Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 192 (2009) 468–478
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Stochastics and Statistics

A matching algorithm for generation of statistically dependent random variables with arbitrary marginals

Nesa Ilich *

University of Calgary, 7128-5 Street NW, Calgary, Alberta, Canada T2K 1C8

Received 8 March 2006; accepted 14 September 2007
Available online 25 September 2007

Abstract

Simulation has gained acceptance in the operations research community as a viable method for analyzing complex problems. While random generation of variables with various marginal distributions has been studied at length, developing ability to preserve a given degree of statistical dependence among them has been lagging behind. This paper includes a short summary of the previous work and a description of the proposed algorithm for efficient re-arranging of generated random variables such that a desired product moment correlation matrix is induced. The proposed approach is different from similar algorithms that induce a desired rank-order correlation among random variables. The algorithm is demonstrated using three numerical examples, one of which also includes a comparison with @RISK commercial package. Its main features are simplicity, ease of implementation and the ability to handle either theoretical or empirical distribution functions.

Keywords: Simulation; Regression; Stochastic processes; Statistical dependence; Correlation

1. Introduction

The need to generate statistically dependent random variables arises in various fields where simulation has proven to be a useful tool, such as finance, production, natural resources management or scheduling. This paper offers a new approach for generating stochastic variables with desired correlation structure and arbitrary marginal distributions aimed at preserving the product moment correlations.

The basic premise for generating dependencies among random variables is to formulate the process of generation of new dependent variables as linear combinations of independent random variables. The notion of ''independent random variables'' here denotes de facto random variables that were already generated in previous steps of the generation process. This approach is known to work well with normal distribution. However, Iman and Conover (1982) identified a difficulty associated with the case of stratified samples where the intent is to preserve the desired bounds on the generated variables. The bounds may be violated since the normalized random component in a regressive process must be unbounded to preserve the desired correlation structure and normal distribution. For example, in some regressive processes variables that are not allowed to be negative may actually become negative after addition of normalized random terms. Also, many processes cannot be adequately represented with a normal distribution, while the use of linear combinations in the generating scheme is only guaranteed to preserve normal distribution. Hence, much of the previous research was based on an attempt to find a universal transformation function from a normal distribution to an arbitrary distribution such that both the distribution properties and desired correlations are preserved. The anticipation was that such a transformation would allow a mathematical transition from a set of correlated variables with normal distribution to a set of correlated variables with arbitrary distribution. The following section contains a survey of published work in this regard.

* Tel.: +1 403 7304480; fax: +1 403 2744031.
E-mail address: nilich@optimal-solutions-ltd.com

0377-2217/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2007.09.024

Early efforts by Mardia (1970) were restricted to finding transformations of bivariate random variables with normal distribution into other distributions. Johnson and Ramberg (1977) also considered marginal distributions as functional transformations of normal distributions. They first transformed the original normally distributed vector to a correlated multivariate normal vector, and then converted it from normal into desired marginal distributions. The problem was that the statistics of the transformed vector (i.e. means, variances and correlation) could not be easily controlled and often deviated from the desired targets. A mathematical treatment for some types of distributions (e.g. lognormal) was developed, however this approach lacked generality since it was not applicable to all marginal distributions. It has been recognized that correlated multivariate random vectors and marginal distributions from the same family of distributions have been covered in the literature (Devorye, 1986; Johnson, 1987). However, correlated random variables with distributions that do not originate from the same family have been given much less attention.

The work of Iman and Conover (1982) provided an algorithm which is used today by two commercial simulation software vendors that provide general purpose simulation models, although with a disclaimer that they are only capable of matching the rank correlations between the generated random variables. A valuable aspect of this approach is that the marginal distributions of the original random vectors remain intact, the algorithm merely provides for a key to re-ordering of the elements of the original vectors. In that sense, this was the first truly "distribution free" algorithm since it guaranteed that the original marginal distributions would not change. In this approach the target correlation matrix contains rank correlations, not the Person correlations. Although rank correlation is most frequently used as a measure of statistical dependence, in certain simulation studies a desired goal is to matching the product moment correlations. In such cases the use of the available commercial packages can only be made under the assumption that matching rank correlations was a sufficiently close approximation to matching the Pearson product moment correlations. This assumption may lead to errors of unacceptable magnitude.

Cario and Nelson (1996, 1997) designed the NORTA ("Normal to Anything") method, which has generated significant interest in the research community, although to this date it has found no application in a general commercial simulation software, but is rather restricted to specific applications related to finance. This algorithm begins with generation of a random vector with multivariate normal distribution, which is then transformed to a random vector with desired marginal distributions and correlation matrix. The authors developed a numerical procedure which determines the correlation structure of the initial normal vector such that the correlation structure of the resulting transformed vector with desired marginal distributions is maintained. However, as documented by Li and Hammond (1975) as well as Lurie and Goldberg (1998), some attempts to generate random vectors with arbitrary marginal distributions and with arbitrary feasible correlation have failed. Ghosh and Henderson suggested an adjustment to the method, and a different adjustment was also suggested by Clemen and Reilly (1999). The recent variants of this approach are the QUARTA method ("Quasi-Random to Anything") from Henderson et al., 2000) and VARTA ("Vector Auto-Regressive to Anything") from Biller and Nelson (2003) which relies on the approximation of input vector with Johnson type distributions. The common feature to all variants is an iterative numerical procedure for matching the correlation structure of the initial random vector until the desired correlation structure of the resulting marginal vector is achieved. To help prevent the iterative procedure from failing, Ghosh and Henderson (2002) resorted to the use of semidefinite programming (SDP). Much of their recent efforts were related to developing a procedure that would determine if the target correlation matrix was feasible for a set of given random variables with arbitrary distribution functions. They introduced the notion of 'NORTA defective correlation matrices' if they are feasible and yet cannot be matched using the NORTA method, and conducted numerical experiments in which the failures of the NORTA method were related to increase in dimensionality of the generated random vector (Ghosh and Henderson, 2003). They noted that the probability of failure of the NORTA method is over 80% for random vectors with dimensions that are above 10. Additionally, the recent use of SDP that they proposed has significantly slowed down the execution, reporting for example 10 min run times for simulating correlated random vectors of dimension 10 (Ghosh and Henderson, 2003).

In their work, Ghosh and Henderson (2003) repeatedly state that "for two-dimensional random vectors, the NORTA method can match any feasible correlation matrix. This follows immediately from the characterizations in Whitt (1975)." The idea in this paper is based on extending this concepts for vectors which are multi-dimensional, using a systematic approach of re-arranging the elements of vectors $X_k, k = 2, \ldots, n$ based on the use of multiple regression fit as a measure of statistical dependence. Hence, the focus of this paper is a method of re-arranging the elements of each generated random variable in order to induce a desired statistical dependence. In addition to execution speed and the ease of implementation, additional advantages of the proposed method are

(a) The method preserves the Pearson correlations instead of the rank correlations. This may sometimes be preferable to preserving the rank correlation.
(b) The method can be used to induce desired statistical dependence among random variables which are derived from empirical distributions. Recent advances in kernel distribution functions (Silverman, 1986; Scott, 1992) have gained momentum among researchers since they offer more flexibility for statis-

tical representation of the processes that are difficult to model using the existing theoretical distributions.

The rest of the paper is divided as follows: Section 2 provides some practical aspects related to problem formulation; Section 3 gives theoretical considerations for the basis of the algorithm, Section 4 explains the algorithm, Section 5 provides results of numerical experiments and Section 6 provides conclusions, followed by acknowledgement and references.

2. Problem formulation

The problem under consideration deals with random generation of vectors which have different probability distributions and which exhibit mutual statistical dependence. It is assumed that random vectors represent processes for which either observed data or underlying theoretical knowledge is available, such that probability distribution functions and parameters of statistical dependence can be estimated.

In general, the above problem can be approached in two ways. One way is to try to estimate the joint probability densities for two or more variables, and the other is to first independently generate the random variables as univariate processes, and then impose the desired correlation structure by re-ordering sequence of their elements. The latter approach ensures that the properties of the initial marginal distributions remain unaffected, and it is also easier to implement due to the multitude of available univariate distributions and fitting techniques which are well known and are therefore not discussed further in this paper. The former approach requires estimation of the parameters and functional form of the multivariate frequency distribution function based on the observed data – a much more difficult task given the uncertainties associated with the functional form, and with typically insufficient length of the available data for fitting multivariate distribution functions.

It is common to define dependence between a series of random vectors using a correlation matrix. When used to model real-world processes for which data are available, this matrix is constructed by calculating the correlation coefficient for each par of random variables under consideration. Another way to represent dependence between observed data is to estimate the matrix of regression coefficients and the regression error term. Usually, both the correlation coefficient matrix and regression coefficient matrix are estimated based on the observed data. This is emphasized since the approach presented in this paper uses the matrix of regression coefficients to represent statistical dependence. Determination of regression coefficients from observed data follows known methodology already covered in standard statistical textbooks (Devore, 1991). It is also possible to estimate a matrix of regression coefficients based on a given Pearson correlation matrix along with the means and standard deviations of correlated variables (Cooley and Lohnes, 1971), and a computer programs that can provide this transformation are available (Cooley and Lohnes, 1971; UNESCO, 2004). Therefore, in this paper we shall assume that the desired statistical dependence can be represented either as a target matrix of product moment correlations or as the lower triangular matrix of regression coefficients.

3. Theoretical basis

Correlation between two independent random variables is zero. Assuming that vectors x and y of dimension n are generated independently, it may be possible to induce a desired correlation by systematic re-ordering of the elements of random vector y. Inducing a desired correlation can be viewed as a combinatorial problem, for which suitable tools can range from mathematical programming to heuristics. In general, there is no unique solution for the sequence of elements of vector y provided that the target correlations are less than the maximum. The goal is to find a suitable solution with a minimum computational effort. It is important to note that for two independent random variables x and y the highest possible correlation that can be obtained by re-arranging their elements to correspond to the pairs of elements of x and y obtained from the sorted sequence of both vectors. Whitt (1976) provided a proof that the sorted sequence of vectors x and y provides the matching of their elements with maximum correlation. The correlation coefficient $\rho_{x,y}$ is a measure of statistical dependence between vectors x and y:

$$\rho_{xy} = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{n\sigma_x\sigma_y} \tag{1}$$

where $\bar{x}$ and $\bar{y}$ are mean values while σ_x and σ_y are standard deviations of sample vectors x and y. It is obvious from Eq. (1) that the value of $\rho_{x,y}$ would change if any two elements y_i and y_k were to swap their positions ($i, k = 1,\ldots, n$). One could therefore envisage an algorithm that consists of a set of systematic trials aimed to swap the positions of the elements of vector y_i, such that only trials resulting in a desired change in the value of $\rho_{x,y}$ are accepted, while the others are discarded. In this fashion the algorithm would be similar to a bubble sort procedure which is known as the simplest sorting algorithm. The maximum correlation that could be enforced in this way would correspond to the correlation obtained after both vectors x and y are sorted in the same sequence. However, the desired target correlation is typically less than maximum, hence the algorithm terminates when the correlation between vector x and y has reached its target. Note that such an algorithm would not require a complete recalculation of all terms on the right hand side of Eq. (1), but instead only the net change in the value of $\rho_{x,y}$ caused by swapping of its two elements.

Although the above suggestion would work well for two vectors, re-arranging an arbitrary vector such that its

correlation coefficients with n other vectors are induced is a more difficult proposition. Eq. (1) would have to be written n times and any permutation of elements that improves some of the correlations may worsen the others. To avoid this, the proposed algorithm exploits the known relationships between coefficient of determination associated with multiple regression and correlation.

Consider a simple linear regression model of the form:

$$y_i = a_0 x_i + a_1 + N[0, \varepsilon_y] \tag{2}$$

where x_i is an independent variable, a_0 and a_1 are parameters of estimation, ε_y is the standard error of estimate and $N[0, \varepsilon_y]$ is a normally distributed random term with zero mean and standard deviation of ε_y. Assume that a_0, a_1 and ε_y can be estimated based on observed data, much $\rho_{x,y}$. A principal difference between correlation and regression is that in regression one must distinguish between independent variable x_i and statistically dependent variable y_i, and that regression coefficients change if the independent–dependent formulation is reversed (i.e. y_i is considered independent and x_i is dependent). On the other hand, a correlation coefficient shows statistical dependence between the two variables without considering independent–dependent nature that may exist between them. Therefore, if we are to use regression as an aid to enforce a given correlation between vectors x and y by permutation of elements of one of the vectors, the choice of a statistical dependence for either vector x or y is arbitrary.

The standard error of estimate in Eq. (2) for a sample of n elements is defined as:

$$\varepsilon_y = \sqrt{\frac{\sum_{i=1}^{n}(y_i - y_i')^2}{n-2}} \tag{3}$$

where y_i' represents the target values which would fit the regression line $a_0 x_i + a_1$. It is known from elementary statistics (Devore, 1991) that the square of the correlation coefficient $\rho_{x,y}$ gives the value of the coefficient of determination r^2 of linear regression between variables x and y. The coefficient of determination is defined as:

$$r^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - y_i')^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

For a given vector y only the error sum of squares $(y_i - y_i')^2$ represented as the numerator portion of the right hand side term in Eq. (4) can be modified by manipulating the order of elements y_i. The value of the denominator remains unchanged regardless of the ordering sequence of elements y_i. Note that the error sum of squares is also featured in the sample standard error of estimate ε_y in Eq. (3) for vectors x and y. As the term ε_y approaches zero, the coefficient of determination r^2 approaches its maximum value of 1, and correlation coefficient $\rho_{x,y}$ approaches its maximum or minimum values of 1 or -1 (minimum value in case of negative statistical dependence between variables x and y). Hence, an important observation for the purpose of building algorithms is that changing ε_y by re-ordering of y_i also changes the correlation coefficient between vectors x and y. An increase of ε_y causes reduction of the correlation between x and y, and vice versa – a reduction of ε_y causes in an increase of the correlation between x and y. The above considerations are also valid for multiple correlation and regression, where multiple r^2 can serve as a composite measure of the fitness when one dependent variable is regressed against several independent variables.

With the above considerations, it is possible to formulate an efficient algorithm for inducing desired correlation between vectors x and y by permutation of the elements y_i. The simplicity of the proposed algorithm is that it is based in part on the above notions from elementary statistic. No additional theoretical considerations are required.

4. Algorithm

In this paper the use of terms such as vectors x, y, z is synonymous with random variables x, y, z. As previously mentioned, the details regarding the generation of random variables with chosen statistical distribution functions are not dealt with here. Rather, the focus is on the procedure for re-ordering of the elements of randomly generated vectors such that their product moment correlation matrix closely matches a desired set of target correlations. The proposed method requires the lower triangular regression matrix containing the target regression coefficients, the standard errors of estimate and coefficients of determination. These can be estimated in the same way a target correlation matrix is estimated, typically by using the observed data sample or by utilizing knowledge of the process which is being modeled. A target correlation matrix is also required for verification of the algorithm.

The algorithm proceeds through the steps described below on the simplest example of re-arranging of elements y_i with respect to vector x until a desired correlation is induced. Once this is accomplished, the algorithm retains the new arrangement of the elements of vector y and proceeds with re-arranging elements of vector z with respect to the unchanged initial vector x and re-arranged vector y. The procedure is thus repeated for other random variables for which a specified correlation structure is desired. Hence, the conceptual procedure explained for permutations of elements of vector y with respect to vector x is subsequently repeated for all other vectors without modifications.

The permutation procedure of elements y_i proceeds in a few distinct steps:

Step 1. Using the available regression parameters a_0, a_1 and ε_y generate a random set of n elements of y_i'' such that

$$y_i'' = a_0 x_i + a_1 + N[0, \varepsilon_y] \tag{5}$$

The idea is to use vector y'' as a temporary aid in re-arranging the elements of vector y, since vector y'' has a desired correlation structure to vector x.

The n realizations of the random term N[0, ε_y] in Eq. (5) are saved as elements of vector τ for future re-use, and the n realizations of the sum of target values $a_0\ x_i + a_1$ are also saved as elements y'_i. The purpose of this will become apparent in subsequent steps.

Step 2. Sort out both vector y'' and the original vector y which has the desired marginal distribution. The sorted order of elements of both vectors will result in the highest possible correlation between the two vectors.

Step 3. Replace the elements y''_i by their counterpart elements y_i obtained from the sorted order of both vectors. This has effectively created a copy of vector y_i with a changed order of its elements which brought vector y closer to matching the desired statistical dependence with vector x.

Step 4. Re-calculate the coefficient of determination r'^2 that corresponds to the new order of elements y_i using Eq. (4).

Step 5. Compare the coefficient of determination r'^2 obtained in step 4 with the target coefficient of determination r^2. If they are sufficiently close (e.g. if $|r^2 - r'^2| < 0.005$), stop. If not, go to step 6.

Step 6. Calculate the standard error of estimate ε'_y that corresponds to the new order of elements y_i obtained in step 3 using Eq. (3) and estimate δ such that:

$$\delta = \frac{\varepsilon'_y}{\varepsilon_y} \tag{6}$$

Step 7. Adjust the elements of vector τ such that $\tau' = \tau\delta$ and re-generate estimates of y''_i by using:

$$y''_i = a_0 x_i + a_1 + \tau' \tag{7}$$

Proceed to steps 2 through 7 until a convergence is achieved or until no further improvement is possible, measured by ε'_y approaching values close to zero. The process normally converges after one or two iterations. To ensure faster convergence on difficult problems, the algorithm can be improved by using an improved first guess and a sophisticated iteration algorithm explained below. Depending on the desired accuracy, the algorithm can be set to exit after a sufficient number of iterations which would lead to either the desired statistical dependence or to the maximum possible dependence measured by the elements of ε converging to zero. It should also be noted that the setting of the threshold $|r^2 - r'2|$ allows more or less accuracy of the final arrangement of all elements. With a smaller value of $|r^2 - r'^2|$ the match with the target correlation matrix is closer, but this may require a few extra iterations. The user can set the desired accuracy of the algorithm by selecting a value for $|r^2 - r'^2|$. This feature is not available in the algorithm of Iman and Conover (1982).

In the above formulation, vectors x and y represent random variables generated using their marginal distributions. Once the elements of vector y have been arranged to give the desired statistical dependence, the process can be repeated for additional variables in a sequential manner one variable at a time, e.g. for variable z correlated to x and y the corresponding multiple regression equation is:

$$z''_i = a_0 x_i + a_1 y_i + a_2 + N[0, \varepsilon_z] \tag{8}$$

In addition to the use of Eq. (8), the model also uses the multiple coefficient of determination, which ensures that re-ordering of elements z_i conforms simultaneously to correlations between vectors z and x as well as between z and y. All other steps of the above procedure remain intact.

One of the possible difficulties with the application of the proposed algorithm is associated with the dimension n of the random vectors. If the desired value for n is too small, this may provide insufficient pool of possible values to guarantee that the desired correlation structure can be induced. However, this is easy to resolve by generating a sufficiently large sample, re-ordering the elements of all vectors as explained above, and then adopting as a desired output only the first n elements of each vector.

It should also be noted that the proposed method is transparent to inducing linear or non-linear regressive relationship between random variables, since r'^2 is evaluated in the same way for either linear or non-linear regression. Although no attempt was made to test the algorithm with a target non-linear regression in this research, other researchers may wish to further explore this possibility.

Additional improvements related to the convergence of the algorithm can be achieved using the following two options:

(a) Improved first guess. It was found that for some problems the following formulation for δ is often closer to the converged value of δ thus reducing the number of required iterations:

$$\delta = 1 \pm \sqrt{\frac{|\varepsilon_y - \varepsilon'_y|}{\varepsilon_y}} \tag{9}$$

where the choice between + or − depends on the sign of $\varepsilon_y - \varepsilon'_y$.

(b) Faster Iterative Scheme. In addition to an improved first guess for δ, other guesses can also be improved by using a linear interpolation of the two previous guesses. Denote with an g_k a guessed value of δ at iteration k and with a c_k its calculated value obtained as:

$$c_k = g_k + \frac{r'^2 - r^2}{r^2} \tag{10}$$

Hence, when the relative error between the targeted and calculated coefficient of determination r^2 becomes negligibly small (i.e $|\ r^2–r'^2| \sim 0$, then $c_k–g_k$ and the convergence is achieved. The evaluation of guessed and calculated parameters proceeds as follows:

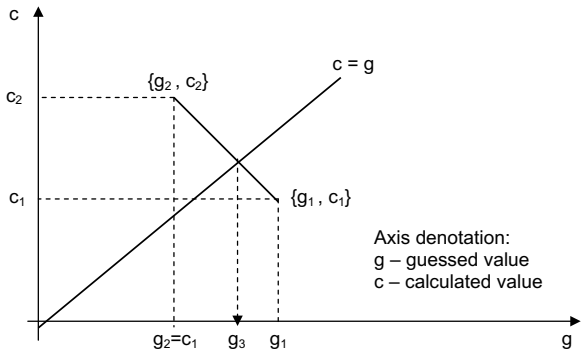

Fig. 1. Graphical interpretation of convergence mechanism.

Iteration 1: Set g_1 using Eq. (9);
Iteration 2: Set g_2 using Eq. (10) and set $c_1 = g_1$;
Iteration 3: Update the term r'^2, set c_2 using Eq. (10) and set the subsequent guess g_3 using the following equation:

$$g_3 = \frac{c_1 g_2 - c_2 g_1}{g_2 - g_1 - c_2 + c_1} \tag{11}$$

The above expression for g_3 represents the g_3 coordinate of an intersection between two lines in the $\{g, c\}$ coordinate system, where g is the x-axis and c is the y-axis. One of the two lines is defined by a pair of points $[g_1,c_1]$ and $[g_2,c_2]$ and the other by $c = g$, as shown in Fig. 1. All subsequent guesses are made using Eq. (11) with updated values of g_1, g_2 and c_1 as shown below:

$$g_1 = g_2 \tag{12}$$

$$g_2 = g_3 \tag{13}$$

$$c_1 = c_2 \tag{14}$$

The effectiveness of the above convergence scheme is demonstrated on the first numerical example in Section 5.

5. Numerical examples

Three test problems are presented. The first is aimed to demonstrate the proposed convergence scheme presented in Section 4. The second test compares the accuracy of the proposed algorithm with the algorithm of Iman and Conover that is currently used in commercial simulation packages, while the third is aimed to demonstrate conversion of correlation matrix into a matrix of regression coefficients.

Table 1
Definition of distribution functions

Variable	Distribution and parameters	Description of parameters
Var 1	Weibull (2.65, 10.33)	Shape, scale
Var 2	Extreme Value (7.65, 2.76)	Location, shape
Var 3	Log Normal (13.26, 4.53)	Mean, standard deviation
Var 4	Binomial (19, 0.46)	No. of draws, probability of success of each draw
Var 5	Gamma (4.48, 1.24)	Shape, scale
Var 6	Poisson (8.26)	Lambda for Poisson distribution
Var 7	Pearson V (7.45, 60.15)	Shape, scale
Var 8	Chi Square (10)	No. of degrees of freedom

Test Problem 1

To demonstrate the universal nature of the algorithm, a test problem with eight variables was selected with a mix of positive and negative correlations, as well as a mix of random vectors with various marginal distributions that included both floating point and integer variables. The following input data statistics were compiled to define the test problem:

(a) Table 1 contains the selected statistical distribution and its parameters for each variable;
(b) Table 2 contains the lower triangular matrix of regression coefficients and error terms; and,
(c) Table 3 contains the correlation matrix.

The objective of this test problem is to demonstrate the ability of the algorithm to generate eight-dimensional random vector with marginal distributions given in Table 1 and with the statistical dependence structure given in Table 3. Table 2 is required for the working of the algorithm, but as stated earlier, Table 2 could have been obtained either by using Table 3 as input or on the basis of solving the normal equations for the same observed empirical datasets from which the correlation in Table 3 was generated in case the simulation process is based on empirical observations.

The goal of this test run was to provide re-ordering of elements of randomly generated vectors. For each vector

Table 2
Regression coefficients and standard errors of estimate

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	ε_y
1.190249	0.876881	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.600622
3.767440	−0.464989	1.491430	0.000000	0.000000	0.000000	0.000000	0.000000	2.442897
3.229037	0.202747	−0.384836	0.549876	0.000000	0.000000	0.000000	0.000000	1.015482
15.911602	−0.008384	−0.085425	0.195606	−1.383125	0.000000	0.000000	0.000000	1.098559
4.632482	−0.078959	0.005707	0.065370	0.667744	−0.431165	0.000000	0.000000	0.865812
−8.486017	0.063402	−0.214972	0.232695	−0.156481	0.538247	1.752190	0.000000	1.402440
−5.560172	−0.031397	−0.016832	−0.120967	0.431866	0.205193	0.892046	0.563336	1.330376

Table 3
Target correlation matrix

	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8
Var 1	1.000	0.901	0.684	0.567	−0.521	0.487	0.418	0.393
Var 2		1.000	0.838	0.648	−0.570	0.577	0.519	0.483
Var 3			1.000	0.866	−0.738	0.800	0.770	0.734
Var 4				1.000	−0.910	0.938	0.857	0.877
Var 5					1.000	−0.919	−0.788	−0.822
Var 6						1.000	0.926	0.940
Var 7							1.000	0.942
Var 8								1.000

1000 elements were generated according to the marginal distributions in Table 1 using the @RISK package from Palisade Corporation. These were read as input into the model. Although the required dimension for the test problem was only 100, the model re-ordered 1000 elements of each vector, and this output was then split into 10 separate subsets. This provided 10 possible solutions sets, which could be compared and analyzed. A correlation matrix was constructed for each of the 10 outputs, and the average values of each correlation coefficient are given in Table 4, while Table 5 shows standard deviations of each correlation coefficient based on the 10 correlation matrices generated for each of the 10 outputs. It is obvious that the proposed algorithm can re-order the input vectors such that their resulting correlation matrix closely resembles the target. The threshold $|r^2–r'^2|$ was set to 0.005 for this problem.

Since the algorithm employs an iterative scheme, it is prudent to check its convergence. Table 6 provides a summary of all iteration steps, with the interim values of r^2, r'^2 and δ. It can be seen that the initial guess already provided convergence for variables 3, 7 and 8, while variable 5 required only one additional trial to converge. Successful re-ordering of variables 2, 4 and 6 required up to 5 additional iterations to converge. Convergence is achieved when r^2 and r'^2 become sufficiently close, i.e. when the absolute value of their difference becomes less than 0.005.

Test Problem 2

This problem involves re-arrangement of 52 random variables, each with 1000 elements. Each variable is generated using log normal distribution with the mean and standard deviation obtained from empirical data samples available for all 52 variables. Also obtained from the data sample are the Spearman rank correlation (to be used as input into @RISK model), Pearson product moment correlation matrix, and a lower triangular regression matrix obtained by solving the normal equations which were setup based on the available data sample (Devore, 1991). The target rank and Person correlation matrices, along with the historic data sample and the simulated random vectors

Table 4
Average resulting correlation matrix

	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8
Var 1	1.000	0.902	0.671	0.547	−0.477	0.488	0.460	0.418
Var 2		1.000	0.833	0.631	−0.525	0.586	0.588	0.526
Var 3			1.000	0.861	−0.725	0.829	0.834	0.790
Var 4				1.000	−0.916	0.958	0.860	0.910
Var 5					1.000	−0.903	−0.734	−0.823
Var 6						1.000	0.925	0.959
Var 7							1.000	0.938
Var 8								1.000

Table 5
Standard deviation of correlation coefficients

	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8
Var 1	0.000	0.017	0.056	0.061	0.074	0.078	0.086	0.090
Var 2		0.000	0.036	0.056	0.061	0.072	0.090	0.090
Var 3			0.000	0.026	0.033	0.032	0.044	0.049
Var 4				0.000	0.010	0.009	0.028	0.017
Var 5					0.000	0.012	0.045	0.027
Var 6						0.000	0.021	0.008
Var 7							0.000	0.010
Var 8								0.000

Table 6
Values of r^2 r'^2 and δ in each iteration

Variable	Iteration	r^2	r'^2	Δ
2	1	0.9011	0.8701	1.1051
	2		0.8762	1.0775
	3		0.8811	1.0432
	4		0.8871	0.9631
	5		0.9000	0.9568
3	1	0.8541	0.8565	0.9190
4	1	0.8895	0.9135	0.6421
	2		0.9075	0.6623
	3		0.9044	0.6881
	4		0.9026	0.8803
	5		0.8759	0.7823
	6		0.8876	0.7663
5	1	0.9175	0.9229	0.7702
	2		0.9223	0.7753
6	1	0.9538	0.9643	0.4925
	2		0.9639	0.5031
	3		0.9637	0.5231
	4		0.9630	0.7719
	5		0.9470	0.6667
	6		0.9541	0.6719
7	1	0.9431	0.9438	0.5108
8	1	0.9630	0.9617	0.7853

obtained from @RISK and from the proposed algorithm can be obtained upon request.

The first step is to generate 1000 elements of each random variable according to the given parameters of lognormal distribution. This was done using one of the standard methods available in the literature (Law and Kelton, 2000), but it could have also been done as an independent step by running @RISK on some other input generation model without including any statistical dependence among the variables. Hence, the results of this step are not in question, as the author merely relied on the proven methods developed by other researchers. Also, the available simulation outputs posted on the web allow for easy verification of the generated marginal distributions.

The same lognormal parameters were used for generation of all 52 variables using the @RISK spreadsheet add-on, along with a Spearman rank correlation matrix as input obtained from the raw data. A comparison with the @RISK package regarding the execution speed was not carried out, since @RISK needs to use the rank-order correlation matrix that first needs to be generated. It then runs within a spreadsheet, which adds additional disadvantage to its execution speed. To make the comparison between the two algorithms more transparent, the convergence criteria for each variable were loosely set to $|r^2 - r'^2| < 0.1$ since @RISK has no similar mechanism for fine tuning the re-ordering process. With the exception of only one variable that needed a single iteration to converge, the first guess of δ for all other variables provided a solution that already satisfies this condition. The proposed algorithm takes less than 7 seconds of CPU time on a 3.2 GHz PC to re-arrange all 52 random variables such that the desired correlation matrix is induced.

Table 7
Comparison of absolute errors

Probability	Proposed algorithm	Algorithm of Iman & Conover
0.001	−0.152	−0.350
0.005	−0.133	−0.312
0.010	−0.123	−0.278
0.020	−0.110	−0.227
0.025	−0.107	−0.210
0.030	−0.103	−0.193
0.050	−0.096	−0.170
0.100	−0.077	−0.136
0.150	−0.064	−0.107
0.200	−0.054	−0.089
0.250	−0.046	−0.074
0.300	−0.039	−0.059
0.350	−0.032	−0.043
0.400	−0.025	−0.031
0.450	−0.018	−0.017
0.500	−0.010	−0.006
0.550	−0.002	0.009
0.600	0.008	0.022
0.650	0.018	0.038
0.700	0.030	0.057
0.750	0.041	0.080
0.800	0.053	0.109
0.850	0.069	0.141
0.900	0.091	0.176
0.950	0.133	0.218
0.960	0.146	0.230
0.970	0.158	0.250
0.980	0.170	0.276
0.990	0.190	0.317
0.999	0.238	0.457
Mean	0.001	0.008
Standard Deviation	0.069	0.122

Once the output was generated by both algorithms, it was possible to compare the Pearson product moment correlation matrix created on the basis of both algorithms with the target Pearson correlation matrix obtained from the raw data. A deviation from the target values constitutes an absolute error. Table 7 provides statistical summary of the absolute errors obtained from both algorithms, including the resulting cumulative probability, mean and standard deviation. The mean absolute errors of both algorithms are close to zero, which was to be expected. However, the standard deviation of absolute error obtained from the algorithm of Iman and Conover is roughly two times higher than that obtained by the proposed algorithm. About 20% of all absolute errors in the algorithm of Iman and Conover are greater than 0.15 – which means, for example, that if the target Pearson correlation is 0.7 the algorithm of Iman and Conover will achieve either below 0.55 or above 0.85 for one in five correlations on average. This level of accuracy may not be acceptable in all applications, especially where generation of random variables with desired statistical dependence is only one step in a larger simulation process with other random components.

The impact of significant increase of dimensions of the generated random vector with a desired correlation structure is one area of study that will have to be addressed in the future. At this point, it can be said that Test Problem 2 performs well with 52 dimensions, but this performance is also satisfactory for @RISK model as far as matching the desired rank correlation is concerned. The author has successfully tested both the proposed algorithm and @RISK on generation of as many as 208 correlated variables with skewed marginal distributions. This larger test has been published (Ilich and Despotovic, 2007) and it is therefore not dealt with in this paper. It would appear that Test Problem 2 already performs well with vectors of 52 dimensions, which seems much better than the performance of the NORTA method reported by Ghosh and Henderson (2003).

Test Problem 3

This test problem demonstrates how to find the regression coefficients for a given correlation matrix in Table 8 and for the assumed probability distribution parameters, which are: Poisson with mean equal to 7 and standard deviation of 2.6587 for variable 1; Normal distribution with a mean of 5 and standard deviation of 3 for variable 2; and Exponential distribution with a mean of 0.1 and standard deviation of 0.1 for variable 3. Both the means and standard deviations are required for calculation of regression coefficients. The following sections follows closely the established procedures found in a range of statistical textbooks. It begins by partitioning the given correlation matrix $\mathbf{R}$ into $\mathbf{R}_{11}$, $\mathbf{R}_{12}$, $\mathbf{R}_{21}$ and a single last diagonal coefficient 1.0 in the final row, i.e.

$$\mathbf{R} = \left[\frac{\mathbf{R}_{11}|\mathbf{R}_{12}}{\mathbf{R}_{21}|1.0}\right] \tag{15}$$

implying that the last random variable is dependent and regressed to the remaining variables. With this assumption, a set of standardized regression weights $\mathbf{b}$ can be found by solving a system of linear equations defined in (16):

$$\mathbf{R}_{11}\mathbf{b} = \mathbf{R}_{12} \tag{16}$$

where $\mathbf{b}$ is a column matrix of standardized regression weights. This process is repeated $p-1$ times, where p is the assumed rank of the given correlation matrix. Each time the process defines one row of coefficients in the regression matrix. To convert the standardized regression weights into the desired raw weights it is necessary to use the means m_p and standard deviations s_p of the random variables and apply the following formulas:

Table 8
Correlation matrix for test Problem 3

	Var 1	Var 2	Var 3
Var 1	1.000	−0.80	0.50
Var 2		1.000	−0.40
Var 3			1.000

$$\widehat{X}_{pi} = \frac{s_p}{s_1}b_1X_{1i} + \frac{s_p}{s_2}b_2X_{2i} + \ldots + \frac{s_p}{s_{p-1}}b_{p-1}X_{p-1,i} + a_i \tag{17}$$

where intercept a_i is found using

$$a_i = m_p - \frac{s_p}{s_1}b_1m_1 - \frac{s_p}{s_2}b_2m_2 - \ldots - \frac{s_p}{s_{p-1}}b_{p-1}m_{p-1} \tag{18}$$

while the regression error of estimate e_i is related to the standard deviation s_i of each variable through the multiple correlation coefficient R as defined in Eq. (19):

$$e_i = s_i\sqrt{1-R^2} \tag{19}$$

Label the variables 1, 2 and 3 in this problem as X, Y and Z. The calculation then proceeds as follows:

Using stepwise approach, solve for standardized regression coefficients b_i using the matrix Eq. (16). For regression variable X to Y in the given sample problem, the single equation is

$$1.0 \cdot b_1 = -0.8 \tag{20}$$

This is a regression coefficient of standardized variables x and y, which can be converted to the regression coefficients of the raw variables X and Y according to Eq. (17):

$$a_1 = (\text{St.Dev.}Y)/(\text{St.Dev.}X) \cdot b_1 = 3/2.6587 \cdot (-0.8) = -0.9027 \tag{21}$$

Hence, regression coefficient $a_1 = -0.9027$

We can now calculate the intercept a_0 using Eq. (18):

$$a_0 = m_2 - a_1m_1 \tag{22}$$

where m_2 and m_1 are the means of Y and X, respectively. Hence,

$$a_0 = 5 - (-0.9027) \cdot 7 = 11.3189 \tag{23}$$

Therefore, the first row of the regression matrix coefficients contains 11.3189 and −0.9027 for a_0 and a_1, respectively (a_2 is zero by default since we are only regressing Y to X in the first equation). Calculation of the second row coefficients proceeds below. Finding the second set of regression coefficients requires finding a solution of a system of two equations:

$$1.0b_1 - 0.8b_2 = 0.5 \tag{24}$$

$$-0.8b_1 + 1.0b_2 = -0.4 \tag{25}$$

The solution for the above system is $b_1 = 0.5$ and $b_2 = 0$. Hence,

$$a_1 = 0.1/2.6587 \cdot 0.5 = 0.0189 \tag{26}$$

$$a_2 = 0.1/3 \cdot 0 = 0.0 \tag{27}$$

while,

$$a_0 = m_3 - a_2m_2 - a_1m_1 \tag{28}$$

$$a_0 = 0.1 - 0.0 \cdot 5 - 0.0189 \cdot 7 = -0.0323 \tag{29}$$

Hence, the second row of the regression matrix coefficients contains −0.0323, 0.0189 and 0.0 for coefficients a_0, a_1 and a_2, respectively. The term $1-R^2$ is found on the diagonal elements of matrix $\mathbf{C} = \left(\mathbf{R}_{11}^T\mathbf{R}_{11}\right)^{-1}$ which also has to be

calculated $p-1$ times due to the need to redefine matrix $\mathbf{R}_{11}$ for each dependent regression variable, which can be done using standard matrix calculation techniques. The results for the term $1-R^2$ for the first and the second row are 0.36 and 0.75, respectively. The regression standard errors of estimate are then:

$$e_y = 3.0\sqrt{0.36} = 1.8 \quad \text{and} \tag{30}$$

$$e_z = 0.1\sqrt{0.75} = 0.0866 \tag{31}$$

where values 3.0 and 0.1 are the standard deviations of variables Y and Z given in this problem. It should be noted that the term $1-R^2$ inside the square root allow us to define multiple R as 0.8 and 0.5 for variables Y and Z, respectively. Also, the important feature of the algorithm is that multiple R values are matched with the R values calculated at the end of each iterative step during the search process, as explained below.

In this example, we start by generating randomly all three variables. The first variable is then taken as input X into the above regression equations to generate re-ordering keys for both Y and subsequently Z. At the end of this step, the resulting sequence of 25 elements of all three vectors and their correlation matrix is given in Table 9. Note that the proposed algorithm does not calculate the correlation matrix, which is given here for discussion purposes. It can be seen that the values of correlation coefficients are far from the desired targets. Rather than calculate the correlation matrix in every step, the proposed algorithm recognizes this discrepancy by calculating multiple Ry and Rz which are in this case equal to 0.945 and 0.842, respectively, while their target values are 0.8 and 0.5. In other words, the first attempt resulted in a fit that is above target. The algorithm then proceeds according to the steps 4 through 7 in Section 4 to iteratively generate new regressed targets by enlarging the initial values of e_y and e_z until an order of the elements of Y and Z vectors has been found such that their respective multiple R values are sufficiently close to the desired targets. For assumed respective values of e_y and e_z of 6.2 and 0.29, the order of the elements of vectors Y and Z is shown in Table 10. The calculated values of Ry and Rz statistics for this configuration of elements is 0.857 and 0.521, fairly close to their respective targets of 0.8 and 0.5. Also, the correlation matrix in Table 10 shows a reasonably good fit with the target correlation matrix, especially given the small size of the sample vectors of only 25 elements. A better fit is possible by increasing the number of iterations by using a tighter convergence criterion, as well as by increasing the number of elements of random vectors.

It is interesting to note that an attempt was made to find an approximate solution for this test problem with @RISK

Table 9
Solution of the first iteration for test Problem 3

X	Y	Z
4	8.2168	0.0258
8	3.7564	0.1383
4	6.7770	0.0032
15	−2.0836	0.1752
8	3.6570	0.0312
9	2.3638	0.0765
12	−1.2004	0.1649
4	9.8237	0.0018
6	4.8026	0.0508
2	10.0136	0.0149
6	4.9422	0.0672
6	5.2477	0.1174
5	9.5011	0.0079
6	6.2770	0.0529
1	14.9360	0.0004
5	8.4465	0.0245
11	−0.3986	0.1702
11	2.2214	0.1704
8	2.6708	0.1372
7	4.5181	0.1021
10	1.2746	0.0917
8	4.5299	0.1466
8	3.3795	0.0614
5	6.4591	0.0583
6	4.9514	0.0321

Pearson correlations:

Var 1	Var 2	Var 3
1.000	−0.945	0.840
	1.000	−0.811
		1.000

Table 10
Solution of the final iteration for test Problem 3

X	Y	Z
4	4.9422	0.0018
8	2.2214	0.0508
4	10.0136	0.0032
15	−2.0836	0.1174
8	6.4591	0.1021
9	1.2746	0.1704
12	−1.2004	0.1372
4	9.8237	0.0583
6	3.7564	0.0765
2	9.5011	0.0004
6	4.8026	0.0917
6	4.5181	0.0079
5	8.2168	0.1752
6	5.2477	0.0312
1	14.9360	0.0672
5	4.5299	0.0258
11	−0.3986	0.1702
11	3.6570	0.1649
8	4.9514	0.0245
7	2.3638	0.0149
10	3.3795	0.0529
8	6.7770	0.0321
8	2.6708	0.1466
5	8.4465	0.1383
6	6.2770	0.0614

Pearson correlations:

Var 1	Var 2	Var 3
1.000	−0.858	0.496
	1.000	−0.343
		1.000

package, an approximation arising from the fact that the input correlation matrix for @RISK contains rank correlations instead of the product moment correlations. The @RISK program reported that the given correlation matrix was infeasible and failed to provide a reasonable solution.

6. Conclusions

This paper presents an algorithm for inducing Pearson correlations among random vectors with any statistical distribution functions. The method is easy to implement. It can be used to induce statistical dependence among variables with arbitrary marginal distributions, including empirical distributions with no explicit formulations. Future research may involve attempts to induce non-linear statistical dependence among random variables with arbitrary distribution functions, examine the impact of high dimensions on the accuracy of the algorithm, or focus on additional algorithmic refinements.

Acknowledgement

The author wishes to express appreciation to the National Scientific and Engineering Research Council (NSERC) of Canada and to Golder Associated Ltd. of Calgary, Alberta, for jointly sponsoring this research project.

References

Biller, B., Nelson, B.L., 2003. Modeling and generation multivariate time series input processing using a vector autoregressive technique. ACM Transactions on Modeling and Computer Simulation 18 (3), 211–237.

Cario, M.C., Nelson, B.L., 1996. Autoregressive to anything: Time series input processes for simulation. Operations Research Letters 9, 51–58.

Cario, M.C., Nelson, B.L., 1997. Numerical methods for fitting and simulation autoregressive-to-anything processes. INFORMS Journal of Computing 10, 72–81.

Clemen, R.T., Reilly, T., 1999. Correlations and copulas for decision and risk analysis. Management Science 45, 208–224.

Cooley, W.W., Lohnes, P.R., 1971. Multivariate Data Analysis. Robert E. Krieger Publishing Company, Malabar, FL, Chapter 3.

Devore, J., 1991. Probability and Statistics for Engineering and Sciences. Brooks Cole.

Devorye, L., 1986. Non-Uniform Random Variate Generation. Springer-Verlag, New York.

Ghosh, S., Henderson, S.G., 2002. Chessboard distributions and random vectors with specified marginals and covariance matrix. Operations Research 50 (5), 820–834.

Ghosh, S., Henderson, S.G., 2003. Behavior of the NORTA method for correlated random vector generation as the dimension increases. ACM Transactions on Modeling and Computer Simulation 13 (3), 276–294.

Henderson, S.G., Chieara, B.A., Cooke, R.M., 2000. Generating "Dependent" Quasi-Random Numbers. In: Jones, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), Proceedings of the 2000 Winter Simulation Conference. Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 527–536.

Ilich, N., Despotovic, J., 2007. A Simple Method for Effective Multi-Site Generation of Stochastic Hydrologic Time Series. Journal or Stochastic Environmental Research and Risk Assessment, doi:10.1007/s00477-007-0113-6.

Iman, R., Conover, W., 1982. A distribution free approach to inducing a rank correlation among input variables. Communications in Statistics – Simulation and Computation 11 (3), 311–334.

Johnson, M.E., 1987. Multivariate Statistical Simulation. John-Wiley, New York.

Johnson, M.E., Ramberg, J.S., 1977. Transformations of the Multivariate Normal Distribution with Applications to Simulation. Technical Report LA-UR-77-2595, Los Alamos Scientific Laboratory, New Mexico.

Law, A.M., Kelton, W.D., 2000. Simulation Modeling and Analysis, 3rd ed. McGraw-Hill, New York.

Li, S.T., Hammond, J.L., 1975. Generation of pseudo random numbers with specified univariate distributions and correlation coefficients. IEEE Transactions on Systems, Man and Cybernetics 5, 557–561.

Lurie, P.M., Goldberg, M.S., 1998. An approximate method for sampling correlated random variables from partially specified distributions. Management Science 44, 203–218.

Mardia, K.V., 1970. A translation family of Bivariate distributions and Fréchet's Bounds. Sankhya. The Indian Journal of Statistics 32, 119–122, Series A.

@RISK Software for Decision and Risk Analysis, Palisade Corporation, www.palisade.com.

Scott, D.W., 1992. Multivariate Density Estimation, Theory Practice and Visualisation. John Wiley, New York.

Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York.

UNESCO. 2004. IDAMS: Internationally Developed Data Analysis and Management Software Package – WinIDAMS Reference Manual, Release 1.2, www.unesco.org/idams.

Whitt, W., 1976. Bivariate distributions with given marginals. The Annals of Statistics 4, 1280–1289.