УДК 519.21

А.С. Вамболь

Национальный аэрокосмический университет имени Н.Е. Жуковского ”ХАИ”, Харьков

МАЖОРАНТО-СУПЕРПОЗИЦИОННЫЙ АЛГОРИТМ ГЕНЕРАЦИИ СЛУЧАЙНОЙ ВЕЛИЧИНЫ, ИМЕЮЩЕЙ РОБАСТНОЕ РАСПРЕДЕЛЕНИЕ СОЛИТОНА

В статье предложен мажорантно-суперпозиционный алгоритм генерации дискретной случайной величины, имеющей робастное распределение солитона, совмещающий в себе метод суперпозиции, мажорантный метод исключения, метод обратного преобразования и стандартный алгоритм генерации дискретной случайной величины. Установлено, что в сравнении со стандартным алгоритмом предлагаемый алгоритм имеет преимущество в отношении производительности. Мажорантно-суперпозиционный алгоритм может быть использован в программной реализации кодера кодов LT.

Ключевые слова: *мажорантно-суперпозиционный алгоритм, робастное распределение солитона, фонтанные коды, коды LT, генерация случайных величин, повышение производительности.*

Введение

Помехоустойчивое кодирование цифровых данных имеет важное значение, поскольку большинство современных систем связи основано на передаче сообщений в цифровом виде. В области помехоустойчивого кодирования цифровой информации практическую ценность представляет новый класс помехоустойчивых кодов – фонтанные коды. Кодами этого класса можно закодировать любое сообщение конечного размера потенциально неограниченным потоком независимых пакетов. Эти коды имеют простые алгоритмы декодирования и позволяют на практике получать результаты, близкие к предельным возможностям помехоустойчивого кодирования [1].

Случайный фонтанный код является несистематическим. В каждый из кодовых символов с вероятностью 0.5 входит каждый из K исходных символов. Сложением битов по модулю 2 (операцией XOR) всех входящих исходных символов вычисляются значения k битов кодового символа. В результате кодер генерирует случайный код (random fountain), в среднем используя $K/2$ операций XOR для вычисления одного кодового символа. Эта величина, которая называется стоимостью кодирования, оказывается сравнимой с K, и поэтому ее следует считать большой [1]. Количество исходных символов, входящих в кодовый символ, называется его степенью.

Исторически первым классом фонтанных кодов стали коды LT (Luby transform code), изобретённые М. Лаби (Michael Luby) в 1998 году и опубликованные в 2002 году. Алгоритм кодирования аналогичен алгоритму случайного фонтанного кода, однако имеет принципиально иное распределение степени кодового символа $\rho(d)$. Суть LT-кода именно в сочетании этого "хорошего" распределения с простым и быстрым алгоритмом декодирования [1]. Распределение степени кодового символа $\rho(d)$ является важнейшей частью алгоритма. Часть кодовых символов должна иметь высокую степень (близкую к K), чтобы гарантировать, что нет исходных символов, не входящих ни в один из кодовых. Многие кодовые символы должны иметь низкую степень, чтобы процесс декодирования мог запуститься и поддерживаться. Кроме того, общее количество операций сложения в процессах кодирования и декодирования должно оставаться небольшим [2].

Изначально в качестве закона распределения степени кодового символа $\rho(d)$ в данном коде предполагалось использовать идеальное распределение солитона (ideal soliton distribution). Дальнейшие исследования продемонстрировали, что на практике это распределение работает плохо, потому что флуктуации вокруг ожидаемого поведения делают сбой процесса декодирования очень вероятным. Это привело к созданию робастного распределения солитона (robust soliton distribution), используемого в кодах LT в качестве закона распределения степени кодового символа [2].

Таким образом, проблема создания высокопроизводительных алгоритмов генерации дискретной случайной величины, подчиняющейся робастному распределению солитона, является актуальной.

Целью статьи является описание разработанного мажорантно-суперпозиционного алгоритма генерации дискретной случайной величины, имеющей робастное распределение солитона, и исследование его производительности. Предлагаемый алгоритм разработан в качестве средства уменьшения времени генерации случайной величины, подчиняющейся данному распределению, по сравнению со стандартным алгоритмом, являющимся универсальным средством генерации дискретных случайных величин.

1. Особенности робастного распределения солитона

Идеальное распределение солитона является однопараметрическим и имеет вид:

$$\rho(d) = \begin{cases} \dfrac{1}{K} & \text{для } d = 1, \\ \dfrac{1}{d(d-1)} & \text{для } d = 2,3,\ldots,K, \end{cases}$$

где параметром является K – количество исходных символов [2].

Робастное распределение солитона является трёхпараметрическим. Его дополнительными параметрами являются δ – верхняя граница вероятности сбоя процесса декодирования при условии получения K' = KZ кодовых символов и c – константа порядка единицы, которая на практике может рассматриваться в качестве свободного параметра, дающего хорошие результаты при значении немного меньше единицы [2]. Для аналитического задания его функции вводится вспомогательное трёхпараметрическое распределение τ(d). Робастное распределение солитона имеет вид [2]:

$$\mu(d) = \frac{\rho(d) + \tau(d)}{Z},$$

где
$$Z = \sum_{d=1}^{K} \left(\rho(d) + \tau(d)\right),$$

$$\tau(d) = \begin{cases} \dfrac{S}{K} \cdot \dfrac{1}{d} & \text{для } d = 1,2,\ldots,\left(\dfrac{K}{S}\right) - 1; \\ \dfrac{S}{K} \ln\left(\dfrac{S}{\delta}\right) & \text{для } d = \dfrac{K}{S}; \\ 0 & \text{для } \dfrac{K}{S} < d \le K, \end{cases}$$

где
$$S = c \cdot \ln\left(\frac{K}{\delta}\right)\sqrt{K}.$$

2. Стандартный алгоритм генерации дискретной случайной величины

Стандартный алгоритм генерации дискретной случайной величины с заданным законом распределения, принимающей конечное число значений N, основан на разбиении единичного отрезка на N интервалов, длины которых равны вероятностям появления соответствующих значений случайной величины. Процедура генерации осуществляется следующим образом: генерируется случайная величина со стандартным равномерным распределением, определяется интервал, в который она попала, и возвращается значение, соответствующее данному интервалу [3].

В случае применения данного алгоритма для генерации робастного распределения солитона критерием попадания стандартной равномерно распределённой случайной величины r в интервал, которому соответствует значение d, является выполнение условия:

$$\sum_{i=1}^{d-1} \mu(i) < r \le \sum_{i=1}^{d} \mu(i).$$

Следовательно, для робастного распределения солитона стандартный алгоритм имеет вид, представленный на рис. 1.

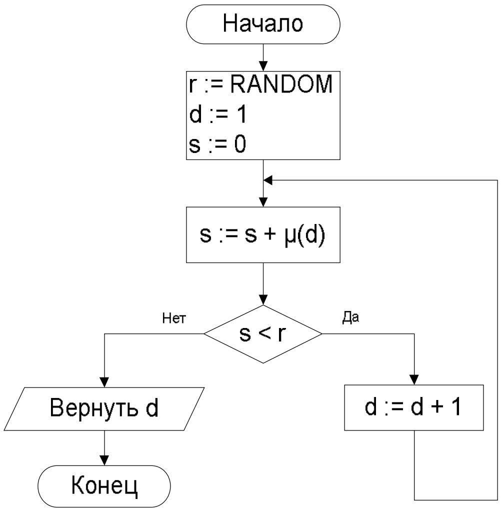

Рис. 1. Стандартный алгоритм генерации дискретной случайной величины для робастного распределения солитона

В качестве примера практического применения стандартного алгоритма, представленного на рис. 1, может быть приведена реализация генератора робастного распределения солитона в рамках проекта по разработке кодера и декодера кодов LT на языке C# с открытым исходным кодом [4].

3. Мажорантно-суперпозиционный алгоритм генерации робастного распределения солитона

Пусть:

$$p_1 = \frac{1+S}{KZ},$$

$$p_2 = \frac{S+K}{2KZ}, \qquad p_s = \frac{S}{KZ}\ln\left(\frac{S}{\delta}\right),$$

$$p_\rho = \frac{0.5K - 1}{KZ}, \qquad p_\tau = \frac{S}{KZ}\sum_{d=3}^{[K/S]-1} \frac{1}{d}.$$

Поскольку

$$\sum_{d=1}^{K}\rho(d)=\frac{1}{K}+\sum_{d=2}^{K}\frac{1}{d(d-1)}=$$

$$=\frac{1}{K}+\sum_{d=2}^{K}\left(\frac{1}{d-1}-\frac{1}{d}\right)=\frac{1}{K}+\sum_{d=2}^{K}\frac{1}{(d-1)}-\sum_{d=2}^{K}\frac{1}{d}=$$

$$=\frac{1}{K}+\sum_{d=1}^{K-1}\frac{1}{d}-\sum_{d=2}^{K}\frac{1}{d}=\frac{1}{K}+1-\frac{1}{K}=1,$$

то $$Z=\sum_{d=1}^{K}\left(\rho(d)+\tau(d)\right)=1+\sum_{d=1}^{K}\tau(d)=$$

$$=1+\frac{S}{K}\left(\sum_{d=1}^{[K/S]-1}\frac{1}{d}+\ln\left(\frac{S}{\delta}\right)\right).$$

Следовательно,

$$p_1+p_2+p_s+p_\rho+p_\tau=$$

$$=\frac{1+S}{KZ}+\frac{S+K}{2KZ}+\frac{S}{KZ}\ln\left(\frac{S}{\delta}\right)+\frac{0.5K-1}{KZ}+$$

$$+\frac{S}{KZ}\sum_{d=3}^{[K/S]-1}\frac{1}{d}=\frac{1}{Z}+\frac{S}{KZ}\sum_{d=1}^{[K/S]-1}\frac{1}{d}+\frac{S}{KZ}\ln\left(\frac{S}{\delta}\right)=$$

$$=\frac{1}{Z}\left(1+\frac{S}{K}\left(\sum_{d=1}^{[K/S]-1}\frac{1}{d}+\ln\left(\frac{S}{\delta}\right)\right)\right)=1.$$

Определим 2 дискретные случайные величины:

$$f_\rho(d)=\begin{cases}0 & \text{для } d=1,2\\ \dfrac{1}{p_\rho Z}\dfrac{1}{d(d-1)} & \text{для } d=3,\ldots,K,\end{cases}$$

$$f_\tau(d)=\begin{cases}0 & \text{для } d=1,2,\left[\dfrac{K}{S}\right],\ldots,K\\ \dfrac{S}{p_\tau KZ}\dfrac{1}{d} & \text{для } d=3,\ldots,\left[\dfrac{K}{S}\right]-1.\end{cases}$$

Дискретная случайная величина, имеющая закон распределения $f_\rho(d)$ или $f_\tau(d)$, может существовать, поскольку

$$\sum_{d=1}^{K}f_\rho(d)=\frac{1}{p_\rho Z}\sum_{d=3}^{K}\rho(d)=\frac{1}{p_\rho Z}\left(\sum_{d=1}^{K}\rho(d)-\right.$$

$$\left.-\rho(1)-\rho(2)\right)=\frac{0.5K-1}{K}\left(0.5-\frac{1}{K}\right)=1,$$

$$\sum_{d=1}^{K}f_\tau(d)=\frac{1}{p_\tau}\frac{S}{KZ}\sum_{d=3}^{[K/S]-1}\frac{1}{d}=\frac{p_\tau}{p_\tau}=1.$$

Также определим три вырожденные дискретные случайные величины:

$$f_1(d)=\begin{cases}1 & \text{для } d=1,\\ 0 & \text{для } d=2,\ldots,K,\end{cases}$$

$$f_2(d)=\begin{cases}1 & \text{для } d=2,\\ 0 & \text{для } d=1,3,4,\ldots,K,\end{cases}$$

$$f_s(d)=\begin{cases}1 & \text{для } d=\left[\dfrac{K}{S}\right],\\ 0 & \text{для } d=1,2,\ldots,\left[\dfrac{K}{S}\right]-1,\left[\dfrac{K}{S}\right]+1,\ldots,K.\end{cases}$$

Тогда для d = 1, 2, ..., K верно следующее равенство (табл. 1):

$$\mu(d)=p_1f_1(d)+p_2f_2(d)+$$
$$+p_sf_s(d)+p_\rho f_\rho(d)+p_\tau f_\tau(d).$$

Таблица 1

Результаты доказательства представимости $\mu(d)$ в виде линейной комбинации функций распределений

Значения d	Значения обеих частей данного равенства
1	$\frac{1+S}{KZ}$
2	$\frac{S+K}{2KZ}$
$3,\ldots,\left[\frac{K}{S}\right]-1$	$\frac{1}{Zd}\left(\frac{1}{d-1}+\frac{S}{K}\right)$
$\left[\frac{K}{S}\right]$	$\frac{1}{Z}\left(\frac{1}{d(d-1)}+\frac{S}{K}\ln\left(\frac{S}{\delta}\right)\right)$
$\left[\frac{K}{S}\right]+1,\ldots,K$	$\frac{1}{Z}\frac{1}{d(d-1)}$

Поскольку функция робастного распределения солитона $\mu(d)$ представима в виде линейной комбинации функций вероятностей, коэффициенты при которых неотрицательны и в сумме дают 1, для генерации случайной величины, подчиняющейся закону $\mu(d)$, может быть использован метод суперпозиции. Его суть состоит в случайном выборе функции вероятности, входящей в данную линейную комбинацию, с последующей генерацией случайной величины в соответствии с выбранной функцией вероятности, причём частота выбора каждой функции распределения должна быть равна коэффициенту при ней [3].

Случайный выбор функции вероятности может быть реализован с использованием стандартного алгоритма генерации дискретной случайной величины. Обозначим верхние границы интервалов в пределах единичного отрезка, при попадании стандартной равномерно распределенной величины в которые выбираются соответствующие им функции вероятности:

$B_2 = p_2$,

$B_S = B_2 + p_S$,

$B_1 = B_S + p_1$,

$B_\tau = B_1 + p_\tau$,

$B_\rho = B_\tau + p_\rho = 1$.

Тогда случайный выбор функции вероятности может быть реализован следующим образом. Генерируется стандартная равномерно распределенная величина r. При попадании r в интервал $[0; B_2]$ выбираем распределение $f_2(d)$, интервалу $(B_2; B_S]$ соответствует распределение $f_S(d)$, интервалу $(B_S; B_1]$ – распределение $f_1(d)$, интервалу $(B_1; B_\tau]$ – распределение $f_\tau(d)$, в случае попадания r в интервал $(B_\tau; 1]$ выбирается распределение $f_\rho(d)$.

Очевидно, что генерация вырожденных распределений $f_1(d)$, $f_2(d)$ и $f_S(d)$ заключается в возврате значения, имеющего единичную вероятность для соответствующего распределения.

Распределения $f_\rho(d)$ и $f_\tau(d)$ можно генерировать с использованием мажорантного метода исключения. Данный метод применительно к генерации непрерывных случайных величин с заданной плотностью распределения описан в [5]. Он может быть модифицирован для решения задачи генерации дискретной случайной величины с заданным законом распределения, принимающей целые значения a, a + 1, ..., b. Данная модификация мажорантного метода исключения может быть описана следующим образом.

Выбирается функция g(x), график которой полностью повторяет форму гистограммы частот генерируемой дискретной случайной величины f(x) на интервале [a; b + 1). Таким образом, для g(x) участок области определения [a; b + 1) разбит на интервалы единичной длины, в пределах которых функция сохраняет значение, пропорциональное значению f(x) от целой части аргумента, принадлежащего соответствующему интервалу. Следовательно, функцию g(x) можно выбрать по формуле

$$g(x) = kf(\lfloor x \rfloor),$$

где k – произвольное положительное число.

Затем выбирается функция m(x), являющаяся мажорантой для g(x) на интервале [a; b + 1), таким образом, что для

$$M(x) = \int_a^x m(z)dz$$

существует обратная функция $M^{-1}(x)$.

Следует отметить, что если f(x) определена всюду на интервале [a – 1; b + 1) и является строго монотонно убывающей, то в качестве мажоранты для g(x) можно выбрать $m(x) = kf(x-1)$. Поскольку для любого числа x выполняется неравенство $x - 1 < \lfloor x \rfloor$ и f(x) строго монотонно убывает, то $kf(\lfloor x \rfloor) < kf(x-1)$, следовательно, g(x) < m(x) для любого x, принадлежащего интервалу [a; b + 1).

Генерация дискретной случайной величины f(x) осуществляется путём генерации точки, равномерно распределённой в области A, ограниченной осью OX, мажорантой m(x) и прямыми

$$x = a \quad \text{и} \quad x = b + 1.$$

В качестве результата возвращается значение генерируемой случайной величины, соответствующее столбцу, являющемуся областью под графиком g(x), в котором оказалась сгенерированная точка. Из определения функции g(x) следует, что значение, соответствующее столбцу, в который попала сгенерированная точка, равно целой части её координаты x. Если сгенерированная точка окажется над графиком g(x), генерация точки осуществляется повторно.

Так как столбцы имеют одинаковую ширину, а их высоты пропорциональны частотам соответствующих значений случайной величины f(x), отношение вероятности попадания сгенерированной точки в выбранный столбец к вероятности её попадания в область под графиком g(x) равно частоте соответствующего значения случайной величины f(x), что свидетельствует о правильности алгоритма.

Необходимо отметить, что мажоранту m(x) следует выбирать таким образом, чтобы как можно большая часть площади области A находилась под графиком g(x). Такой подход к выбору m(x) направлен на повышение производительности мажорантного метода исключения, поскольку в результате его применения достигается уменьшение вероятности попадания сгенерированной точки в область над графиком g(x), а следовательно, и уменьшение вероятности повторной генерации точки.

Генерация точки, равномерно распределённой в области A, может быть осуществлена следующим образом.

Пусть

$$G = \int_a^{b+1} m(x)dx, \qquad q(x) = \frac{m(x)}{G},$$

тогда координата x генерируется в соответствии с распределением непрерывной случайной величины с функцией плотности q(x) и областью значений [a; b + 1). Затем генерируется координата y в соответствии с непрерывным равномерным распределением случайной величины с областью значений $[0; m(x_r))$, где x_r – сгенерированное значение координаты x [5].

Для генерирования случайной величины, в соответствии с которой генерируется координата x точки, может быть применён метод обратного преобразования. В соответствии с данным методом, координата x генерируется по формуле $x_r = Q^{-1}(r)$, где r – случайная величина, имеющая стандартное равномерное распределение, $Q^{-1}(x)$ – функция квантилей распределения координаты x, являющаяся обратной для функции распределения данной случайной величины, имеющей вид [5]:

$$Q(x) = \int_a^x q(z)\,dz.$$

Применительно к распределениям $f_\rho(d)$ и $f_\tau(d)$ мажорантный метод исключения может быть описан следующим образом. Поскольку случайные величины с распределениями $f_\rho(d)$ и $f_\tau(d)$ принимают с ненулевой вероятностью значения 3, ..., K и 3, ..., $\left[\frac{K}{S}\right] - 1$ соответственно, то $a_\rho = 3$, $b_\rho = K$, $a_\tau = 3$ и $b_\tau = \left[\frac{K}{S}\right] - 1$.

Выберем

$$g_\rho(x) = \frac{1}{\lfloor x \rfloor\left(\lfloor x \rfloor - 1\right)}, g_\tau(x) = \frac{1}{\lfloor x \rfloor}.$$

Поскольку $f_\rho(d)$ и $f_\tau(d)$ строго монотонно убывают на интервалах $[a_\rho - 1;\ b_\rho + 1)$ и $[a_\tau - 1;\ b_\tau + 1)$ соответственно, то в качестве мажорант выберем

$$m_\rho(x) = \frac{1}{(x-1)(x-2)}, \quad m_\tau(x) = \frac{1}{x-1}.$$

Пусть

$$U = 2 - \frac{2}{K}, \qquad V = \frac{1}{2}\left[\frac{K}{S}\right] - 0.5,$$

тогда

$$G_\rho = \int_3^{K+1} \frac{dx}{(x-1)(x-2)} = \ln(U),$$

$$G_\tau = \int_3^{\left[\frac{K}{S}\right]} \frac{dx}{x-1} = \ln(V),$$

$$q_\rho(x) = \frac{1}{\ln(U)(x-1)(x-2)},$$

$$q_\tau(x) = \frac{1}{\ln(V)(x-1)},$$

$$Q_\rho(x) = \int_3^x \frac{dz}{\ln(U)(z-1)(z-2)} =$$

$$= \log_U\left(2 - \frac{2}{x-1}\right),$$

$$Q_\tau(x) = \int_3^x \frac{dz}{\ln(V)(z-1)} = \log_V(0.5x - 0.5),$$

$$Q_r^{-1}(x) = 1 + \frac{2}{2 - U^x},$$

$$Q_t^{-1}(x) = 1 + 2V^x.$$

Таким образом, мажорантно-суперпозиционный алгоритм генерации дискретной случайной величины, подчиняющейся робастному распределению солитона, представлен на рис. 2, состоящем из трёх частей.

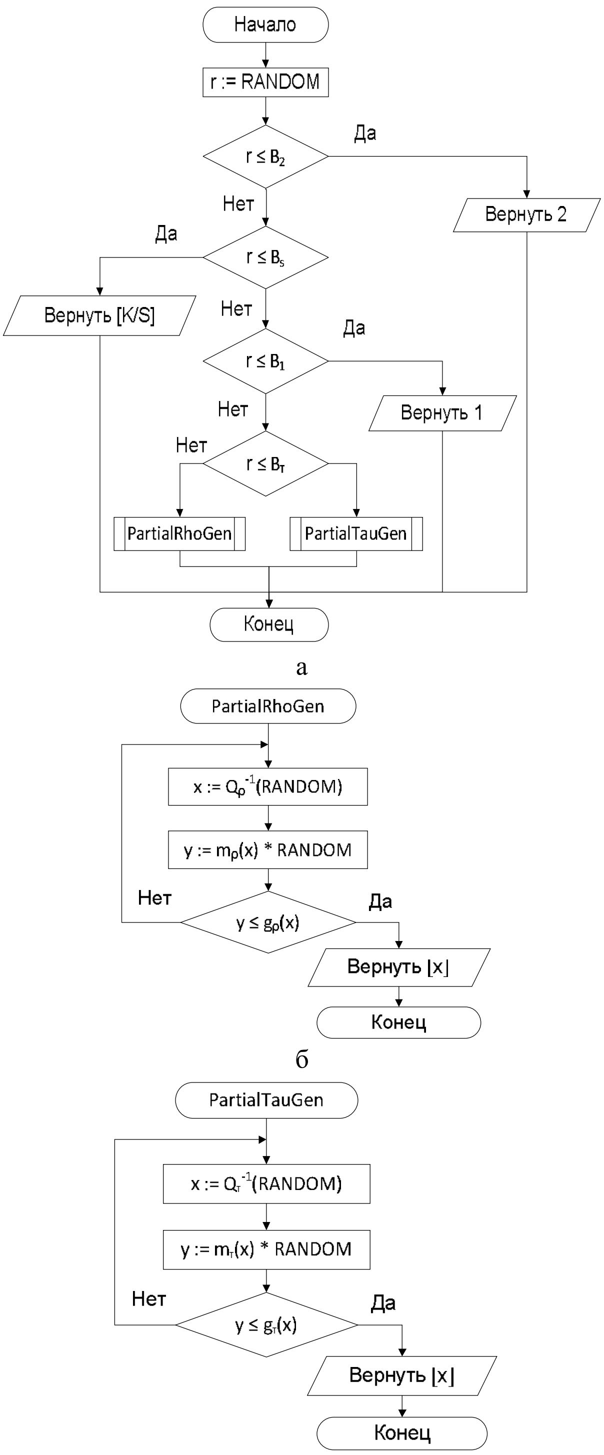

Рис. 2. Мажорантно-суперпозиционный алгоритм

4. Сравнение производительности алгоритмов

Сравнение производительности мажорантно-суперпозиционного алгоритма генерации дискретной случайной величины, подчиняющейся робастному распределению солитона, и стандартного алгоритма генерации дискретной случайной величины, применяемого для данного распределения, было выполнено путём измерения времени генерации 10^7 случайных значений при помощи реализаций данных алгоритмов на языке C# для заданных значений параметра K при фиксированных значениях параметров c = 0.2 и δ = 0.05. В качестве реализации стандартного алгоритма была использована реализация генератора робастного распределения солитона в рамках проекта по разработке кодера и декодера кодов LT на языке C# с открытым исходным кодом [4]. Измерения осуществлялись на компьютере с процессором Intel Core i5-2430M 2.4GHz. На основании полученных измерений был построен график, представленный на рис. 3.

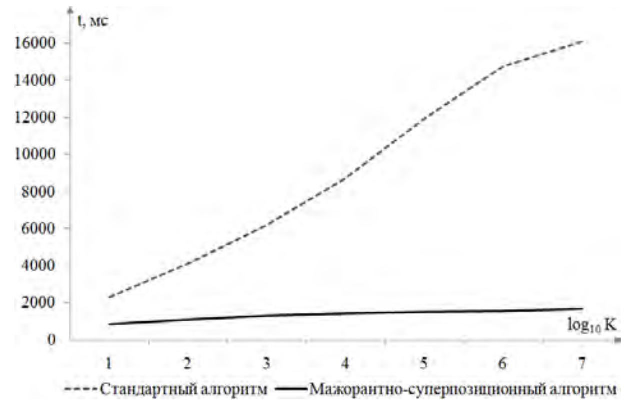

Рис. 3. График зависимости времени генерации 10^7 случайных значений от параметра K для сравниваемых алгоритмов

Из графика следует, что мажорантно-суперпозиционный алгоритм в сравнении со стандартным алгоритмом имеет более высокую производительность для любых значений параметра K и демонстрирует более низкие темпы роста времени выполнения при увеличении значения параметра K.

Заключение

Мажорантно-суперпозиционный алгоритм генерации дискретной случайной величины, подчиняющейся робастному распределению солитона, совмещает в себе метод суперпозиции, мажорантный метод исключения, метод обратного преобразования и стандартный алгоритм генерации дискретной случайной величины. В сравнении со стандартным алгоритмом мажорантно-суперпозиционный алгоритм имеет преимущество как в отношении производительности при любых значениях параметра K, так и в отношении темпов роста времени выполнения при увеличении значения параметра K. Предлагаемый алгоритм может быть использован в программной реализации кодера кодов LT.

Список литературы

1. Жураковський Б.Ю. Дослідження використання нових завадостійких кодів для каналів зі стиранням / Б.Ю. Жураковський [Электронный ресурс] // Вісник Державного університету інформаційно-комунікаційних технологій. – 2012. – Т. 10, № 2. – С. 93-96. – Режим доступа : http://nbuv.gov.ua/j-pdf/vduikt_2012_10_2_18.pdf.

2. MacKay D.J.C. Fountain Codes [Электронный ресурс]/ D.J.C. MacKay // IEEE Proc.-Commun. – Vol. 152 – No. 6, December 2005. – Режим доступа : https://www.ma.utexas.edu/users/voloch/Homework/MacKay05.pdf.

3. Нохрина Г.Л. Имитационное моделирование экономических процессов [Электронный ресурс] / Г.Л. Нохрина. – 2013. – Режим доступа : http://simulation.su/uploads/files/default/2013-kurs-lection-nohrina.pdf.

4. Soliton.cs [Электронный ресурс]. – Режим доступа : https /github.com/GusBricker/FountainCodes/blob/master/LubyTransform/Distributions/Soliton.cs.

5. Дополнительные сведения о численном моделировании случайных элементов. Учеб. пос. [Электронный ресурс] / А.В. Войтишек – 2007. – Режим доступа к ресурсу : http://mmf.nsu.ru/sites/default/files/voytishekav-advaced-topics-on-numerical-modelling-of-stochastic-elements.pdf.

Поступила в редколлегию 23.12.2015

Рецензент: д-р техн. наук, проф. А.В. Горбенко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

МАЖОРАНТНО-СУПЕРПОЗИЦІЙНИЙ АЛГОРИТМ ГЕНЕРАЦІЇ ВИПАДКОВОЇ ВЕЛИЧИНИ, ЩО МАЄ РОБАСТНИЙ РОЗПОДІЛ СОЛІТОНА

О.С. Вамболь

У статті запропоновано мажорантно-суперпозиційний алгоритм генерації дискретної випадкової величини, що має робастний розподіл солітона, що суміщає в собі метод суперпозиції, мажорантний метод виключення, метод зворотного перетворення і стандартний алгоритм генерації дискретної випадкової величини. У порівнянні зі стандартним алгоритмом пропонований алгоритм має перевагу щодо продуктивності. Мажорантно-суперпозиційний алгоритм може бути використаний в програмній реалізації кодера кодів LT.

Ключові слова: *мажорантно-суперпозиційний алгоритм, робастний розподіл солітона, фонтанні коди, коди LT, генерація випадкових величин, підвищення продуктивності.*

MAJORANT-SUPERPOSITION ALGORITHM FOR GENERATING A RANDOM VARIABLE THAT HAS A ROBUST SOLITON DISTRIBUTION

A.S. Vambol

The majorant-superposition algorithm for generating a discrete random variable that has a robust soliton distribution which combines the method of superposition, majorant elimination method, inverse transform sampling and a standard algorithm for generating a discrete random variable is proposed in the paper. The proposed algorithm has an advantage in terms of performance in comparison with a standard algorithm. The majorant-superposition algorithm can be used in a software implementation of the LT codes encoder.

Keywords: *majorant-superposition algorithm, robust soliton distribution, fountain codes, LT codes, generating random variables, performance enhancement.*