Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

A METHOD FOR FAST GENERATION OF BIVARIATE POISSON RANDOM VECTORS

Kaeyoung Shin
Raghu Pasupathy

Industrial and Systems Engineering
250 Durham Hall, Virginia Tech
Blacksburg, VA 24060, U.S.A.

ABSTRACT

It is well known that trivariate reduction — a method to generate two dependent random variables from three independent random variables — can be used to generate Poisson random variables with specified marginal distributions and correlation structure. The method, however, works only for positive correlations. Moreover, the proportion of feasible positive correlations that can be generated through trivariate reduction deteriorates rapidly as the discrepancy between the means of the target marginal distributions increases. We present a specialized algorithm for generating Poisson random vectors, through appropriate modifications to trivariate reduction. The proposed algorithm covers the entire range of feasible correlations in two dimensions, and preliminary tests have demonstrated very fast preprocessing and generation times.

1 INTRODUCTION

Johnson, Kotz, and Balakrishnan (1997) define the bivariate Poisson distribution as the joint distribution of the random variables

$$X_1 = Y_1 + Y_{12} \text{ and } X_2 = Y_2 + Y_{12}, \tag{1}$$

where Y_1, Y_2, and Y_{12} are mutually independent Poisson random variables with means λ_1, λ_2, and λ_{12} respectively. As shown in (Johnson, Kotz, and Balakrishnan 1997), the resulting joint probability mass function of (X_1, X_2) is

$$\Pr\{X_1 = x_1, X_2 = x_2\} = e^{-(\lambda_1+\lambda_2+\lambda_{12})} \sum_{i=0}^{\min(x_1,x_2)} \frac{\lambda_1^{x_1-i}\lambda_2^{x_2-i}\lambda_{12}^{i}}{(x_1-i)!(x_2-i)!i!}. \tag{2}$$

It can also be easily shown that X_1 and X_2 have Poisson distributions with means $\lambda_1+\lambda_{12}$ and $\lambda_2+\lambda_{12}$ respectively, and correlation $\mathrm{Corr}(X_1, X_2)$ given by

$$\frac{\mathrm{Var}(Y_{12})}{\sqrt{\mathrm{Var}(X_1)\mathrm{Var}(X_2)}} = \frac{\lambda_{12}}{\sqrt{(\lambda_1+\lambda_{12})(\lambda_2+\lambda_{12})}}.$$

It is well-known amongst simulation researchers that this definition of a bivariate Poisson distribution automatically provides a method to generate correlated Poisson random variates: to generate the random vector (X_1, X_2) such that X_1 and X_2 have Poisson distributions with specified means λ and λ', and specified correlation $\rho > 0$, simply generate three independent Poisson random variables Y_1, Y_2, and Y_{12}, with carefully chosen means λ_1, λ_2 and λ_{12}, and obtain (X_1, X_2) through the two operations in (1). Specifically, the parameters λ_1, λ_2, and λ_{12} are obtained by solving the following three equations corresponding to matching two target means, and one target correlation:

$$\begin{aligned} \lambda &= \lambda_1 + \lambda_{12}, \\ \lambda' &= \lambda_2 + \lambda_{12}, \\ \rho &= \frac{\lambda_{12}}{\sqrt{(\lambda_1+\lambda_{12})(\lambda_2+\lambda_{12})}}. \end{aligned} \tag{3}$$

Solving the system (3) gives us

$$\lambda_{12} = \rho\sqrt{\lambda\lambda'}, \lambda_1 = \lambda - \lambda_{12}, \lambda_2 = \lambda' - \lambda_{12}. \tag{4}$$

While elegant, the above *trivariate reduction* (TR) method has two important drawbacks when used in the context of generating correlated Poisson random vectors.

D.1. TR cannot be used when the target correlation ρ is negative;

D.2. Even when the target correlation ρ is positive, the vector (X_1, X_2) obtained through TR may not be able to attain the target correlation, while also achieving the specified marginal distributions.

The disadvantage D.1 is fairly easy to internalize upon noticing that the correlation between X_1 and X_2 is induced through a common random variable Y_{12}. The other two random variables involved, Y_1 and Y_2, are independent. Therefore, the resulting correlation between X_1 and X_2 ought to be positive.

As a possible remedy to D.1, an alternate generation scheme such as $X_1 = Y_1 + Y_{12}, X_2 = Y_2 - Y_{12}$ can be arranged to induce the desired negative correlation between X_1 and X_2. This will not work, however, because the resulting X_2 will not have a Poisson distribution.

The disadvantage D.2 is slightly more subtle. To ease exposition, we first present the following definition introduced by Ghosh and Henderson (2003).

Definition 1 *A product-moment (rank) correlation matrix Σ is* feasible *for a given set of marginal distributions $F_1, F_2, \ldots, F_d$ if there exists a random vector X with marginal distributions $F_1, F_2, \ldots, F_d$ and product-moment (rank) correlation matrix Σ.*

To illustrate feasible correlation matrices, consider two Poisson random variables X_1 and X_2 having respective means $\lambda = 0.5$ and $\lambda' = 0.5$. It can be shown that the largest achievable positive correlation between X_1 and X_2 is 1, while the largest achievable negative correlation between X_1 and X_2 is -0.5. Therefore, any correlation matrix

$$\Sigma = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix},$$

with r values in the interval $[-0.5, 1]$ is a *feasible* correlation matrix for the vector (X_1, X_2). The matrix Σ is a correlation matrix, but not feasible, if r lies in the interval $[-1, -0.5)$.

More generally, as shown in (Whitt 1976), if random variables X_1 and X_2 have the cumulative distribution function (cdf) $F(x)$, and U is a random variable that is uniformly distributed between 0 and 1, then $\text{Corr}\left(F^{-1}(U), F^{-1}(U)\right)$ is the maximum achievable, and $\text{Corr}\left(F^{-1}(U), F^{-1}(1-U)\right)$ the minimum achievable, correlations between X_1 and X_2 respectively. Therefore the *feasible* set of correlations between X_1 and X_2 is $\left[\text{Corr}\left(F^{-1}(U), F^{-1}(1-U)\right), \text{Corr}\left(F^{-1}(U), F^{-1}(U)\right)\right]$. Figure 1 depicts this feasible set when X_1 and X_2 are Poisson random variables. The figure is plotted as a function of the larger desired mean λ (assumed without loss of generality) of X_1 and X_2, and the ratio k of the smaller to the larger desired means of X_1 and X_2. So, for a given λ and k, a vertical line between the corresponding upper and lower curves depicts the range of feasible correlations.

Continuing our discussion of disadvantage D.2, recall that we have denoted λ and λ' as the target means, and ρ as the target correlation for the Poisson random variables X_1 and X_2. The solution (4), to the system of equations in (3), implies that λ, λ' and ρ should satisfy $\lambda \geq \rho\sqrt{\lambda\lambda'}$, and $\lambda' \geq \rho\sqrt{\lambda\lambda'}$. Otherwise, one of λ_1 and λ_2 will be negative, implying that TR cannot be used to generate the vector (X_1, X_2) with the desired marginal distributions and correlation. We state this formally through Propositions 1 and 2.

Proposition 1 *Let X_1 and X_2 be Poisson random variables with means λ and λ'. Assume, without loss in generality, that $\lambda \geq \lambda'$. Denote the maximum and minimum achievable correlation between X_1 and X_2 as $\rho^+(\lambda, \lambda')$ and $\rho^-(\lambda, \lambda')$ respectively. Also, denote $k = \lambda'/\lambda$. Then,*

(i) *for fixed k,*

$$\lim_{\lambda\to\infty} \rho^+(\lambda, k\lambda) = 1, \lim_{\lambda\to\infty} \rho^-(\lambda, k\lambda) = -1;$$

(ii) *for fixed k,*

$$\lim_{\lambda\to 0} \rho^+(\lambda, k\lambda) = \sqrt{k}, \lim_{\lambda\to 0} \rho^-(\lambda, k\lambda) = 0.$$

Proposition 2 *Following the notation of Proposition 1, the range of correlations that can be generated using TR is $[0, \sqrt{k}]$.*

Propositions 1 and 2 point to a rather serious problem in TR — as the discrepancy between the desired means λ and λ' increases, assertions (i) and (ii) in Proposition 1 suggest that the range of feasible positive correlations expands to the interval $[0, 1)$. By contrast, as Proposition 2 suggests, the range of correlations that can be generated using TR is always $[0, \sqrt{k}]$. For example, for $k = 0.01$, as $\lambda \to \infty$, the feasible range of positive correlations tends to $[0, 1)$, while the range of correlations that can be generated by TR remains $[0, 0.1]$.

In this paper, we introduce an algorithm that resolves both of the issues D.1 and D.2, through an appropriate modification of TR. The result is an algorithm (Section 2) which can generate every feasible correlation $\rho \in [\rho^-(\lambda, k\lambda), \rho^+(\lambda, k\lambda)]$, for given k and λ. Like the "NORmal To Anything" (NORTA) method (Cario, Nelson, Roberts, and Wilson 2002), the proposed algorithm has a preprocessing step, which we solve using a fast numerical procedure. We detail this procedure in Section 3, and briefly report on numerical performance in Section 4. We summarize and discuss ongoing research in Section 5. A MATLAB implementation of the proposed generation algorithm is available for download at `<https://filebox.vt.edu/users/pasupath/pasupath.htm>`.

2 ALGORITHM DESCRIPTION

Recall that the objective is to generate the random vector (X_1, X_2) such that X_1 has a Poisson distribution with mean λ, X_2 has a Poisson distribution with mean λ', and $\text{Corr}(X_1, X_2) = \rho$, where $\lambda, \lambda' > 0$ and $\rho \in (-1, 0) \cup (0, 1)$

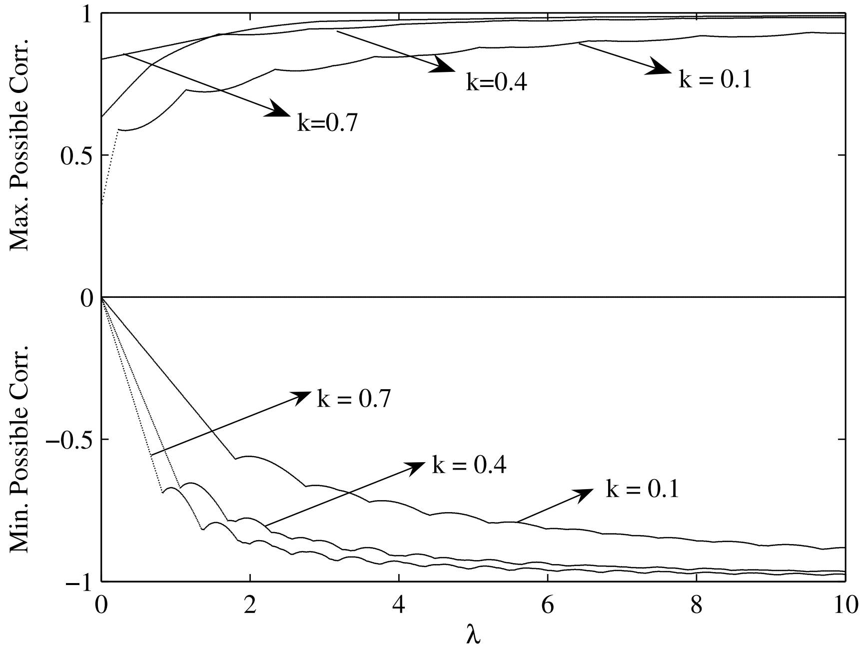

Figure 1: Maximum and minimum achievable correlations between two Poisson random variables. In the figure, λ is the larger of the two desired means, and k is the ratio of the smaller desired mean to the larger desired mean.

are given. We assume, without loss in generality, that $\lambda \geq \lambda'$ and denote $k = \lambda'/\lambda$.

Our assumption about $\rho \in (-1,0) \cup (0,1)$ creates the possibility of the desired correlation being infeasible, i.e., $\rho > \rho^+(\lambda, k\lambda)$ or $\rho < \rho^-(\lambda, k\lambda)$. This problem of infeasibility is not a complication because it is automatically detected at the end of the preprocessing step. In other words, the proposed algorithm is such that nothing special needs to be done to check for an infeasible problem.

Denote $F_\lambda^{-1}(y) = \inf\{x : F(x) > y\}$, where $F_\lambda(x)$ is the Poisson cdf with mean λ. Let U be a random variable that is uniformly distributed between 0 and 1. Then the proposed algorithm takes the following form.

$$\begin{aligned} &X_1 = Y_1 + F_{\lambda^*}^{-1}(U), \quad X_2 = Y_2 + F_{k\lambda^*}^{-1}(U) \qquad \text{if } \rho > 0; \\ &X_1 = Y_1 + F_{\lambda^*}^{-1}(U), \quad X_2 = Y_2 + F_{k\lambda^*}^{-1}(1-U) \quad \text{if } \rho < 0; \end{aligned} \tag{5}$$

We draw attention to three aspects of the proposed operations. First, when $\rho > 0$, i.e., when positive correlation between X_1 and X_2 is sought, we use common random numbers, as in TR. When $\rho < 0$, the operations suggest using antithetic variates to induce negative correlation between X_1 and X_2.

Second, we note that for both cases, $\rho > 0$ and $\rho < 0$, unlike TR, there is no "common random variable." Instead, the random variables inducing correlation are obtained through inversion of two different Poisson cdfs. The means of these Poisson cdfs are in the same ratio as the target means λ and λ'.

Third, the value of λ^* needs to be determined as part of the preprocessing step, so that the resulting random variables X_1, X_2 attain the target means, and the target correlation.

2.1 Algorithm Listing

Generating a random vector (X_1, X_2) through the proposed method is easy, at least in principle. For given $\lambda > 0, \lambda' > 0$, and $\rho \in (-1,0) \cup (0,1)$:

1. Solve for λ^* through the preprocessing step;
2. Generate $U_1 \sim U(0,1), U_2 \sim U(0,1), U_3 \sim U(0,1)$ independently;
3. $Y_1 \leftarrow F_{\lambda-\lambda^*}^{-1}(U_1)$, $Y_2 \leftarrow F_{\lambda'-k\lambda^*}^{-1}(U_2)$;
4. $Y_{12} \leftarrow F_{\lambda^*}^{-1}(U_3)$;
5. If $\rho > 0$, then $Y_{12}' \leftarrow F_{k\lambda^*}^{-1}(U_3)$;
   if $\rho < 0$ then $Y_{12}' \leftarrow F_{k\lambda^*}^{-1}(1-U_3)$;
6. $X_1 \leftarrow Y_1 + Y_{12}, X_2 \leftarrow Y_2 + Y_{12}'$;
7. Deliver (X_1, X_2).

We discuss Step 1 in detail in Section 3. Inverting a Poisson cdf, required in Steps 3, 4, and 5, can be done very efficiently through existing, mature, Poisson random variate generation routines (Kemp and Kemp 1991, Schmeiser and Kachitvichyanukul 1981, Devroye 1986).

2.2 Rationale

It is clear that the disadvantage D.1 is addressed through the proposed algorithm. What is less intuitive is to what extent the proposed algorithm addresses disadvantage D.2.

For given means λ, λ', the proposed algorithm is capable of generating any specified correlation in the feasible range $[\rho^-(\lambda, k\lambda), \rho^+(\lambda, k\lambda)]$. To see this, consider the $\rho > 0$ operation in (5). It is clear from construction that the random variables Y_1, Y_2, $F_{\lambda^*}^{-1}(U)$, and $F_{k\lambda^*}^{-1}(U)$ are each Poisson distributed with respective means $\lambda - \lambda^*, \lambda' - k\lambda^*, \lambda^*$, and $k\lambda^*$. Therefore, the random variables X_1 and X_2 will have the correct marginal distributions, provided the quantities $\lambda - \lambda^*$ and $\lambda' - k\lambda^*$ remain positive. This, however, can be ensured by restricting λ^* to the interval $[0, \lambda]$, after recalling that $\lambda \geq \lambda^*$ and $k \leq 1$. A similar argument holds for the $\rho < 0$ case as well.

What range of correlations are covered if we restrict λ^* to the interval $[0, \lambda]$? To answer this question, again consider the $\rho > 0$ case in (5). As $\lambda^* \to 0$, we have $\text{Corr}(X_1, X_2) \to 0$, giving us the trivial uncorrelated case. On the other extreme, as $\lambda^* \to \lambda$, we have $\lambda - \lambda^* \to 0$, $\lambda' - k\lambda^* \to 0$, and $k\lambda^* \to \lambda'$. These three implications together mean that Y_1 and Y_2 vanish, and $\text{Corr}(X_1, X_2) \to \text{Corr}(F_\lambda^{-1}(U), F_{k\lambda}^{-1}(U)) = \rho^+(\lambda, k\lambda)$. Furthermore, it can be shown that $\text{Corr}(X_1, X_2)$ is a continuous function of λ^*. These three facts — $\text{Corr}(X_1, X_2) \to 0$ as $\lambda^* \to 0$, $\text{Corr}(X_1, X_2) \to \rho^+(\lambda, k\lambda)$ as $\lambda^* \to \lambda$, and the continuity of $\text{Corr}(X_1, X_2)$ as a function of λ^* — ensure that the entire range of positive correlations $[0, \rho^+(\lambda, k\lambda)]$ can be achieved through the proposed algorithm. Similar arguments for the $\rho < 0$ case imply that the entire range of negative correlations $[\rho^-(\lambda, k\lambda), 0]$ can also be achieved through the proposed algorithm.

Before we state the above arguments formally through Proposition 3, we also note in passing that we can achieve a similar effect, i.e., obtaining the entire range of feasible correlations, through

$$\begin{array}{lll} X_1 = Y_1 + F_{\lambda_1^*}^{-1}(U), & X_2 = Y_2 + F_{\lambda_2^*}^{-1}(U) & \text{if } \rho > 0; \\ X_1 = Y_1 + F_{\lambda_1^*}^{-1}(U), & X_2 = Y_2 + F_{\lambda_2^*}^{-1}(1-U) & \text{if } \rho < 0; \end{array} \tag{6}$$

instead of (5). The operation (6), however, provides no advantages over (5), at least in two dimensions. It does have the disadvantage of making the preprocessing step a two-dimensional search, as opposed to the monotone one-dimensional search afforded by (5).

Proposition 3 *Let Y_1, Y_2 be Poisson random variables with means $\lambda - \lambda^*$ and $\lambda' - k\lambda^*$ respectively, where $k = \lambda'/\lambda \leq 1$, and $0 < \lambda^* \leq \lambda$. Let U be a random variable that is mutually independent with Y_1 and Y_2, and uniformly distributed between 0 and 1. Also, denote $X_1 = Y_1 + F_{\lambda^*}^{-1}(U)$. Then*

(i) $\lim_{\lambda^* \to \lambda} Corr\left(X_1, Y_2 + F_{k\lambda^*}^{-1}(U)\right) = \rho^+(\lambda, k\lambda)$;

(ii) $\lim_{\lambda^* \to \lambda} Corr\left(X_1, Y_2 + F_{k\lambda^*}^{-1}(1-U)\right) = \rho^-(\lambda, k\lambda)$;

(iii) *the two functions $Corr\left(X_1, Y_2 + F_{k\lambda^*}^{-1}(U)\right)$ and $Corr\left(X_1, Y_2 + F_{k\lambda^*}^{-1}(1-U)\right)$ are continuous in λ^*.*

3 PREPROCESSING STEP (SOLVING FOR λ^*)

We see from (5) that X_1 and X_2 have the correct marginal distributions. The more challenging question is that of identifying λ^* so that the target correlation ρ is attained. In this section, we detail a fast numerical procedure that can be used to identify λ^*.

From (5), the correlation $\text{Corr}(X_1, X_2)$ as a function of λ, k, and λ^* is given by

$$\text{Corr}(X_1, X_2) = \begin{cases} \frac{1}{\lambda\sqrt{k}} \left(\mathrm{E}\left[F_{\lambda^*}^{-1}(U) F_{k\lambda^*}^{-1}(U)\right] - k\lambda^{*2} \right) & \text{if } \rho > 0; \\ \frac{1}{\lambda\sqrt{k}} \left(\mathrm{E}\left[F_{\lambda^*}^{-1}(U) F_{k\lambda^*}^{-1}(1-U)\right] - k\lambda^{*2} \right) & \text{if } \rho < 0. \end{cases}$$

From the above expression for $\text{Corr}(X_1, X_2)$, identifying λ^* satisfying $\text{Corr}(X_1, X_2) = \rho$ amounts to solving the following generic root-finding problems: given λ, k, ρ,

$$\text{find } x = \lambda^* \text{ satisfying } h(x) = \rho\lambda\sqrt{k}, \tag{7}$$

where

$$h(x) = \begin{cases} \mathrm{E}\left[F_x^{-1}(U) F_{kx}^{-1}(U)\right] - kx^2 & \text{if } \rho > 0; \\ \mathrm{E}\left[F_x^{-1}(U) F_{kx}^{-1}(1-U)\right] - kx^2 & \text{if } \rho < 0. \end{cases}$$

In this section, we detail a solution for the $\rho < 0$ case of the root-finding problem (7). Details for the $\rho > 0$ case are very similar to the $\rho < 0$ case.

3.1 Recursion, Function and Derivative Computation

For solving root-finding problem (7), we use the following Newton recursion on $h(x)$:

$$x = x + \frac{1}{h'(x)} \left(\rho\lambda\sqrt{k} - h(x) \right). \tag{8}$$

The efficiency of recursion (8) is immensely helped by two aspects: (i) the function $h(x)$ is monotone decreasing in $(0, \infty)$, and its derivative $h'(x)$ exists everywhere in $(0, \infty)$ except for a countable set; and (ii) the function $h(x)$, and its derivative $h'(x)$, can be computed efficiently. In what follows, we elaborate on (ii). We do not provide a proof for (i).

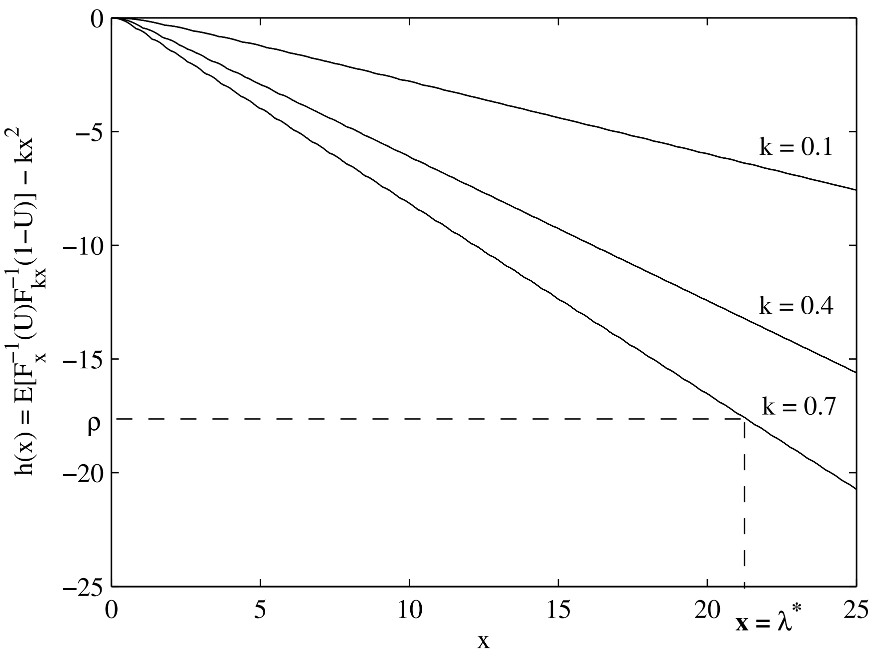

Figure 2: Preprocessing through one-dimensional root-finding search on the covariance function $h(x)$.

We first note that, corresponding to the linear portion of the curves in Figure 1, $\mathrm{E}\left[F_x^{-1}(U)F_{kx}^{-1}(1-U)\right]=0$ if

$$F_x(0)+F_{kx}(0)=e^{-x}+e^{-kx}\geq 1. \tag{9}$$

In such a case, $h(x)=-kx^2$ implying that $\lambda^*=\sqrt{-\rho\lambda/\sqrt{k}}$.

If (9) does not hold, we estimate $h(x)=\int_0^1 F_x^{-1}(u)F_{kx}^{-1}(1-u)\,du-kx^2$ starting from the "middle region" of the integral, and progressively summing out to the upper and lower tails. Specifically, we first compute $m=\mathrm{Max}\{k\in Z^+ : F_x(k-1)\leq 0.5\}$, and the cumulative probability $F_x(m)$, where $Z^+=\{0,1,2,\ldots\}$ denotes the set of non-negative integers. One efficient way to compute m (contrary to appearance, m as defined is not the median) and $F_x(m)$ is through J-fraction approximations given in (Kemp and Kemp 1991). These approximations are highly accurate analytic expressions for $F_x(r)$, where r is the "round-off" value of x, i.e., the integer that satisfies $r+\alpha=x, -0.5\leq\alpha<0.5$. For example, we have found that for $x>15$, the approximation for $F_x(r)$ is generally accurate to within 10^{-6}.

In order to ease computation of $h(x)$, we express

$$h(x)=\int_0^{0.5}F_x^{-1}(u)F_{kx}^{-1}(1-u)\,du+\int_0^{0.5}F_{kx}^{-1}(u)F_x^{-1}(1-u)\,du-kx^2. \tag{10}$$

The first of the integrals on the right-hand side of (10) can be written as

$$\int_0^{0.5}F_x^{-1}(u)F_{kx}^{-1}(1-u)\,du=\sum_{j=m}^{1}\left(B_j+C_j+\sum_{i=l_j}^{u_j}T_{ji}\right), \tag{11}$$

where

$$l_j=\begin{cases}\mathrm{Min}\{n\in Z^+ : 1-F_{kx}(n)\leq F_x(j)\}, & \text{if } j=0,1,2,\ldots,m-1;\\ \mathrm{Min}\{n\in Z^+ : 1-F_{kx}(n)\leq 0.5\}, & \text{if } j=m;\end{cases}$$

$$B_j=\begin{cases}jl_j\left(F_x(j)-\mathrm{Max}\left(F_x(j-1),1-F_{kx}(l_j)\right)\right), & \text{if } j=0,1,2,\ldots,m-1;\\ ml_m\left(0.5-\mathrm{Max}\left(1-F_{kx}(l_m),F_x(m-1)\right)\right), & \text{if } j=m;\end{cases}$$

$$C_j=\begin{cases}jl_{j-1}\left(\mathrm{Min}(1-F_{kx}(l_{j-1}-1),F_x(j))-F_x(j-1)\right), & \text{if } j=0,1,2,\ldots,m-1;\\ ml_{m-1}\left(\mathrm{Min}\left(0.5,1-F_{kx}(l_{m-1}-1)\right)-F_x(m-1)\right), & \text{if } j=m;\end{cases}$$

$$T_{ij}=\ (i+1)\,j\left(F_{kx}(i+1)-F_{kx}(i)\right).$$

The second of the integrals on the right-hand side of (10) can be expressed similarly.

The terms B_j, C_j, and T_j appearing within (11) should not be computed explicitly during actual implementation. Instead, they can be computed on fly, once the cumulative probability $F_x(m)$ and the quantity m are available. For brevity, we do not go into these programming details.

The derivative $h'(x)$ can be obtained through direct differentiation of the summation expressions for $h(x)$, and noting that the probability mass $P_{kx}(i) = F_{kx}(i) - F_{kx}(i-1)$, and its derivative with respect to x,

$$P'_{kx}(i) = k\left(P_{kx}(i-1) - P_{kx}(i)\right).$$

3.2 Bounds on Truncation Error

Computing $h(x)$ is based on a finite-sum, as described in Section 3.1. However, the number of terms $m-1$ within the outside summation could become very large, when x is large. Therefore, from a computational standpoint, it would be useful to appropriately truncate this summation, making sure that the terms excluded add to less than a prespecified tolerance ε.

The following bounds are the basis of our proposed truncation rules.

Proposition 4 *If $P_x(j)$ and $F_x(j)$ denote the probability mass function and cumulative distribution function of the Poisson distribution with mean x, then*

$$(i) \sum_{j=s}^{\infty} jP_x(j) = x\left(1 - F_x(s-2)\right), s \geq 2;$$

$$(ii) \sum_{j=s}^{\infty} j^2 P_x(j) = x^2\left(1 - F_x(s-3)\right) + x\left(1 - F_x(s-2)\right), s \geq 3.$$

From Proposition 4, we can show that

$$\sum_{j=s_\varepsilon+1}^{1} \left(B_j + C_j + \sum_{i=l_j}^{u_j} T_{ji}\right) \leq s_\varepsilon kx\left(1 - F_{kx}(r_\varepsilon - 2)\right), \quad (12)$$

where $r_\varepsilon = \mathrm{Max}\{n \in Z^+ : 1 - F_{kx}(n) \geq F_x(s_\varepsilon - 1)\}$. Therefore, if the integral $\int_0^{0.5} F_x^{-1}(u) F_{kx}^{-1}(1-u)\, du$ comprising $h(x)$ needs to be computed to within tolerance ε, stop the outside summation in (11) when the right-hand side of (12) falls below ε. A similar procedure can be adopted when computing the other integral $\int_0^{0.5} F_{kx}^{-1}(u) F_x^{-1}(1-u)\, du$ that comprises $h(x)$.

3.3 Initial Guess

Motivated by Figure 2, the initial guess x_0 for the recursion (8) is obtained through a linear approximation $l(x)$ to the function $h(x)$. As the following proposition asserts, the "global slope" of $h(x)$ is $-\sqrt{k}$. Therefore, the slope of the linear approximation $l(x)$ is assumed to be $-\sqrt{k}$.

Proposition 5 *The function*

$$h(x) = E\left[F_x^{-1}(U) F_{kx}^{-1}(1-U)\right] - kx^2$$

satisfies

$$\lim_{x \to \infty} \frac{h(x)}{x} = -\sqrt{k}.$$

The intercept c of the linear approximation $l(x)$ is estimated empirically, using a multi-regime regression line.

For a given problem instance, i.e., for given λ, k, and ρ, the initial guess x_0 for the recursion (8) is obtained by solving for x from the equation

$$l(x) = -\sqrt{k}x + c = \rho\lambda\sqrt{k},$$

to obtain $x_0 = (c - \rho\lambda\sqrt{k})/\sqrt{k}$.

4 NUMERICAL PERFORMANCE

Recall that the proposed algorithm has two phases: (i) a preprocessing step where Equation (7) is solved to identify the parameter λ^* to prescribed tolerance; (ii) the identified parameter is then used appropriately to generate the required Poisson random variables. In this section, we report a portion of our ongoing tests to assess the performance in (i) and (ii).

In testing the preprocessing step, we systematically varied the three problem parameters λ, k, and ρ. Specifically, to generate Figure 3, we varied λ in the range $(0, 100)$ in steps of 0.1, k in the range $(0, 1)$ in steps of 0.1, and ρ in the range $(-1, 0)$ in steps of 0.01. The plots show the 99th percentile of the CPU time in seconds, and the number of steps in recursion (8), as a function of k. The stipulated tolerance for recursion (8) was 10^{-4}, and the tests were performed through a MATLAB compiler on an Intel 1.67GHz processor.

As can be seen from Figure 3, an overwhelming majority of the generated problems are solved to stipulated tolerance within 4.1×10^{-4} CPU seconds, and take 4 iterations or less. The maximum CPU time, and the maximum number of iterations, not depicted in Figure 3, were uniformly less than 10^{-2} seconds, and 5 respectively. In addition to those reported in Figure 3, our ongoing tests on a few million problems, generated with λ values up to 1000, have revealed no problems where the preprocessing time took more than 0.02 seconds, and greater than 8 Newton iterations.

Figure 4 depicts generation times for a sample problem with $\lambda = 100$, $\rho = -0.75$ and across different k values. By generation times, we mean the time taken to execute Steps 2 through 7 in the algorithm listing shown in Section 2.1. Steps 3, 4, and 5 were executed using the Poisson cdf inversion technique described in (Kemp and Kemp 1991), but without incorporating the "squeeze" technique.

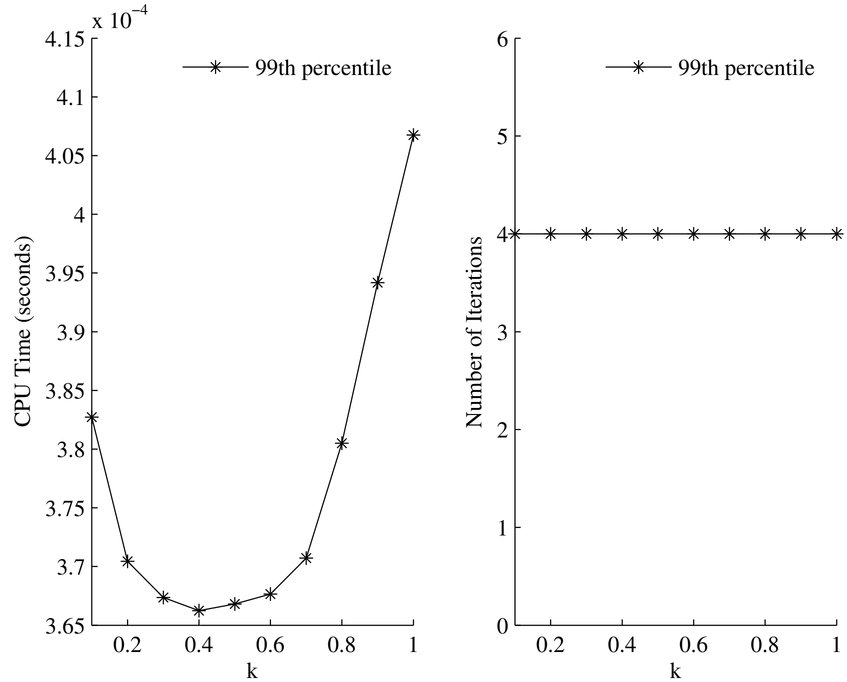

Figure 3: Performance of the proposed numerical procedure for executing the preprocessing step. The reported CPU times were obtained from execution through a MATLAB compiler on an Intel 1.67GHz processor.

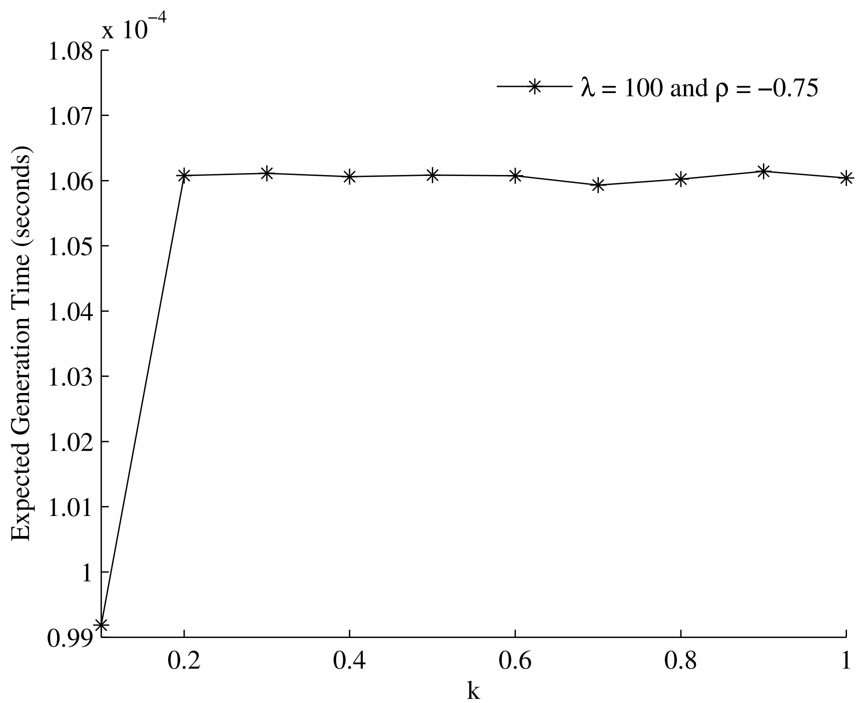

Figure 4: Estimated expected generation times of the proposed algorithm on a sample problem. The reported times were obtained from execution through a MATLAB compiler on an Intel 1.67GHz processor.

5 SUMMARY AND ONGOING RESEARCH

In this paper, we present a specialized method for generating correlated Poisson random vectors. The method, like trivariate reduction, uses extra random variables to induce the required correlation. Specifically, the extra random variables use common random numbers to induce positive correlation, antithetic variates to induce negative correlation, and have appropriately-scaled means. This choice of means is accomplished through a preprocessing step, involving a fast one-dimensional recursive search.

Unlike trivariate reduction, and like NORTA, the proposed method has complete coverage in two dimensions. Furthermore, preliminary tests involving Poisson marginal distributions with means in the interval $(0, 100)$, and correlations in the interval $(-1, 0)$, suggest that, in an overwhelming majority of the cases, the preprocessing step takes of the order of 10^{-4} seconds, when executed through MATLAB on an Intel 1.67GHz processor. The actual generation of the Poisson random vectors is executed through well-established Poisson cdf inversion techniques.

In ongoing research, we are investigating four important issues.

(i) How does the method perform in higher dimensions? We are specifically interested in understanding how the coverage area deteriorates as the dimension increases. Given that the proposed method is specialized for Poisson random vectors, it is unsurprising that the preprocessing step is a few orders of magnitude faster than in traditional implementations of NORTA. What is more interesting, however, is how the proposed method and NORTA compare in terms of coverage area in high dimensions.

(ii) To aid understanding within a modeling context, we would like to know the properties of the regression lines inherent to the bivariate distribution induced by the proposed procedure, e.g., the functions $\mathrm{E}[X_1|X_2 = x_2]$, $\mathrm{Var}[X_1|X_2 = x_2]$.

(iii) Johnson, Kotz, and Balakrishnan (1997) state that "the negative binomial distribution has become increasingly popular as a more flexible alternative to the Poisson distribution." One reason is that the Poisson distribution is sometimes limiting due to its implicit restriction of the equality of the mean and variance.

A useful characterization of the negative binomial distribution is as a "Poisson distribution whose parameter λ is gamma distributed." This close relationship between the Poisson and the negative binomial distributions leads to interesting questions about whether specialized methods can be developed for fast generation of correlated negative binomial random vectors, especially by exploiting the developed methods.

(iv) We would like to further understand numerical performance of the proposed algorithm through more extensive testing. Specifically, we would like to generate random problems within a much larger range of λ values.

ACKNOWLEDGMENTS

We wish to thank Professor Bruce Schmeiser, whose original ideas formed the basis of this research.

REFERENCES

Cario, M. C., B. L. Nelson, S. Roberts, and J. R. Wilson. 2002. Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical report, Department of Industrial Engineering and Management Sciences, Northwestern University.

Devroye, L. 1986. *Non-uniform random variate generation*. New York, NY.: Springer.

Ghosh, S., and S. G. Henderson. 2003. Behavior of the NORTA method for correlated random vector generation as the dimension increases. *ACM TOMACS* 13 (3): 276–294.

Johnson, N. L., S. Kotz, and N. Balakrishnan. 1997. *Discrete multivariate distributions*. New York, NY.: John Wiley & Sons, Inc.

Kemp, C. D., and A. W. Kemp. 1991. Poisson random variate generation. *Applied Statistics* 40 (1): 143–158.

Schmeiser, B. W., and V. Kachitvichyanukul. 1981. Poisson random variate generation. Technical report, Purdue University, School of Industrial Engineering.

Whitt, W. 1976. Bivariate distributions with given marginals. *The Annals of Statistics* 4 (6): 1280–1289.

AUTHOR BIOGRAPHIES

KAEYOUNG SHIN is doctoral student in the Industrial and Systems Engineering Department at Virginia Tech. His research interests include Monte Carlo methods. He is a member of INFORMS and IIE.

RAGHU PASUPATHY is an assistant professor in the Industrial and Systems Engineering Department at Virginia Tech. His research interests include Monte Carlo methods and simulation optimization. He is the current associate newsletter editor for the INFORMS Simulation Society.