

# Bounding the Results of Arithmetic Operations on Random Variables of Unknown Dependency using Intervals

(Published in: *Reliable Computing* 4 (2) (1998), pp. 147-165)

Daniel Berleant<sup>1</sup> and Chaim Goodman-Strauss<sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering  
Iowa State University  
Ames, IA 50011  
`berleant@iastate.edu`

<sup>2</sup>Department of Mathematics  
University of Arkansas  
Fayetteville, AR 72701

## Abstract

Many real problems involve calculations on random variables, yet precise details about the correlations or other dependency relationships among those variables are often unknown.

For example consider analyzing the *cancer risk* associated with an environmental contaminant. The dependency of an individual's *cumulative exposure* on the less useful (but more obtainable) *current exposure level* will be uncertain. In this and many other cases, data points from which to derive such dependencies are sparse, and obtaining additional data is prohibitively expensive or difficult. Thus manipulating variables whose dependencies are unspecified is a problem of significance.

This paper describes a new approach to bounding the results of arithmetic operations on random variables when the dependency relationship between the variables is unspecified. The bounds enclose the space in which the result's distribution function can be.

## 1 Introduction

Consider finding conservative bounds on the results of combining random variables, when the dependency — if any — of one random variable on the other is unknown. For example, we might want to add two random variables and describe the resulting sum, or multiply them, or apply some more complex function to them. While we wish to conservatively bound the result, we also wish to avoid the problem of “hyperconservative” bounds that are excessively wide (Ferson 1995 [7]).

To set the stage, we first describe the previous work. Then to motivate this paper we compare previous work with the results achievable with our approach, in Section 3. Then we describe our approach. This begins with discretizing the probability distributions representing random variables, in Section 4. Then we review arithmetic operations when the dependency relationship between operands is known, in section 5, to aid in understanding the more general problem of operations when the dependency relationship is unknown, in section 6. The method makes use of linear programming (cf. [10]).

## 2 Previous Work

Monte Carlo simulation is the traditional approach in problems requiring arithmetic operations on independent random variable operands. However Monte Carlo methods have several limitations (Ferson 1996 [3]), including uncertainty about the error in the result. Discretization methods are an alternative approach which can support providing crisp error bounds (Berleant 1993 [1]). Pointers to much of the work involving discretization are given in [1]. To those we now add a few more references. Software for the method described in [1] is illustrated in Berleant (1996 [2]). Williamson and Downs’ (1990 [15]) method has been implemented and extended in Risk Calc, a commercial product (Kuhn and Ferson [14]). Yeh (1989 [16]) and Pesonen and Hyvönen (1996 [13]), using different techniques, bound density functions (PDFs) resulting from arithmetic operations on random variables without relying on the cumulative distributions (CDFs) of the results. Gerasimov et al. (1991 [9]) describe operations on histograms, which can represent density functions.

In recent years the problem of bounding results when operands cannot be assumed to be independent or to have any other dependency relation-

ship has received increasing attention. This is because applications often are characterized by lack of knowledge about the dependency relationships between random variables that must be combined arithmetically. For example, risk analysis problems are frequently characterized by sparse data, unknown dependency relationships among variables, and excessively conservative traditional analyses (e.g. Ferson 1995 [4] [5]). Traditional risk analysis techniques such as worst case analysis and Monte Carlo simulation can provide problematic results, and new approaches that avoid their shortcomings are needed (e.g., [3, 7, 5, 4] and their references).

Previous work by others in this area has used copulas (see Nelsen 1995 [12] for a relatively readable introduction to copulas). Frank et al. (1987 [8]) show how to bound sums of random variables, whose dependency relationship is unknown, using copulas. Williamson and Downs extend Frank et al. by developing algorithms for bounding the results of adding, subtracting, multiplying, and dividing random variables. Risk Calc extends the Williamson and Downs approach to additional arithmetic operations as part of a commercial package.

### 3 Comparison of the Interval Approach and the Copula Approach

A comparison with the aforementioned approach to arithmetic operations on random variables of unknown dependency, which we call Copula Based Dependency Bounds Analysis, helps to motivate the new approach we present, which we call Interval Based Dependency Bounds Analysis.

The copula based approach is relatively well developed and available in a commercial product, while the interval based approach is currently not. However the interval based approach is promising for several reasons.

1. Like the copula based approach (Ferson et al. [6]) it allows for conservative representation of a random variable (using a discretization which is information-losing rather than approximating), not only when a distribution function for that variable is given but also when a representation other than a density function is appropriate (e.g. Kreinovich 1993 [11]), as is often true when data are limited [6]. For example, Kolmogorov-Smirnov bounds can be represented rather easily because they consist of staircase-shaped upper and lower bounding curves, which are close to the interval based representation.

2. It promises to allow tighter bounds than the copula based approach on the results of evaluating expressions in which random variables appear multiple times. For example in the expression  $(X^2 - YX)$  the copula based approach assumes no knowledge of the dependency relationship between  $X^2$  and  $YX$  whereas the interval based approach, which relies on interval calculations, could use interval techniques for removing excess width in the interval calculations (e.g., if  $X \in [0, 1]$  and  $Y \in [0, 1]$ , then  $X^2 - YX \in [-0.25, 1]$ ). More work is needed to further develop this potential.
3. It can use partial information about dependencies among input operands. A simple example will be shown later (Section 7), but more work is needed in this area. With the copula based approach, it is not currently known how to use such partial information.
4. It promises to allow operands with infinite supports, which often occur, e.g. in the normal and exponential curves, because interval arithmetic is easily extended to handle possible distribution end points of  $\pm\infty$ .<sup>1</sup> The copula based approach has not been extended to allow infinities.
5. It uses algorithms that work when samples of operands are negative, or can be either negative or positive, whereas in the copula based approach such cases would require modified algorithms which, while they could probably be developed, do not exist at present.
6. The intuitions behind it seem easier to understand.

Interval Based Dependency Bounds Analysis thus has significant potential to address important issues in a broader context: the need to infer as much as possible from partial information about the world. Its basic techniques are now available and described here.

## 4 Representing Random Variables

Figure 1 shows how the distribution functions for random variables are represented. By thus discretizing the distribution functions associated with

---

<sup>1</sup>Here are some representative examples. Addition:  $[2, \infty) + [3, 4] = [5, \infty)$ ,  $(-\infty, 1] + [1, 2] = (-\infty, 3]$ ,  $(\infty, \infty) + (-\infty, 2] = (-\infty, \infty)$ , etc. Subtraction: add the negation. Negation:  $-[1, \infty) = (-\infty, -1]$ , etc. Multiplication:  $[1, \infty) * (-\infty, -1] = (-\infty, -1]$ , etc. Division: multiply by the reciprocal. Reciprocation:  $\frac{1}{[2, 3]} = [\frac{1}{3}, \frac{1}{2}]$ ,  $\frac{1}{[-1, 1]} = (-\infty, \infty)$ , etc.

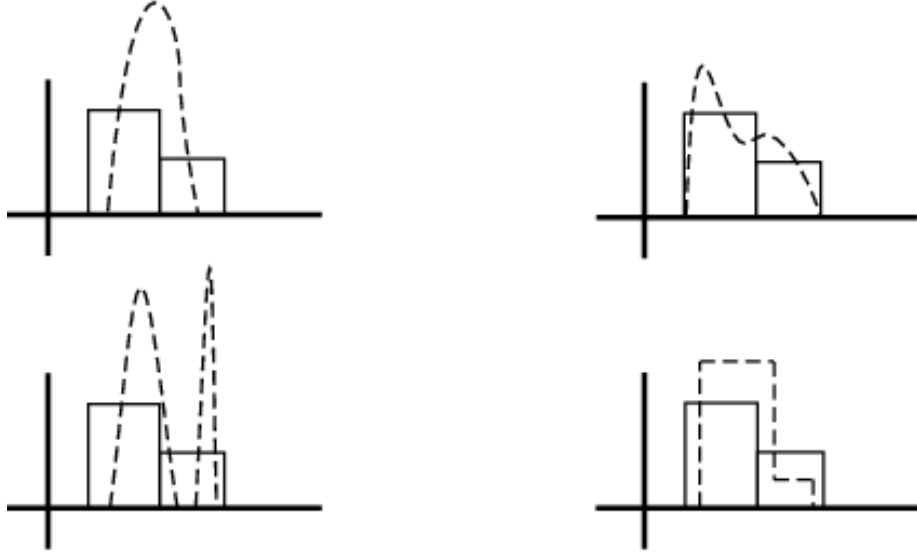


Figure 1: The dotted lines show a few members of the family of density functions corresponding to the same discretization (from Berleant 1993 [1]).

random variables and then manipulating them numerically, a wide variety of distributions can be handled. Operations on distributions become operations on discretizations, which the algorithms reduce to interval operations on the bars of the histogram-like discretizations.

Since we will automatically derive conservative bounds during the course of the calculations, the inputs to those calculations should also be conservative. Fortunately, these inputs are discretizations (Figure 1), which conservatively represent the original undiscretized curves. Each bar in a discretization associates a probability mass with an interval, but does not constrain how the mass is distributed within the interval. Thus a discretization of a given distribution also represents the other members of a family of distributions, and is rigorous in that the family includes the original distribution but at the price of some information loss in that the original distribution cannot be recovered.<sup>2</sup>

---

<sup>2</sup>We do not call the discretization a histogram, because unlike a histogram the bars are allowed to overlap each other, and often do overlap when a discretization is the result of multiplying or dividing other discretizations. Furthermore, a bar leaves undefined how its mass is distributed over the interval of its domain on the  $x$ -axis (Figure 1).

## 5 Arithmetic on Random Variables with Known Dependencies

This section reviews interval based arithmetic on random variables (described in detail in Berleant (1993 [1])), using a basic example. The random variable operands are represented using discretizations. We assume they are independent, perfectly positively correlated, or perfectly negatively correlated. Later we will generalize to the case where the dependency relationship is unknown.

Consider multiplying the two discretized random variables shown in Figure 2. If the operands are independent, the chance that a sample of  $X$  is within the interval of a particular bar of its discretization is the probability mass of that bar, and the chance that a sample of  $Y$  is *also* within the interval of a particular bar of *its* discretization, is simply the product of the probability masses of the two bars. This probability mass is associated with the interval that results from applying the arithmetic operation of interest (multiplication in this case) to the intervals of the two bars. This leads to the joint distribution matrix of Figure 2, middle row, left. A joint distribution matrix contains  $m \times n$  entries if the two discretizations at its margins contain  $m$  and  $n$  bars respectively.

If the random variables are perfectly positively rank correlated, then the value of a sample of one of them determines the other — the lowest possible sample of one will be associated with the lowest possible sample of the other, and, more generally, the value at any fractile of one will be associated with the value of the other at the same fractile. This situation is shown in the joint distribution matrix of Figure 2, middle.

If the random variables are perfectly negatively correlated, each fractile  $r$  of one is associated with fractile  $1 - r$  of the other. This situation is shown in the joint distribution matrix of Figure 2, right.

A joint distribution matrix represents the result of an arithmetic operation on random variables, and constrains considerably (but not completely) the cumulative distribution function for the result. A matrix is consistent with any cumulative distribution in a family of cumulative distributions. The bounding curves of those families are shown in Figure 2, bottom. An upper bounding curve is obtained by assuming the probability mass in each matrix entry is concentrated at the low end of its interval, thereby making the curve rise as quickly as possible. A lower curve assumes each probability mass is concentrated at the high end of its corresponding interval, thereby

making the curve rise as slowly as possible.

## 6 Arithmetic on Random Variables with Unspecified Dependencies

The joint distribution matrices shown in Figure 2 illustrate that the probability value in each entry in the matrix is affected by the dependency relationship between its marginals, which are shown along the right column and bottom row of each table. Indeed, the configuration of probability values defines the dependency relationship expressed by a joint distribution matrix. However the *interval* associated with each entry is unaffected by the dependency relationship. Rather, the intervals are determined by the arithmetic operation performed on the intervals in the marginals.

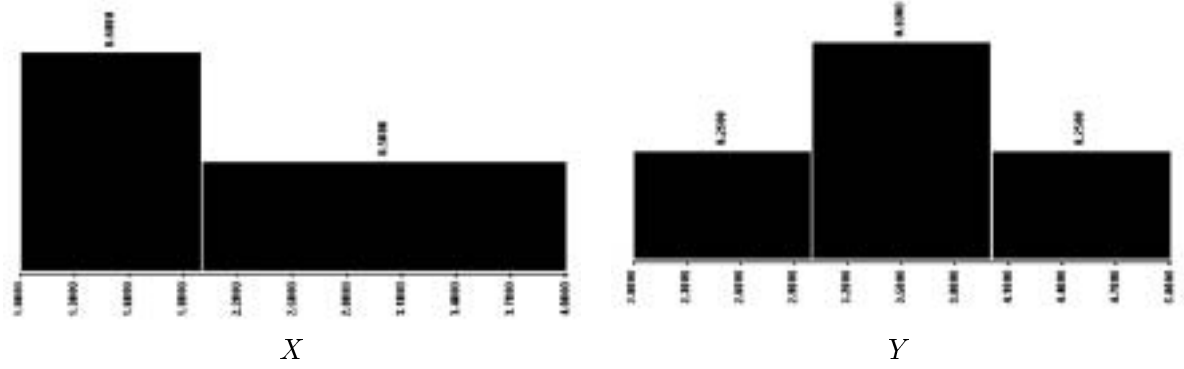
Different dependency relationships imply different assignments of probability masses to joint distribution matrix cells, which in turn imply different pairs of bounding curves. We wish to bound the space of all such curves. These bounds are called *dependency bounds* and they conservatively describe the space of possible results of an operation on marginal discretizations when their dependency relationship is unknown. Each point on an upper or lower dependency bound will also be a point on an upper or lower bounding curve resulting from some dependency relationship between the operands. However, this does not imply that a bounding curve from any single dependency relationship would be the same as the upper or lower dependency bound.

To derive the two dependency bounds we must know, for each point  $z$  on the domain of the random variable  $Z$  representing the result of an operation, the highest and lowest cumulative probabilities that are possible for any dependency relationship between the operands. This requires first finding a pair of joint distribution matrices, one providing the highest possible cumulative probability for  $z$ , and another providing the lowest. After calculating the maximum probability at a finite number of selected values of  $z$ , it is then straightforward to graph the resulting high dependency bound, and similarly for the low dependency bound.

We now describe the method and apply it to an example.

### 6.1 The Joint Distribution Matrix

**Definition 1** A joint distribution matrix (for marginal discretizations  $X$  and  $Y$  with bar probability masses  $\{x_i \mid 1 \leq i \leq m\}$  and  $\{y_j \mid 1 \leq j \leq n\}$



$xy \in [2, 6]$ $p = 1/8$	$xy \in [4, 12]$ $p = 1/8$	$y \in [2, 3]$ $p = 1/4$
$xy \in [3, 8]$ $p = 1/4$	$xy \in [6, 16]$ $p = 1/4$	$y \in [3, 4]$ $p = 1/2$
$xy \in [4, 10]$ $p = 1/8$	$xy \in [8, 20]$ $p = 1/8$	$y \in [4, 5]$ $p = 1/4$
$x \in [1, 2]$ $p = 1/2$	$x \in [2, 4]$ $p = 1/2$	$\leftrightarrow$ $X$

( $X$  and  $Y$  independent)

$xy \in [2, 6]$ $p = 1/4$	$xy \in [4, 12]$ $p = 0$	$y \in [2, 3]$ $p = 1/4$
$xy \in [3, 8]$ $p = 1/4$	$xy \in [6, 16]$ $p = 1/4$	$y \in [3, 4]$ $p = 1/2$
$xy \in [4, 10]$ $p = 0$	$xy \in [8, 20]$ $p = 1/4$	$y \in [4, 5]$ $p = 1/4$
$x \in [1, 2]$ $p = 1/2$	$x \in [2, 4]$ $p = 1/2$	$\leftrightarrow$ $X$

( $X$  and  $Y$  positively correlated)

$xy \in [2, 6]$ $p = 0$	$xy \in [4, 12]$ $p = 1/4$	$y \in [2, 3]$ $p = 1/4$
$xy \in [3, 8]$ $p = 1/4$	$xy \in [6, 16]$ $p = 1/4$	$y \in [3, 4]$ $p = 1/2$
$xy \in [4, 10]$ $p = 1/4$	$xy \in [8, 20]$ $p = 0$	$y \in [4, 5]$ $p = 1/4$
$x \in [1, 2]$ $p = 1/2$	$x \in [2, 4]$ $p = 1/2$	$\leftrightarrow$ $X$

( $X$  and  $Y$  negatively correlated)

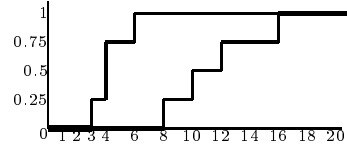
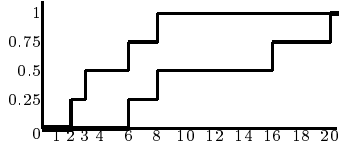
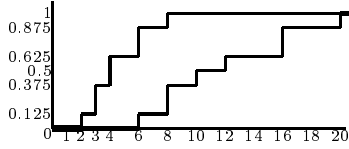


Figure 2: The left discretization represents the density function of a random variable  $X$  whose samples  $x$  satisfy  $\{p(x \in [1, 2]) = \frac{1}{2}, p(x \in [2, 4]) = \frac{1}{2}\}$ . Similarly, the right discretization represents a random variable  $Y$ . Together they have a discrete joint distribution matrix with marginals  $X$  and  $Y$ . Each cell in a discrete joint distribution matrix contains a probability mass determined by the dependency relationship between  $X$  and  $Y$ . Since we are calculating the product of  $X$  and  $Y$  in this case, each cell is also annotated with the product of its associated marginal intervals. For  $X$  and  $Y$  independent (leftmost matrix) the mass of a cell will be the product of the masses of the corresponding bars of marginals  $X$  &  $Y$  (see [1] for details). For  $X$  and  $Y$  perfectly correlated (that is, the value of sample  $x$  determines the value of sample  $y$ ), positively or negatively, the masses are shown in the middle and rightmost matrices, respectively. Integrating these matrices to get the low bounding curve (for the probability mass in each entry concentrated at the high end of its interval) and high bounding curve (for the probability mass in each entry concentrated at the low end of its interval) of the cumulative distribution for  $X \times Y$  produces graphs for the case of  $X$  and  $Y$  independent (lower left [1]), positively correlated (lower middle), and negatively correlated (lower right). 8



respectively) is an  $m \times n$  matrix with entries  $0 \leq p_{ij} \leq 1$  such that

$$x_i = \sum_{j=1}^n p_{ij}$$

(adding up entries in a row) and

$$y_j = \sum_{i=1}^m p_{ij}$$

(adding up entries in a column).

**Example 1** Here is a  $5 \times 4$  joint distribution matrix:

					X
	.04	.02	.03	.01	.10
	.02	.08	.04	.01	.15
	.02	.05	.08	.10	.25
	0	.01	.09	.20	.30
	.02	.04	.06	.08	.20
Y	.10	.20	.30	.40	

The probabilities in a joint distribution matrix might happen to each be associated with an interval, as occurred in Figure 2, depending on the circumstances.

## 6.2 Initializing the Matrix

We define and provide an example of a “vertex” joint distribution matrix, then show how to find one from which to start a search for the extreme ones useful in obtaining dependency bounds. We will see that dependency bounds are obtainable from vertex matrices, allowing us to ignore the infinite number of non-vertex matrices when searching for dependency bounds.

**Definition 2** A vertex is a joint distribution matrix such that at least  $(m - 1)(n - 1)$  of its entries are equal to zero.

**Example 2** For example, the following joint distribution matrix is a vertex:

					$X$
	.10	0	0	0	.10
	0	.15	0	0	.15
	0	0	.15	.10	.25
	0	0	0	.30	.30
	0	.05	.15	0	.20
$Y$	.10	.20	.30	.40	

We wish to generate a vertex given just its marginal discretizations. The Cross Off Algorithm does this:

**Algorithm 1** [The Cross Off Algorithm] (Completes a matrix by repeatedly filling in a remaining row or column with zeros and crossing it off from further consideration.)

1. Given: distributions represented as discretizations  $X$  with  $\{x_i \mid 1 \leq i \leq m\}$  and  $Y$  with  $\{y_j \mid 1 \leq j \leq n\}$ .
2. Assign to each matrix entry  $p_{ij}$  the placeholder symbol “\*.”
3. Initialize variables  $x'_i$  to the corresponding constants  $x_i$  and variables  $y'_j$  to the corresponding constants  $y_j$ .
4. Repeat until all placeholders are replaced by a number:
  - (a) Let  $x'_I := \max_i x'_i$  and  $y'_J := \max_j y'_j$ .
  - (b) If  $x'_I \geq y'_J$ :
    - i.  $p_{IJ} := y'_J$
    - ii. For each  $p_{iJ}$ , if its value is \* set it to 0
  - (c) If  $x'_I \leq y'_J$ :
    - i.  $p_{IJ} := x'_I$
    - ii. For each  $p_{Ij}$ , if its value is \* set it to 0
  - (d)  $x'_I := x'_I - \min(x'_I, y'_J)$
  - (e)  $y'_J := y'_J - \min(x'_I, y'_J)$

**Lemma 1** *The Cross Off Algorithm produces a vertex, given  $X$  and  $Y$ , after no more than  $n + m - 1$  iterations.*

This is true because there are only  $n + m$  rows and columns to be crossed off, and the last entry to be filled crosses off both the last remaining row and the last remaining column. Each crossed off row or column satisfies and removes from further consideration the constraint imposed by the marginal probability mass associated with it.

**Example 3** Finding a vertex. The left column contains snapshots after each step 4a; the right column contains snapshots after each step 4e.

					$X$					$X'$		
		*	*	*	*	.10		*	*	*	*	.10
		*	*	*	*	.15		*	*	*	*	.15
		*	*	*	*	.25		*	*	*	*	.25
		*	*	*	*	.30		0	0	0	.30	0
		*	*	*	*	.20		*	*	*	*	.20
$Y$	.10	.20	.30	.40				$Y'$	.10	.20	.30	.10
						$X'$						$X'$
		*	*	*	*	.10		*	*	*	*	.10
		*	*	*	*	.15		*	*	*	*	.15
		*	*	*	*	.25		0	0	.25	0	0
		0	0	0	.30	0		0	0	0	.30	0
		*	*	*	*	.20		*	*	*	*	.20
$Y'$	.10	.20	.30	.10				$Y'$	.10	.20	.05	.10
						$X'$						$X'$
		*	*	*	*	.10		*	0	*	*	.10
		*	*	*	*	.15		*	0	*	*	.15
		0	0	.25	0	0		0	0	.25	0	0
		0	0	0	.30	0		0	0	0	.30	0
		*	*	*	*	.20		0	.20	0	0	0
$Y'$	.10	.20	.05	.10				$Y'$	.10	0	.05	.10
						$X'$						$X'$
		*	0	*	*	.10		0	0	*	*	.10
		*	0	*	*	.15		.10	0	*	*	.05
		0	0	.25	0	0		0	0	.25	0	0
		0	0	0	.30	0		0	0	0	.30	0
		0	.20	0	0	0		0	.20	0	0	0
$Y'$	.10	0	.05	.10				$Y'$	0	0	.05	.10
						$X'$						$X'$
		0	0	*	*	.10		0	0	0	.10	0
		.10	0	*	*	.05		.10	0	*	0	.05
		0	0	.25	0	0		0	0	.25	0	0
		0	0	0	.30	0		0	0	0	.30	0
		0	.20	0	0	0		0	.20	0	0	0
$Y'$	0	0	.05	.10				$Y'$	0	0	.05	0
						$X'$						$X'$
		0	0	0	.10	0		0	0	0	.10	0
		.10	0	*	0	.05		.10	0	.05	0	0
		0	0	.25	0	0		0	0	.25	0	0
		0	0	0	.30	0		0	0	0	.30	0
		0	.20	0	0	0		0	.20	0	0	0
$Y'$	0	0	.05	0				$Y'$	0	0	0	0

### 6.3 Maximizing and Minimizing the Matrix

**Definition 3** Let  $A$  be a subset of the entries in the joint distribution matrix:

$$A \subseteq \{(i, j) \mid 1 \leq i \leq n, 1 \leq j \leq m\}$$

$A$  would be chosen to include each entry whose associated interval's low bound is at or below a number  $z$  at which the cumulative probability of the result of the arithmetic operation is to be maximized, to get the upper dependency bound at  $z$  (or whose associated interval's *high* bound is at or *above* the number  $z$  at which the cumulative probability is to be *minimized* to get the *lower* dependency bound at  $z$ ).

For example, consider multiplying  $X \times Y$  with samples  $x$  and  $y$ , from Figure 2. The maximum cumulative probability at  $x \times y = z = 4$  is derived from  $A$  being those entries associated with bolded intervals:

<b><math>xy \in [2, 6]</math></b> $p_{11} = 1/4$	<b><math>xy \in [4, 12]</math></b> $p_{12} = 0$	$y \in [2, 3]$ $p = 1/4$
<b><math>xy \in [3, 8]</math></b> $p_{21} = 1/4$	$xy \in [6, 16]$ $p_{22} = 1/4$	$y \in [3, 4]$ $p = 1/2$
<b><math>xy \in [4, 10]</math></b> $p_{31} = 0$	$xy \in [8, 20]$ $p_{32} = 1/4$	$y \in [4, 5]$ $p = 1/4$
$x \in [1, 2]$ $p = 1/2$	$x \in [2, 4]$ $p = 1/2$	$\leftarrow \quad \uparrow$ $X \quad Y$

Given this specification for  $A$ , a set of initial probabilities for the variables  $p_{ij}$ , such as those shown, must be modified if necessary to maximize their sum over  $A$  while remaining consistent with the marginals  $X$  and  $Y$ .

Similarly, the *minimum* cumulative probability, at  $x \times y = z = 10$ , is derived from  $A$  being these entries associated with bolded intervals:

$xy \in [2, 6]$ $p_{11} = 1/4$	<b><math>xy \in [4, 12]</math></b> $p_{12} = 0$	$y \in [2, 3]$ $p = 1/4$
$xy \in [3, 8]$ $p_{21} = 1/4$	<b><math>xy \in [6, 16]</math></b> $p_{22} = 1/4$	$y \in [3, 4]$ $p = 1/2$
<b><math>xy \in [4, 10]</math></b> $p_{31} = 0$	<b><math>xy \in [8, 20]</math></b> $p_{32} = 1/4$	$y \in [4, 5]$ $p = 1/4$
$x \in [1, 2]$ $p = 1/2$	$x \in [2, 4]$ $p = 1/2$	$\leftarrow \quad \uparrow$ $X \quad Y$

By maximizing the sum  $P$  of the  $p_{ij}$ 's in this  $A$ , we minimize the sum  $1 - P$  for  $A$ 's complement.  $1 - P$  is then the minimum cumulation desired. We now describe how to maximize, given  $A$ .

**Definition 4** *Given two joint distribution matrices  $d$  and  $d'$  with entries  $\{p_{ij}\}$  and  $\{p'_{ij}\}$ , respectively, if*

$$\sum_{(i,j) \in A} p_{ij} > \sum_{(i,j) \in A} p'_{ij}$$

*we say  $d$  is higher than  $d'$  and  $d'$  is lower than  $d$ ; these sums are the heights of the matrices.*

We would like to find, given  $A$  as described in definition 3, matrices  $\bar{d}$  and  $\underline{d}$ , where  $\bar{d}$  is as high as any matrix consistent with some dependency relationship between marginals  $X$  and  $Y$ , and  $\underline{d}$  is as low as any such matrix.

Algorithm 2 does this. But first we note as support a definition and two lemmas.

**Definition 5** *Two vertices are adjacent if they have at least  $(m-1)(n-1)-1$  zeros in common among their entries.*

We search for  $\bar{d}$  (or  $\underline{d}$ ) by repeatedly traveling from one adjacent vertex to another.

**Lemma 2** *Given any vertex  $d_0$ , there is a chain of vertices  $d_0, d_1, d_2, \dots, d_k, \dots, d_K = \bar{d}$  such that each vertex in the chain is adjacent to, and higher, than the preceding vertex.*

A similar lemma provides for a chain ending in  $\underline{d}$ .

The adjacent vertices are found using standard linear programming techniques, along with the following lemma:

**Lemma 3** *Given the values of  $(m-1)(n-1)$  of the entries in a joint distribution matrix such that no column or row has all entries specified, then the values of the other entries are determined.*

See the appendix for the proof.

Now we can present the main algorithm. This algorithm always works to construct a chain. The technique used is linear programming (cf. [10]). (See Appendix for the proof.)

**Algorithm 2** [Main Algorithm]

1. Using the Cross Off Algorithm, produce an initial vertex  $d_0$ .
2. Initialize  $k := 0$  and begin the loop:
  - (a) Specify the locations of the zeros for each potential adjacent vertex.
  - (b) For each potential adjacent vertex, apply the proof of lemma 3 (in the Appendix) to find the values of the remaining entries. Discard any resulting matrices containing any  $p_{ij}$  not satisfying  $0 \leq p_{ij} \leq 1$ .
  - (c) Calculate the heights of the remaining adjacent vertices (using Definition 4) and let  $d_{k+1}$  be an adjacent vertex as high or higher than any other adjacent vertex.
  - (d) If  $d_{k+1}$  is higher than  $d_k$  then set  $k := k + 1$  and repeat the loop, else set  $\bar{d} = d_K = d_k$  and stop.

A similar algorithm locates the vertex  $\underline{d}$ .

For example, applying algorithm 2 to maximize the height of the matrix on the left, for  $A$  containing those bolded entries implied by  $z = 4$ , produces a matrix with maximal height such as the one on the right:

$xy \in [2, 6]$ $p = 1/4$	$xy \in [4, 12]$ $p = 0$	$y \in [2, 3]$ $p = 1/4$	$xy \in [2, 6]$ $p = 0$	$xy \in [4, 12]$ $p = 1/4$	$y \in [2, 3]$ $p = 1/4$
$xy \in [3, 8]$ $p = 1/4$	$xy \in [6, 16]$ $p = 1/4$	$y \in [3, 4]$ $p = 1/2$	$xy \in [3, 8]$ $p = 1/2$	$xy \in [6, 16]$ $p = 0$	$y \in [3, 4]$ $p = 1/2$
$xy \in [4, 10]$ $p = 0$	$xy \in [8, 20]$ $p = 1/4$	$y \in [4, 5]$ $p = 1/4$	$xy \in [4, 10]$ $p = 0$	$xy \in [8, 20]$ $p = 1/4$	$y \in [4, 5]$ $p = 1/4$
$x \in [1, 2]$ $p = 1/2$	$x \in [2, 4]$ $p = 1/2$	$\leftarrow \quad \uparrow$ $X \quad Y$	$x \in [1, 2]$ $p = 1/2$	$x \in [2, 4]$ $p = 1/2$	$\leftarrow \quad \uparrow$ $X \quad Y$

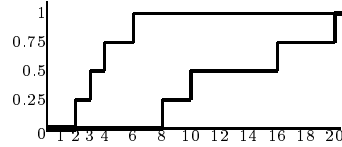
## 6.4 Building the Upper and Lower Dependency Bounds

For each point  $z$  on the domain of the result  $Z$  of the arithmetic operation on the random variables, find the maximum cumulative probability according to the methods of Section 6.3. Connect these maxima to form the upper dependency bound.

Similarly, for each point  $z$  find the minimum cumulative probability according to the methods of Section 6.3. Connect these minima to form the lower dependency bound.

Since the domain of  $Z$  is a segment of the real number line, there are an infinite number of points  $z$  to handle. However, the bounds have a staircase shape, so handling a limited number of different values of  $z$  judiciously chosen using knowledge of the places where the discontinuities in the staircases occur is useful. These discontinuities occur at endpoints of the intervals associated with the entries of the joint distribution matrix.

The next figure shows the dependency bounds that bound the result of multiplying  $X \times Y$ , given  $X$  and  $Y$  as described in Figure 2:



Note that in the general case, a given point on a high or low dependency bound might not also happen to be on a bound obtained under the assumption of perfect positive or negative correlation.

## 7 Incorporating Partial Information about Dependency

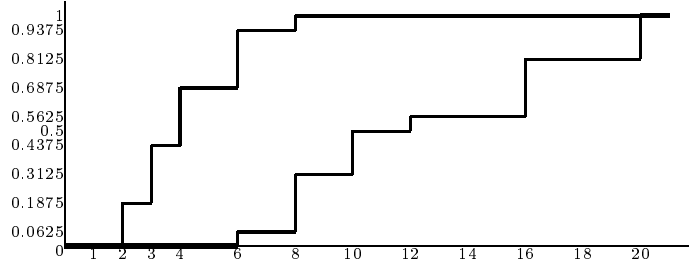
We start with a simple example to illustrate the issue, and end with a call for further work. Consider a situation where:

- 50% of the time, when a pair of samples is drawn, one from  $X$  and one from  $Y$ , they can be assumed independent, and
- 50% of the time we can assume nothing about their kind (or degree) of dependency.

(A situation like this might arise in the actual experiments when the set of samples derives 50% from males and 50% from females.) The bounds for  $Z = X \times Y$  would then be an average of the bounds of the two types:

- bounds for  $X$  and  $Y$  independent, shown in Figure 2 (lower left graph) and
- bounds for  $X$  and  $Y$  that are possibly dependent in some unspecified way, shown in the figure of Section 6.4.

The resulting combined bounds look like this:



Partial information about dependency can take other forms such as “positive to an unknown degree”, various correlation measures, etc. Using partial information about dependencies in the context of Interval Based Dependency Analysis is, we believe, one fruitful direction for further work.

## 8 Acknowledgements

Many individuals in the international interval computations community have made interval computations a field which supports and encourages participation and productivity. Scott Ferson (Applied Biomathematics) provided numerous valuable comments. Mark Arnold (U. of Arkansas, Fayetteville) provided useful advice on linear programming. Vladik Kreinovich provided energy and practical information which helped make the interval community a rewarding one to contribute to. Lizhi Xie provided a valuable critique, and chose to implement this for his thesis.

## Appendix: Proofs

We draw on well known principles from linear algebra and linear programming (cf. [10]).

**Proof of Lemma 3.** Given: the values of  $(m - 1)(n - 1)$  of the entries in a joint distribution matrix such that no row (or column) has all  $n$  (or  $m$ ) entries specified.



There are  $mn - (m - 1)(n - 1) = m + n - 1$  unknown entries remaining, with at least one in each row and column. There is a row or column with only one unknown entry, because if each row and column contained at least two then this would require at least  $\frac{2(m+n)}{2} = m + n$  unknown entries (two for each of the  $m + n$  rows and columns with each appearing in both a row and a column), and this is a contradiction. Since the sum of all the entries in each row or column is specified, one can immediately solve for the value if this value is unknown. Crossing off the row or column we just completed, we now have a smaller matrix. Repeat until done. This algorithm usually works (see the next example).

To prove the lemma, note one can use Gauss-Jordan elimination to solve the  $m + n - 1$  independent equations in  $m + n - 1$  unknowns (each of the  $n + m$  rows and columns produces an equation but since a row or column can be reconstructed given the other rows and columns, there are only  $n + m - 1$  independent equations).

**Example 4** Solve for the starred entries. At each of the steps shown (going across first, then down), each underlined star (“\*”) is about to be given a value since it is the only unknown entry in a row or column (we complete more than one row or column at a time, given the opportunity).

				X					X		
	*	.02	*	.03	.10		*	.02	*	.03	.10
	*	*	.04	.01	.15		*	<u>*</u>	.04	.01	.15
	.02	<u>*</u>	.10	.08	.25		.02	.05	.10	.08	.25
	0	.01	*	<u>*</u>	.30		0	.01	<u>*</u>	.20	.30
	.02	<u>*</u>	.06	.08	.20		.02	.04	.06	.08	.20
Y	.10	.20	.30	.40		Y	.10	.20	.30	.40	

				X					X		
	*	.02	<u>*</u>	.03	.10		<u>*</u>	.02	.01	.03	.10
	<u>*</u>	.08	.04	.01	.15		.02	.08	.04	.01	.15
	.02	.05	.10	.08	.25		.02	.05	.10	.08	.25
	0	.01	.09	.20	.30		0	.01	.09	.20	.30
	.02	.04	.06	.08	.20		.02	.04	.06	.08	.20
Y	.10	.20	.30	.40		Y	.10	.20	.30	.40	

**Linear Programming** The Simplex method of linear programming is widely known and highly developed. At worst, the algorithm has exponen-

tial complexity, however in practice it is usually much faster than that. A general, readable reference is [10].

Here is the idea of the Simplex Algorithm:

- First, a set of linear equalities and inequalities over  $k$  variables describes a polyhedron in  $k$ -dimensional space; that is, the coordinates of the points in the polyhedron are the values of variables that satisfy the equations.
- Second, if we wish to maximize a linear function given these constraints, a maximum will occur at a vertex of the polyhedron.
- Finally, to find the right vertex, one can walk along the edges of the polyhedron: find one vertex, then move to whichever adjacent vertex gives the best value of our function, and repeat. Soon, one has found the “highest” vertex.

We refer the reader to [10] for a more complete discussion of the Simplex Algorithm. Here we merely point out the connections to the problem at hand.

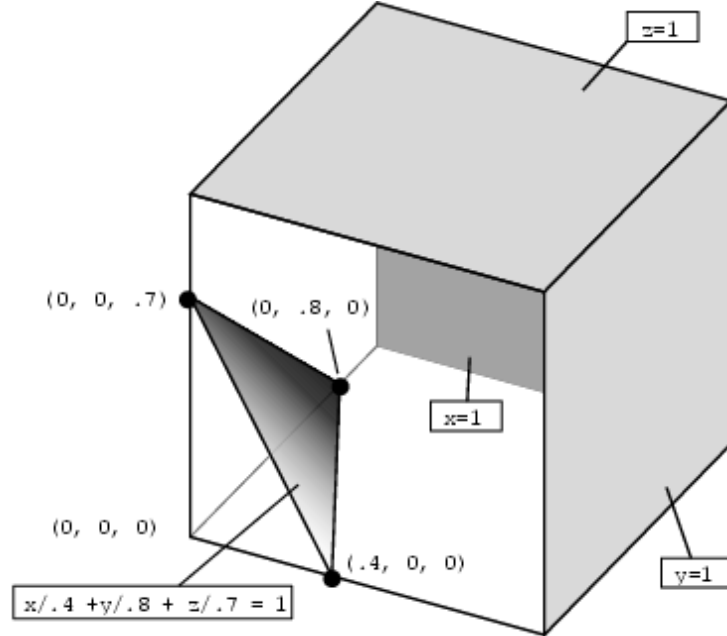


Figure 3: Illustrating the Simplex Algorithm.

First, the joint distribution matrices have  $mn$  entries. We thus can regard them as points in  $mn$  dimensional space. Second, these entries must satisfy the inequalities  $0 \leq p_{ij} < 1$ , strictly less than one because the matrix has more than one entry (else we would have a degenerate case not requiring this analysis). Thus the points corresponding to the joint distribution matrices must lie in the hypercube with corners  $(c_1, \dots, c_{mn})$ ,  $c_i \in [0, 1)$ . In fact, since each  $p_{ij} \neq 1$ , we are restricted to points that are not in the “upper” faces of the hypercube (that is, the faces for which a coordinate has value 1).

In Figure 3 we have drawn a 3-dimensional “hypercube” to illustrate.

Next, the constraint equations

$$x_i = \sum_{j=1}^n p_{ij} \quad y_j = \sum_{i=1}^m p_{ij}$$

each reduce the dimensionality of points satisfying the constraints by 1, just as in 3-D  $x$ - $y$ - $z$ -space a constraint equation such as  $x + y = 1$  restricts the points under consideration to a 2-D plane. Thus we restrict our points to a  $mn - (m + n - 1) = (m - 1)(n - 1)$  dimensional hyperplane intersecting the hypercube; this is because we have  $m + n - 1$  independent linear equations in the  $mn$  variables. Note that this hyperplane misses the upper faces of the hypercube, since no non-degenerate joint distribution matrix will have a  $p_{ij} = 1$ .

To illustrate, in Figure 3 we have sketched a 2-dimensional plane intersecting a cube; the plane is given by the equation shown.

In Figure 3 the 2-D plane intersects the cube forming a 2-D polygon. Similarly, those points on the  $(m - 1)(n - 1)$  dimensional hyperplane that are in the hypercube form a  $(m - 1)(n - 1)$  dimensional polyhedron. The vertices of the 2-D polygon in the 3-D cube of Figure 3 lie on edges of dimensionality  $3 - 2 = 1$ . Similarly, the vertices of the polyhedron in the hypercube lie on hyperedges of dimensionality  $mn - (m - 1)(n - 1)$ , and thus are the points satisfying the constraints such that at least  $(m - 1)(n - 1)$  coordinates are zero (or one, except that no coordinate can be one in our particular problem). Thus, these vertices are the vertices of Definition 2.

Finally, in Figure 3 a linear darkness gradient has been placed on the polyhedron in the cube. This gradient corresponds to the value of some optimization function which is to be maximized or minimized, in this paper the height of a vertex. Notice that the highest and lowest values of the

gradient occur at the vertices of the polyhedron (these values may occur at other points as well, but will certainly occur at vertices).

This establishes Lemma 2 and Algorithm 2.

## References

- [1] Berleant, D., Automatically verified reasoning with both intervals and probability density functions, *Interval Computations* (1993 No. 2) 48–70.
- [2] Berleant, D., Automatically verified arithmetic on probability distributions and intervals, in B. Kearfott and V. Kreinovich, eds., *Applications of Interval Computations*, Kluwer Academic Publishers, 1996, pp. 227–244.
- [3] Ferson, S., What Monte Carlo methods cannot do, *Human and Ecological Risk Assessment* **2** (1996), pp. 990–1007.
- [4] Ferson, S., Quality assurance for Monte Carlo risk assessment, in B. M. Ayyub, ed., *Proceedings of ISUMA-NAFIPS '95*, IEEE Computer Society Press, 1995.
- [5] Ferson, S., and M. A. Burgman, Correlations, dependency bounds and extinction risks, *Biological Conservation* **73** (101–105) (1994).
- [6] Ferson, S., L. Ginzburg, and R. Akçakaya, Whereof one cannot speak: when input distributions are unknown, *Risk Analysis*, accepted.
- [7] Ferson, S. and T. Long, Conservative uncertainty propagation in environmental risk assessments, in J. Hughes, G. Biddinger, and E. Mones, eds., *Environmental Toxicology and Risk Assessment – Third Volume, ASTM STP 1218*, American Society for Testing and Materials, 1995.
- [8] Frank, M. J., R. B. Nelsen, and B. Schweizer, Best-possible bounds for the distribution of a sum — a problem of Kolmogorov, *Probability Theory and Related Fields* **74** (199–211) (1987).
- [9] V. A. Gerasimov, B. S. Dobronets, and M. Yu. Shustrov, “Numerical Calculations for Histogram Operations and their Applications,” *Avtomatika i Telemekhanika*, 1991, Vol. 52, No. 2. (In Russian.)
- [10] Karloff, H., *Linear Programming*, Birkhauser, Boston, 1991.

- [11] Kreinovich, V. and K. Villaverde, Interval, mean value, standard deviation, what else? Group-theoretic approach to describing uncertainty of measurements. Technical report tr93-8, Dept. of Computer Science, U. Texas El Paso, <ftp://cs.utep.edu/pub/reports/tr93-8.tex>. Abstract in *Abstracts for a Workshop on Interval Methods in Artificial Intelligence*, 1993, Lafayette, Louisiana.
- [12] Nelsen, R. B., Copulas, characterization, correlation, and counterexamples, *Mathematics Magazine* **68** (3) (June 1995) 193–198.
- [13] Pesonen, J. and E. Hyvönen, Interval Approach Challenges Monte Carlo Simulation, *Reliable Computing* **2** (2), 1996, 155–160.
- [14] Kuhn, R. and S. Ferson, Risk Calc, Applied Biomathematics, Setauket, NY (commercial software product).
- [15] Williamson, R. and T. Downs, Probabilistic arithmetic i: numerical methods for calculating convolutions and dependency bounds, *International Journal of Approximate Reasoning*, **4** (89–158) (1990).
- [16] Yeh, A., Finding the likely behaviors of static continuous nonlinear systems, presented at the 2nd International Workshop on Artificial Intelligence and Statistics, Jan. 1989. Accepted to corresponding issue of *Annals of Mathematics and Artificial Intelligence*.