УДК 044: 519.711

ГЕНЕРАЦИЯ ПЕРЕСТАНОВОК В СТОХАСТИЧЕСКИХ ГЕНЕРАТОРАХ

Лисицына Е. С., *к.т.н., ст. преподаватель*
Черкасский государственный технологический университет
бул. Шевченко, 460, г. Черкассы, 18006
alyona.lisitsyna@gmail.com

Анотація. *В работе решаются проблемы получения последовательностей случайных чисел, равномерно распределенных на интервале* $\overline{[0,(N-1)]}$, *с периодом повторения, исчисляемым десятками, сотнями или тысячами лет и обладающих стохастическими свойствами на этом интервале. Исследуются методы генерации перестановок в стохастических генераторах, обеспечивающие период повторения последовательности равный N! слов и равномерный закон распределения чисел в этих последовательностях. Рассматриваются методы приближенной реализации периода повторения перестановок к значению N!.*

Ключевые слова: *генератор конгруэнтных чисел, генератор М-последовательности, стохастический цикл, перестановки.*

GENERATION OF PERMUTATIONS IN STOCHASTIC GENERATORS

Lisitsyna O., *Ph.D. (Engineering), senior lecturer*
Cherkassy State Technological University
Shevchenko boul., 460, Cherkassy, 18006
alyona.lisitsyna@gmail.com

Abstract. *The article deals with the problems of obtaining sequences of random numbers that are equally distributed in the interval* $\overline{[0,(N-1)]}$, *with a repetition period that is counted in dozens, hundreds or thousands of years, and have stochastic properties in this interval. The methods of generating permutations in stochastic oscillators that provide the repetition period of the sequence that is equal to N! words, and even the distribution of numbers in these sequences have been researched in the article. Methods of approximate realization of the repetition period of permutations to the value N! have been considered.*

Keywords*: generator of congruent numbers, M-sequence generator, stochastic cycle, permutations.*

Введение. Последовательности случайных чисел широко применяются для решения множества прикладных задач, например, таких как имитационное моделирование, испытания коммуникационных систем, создание компьютерных игр, а так же при решении задач криптографической защиты. При создании генераторов случайных чисел (ГСЧ) важно обеспечить получение заданного закона распределения случайной величины с определенной степенью точности воспроизведения заданного закона. Наиболее распространены генераторы равномерно распределенной на интервале $\overline{[0,(N-1)]}$ случайной величины, поскольку все другие законы распределения дискретной случайной величины получают преобразованием случайных величин с равномерным законом распределения. Наиболее полно требования к последовательности случайных чисел сформулированы в [1, с. 27-38], здесь упомянем основные:

- минимальная (лучше – нулевая) ошибка воспроизведения закона распределения дискретной случайной величины;
- отсутствие корреляции между словами последовательности;
- воспроизводимость – способность точного воспроизведения последовательности при разнесении в пространстве и (или) времени моментов создания/воссозданья;
- непредсказуемость – невозможность восстановления последовательности по ее отрезку.

Строго говоря, не существует способов выполнения данных требований в полном объеме. Так, естественные (природные) источники случайных процессов, например, такие как некоге-

рентное световое излучение солнца или пламени свечи удовлетворяет всем требованиям, кроме требования к воспроизводимости – ни один фрагмент случайного процесса не может быть точно восстановлен в разных точках пространства или в разное время.

Конечные автоматы, обеспечивают воспроизводимость генерируемых случайных последовательностей, и именно это определяет их широкую применимость для решения практических задач, но эти последовательности зачастую имеют значительную ошибку закона распределения дискретной псевдослучайной величины (а именно, от 50 % и более), всегда периодичны и, следовательно, предсказуемы, а слова в них коррелированны между собой.

Предложенные в [2, с. 3–8], решения по созданию стохастических ГСЧ на основе генераторов конгруэнтных чисел и генераторов на регистрах сдвига потенциально обеспечивают возможность генерации равномерно распределенных некоррелированных и непредсказуемых последовательностей случайных чисел с периодом повторения в десятки, сотни и тысячи лет. Основным элементом такого рода генераторов, определяющим его свойства, является блок (устройство) для перестановки (перемешивания, тасовки) натуральной последовательности чисел заданного интервала. Учитывая важность решения задачи выполнения перестановок, в настоящей работе исследуются методы их генерации.

Анализ существующих решений. К настоящему времени математические аспекты реализации перестановок хорошо изучены, созданы механические аналоги для их реализации. Эти устройства называют лототронами и публично используются, например, при розыгрыше лотерей, при жеребьевке спортивных состязаний и в других аналогичных ситуациях. Здесь вопрос периодичности повторения последовательности извлекаемых шаров (при многократном повторении процесса перемешивания и выбора шаров) не представляет интереса и потому не исследован. Задача оценки и обеспечения максимального периода повторения последовательности слов в стохастических генераторах актуальна и определяется следующими соображениями.

Пусть, например, имеется генератор последовательности конгруэнтных чисел, вычисляющий каждое последующее число (слово) по его предшествующему значению:

$$S(n) = \left| KS(n-1) + C \right|_M \qquad (1)$$

где: *K*, *C*, *M* – параметры генератора, $S(n)$ и $S(n-1)$ – слова, порожденные в текущий – «*n*» – и в предшествующий – «*n-1*» – дискретные моменты времени.

Пусть, например, параметры K, C, M подобраны так, что генератор порождает циклически повторяемую последовательность случайных чисел интервала $\overline{[0,(M-1)]}$, при этом каждое слово этого интервала применяется в цикле ровно по одному разу. В этом случае генератор формирует периодически повторяемую (с периодом T=M) равномерно распределенную последовательность чисел, при этом ошибка воспроизведения закона распределения случайной величины, при объеме выборки кратном M, равна нулю. Недостаток этого генератора заключается в том, что последовательность слов предсказуема (предсказуемость определена уравнением (1)), а период повторения последовательности мал и равен M. В стохастическом генераторе [2], вычисляется не $S(n) = f[S(n-1)]$, а $S(n) = f[S(0+t)]$,т.е. слово, смещенное на некоторое случайное t, относительно, наперед заданного слова $S(0)$, при этом $t \in \overline{[0,(M-1)]}$ и каждое из значений t в процессе формирования цикла (стохастического цикла из M слов) применяется только по одному разу. Этот процесс полностью соответствует механической модели лототрона, в который загружают M пронумерованных цифрами 0, 1, 2, 3, 4…(M-1) шаров, которые перемешивают и поочередно извлекают, пока не будет извлечен последний шар. Так получают равномерно распределенную случайную последовательность чисел интервала $\overline{[0,(M-1)]}$, непредсказуемую и некоррелированную, которую (в электронной модели) будем называть стохастическим циклом. После завершения формирования первого стохастического цикла шары вновь загружают, перемешивают и вновь извлекают их случайным выбором. Так формируется каждый стохастический цикл до завершения работы лототрона. Вот здесь и возникает вопрос, а чему будет равен период повторения стохастических циклов, каким образом его максимизировать и как выполнить его электронную модель.

Постановка задачи. Задачей исследования является определение периода повторения стохастических циклов, путей его максимизации, создание генератора параметра t для стохастических генераторов.

Решение задачи. Прежде всего, отметим, что по определению параметр t в каждом стохастическом цикле пробегает все значения интервала $\overline{[0,(M-1)]}$ и каждое его значение применяется только по одному разу. Отсюда следует, что:

- случайная величина t равномерно распределена на интервале $\overline{[0,(M-1)]}$, ошибка воспроизведения закона распределения случайной величины t равна нулю;
- максимальное число разных последовательностей слов, носителей параметра t (число разных стохастических циклов) равно M!;
- период повторения стохастических циклов быстро растет с ростом размерности слов t, или, что то же самое, с ростом размерности интервала определения случайной величины.

В частности для стохастического генератора с $M=2^8=256$ (генератора с разрядностью слова m=8) период повторения стохастических циклов составит $T=256!=8{,}5\cdot10^{505}$, а для стохастического генератора с $M=2^{16}=65536$(генератора с разрядностью слова m=16) период повторения стохастических циклов будет равен $T=65536!=5{,}16\cdot10^{287193}$. Пусть эту последовательность производит многотысячная (например, десятитысячная) группировка высокопроизводительных ЭВМ, например, производительностью 10^{10} слов/сек. Учтем, что год содержит, примерно, $3{,}15\cdot10^7$ сек. Тогда период повторения стохастических циклов генератора с M=256 будет равен $T_{лет}=\frac{8{,}5\cdot10^{505}}{10^4\cdot10^{10}\cdot3{,}15\cdot10^7}=2{,}7\cdot10^{484}\,лет$, а стохастический генератор с M=65536 будет иметь период повторения равный $T_{лет}=\frac{5{,}16\cdot10^{287193}}{10^4\cdot10^{10}\cdot3{,}15\cdot10^7}=1{,}6\cdot10^{287172}\,лет$.

Такое значение периода повторения физически невозможно зафиксировать. Это, в свою очередь, говорит о том, что можно допустить приближенную реализацию процедуры перебора при условии, если этот период выходит за границы физически реализуемой оценки.

Очевидным, также, является то, что для создания стохастического генератора случайной величины «X», равномерно распределенной на интервале $\overline{[0,(M-1)]}$, необходим стохастический генератор случайной величины $t\in\overline{[0,(M-1)]}$. Степень сложности создания стохастического генератора случайной величины t, эквивалента степени сложности создания стохастического генератора случайной величины X. Решение одной из этих задач, автоматически исключает необходимость решения другой задачи. В силу этого обстоятельства будем искать детерминированные процедуры реализации перестановок случайной величины t или псевдослучайные (псевдостохастические) процедуры, т.е. процедуры, сочетающие фрагменты детерминированных и псевдослучайных процедур.

Для решения задачи в такой постановке, прежде всего, рассмотрим некоторые свойства перестановок и методы их генерации.

1. Рекурсивный метод генерации перестановок

Известно, что

$$(m+1)!=(m+1)m! \tag{2}$$

Из этого следует рекурсивный метод генерации перестановок:

- пусть m=2 и представляет множество из двух чисел-1 и 2, тогда $m!=2!=2$, это перестановки 12 и 21. Запомним их.
- пусть m=3, множество чисел – 1, 2, 3 и 3!=3x2!=6 .

Вновь появившееся слово 3 может в перестановках стоять на одном из трех мест (быть первым, вторым или третьим), остальные две позиции занимают известные перестановки из двух слов (12, 21). Тогда повторим 6 раз перестановки из двух слов, а у каждой пары ставим слово 3 на первую, вторую и третью позицию. Получим 312, 321, 132, 231, 123, 213. Запомним их.

Пусть m=4, множество чисел – 1, 2, 3, 4 и 4!=4x3!=24. Вновь появившееся слово 4 может в перестановках стоять на одном из четырех мест (быть первым, вторым, третьим или четвертым), остальные три позиции занимают известные перестановки из трех слов (312, 321, 132,

231, 123, 213). Тогда алгоритм получения перестановок 4! следующий: повторим 24 раза перестановки из трех слов, а у каждой группы из 6 слов ставим слово 4 на первую, вторую, третью и четвертую позицию. В результате получим:

Таблица 1

4312	4321	4132	4231	4123	4213
3412	3421	1432	2431	1423	2413
3142	3241	1342	2341	1243	2143
3124	3214	1324	2314	1234	2134

Рекурсивное продолжение процедуры вычисления перестановок по выражению (2) позволяет строить таблицы перестановок любой размерности, но с ростом размерности перестановки быстро растет размер памяти, необходимый для хранения результирующих перестановок. Размер этой памяти должен быть равен m!, что для m=8 дает 8!= 40 320 слов (что требует 40 Кбайт памяти), для m=9 дает 9!=362 880 слов (что требует 363 Кбайт памяти), для m=10 дает 10!= 3628800 слов (что требует 3,63 Мбайт памяти). В целом, данный метод годится для алфавитов малой мощности.

2. Метод перестановок на основе слов вдвое меньшей размерности

Вычислим:

$$(2n)! = 2n \cdot (2n-1) \cdot (2n-2) \cdot (2n-1-3) \cdots (2n-(n-1)) \cdot n! \qquad (3)$$

Учтем, что $C_{2n}^{n} = \dfrac{2n!}{(2n-n)! \cdot n!} = \dfrac{2n \cdot (2n-1) \cdot (2n-2) \cdot (2n-1-3) \cdots (2n-(n-1)) \cdot n!}{(n!)^2}$,

отсюда

$$(2n)! = (n!)^2 \cdot C_{2n}^{n} \qquad (4)$$

Теперь сконструируем алгоритм вычисления перестановок из 8 слов, по известным таблицам перестановок 4 слов (табл. 1). Увеличение размера слова до 8 бит будем производить конкатенацией двух полуслов по 4 бита каждое.

Представим 8! как 24х24х70. Первые два сомножителя (24х24) определяют поочередный перебор перестановок из табл.1, а третий сомножитель C_8^4 – это число слов веса 4 в последовательности 8 разрядных слов или, что тоже самое, восьмиразрядное слово из двух четырехразрядных может быть образовано 70 способами. Для реализации этой части 1 процедуры вычисления 8! ведем восьмиразрядный двоичный счетчик, который после перебора всех24х24=576 перестановок добавляет «+1» к состоянию счетчика, после чего проверяется вес (wtA) слова А в счетчике. Производится проверка wtA= 4?, если нет – добавляем «+1» в счетчик и снова проверяем его состояние, так продолжается до тех пор пока не получим wtA=4. Тогда единичные разряды числа А определяют первое четырехразрядное полуслово, а остальные (нулевые разряды) второе полуслово восьмиразрядного слова. Например, слово $10=2^3+2^1$ в счетчике имеет вес wtA≠4 и должно игнорироваться, число $15=2^3+2^2+2^1+2^0$ имеет вес 4 и говорит, что первое полуслово конкатенации образуется 1, 2, 3, 4 разрядами, а второе полуслово – 5, 6, 7, 8 разрядами, а слово $23=2^4+2^2+2^1+2^0$ говорит, что первое полуслово конкатенации образуется 1, 2, 3, 5 разрядами, а второе полуслово – 4, 6, 7, 8 разрядами.

С учетом этого получим простой алгоритм:

1) перебирая из табл. 1 два разных словосочетания из четырех слов им расставив их по разрядам слова конкатенации (в соответствии со значением разрядов слова А веса 4) выполним 576 перестановок. Затем, выбрав из счетчика следующее слово веса 4, производим следующие 576 перестановок, при этом переставляя разряды слова, полученного путем конкатенации, в соответствии с весами единичных и нулевых разрядов следующее слово веса 4. В результате получим стохастическую последовательность с периодом повторения 8!, что и свидетельствует о достижении поставленной цели.

3. Синтез таблиц перестановки большой размерности на основе перестановок малой размерности

Сформулируем задачу в следующем виде:

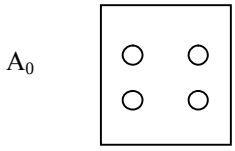
Имея в распоряжении таблицу перестановок 4 слов (табл. 1) построить таблицу перестановок большей размерности (например, 256 слов).

Полученную таблицу перестановок использовать в стохастическом генераторе, порождающем равномерно распределенную последовательность чисел в интервале $\overline{[0.255]}$.

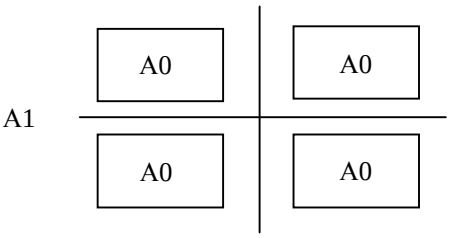
Решение задачи.

Используем для целей построения таблицы большой размерности метод «вложения» множеств.

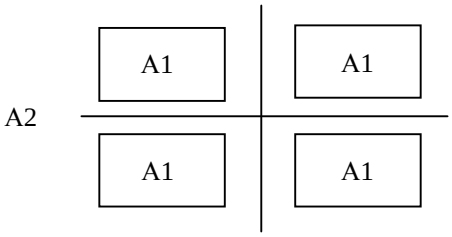
Пусть имеется множество A_0, состоящее из 4 объектов (например, чисел 1, 2, 3, 4)



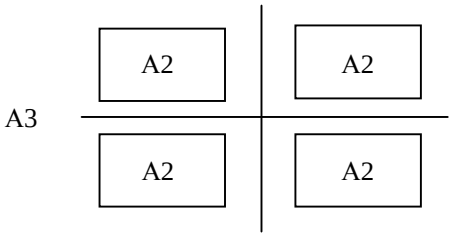
Создадим множество A_1 вложением множестваA_0 в каждый из квадрантов плоскости, получим множество из 16 слов.



Создадим множество A_2 вложением множества A_1 в каждый из квадрантов плоскости, получим множество из 4x16=64 слов.



Создадим множество A_3 вложением множества A_2 в каждый из квадрантов плоскости, получим множество из 4x64=256 слов.



Каждый из объектов множества A_3 может быть определен восьми битами, при этом первая пара бит определяет номер квадранта во множестве A_3, вторая – номер квадранта во множестве A_2, третья пара – номер квадранта во множестве A_1, а четвертая – номер объекта во множестве A_0.

Каждый из объектов множества A_3также может быть определен при записи десятичного номера объекта в четверичной системе счисления. Например, объект 217 может быть определен как $217=3\cdot4^3+1\cdot4^2+2\cdot4^1+1\cdot4^0$. Это значит, что объект 217 находится:

- в 3 квадранте множества A_3;
- в 1 квадранте множества A_2;
- во 2 квадранте множества A_1;
- является объектом 1 множества A_0.

Положим так же, что есть натуральная последовательность чисел 0, 1, 2, 3 … 255, которая хранится в регистре сдвига с циклической перезаписью.

Пусть ее как то перемешали и, выбирая из нее слово за словом, загружали в генератор конгруэнтных чисел, как параметр t для вычисления слова S(0+t). Это слово выводится на выход устройства, как конечный продукт работы генератора. После того, как сформировали стохастический цикл (т.е. вывели 256 сгенерированных слов) последовательность слов в регистре сдвига надо как то перемешать, после чего ее снова можно использовать для генерации следующего стохастического цикла. Перемешивание блока из 256 слов можно производить разными способами, в том числе по частям, как показано в [3]. В результате тасовки здесь получают некоторое трехзначное десятичное число, представление которого в четверичной форме определяет используемые перестановки табл.1. Общее число полученных перестановок и, соответственно, период повторения стохастической последовательности является приближенным значением числа 256!. Точное значение числа перестановок, существенно зависит от выбранных параметров декоррелятора.

В целом, следует отметить, что можно предложить множество способов разложения числа n! на сомножители и тем более его приближенного значения. Каждый из этих сомножителей может быть физически реализован, следовательно, может быть реализовано либо приближенное, либо точное значение числа перестановок равное n!.

Полученные результаты. Проведенное исследование определяет способ построение таблиц перестановок для стохастических генераторов, обеспечивающие получение приближенного или точного значения периода повторения стохастических циклов равное n!.

Выводы. Решенная задача генерации таблиц перестановок для стохастических генераторов решает задачу построения генераторов равномерно распределенных случайных чисел, удовлетворяющих всем поставленным здесь требованиям, последовательностей с периодом повторения равной n!, где n – разрядность генерируемого слова.

Список литературы

1. Бараш Л. Генерация случайных чисел и параллельных потоков случайных чисел для расчетов Монте-Карло // Безопасность информационных технологий. – 2005, № 2. – С. 27–38.
2. Береза А. С. Генерация конгруэнтных последовательностей чисел с наперед заданными свойствами / Береза А. С., Лавданский А. А., Швыдкий В. В., Фауре Э. В. / Вісник ЧДТУ – 2012, № 2. – С. 3–8.
3. Пат. 40649 Україна, МПК G 06 F 7/58. Пристрій декореляції випадкової послідовності чисел / заявник та патентовласник ЧДТУ – № u200811384; заявл. 22.09.2008; опубл. 27.04.2009, Бюл. № 8. Мітянкіна Т. В., Швидкий В. В., Мітянкін М. О.

References

1. Barash L. Generation of random numbers and parallel streams of random numbers for Monte-Carlo calculation / / Safety of information technology. – 2005, № 2. – P. 27–38.
2. Bereza A., Lavdanskiy A., Shvidkiy V., Faure E., Generation of congruent number sequences with prescribed properties // Bulletin CDTU – 2012, № 2. – P. 3–8.
3. Pat. 40649 Ukraine, IPC G 06 F 7/58. The device decorrelation random sequence of numbers / applicant and patentee CDTU – № u200811384; appl. 22.09.2008, publ. 27.04.2009, Bull. #8. Mityankina T., Shvidkiy V., Mityankin M.

Стаття надійшла до редакції 23.04.2013.

Відомості про автора:

Лісіцина О. С., кандидат технічних наук, старший викладач, Черкаський державний технологічний університет