Оглавление

[Описание математического аппарата для тестового задания 1](#_Toc432407107)

[Требования 1](#_Toc432407108)

[Немного теории 3](#_Toc432407109)

[Исходники 6](#_Toc432407110)

# Описание математического аппарата для тестового задания

Автомат имеет **5** барабанов по **3** символа в видимой области, т.е. игровое поле представляется матрицей 3x5  
 Примерная таблица выигрышей для комбинаций, сложившихся на линии, при ставке на линию, равной **1**:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Кол-во символов в комбинации* | *Символы Si, i =1..9* | | | | | | | | |
| ***S1*** | ***S2*** | ***S3*** | ***S4*** | ***S5*** | ***S6*** | ***S7*** | ***S8*** | ***S9*** |
| ***5*** | *45* | *75* | *150* | *250* | *350* | *500* | *750* | *1000* | *2000* |
| ***4*** | *15* | *30* | *50* | *60* | *75* | *90* | *120* | *150* | *200* |
| ***3*** | *3* | *5* | *7* | *9* | *12* | *15* | *20* | *30* | *50* |

Выигрыши считаются слева направо. В выигрышной ситуации принимают участие только одинаковые символы, расположенные подряд.

## Требования

1. Создать математический аппарат, который позволяет (псевдо)случайно выдавать матрицу автомата 5x3, таким образом, чтобы:

- МО процента возврата было бы равно 0.90 при игре по 1 линии (т.е. ставке 1)

- Hit для был равен 0.2 (вероятность непроигрышного исхода партии)

- Оценить среднеквадратическое отклонение процента возврата

1. Реализовать модель автомата, базирующуюся на выполненном расчете.

Модель должна иметь:

- видимое поле 5x3, в котором показывается текущая игровая ситуация

- кнопка «Старт», генерирующая новую матрицу

- индикаторы «текущий выигрыш», «суммарный выигрыш», «суммарная ставка».

## Немного теории

Можно показать, что игровой процесс с точки зрения генерации поля и с точки зрения игрока эквивалентен такому процессу:

Простоты ради будем рассматривать такое пространство вариантов:

Все невыигрышные (выигрыш равен 0) ситуации объединим в элемент **o28**. Вероятность выпадения этого элемента у нас определяется как **1-Phit=0.8**. i.e. с вероятностью 0.8 должен выпадать исход ***o28***.

o1, o2, o3 –события выпадения трех, четырех и пяти символов S1 соответственно. Каждое из этих событий является подмножеством множества элементарных событий выпадения матрицы экрана 3x5.

o4, o5, o6 –события выпадения трех, четырех и пяти символов S2 соответственно

………………………………………………………………………………………………………………………..

o25, o26, o27 –события выпадения трех, четырех и пяти символов S9 соответственно

**O**={ o1, o2,…, o28}

Вероятности выпадения каждого обозначим через p1=P(o1), p2=P(o2), etc

Соответствующие выигрыши при единичной ставке обозначим через u1, u2,… u28 при этом u28=0

Очевидно, что существует бесконечное количеством наборов ui и pi, которые удовлетворяют предъявляемому нами условию Phit=0.2, M=0.9, т.е. таких, что

P28∙u28+∑ui∙pi = 0.9 (где i=1..28) и одновременно ∑pi=0.2

Решения нашей системы двух линейных уравнений (относительно вероятностей pi) образуют линейное пространство, но оно включает в себя и те решения, где некоторые из pi отрицательны. Нас, однако, интересуют только положительные решения.

В нашем случае – а именно тогда, когда таблица выигрышей фиксирована – мы ставим себе задачей определить такие вероятности pi, при которых будет выполняться требования к матожиданию выигрыша и вероятности выигрышного исхода партии.

Решение, очевидно, будет существовать не для каждого заданного набора выигрышей. Критерий существования набора p1,p2,..p27 , таких,что выполняются два равенства

∑pi=M и

∑ui∙pi=A , где i=1..28

Очевидно, можно записать так:

если мы обозначим через imin и imax такие индексы, что uimin и uimax, что являются соответственно минимальными и максимальными среди u1,u2,…u28, то очевидно  
M ∙ uimin ≤ ∑ui∙pi

M ∙ uimax ≥ ∑ui∙pi

Итак, необходимое и достаточное условие существования положительного набора pi системы двух уравнений

∑pi=M и

∑ui∙pi=A

можно записать

uimin≤A/M  
uimax≥A/M

В общих чертах поиск положительного решения у нас будет происходить так:

сначала мы выбираем p1 в пределах, определяемых условием существования положительного решения:  
p1 должно лежать в пределах [pmin,pmax]. Выбираются они так, чтобы система в которой количество переменных уменьшилось на одно всё еще удовлетворяла условию существования двух решений. При этом, для простоты, p1 можно выбирать из середины промежутка [pmin,pmax]. Дальше мы снова имеем систему уравнений, тоже имеющую положительные решения и просто повторяем всю процедуру.

Таким образом мы получаем набор вероятностей {pi}, таких что

∑ui∙pi=0.9

Итак, автомату необходимо (псевдо)случайным образом выбрать либо проигрыш с вероятностью 0.8 (u=0) либо S7x4, S1x3, S5x5 или S5x5 с соответствующим распределением вероятности. Для этого нам будет служить несложный алгоритм генерации чисел от 1 до 5 с соответствующими вероятностями, используя функцию, возвращающую равномерно распределенные от 0 до 1 псевдослучайные числа:

Отрезок от (0,1) делим на промежутки соответственно заданным при инициализации вероятностям. Нумеруем эти промежутки.

Генерируем случайное число стандартного генератора случайных чисел

Определяем номер промежутка, в котором оказалось сгенерированное псевдослучайное число. Возвращаем номер этого промежутка в качестве номера «выпавшего» события.

Итак, мы получили «случайный» исход, при этом «настроили» вероятности так, что матожидание выигрыша при ставке в один рубль составляет 0.9 рубля. Матрицу экрана мы генерируем исходя из того, какая комбинация выпала. Сначала заполняем призовые символы строки, остальные символы строки заполняем равномерно случайно. Строки, не принимающие участие в игре, будем заполнять так как будто каждая колонка является круговым барабаном с обозначениями S1,S2,..,S9.

В программе при 109 тестовых игр (компилятор gcc 4.8.4) матожидание выигрыша выходит 0.900005. Среднеквадратичное отклонением – 2.186, что согласуется с теоретическим подсчетом при использовании полученных в программе вероятностях выигрышных исходов.

## Исходники

В приложении:

Класс WeighedRandom – класс для генерации псевдослучайных исходов с весами

Класс LeverBalancer – класс «балансировщика» линейной системы – поиск положительных решений системы. Используя этот класс, мы определяем вероятности для таблицы выигрышей такие, что матожидание выходит равным 0.9.

Класс BaseSlotMachine – класс минимального игрового автомата

Machine.cpp – программа, сначала проводящая некоторое количество экспериментальных игр в автоматическом режиме для демонстрации того, что матожидание сходится к 0.9. Далее начинается игра – при нажатии Enter генерируется поле и рассчитывается выигрыш. Подразумевается, что каждый раз игрок делает ставку равную единице – которая и списывается со счета игрока. Средний выгрыш выводится ниже. При достаточно большом количестве партий он стремится к 0.9

Make.bash – bash-файл для компиляции