

Simulating Multivariate Nonhomogeneous Poisson Processes Using Projections

EVAN A. SALTZMAN

RAND Corporation

and

JOHN H. DREW and LAWRENCE M. LEEMIS

The College of William & Mary

and

SHANE G. HENDERSON

Cornell University

Established techniques for generating an instance of a multivariate nonhomogeneous Poisson process (NHPP) such as thinning can become highly inefficient as the dimensionality of the process is increased, particularly if the defining intensity (or rate) function has a pronounced peak. To overcome this inefficiency, we propose an alternative approach where one first generates a projection of the NHPP onto a lower-dimensional space, and then extends the generated points to points in the original space by generating from appropriate conditional distributions. One version of this algorithm replaces a high-dimensional problem with a series of one-dimensional problems. Several examples are presented.

Categories and Subject Descriptors: I.6.8 [**Simulation and Modeling**]: Types of Simulation—*Discrete Event*; G.3 [**Probability and Statistics**]: Stochastic Processes; Random Number Generation

General Terms: Algorithms, Theory, Reliability

Additional Key Words and Phrases: nonhomogeneous Poisson process, random process generation, random variate generation, input modeling, simulation

Author's addresses: E. Saltzman, RAND Corporation, 1200 South Hayes Street, Arlington, VA 22202-5050.

J. Drew and L. Leemis, Department of Mathematics, The College of William & Mary, P.O. Box 8795, Williamsburg, VA 23187-8795.

S. Henderson, School of Operations Research and Information Engineering, Cornell University, Ithaca NY 14850

Henderson's work was partially supported by National Science Foundation grants CMMI-0800688 and CMMI-0926814. He would also like to acknowledge the hospitality and support of the Mathematical Sciences Institute at Australia National University where he was visiting when he completed much of his work on this paper.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 20YY ACM 0000-0000/20YY/0000-0001 \$5.00

1. INTRODUCTION

A multivariate (or multidimensional) nonhomogeneous Poisson process is a useful model for systems in which points occur over multiple dimensions at a varying rate. For example, the multivariate NHPP can be used to model how the rate of warranty claims for an item varies over both the item's age and time in use [Murthy et al. 1995]. Other application areas of the multivariate NHPP include queuing theory [Disney and Konig 1985; Jennings et al. 1996], health care [De Angelis and Gilks 1994; Lindsey 1996; Henderson and Mason 2004], labor economics [Bissinger 1981], risk management [Cooil 1991], and reliability [Agustin and Pena 1999; Arjas et al. 1991].

Following Resnick [1992] we can define a multivariate NHPP as a random point process as follows. (Resnick gives a more general definition but the following will suffice for our purposes.) Let $(X(i) : i \geq 0)$ be a countable collection of random points in the d -dimensional rectangle $V = (a_1, b_1] \times (a_2, b_2] \times \cdots \times (a_d, b_d]$, and let $N(A)$ be the (random) number of points that fall in the set $A \subseteq V$. Then N is said to be a point process on V . (Technically, we should restrict attention to measurable subsets A but we suppress measurability requirements in what follows.)

Definition 1.1. An NHPP with intensity function $\lambda(\cdot) \geq 0$ on V with $\int_V \lambda(x) dx < \infty$ is a point process N on V that has the following properties.

- (1) For all $A \subseteq V$, $N(A)$ is Poisson distributed with mean $\mu(A)$, where $\mu(A) = \int_A \lambda(x) dx$.
- (2) If A_1, A_2, \dots, A_k are disjoint subsets of V , then the random variables $N(A_1), N(A_2), \dots, N(A_k)$ are independent.

Hence, the number of points in a region of V is Poisson distributed, and the number of points in disjoint regions are independent of one another.

Resnick's definition allows the NHPP to be defined on unbounded domains, but we restrict attention to the bounded region V . The intensity function $\lambda(\cdot)$ determines the intensity with which points occur in V , with more points tending to arise in areas where $\lambda(\cdot)$ is large. Associated with the intensity function λ is the *mean measure* μ that gives the expected number of points in a set.

We consider the problem of generating a realization of the NHPP. Several methods have previously been proposed for this problem. One such method is an extension of the "order-statistic property" of Poisson processes in one dimension. This method first generates a Poisson random variable M with mean $\mu(V) = \int_V \lambda(x) dx$, and then generates M independent and identically distributed (i.i.d.) points in V with density $\lambda(\cdot)/\mu(V)$; see, e.g., p. 341 of Resnick [1992]. To implement this method one needs $\mu(V)$ in closed form and the ability to generate from the density $\lambda(\cdot)/\mu(V)$. Another algorithm is an extension of the widely-used thinning technique [Lewis and Shedler 1979]. Thinning requires determining a majorizing function $\lambda^*(x)$ such that $\lambda^*(x) \geq \lambda(x)$ for all $x \in V$. Typically, simulation practitioners assume a constant majorizing function such that $\lambda^* \geq \sup_{x \in V} \lambda(x)$. Each point $x \in V$ of the homogeneous Poisson process (HPP) N^* with constant rate λ^* is retained, independent of all other points, with probability $\lambda(x)/\lambda^*$ or "thinned" with probability $1 - \lambda(x)/\lambda^*$ to form the NHPP N with intensity function $\lambda(\cdot)$. Although this form of thinning algorithm can be applied for any bounded inten-

sity function, it is particularly inefficient if $\lambda^* \gg \lambda(x)$ over much of V , e.g., the intensity function has a high peak. This inefficiency becomes more pronounced as the dimensionality of the multivariate NHPP is increased because the rejection probability typically increases. A different algorithm for bivariate *homogeneous* Poisson process generation, proposed by Ross [2003], employs a “fanning out” procedure. Ross’s algorithm for homogeneous Poisson processes is closely related to the one we present here for *nonhomogeneous* Poisson processes, and one could view the present algorithm as an extension of that in Ross [2003]. Other studies by Merzbach and Nualart [1986], Nair [1990], and Schoenberg [1999] have considered using time rescaling methods for process generation of more general multivariate point processes.

As an alternative strategy, we first generate a realization of an NHPP that arises as the projection of the points of the target NHPP onto a lower-dimensional space. We use the projection that maps a point in \mathbb{R}^d into the p -dimensional point consisting of the first $p < d$ coordinates of the point. (For clarity we state and prove results only for $p = 1$, but the proofs go through with only notational changes for $p > 1$.) Proposition 2.1 proves that the projected points are those of an NHPP which we subsequently call the *marginal* NHPP. Then we extend the p -dimensional points of the marginal NHPP back to d dimensions by generating the remaining coordinates from an appropriate conditional distribution given the first p coordinates. If $p = 1$ then the marginal NHPP is one-dimensional so any standard method for generating one-dimensional NHPPs can be employed, including inversion, thinning, composition or other techniques. Likewise, any convenient method can be employed to generate the remaining coordinates of the points. Determining what projection to use, i.e., which coordinates to generate in the marginal NHPP, and which methods to use to generate the remaining coordinates, depends on the functional form of the intensity function. Our approach can, in many cases, reduce the dimensionality of the process-generation problem, and therefore offers a potentially more efficient method for simulation of multivariate NHPPs in those cases than more standard techniques like thinning.

The primary contribution of this paper is the algorithm defined in the next section and the proof that the algorithm indeed generates an NHPP with the desired rate function, i.e., that the algorithm is correct. A secondary contribution of the paper is to demonstrate the benefits of using point-process theory in analyzing Poisson processes, as in Resnick [1992]. Indeed, this makes it straightforward to state and prove the key results.

The remainder of this paper is organized as follows. Section 2 presents the algorithm for our process generation approach, proves that it is correct, and provides some specializations of the algorithm. Section 3 provides some examples. Implications and extensions are discussed in Section 4.

2. THE ALGORITHM

Suppose that N is an NHPP on V as defined in Section 1, with points $\{X(1), X(2), \dots, X(M)\}$, where M is Poisson distributed with mean $\mu(V) = \int_V \lambda(x) dx$. Our algorithm is based on two observations. First, roughly speaking, the set $\{X_1(1), X_1(2), \dots, X_1(M)\}$ of first coordinates of the points is an NHPP

on the interval $(a_1, b_1]$. Second, given the initial coordinate of a point, the remaining coordinates of the point can be obtained by sampling from an appropriate conditional distribution. We first define a few necessary quantities, then state the algorithm, and finally prove that it is correct, i.e., establish that the stochastic process that is generated by the algorithm is an NHPP with the desired rate function.

For $z \in (a_1, b_1]$, let $V(z)$ be the slice through the hyperrectangle V where all points on the slice have first coordinate z , i.e., $V(z) = \{(z, y) = (z, y_1, y_2, \dots, y_{d-1}) : a_{i+1} < y_i \leq b_{i+1}, i = 1, 2, \dots, d-1\}$. Then $V(z)$ is a $(d-1)$ -dimensional hyperrectangle for each z . Define $m(z)$ to be the $((d-1)$ -dimensional) integral of the intensity function $\lambda(\cdot)$ over $V(z)$, i.e.,

$$m(z) = \int_{y \in (a_2, b_2] \times \dots \times (a_d, b_d]} \lambda(z, y) dy.$$

Finally, given $z \in (a_1, b_1]$, define the conditional density $f(\cdot; z)$ of the remaining $d-1$ coordinates given that the first coordinate of a point is z as $f(y; z) = \lambda(z, y_1, y_2, \dots, y_{d-1})/m(z)$ for $y \in (a_2, b_2] \times (a_3, b_3] \times \dots \times (a_d, b_d]$. In the points of an NHPP with rate function λ , first coordinates z for which $m(z) = 0$ arise with probability 0, but to be precise in the case where $m(z) = 0$, define $f(\cdot; z)$ arbitrarily as that of a uniform distribution on $(a_2, b_2] \times (a_3, b_3] \times \dots \times (a_d, b_d]$. Our algorithm is then as follows.

- (1) Generate a realization of an NHPP on the interval $(a_1, b_1]$ with intensity function $(m(z) : a_1 < z \leq b_1)$, yielding the points $Z(1), Z(2), \dots, Z(M) \in (a_1, b_1]$, where M is Poisson distributed with mean $\int_{(a_1, b_1]} m(x) dx = \int_V \lambda(x) dx = \mu(V)$. These points give the first coordinates of all points in V .
- (2) For each i ($i = 1, 2, \dots, M$), generate the remaining coordinates $(Y_1(i), Y_2(i), \dots, Y_{d-1}(i))$ according to the density $f(\cdot; Z(i))$ conditional on $Z(i)$, and set the i th point $X(i)$ equal to $(Z(i), Y_1(i), Y_2(i), \dots, Y_{d-1}(i))$.

N.B. In Step 2 above we generate the remaining candidates according to the (multivariate) conditional density $f(\cdot; Z(i))$. This can be achieved through a variety of algorithms depending on the form of f . For example, one might use acceptance-rejection to generate the remaining $(d-1)$ components simultaneously, or one might first generate $Y_1(i)$ conditional on $Z(i)$, then $Y_2(i)$ conditional on $(Z(i), Y_1(i))$ and so forth until all remaining components are generated.

To prove that this algorithm is correct, we state and prove two propositions that rely on standard results in Poisson-process theory. See Chapter 4 of Resnick [1992] for an excellent introduction to this area.

Our first result proves that Step 1 of the algorithm is correct, in the sense that the set of one-dimensional points obtained by projecting the points of the target NHPP onto their first coordinate are those of an NHPP with the rate function $(m(z) : z \in (a_1, b_1])$. Hence we can generate these coordinates via Step 1 of the algorithm, and then generate the remaining coordinates from their conditional distribution.

PROPOSITION 2.1. *Let $\{X_1(1), X_1(2), \dots, X_1(M)\}$ be the first coordinates of the points of the NHPP defined in Section 1 above. Then these are the points of an NHPP defined on the interval $(a_1, b_1]$ with rate function $(m(z) : z \in (a_1, b_1])$.*

PROOF. Let $\{X(1), X(2), \dots, X(M)\}$ be the points of the NHPP defined in Section 1 above. Let us call this NHPP the *target* NHPP. Define the mapping $g(x) = g(x_1, x_2, \dots, x_d) = x_1$ that returns the first coordinate of the point x . Then, from Proposition 4.3.1 of Resnick [1992], the first coordinates

$$\{X_1(i), i = 1, 2, \dots, M\} = \{g(X(1)), g(X(2)), \dots, g(X(M))\}$$

are the points of a new NHPP on $(a_1, b_1]$ where the expected number of points in $A \subseteq (a_1, b_1]$, given by $\mu_1(A)$, is the expected number of points $\mu(\tilde{A})$ of the target NHPP in

$$\tilde{A} = A \times (a_2, b_2] \times (a_3, b_3] \times \dots \times (a_d, b_d],$$

since $\mu_1 = \mu \circ g^{-1}$. But $\mu(\tilde{A})$ is given by

$$\int_{\tilde{A}} \lambda(x) dx = \int_A \left[\int_{(a_2, b_2] \times \dots \times (a_d, b_d]} \lambda(x_1, x_2, \dots, x_d) dx_2 \dots dx_d \right] dx_1 = \int_A m(x_1) dx_1,$$

where we apply Tonelli's theorem. Hence we have established that the intensity of the transformed points is $(m(z) : a_1 < z \leq b_1)$ as required. \square

We now complete the proof of correctness with the following result. The key to this result is Proposition 4.10.1 of Resnick [1992] which, roughly speaking, states that if one attaches conditionally independent “marks” (extra dimensions) to the points of an NHPP, then one obtains a new NHPP on an enlarged space.

PROPOSITION 2.2. *The points generated in our algorithm are those of an NHPP with intensity function $\lambda(\cdot)$ in V .*

PROOF. Let $X(1), X(2), \dots, X(M)$ be the points generated by the algorithm. Step 1 of the algorithm constructs the first coordinates of these points to be those of an NHPP defined on the interval $(a_1, b_1]$ with rate function $(m(z) : z \in (a_1, b_1])$. Step 2 of the algorithm “marks” each first-coordinate z with a $(d-1)$ -dimensional random vector with density $f(\cdot; z)$. Hence, from Part (b) of Proposition 4.10.1 of Resnick [1992], the resulting set of points are those of an NHPP on V with intensity ν on V , where

$$\begin{aligned} \nu(x) &= \nu(x_1, x_2, \dots, x_d) \\ &= m(x_1) f(x_2, x_3, \dots, x_d; x_1). \end{aligned}$$

If $m(x_1) > 0$, then

$$\begin{aligned} m(x_1) f(x_2, x_3, \dots, x_d; x_1) &= m(x_1) \frac{\lambda(x_1, x_2, \dots, x_d)}{m(x_1)} \\ &= \lambda(x), \end{aligned}$$

while if $m(x_1) = 0$, then $\nu(x) = 0$. Thus $\nu(x) = \lambda(x)$ except potentially on the set $\{x \in V : m(x_1) = 0\}$. The intensities ν and λ therefore agree except on a set of μ measure 0, and this completes the proof. \square

In the algorithm as presented above we generate the first coordinates of each point via a one-dimensional NHPP generator, and then fill in the remaining coordinates from the appropriate conditional distribution. One could instead generate a subset

of the coordinates of the points (not necessarily just a single coordinate, and not necessarily including the first coordinate) as an NHPP, and fill in the remaining coordinates of the generated points by sampling from the appropriate conditional distribution. Of course, one would need to be able to sample from the appropriate NHPP and conditional distributions.

3. ALGORITHMS AND EXAMPLES

In this section, we provide a more explicit implementation of our projection algorithm using inversion, several illustrative examples, and an extension based on composition.

3.1 Inversion Algorithm

We provide a specialized version of our projection algorithm for generating bivariate NHPPs based on inversion. We specialize to bivariate NHPPs for clarity, and because of their ubiquity in applications. We use inversion to generate both the points of the marginal NHPP and the second coordinates of the points conditional on the first. For $z \in (a_1, b_1]$, define

$$\Lambda_1(z) = \int_{(a_1, z]} m(v) dv = \int_{(a_1, z]} \int_{(a_2, b_2]} \lambda(v, y) dy dv$$

to be the cumulative intensity function of the marginal NHPP. Also, define

$$F(y; z) = \int_{(a_2, y]} f(w; z) dw = \frac{\int_{(a_2, y]} \lambda(z, w) dw}{\int_{(a_2, b_2]} \lambda(z, w) dw} \quad (1)$$

to be the conditional cumulative distribution function of the second coordinate conditional on the first. If the integral in the denominator of (1) is 0, then we arbitrarily define $F(y; z)$ to be that of a uniform distribution on $(a_2, b_2]$.

Inversion Algorithm

```

s1 ← 0                                # initialize variables
k ← 0
repeat {
  u1 ← Random(0, 1)                    # generate first coordinate
  s1 ← s1 + ln(u1)
  if (s1 ≤ Λ1(b1)) {
    k ← k + 1
    X1(k) ← Λ1-1(s1)
    u2 ← Random(0, 1)                    # generate second coordinate
    X2(k) ← F-1(u2; X1(k))
  }
} until (s1 > Λ1(b1))
return X(1), X(2), . . . , X(k)

```

This inversion algorithm can be extended to generate higher-dimensional NHPPs. One simply replaces the generation of the second coordinate from its conditional

distribution given the first with the successive generation of the j th coordinate given the first $j - 1$ coordinates, for $j = 2, 3, \dots, d$. This assumes, of course, that inversion can be used with these conditional distributions, which is not always the case.

Using an implementation of the inversion algorithm in S-Plus, we provide two examples of generating random points of a bivariate NHPP in Figure 1. Figure 1(a) shows a contour plot of the intensity function $\lambda(x_1, x_2) = 6x_1^2x_2$ of a bivariate NHPP defined on $(0, 2] \times (0, 2]$, with darker shading corresponding to higher intensity. Applying the inversion algorithm, we generate a realization of random points and plot them in Figure 1(b). Note how points occur more often as the values of x_1 and x_2 increase, as predicted in Figure 1(a). Moreover, the theoretical expected number of points in the region $(0, 2] \times (0, 2]$, given by $\mu(V) = 32$, is close to the 35 points generated in this realization. A series of Monte Carlo simulations yields results that are consistent with the expected number of points being 32. For one such simulation of $n = 10000$ realizations, the sample mean $\hat{m} = 31.98$ and the sample variance $s_M^2 = 31.84$. Note that $\hat{m} \approx s_M^2$ as expected since the number of points m generated in a realization is Poisson distributed (and hence the theoretical mean and variance are equal). The standard error of the mean is given by $s_M/\sqrt{n} = 0.0564$.

A second example of a bivariate NHPP with intensity function $\lambda(x_1, x_2) = 10x_2/(x_1 + 1)$ is shown in Figures 1(c) and 1(d). From Figure 1(c), we observe that the intensity decreases with x_1 , but increases with x_2 . As shown in Figure 1(d), the realization generated by inversion reflects this theoretical prediction. Furthermore, the expected number of points in the region $(0, 2] \times (0, 2]$, given by $\mu(V) = 20 \ln 3 \approx 22$, is close to the 21 points generated in this realization. Once again, a battery of Monte Carlo simulations yields results that are consistent with the theoretical mean for the number of points in the region $(0, 2] \times (0, 2]$. For one such simulation of $n = 10000$ realizations, the sample mean $\hat{m} = 21.97$ and the sample variance $s_M^2 = 22.00$. Once again $\hat{m} \approx s_M^2$, as expected. The standard error of the mean is given by $s_M/\sqrt{n} = 0.0469$.

If a constant majorizing function $\lambda^*(x_1, x_2) = \lambda(2, 2) = 48$ associated with a bivariate thinning algorithm were used for the first bivariate NHPP, the probability of accepting a particular pair of observations generated is given by

$$\Pr[\text{accept}] = \frac{\mu(V)}{4 \cdot 48} = \frac{32}{192} = \frac{1}{6}.$$

Discarding five out of every six pairs generated might be unacceptable in terms of execution time. This inefficiency is exacerbated in the case of a higher peak or higher dimensions.

Now we compare the required computation time of the Inversion Algorithm and bivariate thinning using a constant majorizing function, i.e., $\lambda^* = \sup\{\lambda(x_1, x_2) : (x_1, x_2) \in V\}$, to generate 10,000 realizations in the two-dimensional space $U = (1, 3] \times (1, 3]$ for a family of intensity functions. In particular, we consider a bivariate extension of the Weibull distribution proposed by Hougaard [1986]. The Weibull distribution is widely used in applications of Poisson processes in reliability engineering [Leemis 1995]. Following Hougaard [1986], let the survivor function of the

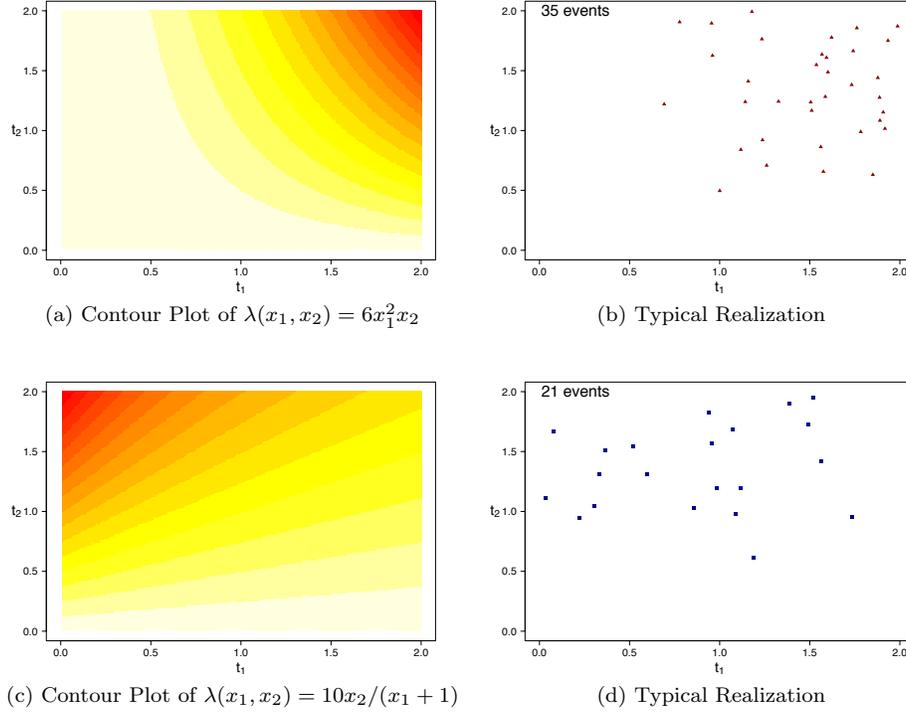


Fig. 1. Generating a Realization of a Bivariate NHPP Using Inversion

bivariate Weibull distribution be given by

$$S(x_1, x_2) = \exp \left\{ - \left[(x_1/\theta_1)^{\beta_1} + (x_2/\theta_2)^{\beta_2} \right] \right\},$$

where $\theta_1, \theta_2, \beta_1, \beta_2, x_1, x_2 > 0$. For simplicity, we will assume that $\theta_1 = \theta_2 = 1$ and consider only variations in the parameters β_1 and β_2 . Given these restrictions on the parameter space, the associated intensity function for $x_1, x_2 > 0$ is given by (note that the power law process uses the parametric form of the hazard function for the Weibull distribution for a nonrepairable item in one dimension with the intensity function for a repairable item)

$$\lambda(x_1, x_2) = \beta_1 \beta_2 x_1^{\beta_1 - 1} x_2^{\beta_2 - 1}.$$

Observe that if $\beta_1 = \beta_2 = 1$, the bivariate Weibull (power law) intensity function is constant and hence defines a bivariate HPP with constant intensity $\lambda = \lambda(x_1, x_2) = 1$. The bivariate Weibull intensity function is increasing (decreasing) in x_i if $\beta_i > 0$ ($\beta_i < 0$) for $i = 2$. It is important to note that for $\beta_i < 0$, the bivariate Weibull intensity function is not defined for $x_i = 0, i = 1, 2$. Hence, we restrict our attention to the two-dimensional space U .

In Table I, we summarize the computation time required to generate 10,000 realizations for the two algorithms for several specifications of the bivariate Weibull

Table I. Comparing the Computation Time of the Inversion Algorithm and Bivariate Thinning (10,000 Realizations in the Space $U = (1, 3] \times (1, 3]$)

(β_1, β_2)	$E[X^U]$	$\Pr[A]$	Inv. (sec.)	Thinning (sec.)	Reduction
(3, 3)	676	0.23	461.03	1871.92	75.4%
(3, 2)	208	0.32	138.73	345.53	59.9%
(2, 2)	64	0.44	42.08	75.11	44.0%
(3, 1)	52	0.48	34.01	55.2	38.4%
(2, 0.25)	2.53	0.42	2.49	3.67	32.2%
(1.5, 1.5)	17.61	0.65	12.25	14.46	15.3%
(2, 1)	16	0.67	10.80	12.70	15.0%
(1.5, 0.5)	3.07	0.59	2.78	3.15	11.7%
(0.75, 0.75)	1.64	0.73	1.63	1.81	9.9%
(0.5, 0.5)	0.54	0.54	1.14	1.20	5.0%
(1, 1)	4	1	2.82	2.67	-5.6%
(0.5, 0.25)	0.23	0.46	0.91	0.83	-9.6%
(0.25, 0.25)	0.10	0.40	0.80	0.63	-27.0%

intensity function. The relative performance of the algorithms is influenced by two key factors: the expected number of points generated in U and the peakedness of the intensity function. In general, the expected number of points generated in U , denoted $E[X^U]$, increases with both β_1 and β_2 , as shown in the second column of Table I. The main loop of the Inversion Algorithm will execute, on average, $E[X^U]$ times. By contrast, the main loop of the bivariate thinning algorithm will execute, on average, $E[X^U]/\Pr[A]$ times, where A is the event that a point generated in the bivariate HPP with rate λ^* is accepted. Since $\Pr[A] \leq 1$, the main loop of the thinning algorithm must execute at least as many times as the main loop of the Inversion Algorithm. As opposed to the expected number of points generated in U , the peakedness of the intensity function decreases as β_1 and β_2 approach 1 and then increases as β_1 and β_2 exceed 1. To give a practical measure of the peakedness of the intensity function, we display $\Pr[A]$ in the third column of Table I.

As Table I indicates, the performance of the Inversion Algorithm is far superior if the number of points generated in U is large and the intensity function is highly peaked. For $\beta_1 = \beta_2 = 3$, the Inversion Algorithm achieves a computation time reduction of over 75%. As β_1 and β_2 are decreased towards 1, the advantage of the Inversion Algorithm is gradually reduced. Once β_1 and β_2 begin to decrease below 1, the advantage of the Inversion Algorithm begins to widen again as $\Pr[A]$ decreases. However, as β_1 and β_2 get closer to 0, thinning becomes slightly more efficient than the Inversion Algorithm. Each loop of the Inversion Algorithm requires a few more computations than bivariate thinning. Hence, thinning may be faster if $E[X^U]$ is very small, as is the case for the last two (β_1, β_2) -pairs in Table I.

We end this subsection with generating a trivariate NHPP, which extends the bivariate notation developed thus far. Such a situation might arise in the following settings. The points in the process are landfalls of hurricanes of category 3 or higher. The first dimension corresponds to the time of landfall (hence a_1 and b_1 denote the beginning and ending times of hurricane season). The second dimension corresponds to the position on the coastline where landfall occurs (stretched in a linear fashion from the furthest possible landfall position south a_2 to the furthest possible landfall position north b_2). The third dimension is the maximum windspeed

at landfall (hence a_3 is the lowest possible windspeed for a category 3 hurricane and b_3 is ∞). As a second example, x_1 might be the time of a 911 emergency call, and x_2 and x_3 might be the longitude and latitude of the location of the call. An example illustrating the generation of a realization of a trivariate process follows.

Consider the intensity function with $d = 3$ given by

$$\lambda(x_1, x_2, x_3) = x_1 + 2x_2^2 + 3x_3^3$$

defined on the rectangular region $(0, 1] \times (0, 2] \times (0, 3]$. The cumulative intensity function is

$$\begin{aligned} \Lambda(x_1, x_2, x_3) &= \int_0^{x_1} \int_0^{x_2} \int_0^{x_3} (t_1 + 2t_2^2 + 3t_3^3) dt_3 dt_2 dt_1 \\ &= \frac{1}{2}x_1^2 x_2 x_3 + \frac{2}{3}x_1 x_2^3 x_3 + \frac{3}{4}x_1 x_2 x_3^4, \end{aligned}$$

so the expected number of points is $\mu(V) = \Lambda(1, 2, 3) = 281/2$. The cumulative intensity function of the marginal NHPP of the first coordinate is

$$\Lambda_1(z) = \int_0^z \int_0^2 \int_0^3 (t_1 + 2t_2^2 + 3t_3^3) dt_3 dt_2 dt_1 = 3z^2 + \frac{275}{2}z$$

for $0 < z < 1$. The inverse cumulative intensity function of the marginal NHPP is

$$\Lambda_1^{-1}(s_1) = \frac{-275 + \sqrt{75625 + 48s_1}}{12}$$

for $0 < s_1 < 281/2$. This function will be used to generate the first coordinate of the process. In order to generate the second and third coordinates of the process, it is necessary to find the associated conditional distributions. The conditional cumulative distribution function of the second coordinate conditioned on the first coordinate is

$$\begin{aligned} F_{2|1}(y; z) &= \frac{\int_{(a_3, b_3]} \int_{(a_2, y]} \lambda(z, t_2, t_3) dt_2 dt_3}{\int_{(a_3, b_3]} \int_{(a_2, b_2]} \lambda(z, t_2, t_3) dt_2 dt_3} \\ &= \frac{\int_0^3 \int_0^y \lambda(z, t_2, t_3) dt_2 dt_3}{\int_0^3 \int_0^2 \lambda(z, t_2, t_3) dt_2 dt_3} \\ &= \frac{3yz + 2y^3 + 243y/4}{6z + 275/2} \end{aligned}$$

for $0 < y < 2$, which has inverse

$$F_{2|1}^{-1}(u_2; z) = \frac{c^{1/3}}{4} - \frac{2z + 81/2}{c^{1/3}},$$

where $c = 2200u_2 + 96u_2z + 2 \left[\left(\sum_{j=0}^3 a_{1j} z^j \right) + u_2^2 \left(\sum_{j=0}^2 a_{2j} z^j \right) \right]^{1/2}$ for $0 < u_2 < 1$, and the a_{1j} and a_{2j} values are given in Table II. Finally, the conditional cumulative distribution function of the third coordinate conditioned on the values of the first

Table II. Constant Values

j	a_{1j}	a_{2j}
0	1062882	1210000
1	157464	105600
2	7776	2304
3	128	•

and second coordinates is

$$\begin{aligned}
F_{3|1,2}(x; z, y) &= \frac{\int_{(a_3, x]} \lambda(z, y, t_3) dt_3}{\int_{(a_3, b_3]} \lambda(z, y, t_3) dt_3} \\
&= \frac{\int_0^x \lambda(z, y, t_3) dt_3}{\int_0^3 \lambda(z, y, t_3) dt_3} \\
&= \frac{x(4z + 8y^2 + 3x^3)}{3(4z + 8y^2 + 81)}
\end{aligned}$$

for $0 < x < 3$. When this function is equated to u_3 in order to find the conditional inverse cumulative distribution function, the resulting equation is the quartic

$$3x^4 + (4z + 8y^2)x - 3u_3(4z + 8y^2 + 81) = 0.$$

This can be solved exactly using Ferrari's method resulting in

$$\begin{aligned}
F_{3|1,2}^{-1}(u_3; z, y) &= \frac{\sqrt{2}}{2} \sqrt{\frac{4z + 8y^2}{18c_1^{1/3} - 6u_3(4z + 8y^2 + 81)/c_1^{1/3}} - y} \\
&\quad - \frac{1}{6} \sqrt{18c_1^{1/3} - 6u_3(4z + 8y^2 + 81)/c_1^{1/3}}
\end{aligned}$$

for $0 < u_3 < 1$, where

$$\begin{aligned}
c_1 &= (4z + 8y^2)^2 / 144 + (z^4 + 8z^3y^2 + 24z^2y^4 + 32zy^6 + 16y^8 + 192u_3^3z^3 + \\
&\quad 1152u_3^3z^2y^2 + 11664u_3^3z^2 + 2304u_3^3zy^4 + 46656u_3^3zy^2 + 236196u_3^3z + \\
&\quad 1536u_3^3y^6 + 46656u_3^3y^4 + 472392u_3^3y^2 + 1594323u_3^3)^{1/2}.
\end{aligned}$$

The speed of evaluating these expressions should be weighed against numerically inverting the conditional cumulative distribution function. Using these functions, the inversion algorithm is easily modified as shown below.

Inversion Algorithm

```

 $s_1 \leftarrow 0$                                      # initialize variables
 $k \leftarrow 0$ 
repeat {
   $u_1 \leftarrow \text{Random}(0, 1)$                  # generate first coordinate
   $s_1 \leftarrow s_1 - \ln(u_1)$ 
  if ( $s_1 \leq \Lambda_1(b_1)$ ) {
     $k \leftarrow k + 1$ 
     $X_1(k) \leftarrow \Lambda_1^{-1}(s_1)$ 
     $u_2 \leftarrow \text{Random}(0, 1)$                  # generate second coordinate
  }
}

```

```

     $X_2(k) \leftarrow F_{2|1}^{-1}(u_2; X_1(k))$ 
     $u_3 \leftarrow \text{Random}(0, 1)$  # generate third coordinate
     $X_3(k) \leftarrow F_{3|1,2}^{-1}(u_3; X_1(k), X_2(k))$ 
  }
} until( $s_1 > \Lambda_1(b_1)$ )
return  $X(1), X(2), \dots, X(k)$ 

```

If the cumulative marginal intensity function and the conditional cumulative distribution functions do not have closed-form inverses, then using the inversion algorithm may be problematic. For these cases, composition or one-dimensional thinning could be used to generate the marginal NHPP while composition or the acceptance-rejection method could be applied to generate the other coordinate. Another alternative is to use composition at the level of the multivariate NHPP, an approach that we discuss next.

3.2 Composition Algorithm

In some instances, composition may offer a solution to the “invertibility problem” by employing a “divide and conquer” strategy. Sometimes an intensity function $\lambda(\cdot)$ can be written as the sum of intensity functions $\sum_{j=1}^k \lambda_j(\cdot)$. For instance, in reliability applications, there are often competing independent risks that contribute to the occurrence of part failures or system failures [Leemis 1995]. Instead of applying inversion to an NHPP N with intensity function $\lambda(\cdot)$, the composition algorithm generates independent realizations for the k individual NHPPs N_j with intensity function $\lambda_j(\cdot)$ and then superimposes them to form a realization for N . (A formal definition of *independent* NHPPs can be obtained via the theory in Section 4.7 of Resnick [1992], but would represent something of a distraction. Suffice it to say that independent NHPPs are those that arise when generated from independent streams of random numbers.) The (multivariate) composition algorithm is analogous to the univariate composition algorithm described in Devroye [1986] and is simply to independently generate points from each NHPP in turn and then return the union of the points generated. Superposition for multivariate NHPPs is possible due to the following well-known result; see, e.g., Exercise 4.44 on p. 363 of Resnick [1992]. We provide a proof for clarity, and to further demonstrate the elegance of the point-process definition of a Poisson process that we have adopted.

THEOREM 3.1. *Let N_1, N_2, \dots, N_k be independent multivariate NHPPs with intensity functions $\lambda_1(\cdot), \lambda_2(\cdot), \dots, \lambda_k(\cdot)$ respectively. Then their superposition $N = \sum_{i=1}^k N_i$ is a multivariate NHPP with intensity function $\sum_{i=1}^k \lambda_i(\cdot)$.*

PROOF. For any set A , $N(A) = \sum_{i=1}^k N_i(A)$ is Poisson distributed with mean $\mu(A) = \sum_{i=1}^d \mu_i(A)$, where $\mu_i(A) = \int_A \lambda_i(x) dx$, because the sum of independent Poisson random variables is also Poisson distributed. The second property (independence in disjoint regions) is immediate. \square

Figure 2 demonstrates the use of the composition algorithm for the bivariate NHPP N defined on $[0, 2] \times [0, 2]$ with intensity function $\lambda(x_1, x_2) = 6x_1^2x_2 + 10x_2/(x_1 + 1)$. The appropriate cumulative intensity and distribution functions do

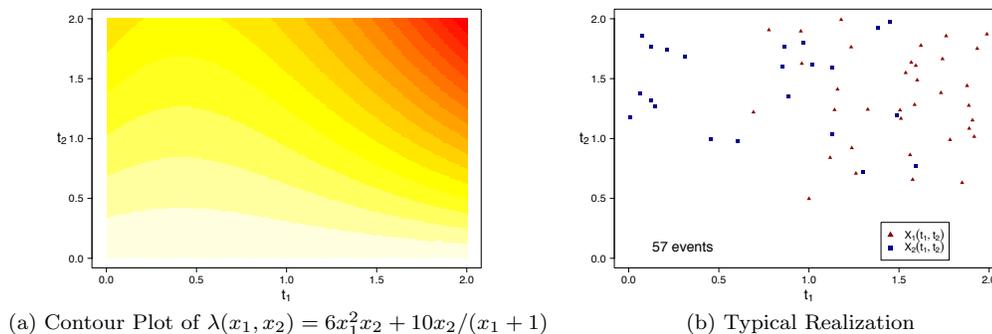


Fig. 2. Generating a Realization of a Bivariate NHPP Using Composition

not have closed-form inverses, so inversion is ill-suited for this problem. Random points can be generated in a straightforward fashion, however, using the composition algorithm by associating N_1 with intensity function $\lambda_1(x_1, x_2) = 6x_1^2x_2$ and N_2 with intensity function $\lambda_2(x_1, x_2) = 10x_2/(x_1 + 1)$. A realization of the composition algorithm is shown in Figure 2(b). The theoretical expected number of points, given by $\mu(V) = 32 + 20 \ln 3 \approx 54$, is close to the 57 points generated in the region $(0, 2] \times (0, 2]$. In addition, the clustering of points in the realization accords closely with the intensity function contour plot in Figure 2(a).

As our example has demonstrated, the composition algorithm applies the inversion algorithm to $m = 2$ (in this case) individual bivariate NHPPs N_1, N_2 that are superimposed to generate a realization of the bivariate NHPP $N_1 + N_2$. Consequently, the “invertibility problem” associated with the inversion algorithm can be overcome in this example in a straightforward manner.

4. DISCUSSION

This paper has presented a potentially efficient approach for generating realizations of a multivariate NHPP. The multivariate extension of the thinning algorithm for generating a realization of points of a multivariate NHPP can become highly inefficient as the dimensionality of the problem is increased, particularly if the intensity function is highly peaked. To overcome this difficulty, we transform the complex multivariate process generation problem to a sequence of smaller and relatively simple one-dimensional process generation problems. First, one generates the first coordinate of each point of the multivariate NHPP using the marginal NHPP of that coordinate. The techniques of inversion, composition, or one-dimensional thinning can be employed. By conditioning on the first coordinate generated, one can then simultaneously or successively generate the remaining coordinates using their conditional distributions. Inversion, composition, or acceptance-rejection techniques may all be used in this process. We also discussed the possibility of applying composition at the level of the multivariate NHPP.

We restricted attention to a hyperrectangle, but one can generalize this approach to regions with other shapes, either by enclosing other shapes in a hyperrectangle,

defining the intensity to be 0 outside the region of interest and employing the algorithm herein, or by breaking down complex regions into simpler regions and employing composition.

We restricted attention to projections along the coordinate axes, but mappings other than projections could also be used. For example, if the intensity function has elliptical contours, then one can first generate a one-dimensional NHPP, the points of which represent the contours that contain points, and then conditional on those contours generate a single point on each contour that is uniformly distributed along the contour. In this sense, the algorithm outlined herein is really a large family of algorithms that can be tailored to particular problem instances.

REFERENCES

- AGUSTIN, M. Z. N. AND PENA, E. A. 1999. Order statistic properties, random generation, and goodness-of-fit testing for a minimal repair model. *Journal of the American Statistical Association* 94, 445, 266–272.
- APOSTOL, T. M. 1974. *Mathematical Analysis*, Second ed. Addison-Wesley, Reading, MA.
- ARJAS, E., HANSEN, C. K., AND THYREGOD, P. 1991. Heterogeneous part quality as a source of reliability improvement in repairable systems. *Technometrics* 33, 1, 1–12.
- BISSINGER, B. H. 1981. A long life analysis. *The Statistician* 30, 1, 15–21.
- ÇINLAR, E. 1975. *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, NJ.
- COOIL, B. 1991. Using medical malpractice data to predict the frequency of claims: A study of Poisson process models with random effects. *Journal of the American Statistical Association* 86, 414, 285–295.
- DE ANGELIS, D. AND GILKS, W. R. 1994. Estimating acquired immune deficiency syndrome incidence accounting for reporting delay. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 157, 1, 31–40.
- DEVROYE, L. 1986. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, NY.
- DISNEY, R. L. AND KONIG, D. 1985. Queuing networks: A survey of their random processes. *SIAM Review* 27, 3, 335–403.
- HENDERSON, S. G. AND MASON, A. J. 2004. Ambulance service planning: simulation and data visualization. In *Operations Research and Health Care: A Handbook of Methods and Applications*, M. L. Brandeau, F. Sainfort, and W. P. Pierskalla, Eds. Kluwer Academic, Boston, MA, 77–102.
- HOUGAARD, P. 1986. A class of multivariate failure time distributions. *Biometrika* 73, 3, 671–678.
- JENNINGS, O. B., MANDELBAUM, A., MASSEY, W. A., AND WHITT, W. 1996. Server staffing to meet time-varying demand. *Management Science* 42, 10, 1383–1394.
- LEEMIS, L. 1995. *Reliability: Probabilistic Models and Statistical Methods*, 1st ed. Prentice Hall, Englewood Cliffs, NJ.
- LEWIS, P. A. W. AND SHEDLER, G. S. 1979. Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly* 3, 26, 403–414.
- LINDSEY, J. K. 1996. Fitting bivariate intensity functions, with an application to modelling delays in reporting immune deficiency syndrome. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 159, 1, 125–131.
- MERZBACH, E. AND NUALART, D. 1986. A characterization of the spatial Poisson process and changing time. *The Annals of Probability* 14, 4, 1380–1390.
- MURTHY, D. N. P., ISKANDAR, B. P., AND WILSON, R. 1995. Two-dimensional failure-free warranty policies: Two-dimensional point process models. *Operations Research* 43, 2, 356–366.
- NAIR, M. G. 1990. Random space change for multiparameter point processes. *The Annals of Probability* 18, 3, 1222–1231.
- RESNICK, S. I. 1992. *Adventures in Stochastic Processes*. Birkhäuser, Boston, MA.
- ROSS, S. M. 2003. *Introduction to Probability Models*, 8th ed. Academic Press, New York, NY.
- ACM Journal Name, Vol. V, No. N, MM 20YY.

SCHOENBERG, F. 1999. Transforming spatial point processes into Poisson processes. *Stochastic Processes and Their Applications* 81, 155–164.

Received Month Year; revised Month Year; accepted Month Year