

ИССЛЕДОВАНИЕ ГЕНЕРАТОРОВ СЛУЧАЙНЫХ ЧИСЛОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ ЗАДАЧ ИЗМЕРЕНИЙ

Приведены результаты исследования генераторов равномерно распределенных псевдослучайных числовых последовательностей по быстродействию и функциональному критерию. Для оценки соответствия генерируемой числовой последовательности равномерному закону распределения применен шахматный алгоритм ее прореживания.

Генераторы случайных числовых последовательностей (ГСЧП) применяются при решении многих инженерных и исследовательских задач. Практическая реализация ГСЧП основана на двух основных способах формирования псевдослучайных и случайных числовых последовательностей (СЧП): схемотехническими алгоритмическом [1].

В генераторах, построенных по первому способу (например, на основе генераторов шума [2]), применяются электронные компоненты с реально протекающими в них случайными процессами и поэтому длина получаемой на выходе ГСЧП числовой последовательности теоретически бесконечно велика. Однако быстродействие таких генераторов ограничено возможностями современных аналого-цифровых преобразователей, применяемых для получения отсчетов случайных напряжений. Кроме того, закон распределения случайных чисел будет полностью соответствовать закону распределения измеряемой случайной величины. Поэтому во многих случаях предпочтение отдают второму способу, который позволяет формировать псевдослучайные числовые последовательности заданной длины с заданным законом распределения. При этом быстродействие ГСЧП повышается за счет того, что, во-первых, не требуется выполнять измерения каких-либо физических величин и, во-вторых, имеется возможность, изменяя алгоритм, управлять скоростью работы ГСЧП, что имеет принципиальное значение в системах реального времени, в частности, предназначенных для решения измерительных задач. При этом возникает вопрос: какому алгоритму отдать предпочтение при решении конкретной задачи?

В статье приведены результаты исследования ГСЧП как с известными алгоритмами формирования числовых последовательностей так и ГСЧП, входящих в состав программного продукта Borland Delphi, алгоритм функционирования которых неизвестен. Сравнение ГСЧП проводилось по двум основным критериям: функциональному и критерию быстродействия.

Применяя метод статистических испытаний к решению измерительных задач, необходимо учитывать, что допустимая погрешность модели не может превышать 1/10 от общей погрешности [3]. Это означает, что для правильного выбора ГСЧП необходимо оценивать как соответствие генерируемой числовой последовательности заданному закону распределения (функциональный критерий), так и погрешность модели, которая зависит от общего числа элементов последовательности. Большой интерес представляют ГСЧП с равномерным законом распределения. И если равномерность распределения оценивается сравнительно просто по частотам вхождения элементов последовательности в соответствующие частичные интервалы, то оценка случайности распределения является намного более сложной задачей. Существует несколько эмпирических критериев, позволяющих ответить на вопрос, является ли данная числовая последовательность случайной [1]. Эти критерии используют различные методы выявления статистической зависимости между элементами числовой последовательности фиксированной длины. Общим недостатком таких критериев является их сложность и большой объем требуемых вычислений.

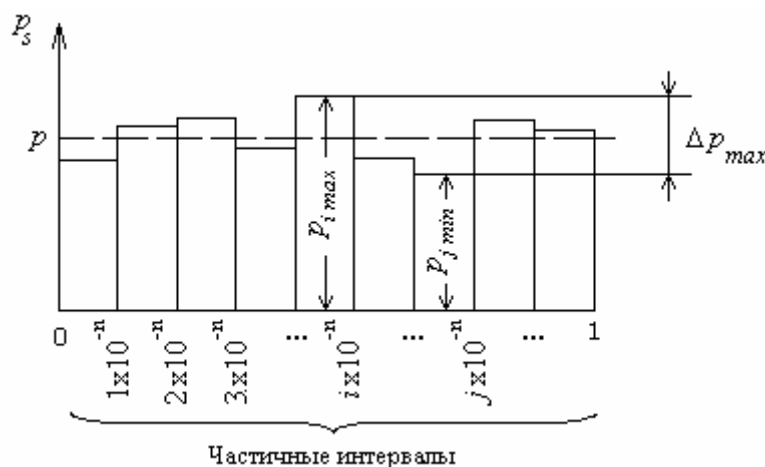
Принципиально отличным от известных эмпирических критериев является рассматриваемый метод прореживания исследуемой СЧП. В основе этого метода лежит следующая гипотеза: *если исходная числовая последовательность распределена равномерно и носит случайный характер, то после ее неслучайного прореживания и, возможно, неслучайных попарных перестановок отдельных элементов, сохраняется равномерность распределения и случайный характер частной последовательности, полученной из исходной.*

Очевидно, что если в результате многократного применения метода прореживания к исходной числовой последовательности выяснится, что каждая частная последовательность, полученная из предыдущей, распределена равномерно, то исходная последовательность также распределена равномерно и носит случайный характер. Таким образом, метод прореживания позволяет прямо оценивать равномерность и косвенно случайный характер распределения исходной числовой последовательности.

Для оценки соответствия числовой последовательности заданному закону распределения будем использовать следующее рабочее определение: числовая последовательность длины $l = N$ (N – число элементов последовательности), является случайной на заданном интервале, если при разбиении интервала на частичные интервалы частоты попадания в него изменяются незначительно при изменении длины последовательности. В пределе, при $N \rightarrow \infty$, частота попадания в частичный интервал равна вероятности попадания в соответствующий интервал.

Для оценки погрешности модели ГСЧП будем рассчитывать частоты попадания элементов последовательности (p_i) в частичные интервалы и сравнивать их с теоретическим значением вероятности попадания в заданный интервал. Методику получения оценки погрешности поясняет рисунок 1.

Рисунок 1 – К определению оценки погрешности ГСЧП



Пусть дана СЧП достаточно большой длины и задана погрешность округления $\epsilon = 10^{-n}$ после нормирования на максимальный элемент последовательности. При этом элементы нормированной последовательности принадлежат отрезку $[0;1]$, а число одинаковых по длине частичных интервалов составляет $m = 10^n$. Тогда для равномерного закона распределения получают теоретическое значение вероятности попадания в частичный интервал $p = \frac{1}{10^n}$. Далее рассчитывают частоты p_s ($s = 1, 2, \dots, m$) и находят максимальный разброс частот $\Delta p_{\max} = p_{i\max} - p_{j\min}$, где $p_{i\max}$ и $p_{j\min}$ – соответственно максимальное и минимальное значения частот вхождения элементов нормированной последовательности в частичные интервалы. Оценку погрешности отклонения от равномерного закона распределения (погрешность модели) рассчитывают по формуле

$$\delta p_{\max} = \frac{\Delta p_{\max}}{p} \cdot 100\% . \quad (1)$$

Рассмотрим алгоритм прореживания числовой последовательности длины $\mathbf{l} = k^2$, которую можно представить в виде матрицы

$$A = (a_{ij}) , \quad (i = 1, 2, \dots, k; \quad j = 1, 2, \dots, k) \quad (2)$$

где в ячейку a_{ij} заносится $(k^{i-1} - 1 + j)$ -й член последовательности генерируемой СЧП.

После чего к матрице (2) применяют шахматный алгоритм.

Рассмотрим шахматный алгоритм прореживания на примере квадратной матрицы, содержащей 16 элементов (рисунки 2 и 3).

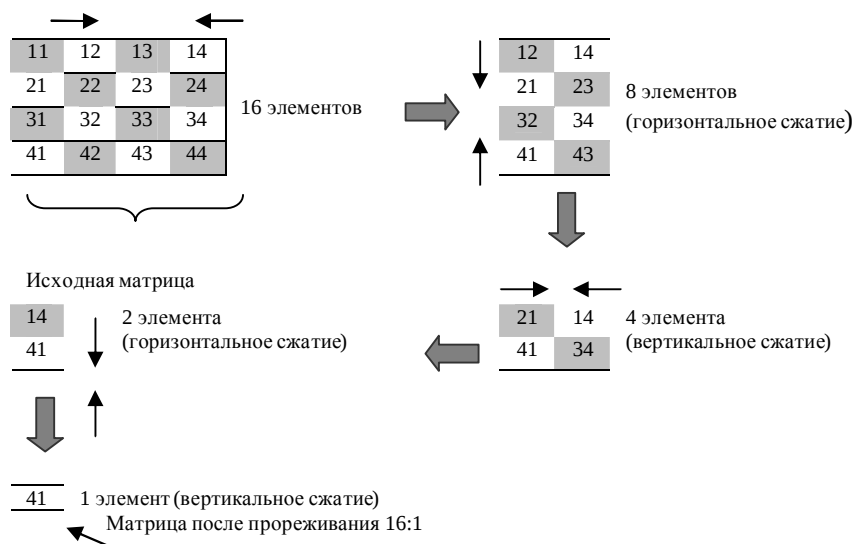


Рисунок 2 – Схема прореживания матрицы 4x4

Затемненные ячейки соответствуют элементам последовательности, которые будут удалены после прореживания. В рассмотренном примере прореживание матрицы производится с чередованием горизонтального и вертикального сжатия. После каждого прореживания общее число элементов частной последовательности уменьшается в два раза. Из рисунка 2 видно, что после второго прореживания произошла попарная перестановка элементов исходной последовательности расположенных в ячейках с номерами 14, 21 и 34, 41.

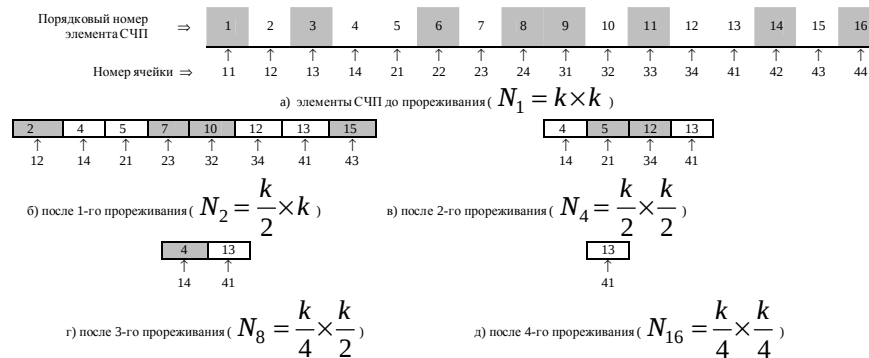


Рисунок 3 – Схема прореживания элементов числовой последовательности ($N_1=16$)

На рисунке 3 изображена схема прореживания исходной числовой последовательности, из которой видно, что прореживание по шахматному алгоритму, хотя и носит неслучайный характер, тем не менее, является более сложной процедурой, чем простое удаление четных или нечетных элементов исходной последовательности.

Для оценки качества генерирования псевдослучайных последовательностей в среде проектирования Delphi 7.0 была создана и отлажена тестовая программа. Разработанное приложение позволяет оценивать соответствие генерируемых ГСЧП массивов случайных чисел заданному закону распределения. В приложении предусмотрено исследование четырех типов генераторов линейного конгруэнтного генератора на основе ряда Фибоначчи с запаздыванием, встроенного в Delphi генератора random, генератора, формирующего числовые последовательности на основе нормального распределения. Учитывая жесткие требования к быстродействию, в качестве основной структуры данных было выбрано бинарное дерево следующего вида:

```
pointer:=^Tree;
Tree=Record
    data : Real;           // значение случайного числа
    freq : integer;        // частота встречаемости
    prob : real;           // вероятность попадания в интервал
    left, right : pointer  // указатель на левое и правое поддерево
end;
...
```

При построении бинарного дерева происходит автоматический подсчет количества вхождений одинаковых элементов в дерево. Далее вызывается процедура обхода дерева, в результате которой подсчитывается вероятность для каждого отдельного элемента и параллельно в отдельном окне выводятся значения полей текущего листа дерева. На быстродействие программы также влияет количество рассматриваемых интервалов, которое пропорционально размеру бинарного дерева.

Первоначально псевдослучайные числа формируются с 16 знаками после запятой, но в программе есть возможность задать определенную точность представления чисел, после нормирования на максимальное сгенерированное число.

Полученные числовые последовательности обрабатываются посредством шахматного алгоритма прореживания:

```
...
c1 := 1; r1 := 1; // индексы новой таблицы
For r:= 1 to SGrid.RowCount-1 do // движение по строкам
    For c := 1 to SGrid.ColCount-1 do // движение по столбцам
        begin // если индекс строки чётный и индекс столбца нечётный или
```

```

// индекс строки нечётный и индекс столбца чётный, то
if ((r mod 2 = 0)and(c mod 2 <> 0))or((r mod 2 <> 0)and(c mod 2 = 0)) then
  begin
    StrGrd.Cells[c1,r1] := SGrid.Cells[c,r]; // переписать текущий элемент в новую таблицу
    inc(c1);
    if c1 mod StrGrd.ColCount = 0 then // преобразование индексов новой таблицы
      begin inc(r1); c1:=1 end // при движении по текущей таблице
    end
  end;
...

```

В программе также есть возможность оценки быстродействия того или иного генератора. Для этого при помощи системной функции GetTickCount в переменную заносится момент времени до запуска процедуры формирования случайных чисел. В цикле через заданное число итераций генерируются новые числа. Затем отмечается текущий момент времени посредством вышеупомянутой функции и рассчитывается время, затраченное на генерацию. Значения выводятся в миллисекундах, что позволяет с приемлемой степенью достоверности оценивать быстродействие исследуемых ГСЧП.

```

begin
  SetLength(aa,temp); // задаём размер временного массива
  start := GetTickCount; // начало отсчёта
  fib_ran_start(round(StrToFloat(EdFstVal.Text)),ran_x); // инициализируем генератор
  i:=0;
  while(i<KK) do
    begin
      aa[i]:=ran_x[i]; // заполняем часть массива числами
      inc(i)
    end;
  while(i<temp) do
    begin
      aa[i]:=(aa[i-KK]-aa[i-LL]) and (MM-1); // заполняем оставшуюся часть массива
      inc(i)
    end;
  vrema := GetTickCount – start // затраченное время
end;

```

На рисунке 4 показана блок-схема прореживания данных в горизонтальном направлении.

С помощью разработанного приложения, установленного на ПК Celeron – 600, были исследованы 4 типа генераторов

А – встроенный в Delphi генератор random;

В – генератор на основе ряда Фибоначчи с запаздыванием [1];

С – линейный конгруэнтный генератор;

Д – генератор, формирующий СЧП на основе нормального распределения.

В таблице 1 приведены результаты исследования ГСЧП четырех типов. Значения погрешностей моделей ГСЧП с начальным значением 12345 получены в результате разбиения отрезка [0; 1], которому принадлежат элементы нормированной СЧП, на десять равных интервалов ($m = 10$). Двадцать и десять процентов отбрасываемых значений соответствуют 80-ти и 90-процентной вероятности того, что погрешность модели не превышает соответствующего числа. В таблице 2 приведены погрешности модели генераторов А и С.

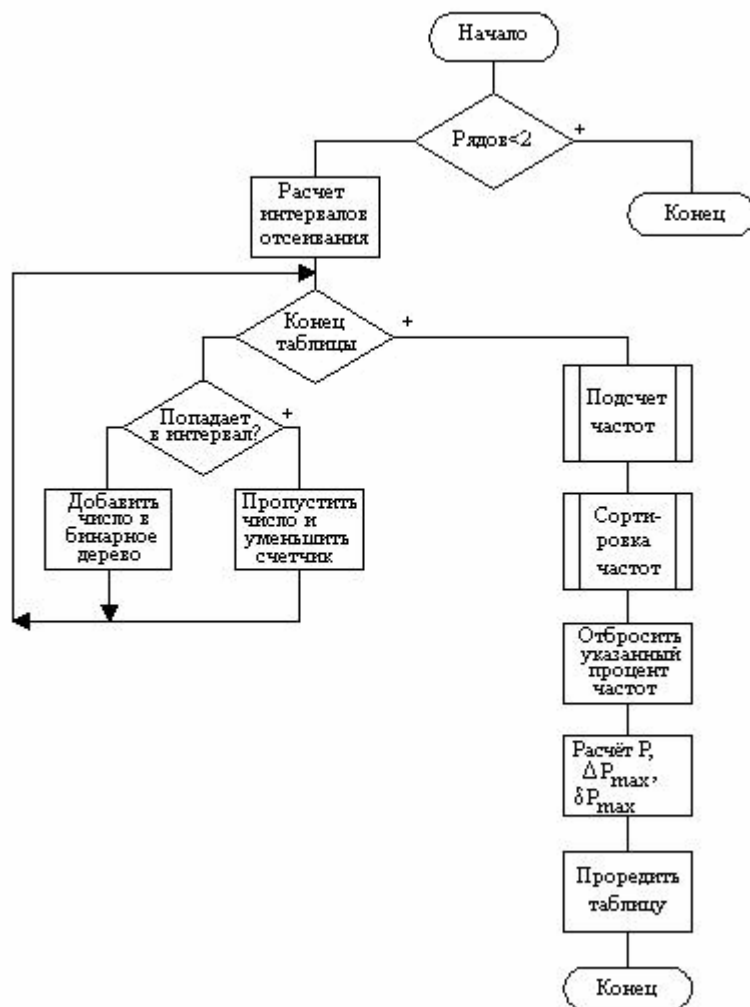


Рисунок 4 – Блок-схема прореживания данных в горизонтальном направлении

Таблица 1 – Результаты исследования ГСЧП

Генератор	Длина СЧП	Время генера- ции,мс	Процент отбра- сывания	Коэффициент прореживания			$\frac{d P_{\max 16}}{d P_{\max 1}}$
				1	4	16	
				Оценка погрешности $d P_{\max}$, %			
A	4194304	97	0	0,39	0,86	1,36	3,5
			10	0,37	0,61	1,24	3,4
			20	0,28	0,31	0,90	3,2
	262144	5	0	2,48	4,65	7,08	2,9
			10	1,42	3,98	6,10	4,3
			20	1,36	2,90	3,36	2,5
B	4194304	145	0	0,33	1,21	2,69	8,2
			10	0,26	1,14	1,90	7,3
			20	0,16	1,13	1,77	11,1
	262144	10	0	2,19	4,65	7,51	3,6
			10	1,96	3,29	4,09	2,1
			20	0,89	3,27	3,78	4,3
C (10 ⁸ ; 2; 0)	4194304	552	0	0,03	0,14	0,50	16,8
			10	0,03	0,08	0,29	11,3
			20	0,02	0,06	0,29	14,5
	262144	33	0	0,44	6,10	8,79	20,0
			10	0,18	5,10	7,14	39,7
			20	0,16	4,75	5,61	35,1

D (10 ⁷ ; 3·10 ⁶)	4194304	2747	0	334,4	334,3	332,7	0,995
			10	273,0	272,6	273,3	1,001
			20	198,0	198,8	198,8	1,004
	262144	170	0	292,8	293,7	293,5	1,002
			10	262,9	262,5	266,1	1,012
			20	185,1	184,6	186,2	1,006

В таблицах 1, 2 в скобках указаны значения модуля множителя и приращения (генератор С), а также среднее значение генерируемых чисел и отклонение от средних значений (генератор D). Коэффициент прореживания – отношение числа элементов исходной СЧП к числу элементов числовой последовательности после прореживания.

Таблица 2 – Результаты исследования генераторов А и С
(оценка погрешности $\delta P_{\max}$, %)

Генератор	Длина СЧП начальное значение	Количество интервалов m	Коэффициент прореживания			$\frac{\delta P_{\max 16}}{\delta P_{\max 1}}$
			1	4	16	
А	<u>4194304</u> 12345	10	0,39	0,86	1,36	3,5
		100	2,47	5,41	12,44	5,0
		1000	9,85	20,69	38,17	3,9
	<u>4194304</u> 56789	10	0,36	1,01	1,60	4,4
		100	2,1	4,32	8,81	4,2
		1000	9,9	20,1	38,9	3,9

Продолжение таблицы 2

С (10 ⁷ ; 2; 0)	<u>4194304</u> 56789	10	0,04	0,19	0,29	7,5
		100	0,37	1,14	2,33	6,3
		1000	3,43	6,01	12,21	3,6
С (10 ⁸ ; 2; 0)	<u>4194304</u> 56789	1000	1,38	6,01	16,02	11,6
С (10 ⁶ ; 2; 0)		10	0,16	0,40	0,82	5,1
С (10 ⁶ ; 113; 0)		10	0,17	0,48	0,63	3,7

Лучшим по быстродействию среди исследованных ГСЧП является генератор А, программа которого написана на Ассемблере. Генераторы В и С работают заметно медленнее, так как их программы написаны на Object Pascal. Наиболее медленным оказался генератор D, хотя его программа также написана на Ассемблере. Это означает, что алгоритм генерации СЧП с нормальным законом распределения сложнее алгоритма генерации равномерно распределенной СЧП. Из таблицы 1 видно, что время генерации СЧП пропорционально числу ее элементов и зависит от типа ГСЧП, а также от быстродействия компьютера.

Разброс погрешности модели зависит от коэффициента прореживания числовой последовательности и типа генератора. Минимальный разброс среди ГСЧП с равномерным распределением имеет генератор А. Большой разброс погрешностей генераторов В и С объясняется тем, что алгоритмы их работы не оптимизированы. Вместе с тем, минимальную погрешность модели обеспечивает генератор С, что подтверждает известные преимущества линейных конгруэнтных генераторов при условии тщательного выбора их параметров. Максимальное значение погрешности среди исследованных ГСЧП имеет генератор D, который вырабатывает СЧП, распределенную по нормальному закону.

Из таблицы 2 видно, что при прореживании исходных СЧП, вырабатываемых генераторами А, В и С, их вероятностные свойства улучшаются. В частности, для генератора А погрешность прореженной в 16 раз исходной СЧП, содержащей 2²² членов, составляет 1,36%, тогда как погрешность этой же СЧП, составленной из первых 2¹⁸ членов равна 2,48%.

Предложенный шахматный алгоритм прореживания СЧП позволяет при решении измерительных задач выбирать ГСЧП, обеспечивающий допустимое значение погрешности модели и необходимую длину числовой последовательности, а также оценивать соответствие числовой последовательности заданному закону распределения.

Задачей дальнейших исследований станет поиск и оптимизация алгоритмов прореживания СЧП произвольной длины, распределенных по заданному закону.

Библиографический список

1. Кнут Д.Э. Искусство программирования. Получисленные алгоритмы / Д.Э. Кнут. — М.: Изд-во «Вильямс», 2001. — 832 с.
2. Тетерич Н.М. Генераторы шума и измерение шумовых характеристик / Н.М. Тетерич. — М.: Энергия, 1968. — 215 с.
3. Брянский Л.Н. Краткий справочник метролога / Л.Н. Брянский, А.С. Дойников. — М.: Изд-во стандартов, 1991. — 79 с.

Поступила в редакцию 07.02.2005 г.

