2 СЛАЙД

Случайные числа

Случайное число играет большую роль в различных задачах и исследованиях. При этом области применения могут быть совершенно разными. Основными являются следующие:

1. Математическое и имитационное моделирование. При моделировании сложных физических, технологических и социально-экономических систем и процессов обойтись без применения источников случайности не представляется возможным.
2. Математическая статистика. Математическая статистика изучает приближенные методы сбора и анализа данных по результатам эксперимента для выявления существующих закономерностей, т. е. нахождение законов распределения случайных величин и их числовых характеристик. Необходимой составляющей выборочных методов является формирование представительных выборок из генеральной совокупности с использованием случайных чисел.
3. Криптография. Криптографические методы являются базовыми в обеспечении информационной безопасности. В криптографии случайные последовательности играют определяющую роль. Они используются, в частности, для получения ключевой последовательности используемого алгоритма шифрования, для генерации гаммы поточных шифров, а также для выработки векторов инициализации (блочных шифров).
4. Другие направления защиты информации. Случайные последовательности незаменимы при формировании паролей и пользовательских ключей (хороший пароль представляет собой короткую последовательность случайных символов). Кроме этого, они могут использоваться для внесения неопределенности в результаты работы различных алгоритмов защиты информации, а также в длительность выполнения шагов алгоритмов для защиты от утечек по побочным каналам. Они также необходимы при формировании случайных запросов при аутентификации и решении многих других задач.
5. Тестирование алгоритмов. Важной задачей является проверка правильности работы программ. Тестирование – достаточно долгий и трудоемкий процесс. Для его осуществления требуется большой объем входных данных. Использование генераторов случайных чисел повышает эффективность тестирования и позволяет экономить время.
6. Сетевые протоколы. Случайные последовательности могут использоваться, например, в качестве сессионных ключей, а также для выработки случайных параметров протокола, что обеспечивает уникальность его различных реализаций.

В любом из этих случаев все начинается со случайного числа. И здесь возникает огромная и сложная задача - задача получения случайного числа. Большое количество научных работ, исследований и экспериментов проведены лишь для того, чтобы дать внятное определение данному понятию. Не будем сильно углубляться в теорию данной задачи, потому что это избыточно в пределах данной работы. Отметим лишь то, что случайные числа, в основном, рассматриваются с точки зрения теории вероятностей и математической статистики. Можно сказать, что случайное число есть часть случайной последовательности. Так намного проще дать определение. Последовательность случайна тогда, когда ни одно ее число нельзя предсказать до генерации. Также числа не должны быть как-либо связано с предыдущими и следующими элементами последовательности и не должны зависеть от них. С точки зрения статистики случайная последовательность должна иметь равномерное распределение. Это, в целом, логично, если вспомнить про вид ее плотности. Отсюда вывод, что получение

абсолютно любого числа в последовательности равновероятно. А раз речь зашла о получении числа, то нужно разобраться с тем, что его может дать.

3 СЛАЙД

Генераторы случайных чисел

И здесь начинается основная информация о генераторах случайных чисел (ГСЧ). Это устройство (программа), которое способно генерировать случайную последовательность чисел. И такая простая (с первого взгляда задача) до сих пор не имеет эталонного решения. Дело в том, что тут проблемой становятся физика и ее законы, время и деньги, принципы работы процессоров и многое другое. Пройдем только по самым базовым ограничениям, не вдаваясь в серьезную теорию. Основная идея является "палкой о двух концах". С одной стороны, мы можем генерировать случайные числа программным способом. Это отличная новость и прекрасная возможность. С другой стороны, сгенерированная последовательность будет либо "не до конца" случайной (об этом будет сказано далее) либо же генерация такой последовательности будет очень долгой и дорогой настолько, что может использоваться в очень ограниченном списке задач. "Не до конца" случайная последовательность на самом деле означает последовательность псевдослучайных чисел. Исходя из этой информации, ГСЧ делятся на ГИСЧ - генераторы истинных случайных чисел, и ГПСЧ - генераторы псевдослучайных чисел. Рассмотрим два типа более подробно.

4 СЛАЙД

Генераторы истинных случайных чисел

ГИСЧ - физическое явление, которое можно описать битовым потоком. Будем говорить, что они, в некотором смысле, генерируют битовый поток. Существуют такие явления, которые доказанно генерируют случайный поток. Такая генерация также называется аппаратной. В данном случае физические явления считываются специальными датчиками и кодируются при помощи компьютера. Этот способ не задействует дополнительные вычислительные ресурсы. Основой для такого генератора может выступать одно из нескольких наиболее распространенных физических явлений:

- Радиоактивный распад атомов
- Дробовой шум - шум в электрических цепях, вызванный дискретностью носителей электрического заряда. Схема такого генератора, основанного на шумах в электронных приборах приведена на слайде:

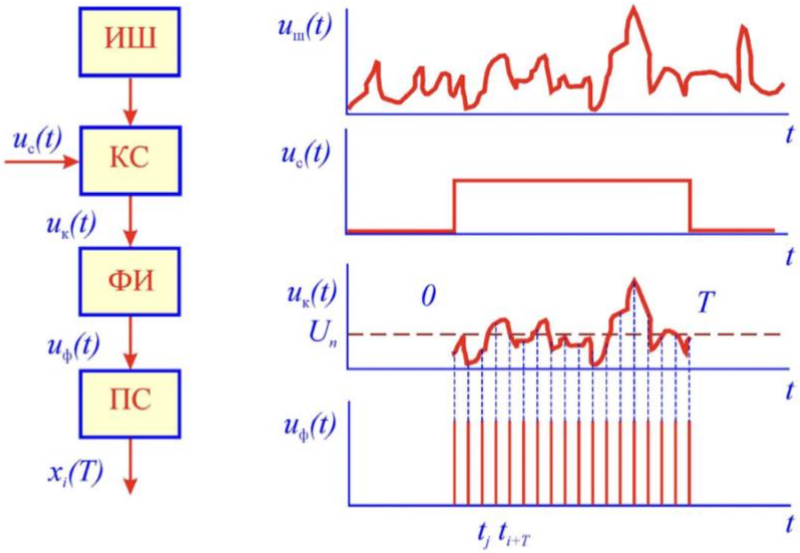


Рисунок 1. Схема ГСЧ, основанного на дробовом шуме

Здесь при усилении шумов на выходе источника шума (ИШ) получается напряжение, которое является случайным процессом. Отрезок реализации данного случайного процесса – ключевая схема (КС) содержит случайное число выбросов. Сравнивая напряжение с пороговым, мы может получить на выходе с формирователя импульсов (ФИ) серию некоторых импульсов. При пересчете в пересчетной схеме (ПС) получаем последовательность случайных чисел. Это были примеры квантовых явлений. Но их сложно фиксировать, поэтому удобнее использовать не квантовые аналоги. Однако, такие явления имеют зависимость от температуры окружающей среды. Ими являются:

- Тепловой шум в резисторе
- Атмосферный шум, измеряемый радиоприемником

В любом случае данные явления не могут обеспечить быструю генерацию случайных последовательностей заданного объема, необходимую в моделировании. Более того, их использование крайне трудно ввиду сложности и высокой стоимости работы.

5 СЛАЙД

Именно поэтому большинство задач использует псевдослучайные последовательности и ГПСЧ. Это так-называемый алгоритмический (программный) способ получения случайных последовательностей. В основе таких генераторов лежат различные алгоритмы, некоторые из которых будут описаны далее. Псевдослучайной называется последовательность, максимально похожая на случайную, но являющаяся детерминированной, подчиняющейся некоторому закону и в пределе зацикливающейся. Все-таки компьютеры не позволяют генерировать бесконечные последовательности, так как ограничены ресурсами и могут находиться в конечном наборе состояний. И все же рассмотрим алгоритмы генерации. С этого момента допустим называть ГПСЧ как ГСЧ ввиду того, что далее рассматриваем только программные способы. В этом месте важно отметить одну из основных характеристик псевдослучайных последовательностей - период - время до начала повторения последовательности.

+ перечислить методы со слайда

Требования к качественному генератору псевдослучайных чисел:

1. Непредсказуемость результатов работы: при неизвестном ключе/начальном состоянии генератора на основе известной конечной части ПСП невозможно определить как ее

последующий элемент (прямая непредсказуемость, или непредсказуемость вправо), так и предыдущий (обратная непредсказуемость, непредсказуемость влево);
2. Неотличимость статистических свойств генерируемых ПСП от аналогичных свойств истинно случайной последовательности;
3. Большой период последовательности;
4. Возможность эффективной аппаратной и программной реализации.

На практике добиться выполнения всех этих условий, как правило, не представляется возможным. Более того, часто эти условия являются взаимоисключающими. Поэтому приходится искать баланс между ними и в первую очередь стремиться к выполнению того, что является наиболее важным в контексте решаемой задачи.

6 СЛАЙД

Алгоритмы генерации псевдослучайных чисел

Одним из наиболее распространенных алгоритмов является линейный конгруэнтный метод. Он лежит в основе ГСЧ большинства языков программирования. Он требует задачи начального значения и последовательность формируется по рекуррентной формуле:

$$X_{n+1} = (aX_n + c)\ mod\ m$$

Здесь m – модуль ($0 < m < 2^{31} - 1$), a – множитель ($0 \leq a \leq m$), c – приращение ($0 \leq c \leq m$), x_0 – начальное значение. 50 итераций этого алгоритма с параметрами c=3, a=2, m=5, x_0=1:

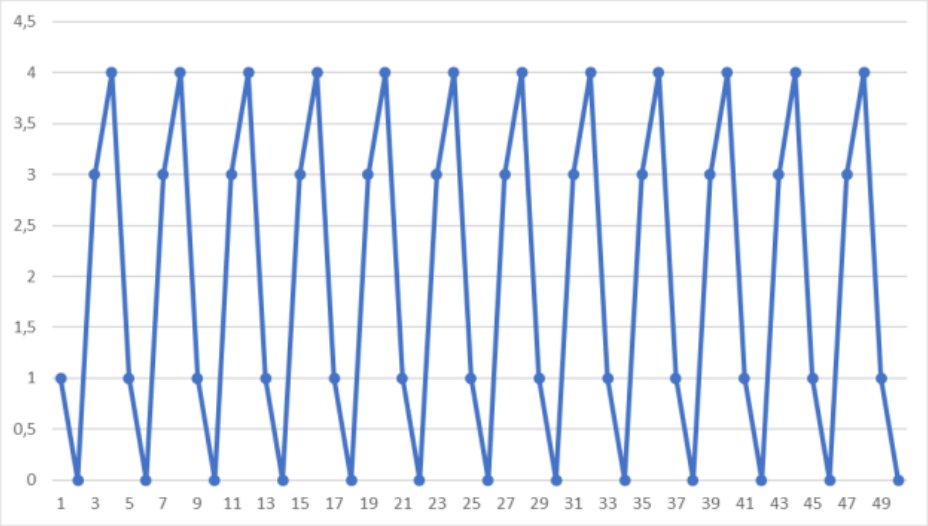


Рисунок 2. Результат работы алгоритма при c=3, a=2, m=5, x_0=1

Видим повторения, попробуем увеличить "уникальную" часть. Для этого модуль (m) должно быть простым числом. Вообще получается интересная ситуация – для "наибольшей случайности", коэффициенты не должны быть случайными. При m = 7877 получаем следующий результат:

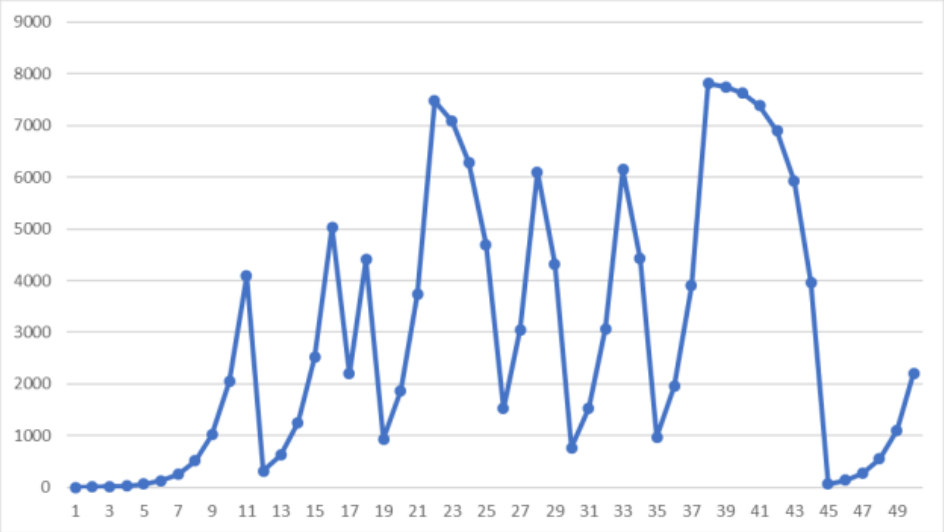


Рисунок 3. Результат работы алгоритма при c=3, a=2, m=7877, x0=1

График больше похож на случайный. К сожалению, данный метод не обделен недостатками. Во-первых - верхний предел периода, он не может превышать машинную длину целого числа ($\max(m) = 2^{31} - 1$). Во-вторых, если говорить о пространстве и случайных векторах, случайные точки, полученные в результате работы алгоритма, имеют размерность меньше, чем исследуемое (моделируемое) пространство.

Частным случаем генератора, основанного на, ЛКГ является генератор Парка-Миллера, имеющий параметры $a = 75$, $b = 0$, $m = 2^{31} - 1$.

7 слайд

Еще одна разновидность ГСЧ, основанная на сдвиге – метод перемешивания. В данном случае используется циклический сдвиг ячейки вправо и влево. Разберем принцип его работы. Пусть в ячейке записано число R_0. Циклически будем сдвигать содержимое ячейки на 1/4 длины ячейки влево, получим число R_0*. Точно такие же операции выполняем и со сдвигом вправо, получаем R_0**. Сумма данных чисел порождает случайное число R_1. Далее R_1 нужно занести в R_0 и повторить операции. Схема:

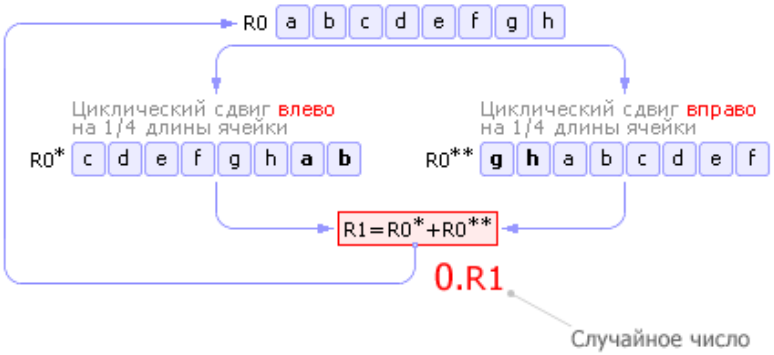


Рисунок 4. Схема работы алгоритма перемешивания

Если новое число не помещается в ячейку, отбрасываются лишние разряды. Пусть $R_0^* = 10010001_2 = 145_{10}$, $R_0^{**} = 10100001_2 = 161_{10}$, тогда $R_0^* + R_0^{**} = 100110010_2 = 306_{10}$. Как видим, число 306 занимает 9 разрядов (в двоичной системе счисления), а ячейка R_1 (как и R_0) может вместить в себя максимум 8 разрядов. Поэтому перед занесением значения в R_1 необходимо убрать один «лишний», крайний левый бит из числа 306, в результате чего в R_1 пойдет уже не 306, а $00110010_2 = 50_{10}$

8 слайд

Проверка ГСЧ

До сих пор не затрагивалась тема проверки пригодности генераторов. На самом деле достаточно сложно "проверить случайность", так как, если говорить тривиально, нет эталона, с которым можно быстро сравнить полученную последовательность. Но все-таки существует два основных типа проверок ГСЧ и случайных последовательностей:

1. Тест на равномерность
2. Тест на статистическую независимость

Проверка на равномерность

Вообще говоря, ГСЧ должен генерировать равномерно распределенную СВ, а ее параметры должны быть максимально близки к следующим:

$$m_r = \frac{\sum_{i=1}^{n} r_i}{n} \approx 0{,}5 - \text{математическое ожидание}$$

$$D_r = \frac{\sum_{i=1}^{n} (r_i - m_r)^2}{n} \approx 0{,}0833 - \text{дисперсия}$$

$$\sigma_r = \sqrt{D_r} = 0{,}2887 - \text{среднеквадратичное отклонение}$$

С помощью частотного теста можно проверить, сколько чисел попало в доверительный интервал в пределах одного стандартного отклонения. Проведя расчеты по эталонным значениям, указанным выше, получаем долю примерно в 57,7% из всех случайно сгенерированных чисел:

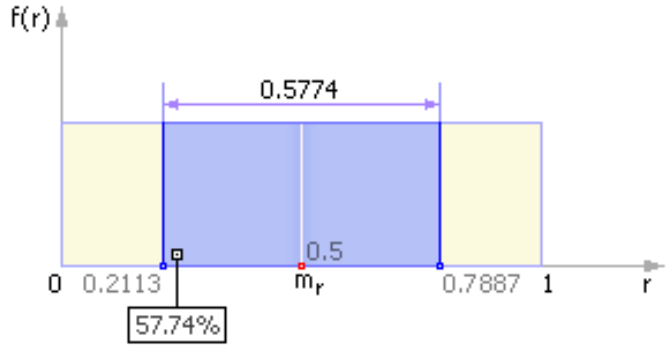


Рисунок 5. Доверительный интервал равномерного распределения

Тест хи-квадрат. Данный критерий позволит нам проверить то, насколько близка полученная последовательность, а значит и наш генератор, к эталонному ГСЧ. Так как закон распределения эталонного ГСЧ равномерный, то (теоретическая) вероятность p_i попадания чисел в i-ый интервал (всего этих интервалов k) равна $p_i = \frac{1}{k}$. И, таким образом, в каждый из k интервалов попадет ровно по $p_i N$ чисел (N — общее количество сгенерированных чисел). Реальный ГСЧ будет выдавать числа, распределенные (не всегда равномерно) по k интервалам и в каждый интервал попадет по n_i чисел (в сумме $n_1 + n_2 + \ldots + n_k = N$). Вполне логично рассмотреть квадраты разностей между полученным количеством чисел n_i и «эталонным» $p_i N$. Сложим их, и в результате получим:

$$\chi^2_{\text{эксп.}} = (n_1 - p_1 N)^2 + (n_2 - p_2 N)^2 + \ldots + (n_k - p_k N)^2$$

Из этой формулы следует, что чем меньше разность в каждом из слагаемых (а значит, и чем меньше значение $\chi^2_{\text{эксп.}}$), тем сильнее закон распределения случайных чисел, генерируемых реальным ГСЧ, тяготеет к равномерному. Далее необходимо провести нормировку каждого i-го слагаемого, поделив его на $p_i N$:

$$\chi^2_{\text{эксп}} = \frac{(n_1 - p_1 * N)^2}{p_1 * N} + \frac{(n_2 - p_2 * N)^2}{p_2 * N} + \cdots + \frac{(n_k - p_k * N)^2}{p_k * N}$$

После упрощения получим следующее уравнение:

$$\chi^2_{\text{эксп}} = \sum_{i=1}^{k} \frac{(n_i - p_i * N)^2}{p_i * N} = \frac{1}{N} \sum_{i=1}^{k} \left(\frac{n_i^2}{p_i}\right) - N$$

Мы получили значение критерия хи-квадрат для экспериментальных данных. По таблице квантилей распределения хи-квадрат определяется то, насколько генератор удовлетворяет требованию равномерного распределения. Если теоретическое значение сильно больше или сильно меньше, то ГСЧ не проходит тест. А если значение лежит между двумя значениями, где p (доверительная вероятность) находится в интервале от 0.1 до 0.9, то есть основания полагать случайность полученных чисел. Однако нужно отметить, что для таких проверок нужна достаточно большая выборка. Дополнительно отметим, что чем ближе доверительная вероятность к "середине", то есть к 0.5, тем лучше.

9 слайд

Проверка на независимость

Немного о проверке на независимость. Один из вариантов – проверка на частоту появления цифры в последовательности. Рассмотрим пример. Случайное число 0.2463389991 состоит из цифр 2463389991, а число 0.5467766618 состоит из цифр 5467766618. Соединим последовательности: 246338999154677666618. Понятно, что теоретическая вероятность p_i выпадения i-ой цифры (от 0 до 9) равна 0.1. Далее следует вычислить частоту появления каждой цифры в выпавшей экспериментальной последовательности. Например, цифра 1 выпала 2 раза из 20, а цифра 6 выпала 5 раз из 20. Далее считаем значение критерия хи-квадрат и принимаем решение.

Отдельно стоит отметить то, что существует и третий способ генерации случайных чисел, называемый табличным, основанный на формировании файлов о помещении их в память. Он не получил широкого применения и его сложно классифицировать как ГИСЧ или ГПСЧ.

10 слайд

Пакет статистических тестов NIST STS

В 1999 г. были выполнены исследовательские работы (NIST – Национальный институт стандартов и технологий, США), в результате которых был сформирован набор из 15 статистических тестов для проверки свойств псевдослучайных последовательностей. Данное решение считается наиболее эффективным в плане строгости оценок и использования вычислительных ресурсов. Пакет состоит из 15 статистических тестов, которые объединены одной методикой анализа двоичных генераторов псевдослучайных последовательностей. В него также входят многие генераторы, рассмотренные ранее.

Тесты основываются на проверке гипотезы о случайности проверяемой последовательности. Альтернативная гипотеза утверждает о том, что последовательность неслучайна. В каждом тесте вычисляется определенная внутри теста статистика и сравнивается с теоретическим значением (из эталонного распределения), а также после каждого теста нулевая гипотеза принимается или отклоняется. Эталонными распределениями являются:

- Распределение χ^2 (10 из 15 тестов)
- Одностороннее усеченное нормально распределение (3 из 15 тестов)
- Нормальное распределение (2 из 15 тестов)

По этим же статистикам рассчитываются p-value. Все тесты в пакете параметрические – необходимо корректно выбрать соответствующие параметры. Далее приведен список тестов и их краткое описание.

1. **Частотный (монобитный) тест (The Frequency (Monobit) Test)**, определяющий нормализованную абсолютную сумму значений элементов последовательности. Тест показывает, не содержится ли в последовательности слишком много нулей или единиц.
2. **Частотный тест внутри блока (Frequency Test within a Block).** Определяет меру согласования количества единиц внутри блока с теоретически ожидаемым значением. Показывает локализованные отклонения частоты появления единиц в блоке от теоретического значения.
3. **Проверка накопленных сумм (The Cumulative Sums (Cusums) Test).** Вычисляется максимальное отклонение накопленной суммы элементов последовательности от начальной точки отсчета. Тест определяет, не слишком ли большое количество единиц или нулей находится в начале или в конце последовательности.
4. **Проверка серий (The Runs Test).** Определяет количество непрерывных серий одинаковых битов на всей длине последовательности. Показывает, есть ли слишком быстрая или слишком медленная перемена серий одинаковых битов в последовательности.
5. **Проверка максимальной длины серии в блоке (Tests for the Longest-Run-of-Ones in a Block).** Определяет меру согласования наблюдаемого значения максимальной длины единичной серии с теоретически ожидаемым значением. Показывает отклонение максимальных длин серий единиц от теоретического закона распределения.
6. **Проверка ранга двоичной матрицы (The Binary Matrix Rank Test).** Определяет меру согласования наблюдаемого значения рангов различного порядка с теоретически ожидаемым. Выявляет зависимость символов в последовательности по отклонению эмпирического закона распределения значений рангов матрицы от теоретического.
7. **Спектральный тест на основе дискретного преобразования Фурье (The Discrete Fourier Transform (Spectral) Test).** Вычисляет нормализованную разницу между наблюдаемым и ожидаемым количеством частотных компонент, превышающих 95% порогового уровня. Выявляет периодические составляющие в двоичной последовательности.
8. **Проверка перекрывающихся шаблонов (The Overlapping Template Matching Test).** Определяет меру согласования наблюдаемого количества перекрывающихся шаблонов в последовательности с их теоретическим значением. Показывает наличие в последовательности большого количества m-битных серий из единиц.
9. **Универсальный тест Маурера (Maurer's «Universal Statistical»).** Вычисляет сумму логарифма расстояния между однобитными шаблонами. Указывает на сжимаемость последовательности.
10. **Энтропийный тест (The Approximate Entropy Test).** Определяет меру согласования наблюдаемого значения энтропии источника с теоретически ожидаемым значением для случайного источника. Характеризует неравномерность распределения m-битных слов в последовательности.
11. **Проверка случайных отклонений (The Random Excursions Test).** Определяется мера согласования наблюдаемого количества посещений при случайном блуждании в заданное состояние внутри цикла с теоретически ожидаемым количеством. Показывает

отклонение от теоретического закона распределения посещений в конкретное состояние при случайном блуждании.

12. **Проверка случайных отклонений (вариантный) (The Random Excursions Variant Test).** Определяется общее количество посещений в различные состояния при случайном блуждании. Показывает отклонение от теоретического ожидаемого общего количества посещений при случайном блуждании в различные состояния.
13. **Тест на подпоследовательности (Serial Test).** Определяет меру согласования наблюдаемого количества всех встретившихся вариантов m-битных шаблонов с теоретически ожидаемым. Показывает неравномерность распределения m-битных слов в последовательности.
14. **Проверка неперекрывающихся шаблонов (The Non-overlapping Template Matching Test).** Определяется мера согласования наблюдаемого количества непериодических шаблонов в последовательности с теоретическим значением. Показывает наличие большого количества заданных непериодических шаблонов в последовательности.
15. **Проверка линейной сложности (The Linear Complexity Test)**. Определяется мера согласования наблюдаемого количества событий, заключающихся в появлении фиксированной длины эквивалентного линейного рекуррентного регистра (ЛРР) для заданного блока с теоретически ожидаемым. Показывает наличие отклонения эмпирического распределения длин эквивалентных ЛРР для последовательности фиксированной длины от теоретического закона распределения для случайной последовательности, что указывает на недостаточную сложность тестируемой последовательности.

Каждый тест выполняет расчет определённой статистической характеристики и соответствующих значений вероятности того, что гипотеза о случайности ПСП является правильной.

Тесты различаются по сложности используемых в них алгоритмов и по объему обрабатываемых участков последовательности.

Часть тестов работает с отдельными битами, проверяя наиболее простые характеристики последовательности (количество нулей или единиц, накопленную сумму элементов последовательности, количество серий одинаковых битов в последовательности и т. д.).

Тесты второй группы обрабатывают m-битные блоки. Это тесты на встречающиеся пересекающиеся и непересекающиеся шаблоны, универсальный тест Мауэра, тест на подпоследовательности, энтропийный тест.

Самые сложные тесты обрабатывают большие блоки (длиной более 1000 битов): проверка линейной сложности, спектральный тест, проверка ранга двоичной матрицы.

11 СЛАЙД

Выводы о прохождении теста делаются по результатам проверки двух условий:

1. Попадание величины P_r(доля последовательностей, сгенерированных с помощью ГСЧ, прошедших тест. Определяется как отношение успешных попыток к общему числу тестов) в доверительный интервал:

$$\left[(1-\ \alpha) - 3\sqrt{\frac{\alpha(1-\alpha)}{m}},\qquad (1-\alpha) + 3\sqrt{\frac{\alpha(1-\alpha)}{m}}\right], m - \text{объем выборки}$$

2. Равномерность распределения вероятностей p-value на отрезке [0, 1]. Для этого интервал разбивается на *k=10* подинтервалов длины 0.1 и вычисляется статистика:

$$\chi^2 = \frac{\sum_{i=1}^{k}\left(v_i - {}^{m}/_{k}\right)^2}{{}^{m}/_{k}}, v_i - \text{количество } p\ value \text{ на } i - \text{ом интервале}$$

Если число прошедших тест последовательностей велико, распределение этой статистики должно приближаться к распределению χ^2 с числом степеней свободы (k-1).

12 СЛАЙД
Тестовый запуск на тестовых данных

13 СЛАЙД результаты теста
Можно показать файлы по папкам

14 СЛАЙД

СО СЛАЙДА

Генераторы случайных чисел в моделировании

15 слайд

Метод Монте-Карло

В самом начале важно отметить, что основой статистического моделирования является метод Монте-Карло. А основой метода Монте-Карло составляют случайные числа. Вкратце опишем метод для дальнейшего понимания. Допустим, существует некий интеграл вида $y = \int_{x_1}^{x_2} f(x)dx$, который мы не можем вычислить аналитически. У нас есть график функции. Следовательно, вычислить интеграл – значит найти площадь под графиком функции.

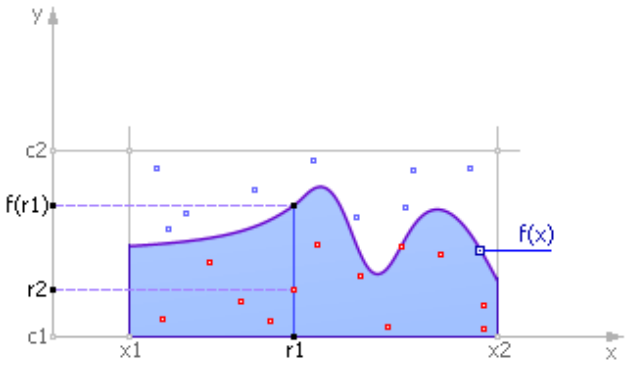


Рисунок 6. Графическое представление метода Монте-Карло

Ограничиваем кривую сверху, справа и слева. Случайным образом распределяем точки в прямоугольнике поиска. Обозначим через N_1 количество точек, принятых для испытаний (то есть попавших в прямоугольник, эти точки изображены на графике красным и синим цветом), и через N_2 — количество точек под кривой, то есть попавших в закрашенную площадь под функцией (эти точки изображены на графике красным цветом). Тогда естественно предположить, что количество точек, попавших под кривую по отношению к общему числу точек пропорционально площади под кривой (величине интеграла) по отношению к площади испытуемого прямоугольника. Математически это можно выразить так:

$$\frac{N_2}{N_1} = \frac{y}{(x_2 - x_1)(c_2 - c_1)}$$

Чем больше количество точек, тем вернее данное предположение. Алгоритм выглядит следующим образом

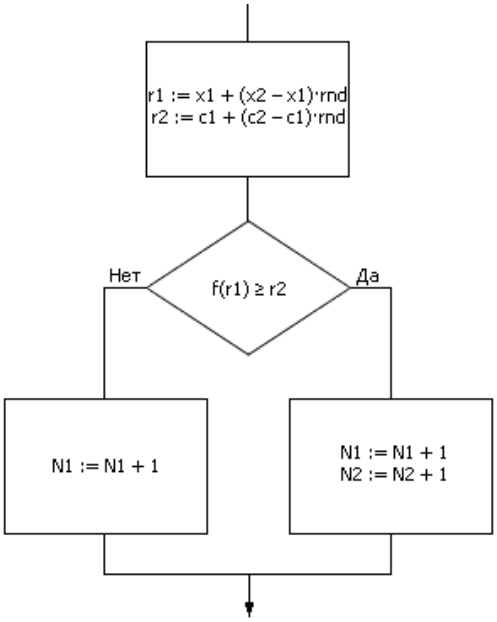


Рисунок 7. Блок-схема метода Монте-Карло

Значения r_1 и r_2 в алгоритме являются равномерно распределенными случайными числами из интервалов (x_1; x_2) и (c_1; c_2) соответственно.

Метод Монте-Карло чрезвычайно эффективен, прост, но необходим «хороший» генератор случайных чисел. Вторая проблема применения метода заключается в определении объема выборки, то есть количества точек, необходимых для обеспечения решения с заданной точностью. Эксперименты показывают: чтобы увеличить точность в 10 раз, объем выборки нужно увеличить в 100 раз; то есть точность примерно пропорциональна корню из объема выборки:

$$\text{точность} \cong \sqrt{\text{объем выборки}}$$

16 слайд

Схема использования метода Монте-Карло при исследовании систем со случайными параметрами

Построив модель системы со случайными параметрами, на ее вход подают входные сигналы от генератора случайных чисел (ГСЧ), как показано на рисунке ниже. ГСЧ устроен так, что он выдает равномерно распределенные случайные числа r_{pp} из интервала [0; 1]. Так как одни события могут быть более вероятными, другие — менее вероятными, то равномерно распределенные случайные числа от генератора подают на преобразователь закона случайных чисел (ПЗСЧ), который преобразует их в заданный пользователем закон распределения вероятности, например, в нормальный или экспоненциальный закон. Эти преобразованные случайные числа x подают на вход модели. Модель отрабатывает входной сигнал x по некоторому закону y = φ(x) и получает выходной сигнал y, который также является случайным.

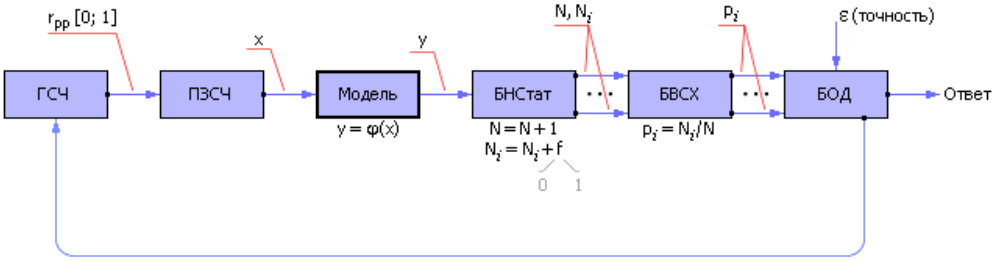


Рисунок 8. Схема использования метода Монте-Карло для систем со случайными параметрами

В блоке накопления статистики (БНСтат) установлены фильтры и счетчики. Фильтр (некоторое логическое условие) определяет по значению y, реализовалось ли в конкретном опыте некоторое событие (выполнилось условие, f = 1) или нет (условие не выполнилось, f = 0). Если событие реализовалось, то счетчик события увеличивается на единицу. Если событие не реализовалось, то значение счетчика не меняется. Если требуется следить за несколькими разными типами событий, то для статистического моделирования понадобится несколько фильтров и счетчиков N_i. Всегда ведется счетчик количества экспериментов — N.

Далее отношение N_i к N, рассчитываемое в блоке вычисления статистических характеристик (БВСХ) по методу Монте-Карло, дает оценку вероятности p_i появления события i, то есть указывает на частоту его выпадения в серии из N опытов. Это позволяет сделать выводы о статистических свойствах моделируемого объекта. Например, событие A совершилось в результате проведенных 200 экспериментов 50 раз. Это означает, согласно методу Монте-Карло, что вероятность совершения события равна: $p_A = 50/200 = 0.25$. Вероятность того, что событие не совершится, равна, соответственно, 1 – 0.25 = 0.75.

Нужно подчеркнуть, что после проведения эксперимента говорят уже о частости появления наблюдений, а не об их вероятности (она употребляется, когда речь о теоретическом понятии без проведенных экспериментов). При большом количестве опытов N частота появления события, полученная экспериментальным путем, стремится к значению теоретической вероятности появления события.

В блоке оценки достоверности (БОД) анализируют степень достоверности статистических экспериментальных данных, снятых с модели (принимая во внимание точность результата ε, заданную пользователем) и определяют необходимое для этого количество статистических испытаний. Если колебания значений частоты появления событий относительно теоретической вероятности меньше заданной точности, то экспериментальную частоту принимают в качестве ответа, иначе генерацию случайных входных воздействий продолжают, и процесс моделирования повторяется. При малом числе испытаний результат

может оказаться недостоверным. Но чем более испытаний, тем точнее ответ, согласно центральной предельной теореме. Заметим, что оценивание ведут по худшей из частот. Это обеспечивает достоверный результат сразу по всем снимаемым характеристикам модели.

17 слайд

Имитация случайных событий

Самая базовая задача моделирования с использованием ГСЧ – имитация наступления случайных событий. Случайное событие подразумевает, что у некоторого события есть несколько исходов и то, который из исходов произойдет в очередной раз, определяется только его вероятностью. То есть исход выбирается случайно с учетом его вероятности.

Например, допустим, что нам известна вероятность выпуска бракованных изделий $P_б = 0.1$. Смоделировать выпадение этого события можно, разыграв равномерно распределенное случайное число из диапазона от 0 до 1 и установив, в какой из двух интервалов (от 0 до 0.1 или от 0.1 до 1) оно попало (рисунок ниже). Если число попадает в диапазон (0; 0.1), то выпущен брак, то есть событие произошло, иначе — событие не произошло. При значительном числе экспериментов частота попадания чисел в интервал от 0 до 0.1 будет приближаться к вероятности $P = 0.1$, а частота попадания чисел в интервал от 0.1 до 1 будет приближаться к $P_к = 0.9$.

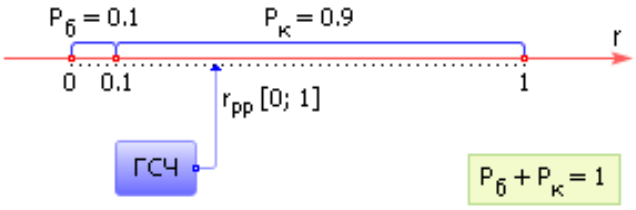


Рисунок 9. Имитация случайного события

Алгоритм в данном случае максимально прост:

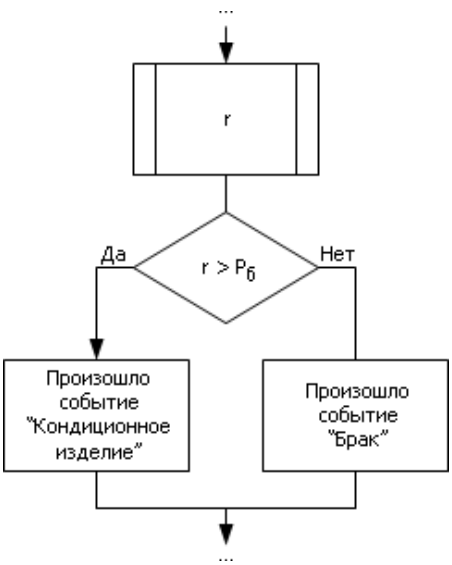


Рисунок 10. Блок-схема имитации случайного события

Заметим, что нам не важно расположение интервала $P_б$ — в начале или в конце, поскольку метод Монте-Карло учитывает только частоту попадания случайных точек в интервал, а она зависит только от величины интервала и не зависит от его месторасположения.

18 слайд

Моделирование полной группы несовместных событий

События называются несовместными, если вероятность появления этих событий одновременно равна 0. Отсюда следует, что суммарная вероятность группы несовместных событий равна 1. Обозначим через a_1, a_2, …, a_n события, а через P_1, P_2, …, P_n — вероятности появления отдельных событий. Так как события несовместны, то сумма вероятностей их выпадения равна единице: $P_1 + P_2 + … + P_n = 1$. Для имитации события используем ГСЧ, выдающий числа из отрезка [0; 1]. Отложим на единичном интервале [0; 1] отрезки P_1, P_2, …, P_n. Понятно, что в сумме отрезки составят точно единичный интервал. Точка, соответствующая выпавшему числу из ГСЧ на этом интервале, укажет на один из отрезков. Соответственно в большие отрезки случайные числа будут попадать чаще.

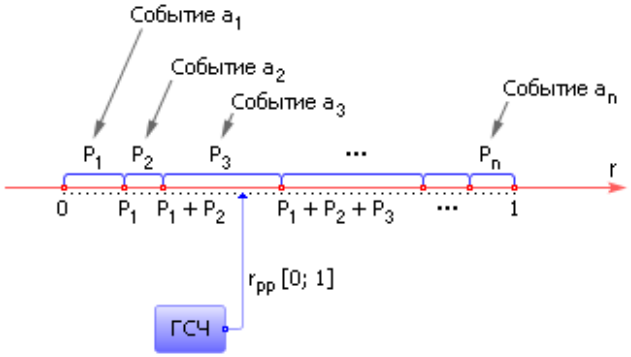


Рисунок 11. Моделирование полной группы несовместных событий

На рисунке ниже показана блок-схема, которая реализует описанный алгоритм. Алгоритм определяет с помощью фильтра, построенного в виде последовательности условных операций (IF), в какой из интервалов — от 0 до P_1, от P_1 до $(P_1 + P_2)$, от $(P_1 + P_2)$ до $(P_1 + P_2 + P_3)$ и так далее — попало число, сгенерированное генератором случайных чисел. Если число попало в какой-то из интервалов (что произойдет всегда и обязательно), то это соответствует выпадению связанного с ним события.

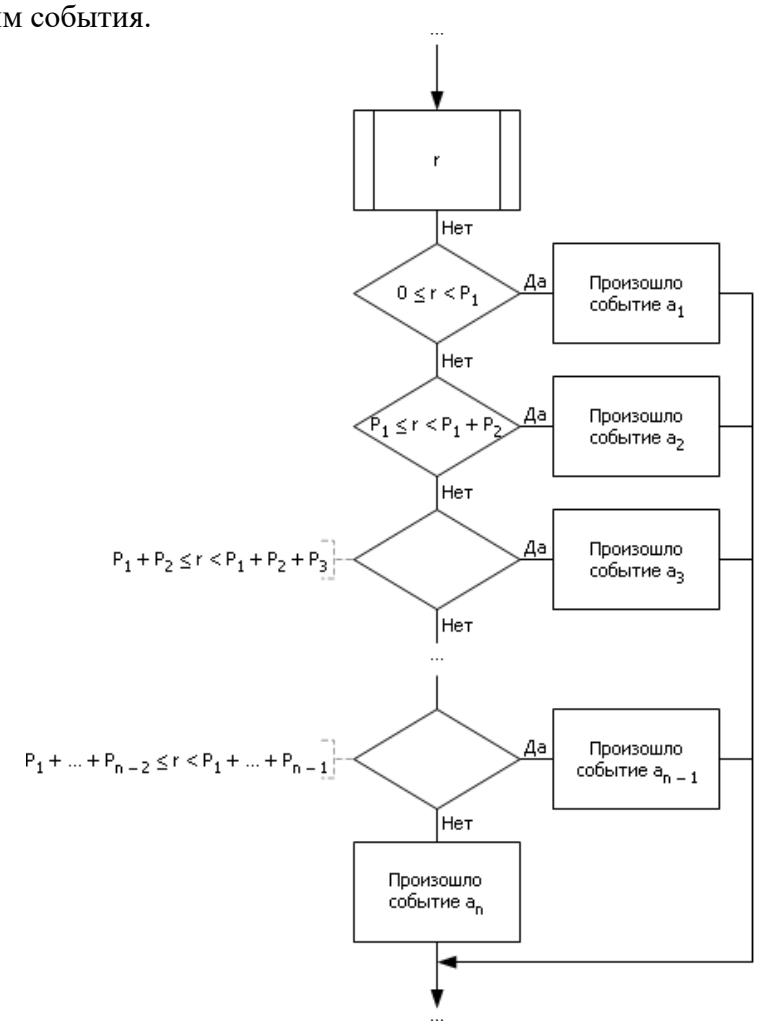


Рисунок 12. Блок-схема алгоритма моделирования

Здесь можно привести простой и понятный пример. Промоделируем выпадение последовательности событий — будем выбирать из колоды карт наугад карту (определять ее масть). Карты в колоду возвращать не будем. В колоде 36 карт четырех мастей по 9 карт каждой масти. Интервал от 0 до 1 разделим на равные четыре части: [0.00—0.25], [0.25—0.50], [0.50—0.75], [0.75—1.00]. Первая часть будет соответствовать картам масти червей (Ч), вторая — картам масти пик (П), третья — картам масти виней (В), четвертая — бубей (Б). Генерируем случайное равномерно распределенное число с помощью ГСЧ. Пусть, например, это будет число 0.597. Данное число попадает в третий интервал, соответствующий масти В. Произошло случайное событие: «Масть выпавшей карты — В». Поскольку теперь в колоде 9 карт масти Ч, 9 карт масти П, 8 карт масти В, 9 карт масти Б, то интервал от 0 до 1 будет разбит на отрезки длиной: 9/35, 9/35, 8/35, 9/35, то есть [0.000—0.257], [0.257—0.514], [0.514—0.743], [0.743—1.000]. Генерируем числа дальше. Например, 0.321. Данное число попадает во второй интервал, соответствующий масти П. Продолжая процесс, можно получить (в зависимости от конкретных случайных чисел), например, такую последовательность: В—П—В—Ч—Б—П—Ч—…

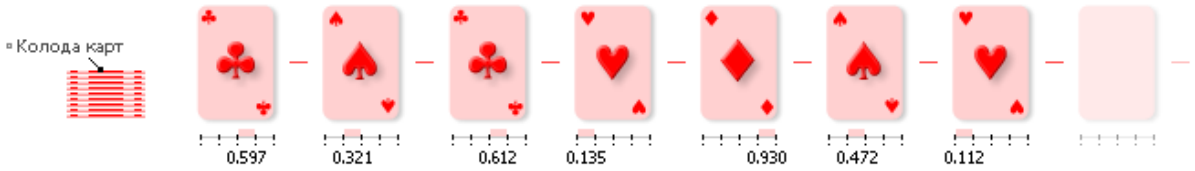


Рисунок 13. Графическая версия полученного результата

19 слайд

Моделирование случайных величин

Одной из основных задач моделирования, использующих ГСЧ, является моделирование СВ с некоторым распределением. С «хорошим» ГСЧ достаточно просто моделировать равномерное распределение, однако это покроет только небольшую часть возможных задач. В идеале необходимо иметь возможность контролировать ГСЧ на предмет типа распределения генерируемой последовательности. Для этого будем использовать метод ступенчатой аппроксимации и дискретизацию (даже в случае непрерывных распределений). Введем обозначения: h_i — высота i-го столбца, f(x) — распределение вероятности (показывает, насколько вероятно некоторое событие x). Значение h_i операцией нормировки необходимо перевести в единицы вероятности появления значений x из интервала $x_i < x \leq x_i + 1$:

$$P_i = \frac{h_i}{(h_1 + h_2 + \ldots + h_i + \ldots + h_n)}$$

Учтем нормировку в условиях работы с вероятностями:

$$\sum_{i=1}^{n} P_i = 1$$

Графически переход от произвольного непрерывного распределения к дискретному, отображение вероятностей на интервал и генерация случайных событий выглядит так:

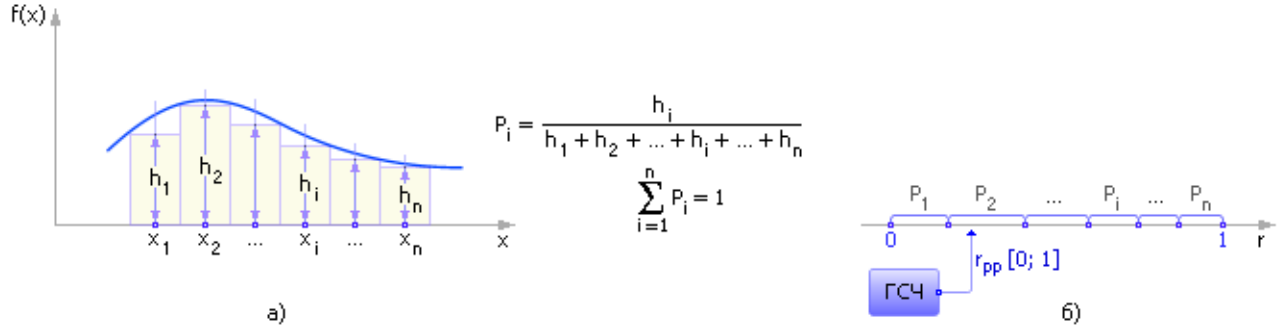


Рисунок 14. Процесс перехода от произвольного распределения к дискретному

Важно отметить, что внутри каждого из интервалов значение x будет дискретным, то есть одинаковым. Это является погрешностью данного метода, потому что мы перешли от непрерывного распределения к дискретному. В данном случае точность будет зависеть от количества интервалов, на которые мы разбили начальные значения. Блок-схема алгоритма:

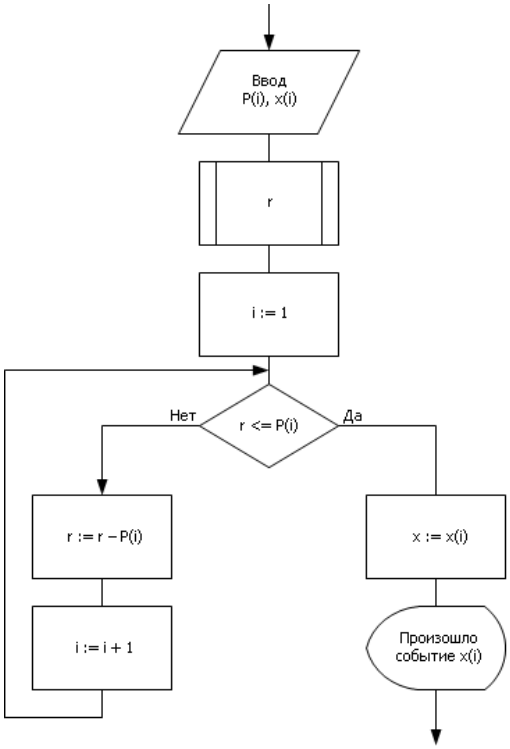


Рисунок 15. Блок-схема алгоритма перехода от произвольного распределения к дискретному

Сначала генерируется случайное число, которое равномерно распределено на [0; 1]. Затем производится сравнение границ отрезков, которые одновременно являются и вероятностями, в результате определяются произошедшие события (из изначально заданного закона распределения).

20 слайд

Генерация случайных чисел с нормальным распределением с использованием центральной предельной теоремы.

В данном случае метод основывается на сложении случайных чисел (любого распределения) и приведении (нормализации) в необходимый диапазон нормального распределения. В результате имитационного моделирования мы хотим получить нормально распределенные случайные числа с заданными математическим ожиданием и среднеквадратичным отклонением. Алгоритм прост и состоит из следующих этапов:

1. Генерация n случайных чисел и их сумма: $S = \sum_{i=1}^{n} r_i$. Согласно центральной предельной теореме, данная сумма образует ряд значений с нормальным распределением (конечно же при достаточно большом значении n). Данный закон имеет МО $\frac{n}{2}$, СКО $\sqrt{\frac{n}{12}}$.
2. Далее производится z-стандартизация (нормализация) с помощью формулы $z = \frac{S - m_S}{\sigma_S}$. Теперь МО = 0, СКО = 1.
3. Теперь сдвигами и масштабированием (из принципов нормального распределения), преобразуем последовательность z в x: $x = z * \sigma_x + m_x$

Приведем производственный пример. Допустим, что имеется некоторый поток изготовляемых изделий. Средняя длина изделия имеет случайную погрешность и МО 35 см, СКО 10 см. Задача – сгенерировать поток заготовок. Производим расчеты, генерируя случайные числа:

$$S = \sum_{i=1}^{6} r_i$$

$$Z = \sum_{i=1}^{6} \frac{s_i - 35}{10}$$

$$x_i = z_i * \sigma_x + m_x,$$

где МО и СКО – параметры желаемого распределения

21 слайд

Моделирование потоков случайных событий

Еще одной важной задачей, моделируемой при помощи ГСЧ, является моделирование некоторого количества событий, которые следуют друг за другом. Такой процесс называется потоком событий. Возникает повсеместно, наиболее знакомый из нашей учебной программы пример – системы массового обслуживания, где таким потоком поступают заявки на обработку (не идеальный, но в общем подходящий пример, далее рассмотрим подробнее полноценный пример на СМО). Поток характеризуется интенсивностью – средним количеством событий в единицу времени. Эталонным в моделировании считается пуассоновский поток. В нем вероятность одновременного появления двух и более событий равна 0, то есть это ординарный поток. Можно рассчитать вероятность появления m событий за некоторый промежуток времени (t0, t0 + Δt):

$$P_m = \frac{a^m e^{-a}}{m!}$$

$$a = \int_{t_0}^{t_0+\Delta} \lambda(t)dt$$

Если $\lambda(t) = const$, то поток стационарный, $a = \lambda t$. В таком случае можем найти вероятность появления m событий за время τ:

$$P_m = \frac{(\lambda\tau)^m e^{-\lambda\tau}}{m!}$$

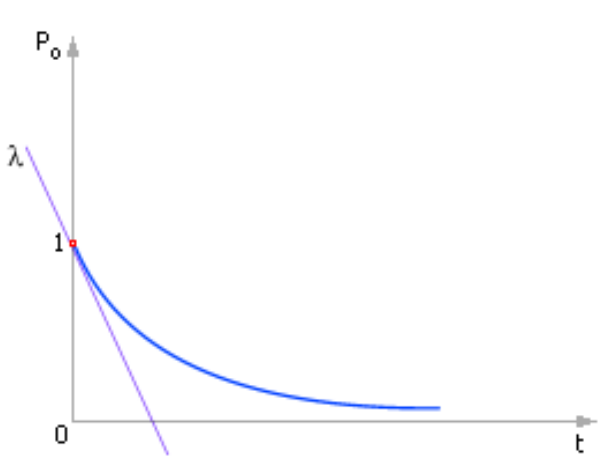
А также вероятность того, что ни одно событие не появится за заданное время:

$$P_0 = \frac{(\lambda\tau)^0 e^{-\lambda\tau}}{0!} = e^{-\lambda\tau}$$

И, соответственно, вероятность появления хотя бы одного события:

$$P_{m>0} = 1 - P_0 = 1 - e^{-\lambda\tau}$$

Графически влияние интенсивности на непоявление или появление хотя бы одного события выглядит следующим образом:



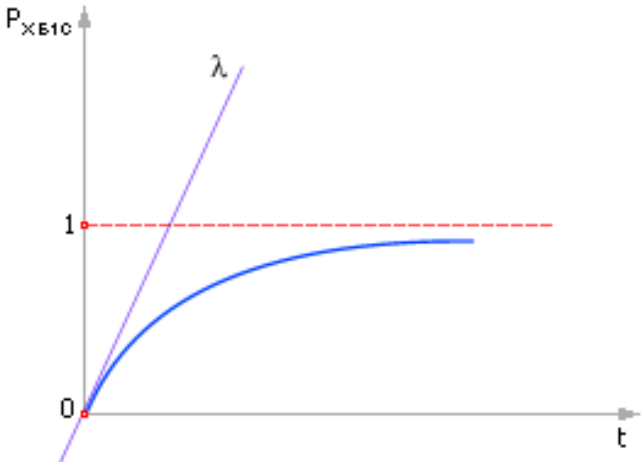


Рисунок 16. Отсутствие события и интенсивность *Рисунок 17. Хотя бы одно событие и интенсивность*

Здесь параметр интенсивности λ обозначена наклоном касательной. Также мы можем выразить τ из уравнения вероятности появления хотя бы одного события:

$$\tau = -\frac{1}{\lambda}\ln(r),$$

где r – равномерно распределенное (от 0 до 1) случайное число, полученное из ГСЧ.

Для примера рассмотрим производственную задачу. Допустим, что некоторые заявки поступают случайным образом, но в среднем восемь штук в сутки. Нужно воспроизвести этот процесс, длительность которого сто часов, модельно.

Для решения вычислим необходимые параметры. $\lambda = \frac{8}{24}$ ед/час, $m = \frac{1}{\lambda} = \frac{24}{8} = 3 = \sigma$.

Имеем алгоритм:

1. $t = 0, N = 0$
2. Получить r из ГСЧ
3. $\tau = -\frac{1}{\lambda}\ln(r)$
4. $t = t + \tau$
5. $N = N + 1$
6. $t \leq T$?
7. Да – возврат к шагу 2, нет – конец

Результат работы алгоритма может быть таким:

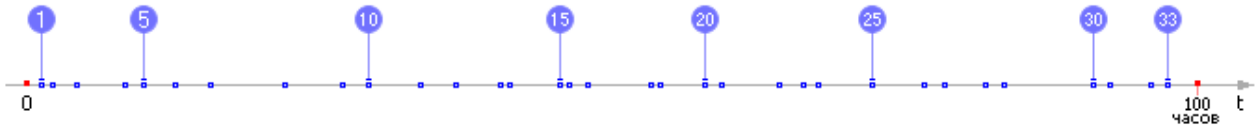


Рисунок 18. Один из возможных результатов моделирования потока случайных событий

В данном случае было получено 33 появившихся события (обработанных заявки), что достаточно близко к среднему значению N= 33,(3). Также расстояния (то есть время) между событиями в среднем будут равны 3.

22 слайд

Моделирование марковских процессов

Случайные события и их появление очень удобно рассматривать с точки зрения системы и ее состояний, при условии, что переход между состояниями зависит только от того состояния, в котором система находится перед переходом. Рассмотрим дискретный марковский процесс. Граф переходов может выглядеть так:

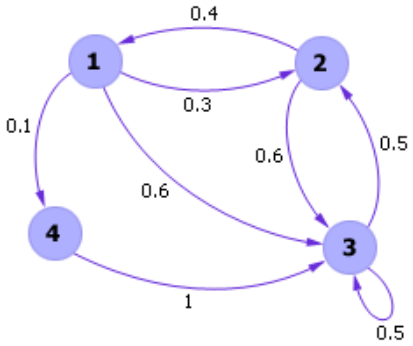


Рисунок 19. Граф переходов марковского процесса

У каждого перехода есть своя вероятность P_{ij}. У каждого состояния сумма таких переходов должна быть равна 1. Сам марковский процесс, а точнее процесс его моделирования, будет выглядеть как цепь (последовательность) переходов между состояниями. Она является случайной и может выглядеть по-разному, например так:

Рисунок 20. Марковская цепь

Для определения нового состояния, в которое произойдет переход из текущего, нужно разбить отрезок [0, 1] на интервалы P_{i1}, P_{i2}, P_{i3}, … ($P_{i1} + P_{i2} + P_{i3} + \ldots = 1$). После этого из ГСЧ нужно получить равномерно распределённое случайное число на этом же отрезке (r_{pp}) и установить интервал, в который оно попало:

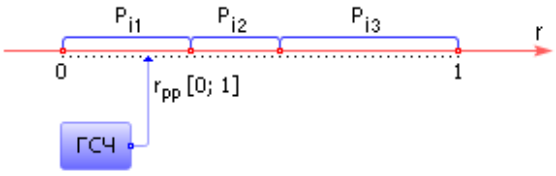


Рисунок 21. Определение следующего состояния

После этого действия повторяются, результатом является марковская цепь.

23 слайд

Для примера можно рассмотреть стрельбу в объект до его разрушения. Пусть имеется три состояния – цель не повреждена, цель повреждена, цель разрушена (S_0, S_1, S_2). Пусть вектор

начальных состояний равен (1, 0, 0). А также имеем матрицу переходных вероятностей марковского процесса:

	S_0	S_1	S_2
S_0	0.45	0.4	0.15
S_1	0	0.45	0.55
S_2	0	0	1

Таблица 1. Матрица переходных вероятностей марковского процесса

Имитируем процесс стрельбы с использованием ГСЧ и находясь в нулевом состоянии. Пусть были получены следующие значения: 0.84, 0.31, 0.15, 0.44, 0.65. Сравнивая их с вероятностями, производим переход в следующее состояние на 1 и 5 шаге, моделирование заканчивается после 5 шагов. Следовательно, понадобилось 5 выстрелов для разрушения объекта.

Заключение

Важно отметить, что ГСЧ действительно распространены широко. Существует множество алгоритмов, на которых они работают. В работе рассмотрены лишь несколько из этого большого множества. Еще больше существует задач, в которых они применяются. Моделирование, различные эксперименты, финансы, бизнес, маркетинг, инфокоммуникации – всего лишь часть областей использования. В работе приведены некоторые задачи для лучшего понимания темы и ее демонстрации с точки зрения моделирования.

Литература

1. Гончарук, В. С. Методы генерации случайных чисел / В. С. Гончарук, Ю. С. Атаманов, С. Н. Гордеев. — Текст : непосредственный // Молодой ученый. — 2017. — № 8 (142). — С. 20-23. — URL: https://moluch.ru/archive/142/40025
2. Л. Ю. Бараш, Л. Н. Щур, Генерация случайных чисел и параллельных потоков случайных чисел для расчетов Монте-Карло, Модел. и анализ информ. систем, 2012, том 19, номер 2, 145–161. — URL: https://www.mathnet.ru/links/2550fc27e7534c6401e6ae465963be09/mais226.pdf
3. Санников С. П. Моделирование систем, рецензент Ордуянц Г. Г., ред. Черных Л. Д. — УГЛТУ, Екатеринбург, 2012. — URL: https://elar.usfeu.ru/bitstream/123456789/948/2/Sannikov_4.pdf
4. http://stratum.ac.ru/education/textbooks/modelir/author.html
5. https://ru.frwiki.wiki/wiki/Générateur_de_nombres_aléatoires
6. https://xreferat.com/33/3934-1-generator-sluchaiynyh-chisel.html
7. https://intuit.ru/studies/courses/940/409/info