УДК 511.215

АНАЛИЗ НЕКОТОРЫХ АЛГОРИТМОВ ПОЛУЧЕНИЯ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

З.И.Маслова, доц.; В.А.Цыбульник, студ.; С.В.Кохан, студ.

Случайно выбранные числа оказываются полезными для самых различных целей: в имитационном моделировании, задачах оптимизации, испытания эффективности различных алгоритмов для вычислительных машин, в теории и стратегии игр. Поэтому в математике и программировании актуальной является задача получения случайных последовательностей.

Раньше учёные, нуждавшиеся в случайных числах, раскладывали карты, бросали кости, вытаскивали шары из урны, составляли специальные таблицы, содержащие случайные цифры, взятые, например, из переписи населения, конструировали специальные машины,

механически вырабатывающие случайные числа. Но все эти методы оказались непригодными при решении вышеперечисленных задач при помощи ЭВМ, что и побудило интерес к получению случайных чисел с помощью арифметических операций вычислительной машины. Для этого были разработаны специальные алгоритмы, которые получали следующий член «случайной» последовательности путём определённого преобразования предыдущего. Метод вызывает очевидное возражение: как может быть случайной последовательность, каждый член которой определён предшественником, а, значит, вся она определена первым членом. Дело в том, что такая последовательность не случайна, а выглядит как случайная, ведь в типичных приложениях обычно не имеет значения, как связаны друг с другом два последующих числа последовательности. В научно-технической литературе последовательности, вырабатываемые такими способами, называются псевдослучайными или квазислучайными.

Первым, кто разработал алгоритм получения псевдослучайных чисел, был Джон фон Нейман, предложивший в 1946 году метод «середины квадрата» [1]. Идея заключалась в том, что предыдущее число возводится в квадрат, а затем из результата извлекаются средние цифры. Этот метод довольно хорошо «перемешивает» предыдущее число, но он является довольно скудным источником случайных последовательностей, поскольку выработанные им последовательности имеют тенденцию превращаться в короткие циклы повторяющихся элементов или вообще вырождаться в ноль. Многие учёные проводили эксперименты с методом середины квадрата, научились контролировать «зацикливание» последовательности, но всё же в данный момент этот метод является одним из наименее эффективных.

Интересны также последовательность Фибоначчи $X_{n+1}=(X_n+X_{n-1})modm$ (хотя тесты показывают, что получаемые последовательности недостаточно случайны) и аддитивный датчик $X_{n+k}=(X_n+X_k)modm$, где k - достаточно большое число, предложенное Грином, Смитом и Клемом [2] (несмотря на кажущееся неудобство в использовании, существует очень эффективная процедура реализации этого датчика, хотя сейчас о нём ещё известно довольно мало).

Наилучшие же из известных сегодня датчиков случайных чисел представляют собой частные случаи следующей схемы, предложенной в 1948 году Лемером. Выберем 4 числа:

X_o - начальное значение, $X_o \geq 0$;

a - множитель, $a \geq 0$;

c - приращение, $c>0$;

m - модуль, $m>X_o$, $m>a$, $m>c$.

Также для удобства вводят $b=a-1$.

Искомая последовательность случайных чисел получается из соотношения $X_{n+1}=(a \cdot X_n+c)modm$, $n \geq 0$ и называется линейной конгруэнтной последовательностью [2]. Выделяют также мультипликативный конгруэнтный метод и смешанный конгруэнтный метод для обозначения линейных конгруэнтных методов с $c=0$ и $c \neq 0$ соответственно.

Такая последовательность в действительности не всегда случайна при произвольном выборе X_o, a, c и m, кроме того, конгруэнтные последовательности всегда «зацикливаются», т.е. в конце концов числа образуют цикл, который повторяется бесконечное число раз. Это свойственно всем последовательностям, имеющим вид $X_{n+1}=f(X_n)$. Повторяющийся цикл называется периодом.

Соотношение $X_{n+k}=(a^k \cdot X_n+(a^k-1) \cdot c/b)modm$, $k \geq 0$, $n \geq 0$, позволяет

выразить $(n+k)$-й член через n-й член последовательности.

Исследуем некоторые принципы выбора значений X_0, a, c, m.

1 Число X_0 может быть произвольным. Если по программе делается несколько просчётов и каждый раз желателен различный источник случайных чисел, присвойте X_0 последнее значение X, полученное во время предыдущего просчёта, или (если это удобно) установите X_0 равным текущему моменту времени и календарной дате.

2 Число m должно быть велико. Удобно выбрать его равным размеру слова вычислительной машины, поскольку при этом эффективно вычисляется $(aX+c) mod m$. Значение $(aX+c) mod m$ следует вычислять точно, без ошибок округления.

3 Если m представляет собой степень двойки, a выбирается так, чтобы $a \bmod 8 = 5$. Если m есть степень 10, то желательно $a \bmod 200 = 21$. При таком выборе величины a, при условии, что c выбрано правильно, гарантируется, что датчик даст все m возможных различных значений X прежде, чем они начнут повторяться и, кроме того, гарантируется высокая «мощность». Мощность линейной конгруэнтной последовательности с максимальным периодом определяется как наименьшее целое число s, такое, что $b^s \equiv 0 (mod\ m)$.

4 Множитель a должен превосходить величину $\sqrt{m}$, желательно, чтобы он был больше $m/100$, но меньше $m-\sqrt{m}$. Последовательность разрядов в десятичном или двоичном представлении a не должна иметь регулярного вида.

5 Постоянная c должна быть равна нечётному числу, когда m есть степень двойки, и не должна быть кратна 5, когда m есть степень 10.

6 Менее значимые первые разряды X не очень хороши с точки зрения случайности, поэтому при использовании числа X основную роль должны играть наиболее значимые разряды. Вообще говоря, лучше рассматривать X как случайную дробь X/m в интервале между 0 и 1, т.е. представить X с десятичной точкой слева, чем как случайное число, расположенное между 0 и $m-1$.

Таким образом, основной задачей является получение последовательностей, похожих на случайные. Получение достаточно большого периода, при котором в конкретных практических задачах последовательность не будет повторяться, еще не означает, что последовательность пригодна для работы. Как же решить, достаточно ли случайна последовательность? В этом нам помогут статистические тесты.

Самым распространенным тестом является критерий χ^2. Он используется как сам по себе, так и как основная часть многих других тестов.

Сформулируем критерий χ^2 в общем виде. Предположим, что все возможные результаты разделены на k категорий. Проводится n независимых испытаний. Пусть Ps - вероятность того, что результат испытаний попадает в категорию s, и пусть Ys - фактическое число попаданий результата в категорию s, тогда $V=\sum_{1\le s\le k}(Ys-n\cdot Ps)^2/(n\cdot Ps)$. Число n должно быть достаточно велико, при котором любые $n\cdot Ps\ge 5$ [3]. «Расшифровать» полученное значение V можно по таблице [2], выбрав строку с соответствующей степенью свободы v, где $v=k-1$, но общий принцип таков: плохи те последовательности, в которых V слишком мало или слишком велико.

Проверив с помощью критерия χ^2 результаты работы линейного конгруэнтного алгоритма, вычислив V и воспользовавшись таблицей, получаем, что некоторые датчики весьма удовлетворительны, некоторые

находятся «на грани», т.е. такая последовательность может получиться в 5% случаев при действительно случайном выборе, а некоторые определённо не прошли испытаний (менее 1% случаев). Этот тест показывает, что с помощью линейного конгруэнтного алгоритма действительно можно получить весьма удовлетворительные последовательности, но нужно быть очень внимательным при подборе значений параметров и придерживаться всех высказанных рекомендаций.

Хотя описанные здесь линейные конгруэнтные последовательности обычно являются удовлетворительным источником случайных чисел, их главное преимущество не в этом. Существуют методы, дающие большую независимость случайных чисел, но линейный конгруэнтный метод даёт очень хорошую возможность достаточно гибко регулировать характеристики получаемой последовательности самому.

В результате проведенного анализа разработана программа, дающая дополнительную практическую помощь при выборе параметров. Введя их значения, можно сразу получить длину периода полученной последовательности и значение χ^2 для неё. Вместе с представленными общими рекомендациями она позволяет наиболее точно и безошибочно определить характеристики конкретной последовательности, облегчая таким образом подбор параметров для реализации той или иной задачи.

SUMMARY

In the article the basic principles of generation of pceudo-casual sequences were considered and the analysis of some most wide-spread algorithms were carried out. The recommendations for an effective use of a linear kongruent method were gave, and the developed tests allow to pick up the demanded characteristics of a received sequence.

СПИСОК ЛИТЕРАТУРЫ

1. Кнут Д. Искусство программирования для ЭВМ. Т. 1. Основные алгоритмы. - М.: Мир, 1976.
2. Кнут Д. Искусство программирования для ЭВМ. Т. 2. Получисленные алгоритмы. - М.: Мир, 1977.
3. Эрдеш П. , Спенсер Дж. Вероятностные методы в комбинаторике. - М.: Мир, 1976.

Поступила в редколлегию 23 сентября 1998 г.