any of the vectors $\boldsymbol{s}$, $\overline{\boldsymbol{S}}$, and $\boldsymbol{d}$ can be computed from any other by means of a bijective linear transformation (see Figure 1).

After recalling some basic formulae in Section 2, in Section 3 we yield these linear transformations explicitly and present them as linear conversion formulae. From these conversion formulae we derive algorithms for the computation of any of these vectors from any other. These algorithms prove to be very efficient since they require at most $\frac{1}{2}n(n+1)$ additions and multiplications.

We also show how the computation of the vectors $\boldsymbol{s}$ and $\overline{\boldsymbol{S}}$ can be easily performed from basic manipulations of function $h(x)$ such as differentiation, reflection, coefficient extraction, and integration. For instance, we establish the polynomial identity (see (26))

$$\sum_{k=1}^{n} \binom{n}{k} s_k x^k = \int_0^x (R^{n-1}h')(t+1)\,\mathrm{d}t, \tag{3}$$

where $h'(x)$ is the derivative of $h(x)$ and $(R^{n-1}h')(x)$ is the polynomial function obtained from $h'(x)$ by switching the coefficients of x^k and x^{n-1-k} for $k = 0, \dots, n-1$. Applying this result to the classical five-component bridge system (see Example 1 below), we can easily see that (3) reduces to

$$5s_1x + 10s_2x^2 + 10s_3x^3 + 5s_4x^4 + s_5x^5 = 2x^2 + 6x^3 + x^4.$$

By equating the corresponding coefficients, we immediately obtain the signature vector $\mathbf{s} = (0, \frac{1}{5}, \frac{3}{5}, \frac{1}{5}, 0)$.

In Section 4 we examine the general non-i.i.d. setting, where the component lifetimes $T_1, \dots, T_n$ may be dependent. We show how a certain modification of the structure function enables us to formally extend almost all the conversion formulae and algorithms obtained in Sections 2 and 3 to the general dependent setting. Finally, we end our paper in Section 5 with some concluding remarks.

2. Preliminaries

Boland [3] showed that every coordinate s_k of the signature vector can be explicitly written in the form

$$s_k = \sum_{A\subseteq C, |A|=n-k+1} \frac{1}{\binom{n}{|A|}}\phi(A) - \sum_{A\subseteq C, |A|=n-k} \frac{1}{\binom{n}{|A|}}\phi(A). \tag{4}$$

Throughout this paper we identify Boolean n-vectors $\mathbf{x} \in \{0,1\}^n$ and subsets $A \subseteq [n]$ in the usual way, i.e. by setting $x_i = 1$ if and only if $i \in A$. Thus, we use the same symbol to denote both a function $f\colon \{0,1\}^n \to \mathbb{R}$ and the corresponding set function $f\colon 2^{[n]} \to \mathbb{R}$ interchangeably. For instance, we write $\phi(0,\dots,0) = \phi(\varnothing)$ and $\phi(1,\dots,1) = \phi(C)$.

As mentioned in the introduction, the reliability function associated with the structure function ϕ is the multilinear function $h\colon [0,1]^n \to \mathbb{R}$ defined by

$$h(\mathbf{x}) = h(x_1,\dots,x_n) = \sum_{A\subseteq C}\phi(A)\prod_{i\in A}x_i\prod_{i\in C\setminus A}(1-x_i). \tag{5}$$

It is easy to see that this function can always be put in the unique standard multilinear form

$$h(\mathbf{x}) = \sum_{A\subseteq C} d(A)\prod_{i\in A}x_i, \tag{6}$$

where, for every $A \subseteq C$, the coefficient $d(A)$ is an integer.

By identifying the variables $x_1,\dots,x_n$ in function $h(\mathbf{x})$, we define its diagonal section $h(x,\dots,x)$, which we have simply denoted by $h(x)$. From (5) and (6) we immediately obtain

$$h(x) = \sum_{A\subseteq C}\phi(A)x^{|A|}(1-x)^{n-|A|} = \sum_{A\subseteq C} d(A)x^{|A|},$$

or, equivalently,

$$h(x) = \sum_{k=0}^{n}\phi_k\, x^k(1-x)^{n-k} = \sum_{k=0}^{n} d_k x^k, \tag{7}$$

where

$$\phi_k = \sum_{A\subseteq C, |A|=k}\phi(A) \quad\text{and}\quad d_k = \sum_{A\subseteq C, |A|=k} d(A) \quad \text{for } k = 0,\dots,n. \tag{8}$$

Clearly, we have $\phi_0 = \phi(\varnothing) = 0$ and $d_0 = d(\varnothing) = h(0) = 0$. As already mentioned, the n-vector $\mathbf{d} = (d_1,\dots,d_n)$ is called the *vector of dominations* of the system.

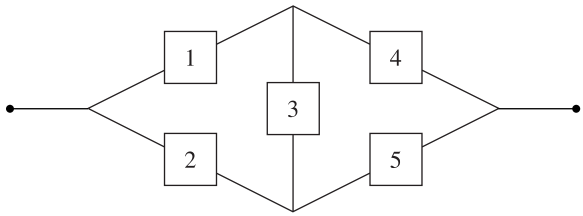

FIGURE 2: Illustration of a bridge structure.

Example 1. Consider the bridge structure as shown in Figure 2. The corresponding structure function is given by

$$\phi(x_1, \dots, x_5) = x_1x_4 \amalg x_2x_5 \amalg x_1x_3x_5 \amalg x_2x_3x_4,$$

where $\amalg$ is the (associative) coproduct operation defined by $x \amalg y = 1 - (1 - x)(1 - y)$. The corresponding reliability function, given in (5), can be computed by expanding the coproducts in ϕ and then simplifying the resulting algebraic expression using $x_i^2 = x_i$. We have

$$\begin{aligned} h(x_1, \dots, x_5) = {} & x_1x_4 + x_2x_5 + x_1x_3x_5 + x_2x_3x_4 - x_1x_2x_3x_4 - x_1x_2x_3x_5 \\ & - x_1x_2x_4x_5 - x_1x_3x_4x_5 - x_2x_3x_4x_5 + 2x_1x_2x_3x_4x_5. \end{aligned}$$

We then obtain its diagonal section $h(x) = 2x^2 + 2x^3 - 5x^4 + 2x^5$ and finally the domination vector $\boldsymbol{d} = (0, 2, 2, -5, 2)$.

In Example 1 we illustrate the important fact that the reliability function $h(\boldsymbol{x})$ of any system can be easily obtained from the minimal path sets simply by first expressing the structure function as a coproduct over the minimal path sets, and then expanding the coproduct and simplifying the resulting algebraic expression (using $x_i^2 = x_i$) until it becomes multilinear. The diagonal section $h(x)$ of the reliability function is then obtained by identifying all the variables.

This observation is crucial since, when combined with an efficient algorithm for converting the polynomial function $h(x)$ into the signature vector, it provides an efficient way to compute the signature of any system from its minimal path sets.

3. Conversion equations

Recall that (6) provides the standard multilinear form of the reliability function $h(\boldsymbol{x})$. As mentioned, for instance, in [9, p. 31], the link between the coefficients $d(A)$ and the values $\phi(A)$ is given through the following linear conversion formulae (obtained from the Möbius inversion theorem)

$$\phi(A) = \sum_{B \subseteq A} d(B) \quad \text{and} \quad d(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} \phi(B). \tag{9}$$

The following proposition yields the linear conversion formulae between the n-vectors $\boldsymbol{d} = (d_1, \dots, d_n)$ and $(\phi_1, \dots, \phi_n)$. Note that an alternative form of (11) was previously obtained by Samaniego [11, Section 6.3].

https://doi.org/10.1239/jap/1437658611 Published online by Cambridge University Press

Proposition 1. *We have*

$$\phi_k = \sum_{j=0}^{k} \binom{n-j}{k-j} d_j \quad \textit{for } k = 1, \ldots, n, \tag{10}$$

and

$$d_k = \sum_{j=0}^{k} (-1)^{k-j} \binom{n-j}{k-j} \phi_j \quad \textit{for } k = 1, \ldots, n. \tag{11}$$

Proof. By (8) and (9), we have

$$\phi_k = \sum_{A \subseteq C, |A|=k} \phi(A) = \sum_{A \subseteq C, |A|=k} \sum_{B \subseteq A} d(B).$$

Permuting the sums and then setting $j = |B|$, we obtain

$$\begin{aligned} \phi_k &= \sum_{B \subseteq C, |B| \leq k} d(B) \sum_{A \supseteq B, |A|=k} 1 \\ &= \sum_{B \subseteq C, |B| \leq k} \binom{n-|B|}{k-|B|} d(B) \\ &= \sum_{j=0}^{k} \binom{n-j}{k-j} \sum_{B \subseteq C, |B|=j} d(B), \end{aligned}$$

which proves (10). We can establish (11) similarly. This completes the proof of the proposition.

We are now ready to establish conversion formulae and algorithms as stated in the introduction.

3.1. Conversions between s and $\overline{S}$

We already know that the linear conversion formulae between the vectors s and $\overline{S}$ are given by (1) and (2). This conversion can also be explicitly expressed by means of a polynomial identity. Let $\sum_{k=1}^{n} s_k x^k$ and $\sum_{k=0}^{n} \overline{S}_k x^k$ be the *generating functions* of vectors s and $\overline{S}$, respectively. Then we have the polynomial identity

$$\sum_{k=1}^{n} s_k x^k = 1 + (x-1) \sum_{k=0}^{n} \overline{S}_k x^k. \tag{12}$$

Indeed, using (2) and summation by parts, we obtain

$$\sum_{k=1}^{n} s_k x^k = \sum_{k=1}^{n} (\overline{S}_{k-1} - \overline{S}_k) x^k = x + \sum_{k=1}^{n} \overline{S}_k (x^{k+1} - x^k),$$

which proves (12).

For instance, for the bridge system described in Example 1, the generating functions of vectors s and $\overline{S}$ are given by $\frac{1}{5}x^2 + \frac{3}{5}x^3 + \frac{1}{5}x^4$ and $1 + x + \frac{4}{5}x^2 + \frac{1}{5}x^3$, respectively. We can easily verify that (12) holds for these functions.

3.2. Conversions between $\overline{\boldsymbol{S}}$ and $\boldsymbol{d}$

Combining (1) with (4) and (8), we observe that

$$\overline{S}_k = \frac{1}{\binom{n}{k}} \sum_{A\subseteq C, |A|=n-k} \phi(A) = \frac{1}{\binom{n}{k}} \phi_{n-k} \quad \text{for } k = 0, \ldots, n. \tag{13}$$

Recall that a *path set* of the system is a component subset A such that $\phi(A) = 1$. It follows from (13) that ϕ_k is precisely the number of path sets of size k, and that $\overline{S}_{n-k}$ is the proportion of component subsets of size k which are path sets. We also observe that the leading coefficient d_n of $h(\boldsymbol{x})$, also known as the *signed domination* [1] of $h(\boldsymbol{x})$, is 0 if and only if there are as many path sets of odd sizes as path sets of even sizes. This observation immediately follows from the identity $d_n = \sum_{j=0}^{n} (-1)^{n-j} \phi_j$, obtained by setting $k = n$ in (11).

Using (10), (11), and (13), we immediately obtain the following conversion formulae between the vectors $\overline{\boldsymbol{S}}$ and $\boldsymbol{d}$.

Proposition 2. *We have*

$$\overline{S}_k = \sum_{j=0}^{n-k} \frac{\binom{n-j}{k}}{\binom{n}{k}} d_j = \sum_{j=0}^{n-k} \frac{\binom{n-k}{j}}{\binom{n}{j}} d_j \quad \textit{for } k = 0, \ldots, n, \tag{14}$$

$$d_k = \binom{n}{k} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} \overline{S}_{n-j} \quad \textit{for } k = 0, \ldots, n. \tag{15}$$

We can rewrite (15) in a simpler form by using the classical difference operator Δ_i which maps a sequence z_i to the sequence $\Delta_i z_i = z_{i+1} - z_i$. Defining the kth difference $\Delta_i^k z_i$ of a sequence z_i recursively as $\Delta_i^0 z_i = z_i$ and $\Delta_i^k z_i = \Delta_i \Delta_i^{k-1} z_i$, we can show by induction on k that

$$\Delta_i^k z_i = \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} z_{i+j}. \tag{16}$$

Comparing (15) with (16) immediately shows that (15) can be rewritten as

$$d_k = \binom{n}{k} (\Delta_i^k \overline{S}_{n-i})|_{i=0} \quad \text{for } k = 1, \ldots, n, \tag{17}$$

and the vector $\boldsymbol{d}$ can then be computed efficiently from a classical difference table (see Figure 3).

Setting $D_{j,k} = \binom{n}{k} (\Delta_i^k \overline{S}_{n-i})|_{i=j}$, from (17) we can easily derive the following algorithm for the computation of $\boldsymbol{d}$. This algorithm requires only $\frac{1}{2} n(n+1)$ additions and multiplications.

Algorithm 1. The following algorithm inputs vector $\overline{\boldsymbol{S}}$ and outputs vector $\boldsymbol{d}$. It uses the variables $D_{j,k}$ for $k = 0, \ldots, n$ and $j = 0, \ldots, n-k$.

Step 1. For $j = 0, \ldots, n$, set $D_{j,0} := \overline{S}_{n-j}$.

Step 2. For $k = 1, \ldots, n$
For $j = 0, \ldots, n-k$
$D_{j,k} := ((n-k+1)/k)(D_{j+1,k-1} - D_{j,k-1})$

Step 3. For $k = 0, \ldots, n$, set $d_k := D_{0,k}$.

https://doi.org/10.1239/jap/1437658611 Published online by Cambridge University Press

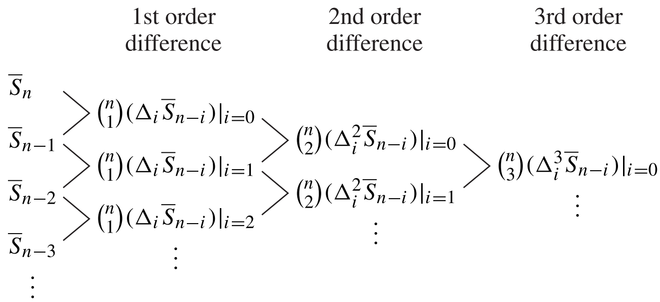

FIGURE 3: The difference table for the computation of $\boldsymbol{d}$ from $\overline{\boldsymbol{S}}$.

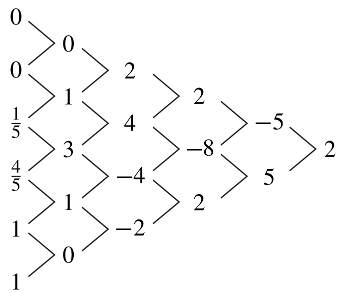

FIGURE 4: The difference table for the computation of $\boldsymbol{d}$ from $\overline{\boldsymbol{S}}$ (Example 2).

Example 2. Consider the bridge system described in Example 1. The corresponding tail signature vector is given by $\overline{\boldsymbol{S}} = (1, 1, \frac{4}{5}, \frac{1}{5}, 0, 0)$. Forming the difference table (see Figure 4) and reading its first row, we obtain the vector $\boldsymbol{d} = (0, 2, 2, -5, 2)$ and therefore the function $h(x) = 2x^2 + 2x^3 - 5x^4 + 2x^5$.

The converse transformation (14) can then be computed efficiently by the following algorithm, in which we compute the quantities

$$S_{j,k} = \sum_{i=0}^{k} \frac{\binom{k}{i}\binom{i+j}{i}}{\binom{n-j}{i}} d_{i+j}.$$

Algorithm 2. The following algorithm inputs vector $\boldsymbol{d}$ and outputs vector $\overline{\boldsymbol{S}}$. It uses the variables $S_{j,k}$ for $k = 0, \dots, n$ and $j = 0, \dots, n-k$.

Step 1. For $j = 0, \dots, n$, set $S_{j,0} := d_j$.

Step 2. For $k = 1, \dots, n$
  For $j = 0, \dots, n-k$
    $S_{j,k} := ((j+1)/(n-j))S_{j+1,k-1} + S_{j,k-1}$

Step 3. For $k = 0, \dots, n$, set $\overline{S}_{n-k} := S_{0,k}$.

3.3. Conversions between $\boldsymbol{s}$ and $\boldsymbol{d}$

The following proposition yields the conversion formulae between the vectors $\boldsymbol{s}$ and $\boldsymbol{d}$. Note that a nonexplicit version of (18) was previously obtained by Boland *et al.* [4] (see [11, Theorem 6.1]).

https://doi.org/10.1239/jap/1437658611 Published online by Cambridge University Press

$$\begin{array}{cccc}
 & \text{1st order difference} & \text{2nd order difference} & \text{3rd order difference} \\
\binom{n}{1}s_n & & & \\
 & \binom{n}{2}(\Delta_i s_{n-i})|_{i=0} & & \\
\binom{n}{1}s_{n-1} & & \binom{n}{3}(\Delta_i^2 s_{n-i})|_{i=0} & \\
 & \binom{n}{2}(\Delta_i s_{n-i})|_{i=1} & & \binom{n}{4}(\Delta_i^3 s_{n-i})|_{i=0} \\
\binom{n}{1}s_{n-2} & & \binom{n}{3}(\Delta_i^2 s_{n-i})|_{i=1} & \vdots \\
 & \binom{n}{2}(\Delta_i s_{n-i})|_{i=2} & \vdots & \\
\binom{n}{1}s_{n-3} & \vdots & & \\
\vdots & & &
\end{array}$$

FIGURE 5: The difference table for the computation of $\boldsymbol{d}$ from $\boldsymbol{s}$.

Proposition 3. *We have*

$$s_k = \sum_{j=0}^{n-k} \frac{\binom{n-j}{k}}{\binom{n}{k}} \frac{j+1}{n-j} d_{j+1} = \sum_{j=1}^{n-k+1} \frac{\binom{n-k}{j-1}}{\binom{n}{j}} d_j \quad \textit{for } k = 1, \dots, n, \tag{18}$$

$$d_k = \binom{n}{k} \sum_{j=0}^{k-1} (-1)^{k-1-j} \binom{k-1}{j} s_{n-j} \quad \textit{for } k = 1, \dots, n. \tag{19}$$

$$d_k = \binom{n}{k} (\Delta_i^{k-1} s_{n-i})|_{i=0} \quad \textit{for } k = 1, \dots, n, \tag{20}$$

Proof. Substituting (14) into (2), we obtain

$$s_k = \overline{S}_{k-1} - \overline{S}_k = \sum_{j=1}^{n-k+1} \frac{\binom{n-k+1}{j}}{\binom{n}{j}} d_j - \sum_{j=1}^{n-k} \frac{\binom{n-k}{j}}{\binom{n}{j}} d_j = \sum_{j=1}^{n-k} \frac{\binom{n-k}{j-1}}{\binom{n}{j}} d_j + \frac{1}{\binom{n}{n-k+1}} d_{n-k+1},$$

which proves (18). By (2), we have $\Delta_i \overline{S}_{n-i} = s_{n-i}$ for $i = 0, \dots, n-1$. Equation (20) then follows from (17). Equation (19) then follows immediately from (20). This completes the proof of the proposition.

From (20), the vector $\boldsymbol{d}$ can be efficiently computed directly from $\boldsymbol{s}$ by means of a difference table (see Figure 5).

Setting $d_{j,k} = \binom{n}{k}(\Delta_i^{k-1} s_{n-i})|_{i=j-1}$, we can also derive the following algorithm for the computation of vector $\boldsymbol{d}$. This algorithm requires only $\frac{1}{2}n(n-1)$ additions and multiplications.

Algorithm 3. The following algorithm inputs vector $\boldsymbol{s}$ and outputs vector $\boldsymbol{d}$. It uses the variables $d_{j,k}$ for $k = 1, \dots, n$ and $j = 1, \dots, n-k+1$.

Step 1. For $j = 1, \dots, n$, set $d_{j,1} := n s_{n-j+1}$.

Step 2. For $k = 2, \dots, n$
  For $j = 1, \dots, n-k+1$
    $d_{j,k} := ((n-k+1)/k)(d_{j+1,k-1} - d_{j,k-1})$

Step 3. For $k = 1, \dots, n$, set $d_k := d_{1,k}$.

Example 3. Consider again the bridge system described in Example 1. The corresponding signature vector is given by $\boldsymbol{s} = (0, \frac{1}{5}, \frac{3}{5}, \frac{1}{5}, 0)$. Forming the difference table (see Figure 6)

https://doi.org/10.1239/jap/1437658611 Published online by Cambridge University Press

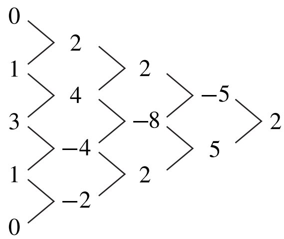

FIGURE 6: The difference table for the computation of $\boldsymbol{d}$ from $\boldsymbol{s}$ (Example 3).

and reading its first row, we obtain the vector $\boldsymbol{d} = (0, 2, 2, -5, 2)$ and, hence, the function $h(x) = 2x^2 + 2x^3 - 5x^4 + 2x^5$.

The converse transformation (18) can then be computed efficiently by the following algorithm, in which we compute the quantities:

$$s_{j,k} = \frac{1}{n} \sum_{i=1}^{k} \frac{\binom{k-1}{i-1}\binom{i+j-1}{i-1}}{\binom{n-j}{i-1}} d_{i+j-1}.$$

Algorithm 4. The following algorithm inputs vector $\boldsymbol{d}$ and outputs vector $\boldsymbol{s}$. It uses the variables $s_{j,k}$ for $k = 1, \ldots, n$ and $j = 1, \ldots, n - k + 1$.

Step 1. For $j = 1, \ldots, n$, set $s_{j,1} := (1/n)d_j$.

Step 2. For $k = 2, \ldots, n$
For $j = 1, \ldots, n - k + 1$
$s_{j,k} := ((j + 1)/(n - j))s_{j+1,k-1} + s_{j,k-1}$

Step 3. For $k = 1, \ldots, n$, set $s_{n-k+1} := s_{1,k}$.

3.4. Conversions between $\overline{S}$ or s and $h(x)$

The conversion formulae between vectors $\boldsymbol{s}$ and $\boldsymbol{d}$ show that the diagonal section $h(x)$ of the reliability function encodes exactly the signature (or, equivalently, the tail signature), no more, no less. Even though the latter can be computed from vector $\boldsymbol{d}$ using (14) and (18), we will now see how we can compute it by direct and simple algebraic manipulations of function $h(x)$.

Let f be a univariate polynomial of degree less than or equal to n,

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.$$

The n*-reflected* of f is the polynomial $R^n f$ obtained from f by switching the coefficients of x^k and x^{n-k} for $k = 0, \ldots, n$; i.e.

$$(R^n f)(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n,$$

or, equivalently, $(R^n f)(x) = x^n f(1/x)$.

Substituting (7) into (13), we obtain (see also [4])

$$h(x) = \sum_{k=0}^{n} \overline{S}_{n-k} \binom{n}{k} x^k (1-x)^{n-k}. \tag{21}$$

https://doi.org/10.1239/jap/1437658611 Published online by Cambridge University Press

From (21), it follows, as it was already observed in [8], that

$$(R^n h)(x+1) = \sum_{k=0}^{n} \binom{n}{k} \overline{S}_k x^k. \tag{22}$$

Thus, $\binom{n}{k}\overline{S}_k$ can be obtained simply by reading the coefficient of x^k in the polynomial function $(R^n h)(x+1)$. Denoting by $[x^k]f(x)$ the coefficient of x^k in a polynomial function $f(x)$, (22) can be rewritten as

$$\binom{n}{k}\overline{S}_k = [x^k](R^n h)(x+1) \quad \text{for } k = 0, \dots, n. \tag{23}$$

From (23), we immediately derive the following algorithm (see also [8]).

Algorithm 5. The following algorithm inputs n and $h(x)$ and outputs $\overline{\mathbf{S}}$.

Step 1. For $k = 0, \dots, n$, let a_k be the coefficient of x^k in the n-degree polynomial $(R^n h)(x+1) = (x+1)^n h(1/(x+1))$.

Step 2. We have $\overline{S}_k = a_k / \binom{n}{k}$ for $k = 0, \dots, n$.

The following proposition yields the analog of (22) and (23) for the signature. Throughout this paper we denote by $h'(x)$ the derivative of $h(x)$.

Proposition 4. *We have*

$$k\binom{n}{k} s_k = [x^{k-1}](R^{n-1}h')(x+1) \quad \textit{for } k = 1, \dots, n, \tag{24}$$

$$\sum_{k=1}^{n} \binom{n}{k} s_k k x^{k-1} = (R^{n-1}h')(x+1), \tag{25}$$

$$\sum_{k=1}^{n} \binom{n}{k} s_k x^k = \int_0^x (R^{n-1}h')(t+1)\,\mathrm{d}t. \tag{26}$$

Proof. By (7), we have $h'(x) = \sum_{j=0}^{n-1}(j+1)d_{j+1}x^j$ and, therefore,

$$\begin{aligned}(R^{n-1}h')(x+1) &= \sum_{j=0}^{n-1}(j+1)d_{j+1}(x+1)^{n-1-j}\\ &= \sum_{j=0}^{n-1}(j+1)d_{j+1}\sum_{k=1}^{n-j}\binom{n-1-j}{k-1}x^{k-1}\\ &= \sum_{k=1}^{n}x^{k-1}\sum_{j=0}^{n-k}\binom{n-1-j}{k-1}(j+1)d_{j+1}.\end{aligned}$$

Thus, the inner sum in the latter expression is the coefficient of x^{k-1} in the polynomial function $(R^{n-1}h')(x+1)$. Dividing this sum by $k\binom{n}{k}$ and then using (18), we obtain s_k. This proves (24) and (25). We obtain (26) by integrating both sides of (25) on the interval $[0, x]$. This completes the proof of the proposition.

https://doi.org/10.1239/jap/1437658611 Published online by Cambridge University Press

From (24) we immediately derive the following algorithm.

Algorithm 6. The following algorithm inputs n and $h(x)$ and outputs $\boldsymbol{s}$.

Step 1. For $k = 1, \dots, n$, let a_{k-1} be the coefficient of x^{k-1} in the $(n-1)$-degree polynomial $(R^{n-1}h')(x+1) = (x+1)^{n-1}h'(1/(x+1))$.

Step 2. We have $s_k = a_{k-1}/(k\binom{n}{k})$ for $k = 1, \dots, n$.

Even though such an algorithm can be easily executed by hand for small n, a computer algebra system can be of great assistance for large n.

Example 4. Consider again the bridge system described in Example 1. We have

$$h'(x) = 4x + 6x^2 - 20x^3 + 10x^4 \quad \text{and} \quad (R^4h')(x) = 10 - 20x + 6x^2 + 4x^3.$$

It follows that $(R^4h')(x+1) = 4x + 18x^2 + 4x^3$ and, hence, $\boldsymbol{s} = (0, \frac{1}{5}, \frac{3}{5}, \frac{1}{5}, 0)$ by Algorithm 6. Indeed, we have, for instance, $s_3 = a_2/(3\binom{5}{3}) = \frac{3}{5}$.

The following proposition, established in [8], provides a necessary and sufficient condition on the system signature for the reliability function to be of full degree (i.e. the corresponding signed domination d_n is nonzero). Here we provide a shorter proof based on (25).

Proposition 5. (See [8].) *Let (C, ϕ) be an n-component semicoherent system with continuous and i.i.d. component lifetimes. Then the reliability function $h(\boldsymbol{x})$ (or, equivalently, its diagonal section $h(x)$) is a polynomial of degree n if and only if*

$$\sum_{k \text{ odd}} \binom{n-1}{k-1} s_k \neq \sum_{k \text{ even}} \binom{n-1}{k-1} s_k.$$

Proof. The function $h(x)$ is of degree n if and only if $h'(x)$ is of degree $n-1$ and this condition holds if and only if $d_n = (1/n)(R^{n-1}h')(0) \neq 0$. By (25), this means that

$$\sum_{k=1}^{n} \binom{n}{k} s_k k(-1)^{k-1} = n \sum_{k=1}^{n} \binom{n-1}{k-1} s_k (-1)^{k-1}$$

is not 0. This completes the proof of the proposition.

The vectors $\boldsymbol{s}$ and $\overline{\boldsymbol{S}}$ can also be computed via their generating functions. The following proposition yields integral formulae for these functions.

Proposition 6. *We have*

$$\sum_{k=0}^{n} \overline{S}_k x^k = \int_0^1 (n+1) R_t^n((R^n h)((t-1)x+1))\,\mathrm{d}t, \tag{27}$$

$$\sum_{k=1}^{n} s_k x^k = \int_0^1 x R_t^{n-1}((R^{n-1}h')((t-1)x+1))\,\mathrm{d}t, \tag{28}$$

where R_t^n is the n-reflection with respect to variable t.

Proof. By (22), we have

$$(R^n h)((t-1)x+1) = \sum_{k=0}^{n} \binom{n}{k} \overline{S}_k (t-1)^k x^k$$

and, hence,

$$R_t^n((R^n h)((t-1)x+1)) = \sum_{k=0}^{n} \binom{n}{k} \overline{S}_k t^{n-k}(1-t)^k x^k.$$

Integrating this expression from $t=0$ to $t=1$ and using the well-known identity

$$\int_0^1 t^{n-k}\,(1-t)^k\,\mathrm{d}t = \frac{1}{(n+1)\binom{n}{k}}, \tag{29}$$

we finally obtain (27). We can prove (28) in a similar way by using (25). This completes the proof of the proposition.

From (28), we immediately derive the following algorithm for the computation of the generating function of the signature. The algorithm corresponding to (27) can be derived similarly.

Algorithm 7. The following algorithm inputs n and $h(x)$ and outputs the generating function of vector $\boldsymbol{s}$.

Step 1. Let $f(t,x) = x(R^{n-1}h')((t-1)x+1)$.

Step 2. We have $\sum_{k=1}^{n} s_k x^k = \int_0^1 (R_1^{n-1} f)(t,x)\,\mathrm{d}t$, where R_1^{n-1} is the $(n-1)$-reflection with respect to the first argument.

The computation of $h(x)$ from $\boldsymbol{s}$ or $\overline{\boldsymbol{S}}$ can be useful if we want to compute the system reliability $h(p)$ directly from the signature and the component reliability p.

We already know that (21) provides the polynomial $h(x)$ in terms of vector $\overline{\boldsymbol{S}}$. The following proposition yields simple expressions of $h(x)$ and $h'(x)$ in terms of vector $\boldsymbol{s}$. This result was already presented in [6, Section 4] and [8, Remark 2] in alternative forms.

Proposition 7. *We have*

$$h'(x) = \sum_{k=1}^{n} s_k k \binom{n}{k} x^{n-k}(1-x)^{k-1}, \tag{30}$$

$$h(x) = \sum_{k=1}^{n} s_k I_x(n-k+1,k) = \sum_{k=1}^{n} s_k \sum_{i=n-k+1}^{n} \binom{n}{i} x^i (1-x)^{n-i}, \tag{31}$$

where $I_x(a,b)$ is the regularized beta function defined for any $a, b, x > 0$, by

$$I_x(a,b) = \frac{\int_0^x t^{a-1}(1-t)^{b-1}\,\mathrm{d}t}{\int_0^1 t^{a-1}(1-t)^{b-1}\,\mathrm{d}t}.$$

Proof. Equation (30) immediately follows from (25). Then, from (29) and (30), we immediately derive the first equality in (31) since $h(x) = \int_0^x h'(t)\,\mathrm{d}t$. The second equality follows from (1) and (21). This completes the proof of the proposition.

The following proposition provides alternative expressions of $h(x)$ and $h'(x)$ in terms of $\boldsymbol{\overline{S}}$ and $\boldsymbol{s}$, respectively.

Proposition 8. *We have*

$$h(x) = ((x\Delta_i + I)^n \overline{S}_{n-i})|_{i=0}, \tag{32}$$

$$h'(x) = n((x\Delta_i + I)^{n-1} s_{n-i})|_{i=0}, \tag{33}$$

where I denotes the identity operator.

Proof. By (17), we have

$$h(x) = \sum_{k=0}^{n} d_k x^k = \sum_{k=0}^{n} \binom{n}{k} x^k (\Delta_i^k \overline{S}_{n-i})|_{i=0},$$

which proves (32) as we can immediately see by formally expanding the binomial operator expression $(x\Delta_i + I)^n$. Equation (33) then immediately follows from (32). This completes the proof of the proposition.

In Proposition 8, we show that the functions $h(x)$ and $h'(x)$ can be computed from difference tables. Setting

$$D_{j,k}(x) = ((x\Delta_i + I)^k \overline{S}_{n-i})|_{i=j} \quad \text{and} \quad d_{j,k}(x) = n((x\Delta_i + I)^{k-1} s_{n-i})|_{i=j-1},$$

we can derive the following algorithms for the computation of $h(x)$ and $h'(x)$.

Algorithm 8. The following algorithm inputs vector $\boldsymbol{\overline{S}}$ and outputs function $h(x)$. It uses the functions $D_{j,k}(x)$ for $k = 0, \ldots, n$ and $j = 0, \ldots, n-k$.

Step 1. For $j = 0, \ldots, n$, set $D_{j,0}(x) := \overline{S}_{n-j}$.

Step 2. For $k = 1, \ldots, n$
For $j = 0, \ldots, n-k$
$D_{j,k}(x) \; := \; xD_{j+1,k-1}(x) + (1-x)D_{j,k-1}(x)$

Step 3. $h(x) := D_{0,n}(x)$.

Algorithm 9. The following algorithm inputs vector $\boldsymbol{s}$ and outputs function $h'(x)$. It uses the functions $d_{j,k}(x)$ for $k = 1, \ldots, n$ and $j = 1, \ldots, n-k+1$.

Step 1. For $j = 1, \ldots, n$, set $d_{j,1}(x) := ns_{n-j+1}$.

Step 2. For $k = 2, \ldots, n$
For $j = 1, \ldots, n-k+1$
$d_{j,k}(x) \; := \; xd_{j+1,k-1}(x) + (1-x)d_{j,k-1}(x)$

Step 3. $h'(x) := d_{1,n}(x)$.

In Table 1 we summarize the main conversion formulae obtained thus far. They are given by the corresponding equation numbers. For instance, formulae to compute $\boldsymbol{s}$ from $\boldsymbol{d}$ or $h(x)$ are given in (18), (24), (26), and (28).

https://doi.org/10.1239/jap/1437658611 Published online by Cambridge University Press

TABLE 1: Conversion formulae.

	$\boldsymbol{d}$ or $h(x)$	$\boldsymbol{s}$	$\overline{\boldsymbol{S}}$
$\boldsymbol{d}$ or $h(x)$	—	(19), (20), (31)	(15), (17), (21), (32)
$\boldsymbol{s}$	(18), (24), (26), (28)	—	(2), (12)
$\overline{\boldsymbol{S}}$	(14), (23), (27)	(1), (12)	—

3.5. Conversions based on the dual structure

We end this section by presenting conversion formulae involving the dual structure of the system. Let $\phi^{\mathrm{D}}\colon \{0,1\}^n \to \{0,1\}$ be the dual structure function defined as $\phi^{\mathrm{D}}(\boldsymbol{x}) = 1 - \phi(\mathbf{1}-\boldsymbol{x})$, where $\mathbf{1}-\boldsymbol{x} = (1-x_1,\dots,1-x_n)$, and let $h^{\mathrm{D}}\colon [0,1]^n \to \mathbb{R}$ be its corresponding reliability function, i.e. $h^{\mathrm{D}}(\boldsymbol{x}) = 1 - h(\mathbf{1}-\boldsymbol{x})$.

Straightforward computations yield the following conversion formulae, where the upper index, D, always refers to the dual structure and δ stands for the Kronecker delta:

$$
\begin{aligned}
d_k^{\mathrm{D}} &= \delta_{k,0} - (-1)^k \sum_{j=k}^{n} \binom{j}{k} d_j \quad \text{for } k=0,\dots,n,\\
d_k &= \delta_{k,0} - (-1)^k \sum_{j=k}^{n} \binom{j}{k} d_j^{\mathrm{D}} \quad \text{for } k=0,\dots,n,\\
\overline{S}_k &= 1 - \overline{S}_{n-k}^{\mathrm{D}} = 1 - \sum_{j=0}^{k} \frac{\binom{k}{j}}{\binom{n}{j}} d_j^{\mathrm{D}} \quad \text{for } k=0,\dots,n,\\
s_k &= s_{n-k+1}^{\mathrm{D}} = \sum_{j=1}^{k} \frac{\binom{k-1}{j-1}}{\binom{n}{j}} d_j^{\mathrm{D}} \quad \text{for } k=1,\dots,n,\\
d_k^{\mathrm{D}} &= \delta_{k,0} - \binom{n}{k} (\Delta_i^k S_i)|_{i=0} \quad \text{for } k=0,\dots,n,\\
d_k^{\mathrm{D}} &= \binom{n}{k} (\Delta_i^{k-1} s_i)|_{i=1} \quad \text{for } k=1,\dots,n.
\end{aligned}
\tag{34}
$$

Recall that ϕ_k provides the number of path sets of size k. Combining (13) with (22), we obtain the identity $\sum_{k=0}^{n} \phi_{n-k} x^k = (R^n h)(x+1)$, from which we immediately derive the following generating function

$$
\sum_{k=0}^{n} \phi_k x^k = R^n((R^n h)(x+1)) = (x+1)^n h\biggl(\frac{x}{x+1}\biggr).
$$

Note that this function can also be obtained by using (13), (34), and the dual version of (22). Indeed, we have

$$
\sum_{k=0}^{n} \phi_k x^k = \sum_{k=0}^{n} \binom{n}{k} \overline{S}_{n-k} x^k = \sum_{k=0}^{n} \binom{n}{k} x^k - \sum_{k=0}^{n} \binom{n}{k} \overline{S}_k^{\mathrm{D}} x^k = (x+1)^n - (R^n h^{\mathrm{D}})(x+1).
$$

4. The general dependent case

In this section we drop the i.i.d. assumption and consider the general dependent setting, assuming only that there are no ties among the component lifetimes (i.e. $\mathbb{P}\{T_i = T_j\} = 0$ whenever $i \neq j$). As a consequence, the function $h(\boldsymbol{x})$ may no longer express the reliability of the system in terms of the component reliabilities.

Two concepts of signature emerge in this general setting. First, we can consider the *structure signature*, i.e. the n-vector $\boldsymbol{s} = (s_1, \dots, s_n)$ whose kth coordinate is given by Boland's formula (4). Of course, the conversion formulae and algorithms obtained in Sections 2 and 3 can still be used 'as is', even if the i.i.d. assumption is dropped. Secondly, we can consider the *probability signature*, i.e. the n-vector $\boldsymbol{p} = (p_1, \dots, p_n)$ whose kth coordinate is given by $p_k = \mathbb{P}\{T_{\mathrm{S}} = T_{\{k:\, n\}}\}$.

We now elaborate on this latter case, and show that a modification of the structure function enables us to formally extend almost all the conversion formulae and algorithms obtained in Sections 2 and 3 to the general dependent setting.

It was recently shown [7] that

$$p_k = \sum_{A \subseteq C, |A|=n-k+1} q(A)\phi(A) - \sum_{A \subseteq C, |A|=n-k} q(A)\phi(A), \tag{35}$$

where the function $q\colon 2^{[n]} \to \mathbb{R}$, called the *relative quality function* associated with the system, is defined by

$$q(A) = \mathbb{P}\Big\{\max_{i \notin A} T_i < \min_{i \in A} T_i\Big\},$$

and has the property $\sum_{|A|=k} q(A) = 1$ for $k = 0, \dots, n$. Thus, for any subset $A \subseteq C$, the number $q(A)$ is the probability that the best $|A|$ components of the system are precisely those in A.

In the special case when the component lifetimes are i.i.d., or even exchangeable, the number $q(A)$ is exactly $1/\binom{n}{|A|}$ and, therefore, by comparing (4) and (35), we immediately see that the vector $\boldsymbol{p}$ then reduces to $\boldsymbol{s}$. As mentioned in [7], this observation motivates the introduction of the *normalized relative quality function* $\tilde{q}\colon 2^{[n]} \to \mathbb{R}$, defined by $\tilde{q}(A) = \binom{n}{|A|} q(A)$. We then have $\tilde{q}(A) = 1$ whenever the component lifetimes are i.i.d. or exchangeable.

Following a suggestion by Pierre Mathonet (University of Liège), we now assign to the system a pseudo-structure function $\psi\colon 2^{[n]} \to \mathbb{R}$ defined so as to have

$$\sum_{A \subseteq C, |A|=k} \frac{1}{\binom{n}{|A|}} \psi(A) = \sum_{A \subseteq C, |A|=k} q(A)\,\phi(A) \quad \text{for } k = 0, \dots, n. \tag{36}$$

Definition 1. Let (C, ϕ) be an n-component system with relative quality function q. The *q-structure function* associated with the system is the set function $\psi\colon 2^{[n]} \to \mathbb{R}$ defined by

$$\psi(A) = \tilde{q}(A)\phi(A) = \begin{cases} \dbinom{n}{|A|} q(A) & \text{if } A \text{ is a path set,} \\ 0 & \text{otherwise.} \end{cases}$$

It is clear that ψ reduces to ϕ whenever the component lifetimes of the system are i.i.d. or exchangeable. In the general dependent case, the function ψ is a pseudo-Boolean function, i.e. a function from $\{0, 1\}^n$ to $\mathbb{R}$. As such, it has the following multilinear form:

$$\psi(\boldsymbol{x}) = \sum_{A \subseteq C} \psi(A) \prod_{i \in A} x_i \prod_{i \in C \setminus A} (1 - x_i), \qquad \boldsymbol{x} \in \{0, 1\}^n.$$

Just as $h(\boldsymbol{x})$ is the multilinear extension of $\phi(\boldsymbol{x})$, we can also define the multilinear extension $g\colon [0,1]^n \to \mathbb{R}$ of $\psi(\boldsymbol{x})$; i.e.

$$g(\boldsymbol{x}) = \sum_{A\subseteq C} \psi(A) \prod_{i\in A} x_i \prod_{i\in C\setminus A} (1-x_i), \qquad \boldsymbol{x} \in [0,1]^n.$$

This function can always be expressed in the unique standard multilinear form

$$g(\boldsymbol{x}) = \sum_{A\subseteq C} c(A) \prod_{i\in A} x_i,$$

where, by the Möbius inversion theorem, the coefficient $c(A)$ is given by

$$c(A) = \sum_{B\subseteq A} (-1)^{|A|-|B|} \psi(B).$$

Thus, in this general setting we readily see that (36) holds and, consequently, that (4) immediately extends to (35).

Now, if we also define the values $\overline{P}_k$, ψ_k, and c_k for $k = 0, \dots, n$ as

$$\overline{P}_k = \sum_{i=k+1}^{n} p_i, \qquad \psi_k = \sum_{A\subseteq C, |A|=k} \psi(A), \qquad c_k = \sum_{A\subseteq C, |A|=k} c(A),$$

then we can formally extend all our formulae and algorithms from (1) to (33) *mutatis mutandis* to the general dependent setting.

More formally, we have the following straightforward theorem.

Theorem 1. *Equations (1)–(33) still hold if we replace s_k, $\overline{S}_k$, $h(\boldsymbol{x})$, $h(x)$, $\phi(A)$, $d(A)$, ϕ_k, and d_k with p_k, $\overline{P}_k$, $g(\boldsymbol{x})$, $g(x)$, $\psi(A)$, $c(A)$, ψ_k, and c_k, respectively.*

Let us illustrate how this theorem can be applied. Considering, for instance, (13), from Theorem 1 this equation can be translated in the general setting into

$$\overline{P}_k = \frac{1}{\binom{n}{k}} \sum_{A\subseteq C, |A|=n-k} \psi(A) = \frac{1}{\binom{n}{k}} \psi_{n-k} \quad \text{for } k = 0, \dots, n.$$

Similarly, from (26) we immediately derive the identity

$$\sum_{k=1}^{n} \binom{n}{k} p_k x^k = \int_0^x (R^{n-1} g')(t+1)\,\mathrm{d}t. \tag{37}$$

Example 5. Consider a three-component system whose structure function is given by

$$\phi(x_1, x_2, x_3) = x_1(x_2 \amalg x_3) = x_1x_2x_3 + x_1x_2(1-x_3) + x_1(1-x_2)x_3.$$

The q-structure function is then given by

$$\psi(x_1, x_2, x_3) = x_1x_2x_3 + 3q(\{1,2\})x_1x_2(1-x_3) + 3q(\{1,3\})x_1(1-x_2)x_3.$$

Using (37), we finally obtain

$$\begin{aligned} p_1 &= 1 - q(\{1,2\}) - q(\{1,3\}) = q(\{2,3\}), \\ p_2 &= q(\{1,2\}) + q(\{1,3\}), \\ p_3 &= 0. \end{aligned}$$

It is noteworthy that in practice the function $g(\boldsymbol{x})$ is much heavier to handle than the function $h(\boldsymbol{x})$ (consider, for instance, Example 1). Moreover, the function $g(\boldsymbol{x})$ need not be nondecreasing in each argument and, hence, it cannot be easily expressed as a coproduct over the minimal path sets.

However, despite these observations, Theorem 1 shows that this formal extension of the conversion formulae is mathematically elegant and might have theoretical applications.

5. Concluding remarks

We have provided various conversion formulae between the signature and the reliability function for systems with continuous and i.i.d. component lifetimes, and we have extended theses formulae to the general dependent case. This study can be regarded as the continuation of Marichal and Mathonet [8], where (22) and (23), Algorithm 5, and Proposition 5 were already presented and established.

We conclude this paper with the following two observations which are worth particular mention.

- It is a well-known fact that, under the i.i.d. assumption, both the structure signature $\boldsymbol{s} = (s_1, \ldots, s_n)$ and the reliability function $h(\boldsymbol{x})$ are purely combinatorial objects associated with the structure function of the system. As a consequence, the developments and results presented in Sections 2 and 3 are based only on combinatorial and algebraic arguments, and do not really require any stochastic setting, even if such a setting has to be considered to define the component lifetimes.

- The q-structure function of a system as introduced in Definition 1 is simply a convenient transformation of the structure function of the system which enables us to extend (1)–(33) to the general dependent case. Even though the q-structure function $\psi(\boldsymbol{x})$ and its corresponding multilinear extension $g(\boldsymbol{x})$ are heavier to handle than their i.i.d. counterparts $\phi(\boldsymbol{x})$ and $h(\boldsymbol{x})$, Theorem 1 suggests that this extension is interesting more from a conceptual than applied viewpoint.

Acknowledgements

The author would like to thank the anonymous reviewer for timely and helpful suggestions for improving the organization of this paper. This research is supported by the internal research project F1R-MTH-PUL-12RDO2 of the University of Luxembourg.

References

[1] Barlow, R. E. and Iyer, S. (1988). Computational complexity of coherent systems and the reliability polynomial. *Prob. Eng. Inf. Sci.* **2,** 461–469.

[2] Barlow, R. E. and Proschan, F. (1975). *Statistical Theory of Reliability and Life Testing*. Holt, Rinehart and Winston, New York.

[3] Boland, P. J. (2001). Signatures of indirect majority systems. *J. Appl. Prob.* **38,** 597–603.

[4] Boland, P. J., Samaniego, F. J. and Vestrup, E. M. (2003). Linking dominations and signatures in network reliability theory. In *Mathematical and Statistical Methods in Reliability* (Ser. Qual. Reliab. Eng. Statist. **7**), World Scientific, River Edge, NJ, pp. 89–103.

[5] Gertsbakh, I., Shpungin, Y. and Spizzichino, F. (2011). Signatures of coherent systems built with separate modules. *J. Appl. Prob.* **48,** 843–855.

[6] Marichal, J.-L. (2006). Cumulative distribution functions and moments of lattice polynomials. *Statist. Prob. Lett.* **76,** 1273–1279.

[7] Marichal, J.-L. and Mathonet, P. (2011). Extensions of system signatures to dependent lifetimes: explicit expressions and interpretations. *J. Multivariate Anal.* **102,** 931–936.

[8] MARICHAL, J.-L. AND MATHONET, P. (2013). Computing system signatures through reliability functions. *Statist. Prob. Lett.* **83,** 710–717.
[9] RAMAMURTHY, K. G. (1990). *Coherent Structures and Simple Games* (Theory Decision Library Ser. C Game Theory Math. Prog. Operat. Res. **6**). Kluwer, Dordrecht.
[10] SAMANIEGO, F. J. (1985). On closure of the IFR class under formation of coherent systems. *IEEE Trans. Reliab.* **34,** 69–72.
[11] SAMANIEGO, F. J. (2007). *System Signatures and Their Applications in Engineering Reliability* (Internat. Ser. Operat. Res. Manag. Sci. **110**). Springer, New York.