Efficient Initialization of the Correlation Matrix in NORTA Using Quasi-Monte Carlo and Updating Techniques

Benjamas Tulyanitikul, Praphan Klompon, and Patchanok Srisuradetchai

Abstract— **Simulating multivariate random variables is essential in data analytics as it allows for more accurate modeling, improved decision-making, and a better understanding of complex systems and processes. The NORTA algorithm, a widely used method, can accomplish this task. However, it requires an initial correlation matrix to produce multivariate random variables with the desired correlation matrix, and both matrices usually differ. This paper presents an efficient simulation-based algorithm for determining the initial correlation matrix, leveraging quasi-Monte Carlo integration with the Halton sequence, an adaptive sum of squared errors, and some probability distributions. The proposed algorithm is tested to generate many cases of multivariate random variables with different distributions and different correlation matrices. The results show that this algorithm can substantially reduce time compared to the traditional simulation-based method.**

I. Introduction

Understanding and simulating correlations among random variables plays a pivotal role in big data analytics and practices across various fields, such as finance, engineering, and machine learning. The interconnected nature of complex systems necessitates the accurate simulation of correlated variables to enhance prediction quality, inform decision-making, and optimize risk assessment. In the context of big data analytics in finance, Leduc et al. [1] underscored the significance of incorporating correlations in systemic risk assessment. Within engineering, Qiu et al. [2] employed copula-based modeling to design reliable water distribution systems, emphasizing the importance of accounting for correlations in accurate estimation and infrastructure planning. Goodfellow et al. [3] investigated how generative adversarial networks (GANs) can be used to simulate correlated random variables in image synthesis tasks. This shows how important it is to understand correlations to improve the quality and realism of images.

There are various techniques for generating multivariate random variables. According to Niaki and Abbasi [4], generating methods can be classified into three categories: 1) the analytical approach, which utilizes marginal distribution functions and conditional distribution functions to generate multivariate random variables. This method has its limitations when it is not feasible to find conditional distribution functions; 2) the numerical approach, which involves the acceptance/rejection method and requires the selection of a joint probability density function that covers the joint probability density function of the desired multivariate random variables; and 3) the simulation approach, which employs specific properties in the transformation process, such as the NORTA (NORmal-To-Anything) transformation, also known as the Nataf transformation. This method utilizes marginal probability density functions and a correlation matrix, which have specific properties for generating multivariate random variables. The NORTA transformation was proposed by Cario and Nelson [5], and its conceptual origins come from the works of Mardia [6] and Li and Hammond [7].

The NORTA transformation aims to generate a vector of multivariate random variables with desired distributions and a correlation matrix by transforming a vector of multivariate normal random variables. This involves generating a vector of standard normal random variables, $\mathbf{Y} = (Y_1, \ldots, Y_m)^T$. Then, perform a Cholesky decomposition of the initial correlation matrix $\mathbf{\Sigma}_Z$ to obtain a lower triangular matrix $\mathbf{L}$ such that $\mathbf{\Sigma}_Z = \mathbf{L}\mathbf{L}^T$. Note that $\mathbf{\Sigma}_Z$ is a symmetric matrix:

$$\mathbf{\Sigma}_Z = \begin{bmatrix} \rho_Z(1,1) & \rho_Z(1,2) & \cdots & \rho_Z(1,m) \\ & \rho_Z(2,2) & \cdots & \rho_Z(2,m) \\ & & \ddots & \vdots \\ & & & \rho_Z(m,m) \end{bmatrix}.$$

Next, create a transformed vector by applying the Cholesky decomposition to the standard normal random variables: $\mathbf{Z} = \mathbf{L}\mathbf{Y}$. Finally, generate a vector of the multivariate random variable $\mathbf{X} = (X_1, \ldots, X_m)^T$, where $X_i = F_i^{-1}(z_i)$ and $F_i^{-1}(\cdot)$ is the inverse of the cumulative distribution of X_i. Finally, $\mathbf{X} = (X_1, \ldots, X_m)^T$ is claimed to be a vector of multivariate random variables where each $X_i, i = 1, 2, ..., m$ has a given marginal distribution and the correlation matrix is $\mathbf{\Sigma}_X$ [5].

Implementing the NORTA transformation requires establishing an initial correlation matrix $\mathbf{\Sigma}_Z$ for a multivariate random variable vector with various distributions, aiming to achieve the targeted correlation matrix $\mathbf{\Sigma}_X$. The initial correlation matrix typically differs from the target, $\mathbf{\Sigma}_X$, necessitating the determination of an initial correlation matrix. Cario and Nelson [5] suggested using the bisection method to determine $\mathbf{\Sigma}_Z$, and later studies focused on finding the relationship between the initial and desired matrices.

The correlation between two variables is measured by:

$$\rho_X(i,j) = Corr[X_i, X_j] = \frac{E[X_i X_j] - E[X_i]E[X_j]}{\sqrt{Var(X_i)Var(X_j)}}$$

B. Tulyanitikul is with Thammasat University, Pathum Thani, 12120 Thailand (e-mail: benjamas@mathstat.sci.tu.ac.th).

P. Klompon was with Thammasat University, Pathum Thani, 12120 Thailand (e-mail: praphan.klo@dome.tu.ac.th).

P. Srisuradetchai is with Thammasat University, Pathum Thani, 12120 Thailand (corresponding author to provide phone: +662-564-4440 ext. 106; e-mail: patchanok@mathstat.sci.tu.ac.th).

979-8-3503-0219-6/23/$31.00 ©2023 IEEE

$$= \frac{E\left[X_i X_j\right]}{\sigma_i \sigma_j} - \frac{\mu_i \mu_j}{\sigma_i \sigma_j},\ i \neq j,\ i, j = 1, 2, ..., m. \quad (1)$$

Term $E\left[X_i X_j\right]$ in (1) can be written as

$$E\left[X_i X_j\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \begin{pmatrix} F_i^{-1}\left[\Phi\left(z_i\right)\right] F_j^{-1}\left[\Phi\left(z_j\right)\right] \\ \phi\left[z_i, z_j, \rho_Z\left(i,j\right)\right] \end{pmatrix} dz_i dz_j, \quad (2)$$

where $\phi\left[z_i, z_j, \rho_Z\left(i,j\right)\right] = \dfrac{\exp\left\{-\dfrac{z_i^2 - 2\rho_Z\left(i,j\right) z_i z_j + z_j^2}{2\left[1-\rho_Z^2\left(i,j\right)\right]}\right\}}{2\pi\sqrt{1-\rho_Z^2\left(i,j\right)}}$

and $\rho_Z\left(i,j\right)$, which is the element in $\mathbf{\Sigma}_Z$, is the correlation between Z_i and Z_j. From (1) and (2), the relationship between $\rho_X\left(i,j\right)$ and $\rho_Z\left(i,j\right)$ can be presented in (3) as follows: $\rho_X\left(i,j\right) =$

$$-\frac{\mu_i \mu_j}{\sigma_i \sigma_j} + \frac{1}{\sigma_i \sigma_j}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \begin{pmatrix} F_i^{-1}\left[\Phi\left(z_i\right)\right] F_j^{-1}\left[\Phi\left(z_j\right)\right] \\ \phi\left[z_i, z_j, \rho_Z\left(i,j\right)\right] \end{pmatrix} dz_i dz_j. \quad (3)$$

Xiao [8] wrote (3), in the form of

$$\rho_X\left(i,j\right) = -\frac{\mu_i \mu_j}{\sigma_i \sigma_j} + \frac{1}{\sigma_i \sigma_j}\int_0^1\int_0^1 H\left[u_i, u_j; \rho_Z\left(i,j\right)\right] du_i du_j, \quad (4)$$

where $H\left[u_i, u_j; \rho_z\left(i,j\right)\right] =$

$F_i^{-1}\left(u_i\right) \cdot F_i^{-1}\left\{\Phi\left[\rho_z\left(i,j\right)\Phi^{-1}\left(u_i\right) + \sqrt{1-\rho_z^2\left(i,j\right)}\Phi^{-1}\left(u_j\right)\right]\right\}$, $Z_i = Y_i$, $Z_j = \rho_Z Y_i + \sqrt{1-\rho_Z^2} Y_i$, $u_i = \Phi\left(Y_i\right)$, and $u_j = \Phi\left(Y_j\right)$. Generally, solving (3) for $\rho_Z\left(i,j\right)$ given the desired correlation matrix $\rho_X\left(i,j\right)$ is not easy because there are $m(m-1)$ equations.

II. Literature Review

Several studies have proposed methods for determining the initial correlation matrix when X_i and X_j are continuous random variables, such as the empirical formulae proposed by Kiureghian and Liu [9], the root finding method proposed by Liu and Kiureghian [10], and the linear search method proposed by Li et al. [11]. However, when X_i or X_j is a discrete random variable, these methods cannot be used to solve the equations. Avramidis et al. [12] studied the correlation function between $\rho_Z\left(i,j\right)$ and $\rho_X\left(i,j\right)$, and they also developed a method to determine the value of $\rho_Z\left(i,j\right)$ for discrete random variables. Avramidis [13] expanded this study to the case of mixed continuous and discrete random variables. Niaki and Abbasi [4] proposed an artificial neural network method for generating correlated multivariate random variables.

Niavarani and Smith [14] proposed a method for generating multivariate random variables using NORTA, which avoids solving a system of equations when n is large. The steps are described from (i) to (vii) as follows:

(i) Create a random symmetric matrix $\mathbf{D}$ with principal diagonal elements equal to 0 and other elements as random numbers in $\left(-1-\rho_X\left(i,j\right),\ 1-\rho_X\left(i,j\right)\right)$;

(ii) Create the initial correlation matrix $\mathbf{\Sigma}_Z$ by adding the matrix $\mathbf{D}$ to $\mathbf{\Sigma}_X$, i.e, $\mathbf{\Sigma}_Z = \mathbf{\Sigma}_X + \mathbf{D}$;

(iii) Check if $\mathbf{\Sigma}_Z$ is a positive definite matrix. If not, create $\mathbf{D}$ again until $\mathbf{\Sigma}_Z$ is a positive definite matrix;

(iv) Use $\mathbf{\Sigma}_Z$ to create a multivariate random variable vector with the desired distribution of 5,000 vectors;

(v) Estimate $\rho_X\left(i,j\right)$ by using the method of moments in the following:

$$\hat{\rho}_X\left(i,j\right) = \frac{\Sigma\left(X_i - \bar{X}_i\right)\left(X_j - \bar{X}_j\right)}{\sqrt{\Sigma\left(X_i - \bar{X}_i\right)^2 \Sigma\left(X_j - \bar{X}_j\right)^2}}; \quad (5)$$

(vi) Calculate the sum of squared errors:

$$SSE = \sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\left[\rho_X\left(i,j\right) - \hat{\rho}_X\left(i,j\right)\right]^2;$$

(vii) Check if SSE is less than the acceptable sum of squared errors, SSE_{target}. If SSE is greater or equal to SSE_{target}, go back to the step of creating a new matrix $\mathbf{D}$ until a multivariate random variable vector can be created with the desired distribution and SSE *is* less than SSE_{target}. This will result in a multivariate random variable vector of 5,000 vectors. This method, developed by Niavarani and Smith [14], will be compared to our method.

The method introduced by Niavarani and Smith [14] is easy to comprehend but may require extensive processing time due to matrix creation with element d_{ij} that must be within a given range $\left(-1-\rho_x\left(i,j\right), 1-\rho_x\left(i,j\right)\right)$. For example, creating multivariate random variables whose marginal distributions are Binomial(n = 3, p = 0.5) and Gamma(shape = 14.4, scale = 0.03424) will produce $\rho_Z\left(i,j\right)$ of approximately 0.5181 for $\rho_X\left(i,j\right)$ of 0.5. Many iterations of different correlations are tried before reaching the desired answer, making the processing time lengthy. Xiao [8] also proposed a method to tackle this issue. The process involves transforming the double integral evaluated for $\rho_X\left(i,j\right)$ into an independent standard uniform space and introducing a quasi-Monte Carlo method to calculate the double integral. For a given $\rho_X\left(i,j\right)$, an appropriate $\rho_Z\left(i,j\right)$ is determined using the false position method. Although this method may be less efficient than existing approaches, it is comparatively easier to implement.

For this reason, this research presents an engaging method that tackles the challenges associated with generating elements in matrix $\mathbf{D}$ that appear in the method proposed by Niavarani and Smith [14]. By utilizing random variables with exponential or half-normal distributions, the method ensures that values in matrix $\mathbf{D}$ are always greater than zero.

Furthermore, we also modify the algorithm by introducing the adaptive sum of squared errors. This approach is employed iteratively to adjust the matrix, aiming to achieve a lower value of *SSE* with each iteration compared to its predecessor. Lastly, for a more accurate estimation of the double integral in (4), a simulation-based Monte Carlo method using Halton sequences is utilized.

III. Quasi-Monte Carlo Method

The conventional Monte Carlo method for assessing multidimensional integrals relies on pseudo-random integration nodes and is commonly employed when quadrature techniques prove too complex or costly to execute. As a more effective alternative, it has been proposed that utilizing quasi-random sequences, which are more uniformly distributed than pseudo-random ones, can lead to reduced error and enhanced convergence rates [15]. One of the most popular quasi-Monte Carlo (QMC) methods employs the Halton sequence, which is known for its low-discrepancy properties [16]. The Halton sequence is a quasi-random number sequence that exhibits a fairly uniform distribution. However, as the dimensionality of the Halton sequence increases, the uniformity of the distribution decreases. The Halton sequence is generated using the base representation of counting numbers with prime bases.

Let $m, m \geq 2$ be a counting number. Any counting number k can be uniquely represented in base m as:

$$k = b_0 + b_1 m + b_2 m^2 + \cdots + b_r m^r,$$

$$i = 0,1,\ldots,r, \;\; 0 \leq b_i \leq m-1,$$

where $m^r \leq k < m^{r+1}$. Moreover, any $c \in (0,1)$ can be uniquely represented in base m as:

$$c = c_0 m^{-1} + c_1 m^{-2} + \cdots, \;\; 0 \leq c_i \leq m-1, \;\; i = 0,1,2,\ldots.$$

Conventionally, this is written as $k = b_r b_{r-1} \cdots b_1 b_0$ and $c = c_0 c_1 \cdots$. Hence, a one-to-one correspondence between counting numbers and real numbers in the interval $(0,1)$ can be established as follows:

$$y_m(k) = b_0 m^{-1} + b_1 m^{-2} + \cdots + b_r m^{-r-1}.$$

The term $y_m(k) \in (0,1)$ is called the radical inverse of k with base m. Then, select prime numbers p_i, $i = 1,2,\ldots,s$, where s is the dimensionality of the Halton sequence. Thus, the sequence

$$\mathbf{x}_k = \left(y_{p_1}(k),\ldots,y_{p_s}(k)\right), \;\; k = 1,2,\ldots \qquad (6)$$

is called the Halton sequence. Fig. 1 depicts the Halton sequence, a type of quasi-random sequence that Xiao [8] used to estimate double integrations in correlation computations. This sequence appears to provide an almost uniform distribution of points when considering the distances between them. It is important to note that the use of the Halton sequence contributes to a reduction in these distances.

IV. Proposed Algorithm

The proposed algorithm utilized quasi-Monte Carlo integration with an update to the sum of squared errors. The steps are the following:

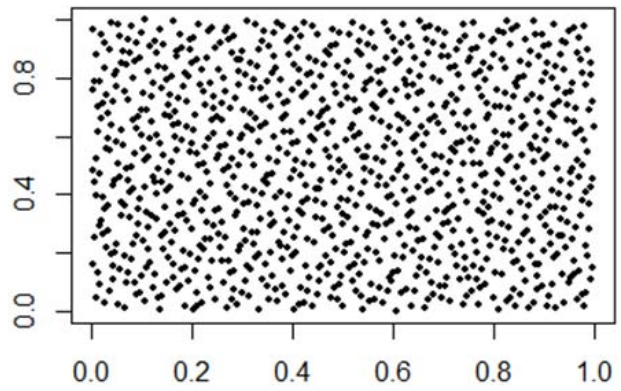

Figure 1. An example of 1,000 points of the Halton sequence.

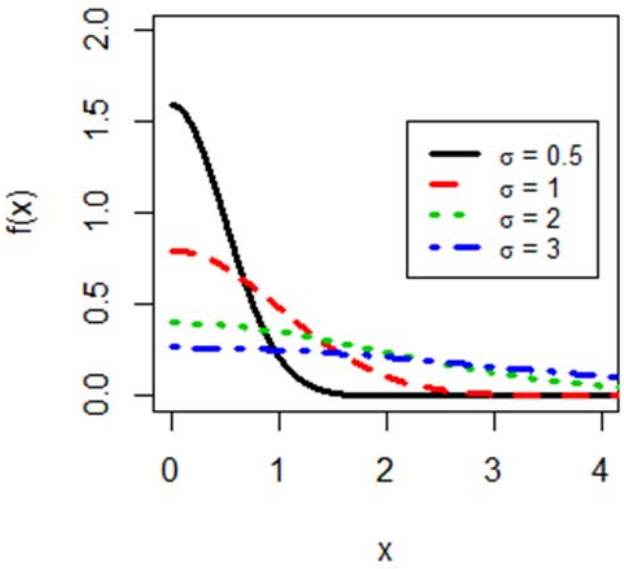

Figure 2. Possible shapes of half-normal distributions.

Step 1. Define the acceptable sum of squared errors (SSE_{target}) of the covariance matrix of the desired random variables and set $SSE_{\text{compared}} = m(m-1)$.

Step 2. Set $\boldsymbol{\Sigma}_{\text{initial}} = \boldsymbol{\Sigma}_X$ and construct matrix $\mathbf{S}_{m \times m}$ with elements s_{ij}, where

$$s_{ij} = \begin{cases} sign\left(\rho_X(i,j)\right), & i \neq j \\ 0, & i = j \end{cases},$$

and $\boldsymbol{\Sigma}_X$ is the desired correlation matrix.

Step 3. Create a random symmetric matrix $\mathbf{D}$ with dimensions equal to $\boldsymbol{\Sigma}_X$ with elements d_{ij} in matrix D. The d_{ij} elements are random variables with a half-normal or exponential distribution with specified parameters.

Step 4. Construct the initial covariance matrix $\boldsymbol{\Sigma}_Z = \boldsymbol{\Sigma}_{\text{initial}} + \mathbf{S} \circ \mathbf{D}$ and make $\boldsymbol{\Sigma}_Z$ symmetric.

Step 5. Check the condition $-1 < \rho_Z(i,j) < 1, \; i \neq j$ and whether the matrix $\boldsymbol{\Sigma}_Z$ is positive definite.

(5.1) If the condition is met, proceed to step 6.

(5.2) If the condition is not met, return to step 3.

Step 6. Create matrix $\mathbf{u}_{50000\times 2}$ using Halton sequences, with $u_{ij} \in (0,1)$ as the element of the i-th row and j-th column of $\mathbf{u}_{50000\times 2}$.

Step 7. Estimate $\rho_X(i,j)$ or (4) from

$$\hat{\rho}_x(i,j) = -\frac{\mu_i \mu_j}{\sigma_i \sigma_j} + \frac{1}{\sigma_i \sigma_j} \cdot \frac{1}{5\times 10^4} \sum_{k=1}^{5\times 10^4} F_i^{-1}(u_{k1}) \times$$

$$F_j^{-1}\left\{\Phi\left[\rho_z(i,j)\Phi^{-1}(u_{k1}) + \sqrt{1-\rho_z^2(i,j)}\Phi^{-1}(u_{k2})\right]\right\},$$

where $F_i^{-1}(\cdot)$ and $F_j^{-1}(\cdot)$ are the inverse cumulative distribution functions of X_i and X_j, respectively, and u_{ij} are obtained from step 6.

Step 8. Calculate $SSE = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \left[\rho_X(i,j) - \hat{\rho}_X(i,j)\right]^2$, where $\rho_X(i,j)$ and $\hat{\rho}_X(i,j)$ are the elements of the i-th row and j-th column of $\mathbf{\Sigma}_X$ and $\hat{\mathbf{\Sigma}}_X$, respectively.

Step 9. Check if $SSE < SSE_{\text{compared}}$.

(9.1) If the condition is met, set $\mathbf{\Sigma}_{\text{initial}} = \mathbf{\Sigma}_Z$, $s_{ij} = sign\left(\rho_X(i,j) - \hat{\rho}_X(i,j)\right)$, and $SSE_{\text{compared}} = SSE$.

(9.2) If the condition is not met, proceed to step 10.

Step 10. Check if $SSE < SSE_{\text{target}}$.

(10.1) If the condition is met, return $\mathbf{\Sigma}_Z$.

(10.2) If the condition is not met, return to step 3.

In the process of updating $\mathbf{\Sigma}_Z$ by adding $\mathbf{S} \circ \mathbf{D}$ to $\mathbf{\Sigma}$ from the previous iteration, we would like to emphasize that the exponential distribution and the half-normal distribution were chosen for generating elements in matrix $\mathbf{D}$ because their values are greater than zero. Fig. 2 presents the possible shapes of the half-normal distributions with varying scale parameters.

The introduction of SSE_{compared} serves to accelerate the process, minimizing the computational time needed to obtain $\mathbf{\Sigma}_Z$ with $SSE < SSE_{\text{target}}$, as opposed to not specifying SSE_{compared}. If the current SSE is less than the previous SSE, i.e., $SSE < SSE_{\text{compared}}$, the current SSE will become the criterion for the subsequent iteration or "new" SSE_{compared}. This comparison consistently takes place before evaluating the current SSE against SSE_{target}, and simulation studies will demonstrate that this approach outperforms the algorithm that solely compares SSE with SSE_{target}.

Consider the task of generating a multivariate random variable $\mathbf{X} = (X_1, X_2)^T$ with $SSE_{\text{target}} = 3\times 10^{-6}$. Here, the marginal distributions of X_1 and X_2 are defined by a Poisson distribution with a rate of 10 and a Gamma distribution with a shape parameter of 2 and a scale parameter of 3, respectively. The algorithm that has been proposed yields a value of approximately –0.5273 for the correlation coefficient $\rho_Z(1,2)$ and this will be utilized within the framework of the NORTA algorithm to generate a set of 10,000 points, as depicted in Fig. 3. From the generated points, the estimated correlation coefficient $\hat{\rho}_X(1,2)$ amounts to –0.499557, and the corresponding SSE is calculated to be 1.96267×10^{-7}.

It becomes evident that the proposed algorithm can be utilized for the generation of multivariate random variables. These variables play a crucial role in fields such as data analytics and machine learning [17].

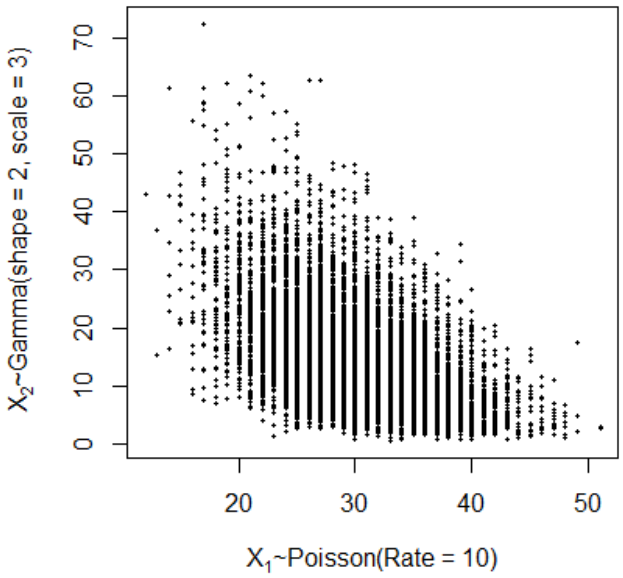

Figure 3. An example of generated points where the marginal distributions follow the Poisson and Gamma distributions.

V. Case Studies of Simulations

For the proposed algorithm, exponential and half-normal distributions are used for generating elements in matrix $\mathbf{D}$, with mean values set to 0.3, 0.15, 0.075, 0.05, 0.025, 0.01, and 0.005. Our algorithm will be compared to the traditional algorithm that estimates the correlation matrix using the method of moments.

In the case of multivariate simulation studies, the number of variables (m) will be set to 2, 3, and 4 variables, with five distributions: Poisson$(\text{rate} = 0.8)$ representing a discrete asymmetric distribution, Uniform$\{1,2,...,10\}$ and Binomial $(n = 3,\ p = 0.5)$ representing discrete symmetric distributions, Exponential$(\text{scale} = 10)$ representing a continuous asymmetric distribution, and Gamma$(\text{shape} = 14.4,\ \text{scale} = 0.0342)$. There are a total of 10 situations listed below:

Situation 1: Discrete asymmetric random variables are divided into 3 cases:

Case 1: $X_1, X_2 \sim$ Poisson;

Case 2: $X_1, X_2, X_3 \sim$ Poisson;

Case 3: $X_1, X_2, X_3, X_4 \sim$ Poisson.

Situation 2: Continuous asymmetric random variables are divided into 3 cases:

Case 4: $X_1, X_2 \sim$ Exponential;

Case 5: X_1, X_2, $X_3 \sim$ Exponential;

Case 6: X_1, X_2, X_3, $X_4 \sim$ Exponential.

Situation 3: Discrete asymmetric and continuous asymmetric random variables are divided into 3 cases:

Case 7: $X_1 \sim$ Poisson, $X_2 \sim$ Exponential;

Case 8: X_1, $X_2 \sim$ Poisson, $X_3 \sim$ Exponential;

Case 9: X_1, $X_2 \sim$ Poisson, X_3, $X_4 \sim$ Exponential.

Situation 4: Discrete asymmetric and discrete symmetric random variables are divided into 3 cases:

Case 10: $X_1 \sim$ Poisson, $X_2 \sim$ Uniform;

Case 11: X_1, $X_2 \sim$ Poisson, $X_3 \sim$ Uniform;

Case 12: X_1, $X_2 \sim$ Poisson, $X_3, X_4 \sim$ Uniform.

Situation 5: Asymmetric continuous and symmetric continuous distributions:

Case 13: $X_1 \sim$ Exponential, $X_2 \sim$ Gamma;

Case 14: X_1, $X_2 \sim$ Exponential, $X_3 \sim$ Gamma;

Case 15: X_1, $X_2 \sim$ Exponential, X_3, $X_4 \sim$ Gamma.

Situation 6: Asymmetric discrete and symmetric continuous distributions:

Case 16: $X_1 \sim$ Poisson, $X_2 \sim$ Gamma;

Case 17: X_1, $X_2 \sim$ Poisson, $X_3 \sim$ Gamma;

Case 18: X_1, $X_2 \sim$ Poisson, X_3, $X_4 \sim$ Gamma.

Situation 7: Symmetric discrete and asymmetric continuous distributions:

Case 19: $X_1 \sim$ Uniform, $X_2 \sim$ Exponential;

Case 20: X_1, $X_2 \sim$ Binomial, $X_3 \sim$ Exponential;

Case 21: X_1, $X_2 \sim$ Binomial, X_3, $X_4 \sim$ Exponential.

Situation 8: Symmetric discrete distributions:

Case 22: X_1, $X_2 \sim$ Binomial;

Case 23: X_1, X_2, $X_3 \sim$ Binomial;

Case 24: X_1, X_2, X_3, $X_4 \sim$ Binomial.

Situation 9: Approximately symmetric continuous distributions:

Case 25: X_1, $X_2 \sim$ Gamma;

Case 26: X_1, X_2, $X_3 \sim$ Gamma;

Case 27: X_1, X_2, X_3, $X_4 \sim$ Gamma.

Situation 10: Discrete symmetric random variables with approximately symmetric continuous variables are divided into 3 cases:

Case 28: $X_1 \sim$ Binomial, $X_2 \sim$ Gamma;

Case 29: $X_1, X_2 \sim$ Binomial, $X_3 \sim$ Gamma;

Case 30: $X_1, X_2 \sim$ Binomial, $X_3, X_4 \sim$ Gamma.

The desired correlation matrix ($\mathbf{\Sigma}_X$) is defined as follows: For the 2-variable cases, $\mathbf{\Sigma}_X$ is set to

$$\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix},$$

and SSE_{target} is 3×10^{-6}. For the 3-variable cases in situations 1 – 7, $\mathbf{\Sigma}_X$ is set to

$$\begin{bmatrix} 1 & 0.2 & -0.6 \\ 0.2 & 1 & 0.2 \\ -0.6 & 0.2 & 1 \end{bmatrix},$$

and SSE_{target} is 6×10^{-5}. For the 3-variable cases in situations 8 – 10, $\mathbf{\Sigma}_X$ is equal to

$$\begin{bmatrix} 1 & 0.2 & -0.8 \\ 0.2 & 1 & 0.2 \\ -0.8 & 0.2 & 1 \end{bmatrix},$$

and the acceptable squared sum of errors is 6×10^{-5}. For the 4-variable cases in situations 1 – 7, $\mathbf{\Sigma}_X$ is equal to

$$\begin{bmatrix} 1 & 0.4 & 0.2 & -0.6 \\ 0.4 & 1 & 0.4 & -0.3 \\ 0.2 & 0.4 & 1 & -0.1 \\ -0.6 & -0.3 & -0.1 & 1 \end{bmatrix},$$

and SSE_{target} is 3.5×10^{-4}. Finally, for the 4-variable cases in situations 8 – 10 , $\mathbf{\Sigma}_X$ is equal to

$$\begin{bmatrix} 1 & 0.7 & 0.5 & -0.9 \\ 0.7 & 1 & 0.7 & -0.6 \\ 0.5 & 0.7 & 1 & -0.3 \\ -0.9 & -0.6 & -0.3 & 1 \end{bmatrix},$$

and SSE_{target} is 3.5×10^{-4}.

The research presented in this paper was conducted using a computer with an Intel(R) Core(TM) i7-8565U CPU, clocked at a speed between 1.80 GHz and 1.99 GHz, and equipped with 16GB of RAM. The corresponding codes were programmed in R, version 4.2.2 [18].

VI. Results and Conclusions

Table I presents the average time in seconds needed to identify the initial correlation for the NORTA algorithm, along with the average of SSEs. These results are derived from the optimal configuration of all potential mean values for both exponential and half-normal distributions. For example, in case 1, the average times for the exponential distribution with means of 0.3, 0.15, 0.075, 0.05, 0.025, and 0.01 are 2.65, 1.51, 1.30, 0.98, 0.82, 1.38, and 2.61, respectively. For the half-

normal distribution with the specified means, the average times are 2.95, 1.67, 1.75, 0.98, 0.74, 1.33, and 2.49, respectively. Consequently, the best average time of 0.74 is obtained from using the half-normal distribution and will only be selected to be shown in Table 1. For such a configuration, the corresponding SSE is about 0.8×10^{-6}, considered to be the best average *SSE*. The notation (H) means the setting with a half-normal distribution yields the lowest time, and (E) means the exponential distribution.

As the number of variables increases, the average time required to complete the process increases significantly. Generating 4-dimensional variables cannot be accomplished within 15 minutes, leading to a lack of *SSE* values in such cases. In all cases, the *SSE*s of the proposed algorithm are lower. While the half-normal distribution may be favored over the exponential distribution, it is not clearly evident which one is superior. Nevertheless, utilizing small random numbers from both distributions as elements of matrix **D** proves more effective than using uniform random numbers, which are employed in traditional simulation-based methods.

In conclusion, using a quasi-Monte Carlo can improve the accuracy of double integration in (4), and besides only determining SSE_{target}, using the additional criterion of SSE_{compared} can reduce the time it takes to identify the initial correlation matrix for NORTA.

TABLE I. AVERAGE TIME AND ERRORS

m	Case	Simulation-based Algorithm		Proposed Algorithm	
		Average Time	*Average SEE*	*Average Time*	*Average SSE*
2	1	8.24 (H)	$5.6\text{x}10^{-6}$	0.74 (H)	$0.8\text{x}10^{-6}$
	4	3.46 (H)	$2.9\text{x}10^{-6}$	0.31 (H)	$1.5\text{x}10^{-6}$
	7	5.2 (H)	$4.0\text{x}10^{-6}$	0.26 (E)	$1.0\text{x}10^{-6}$
	10	2.42 (E)	$9.4\text{x}10^{-6}$	0.27 (H)	$1.2\text{x}10^{-6}$
	13	5.47 (E)	$11.2\text{x}10^{-6}$	1.12 (E)	$2.1\text{x}10^{-6}$
	16	7.29 (H)	$9.4\text{x}10^{-6}$	1.67 (H)	$2.0\text{x}10^{-6}$
	19	1.17 (H)	$8.1\text{x}10^{-6}$	0.21 (H)	$0.5\text{x}10^{-6}$
	22	2.72 (H)	$10.8\text{x}10^{-6}$	0.74 (H)	$0.6\text{x}10^{-6}$
	25	1.82 (E)	$19.7\text{x}10^{-6}$	0.51 (H)	$5.1\text{x}10^{-6}$
	28	4.06 (H)	$16.6\text{x}10^{-6}$	0.89 (H)	$2.2\text{x}10^{-6}$
3	2	275.30 (E)	$4.8\text{x}10^{-5}$	5.99 (E)	$3.6\text{x}10^{-5}$
	5	256.80 (E)	$11.9\text{x}10^{-5}$	5.03 (E)	$9.2\text{x}10^{-5}$
	8	179.44 (E)	$6.9\text{x}10^{-5}$	1.24 (E)	$2.3\text{x}10^{-5}$
	11	184.48 (E)	$11.3\text{x}10^{-5}$	2.77 (H)	$3.1\text{x}10^{-5}$
	14	246.49 (H)	$12.4\text{x}10^{-5}$	5.16 (E)	$5.0\text{x}10^{-5}$
	17	242.74 (E)	$9.1\text{x}10^{-5}$	4.45 (E)	$3.4\text{x}10^{-5}$
	20	197.73 (E)	$7.9\text{x}10^{-5}$	1.34 (H)	$3.3\text{x}10^{-5}$
	23	129.25 (H)	$8.3\text{x}10^{-5}$	2.04 (H)	$3.4\text{x}10^{-5}$
	26	137.92 (E)	$4.4\text{x}10^{-5}$	2.02 (H)	$6.0\text{x}10^{-5}$
	29	213.35 (H)	$5.5\text{x}10^{-5}$	3.10 (H)	$3.0\text{x}10^{-5}$
4	3	> 900	-	9.25 (H)	$2.7\text{x}10^{-4}$
	6	> 900	-	3.13 (H)	$2.8\text{x}10^{-4}$
	9	> 900	-	4.44 (H)	$2.6\text{x}10^{-4}$
	12	> 900	-	1.99 (H)	$2.6\text{x}10^{-4}$
	15	> 900	-	11.36 (E)	$3.7\text{x}10^{-4}$
	18	> 900	-	7.72 (H)	$2.6\text{x}10^{-4}$
	21	> 900	-	2.49 (E)	$2.5\text{x}10^{-4}$
	24	> 900	-	6.51 (H)	$2.5\text{x}10^{-4}$
	27	> 900	-	2.88 (E)	$3.2\text{x}10^{-4}$
	30	> 900	-	7.30 (E)	$2.0\text{x}10^{-4}$

REFERENCES

[1] M. V. Leduc, S. Poledna, and S. Thurner, "Systemic risk management in financial networks with credit default swaps," *J. Netw. Theory Finance*, vol. 2, no. 3, pp. 1-29, 2016.

[2] S. Qiu, M. Zhang, and Y. Yuan, "Performance and reliability analysis of water distribution systems under cascading failures and the identification of crucial pipes," *PLoS ONE*, vol. 9, no. 2, e88445, Feb. 2014.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2672-2680.

[4] S. T. Akhavan Niaki and B. Abbasi, "Generating correlation matrices for normal random vectors in norta algorithm using artificial neural networks," *J. Uncertain Syst.*, vol. 2, no. 3, pp. 192-201, 2008.

[5] M. C. Cario and B. L. Nelson, "Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix," Technical Report, Department of Industrial Engineering and Management Sciences, Northwestern University, IL, 1997.

[6] K. V. Mardia, "Families of Bivariate Distributions," in *Griffin's Statistical Monographs and Courses No. 27*. Darien, CT: Hafner Publishing Co., 1970

[7] S. T. Li and J. L. Hammond, "Generation of pseudorandom numbers with specified univariate distributions and correlation coefficients," in *IEEE Trans.*, Man, and Cybernetics, vol. SMC-5, no. 5, pp. 557-561, Sept. 1975.

[8] Q. Xiao, "Generating correlated random vector involving discrete variables," *Commun. Stat. - Theory Methods*, vol. 46, no. 4, pp. 1594-1605, Feb. 2017.

[9] A. Der Kiureghian and P.-L. Liu, "Structural reliability under incomplete probability information," *J. Eng. Mech.*, vol. 112, no. 1, pp. 15-33, Jan. 1986.

[10] P.-L. Liu and A. Der Kiureghian, "Multivariate distribution models with prescribed marginals and covariances," *Probabilistic Eng. Mech.*, vol. 1, no. 2, pp. 105-112, Jun. 1986.

[11] H. Li, Z.-Z. Lü, and X. Yuan, "Nataf transformation based point estimate method," *Chin. Sci. Bull.*, vol. 53, pp. 2586-2592, 2008.

[12] A. N. Avramidis, N. Channouf, and P. L'Ecuyer, "Efficient correlation matching for fitting discrete multivariate distributions with arbitrary marginals and normal-copula dependence," *INFORMS J. Comput.*, vol. 21, no. 1, pp. 88-106, Feb. 2009.

[13] A. N. Avramidis, "Constructing discrete unbounded distributions with Gaussian-copula dependence and given rank correlation," *INFORMS J. Comput.*, vol. 26, no. 2, pp. 269-279, 2013.

[14] M. R. Niavarani and A. J. Smith, "Modeling and generating multi-variate-attribute random vectors using a new simulation method

combined with NORTA algorithm," *J. Uncertain Syst.*, vol. 7, no. 2, pp. 83-91, 2013.

[15] W. J. Morokoff and R. E. Caflisch, "Quasi-Monte Carlo integration," *J. Comput. Phys.*, vol. 122, no. 2, pp. 218-230, 1995.

[16] J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numer. Math.*, vol. 2, pp. 84-90, 1960.

[17] P. Srisuradetchai and W. Panichkitkosolkul, "Using ensemble machine learning methods to forecast particulate matter (PM2.5) in Bangkok, Thailand," in Multi-disciplinary Trends in *Artificial Intelligence*, MIWAI 2022, O. Surinta and K. Kam Fung Yuen, Eds., Lect. Notes Comput. Sci, vol. 13651. Cham, Switzerland: Springer, 2022.

[18] R Core Team, "R: A language and environment for statistical computing," R Foundation for Statistical Computing, Vienna, Austria, 2022.