

# An Efficient Randomized Quasi-Monte Carlo Algorithm for the Pareto Distribution

by

M. L. Huang<sup>1</sup>, M. Pollanen<sup>2\*</sup> and W. K. Yuen<sup>3\*\*</sup>

<sup>1</sup>Dept. of Mathematics, Brock University, St. Catharines, Ontario, Canada L2S 3A1

E-mail: mhuang@brocku.ca

<sup>2</sup>Dept. of Mathematics, Trent University, Peterborough, Ontario, Canada K9J 7B8

E-mail: marcopollanen@trentu.ca

<sup>3</sup>Dept. of Mathematics, Brock University, St. Catharines, Ontario, Canada L2S 3A1

E-mail: jyuen@brocku.ca

Revised: March 9, 2007

## Abstract

---

**Key Words and Phrases:** Discrepancy, Efficiency, Goodness-of-fit test, Mean square error, Quasi-Monte Carlo methods, Random number generation.

\*Corresponding Author

\*\*Research of Authors are Supported by an NSERC Canada Grant.

This paper studies a new randomized quasi-Monte Carlo method for estimating the mean and variance of the Pareto distribution. In many Monte Carlo simulations, there are some stability problems for estimating the population Pareto variance by using the sample variance. In this paper, we propose a *randomized quasi-random number generator [quasi-RNG]* to generate Pareto random samples, such that the sample mean and sample variance estimators gain more efficiency. The efficiency of this generator relative to the popular *linear congruential random number generator [LC-RNG]* is studied by using the simulation mean square errors. We also compare the results of the Kolmogorov-Smirnov goodness-of-fit tests using these two sample generators.

## 1 Introduction

The Pareto distribution is a power-tailed distribution with many applications in economics, actuarial science, survival analysis, queuing networks and other stochastic models (see e.g. Aban, Meerschaert and Panorska, 2006; Brown, Gans, Baum, Sakov, Shen, Zeltyn and Zhao, 2005; Fowler, 1999). It is important to explore estimation methods for this distribution.

Several types of Pareto distributions have been defined (Kleiber and Kotz, 2003).

In this paper, we only discuss the Pareto Type I distribution with density function

$$f(x) = \frac{\alpha}{x^{(\alpha+1)}}, \quad x \geq 1, \quad \alpha > 0 \tag{1.1}$$

and cumulative distribution function (*CDF*.)

$$F(x) = 1 - \frac{1}{x^\alpha}, \quad x \geq 1, \quad \alpha > 0. \quad (1.2)$$

The mean and variance are given by

$$\mu = \frac{\alpha}{\alpha - 1}, \quad \alpha > 1; \quad (1.3)$$

$$\sigma^2 = \frac{\alpha}{(\alpha - 1)^2(\alpha - 2)}, \quad \alpha > 2. \quad (1.4)$$

One way to compare the quality of the samples obtained from different RNGs is to consider the accuracies of various estimators based on the generated samples. In particular, it is important to obtain good estimates for the known mean and variance in (1.3) and (1.4) from a random sample  $X_1, X_2, \dots, X_n$ ,  $n \geq 3$ . In this paper, we consider the popular moment estimators, i.e. the sample mean and sample variance defined as follows

$$\hat{\mu} = \bar{X} = \frac{1}{n} \sum_{i=1}^n X_i; \quad (1.5)$$

$$\hat{\sigma}^2 = S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2. \quad (1.6)$$

(1.5) and (1.6) are based on the empirical distribution function (*EDF*)  $S_n(x)$  which is a minimum variance unbiased estimator for  $F(x)$ , where

$$S_n(x) = \frac{1}{n} \sum_{i=1}^n I_{(-\infty, x]}(X_i), \quad (1.7)$$

with

$$I_A(x) = \begin{cases} 1, & \text{if } x \in A; \\ 0 & \text{if } x \notin A. \end{cases}$$

an indicator function for the set  $A$ .

Note that when using the sample variance  $S^2$  in (1.6) to estimate the population variance, the variance of  $S^2$  is infinite when  $\alpha \leq 4$  (see Lemma 1.1). As a result, the estimated values of variance are not stable when the value of  $\alpha$  is slightly greater than 4. In addition, in many Monte Carlo simulations, for example, using the popular *LC-RNG* from MAPLE 10, the estimated values tend to be much smaller than the true variance since the generators are not generating large Pareto tail values often. To illustrate this problem, we simulate a sample of size 10 million for the Pareto distribution with  $\alpha = 2.05$  using the inverse distribution function method and two *LC-RNGs*. The sample variance for the first  $k$  sample points is calculated whenever  $k$  is a factor of 100000 to see how the estimator improves as  $k$  increases. The graph of sample variance vs  $k$  is shown in Appendix III. Observe that for both the default *LC-RNGs* in Maple 10 and the GNU implementation of C++, the Pareto sample underestimated the true variance 37.19 substantially (neither of them ever reached 15 once). In fact, in the test simulations we ran, these samples mostly underestimate the true variance substantially and in some very rare instances, the variance is substantially overestimated. Several authors studied different methods for this instability problem. For example, the truncated Pareto distribution has been considered but the results are not significant (Gross, Shortle, Fisher and Masi, 2002).

In this paper, we propose a new *randomized quasi-RNG* by modifying the well-

known van der Corput sequence. We show that the new *quasi-RNG* (without randomization) has good theoretical properties, such as  $d$ -variate uniformity and low discrepancy. In our simulation studies, we obtain randomized quasi samples for the Pareto distribution by transforming the randomized quasi-random numbers using the inverse distribution function method. Our results show that samples obtained with this new method are more efficient than those from *LC-RNGs* when using the sample mean  $\bar{X}$  and sample variance  $S^2$  to estimate the population mean and variance.

In Section 1.1 we provide expected values and variances of the sample mean  $\bar{X}$  and sample variance  $S^2$ . The new randomized *quasi-RNG* and its properties are introduced in Section 2. Section 3 discusses the simulation efficiencies of the samples using the new generator relative to the case of using the *LC-RNG* by considering the estimations of the population mean and variance. In Section 4 we perform Kolmogorov-Smirnov tests by using new and existing methods.

### 1.1 Properties of the Sample Variance $S^2$

In this section, we provide the expected values and variances of the sample mean  $\bar{X}$  in (1.5) and the sample variance  $S^2$  in (1.6) for estimating the population mean in (1.3) and variance in (1.4).

**Lemma 1.1.** *For a random variable  $X$  with density function (1.1), the expected values and the variances of the sample mean  $\bar{X}$  in (1.5) and the sample variance  $S^2$*

in (1.6) are given by

$$E(\bar{X}) = \mu = \frac{\alpha}{\alpha - 1}, \quad \alpha > 1; \quad (1.8)$$

$$\text{Var}(\bar{X}) = \frac{\alpha}{n(\alpha - 1)^2(\alpha - 2)}, \quad \alpha > 2; \quad (1.9)$$

$$E(S^2) = \sigma^2 = \frac{\alpha}{(\alpha - 1)^2(\alpha - 2)}, \quad \alpha > 2; \quad (1.10)$$

$$\text{Var}(S^2) = \frac{2\alpha [6(1 - n) + 6(3 - n)\alpha - (4n + 3)\alpha^2 + (4n - 3)\alpha^3]}{n(n - 1)(\alpha - 1)^4(\alpha - 2)^2(\alpha - 3)(\alpha - 4)}, \quad \alpha > 4. \quad (1.11)$$

**Proof.** (1.8)-(1.10) are straightforward and (1.11) is derived by Lemma 2.3, Huang, Brill and Gross, 2005. ■

## 2 A Quasi-RNG

In this section, we propose a new *quasi-RNG*. We also will study the d-variate uniformity and discrepancy of this generator.

While traditional quasi-RNGs aim to create sequences which minimize some notion of discrepancy, our new quasi-RNG has the dual aim of creating sequences with low-discrepancy and also good variance statistics.

### 2.1 Low-Discrepancy Sequences

**Definition 2.1** Let  $S = \{x_n\}$  be an infinite sequence in  $[0, 1]^d$ . The discrepancies  $D_N(S)$  and  $D_N^*(S)$  of the first  $N$  terms of  $S$  are defined by

$$D_N(S) = D_N(x_1, \dots, x_N) = \sup_J \left| \frac{A_N(J, S)}{N} - \lambda(J) \right| \quad (2.1)$$

and

$$D_N^*(S) = D_N^*(\mathbf{x}_1, \dots, \mathbf{x}_N) = \sup_{J^*} \left| \frac{A_N(J^*, S)}{N} - \lambda(J^*) \right| \quad (2.2)$$

where  $J$  runs through all subintervals of  $[0, 1]^d$  of the form

$$J = \{(y_1, \dots, y_d) \in [0, 1]^d : \alpha_i \leq y_i \leq \beta_i, \text{ for } 1 \leq i \leq d\},$$

and  $J^*$  runs through all subintervals of  $[0, 1]^d$  of the form

$$J^* = \{(y_1, \dots, y_d) \in [0, 1]^d : 0 \leq y_i \leq \beta_i, \text{ for } 1 \leq i \leq d\},$$

where  $0 \leq \alpha_i < \beta_i \leq 1$ . Here  $\lambda$  denotes the  $k$ -dimensional Lebesgue measure and

$$A_N(J, S) = \sum_{n=1}^N I_J(\mathbf{x}_n),$$

counts the number of  $\mathbf{x}_1, \dots, \mathbf{x}_N$  in  $J$ .

$D_N^*(S)$  is referred to as the *star discrepancy* while  $D_N(S)$  is the *extreme discrepancy*. Sequences with lowest known discrepancy have

$$D_N^*(S) = O\left(\frac{\log^d N}{N}\right).$$

It is also known that

$$D_N^*(S) \geq C \frac{\log^{\frac{d-1}{2}} N}{N}, \text{ for some constant } C.$$

In the one-dimensional case, it is known (Niederreiter, 1992) that

$$D_N^*(S) \geq 0.06 \frac{\log N}{N}$$

for any possible sequence.

Thus, it would be reasonable to define low-discrepancy as follows

**Definition 2.2** *A sequence  $S = \{\mathbf{x}_n\}$  in  $[0, 1]^d$  is called a low-discrepancy sequence if*

$$D_N^*(S) = O\left(\frac{\log^d N}{N}\right). \quad (2.3)$$

Note: by the law of the iterated logarithm, truly random sequences have a discrepancy that almost surely satisfies

$$D_N^*(S) = O\left(\sqrt{\frac{\log \log N}{N}}\right). \quad (2.4)$$

It is obvious that  $D_N^*(S) \leq D_N(S)$ , but actually, either measure could be used to define low-discrepancy sequences as the following relationship holds (Niederreiter, 1992):

**Lemma 2.1** *For any sequence  $\{\mathbf{x}_n\}$  in  $[0, 1]^d$ , we have*

$$D_N^*(\{\mathbf{x}_n\}) \leq D_N(\{\mathbf{x}_n\}) \leq 2^d D_N^*(\{\mathbf{x}_n\}). \quad (2.5)$$

We will use this result in the proof of Lemma 2.5, in which we give asymptotic bounds on how many points we may modify in low-discrepancy sequences for them to remain low-discrepancy. The above lemma also shows that Lemma 2.5 will hold if we replace star discrepancy with extreme discrepancy.

As the generation of random sequences from most well-known distributions are based on the transformation of uniformly distributed sequences, it is also important

to compare low-discrepancy sequences with uniformly distributed sequences:

**Definition 2.3** *An infinite sequence  $\{\mathbf{x}_n\}$  is uniformly distributed in  $[0, 1]^d$  if*

$$\lim_{N \rightarrow \infty} \frac{A_N(J, \{\mathbf{x}_n\})}{N} = \lambda(J). \quad (2.6)$$

*for every subinterval  $J$  of  $[0, 1]^d$ .*

The following is a well-known result:

**Proposition 2.2** (Kuipers and Niederreiter, 1974) *A sequence  $\{\mathbf{x}_n\}$  in  $[0, 1]^d$  is uniformly distributed iff*

$$\lim_{N \rightarrow \infty} D_N^* (\{\mathbf{x}_n\}) = 0. \quad (2.7)$$

Therefore the modified quasi-random sequences we define in Definition 2.5, will also inherit the property of uniformity.

## 2.2 Halton Sequences

If  $n$  is any positive integer, then there is a unique representation of  $n$  in base  $b \geq 2$  such that

$$n = \sum_{i=0}^{\infty} a_i(n) b^i$$

where  $a_i(n) \in \mathbb{Z}_b$  for all  $i \geq 0$  and  $a_i(n) = 0$  for sufficiently large  $i$ .

**Definition 2.4** (Niederreiter, 1992) *For any integer  $b \geq 2$ , the inverse-radix function  $\phi_b$  in base  $b$  is defined by*

$$\phi_b(n) = \sum_{i=0}^{\infty} a_i(n) b^{-i-1}, \text{ for all integers } n \geq 0. \quad (2.8)$$

For any integer  $b \geq 2$ , we define the van der Corput sequence as  $S_b = \{\phi_b(i)\}_{i \geq 0}$ .

It has been established that (see Niederreiter, 1992)

$$D_N^*(S_b) = O\left(\frac{\log N}{N}\right), \text{ for all } N \geq 2.$$

In fact

$$\limsup_{N \rightarrow \infty} \frac{N D_N^*(S_b)}{\log N} = \begin{cases} \frac{b^2}{4(b+1) \log b}, & \text{for even } b, \\ \frac{b-1}{4 \log b}, & \text{for odd } b. \end{cases}$$

Thus,  $S_3$  is asymptotically the best van der Corput sequence.

The van der Corput sequence has an obvious extension to a  $d$ -dimensional sequence. If  $b_1, \dots, b_d$  are all integers greater than 2, then we may define the *Halton sequence*  $\{\mathbf{x}_n\}$  as:  $\mathbf{x}_n = (\phi_{b_1}(n), \dots, \phi_{b_d}(n)) \in [0, 1]^d$ , for all  $n \geq 0$ .

**Theorem 2.3** (Niederreiter, 1992) *If  $S_{b_1, \dots, b_d}$  is the Halton sequence in the pairwise relatively prime bases  $b_1, \dots, b_d$ , then*

$$D_N^*(S_{b_1, \dots, b_d}) < \frac{d}{N} + \frac{1}{N} \left( \frac{b_i - 1}{2 \log b_i} \log N + \frac{b_i + 1}{2} \right)$$

for all  $N \geq 1$ .

Thus, the Halton sequence  $S_{b_1, \dots, b_d}$  in  $d$ -dimensions with pairwise relatively prime bases  $b_i$  has discrepancy

$$D_N^*(S_{b_1, \dots, b_d}) = O\left(\frac{\log^d N}{N}\right). \tag{2.9}$$

In the next subsection we provide a criterion for modifying a multidimensional low-discrepancy sequences such that the resulting sequence has the same asymptotic

discrepancy. We also examine the special one-dimensional case of the van der Corput sequence, for which we provide simulation results in transforming these to the Pareto distribution in Section 4.

### 2.3 Modified van der Corput sequence $\{y_n^{-s}\}$

So far we have been working only with the uniform distribution. Samples from general distributions with known distribution function can be generated by transforming samples from the uniform distribution using the inverse distribution function method. Tailed distributions like the Pareto distribution in (1.1) present a few challenges. First of all, when we “transform” the uniform sequence we must make sure that there are no points too close to edge of the cube. This is not a problem for the Halton sequence as:

**Lemma 2.4** *Given a Halton sequence  $\{\mathbf{x}_n\}$  in  $[0, 1]^d$  with relative prime bases  $b_1, \dots, b_d$ , the  $i$ -th coordinate satisfies*

$$\frac{1}{b_i n} \leq x_{ni} \leq 1 - \frac{1}{b_i n} \quad \text{for } n \geq 1. \quad (2.10)$$

**Proof.** Actually, we can prove this for a bit more general class of sequences. Each coordinate of  $\{\mathbf{x}_n\}$  forms its own van der Corput sequence, which is a generalized Niederreiter sequence (see Tezuka, 1995), from whose construction we have (for  $\mathbf{x}_n \neq$

0, which in the case of the Halton sequence is true for  $n > 0$ )

$$x_{ni} = \sum_{j=1}^m y_{nj}^{(i)} b^{-j}, \text{ for } 0 \leq n < b^m.$$

Therefore,  $x_{ni} \geq b^{-m}$ . As this construction holds for any  $m$ , we may assume that  $b^{m-1} \leq n < b^m$ , from which we see that  $b^{-(m-1)} \geq n^{-1}$  and so  $x_{ni} \geq b^{-m} \geq \frac{1}{bn}$ . Also

$$\begin{aligned} \sum_{j=1}^m y_{nj}^{(i)} b^{-j} &\leq \sum_{j=1}^m (b-1) b^{-j} \\ &= \left( \sum_{j=1}^m (b-1) b^{-j} + b^{-m} \right) - b^{-m} \\ &= 1 - b^{-m}. \end{aligned}$$

Now, since  $bn > b^m$ , we have  $\frac{1}{bn} < b^{-m}$  and so  $x_{ni} < 1 - \frac{1}{bn}$ . Thus the desired results follows. ■

While calculating moments, the tail of the Pareto distribution when transformed to the unit cube represents a singularity on the boundary of the cube. In many applications, it is observed that many existing RNGs do not generate points close enough to the boundary so that the Pareto sample points are not far enough into the tail. A major consequence is the underestimation of the true variance with the sample variance. With this in mind, we modify the Halton sequence to obtain another high-quality sequence (i.e. low-discrepancy) that after being transformed to a quasi-random sample from the Pareto distribution, will improve the estimation of the population mean and variance using the sample mean in (1.3) and sample variance in (1.4).

The following key result allows us to modify any low-discrepancy sequence by removing or adding points in a neighbourhood around a point and still have a sequence of  $O\left(\frac{\log^d N}{N}\right)$ :

**Lemma 2.5** *Let  $\{\mathbf{x}_n\}$  be any sequence in  $[0, 1]^d$  with discrepancy  $D_N^*(\{\mathbf{x}_n\}) = O\left(\frac{\log^d N}{N}\right)$  and let  $C_N$  be a sequence of nested subintervals of  $[0, 1]^d$  such that  $\lambda(C_N) = O\left(\frac{\log^d N}{N}\right)$ . If  $\{\mathbf{y}_n\}$  is any subsequence of  $\{\mathbf{x}_n\}$  such that  $\mathbf{x}_k \notin C_N$  for  $k \leq N$  implies  $\mathbf{x}_k \in \{\mathbf{y}_n\}$ , then  $D_N^*(\{\mathbf{y}_n\}) = O\left(\frac{\log^d N}{N}\right)$ . Likewise, let  $\{\mathbf{z}_n\}$  be any supersequence of  $\{\mathbf{x}_n\}$ , such that  $\mathbf{z}_k \notin C_N$  for  $k \leq N$  implies  $\mathbf{z}_k \in \{\mathbf{x}_n\}$ . If  $A_N(C_N, \{\mathbf{z}_n\}) = O(\log N)$ , then  $D_N^*(\{\mathbf{z}_n\}) = O\left(\frac{\log^d N}{N}\right)$ .*

**Proof.** Let  $\{C_N\}$  be a set of nested intervals whose measure converges to zero. Suppose  $\{\mathbf{x}_n\}$  is a sequence in  $[0, 1]^d$  with  $D_N^*(\{\mathbf{x}_n\}) = O\left(\frac{\log^d N}{N}\right)$ . By removing points from the sequence  $\{\mathbf{x}_n\}$  we can create a new sequence  $\{\mathbf{y}_n\}$  such that  $\mathbf{y}_n \notin C_N$  for  $n \leq N$ . Now for  $D_N^*(\{\mathbf{y}_n\}) = O\left(\frac{\log^d N}{N}\right)$  we require  $\lambda(C_N) = O\left(\frac{\log^d N}{N}\right)$ , otherwise

$$\lambda(C_N) = \left| \frac{A_N(C_N, \{\mathbf{y}_n\})}{N} - \lambda(C_N) \right| \leq D_N(\{\mathbf{y}_n\}) \leq 2^d D_N^*(\{\mathbf{y}_n\}),$$

and so we would have  $D_N^*(\{\mathbf{y}_n\}) \neq O\left(\frac{\log^d N}{N}\right)$ . Conversely, suppose that  $\{\mathbf{x}_n\}$  is a sequence in  $[0, 1]^d$  such that  $D_N(\{\mathbf{y}_n\}) \leq C \frac{\log^d N}{N}$  and assume that  $\lambda(C_N) = C' \frac{\log^d N}{N}$ .

Now the intervals  $C_N$  cannot contain more than  $(C + C') \log^d N$  points as

$$\begin{aligned} \left| \frac{A_N(C_N, \{\mathbf{x}_n\})}{N} - \lambda(C_N) \right| &= \left| \frac{A_N(C_N, \{\mathbf{x}_n\})}{N} - C' \frac{\log^d N}{N} \right| \\ &\leq D_N(\{\mathbf{x}_n\}) \leq C \frac{\log^d N}{N}. \end{aligned}$$

Let us create a new sequence  $\{y_n\}$  such that  $y_n \notin C_N$  for  $n < N$ . Suppose that we consider the first  $N$  terms of  $\{x_n\}$  and let  $N'$  represent the number of these terms that appear in  $\{y_n\}$ . We see that

$$N' \geq N - (C + C') \log^d N.$$

Now consider  $D_{N'}^*(\{y_n\})$ . For any interval  $[0, \mathbf{a}) \subset [0, 1]^d$ , since a  $d$ -hypercube has  $2d$  faces, to partition  $[0, \mathbf{a}) \setminus C_N$  into intervals we need at most  $2d$  intervals. Hence

$$\begin{aligned} D_{N'}^*(\{y_n\}) &\leq 2dD_N(\{x_n\}) + \lambda(C_N \cap [0, \mathbf{a})) \\ &= 2dD_N(\{x_n\}) + \frac{C' \log^d N}{N} \\ &\leq K \frac{\log^d N'}{N'}, \end{aligned}$$

for some constant  $K$  and for all  $N'$ , as  $\frac{\log^d N}{N}$  is a decreasing function (for  $N$  sufficiently large). ■

This last lemma allow us to take any low-discrepancy sequence and move around some points and maintain the same asymptotic order of discrepancy. In the case of the Pareto distribution, we will move these points closer to the interval boundary which maps into the tail. In the one-dimensional case, we consider the subsequence of the van der Corput sequence  $S_b$  with points of the form  $b^{-i}$  (note each term of this sub-sequence is smaller than any term preceding the corresponding term in the original sequence). The  $i$ -th point in this sequence is the  $b^i$ -th point in the van der Corput sequence, so by the above theorem we can move it anywhere we wish without

affecting the asymptotic order of discrepancy. In particular, we define the following low-discrepancy sequence:

**Definition 2.5.** *For any integer  $b \geq 2$ , we define the modified van der Corput sequence  $\{y_n^{\rightarrow s}\}_{n \geq 0}$  with shift  $s > 0$  by replacing the  $b^i$ -th point ( $b^{-(i+1)}$ ) in the van der Corput sequence  $S_b$ , by  $b^{-(i+1+s)}$  for some fixed  $s > 0$ .*

**Remark.** Note that the sequence  $\{y_n^{\rightarrow s}\}_{n \geq 0}$  depends on the value of the base  $b$ . However, since we will set  $b = 3$  in all our simulations, to avoid further complication of notation,  $b$  will not be shown in all of the definitions that follow.

### 3 Pareto Randomized Quasi-Random Sample $\{X_k^{\rightarrow s, p}\}$

With this new sequence  $\{y_n^{\rightarrow s}\}$ , we use the inverse distribution function method to define a quasi-random sample for the Pareto distribution as follows:

**Definition 3.1.** *For any integer  $b \geq 2$ , we define a quasi-random sample for the Pareto distribution  $X_1^{\rightarrow s}, X_2^{\rightarrow s}, \dots, X_n^{\rightarrow s}$ ,  $n \geq 3$ , with shift  $s > 0$  by the transformation:*

$$X_k^{\rightarrow s} = F^{-1}(1 - y_k^{\rightarrow s}).$$

A disadvantage of  $\{X_k^{\rightarrow s}\}$  is that it is a fixed sequence so that we will obtain the exact same quasi-random sample for our distribution in every simulation. To avoid this problem, we randomly select a subsequence of  $X_k^{\rightarrow s}$ . One method to do this is to accept each point of the sequence with a probability  $0 < p < 1$ . Formally, we have

**Definition 3.2.** For any integer  $b \geq 2$ , suppose we accept each points in the sequence  $\{X_k^{\rightarrow s}\}$  with shift  $s > 0$  independently with probability  $0 < p < 1$ . Let  $X_1^{\rightarrow s,p}, X_2^{\rightarrow s,p}, \dots, X_n^{\rightarrow s,p}$ ,  $n \geq 3$  be the first  $n$  accepted points of the sequence. We called this sample a randomized quasi-random sample for the Pareto distribution with shift  $s > 0$  and acceptance probability  $0 < p < 1$ .

Clearly, the discrepancy of this random sequence  $\{X_k^{\rightarrow s,p}\}$  is very close to the discrepancy of  $X_k^{\rightarrow s}$  if  $p$  is close to 1, although the resulting sequence is less “random”. The implementation of this method is straightforward: if want to obtain a sample of size  $n$ , we accept  $X_k^{\rightarrow s}$  with probability  $p$  sequentially until we have accepted  $n$  points of the sequence. We denote the resulting sequence by  $\{X_k^{\rightarrow s,p}\}$ . This is the random sequence we shall use in the comparison in the next section.

We simulated a sample of size 10 million and a graph of the sample variance of the first  $k$  sample points vs  $k$  is plotted in Appendix III for the *randomized quasi-RNGs* for  $s = 1, 2$  and  $p = 0.5$ . For the case  $s = 2$ , the sample variance actually reached the true variance 37.19 the first time after roughly 2 million sample points. For the case  $s = 1$ , the sample variance reached 25 once after 7 million sample points. On the other hand, the other two *LC-RNGs* never reached 15 once.

#### 4 Simulation Efficiencies of $\{X_k^{\rightarrow s,p}\}$

In this section, we study the quality of the Pareto random sample  $\{X_k^{\rightarrow s,p}\}$  with base  $b = 3$ , introduced in Section 2.4. We choose  $b = 3$  because  $S_3$  is asymptotically the best van der Corput sequence, as pointed out in Section 2.2. To simplify the notation, we use the notation  $\{X_k^{\rightarrow}\}$  instead of  $\{X_k^{\rightarrow s,p}\}$  and specify the values of  $s, p$  when necessary. This sample is compared to a popular Pareto pseudo-random sample  $X_1, X_2, \dots, X_n$ ,  $n \geq 3$ , generated similarly but using the *linear congruential RNG* instead (*LCG*), with the system clock as the seed, which is identical to the RNG in Maple 10. Note that we have also tried other popular pseudo RNGs such as the Mersenne Twister (Matsumoto and Nishimura, 1998) and built-in system C++ pseudo-RNGs, but the results are similar.

**Definition 4.1.** *A simulation sample variance by using the new randomized quasi-Monte Carlo RNG is denoted by*

$$\widehat{\sigma}_{new}^2 = S_{new}^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i^{\rightarrow} - \overline{X^{\rightarrow}})^2, \text{ where} \quad (4.1)$$

$$\widehat{\mu}_{new} = \overline{X^{\rightarrow}} = \frac{1}{n} \sum_{i=1}^n X_i^{\rightarrow}. \quad (4.2)$$

*A simulation sample variance by using the old linear congruential RNG is denoted by*

$$\widehat{\sigma}_{old}^2 = S_{old}^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \overline{X})^2, \text{ where} \quad (4.3)$$

$$\widehat{\mu}_{old} = \overline{X} = \frac{1}{n} \sum_{i=1}^n X_i. \quad (4.4)$$

We ran simulations on sample sizes of  $n = 50, 100, 1000, 10000, 100000$ . The computational efficiency results are given in Appendix I for selected values of  $\alpha = 2.05, 2.55, 3.05, 3.55, 4.05, 4.55$ . We generated  $m = 1,000$  random samples by using both the new quasi-RNG and old LCG for each  $n$  and  $\alpha$ , a total of 30 cases. In our simulations, we use  $p = 0.5$  for the new generator so that about half the points of  $\{X_k^{\rightarrow s}\}$  are accepted. We also tried various values of  $s$  and only the results for  $s = 1$  are shown in the Appendix I.

The  $m$  times simulation mean square errors are defined by

$$SMSE(\widehat{\mu}_{new}) = \frac{1}{m} \sum_{i=1}^m (\widehat{\mu}_{new,i} - \mu)^2,$$

$$SMSE(\widehat{\mu}_{old}) = \frac{1}{m} \sum_{i=1}^m (\widehat{\mu}_{old,i} - \mu)^2,$$

$$SMSE(\widehat{\sigma}_{new}^2) = \frac{1}{m} \sum_{i=1}^m (\widehat{\sigma}_{new,i}^2 - \sigma^2)^2,$$

$$SMSE(\widehat{\sigma}_{old}^2) = \frac{1}{m} \sum_{i=1}^m (\widehat{\sigma}_{old,i}^2 - \sigma^2)^2.$$

where  $\widehat{\mu}_{new,i}, \widehat{\sigma}_{new,i}^2, \widehat{\mu}_{old,i}, \widehat{\sigma}_{old,i}^2$  are the sample mean and variance values in the  $i$ -th generated random sample using the new (old) generator. Note that  $\widehat{\mu}_{old}, \widehat{\sigma}_{old}^2$  are unbiased estimators of the true mean  $\mu$  and variance  $\sigma^2$ , and  $SMSE(\widehat{\mu}_{old}), SMSE(\widehat{\sigma}_{old}^2)$  are unbiased estimators of the theoretical  $Var(\widehat{\mu}_{old}), Var(\widehat{\sigma}_{old}^2)$  respectively. We also define the simulation efficiencies of  $\widehat{\mu}_{new}$  relative to  $\widehat{\mu}_{old}$  and the

simulation efficiencies of  $\widehat{\sigma}_{new}^2$  relative to the  $\widehat{\sigma}_{old}^2$  as

$$SEFF(\widehat{\mu}_{new}) = \frac{SMSE(\widehat{\mu}_{old})}{SMSE(\widehat{\mu}_{new})}, \quad (4.5)$$

$$SEFF(\widehat{\sigma}_{new}^2) = \frac{SMSE(\widehat{\sigma}_{old}^2)}{SMSE(\widehat{\sigma}_{new}^2)}. \quad (4.6)$$

The simulation results are shown in Appendix I. We have the following interesting remarks:

- $SEFF(\widehat{\mu}_{new}) > 1$  in 25 out of 30 cases.  $SEFF(\widehat{\sigma}_{new}^2) > 1$  in 28 out of 30 cases.
- The fluctuation of the simulation efficiencies is due to the fluctuation of the simulation mean square error. In fact, the result is still unstable for larger value of  $m$  but it is important to note that  $SEFF(\widehat{\mu}_{new}) > 1$  and  $SEFF(\widehat{\sigma}_{new}^2) > 1$  still hold in most cases.
- We do not show the results for  $s = 0$  (i.e. the original van der Corput sequence is used). In that case, the  $\widehat{\mu}_{new}$  and  $\widehat{\sigma}_{new}^2$  are on average more efficient ( $>1$ ) relative to the classical estimators  $\widehat{\mu}_{old}$  and  $\widehat{\sigma}_{old}^2$  in all the cases. It is not surprising due to the regularity of the samples. However, the improvements for  $SEFF(\widehat{\sigma}_{new}^2)$  is not as good as the case of  $s = 2$ , which is our main focus.
- We also tried other values of  $s$ . Generally speaking, when  $\alpha$  is close to 2, a larger  $s$  results in a smaller  $SMSE(\widehat{\sigma}_{new}^2)$ . For other values of  $\alpha$ , a large

shift  $s$  affects the estimation in a negative way, as we “overmodified” the van der Corput sequence.

## 5 Kolmogorov-Smirnov Goodness-of-Fit Test

Finally, we perform the *Kolmogorov-Smirnov (K-S) goodness-of-fit test* (Conover, 1999) on the samples  $X_1^{\rightarrow}, X_2^{\rightarrow}, \dots, X_n^{\rightarrow}$  and  $X_1, X_2, \dots, X_n$ , to determine how they fit the true Pareto distribution. We define *K-S statistics* for the samples using the new and old generator as

**Definition 5.1.** *The empirical distribution functions (EDF)  $S_{n(new)}(x)$  and  $S_{n(old)}(x)$  by using the new randomized quasi-RNG and the old linear congruential RNG are defined as*

$$S_{n(new)}(x) = \frac{1}{n} \sum_{i=1}^n I_{(-\infty, x]}(X_i^{\rightarrow}), \quad x \geq 1; \quad (5.1)$$

$$S_{n(old)}(x) = \frac{1}{n} \sum_{i=1}^n I_{(-\infty, x]}(X_i), \quad x \geq 1. \quad (5.2)$$

**Definition 5.2.** *The K-S statistics  $T_{new}$  and  $T_{old}$  by using the new randomized quasi RNG and the linear congruential RNG are defined as*

$$T_{new} = \sup_{x \geq 1} |S_{n(new)}(x) - F(x)|; \quad (5.3)$$

$$T_{old} = \sup_{x \geq 1} |S_{n(old)}(x) - F(x)|; \quad (5.4)$$

where  $F(x)$  is the true CDF in (1.2).

**Definition 5.3.** The  $p$ -values of K-S statistics  $T_{new}$  and  $T_{old}$  are defined as

$$P_{new} = P \{T_{new} > T_{new(observable)}(X_1^{\rightarrow}, X_2^{\rightarrow}, \dots, X_n^{\rightarrow})\};$$

$$P_{old} = P \{T_{old} > T_{old(observable)}(X_1, X_2, \dots, X_n)\}.$$

Note that the  $p$ -values can be obtained by the following formula

$$P - value = 1 - \left[ 1 - t \sum_{j=0}^{[n(1-t)]} \binom{n}{j} \left(1 - t - \frac{j}{n}\right)^{n-j} \left(t + \frac{j}{n}\right)^{j-1} \right]^2, \quad (5.5)$$

where  $t = T_{obs}$  is the observed value of  $T$  from the random sample.  $[n(1-t)]$  is the greatest integer less than or equal to  $n(1-t)$ . When  $n > 1000$ , we use an asymptotic formula as

$$P - value \approx 1 - \left[ 1 - e^{-2nt^2} \right]^2. \quad (5.6)$$

The  $m$  times simulation  $p$ -values  $SP_{new}$  and  $SP_{old}$  are defined as

$$SP_{new} = \frac{1}{m} \sum_{i=1}^m P_{new,i}, \quad SP_{old} = \frac{1}{m} \sum_{i=1}^m P_{old,i}, \quad (5.7)$$

where  $P_{new,i}$  and  $P_{old,i}$  are respectively the  $p$ -values of K-S statistics of the  $i$ -th random sample using the new quasi-RNG and old LCG generator.

In this simulation, we use the samples generated for Table 1, i.e.,  $X_i^{\rightarrow} \equiv X_i^{\rightarrow 1,0.5}$ .

The values of  $T_{new}$  and  $T_{old}$  and their respective  $p$ -values  $SP_{new}$  and  $SP_{old}$  are showed in Appendix II, Table 2. Our results show that  $T_{new} < T_{old}$  and  $SP_{new}$  are larger than  $SP_{old}$  by probabilities of about 0.2 to 0.3 in all cases, which means that  $X_1^{\rightarrow 1,0.5}, X_2^{\rightarrow 1,0.5}, \dots, X_n^{\rightarrow 1,0.5}$  fit better to the true distribution than  $X_1, X_2, \dots, X_n$  on average.

The simulations were run by using C++ with double precision.

## 6 Conclusion

In this paper we propose a new randomized quasi-random number generator to generate Pareto random sequences to improve common stability problems, due to the heavy tail, in estimating the population Pareto variance by using the sample variance in Monte Carlo simulations. The generator uses randomized van der Corput sequences and is based on slightly modifying the location of the points that will be mapped, under transformation, into the tail of the Pareto distribution. We show by simulation that under this new generator the sample mean and sample variance estimators gain more efficiency and have improved goodness-of-fit in comparison to traditional random number generators.

While the method we propose uses the randomized van der Corput sequence to generate a 1-dimensional Pareto distribution, a similar method might be useful in generating other heavy-tailed distributions. Traditional quasi-Monte Carlo methods rely on minimizing one measure, i.e. the discrepancy, as the criterion for the quality of a sequence. In essence, with our method, we aim to maintain the asymptotic order of discrepancy for a low-discrepancy sequence, but slightly modify the sequence to also obtain better variance statistics. In Lemma 2.5 we derive bounds on how many points can be modified in a general multivariate low-discrepancy sequences to maintain the asymptotic order of discrepancy. This lemma may be useful for creating methods for generating other heavy-tailed distributions.

## References

- Aban, I. B. Meerschaert, M. M. and Panorska, A. K. (2006), "Parameter Estimation for the Truncated Pareto Distributions", *Journal of the American Statistical Association*, Vol. 101, No. 473, 270-277.
- Brown, L., Gans, N., Baum, A. M., Sakov, A., Shen, H., Zeltyn, S. and Zhao, L. (2005), "Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective", *Journal of the American Statistical Association*, Vol. 100, No. 469, 36-50.
- Conover, W. J. (1999), *Practical Nonparametric Statistics, third edition*, John Wiley & Sons, New York.
- Fowler, T. B. (1999), "A Short Tutorial on Fractals and Internet Traffic", *The Telecommunication Review*, (McLean, VA), 10: 1-14.
- Gentle, J. E. (2003), *Random Number Generation and Monte Carlo Methods*, Springer, New York.
- Gross, D., Shortle, J. F., Fisher, M. J. and Masi, D. M. B. (2002), "Difficulties in Simulating Queues with Pareto Service", *Proceedings of the 2002 Winter Simulation Conference*, E. Yiicesam, C. H. Chen, J. L. Snowdon and J. M. Chanes Eds. pp.407-415.

Huang, M. L, Brill, P. H. and Gross, D. (2005), “A Weighted Estimation Method for the Pareto Variance.”, *The Proceeding of American Statistical Association, Nonparametric Statistics Section*, pp.1649-1654, 2005.

Kleiber, C. K. and Kotz, S. (2003), *Statistical Size Distribution in Economics and Actuarial Sciences*. John Wiley & Sons, New York.

Kuipers, L. and Niederreiter, H. (1974), *Uniform Distribution of Sequences*. Wiley, New York.

Matsumoto, M. and Nishimura, T. (1998), “Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator”, *ACM Trans. on Modeling and Computer Simulation*, Vol. 8, No. 1, pp.3-30.

Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia.

Tezuka, S. (1995), *Uniform Random Numbers: Theory and Practice*. Kluwer, Boston.

**APPENDIX I Simulation Efficiencies of the Quasi-Monte Carlo Estimator**

**Table 1: The Simulation Efficiencies of  $\widehat{\mu}_{new}$  Relative to  $\widehat{\mu}_{old}$  and  $\widehat{\sigma}_{new}^2$  Relative to  $\widehat{\sigma}_{old}^2$**

Generated  $m = 1000$  times, Sample size  $n = 50, 100, 1000, 10000, 100000$

*Shift  $s = 1$ , Acceptance Probability  $p = 0.5$ , Parameter  $\alpha = 2.05, 2.55, 3.05, 3.55, 4.05, 4.55$*

$$\text{Simulation Efficiencies } SEFF\left(\widehat{\mu}_{new}\right) = \frac{SMSE\left(\widehat{\mu}_{old}\right)}{SMSE\left(\widehat{\mu}_{new}\right)} \text{ and } SEFF\left(\widehat{\sigma}_{new}^2\right) = \frac{SMSE\left(\widehat{\sigma}_{old}^2\right)}{SMSE\left(\widehat{\sigma}_{new}^2\right)}$$

<b>Sample Size <math>n</math></b>	<b>Parameter <math>\alpha</math></b>	<b>SMSE <math>\left(\widehat{\mu}_{new}\right)</math></b>	<b>SMSE <math>\left(\widehat{\mu}_{old}\right)</math></b>	<b>SEFF <math>\left(\widehat{\mu}_{new}\right)</math></b>	<b>SMSE <math>\left(\widehat{\sigma}_{new}^2\right)</math></b>	<b>SMSE <math>\left(\widehat{\sigma}_{old}^2\right)</math></b>	<b>SEFF <math>\left(\widehat{\sigma}_{new}^2\right)</math></b>
<b>50</b>	<b>2.05</b>	0.133492	0.197843	<b>1.482060</b>	862.9593	20247.11	<b>23.46242</b>
	<b>2.55</b>	0.039314	0.061696	<b>1.569323</b>	2.216711	1567.745	<b>707.2394</b>
	<b>3.05</b>	0.016634	0.011925	<b>0.716886</b>	0.411620	1.595868	<b>3.877042</b>
	<b>3.55</b>	0.009067	0.007732	<b>0.852782</b>	0.111726	0.733781	<b>6.567679</b>
	<b>4.05</b>	0.005360	0.004024	<b>0.750774</b>	0.037124	0.055912	<b>1.506097</b>
	<b>4.55</b>	0.003809	0.002944	<b>0.772944</b>	0.016272	0.016406	<b>1.008257</b>
<b>100</b>	<b>2.05</b>	0.030675	0.088381	<b>2.881226</b>	1034.158	5582.406	<b>5.398022</b>
	<b>2.55</b>	0.010304	0.014928	<b>1.448820</b>	0.615698	8.263912	<b>13.42201</b>
	<b>3.05</b>	0.004794	0.007439	<b>1.551865</b>	0.083555	1.291442	<b>15.45623</b>
	<b>3.55</b>	0.002442	0.003750	<b>1.535182</b>	0.021184	0.529879	<b>25.01300</b>
	<b>4.05</b>	0.001536	0.002249	<b>1.464886</b>	0.007423	0.044573	<b>6.004870</b>
	<b>4.55</b>	0.001010	0.001327	<b>1.313413</b>	0.003072	0.041097	<b>13.37775</b>
<b>1000</b>	<b>2.05</b>	0.003665	0.008972	<b>2.447898</b>	953.9539	7710.696	<b>8.082881</b>
	<b>2.55</b>	0.000934	0.001654	<b>1.770707</b>	0.304350	15.28723	<b>50.22917</b>
	<b>3.05</b>	0.000385	0.000680	<b>1.766091</b>	0.024366	0.404179	<b>16.58774</b>
	<b>3.55</b>	0.000189	0.000331	<b>1.745921</b>	0.004744	0.032159	<b>6.778266</b>
	<b>4.05</b>	0.000125	0.000246	<b>1.974044</b>	0.001416	0.007075	<b>4.994725</b>
	<b>4.55</b>	0.000087	0.000128	<b>1.462881</b>	0.000567	0.001332	<b>2.349912</b>
<b>10000</b>	<b>2.05</b>	0.001056	0.000814	<b>0.771096</b>	586.4469	957.8131	<b>1.633248</b>
	<b>2.55</b>	0.000154	0.000209	<b>1.361064</b>	0.642425	6.015434	<b>9.363641</b>
	<b>3.05</b>	0.000050	0.000072	<b>1.427454</b>	0.035432	0.074996	<b>2.116609</b>
	<b>3.55</b>	0.000024	0.000038	<b>1.570794</b>	0.004418	0.009024	<b>2.042363</b>
	<b>4.05</b>	0.000013	0.000023	<b>1.761931</b>	0.000790	0.000729	<b>0.922745</b>
	<b>4.55</b>	0.000008	0.000014	<b>1.695245</b>	0.000215	0.000163	<b>0.756867</b>
<b>100000</b>	<b>2.05</b>	0.000108	0.000143	<b>1.314563</b>	551.4111	9175.800	<b>16.64058</b>
	<b>2.55</b>	0.000014	0.000019	<b>1.354249</b>	0.203314	0.707471	<b>3.479690</b>
	<b>3.05</b>	0.000004	0.000007	<b>1.704456</b>	0.007049	0.021709	<b>3.079713</b>
	<b>3.55</b>	0.000002	0.000003	<b>1.704249</b>	0.000584	0.000684	<b>1.170044</b>
	<b>4.05</b>	0.000001	0.000002	<b>1.879106</b>	0.000089	0.000106	<b>1.184930</b>
	<b>4.55</b>	0.000001	0.000001	<b>1.773579</b>	0.000019	0.000023	<b>1.180135</b>

APPENDIX II Kolmogorov-Smirnov Goodness-of-Fit Test

Table 2: The K-S Test Statistics and Their P – Values

Generated  $m = 1000$  times, Sample size  $n = 50, 100, 1000, 10000, 100000$

Shift  $s = 1$ , Acceptance Probability  $p = 0.5$ , Parameter  $\alpha = 2.05, 2.55, 3.05, 3.55, 4.05, 4.55$

$$T_{new} = \sup_{x \geq 1} |S_{n(new)}(x) - F(x)|; \quad T_{old} = \sup_{x \geq 1} |S_{n(old)}(x) - F(x)|$$

Sample Size $n$	Parameter $\alpha$	$T_{new}$	$SP_{new}$	$T_{old}$	$SP_{old}$
50	2.05	0.090928	0.647672	0.119939	0.437881
	2.55	0.090771	0.648798	0.117791	0.454035
	3.05	0.091987	0.639900	0.119389	0.442505
	3.55	0.092840	0.632532	0.119458	0.440635
	4.05	0.090471	0.651481	0.119097	0.441897
	4.55	0.090828	0.649300	0.121729	0.421423
100	2.05	0.062274	0.676659	0.085325	0.440504
	2.55	0.062979	0.670237	0.086358	0.430445
	3.05	0.062945	0.669484	0.086305	0.428739
	3.55	0.063994	0.657886	0.084357	0.450080
	4.05	0.062798	0.671370	0.084422	0.448276
	4.55	0.062208	0.677077	0.085279	0.437621
1000	2.05	0.019644	0.690068	0.027045	0.447439
	2.55	0.019778	0.684977	0.027312	0.439479
	3.05	0.019749	0.687266	0.027493	0.435842
	3.55	0.019256	0.704033	0.026971	0.450120
	4.05	0.019460	0.696114	0.027375	0.440065
	4.55	0.019634	0.692032	0.027015	0.443203
10000	2.05	0.006206	0.696346	0.008503	0.454550
	2.55	0.006029	0.714274	0.008620	0.445448
	3.05	0.006232	0.692673	0.008716	0.437637
	3.55	0.006092	0.709325	0.008725	0.435687
	4.05	0.006169	0.700740	0.008841	0.424714
	4.55	0.006085	0.708559	0.008681	0.438954
100000	2.05	0.001946	0.701022	0.002731	0.443344
	2.55	0.001951	0.700214	0.002758	0.436406
	3.05	0.001920	0.710636	0.002791	0.425773
	3.55	0.001941	0.703102	0.002709	0.450199
	4.05	0.001934	0.706422	0.002759	0.433438
	4.55	0.001966	0.695676	0.002755	0.437818

APPENDIX III Sample Variances of the First  $k$  Pareto Sample Points from Pareto Samples of Size 10 Million Using Various RNGs

