Научная статья
УДК 63.009.34
DOI: 10.14529/ctcr240105

АЛГОРИТМЫ ФОРМИРОВАНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ В НАТУРНОМ ЭКСПЕРИМЕНТЕ ПРИ ИМИТАЦИИ ПРОЦЕССОВ ФУНКЦИОНИРОВАНИЯ СЛОЖНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

С.А. Баркалов, *sbarkalov@vgasu.vrn.ru*
Е.А. Серебрякова, *sea-parish@mail.ru*
К.А. Нижегородов, *upr_stroy_kaf@vgasu.vrn.ru*
Воронежский государственный технический университет, Воронеж, Россия

Аннотация. В данной работе рассматривается информационная система – ее структура и факторы воздействия внешней среды, что позволяет описывать ее имитационными средствами. Элементами для описания подобных систем являются алгебраические дифференциальные уравнения, цепи Маркова или аналогичные им. **Цель исследования** заключается в формировании условий проведения натурного эксперимента не на исследуемой системе, а на некоторой другой (с известными параметрами распределения исследуемой случайной величины, например, закона распределения), достаточно «похожей» на данную и такой, экспериментальное функционирование которой не приведет к сколько-нибудь ощутимым издержкам. **Методы исследования.** При имитации процессов Маркова очень часто встречаются трудности, связанные с необходимостью ввода в память машины стохастических матриц бесконечной размерности. Фактически очень трудно представить бесконечную стохастическую матрицу, в которой все элементы выбраны «чисто случайно», т. е. без каких бы то ни было закономерностей. На практике всегда случаются только такие матрицы, в которых, начиная с некоторой строки, проявляется определенная закономерность в образовании элементов. На этом свойстве основывается решение указанной проблемы. **Результаты.** Алгоритм нахождения случайной величины заключается в выборе того из высказываний, которое при данном значении является истинным. Затем следующая реализация находится по формуле, соответствующей выбранному высказыванию. Если при этом в выбранной формуле содержится случайная величина, то с помощью уже известных приемов находится реализация этой случайной величины и подставляется в нее. Чтобы убедиться в правильности предлагаемого алгоритма и понять соображения, на основании которых построены высказывания, достаточно подставить все возможные значения случайных величин и сравнить результат с получаемым с помощью матрицы. Рассмотренный логический прием применим также при имитации поведения объектов, более сложных, чем цепь Маркова. Использование его в более сложных ситуациях – составная часть идеи автоматного моделирования информационных систем. **Заключение.** Полученные в работе алгоритмы формирования псевдослучайных чисел для имитации компонентов информационной системы достаточно адекватны, однако их эффективность определяется отношением части площади прямоугольника, расположенной под кривой, ко всей его площади. Если это отношение мало, то значительная часть машинного времени расходуется на непроизводительное получение случайных точек, лежащих под кривой и отбрасываемых в процессе вычислений. Отсюда следует, в частности, что для экспоненциального распределения указанный метод наименее, а для равномерного – наиболее эффективен. На практике таким методом целесообразно пользоваться, когда указанное отношение превышает величину 0,3. В некоторых случаях его можно применять и тогда, когда это отношение больше 0,1.

Ключевые слова: алгоритм, закон распределения, модель, машина, натурный эксперимент, случайное число

Для цитирования: Баркалов С.А., Серебрякова Е.А., Нижегородов К.А. Алгоритмы формирования последовательности псевдослучайных чисел в натурном эксперименте при имитации процессов функционирования сложных информационных систем // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». 2024. Т. 24, № 1. С. 52–62. DOI: 10.14529/ctcr240105

Original article
DOI: 10.14529/ctcr240105

ALGORITHMS OF FORMING OF THE SEQUENCE OF PSEUDORANDOM NUMBERS IN THE NATURAL EXPERIMENT AT SIMULATION OF PROCESSES OF FUNCTIONING OF COMPLEX INFORMATION SYSTEMS

S.A. Barkalov, *sbarkalov@vgasu.vrn.ru*
E.A. Serebryakova, *sea-parish@mail.ru*
K.A. Nizhegorodov, *upr_stroy_kaf@vgasu.vrn.ru*
Voronezh State Technical University, Voronezh, Russia

Abstract. In this work is considered information systems: its structure and influencing factors of the external environment that allows to describe its simulation means. Elements for the description of similar systems are algebraic differential equations, Markov chains or similar of the **Research objective** consists in forming of conditions of carrying out a natural experiment not on the studied system, and on some other (with the known distribution parameters of the studied random variable, for example, of the distribution law), rather "similar" to given and such which experimental functioning will not lead to a little notable costs. **Research methods.** At simulation of processes of Markov the difficulties connected with need of input in memory of the machine of stochastic matrixes of infinite dimension very often meet. It is actually very difficult to present an infinite stochastic matrix in which all elements are selected "purely accidentally", i.e. without any patterns. In practice there are always only such matrixes in which, since some line, a certain pattern in formation of elements is shown. The solution of the specified problem is based on this property. **Results.** The algorithm of finding of a random variable consists in the choice of that from expressions which at this value is true. Then the following implementation is on the formula corresponding to the selected expression. If at the same time the selected formula contains a random variable, then by means of already known receptions there is implementation of this random variable and is substituted in it. To be convinced of correctness of the offered algorithm and to understand reasons on the basis of which expressions are constructed it is enough to substitute all possible values of random variables and to compare result to received by means of a matrix. The considered logical reception is applicable also at simulation of behavior of the objects, more difficult, than a Markov chain. Its use in more difficult situations – a component of the idea of automatic modeling of information systems. **Conclusion.** The algorithms of forming of pseudorandom numbers received in work for simulation of components of an information system are rather adequate, however their efficiency is defined by the relation of a part of the square of a rectangle located under a curve to all its area. If this relation is not enough, then a considerable part of machine time is spent for unproductive receiving the accidental points lying under a curve and discarded in the course of calculations. From here follows, in particular, that for exponential distribution the specified method least, and for uniform is most effective. In practice it is reasonable to use such method when the specified relation exceeds value 0.3. In certain cases it can be applied and when this relation more than 0.1.

Keywords: algorithm, distribution law, model, machine, natural experiment, random number

For citation: Barkalov S.A., Serebryakova E.A., Nizhegorodov K.A. Algorithms of forming of the sequence of pseudorandom numbers in the natural experiment at simulation of processes of functioning of complex information systems. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*. 2024;24(1):52–62. (In Russ.) DOI: 10.14529/ctcr240105

Введение

Предположим, что все составные части некоторой информационной системы – ее структуры и факторы воздействия внешней среды – имеют математическую природу. В этом случае они являются абстрактными объектами или же структурами со свойствами объекта. В этом случае следует считать такую систему моделируемой имитационными средствами. Элементами для описания подобных систем являются алгебраические дифференциальные уравнения, цепи Маркова или аналогичные им. Имитационное моделирование придает понятию сложности рассматриваемой системы более глубокий смысл. К примеру, из двух моделируемых систем более сложной предпочтительнее считать ту, в которой для её описания при поддержке некоторого алгоритмического языка потребуется больше символов уникода. Само собой разумеется, что вводимое таким способом определение понятия сложности системы зависит от выбора алгоритмического языка, который избыточен. Если подобные системы S_1 и S_2 различны по своей природе, например S_2 математическая, а S_1 информационная система, то математическую систему S_2 будем называть математической моделью информационной системы S_1.

Во многих случаях расчеты методом имитационного моделирования могут привести к существенным неточностям. Это объясняется в основном недостаточной достоверностью данных, используемых в расчетах, а также тем, что применяемый при вычислениях простейший алгоритм не учитывает некоторых закономерностей, имеющих место в действительности. Если полученные (с допущенными неточностями) данные принять за истинные и использовать при проектировании информационной системы, то после запуска система будет функционировать далеко не оптимально. Поэтому одни узлы сети будут работать с повышенной нагрузкой, другие, наоборот, с недостаточной. Для совершенствования системы вновь потребуются количественные расчеты, которые также невозможно выполнить идеально точно. Весь процесс повторится сначала, и система постепенно приведется к такому режиму функционирования и такой структуре, которые в данных условиях будут обеспечивать достаточно эффективную ее работу [1].

Описанный процесс является ни чем иным, как регулированием системы с помощью натурного эксперимента. Оптимизация системы с помощью натурного эксперимента – очень дорогостоящее мероприятие. Она может продолжаться довольно длительное время, в течение которого система будет работать убыточно. Кроме того, в ряде случаев натурный эксперимент вообще невозможен (например, когда его проведение может привести к катастрофическому ущербу или человеческим жертвам).

Тогда целесообразно провести эксперимент не на самой системе, а на некоторой другой, достаточно «похожей» на данную и такой, экспериментальное функционирование которой не приведет к сколько-нибудь ощутимым издержкам [2].

Постановка задачи

Рассмотрим простейшие случаи синтеза имитационных моделей процессов информационного взаимодействия агентов сложных комбинированных информационных систем управления на основе моделирования случайных воздействий. Эта часть метода моделирования является очень важной составляющей практически в любой статистике для формирования адекватных датасетов с последующим машинным обучением [3].

Моделируемая система подвержена случайным воздействиям. Они могут носить разнообразный характер: от отдельных взаимно не зависящих величин до взаимосвязанных и конечных количеств или некоторого произвольного процесса (например, цепи Маркова). В первую очередь, при конструировании автоматных моделей важным является первый из всех перечисленных случаев. Существует несколько приемов для имитации совокупности взаимосвязанных случайных величин, это воспроизведение их значений по заданной матрице и поиск похожих элементов.

Рассмотрим способы и механизмы генерации случайных чисел с заданным законом распределения. Эти числовые последовательности являются эмпирическими и их вероятностное распределение в некотором смысле отличается от определенного теоретического распределителя [4].

Все случайные величины в теории вероятностей можно разделить на дискреты, непрерывную и смешанную. Только определенные отдельные значения имеют дискретные величины. Множество допустимых значений дискретной случайный величины может быть либо конечным, либо счетным. Разумеется, нередки случаи, когда случайная величина принимает только целочисленные значения. Также двоичной называется случайная величина с двумя значениями: 0 и 1.

Дискретная случайная величина задается с помощью распределения вероятностей p_k ($k = 0, 1, 2, ...$) того, что она примет значение, равное k.

Перечислим некоторые наиболее распространенные типы распределений дискретных случайных величин.

1. Распределение Бернулли (для двоичной случайной величины):

$p_0 = 1 - p;\ p_1 = p;\ p_k = 0$ при $k > 1$.

Здесь через p обозначен единственный в данном случае параметр распределения ($0 \le p \le 1$).

2. Распределение Пуассона:

$p_k = \frac{\lambda^k}{k!} e^{-\lambda}$ $(0 < \lambda < +\infty; k = 0, 1, 2, ...)$.

3. Биноминальное распределение:

$p_k = C_n^k p^k (1-p)^{n-k-1}$ при $k = \overline{0, n}$,

$p_k = 0$ при $k > n$ $(0 < p < 1; n$ – целое число).

4. Геометрическое распределение:

$p_k = (1-p)^{n-k-1}$ $(k = 0, 1, 2, ...;\ 0 < p < 1)$.

5. Дискретное равномерное распределение:

$p_k = \frac{1}{n}$ при $k = \overline{1, n}$ и $p_k = 0$ при $k > n$ (n – целое число).

Непрерывный случайный интервал имеет возможность принимать значение из бесконечно малого или большого непрерывного промежутка. Непрерывная функция распределения, в свою очередь, составляется на базе вероятности $F\,(x)$. Вероятность значения случайной величины не может быть более 0 процентов. Если говорить о распределениях непрерывных случайных величин, то можно привести следующие примеры [5].

1. Показательное (экспоненциальное) распределение:

$$F\,(x) = \begin{cases} 0 & \text{при } x \le 0, \\ 1 - e^{-\lambda x} & \text{при } x > 0, \end{cases} \quad (0 < \lambda < +\infty).$$

Случайная величина, распределенная по этому закону, может принимать лишь неотрицательные значения.

2. Нормальное распределение:

$$\Phi(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-(z-a)^2}{2\sigma^2}} dz \quad (-\infty < a < +\infty, \sigma \ge 0).$$

Нормально распределенная случайная величина принимает не только положительные, но и отрицательные значения.

3. Равномерное распределение на отрезке $[a, b]$:

$$F\,(x) = \begin{cases} 0 & \text{при } x \le a, \\ \frac{x-a}{b-a} & \text{при } a < x \le b, \\ 1 & \text{при } x > b. \end{cases} \quad (-\infty < a < b < +\infty),$$

4. Усеченное нормальное распределение с точками усечения c_1 и c_2:

$$F\,(x) = \frac{\Phi(x) - \Phi(c_1)}{\Phi(c_2) - \Phi(c_1)},$$

где $\Phi(x)$ – обычная нормальная функция распределения и $-\infty \le c_1 < c_2 \le +\infty$.

При $c_1 = -\infty$ получается одностороннее усечение справа в точке c_2 при $c_2 = +\infty$ – одностороннее усечение слева в точке c_1.

Если одновременно $c_1 = -\infty$, а $c_2 = +\infty$, то усеченное распределение превращается в неусечённое.

Смешанной случайной величиной назовём такую величину, значение которой может быть как изолированным от прочих точек на числовой оси, так и постоянно расположенным в одной из её частей. Рассмотрим две конкретные разновидности смешанных распределений [6].

Смешанное показательное распределение: с вероятностью p_0 случайная величина принимает значение 0, а с вероятностью $p_0 = 1 - p_0$ – значение, совпадающее с одной из реализаций случайной величины, распределенной по показательному закону.

Смешанное усеченное нормальное распределение: с вероятностью p_0 случайная величина принимает значение 0, а с противоположной вероятностью $p_0 = 1 - p_0$ – значение, совпадающее с реализацией случайной величины, имеющей усеченное нормальнее распределение.

Довольно часто приходится иметь дело с ситуациями, когда случайная величина получается при моделировании не по определенному теоретическому распределению определенного вида, а, скорее, эмпирическому представлению, полученному на основе статистических данных [7].

Такие эмпирические распределения могут иметь произвольный характер (асимметричность, многомодальность и т. д.), они не имеют точной степени приближения к некоторым известным теоретическим распределениям или являются достаточно беспорядочными. Тогда указанное эмпирическое распределение может быть введено в машину посредством значений ординат плотности распределения через произвольные временные интервалы или при помощи некоторого уравнения, приближенно воспроизводящего начальную плотность этого определения.

Для получения значения случайной величины с заданным законом распределения чаще всего пользуются одним или несколькими из множества равно распределенных случайных чисел. В связи с этим получение равномерных распределенных случайных чисел на ЭВМ имеет огромное значение. Существует множество способов, как можно с помощью ЭВМ решить проблему имитации неравномерно распределенных случайных величин. Некоторые результаты, на которых стоит заострить внимание, представлены в работе [8].

Другим способом является ввод в память ЭВМ таблиц равномерно распределенных случайных чисел и считывание их по ходу решения. Применение таблиц, в которых равномерно распределенные числа зачисляются во внешнюю память машины, приводит к росту обращений к работающим устройствам и снижению скорости работы ЭВМ. Это увеличивает сложность программы и уменьшает надёжность решения. Также для решения больших задач зачастую необходимы сотни тысяч и даже миллионы случайных чисел, что во много раз больше объема имеющихся в настоящее время таблиц равно распределенных случайным образом чисел [9].

С помощью генераторов случайных чисел можно получить свободно распределенные случайные числа. Эти устройства создают результаты некоторого случайно выполняющегося физического процесса в последовательности двоичных разрядов машины и дают возможность получения реализации разнообразных значений. При систематическом применении метода статистических испытаний на данной ЭВМ, возможно, потребуется создание датчика случайных чисел. Если иного нет – разумнее применять так называемые псевдослучайные последовательности, вырабатываемые электронной вычислительной машиной при поддержке особых подпрограмм.

Данный метод широко распространен в наше время, заключается он в следующем. Программистский способ получения случайных чисел – в ЭВМ используется некоторое рекуррентное соотношение: каждое последующее число происходит из предыдущего (или группы предыдущих) при помощи использования алгоритма, состоящего из арифметических и логических операций. Так, к примеру, простейший арифметический расчет, предложенный Нейманом, заключается в том, что берется некоторое произвольное число, состоящее из двоичных цифр, и возводится равным квадратом [10]. По итогу мы получили число, состоящее из цифр, в котором после каждого последующего выбрасывается по одной цифре с обоих концов (остается «середина произведения»). Вновь данный процесс возведения в квадрат и отбрасывания цифр многократно повторяется, получаемые «середины квадратов» – это ровно распределенные промежутки (частицы случайного происхождения) последовательностей случайных чисел. На практике чаще используются более сложные алгоритмы [11], позволяющие получить распределение лучшего качества и учитывающие особенности ЭВМ. Каждая из этих последовательностей чисел находит соответствие с известными критериями случайной природы, хотя входящие в них числа взаимно зависимы друг от друга.

Такой способ имеет недостатки. В первую очередь, разработанные при поддержке специальных программ последовательности случайных чисел являются периодическими и потому даже очень длинные их комбинации не будут случайными. Увеличение периода последовательности может несколько снизить скорость работы машины. Также распределение выработанных программным способом случайных чисел в большинстве случаев не отличается от теоретического. Таковое расхождение, незначительное в одномерном случае, начинает влиять на формирование из последовательности равномерно распределенных чисел разнообразных многомерных вычислений [3].

В то же время, несмотря на все недостатки метода Монте-Карло, метод псевдослучайных чисел имеет большое распространение среди исследователей. Рассмотрим вероятность получения псевдослучайных чисел, равномерно распределенных на отрезке. Исходными данными для получения равномерно распределенных на отрезке $[0,1]$ псевдослучайных чисел служат два положительных целых числа p и g, причем p должно быть простым и представляемым в виде $p = 2p_1 + 1$, где p_1 – также некоторое положительное простое число.

При этом g должно быть близким к половине числа p и допускать представление в виде $g = p - 3m$, где m – некоторое положительное целое [12].

Если с помощью фигурных скобок обозначить операцию взятия дробной части числа, а с помощью квадратных – операцию выделения целой части, то последовательность равномерно распределенных на отрезке [0,1] случайных чисел $\xi_1, \xi_2, \xi_3, \ldots$ может быть получена по рекуррентным формулам:

$$R_k = r_{k-1}g,\ \xi_k = \left\{{}^{R_k}/_{p}\right\}, \tag{1}$$
$$r_k = R_k - p\left[{}^{R_k}/_{p}\right]\ (k\ =\ 1, 2, 3, \ldots),$$

где R_k и r_k – промежуточные последовательности и $r_0 = 1$.

Полученные таким образом псевдослучайные числа повторяются с периодом $p - 1$.

Числа p и g, служащие исходными данными для работы подпрограммы, выбираются так, чтобы заведомо избежать повторения в ходе решения всей задачи [13].

Последовательность псевдослучайных чисел, распределенных нормально с параметрами a и σ, можно получить способом, основанным на приближенной имитации условий, при которых оказывается справедливой центральная предельная теорема теории вероятностей. В силу центральной предельной теоремы суммы большого числа случайных слагаемых при выполнении некоторых весьма общих условий имеют асимптотически нормальное распределение. Поэтому для приближенного моделирования можно воспользоваться суммированием чисел исходной совокупности, равномерно распределённых на отрезке [0, 1].

Пусть ξ_i $(i = 1, 2, \ldots)$ – псевдослучайные числа, равномерно распределенные на отрезке [0,1] с математическим ожиданием 0,5 и средним квадратическим отклонением $\frac{1}{2\sqrt{3}}$ [4].

Тогда при достаточно больших m сумма $\eta = \sum_{i=1}^{m} \xi_i$ распределена асимптотически нормально с параметрами $a' = \frac{m}{2}$; $\sigma' = \frac{1}{2}\sqrt{\frac{m}{3}}$, а псевдослучайные числа, распределенные нормально с заданными параметрами a и σ, могут быть найдены по формуле

$$\eta_k = \frac{\sum_{i=1}^{m} \xi_i - \frac{m}{2}}{\frac{1}{2}\sqrt{\frac{m}{3}}}\sigma + a = \left(\frac{2}{\sqrt{m}}\sum_{i=1}^{m} \xi_i - \sqrt{m}\right)\sigma\sqrt{3} + a. \tag{2}$$

Для получения последовательности псевдослучайных чисел, соответствующих показательному закону распределения, используется формула

$$x = -\frac{1}{\lambda}\ln(1 - \xi_i),$$

где λ – параметр распределения; ξ_i – последовательность случайных чисел, равномерно распределенных на отрезке.

Этот способ [14] широко применяется на практике, так как не приводит к загрузке оперативной памяти машины.

При решении практических задач часто приходится оперировать случайными величинами, распределенными по некоторым специальным законам. Приведенные ниже распределения будут полезны при реализации отдельных моделей.

1. Смешанное усеченное нормальное распределение: случайная величина ξ с вероятностью $p_0 (0 \le p_0 \le 1)$ принимает значение 0, а с вероятностью $(1 - p_0)$ – значение, совпадающее с одной из реализаций случайной величины η, распределенной по усеченному нормальному закону:

$$F(x) = P\{\eta \le x\} = \begin{cases} 0 & \text{при } x < c_1, \\ \frac{\Phi(x) - \Phi(c_1)}{\Phi(c_2) - \Phi(c_1)} & \text{при } c_1 \le x < c_2, \\ 1 & \text{при } x \ge c_2, \end{cases} \tag{3}$$

где $\Phi(x) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{(x-a)^2}{2\sigma^2}}dx$ – функция обычного (неусеченного) нормального распределения с математическим ожиданием a $(-\infty < a < +\infty)$ и дисперсией σ^2 $(0 \le \sigma < +\infty)$; c_1 и c_2 – соответственно левая и правая точки усечения.

Таким образом, смешанное усеченное нормальное распределение определяется с помощью пяти параметров: p_0, c_1, c_2, a, σ.

В частном случае, когда распределение не является смешанным, следует положить $p_0 = 0$.

В других частных случаях, когда усечение является односторонним слева, справа или вообще отсутствует, надо принять соответственно:

$$c_2 = k_2,\ c_1 = k_1,\ \begin{cases} c_1 = k_1, \\ c_2 = k_2, \end{cases} \tag{4}$$

где в качестве k_1 берется некоторая достаточно большая по абсолютной величине отрицательная константа; k_2 – достаточно большая положительная константа.

Конкретный выбор k_1 и k_2 осуществляется исходя из разрядности ЭВМ.

2. Смешанное показательное распределение: случайная величина ξ с вероятностью p_0 $(0 \le p_0 \le 1)$ принимает значение 0, а с вероятностью $(1 - p_0)$ совпадает с одной из реализаций случайной величины η, распределенной по показательному закону:

$$G(x) = P\{\eta \le x\} = \begin{cases} 0 & \text{при } x \le 0, \\ 1 - e^{-\lambda x} & \text{при } x > 0, \end{cases} \quad (0 \le \lambda < +\infty). \tag{5}$$

Параметрами смешанного показательного распределения являются два числа: p_0 и λ. Частный случай, когда распределение не является смешанным, осуществляется аналогично изложенному выше.

3. Конечное дискретное распределение: определяется с помощью вероятностей $p_k = P\{\xi = k\}$ $(k = \overline{0, n}; n$ – некоторое фиксированное положительное целое; $0 \le p_k \le 1$; $\sum_{k=0}^{n} p_k = 1)$. Задание этого распределения в общем случае осуществляется с помощью $n + 1$ числа p_k $(k = \overline{0, n})$.

Чтобы получить случайные числа, распределенные по одному из трех перечисленных законов, используемых в дальнейшем при решении сложных экономических систем, составляют специальную подпрограмму псевдослучайных чисел [14].

Основным блоком подпрограммы является блок получения равномерно распределенных на отрезке [0, 1] случайных чисел. Остальные блоки подпрограммы используют результаты работы основного блока и осуществляют преобразование равномерно распределенных случайных чисел в случайные числа, распределенные по требуемому закону.

Псевдослучайные числа, распределенные по смешанному усеченному нормальному закону, получаются с помощью центральной предельной теоремы.

Получение псевдослучайных чисел, распределенных по показательному закону, осуществляется по формуле

$$x = -\frac{1}{\lambda}\ln(1 - \xi_i) \quad (i = 1, 2, \dots). \tag{6}$$

Для получения псевдослучайных чисел, распределенных по дискретному закону, использовался метод последовательного сравнения очередной реализации равномерно распределенного на отрезке [0,1] случайного числа с последовательностью сумм $S_k = \sum_{i=0}^{K} p_i$ $(k = \overline{0, n})$.

Алгоритм указанного метода можно представить следующим образом.

Шаг 1. Формирование суммы $S_0 = p_0$.

Шаг 2. Выборка равномерно распределенного на отрезке [0, 1] случайного числа ξ.

Шаг 3. Сравнение ξ со значением текущей суммы S_k. Если $\xi < S_k$, то производится фиксация $\xi = k$ и осуществляется выход из подпрограммы. В противном случае производится формирование очередной суммы $S_{k+1} = S_k + p_{k+1}$ и переход к выполнению шага 2. Так как $\xi \le 1$ и $\sum_{k=0}^{n} p_k = 1$, то во всех случаях процесс должен закончиться фиксацией некоторого значения x и выходом из подпрограммы.

Подпрограмма состоит из номеров ячеек, в которые помещаются исходные параметры и номера ячейки для передачи управления на некоторый ее вход. При использовании подпрограммы требуется пересылка необходимых значений параметров в отведенные для них ячейки, информации о передаче управления после выполнения операции выборки и номеров ячеек.

Данная подпрограмма занимается получением случайных чисел, которые используются при моделировании случайной величины процессов для решения многих экономических задач. Имея вспомогательные и случайные элементы, она даёт возможность создать произвольное множество чисел или случайных процессов для моделирования достаточно обширного перечня задач.

Опишем алгоритм получения псевдослучайных чисел, распределенных в соответствии с некоторой произвольной плотностью вероятностей $f(x)$, которая может быть определена на основании статистических данных [8].

Для функции $f(x)$ как плотности некоторого распределения справедливо соотношение $\int_{-\infty}^{+\infty} f(x)\,dx = 1$, поэтому она должна быть или сосредоточенной на конечном промежутке, или асимптотически приближаться к горизонтальной оси для достаточно больших и достаточно малых значений x.

С большей или меньшей погрешностью можно указать такой прямоугольник с вершинами $(a, 0), (a, c), (b, c), (b, 0)$, что вся кривая плотности распределения $f(x)$, за исключением, может быть, отбрасываемых близких к горизонтальной оси асимптотических участков, будет заключена внутри этого прямоугольника.

Тогда алгоритм получения случайных чисел может быть представлен в следующей форме.

Шаг 1. Находятся два случайных числа α_1 и α_2, каждое из которых распределено на отрезке [0, 1].

Шаг 2. Найденные случайные числа преобразуются в координаты случайной точки, равномерно распределенной в прямоугольнике с вершинами $(a, 0), (a, c), (b, c), (b, 0)$, т. е. $\beta_1 = a + \alpha_1(b - a)$; $\beta_2 = \alpha_2 c$.

Шаг 3. Проверяется положение полученной случайной точки относительно графика функции $f(x)$; если точка расположена выше кривой, т. е. если $\beta_2 > f(\beta_1)$, то все действия повторяются снова (с шагом 1), в противном случае величина β_1 принимается за очередную реализацию моделируемой случайной величины.

Частоты попадания значений получаемых случайных чисел в интервалы отрезка $[a, b]$ будут пропорциональны площадям криволинейных трапеций, расположенных над этими отрезками под кривой $f(x)$, следовательно, функция $f(x)$ будет плотностью распределения.

При имитации процессов Маркова очень часто встречаются трудности, связанные с необходимостью ввода в память машины стохастических матриц бесконечной размерности. Фактически очень трудно представить бесконечную стохастическую матрицу, в которой все элементы выбраны «чисто случайно», т. е. без каких бы то ни было закономерностей.

На практике всегда случаются только такие матрицы, в которых, начиная с некоторой строки, проявляется определенная закономерность в образовании элементов. На этом свойстве основывается решение указанной проблемы.

Предположим, что рассматриваемая простая однородная цепь Маркова имеет матрицу вероятностей перехода следующего вида (табл. 1).

Таблица 1

Матрица вероятностей перехода

Table 1

Matrix of transition probabilities

	0	1	2	3	4	…
0	p_0	p_1	p_2	p_3	p_4	…
1	p_0	p_1	p_2	p_3	p_4	…
2	0	1	0	0	0	…
3	$q_0 + q_1$	q_2	q_3	q_4	q_5	…
4	q_0	q_1	q_2	q_3	q_4	…
5	0	q_0	q_1	q_2	q_3	…
6	0	0	q_0	q_1	q_2	…
…	…	…	…	…	…	…

Примечание:
p_i и q_k – такие числа, что $0 \le p_i \le 1$; $0 \le q_k \le 1$,
$(i = 0, 1, 2, \dots,\ k = 0, 1, 2, \dots)$;
$\sum_{k=0}^{\infty} p_i = 1$ и $\sum_{k=0}^{\infty} q_k = 1$.

Задача заключается в построении алгоритма нахождения реализации случайной величины $a\,(t + 1)$ по известной реализации случайной величины $a\,(t)$.

Обозначим через ξ и η дискретные взаимно независимые случайные величины, вероятностными распределениями которых служат соответственно последовательности $\{\, p_i\}$ и $\{\, q_k\}$. Ввести в машину эти последовательности не составляет трудности.

Построим следующую (конечную) полную систему логических высказываний и соответствующих этим высказываниям формул (табл. 2).

Таблица 2

Система логических высказываний и формул

Table 2

System of logical expressions and formulas

Высказывание	Формула
$a\,(t) < 2$	$a\,(t\,+\,1) = \xi$
$a\,(t) = 2$	$a\,(t\,+\,1) = 1$
$a\,(t) > 2$	$a\,(t\,+\,1) = \max\{0, a\,(t) + \eta - 4\}$

Алгоритм нахождения $a\,(t\,+\,1)$ чрезвычайно прост. Вначале выбирается то из высказываний, которое при данном значении $a\,(t)$ является истинным. Затем реализация $a\,(t+1)$ находится по формуле, соответствующей выбранному высказыванию. Если при этом в выбранной формуле содержится случайная величина, то с помощью уже известных приемов находится реализация этой случайной величины и подставляется в формулу. Чтобы убедиться в правильности предлагаемого алгоритма и понять соображения, на основании которых построены высказывания и формулы, достаточно подставить все возможные значения случайных величин $a\,(t)$, ξ и η в формулы и сравнить результат с получаемым с помощью матрицы.

Рассмотренный логический прием применим также при имитации поведения объектов, более сложных, чем цепь Маркова. Использование его в более сложных ситуациях – составная часть идеи автоматного моделирования сложных систем.

Выводы

Описанный алгоритм внешне весьма привлекателен, однако при более подробном рассмотрении оказывается, что он эффективен далеко не всегда. Эффективность характеризуется отношением части площади прямоугольника, расположенной под кривой, ко всей его площади. Если это отношение мало, то значительная часть машинного времени расходуется на непроизводительное получение случайных точек, лежащих под кривой и отбрасываемых в процессе вычислений. Отсюда следует, в частности, что для экспоненциального распределения указанный метод наименее, а для равномерного – наиболее эффективен. На практике таким методом целесообразно пользоваться, когда указанное отношение превышает величину 0,3. В некоторых случаях его можно применять и тогда, когда это отношение больше 0,1.

Список литературы

1. Баркалов С.А., Бурков В.Н., Порядина В.Л. Механизмы активной экспертизы в задачах комплексного оценивания // Вестник Воронежского государственного технического университета. 2009. Т. 5, № 6. С. 64–67.

2. Белоусов В.Е., Абросимов И.П., Губина О.В. Алгоритм идентификации состояний многоуровневой технической системы с использованием расплывчатых категорий модели представления знаний // Вестник ВГУ. Серия: Системный анализ и информационные технологии. № 3. 2017. С. 124–129.

3. Jordan M.I. Attractor dynamics and parallelism in a connectionist sequential machine // The Eighth Annual Conference of the Cognitive Science Society. Amherst, MA, 1986. P. 531–546.

4. Горелик А.Л., Скрипкин В.А. Методы распознавания. М.: Высшая школа. 2004. 341 с.

5. Белоусов В.Е., Баркалов С.А., Нижегородов К.А. Ресурсно-временной анализ в задачах календарного планирования строительных предприятий // Материалы XVI Всероссийской школы-конференции молодых ученых «Управление большими системами». Тамбов: Изд-во ТГТУ, 2019. Т. 1. С. 98–101.

6. Вапник В.Н. Восстановление зависимости по эмпирическим данным. М.: Наука, 1979. 295 с.

7. Osherson D.N., Weinstein S., Stoli M. Modular learning // Computational Neuroscience / E.L. Schwartz (ed.). Cambridge, MA: MIT Press, 1990. P. 369–377.

8. Галинская А.А. Модульные нейронные сети: обзор современного состояния разработок // Математические машины и системы. 2003. № 3-4. С. 87–102.

9. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн; пер. с англ. и ред. И.В. Красикова. 2-е изд. М.: Вильямс, 2005. 1296 с.

10. Моделирование системы оценки компетенций в управлении профессорско-преподавательским составом вуза / С.А. Баркалов, В.Е. Белоусов, Н.Ю. Калинина и др. // XXI Международная конференция по мягким вычислениям и измерениям (SCM-2018): сб. докл. в 2 т. СПб.: СПбГЭТУ «ЛЭТИ», 2018. Т. 1. С. 355–358.

11. Белоусов В.Е., Нижегородов К.А., Соха И.С. Алгоритмы получения упорядоченных правил предпочтения в задачах принятия решений при планировании производственных программ // Управление строительством. 2019. № 1 (14). С. 105–111.

12. Афанасьев В.Н., Юзбашев М.М. Анализ временных рядов и прогнозирование. М.: Финансы и статистика, 2001. С. 203–211.

13. Губко М.В., Караваев А.П. Согласование интересов в матричных структурах управления // Автоматика и телемеханика. 2001. № 10. С. 112–119.

14. Hart O.D., Holmstrom B. Theory of contracts // Advances in economic theory. 5-th World Congress. Cambridge: Cambridge Univ. Press; 1987. P. 71–155.

References

1. Barkalov S.A., Burkov V.N., Porjadina V.L. Mechanisms of active examination in problems complex estimation. *Bulletin of Voronezh state technical university*. 2009;5(6):64–67. (In Russ.)

2. Belousov V.E., Abrosimov I.P., Gubina O.V. [An algorithm of identification of conditions of a multilevel technical system with use of indistinct categories of model of representation of knowledge]. *Proceedings of Voronezh state university. Series: Systems analysis and information technologies*. 2017;3:124–129. (In Russ.)

3. Jordan M.I. Attractor dynamics and parallelism in a connectionist sequential machine. In: *The Eighth Annual Conference of the Cognitive Science Society*. Amherst, MA; 1986. P. 531–546.

4. Gorelik A.L., Skripkin V.A. *Metody raspoznavaniya* [Recognition methods]. Moscow: Vysshaya shkola; 2004. 341 p. (In Russ.)

5. Belousov V.E., Barkalov S.A., Nizhegorodov K.A. [Resource timing analysis in problems of scheduling of the construction enterprises]. In: *Materials of the XVI All-Russian school conference of young scientists "Management of big systems"*. Tambov: TSTU Publ.; 2019. Vol. 1. P. 98–101. (In Russ.)

6. Vapnik V.N. *Vosstanovlenie zavisimosti po empiricheskim dannym* [Recovery of dependence according to empirical data]. Moscow: Nauka; 1979. 295 p. (In Russ.)

7. Osherson D.N., Weinstein S., Stoli M. Modular learning. In: Schwartz E.L. (ed.). Computational Neuroscience. Cambridge, MA: MIT Press; 1990. P. 369–377.

8. Galinskaya A.A. [Modular neural networks: review of the current state of developments]. *Mathematical machines and systems*. 2003;3-4:87–102. (In Russ.)

9. Cormen T.H., Leiserson Ch.E., Rivest R.L., Stein C. *Introduction to algorithms*. Transl. from Engl. 2nd ed. Moscow: Vil'yams; 2005. 1296 p. (In Russ.)

10. Barkalov S.A., Belousov V.E., Kalinina N.Yu., Nasonova T.V., Fomina M.A., Leksashov A.V. [Modeling of a system of assessment of competences of management of the faculty of higher education institution]. *XXI International conference on soft calculations and measurements (SCM-2018)*, in 2 volumes. St. Petersburg: Saint Petersburg Electrotechnical University "LETI"; 2018. Vol. 1. P. 355–358. (In Russ.)

11. Belousov V.E., Nizhegorodov K.A., Soha I.S. Algorithms of obtaining the ordered rules of preference in problems of decision-making when planning production programs. *Upravleniye stroitel'stvom*. 2019;1(14):105–111. (In Russ.)

12. Afanas'ev V.N., Yuzbashev M.M. *Analiz vremennykh ryadov i prognozirovanie* [Analysis of time series and forecasting]. Moscow: Finansy i statistika; 2001. P. 203–211. (In Russ.)

13. Goubko M.V., Karavaeva A.P. Interest Reconciliation in Matrix Control Structures. *Automation and Remote Control*. 2001;62(10): 1658–1672. DOI: 10.1023/A:1012414500272

14. Hart O.D., Holmstrom B. Theory of contracts. In: *Advances in economic theory. 5-th World Congress.* Cambridge: Cambridge Univ. Press; 1987. P. 71–155.

Информация об авторах

Баркалов Сергей Алексеевич, д-р техн. наук, проф., заведующий кафедрой управления, Воронежский государственный технический университет, Воронеж, Россия; sbarkalov@vgasu.vrn.ru.

Серебрякова Елена Анатольевна, канд. экон. наук, доц., доц. кафедры цифровой и отраслевой экономики, Воронежский государственный технический университет, Воронеж, Россия; sea-parish@mail.ru.

Нижегородов Кирилл Александрович, аспирант кафедры управления, Воронежский государственный технический университет, Воронеж, Россия; upr_stroy_kaf@vgasu.vrn.ru.

Information about the authors

Sergey A. Barkalov, Dr. Sci. (Eng.), Prof., Head of the Department of Management, Voronezh State Technical University, Voronezh, Russia; sbarkalov@vgasu.vrn.ru.

Elena A. Serebryakova, Cand. Sci. (Econ.), Ass. Prof., Ass. Prof. of the Department of Digital and Industrial Economics, Voronezh State Technical University, Voronezh, Russia; sea-parish@mail.ru.

Kirill A. Nizhegorodov, Postgraduate student of the Department of Management, Voronezh State Technical University, Voronezh, Russia; upr_stroy_kaf@vgasu.vrn.ru.

Статья поступила в редакцию 30.10.2023
The article was submitted 30.10.2023