

PAPER • OPEN ACCESS

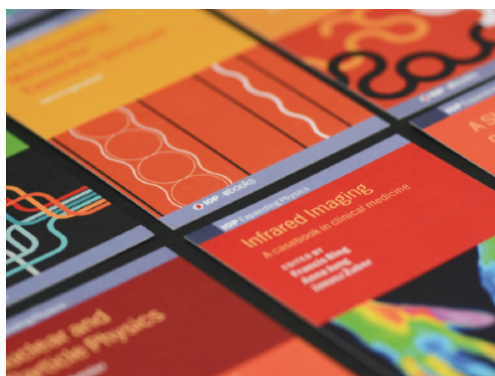
Methods of investigation and elimination of systematic errors in the realization of pseudo-random variables with heavy-tailed distributions

To cite this article: V N Zadorozhnyi *et al* 2021 *J. Phys.: Conf. Ser.* **1791** 012066

View the [article online](#) for updates and enhancements.

You may also like

- [Position Shift Analysis in Resist Reflow Process for Sub-50 nm Contact Hole](#)
Jee-Hye You, Joonwoo Park, Joon-Min Park *et al.*
- [Identification and Filtering of Random Error in Fiber Optic Current Transducer](#)
Zhengqi Tian, Qing Xu, Jian Liu *et al.*
- [One-dimensional numerical simulations of random sound impulses](#)
K Murawski and M Medrek



IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

Methods of investigation and elimination of systematic errors in the realization of pseudo-random variables with heavy-tailed distributions

V N Zadorozhnyi¹, T R Zakharenkova¹ and M Pagano²

¹Omsk State Technical University, 11, Mira Pr., Omsk 644050, Russia

²University of Pisa, 43, Lungarno Pacinotti, Pisa 56126, Italy

E-mail: zwn2015@yandex.ru, ZakharenkovaTatiana@gmail.com, Michele.Pagano@iet.unipi.it

Abstract. Shifts arising in generation of random variables, which are obtained by transformation of standard random numbers, with heavy-tailed distributions are investigated. The formula for calculation of deterministic moments shift for a random variable via a discretization spacing of used standard pseudo-random numbers is derived. The deterministic shift is calculated for a mean value of random variables that have the Pareto distribution and realized by the widely accepted inversion technique. The shift calculation results are verified and confirmed experimentally by means of the rapid Monte-Carlo technique – stratification method. By an example of the average queue length calculation, it is demonstrated how the shifts within distributions in queueing systems affect the queueing properties. The universal method for the realization of non-shifted random variables with heavy-tailed distributions, ARAND method, is developed and investigated. The method is based on the use of uniformly distributed between one and zero pseudo-random variables having a variable discretization spacing and represented in floating-point format.

1. Introduction

A probability distribution, defined by a probability distribution function (d.f.) $F(t)$, has a heavy tail

$\bar{F}(t) = 1 - F(t)$ if any constant $\lambda > 0$ satisfies the condition $\lim_{t \rightarrow \infty} \frac{\bar{F}(t)}{e^{-\lambda t}} = \infty$. Distributions with

asymptotic power tail $\bar{F}(t) \sim ct^{-\alpha}$, ($c > 0$, $\alpha > 0$) are typical examples of heavy-tailed distributions (HTD). Experience of queueing systems simulation shows that generators of having HTD random variables (r.v.) quite often realize these r.v. with significant deterministic shifts of d.f. moments [1, 2]. Such shifts of realized HTD lead to incorrect determining the quantitative and qualitative characteristics of simulated queueing systems [2, 3].

In monograph [3] practically showed how a value of the sampling mean depends on the capabilities of various RNGs. Therefore, an author of the monograph recommends using the best random number generator a researcher has access to.

A cause of moment shifts for HTD realized in simulation modeling (SM) was identified and analyzed in detail in [2]. The main reason is that an ordinary discretization spacing, ranging from $\varepsilon = 10^{-15}$ to $\varepsilon = 10^{-6}$, of used RNGs is sufficiently small for distributions with light tails (LTD) but for HTD it often is unacceptably large. In [2] there are also results of numerous experiments demonstrating the influence of incorrect HTD realization on calculated estimates of different characteristics for queueing



systems with HTD in SM, in particular, for queueing systems M/Pa/n, M/Pa/n/0, Pa/M/n, where symbols M and Pa represent exponential and Pareto distributions.

It is worth mentioning that in the generation of HTD samplings and their processing there is a problem caused by features of heavy tails – the problem of slow convergence of sampling moments to their exact values. This problem should not be confused with a deterministic moment shifts problem for HTD distributions realized in SM. The problem of slow convergence of sampling moments may be solved, for example, by required sampling extension [4] or by applying rapid Monte-Carlo techniques [5, 6, 7, 8, 9, 10]. The deterministic moment shifts problem is caused by weaknesses of program RNGs and can not be solved by a sampling extension. With an increase in the sample size of realized HTD to infinity (or to a period length of RNG), sampling moments converge to values different from the true ones.

In this article, we give the recommendations for shifts elimination of realized HTD that are to use a tail inversion technique and to use a variable discretization spacing ε that becomes smaller as realized standard pseudo-random numbers (PRN) approach to zero. We develop and investigate a universal ARAND method for moment shifts elimination of realized HTD.

We will consider in the article GPSS, as a widely spread and available simulation environment, and AnyLogic, as a simulation environment with wide functionality.

2. Generation of HTD

Value of a random variable (r.v.) ξ distributed according to some distribution function (d.f.) $F(t)$ can be obtained by the formula $\xi = \bar{F}^{-1}(r)$, where $\bar{F}^{-1}$ is the inverse of the function $\bar{F}$, r is standard random variable (SRV) uniformly distributed on (0, 1). Such approach to realization of r.v. values is called *inversion technique*.

Along with that a value of r.v. ξ may be realized by a *distribution tail inversion technique* when the tail $\bar{F}(t) = 1 - F(t)$ of d.f. $F(t)$ is inversed, rather than d.f. $F(t)$ itself. As a result of such inverse transformation for value of r.v. ξ is the following formula $\xi = F^{-1}(r)$.

When generating any r.v. with light-tailed distributions it does not make much difference which inversion technique is used – inversion technique of d.f. or tail inversion technique. But when generating r.v. with HTD, as we will see later in paragraph 5, the difference between these two techniques arises largely due to machine arithmetic properties.

Let us derive the formula for generating r.v. ξ having frequently used HTD – Pareto distribution Pa(K, α) that has the d.f.

$$F(t) = 1 - \left(\frac{K}{t}\right)^\alpha, \quad t \geq K,$$

where $K > 0$ is scale parameter (and the least value of the random variable ξ), α is shape parameter.

We can write the initial formal equation of distribution tail for r.v. $\xi \in \text{Pa}(K, \alpha)$ in the form:

$$\bar{F}(\xi) = \left(\frac{K}{\xi}\right)^\alpha = r,$$

and, by expressing ξ through r , we get the following formula for generating r.v. $\xi \in \text{Pa}(K, \alpha)$:

$$\xi = \frac{K}{\sqrt[\alpha]{r}} = Kr^{-1/\alpha}. \quad (1)$$

As can be seen from formula (1), values of r.v. ξ (large values of ξ) from the tail of the Pareto distribution are functionally dependent on small values of PRN r . Therefore, for correct realization of the distribution tail the quality of SRV generation has to be monitored, particularly in the area of its small values.

3. Shift of the HTD moments in SM

Generators of PRN in the considered SM environments have necessary requirements for conducting simulation experiments: PRN are uniformly distributed in the range (0,1), random and independent. However, when simulating queueing systems and queueing networks with HTD these necessary requirements are not sufficient. The requirement of small discretization spacing must be added to those described.

Due to limitations of the computers bit grid, a continuous SRV, realized by widely accepted RNGs, is generated as discrete PRN r of n -bit. A set of possible values of the PRN r is the set of equally-spaced points on the numerical axis that are located in the interval between zero and one. In view of this, generated by transformation of SRV the continuous r.v. ξ also become discrete. We will denote by ξ^* the discrete realizations of continuous r.v. ξ . Uniform distribution law of PRN r is sometimes called a *quasi-uniform distribution law* because of PRN r discretization. All possible values r in such law follow each other with equal spacing ε called *discretization spacing*.

A primary consideration in simulation of queueing systems and queueing networks with HTD is the shift of HTD moments owing to *discretization spacing* $\varepsilon > 0$ for PRN r . When generating HTD, an «ordinary» discretization spacing sometimes turned out to be too large. In simulation the values of calculated characteristics of queueing systems and queueing networks with HTD become shifted due to a shift of HTD moments.

A shift of HTD moments, causing by PRN discretization, can be expressed through discretization spacing ε of PRN. Pseudo-random numbers r take on numerical values with equal probabilities $\{r_1, r_2, r_3, \dots, r_N\} = \{\varepsilon, 2\varepsilon, 3\varepsilon, \dots, N\varepsilon\}$, where N is the number of possible values of r (N does not exceed the length of T RNG period).

Let us consider a set of r.v. ξ^* values of Pareto distribution realization obtained by formula (1). This r.v. has a set of values $\xi_1, \xi_2, \dots, \xi_N$ with equal probabilities which are obtained by substituting into the formula (1) values $\{r_1, r_2, r_3, \dots, r_N\} = \{\varepsilon, 2\varepsilon, 3\varepsilon, \dots, N\varepsilon\}$ of the PRN r . The first moments of r.v. ξ^* owing to equal probabilities of $\xi_1, \xi_2, \dots, \xi_N$ values can be expressed as follows:

$$M[(\xi^*)^k] = \frac{1}{N} \sum_{i=1}^N \left(\frac{K}{\sqrt[k]{r_i}} \right)^k = \frac{1}{N} \sum_{i=1}^N \left(\frac{K}{\sqrt[k]{i\varepsilon}} \right)^k = \frac{1}{N} \sum_{i=1}^N K^k \left(\frac{i}{N} \right)^{-k/\alpha}. \quad (2)$$

The sum in the formula comprises a large number N of summands, for example, one million (like in GPSS) or 10^{15} (like in AnyLogic). Such sums can be well calculated by means of modern mathematical packages in particular by WolframAlpha.com, and calculation accuracy can be easily controlled, for instance, by approximation of sums or their parts by corresponding integrals.

Let us calculate, for example, the m.e. for considered r.v. ξ^* realizing Pareto distribution with $K = 1$, $\alpha = 1.01$ (it is the case when the tail of Pareto distribution is highly heavy due to α being close to one). In the article, we consider the relevant on practice [11] values of $\alpha \leq 2$, when the second moment equals infinity.

For GPSS RNG we have $N = 10^6$ and, according to formula (2),

$$M(\xi^*) = \frac{1}{N} \sum_{i=1}^N \frac{1}{\sqrt[1.01]{i}} = \frac{1}{1000000} \sum_{i=1}^{1000000} \left(\frac{i}{1000000} \right)^{-1/1.01} = 13.4151. \quad (3)$$

In fact, the m.e. of Pareto distribution with $K = 1$, $\alpha = 1.01$ equals $K\alpha/(\alpha - 1) = 101$. By matching the result of (3) with this exact m.e., we see a significant shift of m.e. for r.v. ξ^* toward a decrement in value. The same calculation for RNG in AnyLogic when $N = 10^{15}$ gives the following:

$$M(\xi^*) = \frac{1}{10^{15}} \sum_{i=1}^{10^{15}} \left(\frac{i}{10^{15}} \right)^{-1/1.01} = 29.6621. \quad (4)$$

Consequently, a m.e. for realized r.v. ξ^* in AnyLogic is significantly shifted, albeit slightly less than in GPSS. A detailed investigation of variation of Pareto distribution moments shifts with discretization spacing ε at different types of tail heaviness is provided in [2]. The biggest shifts of distribution moments are shown to arise in critical areas where a distribution tail is particularly heavy. Herewith, true moments values do not need to be very large.

When realizing any HTD on the basis of standard RNG, one must take into account the features of RNG behavior and identify the “critical areas” where HTD shifts are the most significant.

When realizing any distribution $F(t)$ of r.v. ξ on the basis of standard RNG, samplings are selected from the set of N values $\{\xi_1, \xi_2, \dots, \xi_N\}$ with equal probabilities. The desired shape of the distribution $F(t)$ is achieved with a help of appropriate irregular arrangement of these values with equal probabilities along the numerical axis t .

4. Realization of HTD in simulation modeling

Let us consider the realization of r.v. $\xi \in \text{Pa}(1, 1.1)$ in GPSS World and calculate a shift of the first moment $M(\xi)$ by means of SM. The m.e. is equal to $M(\xi) = 11$ when $K = 1$, $\alpha = 1.1$, but at the same time a realized on practice m.e. $M(\xi^*)$ of r.v. ξ^* is considerably smaller than exact one.

To calculate the $M(\xi^*)$, we use SM. In order to increase an accuracy of SM results, we apply one of the variance reduction technique, namely the stratification method [5]. We split a set Ω of values PRN r into strata Ω_i :

$$\Omega_i = \{r : 10^{-i} < r \leq 10^{-(i-1)}\}, \quad i = 1, 2, \dots \quad (5)$$

(at the same time, the PRN r with 6 decimal bits after the point, as in GPSS, will be used, i.e. discretization spacing for PRN r will be equal to 10^{-6}).

We define bounds of corresponding strata Θ_i for r.v. ξ^* by formula (1). The strata for r and ξ^* are given in table 1.

Table 1. The strata for r and r.v. ξ^* .

Strata Ω_i for PRN r	Strata Θ_i for r.v. $\xi^* \in \text{Pa}(1, 1.1)$
$0.1 < r \leq 1,$	$1^{-1/\alpha} \leq \xi^* < 0.1^{-1/\alpha},$
$0.01 < r \leq 0.1,$	$0.1^{-1/\alpha} \leq \xi^* < 0.01^{-1/\alpha},$
$0.001 < r \leq 0.01,$	$0.01^{-1/\alpha} \leq \xi^* < 0.001^{-1/\alpha},$
$0.0001 < r \leq 0.001,$	$0.001^{-1/\alpha} \leq \xi^* < 0.0001^{-1/\alpha},$
$0.00001 < r \leq 0.0001,$	$0.0001^{-1/\alpha} \leq \xi^* < 0.00001^{-1/\alpha},$
$10^{-6} < r \leq 10^{-5},$	$10^{5/\alpha} \leq \xi^* < 10^{6/\alpha},$
$10^{-7} < r \leq 10^{-6},$	$10^{6/\alpha} \leq \xi^* < 10^{7/\alpha},$
...	...

Owing to such stratification, we can not only ensure the high accuracy for simulation estimate of m.e. $M(\xi^*)$ (for determination of its shift), but also trace changes of conditional m.e. when strata move away in the direction of large values of r.v. ξ^* .

The unconditional m.e. $M(\xi^*)$ is expressed in terms of the conditional m.e. in the following form:

$$M(\xi^*) = \sum_{i=1}^{\infty} w_i M_i^*, \quad (6)$$

where M_i^* is conditional m.e. of r.v. ξ^* in the stratum Θ_i , i.e. $M_i^* = M(\xi^* | \xi^* \in \Theta_i)$, and w_i^* is the probability that r.v. ξ^* will fall into the stratum Θ_i : $w_i^* = P(\xi^* \in \Theta_i)$. As an example, we calculate the probability w_1^* that r.v. ξ^* will fall into the stratum Θ_1 :

$$\begin{aligned} w_1^* &= P(1^{-1/\alpha} \leq \xi^* < 0.1^{-1/\alpha}) = P(0.1 < r \leq 1) = (1000000 - 100000) \cdot \varepsilon \\ &= (1000000 - 100000) \cdot 10^{-6} = 0.9. \end{aligned} \quad (7)$$

We calculate similarly the probabilities of falling into other strata for r.v. ξ^* , therefore, we obtain:

$$\begin{aligned} w_2^* &= 9 \cdot 10^{-2}, \quad w_3^* = 9 \cdot 10^{-3}, \quad w_4^* = 9 \cdot 10^{-4}, \quad w_5^* = 9 \cdot 10^{-5}, \quad w_6^* = 9 \cdot 10^{-6}, \\ w_7^* &= 10^{-6}, \quad w_i^* = 0 \text{ at } i > 7. \end{aligned} \quad (8)$$

The probability w_7^* differs from the previous ones by the form because of the only one from the million values of r.v. ξ^* falls into stratum. The calculation formula for the probability of this stratum takes the form $w_7^* = \dots = (1 - 0) \cdot 10^{-6} = 10^{-6}$.

In GPSS $h = 100$ million experiments with optimal distribution of them along strata were carried out, i.e. the number of experiments h_i with values $r \in \Omega_i$ was defined according to the formula [6]

$$h_i = \frac{w_i^* \sigma_i}{\sum_{j: w_j^* > 0} w_j^* \sigma_j} h, \quad \forall i: w_i^* > 0,$$

where $\sigma_i = \sqrt{D_i}$, $D_i = D(\xi^* | \xi^* \in \Theta_i)$ is the conditional variance of r.v. ξ^* defined when $\xi^* \in \Theta_i$, or, equivalently, when $r \in \Omega_i$.

Estimates $\hat{M}_i$ of conditional m.e. M_i^* we obtained for all strata (see table 2) as a result of such stratification in SM. To obtain an each estimate $\hat{M}_i$, we generated only those values r that are related to a stratum Ω_i . Such realization of values r in strata was carried out by means of built into GPSS generator Duniform(1, c , d), where 1 is the number of generator, c and d are the smallest and the largest values of an integer r.v. For instance, the values of r.v. ξ^* concerned with the first stratum (see table 1) were realized by the following expression:

$$(\text{Duniform}(1, 100001, 1000000) \# 10^{-6})^{(-1/\text{ALPHA})},$$

where ALPHA is the parameter α of Pareto distribution.

The exact values of conditional m.e. M_i^* , that take into account the discretization spacing $\varepsilon = 10^{-6}$ of built-in RNG, are calculated with the following consideration. Since all values of r.v. ξ^* have equal probabilities, they will also have equal probabilities provided that they belong to some stratum. Hence, for a conditional m.e. M_i^* to be calculated it is enough to find the arithmetic mean of r.v. ξ^* values that belong to a stratum Θ_i . Thus, for the first stratum we get the calculation formula:

$$M_1^* = \frac{1}{N_1} \sum_{r \in \Omega_1} r^{-1/\alpha} = \frac{1}{N_1} \sum_{r \in \Omega_1} r^{-1/1.1} = \frac{1}{900000} \sum_{j=100001}^{1000000} (j\varepsilon)^{-1/1.1}, \quad (9)$$

where $N_1 = 900000$ is the number of PRN r in the stratum Ω_1 . In the next strata we have $N_2 = 100000 - 10000 = 90000$, ..., $N_6 = 10 - 1 = 9$, $N_7 = 1 - 0 = 1$. The boundaries of summation in the stratum Ω_2 are from 10001 to 100000, in the stratum Ω_3 are from 1001 to 10000, ..., in the stratum Ω_6 are from 2 to 10, and in the stratum Ω_7 are from 1 to 1.

The results of corresponding calculations are given in the column M_i^* .

Table 2. Conditional and unconditional m.e. of r.v. ξ^* and $\xi \in \text{Pa}(1, 1.1)$ in strata Θ_i .

i	$\hat{M}_i$	M_i^*	$M_i(\xi)$
1	2.3083	2.3084	2.3084
2	18.7220	18.7238	18.7242
3	151.8362	151.8514	151.8774
4	1229.6927	1229.8194	1231.9242
5	9824.1359	9824.4155	9992.5167
6	69394.2800	69383.8378	81052.3836
7	284803.5868	284804.5868	657440.8724
8	0	0	5332705.6105
...	...	...	...
Result	8.0292	8.0297	11

Note that conditional m.e. of squared r.v. ξ^* in stratum Θ_1 may be calculated by formula (2) in the following way:

$$M[(\xi^*)^2 | \xi^* \in \Theta_1] = \frac{1}{N_1} \sum_{r \in \Omega_1} r^{-2/\alpha} = \frac{1}{N_1} \sum_{r \in \Omega_1} r^{-2/1.1} = \frac{1}{900000} \sum_{j=100001}^{1000000} (j\varepsilon)^{-2/1.1}$$

and, further, by subtracting the square of m.e. (9) from it we may find a conditional variance of r.v. ξ^* in this stratum. By this method the conditional variances were calculated for the above formula of optimal distribution of h experiments along strata.

The probabilities w_i of strata Θ_i for (theoretical) r.v. ξ can be calculated as follows:

$$w_1 = P(1^{-1/\alpha} \leq \xi < 0.1^{-1/\alpha}) = F(0.1^{-1/1.1}) - F(1^{-1/1.1}) = \bar{F}(1^{-1/1.1}) - \bar{F}(0.1^{-1/1.1})$$

$$= \left(\frac{1}{1^{-1/1.1}} \right)^{1.1} - \left(\frac{1}{0.1^{-1/1.1}} \right)^{1.1} = 0.9, \quad (10)$$

$$w_2 = 9 \cdot 10^{-2}, \dots, w_i = 9 \cdot 10^{-i}, \dots \quad (11)$$

Sequence 10, 11 differs from sequence (7), (8) beginning from the probability w_7 .

For the conditional m.e. $M_i(\xi)$ to be calculated, we derive an expression for a corresponding conditional tail of d.f. To do this, we use the well-known formula for a conditional d.f. that is determined when a r.v. falls into a given interval (a, b) :

$$F(t | a \leq \xi < b) = \begin{cases} 0, & \text{at } t < a, \\ \frac{F(t) - F(a)}{F(b) - F(a)}, & \text{at } a \leq t < b, \\ 1, & \text{at } b \leq t. \end{cases}$$

Therefore, we obtain a conditional tail of distribution:

$$\bar{F}(t | a \leq \xi < b) = 1 - F(t | a \leq \xi < b) = \begin{cases} 1, & \text{at } t < a, \\ \frac{F(b) - F(t)}{F(b) - F(a)}, & \text{at } a \leq t < b, \\ 0, & \text{at } b \leq t. \end{cases}$$

Given that a tail equals one to the left of a and equals zero to the right of b , we obtain a conditional m.e. $M(\xi | a \leq \xi < b)$ as follows:

$$M(\xi | a \leq \xi < b) = \int_0^{\infty} \bar{F}(\xi | a \leq \xi < b) dt = \int_0^a dt + \int_a^b \frac{F(b) - F(t)}{F(b) - F(a)} dt = a + \int_a^b \frac{F(b) - F(t)}{F(b) - F(a)} dt. \quad (12)$$

For the d.f. $\text{Pa}(1, \alpha)$ we have $F(b) = 1 - b^{-\alpha}$, $F(t) = 1 - t^{-\alpha}$, $F(a) = 1 - a^{-\alpha}$; hence, calculation formula (12) for this d.f. takes the following form:

$$M(\xi | a \leq \xi < b) = a + \int_a^b \frac{1 - b^{-\alpha} - (1 - t^{-\alpha})}{1 - b^{-\alpha} - (1 - a^{-\alpha})} dt = a + \frac{b^{1-\alpha} - a^{1-\alpha}}{a^{-\alpha} - b^{-\alpha}} \cdot \frac{1}{1 - \alpha} - \frac{(b - a)b^{-\alpha}}{a^{-\alpha} - b^{-\alpha}}.$$

Since $a = 10^{i-1/\alpha}$, $b = 10^{i/\alpha}$ in a stratum i , then:

$$M_i(\xi) = 10^{\frac{i-1}{\alpha}} + \frac{10^{\frac{i \cdot 1-\alpha}{\alpha}} - 10^{(i-1) \cdot \frac{1-\alpha}{\alpha}}}{10^{1-i} - 10^{-i}} \cdot \frac{1}{1-\alpha} + 10^{-i} \cdot \frac{10^{\frac{i}{\alpha}} - 10^{\frac{i-1}{\alpha}}}{10^{1-i} - 10^{-i}}. \quad (13)$$

Calculating $M_i(\xi)$ values with $\alpha = 1.1$ and, further, given known probabilities of strata (11), calculating the sum of the initial 100 summands by formula $M(\xi) = \sum_{i=1}^{\infty} w_i M_i(\xi)$, we obtain $M(\xi) \approx 10.99999999$.

The sum of the initial 150 summands yields the approximation $M(\xi) \approx 10.999999999999$. A computer with double precision no longer distinguishes the sum of the initial 160 and more summands from 11.00000... .

The exact theoretical values of conditional m.e. $M_i(\xi)$ for continuous r.v. ξ are given in the last column of table 2.

In the last line of the table (line «Result») the following values of unconditional m.e. are given: (1) the estimate $\hat{M}(\xi^*) = \sum_i w_i^* \hat{M}_i = 8.0292$ obtained by SM in GPSS, (2) an exact value of

$M(\xi^*) = \sum_i w_i^* M_i^* = 8.0297$ and (3) the true value of the unconditional m.e. $M(\xi) = 11$ for a theoretical

r.v. ξ which should have been realized without shifts. Values match for the calculated m.e. $M(\xi^*)$ and the estimate $\hat{M}(\xi^*)$ with an accuracy at 4 significant digits indicates high accuracy of simulation, which is obtained by the stratification method. And at the same time it enhances confidence in the results of accurate calculation of $M(\xi^*)$.

Table 2 shows how the conditional m.e. for r.v. ξ^* is becoming increasingly remote, as strata approach to large values of r.v. ξ^* , from the true conditional m.e. for r.v. ξ . Values of realized on practice r.v. ξ^* do not exceed 284803.6 due to discretization spacing $\varepsilon = 10^{-6}$ which in this case – the case of the HTD Pareto – turns out to be too large. This conclusion is correct for other HTD as well. In the case of LTD, conditional m.e. for r.v. ξ are so fast decreasing that at discretization spacing $\varepsilon = 10^{-6}$ there is no noticeable difference between conditional m.e. for ξ and conditional m.e. for ξ^* . And m.e. $M(\xi^*)$ are known almost not to be shifted.

Shifts of HTD moments may result in considerable errors in SM of queueing systems and networks. Let us consider the following example. Suppose service time ξ have a $\text{Pa}(K, \alpha)$ distribution. If $\alpha \in (1, 2]$, stationary average queue length L of requests will equal infinity at any load coefficient $\rho \in (0, 1)$. According to the Pollaczek-Khinchine formula, here at any $\rho \in (0, 1)$ and at any intensity of arrival request flow $\lambda > 0$ we obtain:

$$L = \frac{\lambda \xi^{(2)}}{2(1-\rho)} = \infty, \quad (14)$$

since the second moment $\xi^{(2)}$ of service time ξ at $\alpha \in (1, 2]$ is equal to infinity. But if we investigate this queueing system by SM, it will be realized the service time ξ^* with shifted moments instead of

time $\xi \in \text{Pa}(K, \alpha)$. For example, at $K = 1$, $\alpha = 2$ (the lightest tail in the range $1 < \alpha \leq 2$) and $\varepsilon = 10^{-15}$ (as in AnyLogic), in accordance with formula (2), the second moment is realized

$$\xi^{*(2)} = M[(\xi^*)^2] = \frac{1}{10^{15}} \sum_{i=1}^{10^{15}} \left[(10^{-15}i)^{-1/2} \right]^2 = 35.116,$$

as a result, according to the Pollaczek-Khinchine formula, with a sufficiently large run length of the model, we get

$$L \approx \frac{35.116 \cdot \lambda}{2(1-\rho)} \approx \frac{17.5 \cdot \lambda}{1-\rho}. \quad (15)$$

This conclusion is also confirmed by the results of simulation experiments. A feature of the considered case is the fundamental difference between the simulation solution (15) and the correct solution (14).

Common values of discretization spacing ε for PRN, laying in the range from $\varepsilon = 10^{-15}$ to $\varepsilon = 10^{-6}$, lead to significant shifts of the HTD moments realized in SM. The significant shifts of the HTD moments, in return, lead to considerable, sometimes to unacceptable, errors of queueing systems and queueing networks simulation results. Therefore, this determines the relevance of methods development for radical decrease of discretization spacing ε for PRN r . Moreover, it is recommended the developed method for decreasing of HTD shifts to be based on existing RNGs in SM systems without any increase in the length for the existing n bit grid of computers.

5. ARAND method

One may use methods of «long arithmetic» and «long weighted sums» [2] to reduce ε and by this to reduce shifts of realized HTD. These methods are based on program emulation of the increase the length of bit grid which is used by RNG. In this case, for a r.v. value to be realized with the reduced discretization spacing it is required turning to the standard RNG several times. Thus, the generation speed of pseudo-random numbers decreases several times, and, consequently, a generation period length is reduced several times. Moreover, as can be seen from (3) and (4), 2–3-fold increase in the bit grid length, which is equivalent to decrease of ε by approximately 10-15 orders of magnitude, may not result in sufficient decrease of moments shift for realized HTD.

The most flexible and effective method of generating PRN and eliminating the HTD shifts is the ARAND method [2]. The ARAND method transforms the realized by existing RNGs n -bit uniformly distributed in the range $[0,1]$ numbers r with a constant discretization spacing ε to uniformly distributed in the range $[0,1]$ numbers r with a variable discretization spacing $\varepsilon(r)$ such that the discretization spacing ε is always approximately proportional to values r . Owing to this, the ARAND allows us to obtain sufficiently large values of realized r.v. ξ^* which are necessary for the realization of unbiased HTD moments.

The ARAND method is generalization of cascade method [2] eliminating the moments shifts for power-law distributions. Figure 1 depicts the algorithm for implementing the ARAND method procedure written in Java language in AnyLogic.

In the basic version of the ARAND algorithm the first call of RNG (see figure 1) is made when scale $C = 1$ and if generated value r falls into $0.1 \leq r < 1$, it will contain n significant decimal digits and will be returned to the calling program as a realized PRN value by the procedure. In this case, the discretization spacing $\varepsilon = 1/N$: in AnyLogic $N \approx 10^{15}$, in GPSS $N = 10^6$. The number of possible values r in this range is $0.9N$, i.e. $9 \cdot 10^{14}$ for AnyLogic and $9 \cdot 10^5$ for GPSS.

If generated value r at the first call of RNG falls into $0 \leq r < 0.1$, then it will be discarded, the scale C will be increased by an order of magnitude, and a value r will be generated again by the second call of RNG. The new value $r \in [0, 1]$ in terms of the scale C is interpreted as value belonging to the range $0 \leq r < 0.1$. In this range the new value r is also distributed uniformly as the previous discarded value does. If the new value falls into $0.1 \leq r < 1$ (in terms of the scale C which is interpreted as $0.01 \leq r < 0.1$), it will contain n significant decimal digits, will be divided by C and will be returned to the calling program. Division by C preserves the n significant decimal digits for r , since division and its result are implemented in floating-point arithmetic. In this case, the discretization spacing

$\varepsilon = 1/(10N)$ is decreased by an order of magnitude. The number of possible values r in the range $0.01 \leq r < 0.1$ is the same as in the previous range.

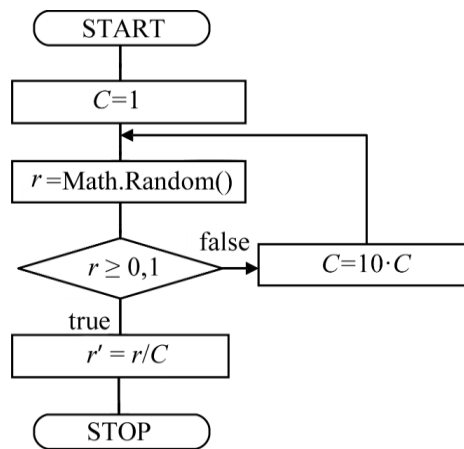


Figure 1. Scheme of the algorithm for the procedure ARAND in Java. The operation $r' \leftarrow r/C$ allows us to generate almost arbitrarily small values r' , transforms them to floating-point numbers, makes a discretization spacing an irregular one, and leaves uniform in the range between 0 and 1 the unconditional distribution for r.v. r' .

There occurs in a similar way transitions of r to the next ranges decreasing by an order of magnitude until the corresponding r values can be represented in floating-point form. At double precision of calculations, the last range lays to the left of the range $10^{-301} \leq r < 10^{-300}$. But with a uniform distribution of r the probability of reaching the range $10^{-301} \leq r < 10^{-300}$, which approximately equal 10^{-300} , is so small that it is an event occurring, on average, only once during 10^{300} calls of RNG, and, according to Bremermann's limit, at any achievable computer speed, this event will not happen even in a billion years. Note that in the range $10^{-301} \leq r < 10^{-300}$ the number of possible values r is still equal to $0.9N$ as in the range $0.1 \leq r < 1$.

For instance, in GPSS the initial 10 numbers from the standard RNG Uniform(1,0,1) equal to 0.842366, 0.777717, 0.880991, 0.260493, 0.463553, **0.083898**, **0.022383**, **0.948592**, 0.344245, 0.929906 are transformed by the ARAND method (basic version) to numbers 0.842366, 0.777717, 0.880991, 0.260493, 0.463553, **0.00948592**, 0.344245, 0.929906.

Due to losses of a part of the numbers in the initial sequence, the period length of the RNG with ARAND is on average $1/0.9$ times less than that of the original RNG Uniform(j , 0,1). A period of the RNG with ARAND, that transforms a sequence from the RNG Uniform(1,0,1), contains 1 843 200 000 numbers, the least of those is $6.54026 \cdot 10^{-11}$ [2]. Generation of PRN with ARAND slows down just $1/0.9$ times.

On the exit of a loop, after i -th passing of it (see figure 1) and division r by the scale C , a PRN r is uniformly distributed on the interval $\Omega_i = [10^{-i}, 10^{-(i-1)}]$, $i = 1, 2, \dots$.

The probability of the loop termination after i -th passing of the loop body equals $0.9 \cdot 0.1^{-(i-1)}$ (see figure 1), i.e. equals the length of an interval Ω_i . Consequently, the probability of r falling into any interval $\{\Omega_i\}$ is equal to the length of the corresponding interval. The probability of falling into any part of the interval Ω_i equals the length of the part. Consequently, the probability of falling into any part $[a, b]$ of the interval $[0, 1]$ is equal to the length $(b - a)$ of this part. And, consequently, the number r on the exit of ARAND procedure is distributed uniformly on the $[0, 1]$.

Note that the choice of specific points (specific r values) on the interval $[0, 1]$ occurs with probabilities depending on the position of these points. Any of the $0.9N$ points on the interval Ω_i has the probability $0.9 \cdot 0.1^{-(i-1)} / 0.9N = 0.1^{-(i-1)} / N$, where the typical values N in different programs belong to the range from 10^6 to 10^{15} . The density of the chosen points increases by an order of magnitude on each of the following intervals Ω_i .

As shown above, when we use classic RNGs, the desired shape of the *non-uniform* distributions $F(t)$ of r.v. ξ is achieved with a help of appropriate irregular arrangement of values ξ with equal probabilities. Now one can see that the ARAND method realizes the *uniform* distribution of r.v. r on the interval $[0, 1]$ by means of an appropriate non-uniform distribution of values r with *non-equal* probabilities.

The considered method of realizing SRV r with a variable discretization spacing $\alpha(r)$ results in correct program HTD realization, since it does not cut off tails of distributions in the area of large values of r.v. in contradistinction to classic RNGs. And, consequently, ensures the correct realization of queueing systems characteristics depending on the distribution moments.

6. Conclusion

The research of HTD realized in GPSS World is carried out on the example of Pareto distribution realization. It is shown that moment shifts for HTD resulted from the disappearance of the r.v. values, which are too large to be neglected, with small probabilities in the generated samplings. Their disappearance is related to not sufficiently small discretization spacing of built-in RNGs in GPSS.

The simulation experiment confirming the calculations was carried out by means of the stratification method that significantly accelerated the calculation of the conditional mathematical expectations within each stratum and the corresponding unconditional mathematical expectation. The problems identified during HTD simulation can be eliminated by means of the ARAND method for the correct realization of HTD.

7. References

- [1] Gross D, Shortle J F, Fischer M J and Masi D 2002 Simulation input analysis: difficulties in simulating queues with Pareto service, *Proc. Winter Simulation Conference: Exploring New Frontiers* (San Diego) (ACM Press) pp 407–15
- [2] Zadorozhnyi V N 2016 *Queues and power-law distributions* (SBRAS Publisher) p 162 (in Russian)
- [3] Feitelson D G 2015 *Workload modeling for computer systems performance evaluation* (New York: Cambridge University Press) p 551
- [4] Zadorozhnyi V N and Zakharenkova T R 2016 Methods of Simulation Queueing Systems with Heavy Tails, ed A Dudin et al *Proc. Information Technologies and Mathematical Modelling (Katun)* (Cham: Springer) pp 168–83
- [5] Kleinen J P C 1974 *Statistical techniques in simulation* (New York: Marcel Decker) vol 1 p 285
- [6] Kovalenko I N and Kuznetsov I Yu 1988 Methods of calculating highly reliable systems (Moscow: Radio and Communications) p 176 (in Russian)
- [7] Huang M L, Pollanen M and Yuen W K 2007 An Efficient Randomized Quasi-Monte Carlo Algorithm for the Pareto Distribution *Monte Carlo Methods and Appl.* **13** pp 1–20
- [8] Zadorozhnyi V N and Zakharenkova T R 2016 Rapid method for calculation of loss probabilities in fractal queueing systems *Proceedings in Cybernetics* **31** pp 102–12 (in Russian)
- [9] Lukashenko O V, Morozov E V and Pagano M 2017 On the Efficiency of Bridge Monte-Carlo Estimator *Informatics and Applications* **11** pp 16–24
- [10] Mikhailov G A and Voytishek A V 2018 *Statistical Modeling. Monte Carlo Methods* (Moscow: Yurayt) p 371 (in Russian)
- [11] Leland W E, Willinger W, Taqqu M S and D V Wilson 1993 On the Self-Similar Nature of Ethernet Traffic, *Proc. SIGCOMM '93 (San Francisco)* (IEEE Publisher) pp 183–93