

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА (МИИТ)»
(РУТ (МИИТ))**

Одобрено кафедрой
«ЖЕЛЕЗНОДОРОЖНАЯ АВТОМАТИКА ТЕЛЕМЕХАНИКА И СВЯЗЬ»

Протокол № ____ от _____ 201__ г.
Автор: _____

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ С МЕТОДИЧЕСКИМИ
УКАЗАНИЯМИ**

ПО ДИСЦИПЛИНЕ

Моделирование систем управления

Уровень ВО: *Бакалавриат*

Форма обучения: *Заочная*

Курс: *4*

Специальность/Направление: *27.03.04 Управление в технических системах (УТб)*

Специализация/Профиль/Магистерская программа: *(УТ) Системы и технические средства автоматизации и управления*

Москва

ОБЩИЕ УКАЗАНИЯ

Курсовой проект посвящен проектированию имитационной модели работы сортировочной станции с заданными параметрами, а также статистической обработкой результатов моделирования. Варианты исходных данных к задачам студент выбирает по двум последним цифрам своего учебного шифра.

К выполнению курсового проекта рекомендуется приступать после изучения теоретической части курса в целом. При выполнении курсового проекта следует привести исходные данные в соответствии с вариантом, краткие пояснения, расчеты и схемы, предусмотренные методическими указаниями. Все схемы должны быть выполнены в графических редакторах. Результаты расчетов представляются в табличной форме, в соответствии с методическими указаниями.

Оформление курсового проекта выполняется на одной стороне стандартных листов А4, с использованием текстовых редакторов, в соответствии с требованиями по оформлению (14 шрифт, полуторный межстрочный интервал, поля справа – 3 см, слева – 3 см, сверху и снизу – 2 см). Листы должны быть сброшюрованы и пронумерованы. В конце контрольной работы необходимо привести список использованной литературы.

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Провести имитационное моделирование работы железнодорожной сортировочной станции, в которой производится расформирование составов на единственной сортировочной горке. В парке прибытия сортировочной станции имеется $N_{\text{приемн. путей}}$ путей, на которых составы ожидают роспуска. При этом станция должна выплачивать штраф в размере $S_{\text{штраф}}$ за один час простоя одного состава. Если эти пути заняты, то пришедший поезд вынужден покинуть эту станцию.

Время моделирования $T_{\text{мод}}$ принимается равным 24 часам.

Значения интервалов между приходящими поездами на станцию, которые необходимо расформировать, а также время роспуска состава на сортировочной горке являются случайными целыми числами, вид распределения которых и параметры этих распределений выбираются из табл. 1 и 2 согласно варианту.

Остальные исходные данные приведены в таблице 3.

По результатам имитационного моделирования необходимо провести их статистическую обработку и определить:

- вероятность отказа в роспуске поезду, при занятости всех приемных путей;
- среднее время пребывания состава на путях парка приема и среднее квадратичное отклонение;
- среднее число составов, ожидающих очереди на расформирование и среднее квадратичное отклонение;
- суммарный суточных штраф, который придется заплатить станции за простоя составов на приемных путях.

Моделирование случайных чисел с параметрами распределения, заданными согласно варианту, проводится на основании генератора базовых случайных чисел. Также должно быть приведено сравнение функций плотности распределения случайных чисел, полученных в результате

указанного моделирования, с функцией плотности этого же распределения случайных чисел, полученных встроенными функциями программной среды моделирования.

Таблица 1. Вид и параметры распределения значений интервалов между входящими на станцию поездами

№ предпоследней цифры шифра	Вид распределения	Параметры распределения
0	экспоненциальное	интенсивность: $\lambda = 0,03$ поезда/мин
1	нормальное	мат. ожидание: $M_x = 30$ мин дисперсия: $D_x = 10$
2	равномерное	минимальное значение: $a = 20$ мин максимальное значение: $b = 40$ мин
3	экспоненциальное	интенсивность: $\lambda = 0,02$ поезда/мин
4	нормальное	мат. ожидание: $M_x = 40$ мин дисперсия: $D_x = 15$
5	равномерное	минимальное значение: $a = 30$ мин максимальное значение: $b = 50$ мин
6	экспоненциальное	интенсивность: $\lambda = 0,025$ поезда/мин
7	нормальное	мат. ожидание: $M_x = 50$ мин дисперсия: $D_x = 20$
8	равномерное	минимальное значение: $a = 30$ мин максимальное значение: $b = 60$ мин
9	нормальное	мат. ожидание: $M_x = 60$ мин дисперсия: $D_x = 5$

Таблица 2. Вид и параметры распределения времени расформирования одного состава на сортировочной горке

№ последней цифры шифра	Вид распределения	Параметры распределения
0	равномерное	минимальное значение: $a = 10$ мин максимальное значение: $b = 15$ мин
1	равномерное	минимальное значение: $a = 10$ мин максимальное значение: $b = 20$ мин
2	равномерное	минимальное значение: $a = 10$ мин максимальное значение: $b = 25$ мин
3	равномерное	минимальное значение: $a = 15$ мин максимальное значение: $b = 20$ мин
4	равномерное	минимальное значение: $a = 15$ мин максимальное значение: $b = 25$ мин
5	нормальное	мат. ожидание: $M_x = 12$ мин дисперсия: $D_x = 5$
6	нормальное	мат. ожидание: $M_x = 15$ мин дисперсия: $D_x = 5$
7	нормальное	мат. ожидание: $M_x = 20$ мин дисперсия: $D_x = 5$
8	нормальное	мат. ожидание: $M_x = 20$ мин дисперсия: $D_x = 10$
9	нормальное	мат. ожидание: $M_x = 25$ мин дисперсия: $D_x = 5$

Таблица 3. Остальные исходные данные

№ последней цифры шифра	Число путей в парке приема, ($N_{\text{приемн. путей}}$)	Штраф за час простоя одного поезда ($S_{\text{штраф}}$), руб.
0	2	2000
1	2	2500
2	3	3000
3	3	3500
4	4	4000
5	2	2250
6	2	2750
7	3	3250
8	3	3750
9	4	4250

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Концептуальная и алгоритмическая модель работы сортировочной станции

Сортировочную станцию в рамках поставленной задачи можно представить в качестве системы массового обслуживания с ожиданием. В этом случае каналом обслуживания является сортировочная горка, а буфером данных – пути парка приема (рис. 1).

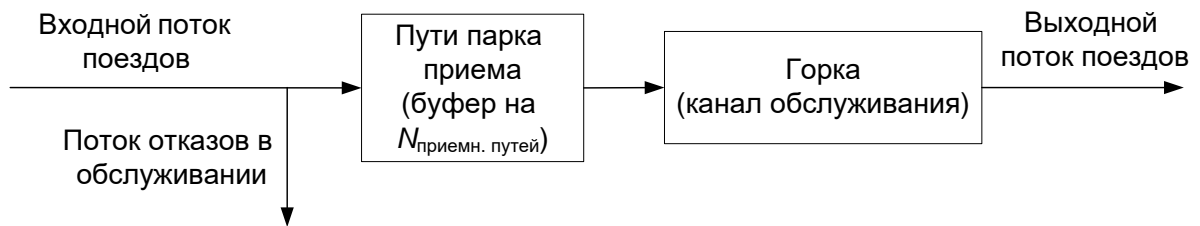


Рисунок 1. Концептуальная модель работы сортировочной станции

Поток отказов в обслуживании возникает в том случае, если все приемные пути заняты поездами (то есть все места в буфере заняты).

С учетом принятого допущения можно составить блок-схему имитационной модели работы сортировочной станции (рис. 2).

Для упрощения написания программы имитационного моделирования работы сортировочной станции составим на основании блок-схемы (рис. 2) следующий пошаговый алгоритм этой имитационной модели.

Шаг 1: ввести начальные параметры модели

Вид и параметры распределения случайных значений интервалов между входящими поездами;

Вид и параметры распределения случайных значений времени роспуска каждого состава;

Время моделирования: $T_{\text{мод}} = 1440$ мин.;

Количество путей в парке приема: $N_{\text{приемн. путей}} = 3$;

Штраф за один час простоя одного поезда на путях парка приема:

$$S_{\text{штрафа}} = 1000 \text{ руб.};$$

Размер финансовых потерь за отказ в расформировании одного поезда: $S_{\text{потерь}} = 9\,000 \text{ руб.};$

установка счетчиков:

$$\text{Количество отказов } N_{\text{отк}} = 0;$$

Моменты времени освобождения путей в парке приема:

$$T_{\text{освоб.пути}} [k] = 0, \text{ при } k=1,2,\dots, N_{\text{кан}};$$

Момент времени освобождения сортировочной горки:

$$T_{\text{освоб. горки}} = 0;$$

$$\text{Счетчик номера входящего поезда: } i = 1;$$

$$\text{Счетчик номера минуты моделирования: } \textit{CurrentMinute} = 0;$$

$$\text{Счетчик количества поездов в очереди на горку: } \textit{Ochered} = 0.$$

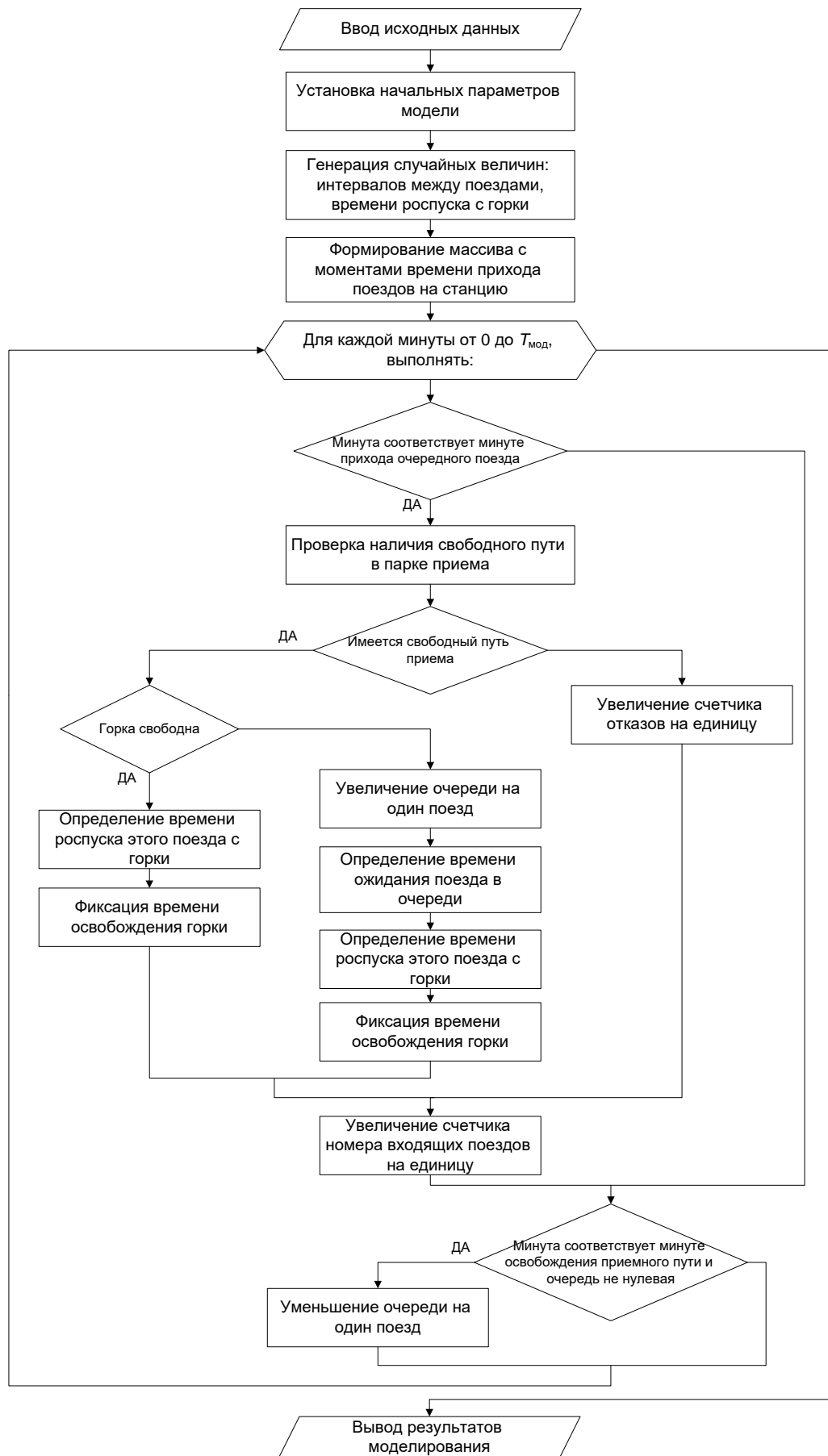


Рис. 2. Блок-схема имитационной модели работы сортировочной станции

Шаг 2: генерация значения интервалов между входящими поездами и формирования массива с этими значениями *IntervalR*.

Шаг 3: Генерация времени роспуска каждого состава и формирования массива данных этого значения *TimeRospuskaR*.

Шаг 4: Формирование массива с моментами времени прибытия каждого поезда на станцию *TimeLineR*.

Шаг 5: Проверка не окончено ли моделирование:

если текущая минута моделирования больше времени моделирования ($CurrentMinute < T_{\text{мод}}$),

то переход на шаг 23.

иначе увеличиваем счетчик номера минуты моделирования на единицу.

($CurrentMinute = CurrentMinute + 1$).

Шаг 6: Проверка, на прибытие поезда на станцию в текущую минуту моделирования:

если значение текущей минуты моделирования не совпадает с прибытием i -го поезда, $CurrentMinute \neq TimeLineR [i]$,

то переход на шаг 19.

Шаг 7: Проверка наличия свободного пути в парке приема:

если ($CurrentMinute < T_{\text{освоб. пути}} [k]$) для всех $k = 1, 2, \dots, N_{\text{приемн. путей}}$,

то переход к шагу шаг 16,

иначе зафиксировать номер свободного пути k .

Шаг 8: Проверка свободности горки в минуту прибытия i -го поезда на приемный путь:

если ($CurrentMinute < T_{\text{освоб. горки}}$),

то переход на шаг 12.

Шаг 9: Зафиксировать момент времени освобождения k -го пути и начала роспуска i -го состава:

$T_{\text{освоб. пути}} [k] = CurrentMinute$

$T_{\text{нач. роспуска}} = CurrentMinute$

Шаг 10: Определения времени отпуска i -го состава и фиксация момента времени освобождения горки:

$$T_{\text{освоб. горки}} = \text{CurrentMinute} + \text{TimeRospuskaR}[i]$$

Шаг 11: Переход на шаг 17.

Шаг 12: Увеличение счетчика очереди на один поезд:

$$\text{Ochered} = \text{Ochered} + 1$$

Шаг 13: Определение времени ожидания в очереди i -го состава и фиксация момента времени освобождения k -го приемного пути и начала отпуска i -го состава:

$$T_{\text{очереди}}[i] = T_{\text{освоб. горки}} - \text{CurrentMinute},$$

$$T_{\text{освоб. пути}}[k] = \text{CurrentMinute} + T_{\text{очереди}}[i] = T_{\text{освоб. горки}}$$

$$T_{\text{нач. отпуска}} = T_{\text{освоб. пути}}[k]$$

Шаг 14: Определения времени отпуска i -го состава и фиксация момента времени освобождения горки:

$$T_{\text{освоб. горки}} = T_{\text{освоб. горки}} + \text{TimeRospuskaR}[i]$$

Шаг 15: Переход на шаг 17.

Шаг 16: Увеличение счетчика отказов в отпуске на единицу:

$$N_{\text{отк}} = N_{\text{отк}} + 1$$

Шаг 17: Фиксация для i -го поезда момента времени прихода на станцию, момента времени начала отпуска состава с горки, времени отпуска состава и момента окончания отпуска состава с горки:

$$T_{\text{вх. на ст.}}[i] = \text{CurrentMinute}$$

$$T_{\text{нач. отпуска}}[i] = T_{\text{нач. отпуска}}$$

$$T_{\text{ропуска}}[i] = \text{TimeRospuskaR}[i]$$

$$T_{\text{оконч. отпуска}}[i] = T_{\text{освоб. горки}}$$

Шаг 18: Увеличение счетчика номера входящего поезда на единицу:

$$i = i + 1$$

Шаг 19: Проверка на соответствие текущей минуты моделирования моменту времени освобождения приемного пути при наличии в очереди на расформирование хотя бы одного поезда:

если ($CurrentMinute = T_{\text{освоб путей}} [k]$, при всех $k = 1, 2, \dots, N_{\text{приемн. путей}}$) И ($Ochered \neq 0$),

то переход к следующему шагу 20,

иначе переход к шагу 21.

Шаг 20: Уменьшение счетчика очереди на один поезд:

$$Ochered = Ochered - 1$$

Шаг 21: Фиксация количества проездов в очереди в текущую минуту моделирования:

$$Ochered[CurrentMinute] = Ochered$$

Шаг 22: Увеличение текущей минуты моделирования на единицу и переход к шагу 5:

$$CurrentMinute = CurrentMinute + 1$$

Шаг 23: Вывод результатов моделирования.

Моделирование случайных непрерывных чисел с заданными параметрами распределения

Значения интервалов между входящими поездами, а также значение времени роспуска состава с горки по условию задания являются случайными величинами, которые имеют заданные законы распределения (равномерный, экспоненциальный или нормальный). При этом предполагается, что инструмент моделирования имеет возможность генерировать случайные числа в пределах от 0 до 1, которые подчиняются равномерному закону распределения. Совокупность $\{R_i\}$, при $i = 1, 2, \dots$ независимых равномерно распределенных на отрезке $[0,1]$ случайных величин называется последовательностью базовых случайных чисел.

Необходимо провести моделирование случайных чисел с параметрами распределения, заданными согласно варианту, на основании генератора базовых случайных чисел.

Для большинства распределений случайных чисел случаев решение задачи основывается на методе инверсий.

Теорема: если случайная величина ξ имеет плотность распределения $p(x)$, то распределение случайной величины

$$\eta = \int_0^{\xi} p(x) dx, \quad (1)$$

является равномерным в интервале $(0,1)$.

Допустим, что требуется получить значения случайной величины $\{\xi_i\}$, распределенной в интервале (a, b) с заданной плотностью вероятности $p(x)$. Тогда согласно методу инверсий значения ξ_i можно определять из следующего интегрального уравнения:

$$\int_a^{\xi_i} p(x) dx = R_i, \quad (2)$$

где R_i – базовое случайное число.

Моделирование равномерно распределенной случайной величины на отрезке (a,b)

Пусть x – это случайное число, распределенное по равномерному закону в интервале (a, b) . Функция плотности распределения этой случайной величины имеет вид:

$$p(x) = \frac{1}{b-a}. \quad (3)$$

Математическое ожидание и дисперсия равномерного распределения соответственно равны:

$$M_x = \frac{a+b}{2}, \quad (4)$$

$$D_x = \frac{(b-a)^2}{12}. \quad (5)$$

Предположим, что нам необходимо составить программу для моделирования входного потока заявок распределенного по равномерному

закону в интервале (a, b) . Тогда уравнение метода инверсии (2) выглядит следующим образом:

$$\int_a^y \frac{dx}{b-a} = R . \quad (6)$$

Решением этого интегрального уравнения является:

$$\frac{y-a}{b-a} = R . \quad (7)$$

Тогда искомое случайное число определяется по формуле:

$$y = a + R(b-a) . \quad (8)$$

Моделирование экспоненциальной случайной величины

Случайная величина, подчиняющаяся экспоненциальному закону, распределена в интервале $(0, \infty)$ и имеет функцию плотности вероятности вида:

$$p(x) = \lambda \cdot e^{-\lambda \cdot x} , \quad (9)$$

где λ – параметр экспоненциального распределения.

Математическое ожидание M_x и дисперсия D_x экспоненциального распределения соответственно равны:

$$M_x = \frac{1}{\lambda} , \quad (10)$$

$$D_x = \frac{1}{\lambda^2} . \quad (11)$$

Для нахождения алгоритма имитации экспоненциально распределенных чисел применяется метод инверсии:

$$\int_0^y \lambda \cdot e^{-\lambda x} dx = R , \quad (12)$$

$$1 - e^{-\lambda y} = R , \quad (13)$$

$$y = -\frac{1}{\lambda} \ln(1-R) . \quad (14)$$

Поскольку случайная величина $(1-R)$ распределена точно так же, как R на отрезке $(0,1)$, то формулу (11) можно заменить:

$$y = -\frac{1}{\lambda} \ln R . \quad (15)$$

Моделирование нормальной случайной величины на основе центральной предельной теоремы

Нормальное (гауссово) распределение – это один из наиболее важных и часто используемых в имитационном моделировании видов непрерывных распределений.

Функция плотности вероятности нормального распределения случайной величины имеет следующий вид:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-M_x)^2}{2\sigma^2}} , \quad (16)$$

где M_x – математическое ожидание,

σ - средне квадратическое отклонение.

Метод инверсии не может быть использован для функций нормального распределения, так как не имеют аналитического выражения. Поэтому алгоритмы моделирования нормальных случайных чисел базируются на центральной предельной теореме теории вероятностей.

Теорема: Пусть совокупность $\{X_i\}$, при $i = 0,1, \dots$, одинаково распределенных случайных чисел с математическим ожиданием M_x и дисперсией $D_x = \sigma^2$. Если случайные величины $\{X_i\}$ независимы и $D_x < \infty$, то

при достаточно большом n закон распределения сумм $S_i = \sum_{k=i-n}^{i+n-1} X_k$ будет

сколь угодно близок к нормальному закону распределения с математическим ожиданием $n \cdot M_x$ и дисперсией $n \cdot D_x$.

Тогда используя базовые числа $\{R_i\}$, определим новую случайную величину:

$$S_1 = R_1 + R_2 \dots + R_n. \quad (17)$$

Тогда, согласно вышеуказанной теореме и формулам (4) и (5) математическое ожидание и дисперсия величины S равны:

$$M_s = \frac{n}{2}, \quad (18)$$

$$D_s = \frac{n}{12}, \quad (19)$$

Тогда случайная величина z , распределенная по нормальному закону с нулевым математическим ожиданием и единичной дисперсией, равна:

$$z = \frac{\left(S - \frac{n}{2}\right)}{\sqrt{\frac{n}{12}}}. \quad (20)$$

Пусть y является случайной непрерывной величиной, которая распределена по нормальному закону, и с математическим ожиданием M_y и дисперсией D_y . Тогда в соответствии с указанным выше n базовым случайным числом:

$$\frac{(y - M_y)}{\sqrt{D_y}} = z = \frac{\left(S - \frac{n}{2}\right)}{\sqrt{\frac{n}{12}}}, \quad (21)$$

$$y = M_y + \frac{\sqrt{D_y} \left(S - \frac{n}{2}\right)}{\sqrt{\frac{n}{12}}} = M_y + \frac{\sqrt{D_y}}{\sqrt{\frac{n}{12}}} \cdot \left(\sum_{i=1}^n R_i - \frac{n}{2}\right). \quad (22)$$

Согласно предельной теореме, нормальность достигается быстро даже при сравнительно небольших значениях n . В практических задачах n обычно принимается равным 12.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Получение плотности распределения случайных чисел с использованием встроенных функций программной среды моделирования

При моделировании случайных чисел с заданным распределением необходимо удостовериться, что случайные числа, полученные с использованием базовых случайных чисел, подчиняются закону распределения с заданными параметрами. Для этого сравниваются функции плотности распределения случайных чисел, полученных в результате указанного моделирования, с функцией плотности этого же распределения случайных чисел, полученных встроенными функциями программной среды моделирования. Пример такого сравнения для экспоненциального распределения с параметром $\lambda=2$ приведен на рис. 2.

```
Параметр:  $\lambda := 2$ 
Количество случайных чисел:  $N := 4000$ 
Интервал от 0 до N:  $interval := 0..N$ 
Точность построения функции:  $int_{interval} := 0.1 \cdot interval$ 
Генерация случайных чисел через
случайные базовые числа  $rand1_{interval} := \frac{-1}{\lambda} \cdot \ln(\text{rnd}(1))$ 
Генерация случайных чисел через
встроенные функции  $rand2 := \text{rexp}(N, \lambda)$ 
Функции плотности вероятности
случайной величины  $p1 := \frac{\text{hist}(int, rand1)}{N}$ 
 $p2 := \frac{\text{hist}(int, rand2)}{N}$ 
```

Рис. 2. Программная реализация сравнения функции плотности распределения случайных чисел

График плотностей распределений указанных случайных чисел приведен на рис. 3.

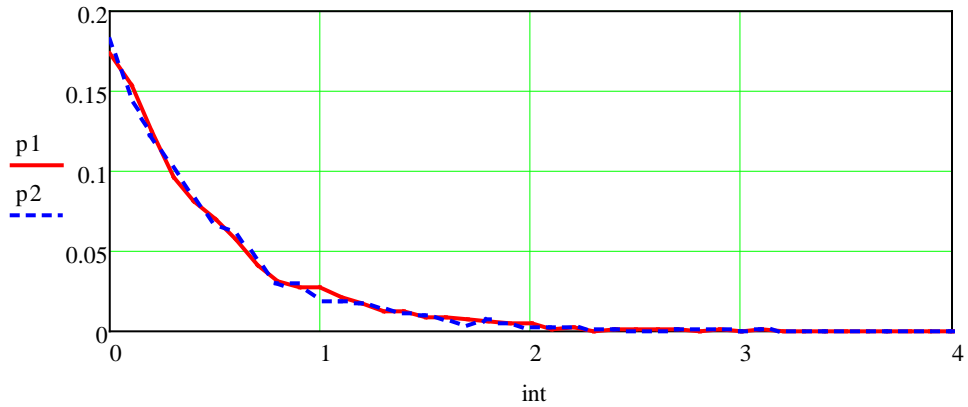


Рис. 3. График плотностей распределений случайных чисел экспоненциального распределения с параметром $\lambda=2$

Моделирование работы сортировочной станции

Моделирование работы сортировочной станции в данной работе проводится в программной среде MathCAD. Требуется реализовать алгоритм работы сортировочной станции, приведенный в параграфе 3, в синтаксисе программной среды.

Допустим, что уже были сформированы массивы значений интервалов между входящими поездами (*IntervalR*) и времени роспуска каждого состава каждого состава (*TimeRospuskaR*). В соответствии с алгоритмом требуется сформировать массив *TimeLineR* значений моментов времени прибытия на станцию каждого поезда. Это можно реализовать с помощью подпрограммы, отдельной от основной программы, приведенной на рис. 4.

$$\text{TimeLineR} := \left| \begin{array}{l} M_0 \leftarrow \text{IntervalR}_0 \\ \text{for } k \in 1..(\text{rows}(\text{IntervalR}) - 1) \\ \quad M_k \leftarrow M_{k-1} + \text{IntervalR}_k \\ M \end{array} \right.$$

Рис. 4. Подпрограмма для формирования массива значений моментов времени прибытия на станцию каждого поезда

В рамках основной программы, необходимо привести моделируемую систему к исходному состоянию. Для этого устанавливаются начальные значения переменным, используемым в системе (рис. 5):

$Notk = 0$ – счетчик количества отказов в обслуживании, когда на момент прибытия поезда на станцию все приемные пути заняты;

$TimeOsvobPriemPut[k] = 0$, для всех $k =$ от 0 до (количество приемных путей – 1) – массив, в котором индексом (k) является номер приемного пути, а значением является момент освобождения k -го пути (номер минуты);

$TimeOsvobGorki = 0$ – момент освобождения горки для роспуска следующего состава (номер минуты);

$ochered = 0$ – количество занятых приемных путей на текущую минуту моделирования;

$CurrentMinute = 0$ – номер текущей минуты моделирования;

$i = 0$ – счетчик поездов прибывающих на станцию.

```

Notk ← 0
for k ∈ 0..(N_priem_put - 1)
    TimeOsvobPriemPutk ← 0
TimeOsvobGorki ← 0
ochered ← 0
CurrentMinute ← 0
i ← 0

```

Рис. 5. Приведение системы к исходному состоянию

Программная реализация шагов №5 и №6 приведена на рис. 6. В этом случае для проверки на окончание моделирование (шаг №5) и организации итерационного процесса моделирования используется цикл с параметром (оператор *for*), в котором последовательно изменяется номер текущей минуты моделирования ($CurrentMinute$) на единицу после каждого выполнения тела этого цикла.

```

for CurrentMinute ∈ 0.. Tmod
  if CurrentMinute = TimeLineR1
    шаг№7
    ...
    шаг№18
  шаги№19 – 22

```

Рис. 6. Программная реализация шагов №5-6

Программная реализация проверки на наличие свободного приемного пути в момент прибытия поезда на станцию приведена, на рис. 7. Переменная *free_priem_put* является флагом наличия или отсутствия свободного приемного пути. То есть после выполнения приведенного фрагмента программы эта переменная примет значение 1 если имеется свободный путь, иначе переменная примет значение 0. Переменная *k_put_zanyatiya* будет содержать номер свободного пути, если *free_priem_put=1*.

```

k ← 0
k_put_zanyatiya ← 0
free_priem_put ← 0
while (free_priem_put = 0) ∧ (k < N_priem_put)
  if TimeOsvobPriemPutk < TimeLineR1
    free_priem_put ← 1
    k_put_zanyatiya ← k
  k ← k + 1

```

Рис. 7. Проверка наличия свободного приемного пути в момент прибытия очередного поезда на станцию

Согласно алгоритму если приемные пути заняты, то необходимо увеличить счетчик количества отказов на единицу (рис. 8). Для этого случая фиксируются следующие переменные: момент времени перехода поезда на сортировочную горку ($T_{per_na_gorku}=0$), время расформирования ($T_{rasform}=0$) и момент времени выхода из сортировочной горки ($T_{vyhoda_iz_gorki}=TimeLineR[i]$). Указанные переменные в дальнейшем

потребуется для формирования массива данных с результатами моделирования, которые будут использоваться для статистической обработки.

```

if free_priem_put = 1
    T_per_na_gorku ← TimeLineR1 if TimeLineR1 > TimeOsvobGorki
    otherwise
        T_per_na_gorku ← TimeOsvobGorki
        ochered ← ochered + 1
        TimeOsvobPriemPutk_put_zanyatiya ← T_per_na_gorku
        T_rasform ← TimeRospuskaRi
        T_vyhoda_iz_gorki ← T_per_na_gorku + T_rasform
        TimeOsvobGorki ← T_vyhoda_iz_gorki
    otherwise
        Notk ← Notk + 1
        T_per_na_gorku ← 0
        T_rasform ← 0
        T_vyhoda_iz_gorki ← TimeLineR1

```

Рис. 8. Программная реализация шагов №№7-16

Если же хотя бы один путь на момент прибытия поезда свободен, то сначала проводится проверка на свободу сортировочной горки для роспуска пришедшего состава в этот же момент времени (сопоставление переменных $TimeOsvobGorki$ и $TimeLineR[i]$). Если горка в рассматриваемый момент свободна, то в переменную $T_per_na_gorku$ записывается значение равное моменту времени прибытия поезда на станцию ($TimeLineR[i]$). Иначе эта переменная примет значение равное моменту времени освобождения сортировочной горки ($T_per_na_gorku=TimeOsvobGorki$).

Затем фиксируются значения переменных: время расформирования ($T_rasform=TimeRospuskaR[i]$) и момент времени выхода из сортировочной горки ($T_vyhoda_iz_gorki = T_per_na_gorku + T_rasform$), который также равен моменту освобождения горки ($TimeOsvobGorki=T_vyhoda_iz_gorki$).

После этого считается, что моделирование обслуживания сортировочной станцией одного i -го поезда окончено, и дополняется

итоговой массив данных ключевыми временными параметрами этого поезда (момент времени прибытия на станцию, момент времени перехода поезда на горку, фактическое время расформирования состава на горке, момент выхода поезда с сортировочной горки). Также увеличивается на единицу счетчик номеров входящих поездов на станцию (рис. 9).

$$\left| \begin{array}{l} M_{i,0} \leftarrow i + 1 \\ M_{i,1} \leftarrow \text{TimeLineR}_1 \\ M_{i,2} \leftarrow T_{\text{per_na_gorku}} \\ M_{i,3} \leftarrow T_{\text{rasform}} \\ M_{i,4} \leftarrow T_{\text{vyhoda_iz_gorki}} \\ i \leftarrow i + 1 \end{array} \right.$$

Рис. 9. Программная реализация шагов №№17-18

В конце каждой итерации цикла минут моделирования необходимо фиксировать количество поездов, ожидающих в эту минуту на приемных путях своей очереди на расформирование с горки (шаги №№19-21). Сначала проводится проверка на совпадение минуты освобождения приемных путей каким-либо поездом (номер которого меньше i -го) с номером моделируемой минуты. Причем, если такой совпадение имеет место, то значение не нулевой очереди (*ochered*) уменьшается на единицу (рис. 10).

$$\left| \begin{array}{l} \text{for } j \in 0..(\text{rows}(M) - 1) \qquad \qquad \qquad \text{if } \text{ochered} > 0 \\ \text{ochered} \leftarrow \text{ochered} - 1 \text{ if } \text{CurrentMinute} = M_{j,2} \\ \text{Ochered}_{\text{CurrentMinute}} \leftarrow \text{ochered} \end{array} \right.$$

Рис. 10. Программная реализация шагов №№19-21

На этом программная реализация алгоритма моделирования работы сортировочной станции окончена. Полный листинг программы представлен в приложении 1.

Статистическая обработка результатов моделирования

Выходные данные имитационного эксперимента представляющие собой, как правило, массивы случайных чисел, описывают соответствующие операционные характеристики модели. Как известно, исчерпывающей характеристикой случайной величины X является ее закон распределения вероятностей $F(x)$ – правило, позволяющее определять вероятность попадания этой величины в любую заданную область ее значений.

Чтобы найти закон распределения экспериментальным путем, нужно располагать достаточно обширным статистическим материалом: порядка нескольких тысяч опытов (или наблюдений, экспериментов). На практике редко удается собрать такое количество экспериментального материала, а иногда это и не требуется – достаточно определить только оценки некоторых основных характеристик (параметров) закона распределения, таких как математическое ожидание, дисперсию. Поэтому необходимо уметь получать статистические оценки при ограниченных опытных данных.

Задачи статистической обработки экспериментальных данных, как правило, решаются в условиях, когда вид закона распределения анализируемой случайной величины не известен. Однако бывают исключения. Так, если вид закона распределения случайной величины известен заранее, то задача статистической обработки упрощается и заключается только в нахождении некоторых параметров закона распределения, которые полностью определяют соответствующий закон распределения.

Например, если априори известно, что закон распределения случайной величины X нормальный (гауссовский), то задача обработки данных сводится к определению двух его параметров: математического ожидания (M_x), – средне квадратического отклонения (σ_x).

В некоторых задачах имитационного моделирования вид закона распределения несущественен, а требуется знать только его числовые

характеристики (параметры). Задачи такого рода называются задачами параметрической оценки.

Следует отметить, что любое значение искомого параметра, вычисленное на основе ограниченного числа экспериментальных данных, всегда будет содержать элемент случайности, а значит и ошибку. Такое приближенное, случайное значение будем называть оценкой параметра. Например, оценкой для математического ожидания случайной величины X может служить среднее арифметическое наблюдаемых значений в n независимых опытах. При очень большом количестве опытов среднее арифметическое будет с большой вероятностью весьма близко к истинному значению математического ожидания. Если же число опытов n невелико, то замена математического ожидания средним арифметическим приводит к некоторой ошибке. То есть эта ошибка обратно пропорциональна числу опытов. Аналогичная ситуация и с оценками других неизвестных параметров.

Математическое ожидание m_x случайной величины $X = \{ x_1, x_2, \dots, x_n \}$ называется сумма произведений всех n возможных значений случайной величины на вероятности p_i этих значений:

$$m_x = \sum_{i=1}^n x_i p_i . \quad (23)$$

В случае гауссовского (нормального) распределения для истинного математического ожидания m_x существует его оценка \tilde{m}_x , вычисляемая по выборке объема n случайной величины $X = (x_1, x_2, \dots, x_n)$:

$$\tilde{m}_x = \frac{1}{n} \sum_{i=1}^n x_i . \quad (24)$$

Для истинной дисперсии D_x (характеристика рассеивания случайной величины около ее математического ожидания) ее оценка \tilde{D}_x при известном математическом ожидании m_x вычисляется так:

$$\tilde{D}_x = \sigma_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - m_x)^2 , \quad (25)$$

где σ_x является средне квадратическим отклонением.

В случае неизвестного математического ожидания дисперсию \tilde{D}_x нужно вычислять по формуле:

$$\tilde{D}_x = \frac{1}{n-1} \left[\sum_{i=1}^n x_i^2 - \frac{1}{n} \left(\sum_{i=1}^n x_i \right)^2 \right]. \quad (26)$$

По приведенным формулам производится расчет требуемых параметров модели (работы сортировочной станции).

Пусть в массиве данных *ResultModel* содержатся временные характеристики входящих на станцию поездов за время моделирования. Тогда вероятность отказа в роспуске поезду, при занятости всех приемных путей:

$$\text{Potkaza} := \frac{\text{Notk}}{\text{rows}(\text{ResultModel})}$$

Для определения среднего времени пребывания состава на путях парка приема и среднего квадратичного отклонения примем N_{obsl} – число составов, которые за время моделирования были расформированы с сортировочной горки:

$$N_{obsl} := \text{rows}(\text{ResultModel}) - \text{Notk}$$

$$M_{vremya_prebyvaniya} := \frac{1}{N_{obsl}} \left[\sum_{k=0}^{\text{rows}(\text{ResultModel})-1} \left(\begin{cases} \text{ResultModel}_{k,2} - \text{ResultModel}_{k,1} & \text{if } \text{ResultModel}_{k,2} > 0 \\ 0 & \text{otherwise} \end{cases} \right) \right]$$

$$\sigma_{vremya_prebyvaniya} := \sqrt{\frac{1}{N_{obsl}-1} \left[\sum_{k=0}^{\text{rows}(\text{ResultModel})-1} \left[\begin{cases} (\text{ResultModel}_{k,2} - \text{ResultModel}_{k,1})^2 & \text{if } \text{ResultModel}_{k,2} > 0 \\ 0 & \text{otherwise} \end{cases} \right] - \frac{1}{N_{obsl}} \cdot (N_{obsl} \cdot M_{vremya_prebyvaniya})^2 \right]}$$

Среднее число составов, ожидающих очереди на расформирование и её среднее квадратичное отклонение:

$$M_{chislo_sostavov} := \frac{1}{T_{mod}} \cdot \sum_{i=0}^{\text{rows}(\text{TimelineOchered})-1} \text{TimelineOchered}_i$$

$$\sigma_{chislo_sostavov} := \sqrt{\frac{1}{T_{mod}-1} \left[\sum_{i=0}^{\text{rows}(\text{TimelineOchered})-1} (\text{TimelineOchered}_i)^2 - \frac{1}{T_{mod}} \cdot (T_{mod} \cdot M_{chislo_sostavov})^2 \right]}$$

Суммарный суточных штраф, который придется заплатить станции за простой:

$$M_{\text{shtraf}} := \frac{M_{\text{вре́мя_пре́бывания}} \cdot S_{\text{shtraf}}}{60}$$

где S_{shtraf} – стоимость за час простоя одного поезда.

Заключение

По результатам статистической обработки необходимо сформулировать заключение о возможной необходимости увеличения или уменьшения приемных путей, о возможной необходимости увеличения или уменьшения интервала между поездами для увеличения эффективности сортировочной станции.

Список используемых источников

1. Советов Б.А., Яковлев С.А. Моделирование систем. Учеб. для вузов — М.: Высшая школа, 2007. — 343 с.
2. Дунин-Барковский И.В., Смирнов Н.В. Теория вероятностей и математическая статистика в технике (общая часть). Гостехиздат, 1955.
3. Тюрин Ю.Н., Макаров А.А. Статистический анализ данных на компьютере. М., ИНФРА., 1998.

Приложение 1

Программа моделирования сортировочной станции

```
TimeLineR := | M0 ← IntervalR0
              | for k ∈ 1..(rows(IntervalR) - 1)
              |   Mk ← Mk-1 + IntervalRk
              | M

ModelSortSt := | Notk ← 0
               | for k ∈ 0..(N_priem_put - 1)
               |   TimeOsvobPriemPutk ← 0
               | TimeOsvobGorki ← 0
               | ochered ← 0
               | CurrentMinute ← 0
               | i ← 0
               | for CurrentMinute ∈ 0..Tmod
               |   if CurrentMinute = TimeLineRi
               |     k ← 0
               |     k_put_zanyatiya ← 0
               |     free_priem_put ← 0
               |     while (free_priem_put = 0) ∧ (k < N_priem_put)
               |       if TimeOsvobPriemPutk < TimeLineRi
               |         free_priem_put ← 1
               |         k_put_zanyatiya ← k
               |         k ← k + 1
               |     if free_priem_put = 1
               |       T_per_na_gorku ← TimeLineRi if TimeLineRi > TimeOsvobGorki
               |       otherwise
               |         T_per_na_gorku ← TimeOsvobGorki
               |         ochered ← ochered + 1
               |       TimeOsvobPriemPutk_put_zanyatiya ← T_per_na_gorku
               |       T_rasform ← TimeRospuskaRi
               |       T_vyhoda_iz_gorki ← T_per_na_gorku + T_rasform
               |       TimeOsvobGorki ← T_vyhoda_iz_gorki
```

```

otherwise
  Notk ← Notk + 1
  T_per_na_gorku ← 0
  T_rasform ← 0
  T_vyhoda_iz_gorki ← TimeLineRi
Mi,0 ← i + 1
Mi,1 ← TimeLineRi
Mi,2 ← T_per_na_gorku
-----
Mi,3 ← T_rasform
Mi,4 ← T_vyhoda_iz_gorki
i ← i + 1
for j ∈ 0..(rows(M) - 1) if ochered > 0
  ochered ← ochered - 1 if CurrentMinute = Mj,2
OcheredCurrentMinute ← ochered

```

$$\begin{pmatrix} M \\ \text{Ochered} \\ \text{Notk} \end{pmatrix}$$

ResultModel := ModelSortSt₀

TimelineOchered := ModelSortSt₁

Notk := ModelSortSt₂

Приложение 2

Пример выполнения курсового проекта

1 Постановка задачи и выбор исходных данных

Провести имитационное моделирование работы железнодорожной сортировочной станции, в которой производится расформирование составов на единственной сортировочной горке. В парке прибытия сортировочной станции имеется $N_{\text{приемн. путей}}$, на которых составы ожидают отпуска. При этом станция должна выплачивать штраф в размере $S_{\text{штраф}}$ за один час простоя одного состава. Если эти пути заняты, то пришедший поезд вынужден покинуть эту станцию.

Время моделирования $T_{\text{мод}}$ принимается равным 24 часам.

Значения интервалов между приходящими поездами на станцию, которые необходимо расформировать, а также время отпуска состава на сортировочной горке являются случайными целыми числами, вид распределения которых и параметры этих распределений выбираются из табл. 1 и 2 согласно варианту.

Остальные исходные данные приведены в таблице 3.

По результатам имитационного моделирования необходимо провести их статистическую обработку и определить:

- вероятность отказа в роспуске поезду, при занятости всех приемных путей;
- среднее время пребывания состава на путях парка приема и среднее квадратичное отклонение;
- среднее число составов, ожидающих очереди на расформирование и среднее квадратичное отклонение;
- суммарный суточных штраф, который придется заплатить станции за простой составов на приемных путях.

Моделирование случайных чисел с параметрами распределения, заданными согласно варианту, проводится на основании генератора базовых случайных чисел. Также должно быть приведено сравнение функций

плотности распределения случайных чисел, полученных в результате указанного моделирования, с функцией плотности этого же распределения случайных чисел, полученных встроенными функциями программной среды моделирования.

Вид и параметры распределения значений интервалов между входящими на станцию поездами

Вид распределения	Параметры распределения
экспоненциальное	интенсивность: $\lambda = 0,03$ поезда/мин

Вид и параметры распределения времени расформирования одного состава на сортировочной горке

Вид распределения	Параметры распределения
равномерное	минимальное значение: $a = 15$ мин максимальное значение: $b = 20$ мин

Число путей в парке приема, ($N_{\text{приемн. путей}}$)	Штраф за час простоя одного поезда ($S_{\text{штраф}}$), руб.
3	3500

3 Концептуальная и алгоритмическая модель работы сортировочной станции

Сортировочную станцию в рамках поставленной задачи можно представить в качестве системы массового обслуживания с ожиданием. В этом случае каналом обслуживания является сортировочная горка, а буфером данных – пути парка приема (рис. 3.1).

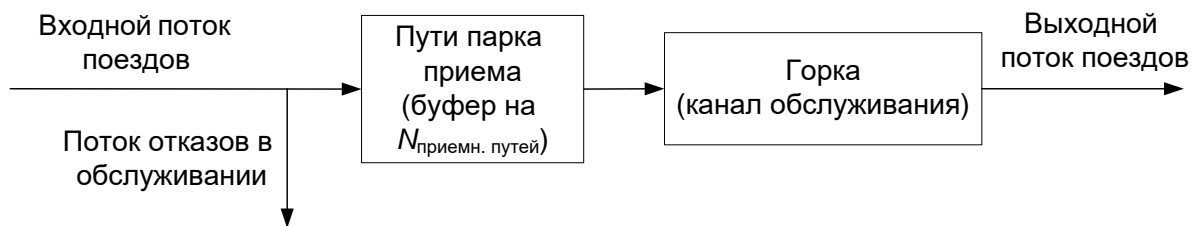


Рисунок 3.1. Концептуальная модель работы сортировочной станции

Поток отказов в обслуживании возникает в том случае, если все приемные пути заняты поездами (то есть все места в буфере заняты).

С учетом принятого допущения можно составить блок-схему имитационной модели работы сортировочной станции (рис. 3.2).

Для упрощения написания программы имитационного моделирования работы сортировочной станции составим на основании блок-схемы (рис. 3.2) следующий пошаговый алгоритм этой имитационной модели.

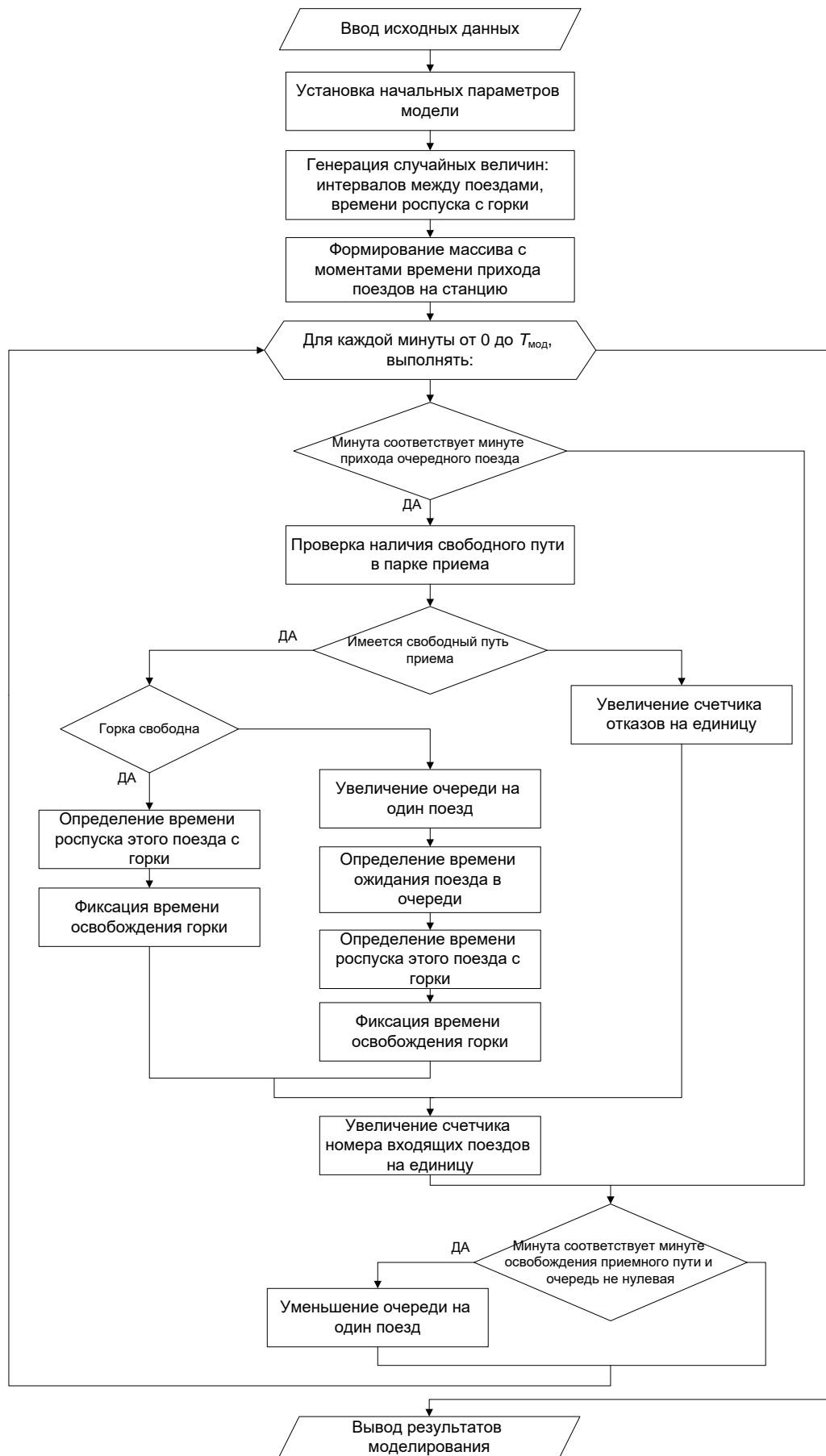


Рис. 3.2. Блок-схема имитационной модели работы сортировочной станции

Шаг 1: ввести начальные параметры модели

Вид и параметры распределения случайных значений интервалов между входящими поездами;

Вид и параметры распределения случайных значений времени роспуска каждого состава;

Время моделирования: $T_{\text{мод}} = 1440$ мин.;

Количество путей в парке приема: $N_{\text{приемн. путей}} = 3$;

Штраф за один час простоя одного поезда на путях парка приема:
 $S_{\text{штрафа}} = 3500$ руб.;

установка счетчиков:

Количество отказов $N_{\text{отк}} = 0$;

Моменты времени освобождения путей в парке приема:
 $T_{\text{освоб.пути}}[k] = 0$, при $k=1,2,\dots, N_{\text{кан}}$;

Момент времени освобождения сортировочной горки:
 $T_{\text{освоб. горки}} = 0$;

Счетчик номера входящего поезда: $i = 1$;

Счетчик номера минуты моделирования: $CurrentMinute = 0$;

Счетчик количества поездов в очереди на горку: $Ochered = 0$.

Шаг 2: генерация значения интервалов между входящими поездами и формирования массива с этими значениями *IntervalR*.

Шаг 3: Генерация времени роспуска каждого состава и формирования массива данных этого значения *TimeRospuskaR*.

Шаг 4: Формирование массива с моментами времени прибытия каждого поезда на станцию *TimeLineR*.

Шаг 5: Проверка не окончено ли моделирование:

если текущая минута моделирования больше времени моделирования
($CurrentMinute < T_{\text{мод}}$),

то переход на шаг 23.

иначе увеличиваем счетчик номера минуты моделирования на единицу.

($CurrentMinute = CurrentMinute + 1$).

Шаг 6: Проверка, на прибытие поезда на станцию в текущую минуту моделирования:

если значение текущей минуты моделирования не совпадает с прибытием i -го поезда, $CurrentMinute \neq TimeLineR [i]$,

то переход на шаг 19.

Шаг 7: Проверка наличия свободного пути в парке приема:

если ($CurrentMinute < T_{освоб. пути [k]}$) для всех $k = 1, 2, \dots, N_{приемн. путей}$,

то переход к шагу шаг 16,

иначе зафиксировать номер свободного пути k .

Шаг 8: Проверка свободности горки в минуту прибытия i -го поезда на приемный путь:

если ($CurrentMinute < T_{освоб. горки}$),

то переход на шаг 12.

Шаг 9: Зафиксировать момент времени освобождения k -го пути и начала роспуска i -го состава:

$$T_{освоб. пути [k]} = CurrentMinute$$

$$T_{нач. роспуска} = CurrentMinute$$

Шаг 10: Определения времени роспуска i -го состава и фиксация момента времени освобождения горки:

$$T_{освоб. горки} = CurrentMinute + TimeRospuskaR[i]$$

Шаг 11: Переход на шаг 17.

Шаг 12: Увеличение счетчика очереди на один поезд:

$$Ochered = Ochered + 1$$

Шаг 13: Определение времени ожидания в очереди i -го состава и фиксация момента времени освобождения k -го приемного пути и начала роспуска i -го состава:

$$T_{очереди[i]} = T_{освоб. горки} - CurrentMinute,$$

$$T_{освоб. пути [k]} = CurrentMinute + T_{очереди[i]} = T_{освоб. горки}$$

$$T_{нач. роспуска} = T_{освоб. пути [k]}$$

Шаг 14: Определения времени отпуска i -го состава и фиксация момента времени освобождения горки:

$$T_{\text{освоб. горки}} = T_{\text{освоб. горки}} + \text{TimeRospuskaR}[i]$$

Шаг 15: Переход на шаг 17.

Шаг 16: Увеличение счетчика отказов в распуске на единицу:

$$N_{\text{отк}} = N_{\text{отк}} + 1$$

Шаг 17: Фиксация для i -го поезда момента времени прихода на станцию, момента времени начала отпуска состава с горки, времени отпуска состава и момента окончания отпуска состава с горки:

$$T_{\text{вх. на ст.}}[i] = \text{CurrentMinute}$$

$$T_{\text{нач. отпуска}}[i] = T_{\text{нач. отпуска}}$$

$$T_{\text{ропуска}}[i] = \text{TimeRospuskaR}[i]$$

$$T_{\text{оконч. отпуска}}[i] = T_{\text{освоб. горки}}$$

Шаг 18: Увеличение счетчика номера входящего поезда на единицу:

$$i = i + 1$$

Шаг 19: Проверка на соответствие текущей минуты моделирования моменту времени освобождения приемного пути при наличии в очереди на расформирование хотя бы одного поезда:

если ($\text{CurrentMinute} = T_{\text{освоб путей}}[k]$, при всех $k = 1, 2, \dots, N_{\text{приемн. путей}}$) И ($\text{Ochered} \neq 0$),

то переход к следующему шагу 20,

иначе переход к шагу 21.

Шаг 20: Уменьшение счетчика очереди на один поезд:

$$\text{Ochered} = \text{Ochered} - 1$$

Шаг 21: Фиксация количества проездов в очереди в текущую минуту моделирования:

$$\text{Ochered}[\text{CurrentMinute}] = \text{Ochered}$$

Шаг 22: Увеличение текущей минуты моделирования на единицу и переход к шагу 5:

$$\text{CurrentMinute} = \text{CurrentMinute} + 1$$

Шаг 23: Вывод результатов моделирования.

4 Моделирование случайных непрерывных чисел с заданными параметрами распределения

Экспоненциальная случайная величина моделируется по формуле:

$$y = -\frac{1}{\lambda} \ln R$$

Примем $N = 3000$ – число точек, $Nm = 30$ число интервалов для построения плотности распределения случайной величины.

Интервалы между входящими на станцию поездами

Экспоненциальное $\lambda := 0.03$

$N := 3000$ $kk := 0..N$

$Nm := 30$ - число интервалов $mm := 0..Nm$

Массив интервалов: $h := \frac{300}{Nm} = 10$ $int_{mm} := h \cdot mm$

Генерация случайных чисел через случайные базовые числа:

$$rand1_{kk} := \frac{-1}{\lambda} \cdot \ln(md(1))$$

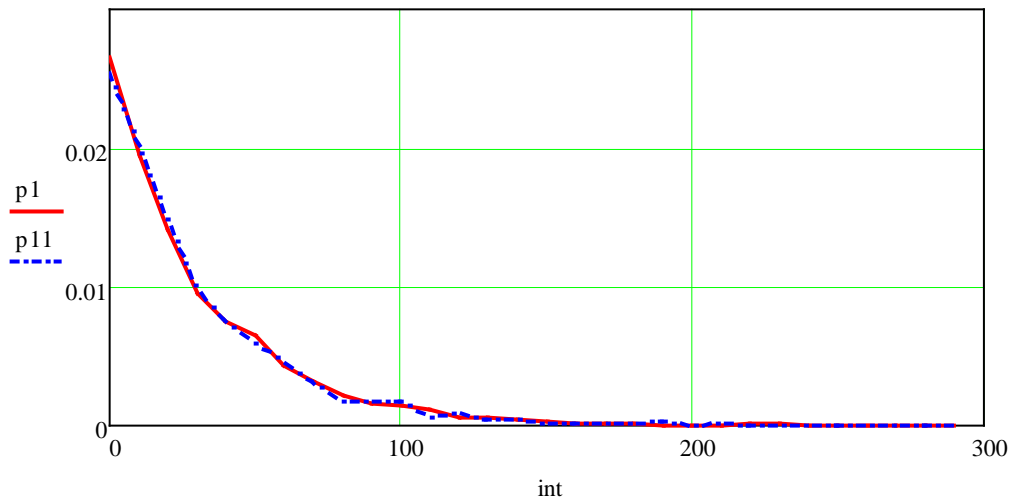
Генерация случайных чисел через встроенные функции:

$rand11 := rexp(N, \lambda)$

Функции плотности вероятности случайной величины:

$$p1 := \frac{hist(int, rand1)}{h N} \quad p11 := \frac{hist(int, rand11)}{h N}$$

График сравнения функций плотности распределения случайных чисел, полученных в результате указанного моделирования, с функцией плотности этого же распределения случайных чисел, полученных встроенными функциями программной среды моделирования:



Равномерная случайная величина моделируется по формуле:

$$y = a + R(b - a)$$

Примем $N = 3000$ – число точек, $Nm = 50$ число интервалов для построения плотности распределения случайной величины.

Время расформирования одного состава на сортировочной горке

Равномерное $a := 15$ $b := 20$ минут

$Nm := 50$ - число интервалов $mm := 0..Nm$

Массив интервалов: $h := \frac{b - a}{Nm} = 0.1$ $int_{mm} := a + h \cdot mm$

Генерация случайных чисел через случайные базовые числа:

$$rand2_{kk} := a + md(1) \cdot (b - a)$$

Генерация случайных чисел через встроенные функции:

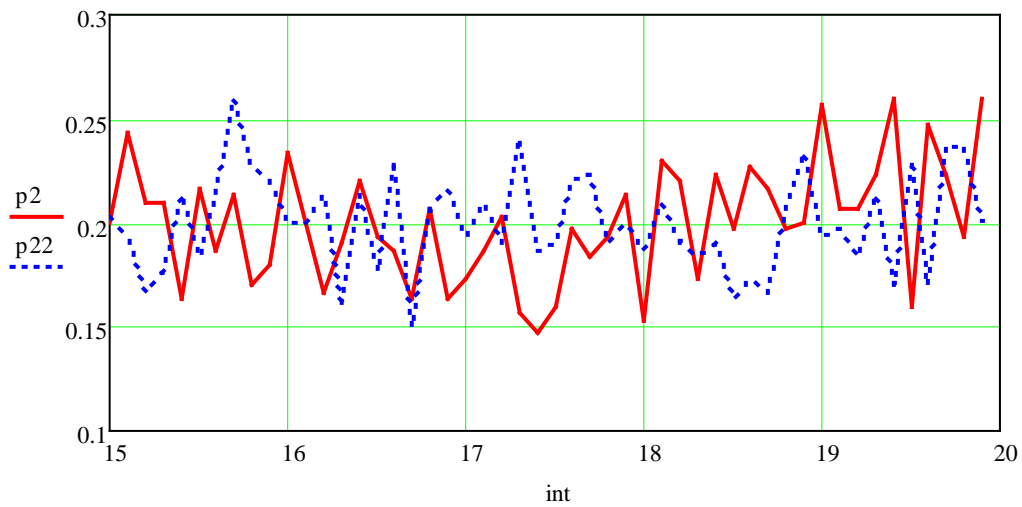
$$rand22 := runif(N, a, b)$$

Функции плотности вероятности случайной величины:

$$p2 := \frac{hist(int, rand2)}{h \cdot N} \quad p22 := \frac{hist(int, rand22)}{h \cdot N}$$

График сравнения функций плотности распределения случайных чисел, полученных в результате указанного моделирования, с функцией плотности

этого же распределения случайных чисел, полученных встроенными функциями программной среды моделирования:



Поскольку дальнейший алгоритм использует целочисленный счётчик времени прибытия поездов на станцию, то необходимо округлить сформированные массивы случайных чисел.

```
IntervalR := ceil(rand1)
```

```
TimeRospuskaR := ceil(rand2)
```

5. Моделирование работы сортировочной станции

Моделирование работы сортировочной станции в данной работе проводится в программной среде MathCAD. Требуется реализовать алгоритм работы сортировочной станции, приведенный в параграфе 3, в синтаксисе программной среды.

Допустим, что уже были сформированы массивы значений интервалов между входящими поездами (*IntervalR*) и времени роспуска каждого состава каждого состава (*TimeRospuskaR*). В соответствии с алгоритмом требуется сформировать массив *TimeLineR* значений моментов времени прибытия на станцию каждого поезда. Это можно реализовать с помощью подпрограммы, отдельной от основной программы, приведенной на рис. 5.1.

$$\text{TimeLineR} := \left\{ \begin{array}{l} M_0 \leftarrow \text{IntervalR}_0 \\ \text{for } k \in 1..(\text{rows}(\text{IntervalR}) - 1) \\ M_k \leftarrow M_{k-1} + \text{IntervalR}_k \\ M \end{array} \right.$$

Рис. 5.1. Подпрограмма для формирования массива значений моментов времени прибытия на станцию каждого поезда

В рамках основной программы, необходимо привести моделируемую систему к исходному состоянию. Для этого устанавливаются начальные значения переменным, используемым в системе (рис. 5.2):

Notk = 0 – счетчик количества отказов в обслуживании, когда на момент прибытия поезда на станцию все приемные пути заняты;

TimeOsvobPriemPut[*k*] = 0, для всех *k* = от 0 до (количество приемных путей – 1) – массив, в котором индексом (*k*) является номер приемного пути, а значением является момент освобождения *k*-го пути (номер минуты);

TimeOsvobGorki = 0 – момент освобождения горки для роспуска следующего состава (номер минуты);

ochered = 0 – количество занятых приемных путей на текущую минуту моделирования;

$CurrentMinute = 0$ – номер текущей минуты моделирования;

$i = 0$ – счетчик поездов прибывающих на станцию.

```
Notk ← 0
for k ∈ 0.. (N_priem_put - 1)
    TimeOsvobPriemPutk ← 0
TimeOsvobGorki ← 0
ochered ← 0
CurrentMinute ← 0
i ← 0
```

Рис. 5.2. Приведение системы к исходному состоянию

Программная реализация шагов №5 и №6 приведена на рис. 5.3. В этом случае для проверки на окончание моделирование (шаг №5) и организации итерационного процесса моделирования используется цикл с параметром (оператор *for*), в котором последовательно изменяется номер текущей минуты моделирования ($CurrentMinute$) на единицу после каждого выполнения тела этого цикла.

```
for CurrentMinute ∈ 0.. Tmod
    if CurrentMinute = TimeLineR1
        шаг№7
        ...
        шаг№18
    шаги№19 – 22
```

Рис. 5.3. Программная реализация шагов №5-6

Программная реализация проверки на наличие свободного приемного пути в момент прибытия поезда на станцию приведена, на рис. 5.4. Переменная $free_priem_put$ является флагом наличия или отсутствия свободного приемного пути. То есть после выполнения приведенного фрагмента программы эта переменная примет значение 1 если имеется свободный путь, иначе переменная примет значение 0. Переменная $k_put_zanyatiya$ будет содержать номер свободного пути, если $free_priem_put=1$.


```

k ← 0
k_put_zanyatiya ← 0
free_priem_put ← 0
while (free_priem_put = 0) ∧ (k < N_priem_put)
    if TimeOsvobPriemPutk < TimeLineRi
        free_priem_put ← 1
        k_put_zanyatiya ← k
    k ← k + 1

```

Рис. 5.4. Проверка наличия свободного приемного пути в момент прибытия очередного поезда на станцию

Согласно алгоритму если приемные пути заняты, то необходимо увеличить счетчик количества отказов на единицу (рис. 5.5). Для этого случая фиксируются следующие переменные: момент времени перехода поезда на сортировочную горку ($T_{per_na_gorku}=0$), время расформирования ($T_{rasform}=0$) и момент времени выхода из сортировочной горки ($T_{vyhoda_iz_gorki}=TimeLineR[i]$). Указанные переменные в дальнейшем потребуются для формирования массива данных с результатами моделирования, которые будут использоваться для статистической обработки.

```

if free_priem_put = 1
    T_per_na_gorku ← TimeLineRi if TimeLineRi > TimeOsvobGorki
    otherwise
        T_per_na_gorku ← TimeOsvobGorki
        ochered ← ochered + 1
    TimeOsvobPriemPutk_put_zanyatiya ← T_per_na_gorku
    T_rasform ← TimeRospuskaRi
    T_vyhoda_iz_gorki ← T_per_na_gorku + T_rasform
    TimeOsvobGorki ← T_vyhoda_iz_gorki
otherwise
    Notk ← Notk + 1
    T_per_na_gorku ← 0
    T_rasform ← 0
    T_vyhoda_iz_gorki ← TimeLineRi

```

Рис. 5.5. Программная реализация шагов №№7-16

Если же хотя бы один путь на момент прибытия поезда свободен, то сначала проводится проверка на свободу сортировочной горки для роспуска пришедшего состава в этот же момент времени (сопоставление переменных $TimeOsvobGorki$ и $TimeLineR[i]$). Если горка в рассматриваемый момент свободна, то в переменную $T_per_na_gorku$ записывается значение равное моменту времени прибытия поезда на станцию ($TimeLineR[i]$). Иначе эта переменная примет значение равное моменту времени освобождения сортировочной горки ($T_per_na_gorku=TimeOsvobGorki$).

Затем фиксируются значения переменных: время расформирования ($T_rasform=TimeRospuskaR[i]$) и момент времени выхода из сортировочной горки ($T_vyhoda_iz_gorki = T_per_na_gorku + T_rasform$), который также равен моменту освобождения горки ($TimeOsvobGorki=T_vyhoda_iz_gorki$).

После этого считается, что моделирование обслуживания сортировочной станцией одного i -го поезда окончено, и дополняется итоговой массив данных ключевыми временными параметрами этого поезда (момент времени прибытия на станцию, момент времени перехода поезда на горку, фактическое время расформирования состава на горке, момент выхода поезда с сортировочной горки). Также увеличивается на единицу счетчик номеров входящих поездов на станцию (рис. 5.6).

$$\begin{array}{l}
 M_{i,0} \leftarrow i + 1 \\
 M_{i,1} \leftarrow TimeLineR_i \\
 M_{i,2} \leftarrow T_per_na_gorku \\
 M_{i,3} \leftarrow T_rasform \\
 M_{i,4} \leftarrow T_vyhoda_iz_gorki \\
 i \leftarrow i + 1
 \end{array}$$

Рис. 5.6. Программная реализация шагов №№17-18

В конце каждой итерации цикла минут моделирования необходимо фиксировать количество поездов, ожидающих в эту минуту на приемных путях своей очереди на расформирование с горки (шаги №№19-21). Сначала проводится проверка на совпадение минуты освобождения приемных путей каким-либо поездом (номер которого меньше i -го) с номером моделируемой минуты. Причем, если такой совпадение имеет место, то значение не нулевой очереди (*ochered*) уменьшается на единицу (рис. 5.7).

```

for j ∈ 0..(rows(M) - 1)           if ochered > 0
  ochered ← ochered - 1 if CurrentMinute = Mj,2
OcheredCurrentMinute ← ochered

```

Рис. 5.7. Программная реализация шагов №№19-21

На этом программная реализация алгоритма моделирования работы сортировочной станции окончена.

6. Статистическая обработка результатов моделирования

Выходные данные имитационного эксперимента представляющие собой, как правило, массивы случайных чисел, описывают соответствующие операционные характеристики модели. Как известно, исчерпывающей характеристикой случайной величины X является ее закон распределения вероятностей $F(x)$ – правило, позволяющее определять вероятность попадания этой величины в любую заданную область ее значений.

Чтобы найти закон распределения экспериментальным путем, нужно располагать достаточно обширным статистическим материалом: порядка нескольких тысяч опытов (или наблюдений, экспериментов). На практике редко удается собрать такое количество экспериментального материала, а иногда это и не требуется – достаточно определить только оценки некоторых основных характеристик (параметров) закона распределения, таких как математическое ожидание, дисперсию. Поэтому необходимо уметь получать статистические оценки при ограниченных опытных данных.

Задачи статистической обработки экспериментальных данных, как правило, решаются в условиях, когда вид закона распределения анализируемой случайной величины не известен. Однако бывают исключения. Так, если вид закона распределения случайной величины известен заранее, то задача статистической обработки упрощается и заключается только в нахождении некоторых параметров закона распределения, которые полностью определяют соответствующий закон распределения.

Например, если априори известно, что закон распределения случайной величины X нормальный (гауссовский), то задача обработки данных сводится к определению двух его параметров: математического ожидания (M_x), – средне квадратического отклонения (σ_x).

В некоторых задачах имитационного моделирования вид закона распределения несущественен, а требуется знать только его числовые

характеристики (параметры). Задачи такого рода называются задачами параметрической оценки.

Следует отметить, что любое значение искомого параметра, вычисленное на основе ограниченного числа экспериментальных данных, всегда будет содержать элемент случайности, а значит и ошибку. Такое приближенное, случайное значение будем называть оценкой параметра. Например, оценкой для математического ожидания случайной величины X может служить среднее арифметическое наблюдаемых значений в n независимых опытах. При очень большом количестве опытов среднее арифметическое будет с большой вероятностью весьма близко к истинному значению математического ожидания. Если же число опытов n невелико, то замена математического ожидания средним арифметическим приводит к некоторой ошибке. То есть эта ошибка обратно пропорциональна числу опытов. Аналогичная ситуация и с оценками других неизвестных параметров.

Математическое ожидание m_x случайной величины $X = \{ x_1, x_2, \dots, x_n \}$ называется сумма произведений всех n возможных значений случайной величины на вероятности p_i этих значений:

$$m_x = \sum_{i=1}^n x_i p_i . \quad (6.1)$$

В случае гауссовского (нормального) распределения для истинного математического ожидания m_x существует его оценка \tilde{m}_x , вычисляемая по выборке объема n случайной величины $X = (x_1, x_2, \dots, x_n)$:

$$\tilde{m}_x = \frac{1}{n} \sum_{i=1}^n x_i . \quad (6.2)$$

Для истинной дисперсии D_x (характеристика рассеивания случайной величины около ее математического ожидания) ее оценка \tilde{D}_x при известном математическом ожидании m_x вычисляется так:

$$\tilde{D}_x = \sigma_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - m_x)^2 , \quad (6.3)$$

где σ_x является средне квадратическим отклонением.

В случае неизвестного математического ожидания дисперсию \tilde{D}_x нужно вычислять по формуле:

$$\tilde{D}_x = \frac{1}{n-1} \left[\sum_{i=1}^n x_i^2 - \frac{1}{n} \left(\sum_{i=1}^n x_i \right)^2 \right]. \quad (6.4)$$

По приведенным формулам производится расчет требуемых параметров модели (работы сортировочной станции).

Пусть в массиве данных *ResultModel* содержатся временные характеристики входящих на станцию поездов за время моделирования. Тогда вероятность отказа в роспуске поезду, при занятости всех приемных путей:

$$Potkaza := \frac{Notk}{rows(ResultModel)}$$

Для определения среднего времени пребывания состава на путях парка приема и среднего квадратичного отклонения примем N_{obs1} – число составов, которые за время моделирования были расформированы с сортировочной горки:

$$Nobs1 := rows(ResultModel) - Notk$$

$$Mvremya_prebyvaniya := \frac{1}{Nobs1} \left[\sum_{k=0}^{rows(ResultModel)-1} \left(\begin{cases} ResultModel_{k,2} - ResultModel_{k,1} & \text{if } ResultModel_{k,2} > 0 \\ 0 & \text{otherwise} \end{cases} \right) \right]$$

$$\sigma vremya_prebyvaniya := \sqrt{\frac{1}{Nobs1-1} \left[\sum_{k=0}^{rows(ResultModel)-1} \left[\begin{cases} (ResultModel_{k,2} - ResultModel_{k,1})^2 & \text{if } ResultModel_{k,2} > 0 \\ 0 & \text{otherwise} \end{cases} \right] - \frac{1}{Nobs1} \cdot (Nobs1 \cdot Mvremya_prebyvaniya)^2 \right]}$$

Среднее число составов, ожидающих очереди на расформирование и её среднее квадратичное отклонение:

$$Mchislo_sostavov := \frac{1}{Tmod} \cdot \sum_{i=0}^{rows(TimelineOchered)-1} TimelineOchered_i$$

$$\sigma chislo_sostavov := \sqrt{\frac{1}{Tmod-1} \left[\sum_{i=0}^{rows(TimelineOchered)-1} (TimelineOchered_i)^2 - \frac{1}{Tmod} \cdot (Tmod \cdot Mchislo_sostavov)^2 \right]}$$

Суммарный суточных штраф, который придется заплатить станции за простой:

$$Mshtraf := \frac{Mvremya_prebyvaniya \cdot S_shtraf}{60}$$

где S_{shtraf} – стоимость за час простоя одного поезда.

При одной из реализаций случайных чисел получены следующие результаты:

Число отказов:

$$Notk := ModelSortSt_2 = 1$$

Вероятность отказа:

$$Potkaza := \frac{Notk}{rows(ResultModel)} = 0.03$$

Число обслуженных составов:

$$Nobs1 := rows(ResultModel) - Notk = 32$$

Среднее время пребывания на путях парка приёма:

$$Mvremya_prebyvaniya = 10.906 \text{ минут}$$

Среднее квадратическое отклонение времени пребывания на путях парка приёма:

$$\sigma vremya_prebyvaniya = 14.072 \text{ минут}$$

Среднее число составов, ожидающих очереди:

$$Mchislo_sostavov = 0.242$$

Среднее квадратическое отклонение числа составов, ожидающих очереди:

$$\sigma Mchislo_sostavov = 0.589$$

Суммарный суточный штраф:

$$Mshtraf := \frac{Mvremya_prebyvaniya \cdot S_shtraf}{60}$$

$$Mshtraf = 636.198$$

Таким образом за сутки будут обслужены 32 из 33 прибывших составов, среднее время ожидания (пребывания состава на путях парка приема) составит примерно 10.9 минут.

Заключение

Поскольку вероятность отказа небольшая (около 3%), то рекомендация по увеличению приёмных путей будет зависеть от стоимости её реализации.

В любом случае рекомендуется немного уменьшить среднее время расформирования одного состава на сортировочной горке.