



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Домашняя работа №1
по курсу «Моделирование»
«Применение методов имитационного и стохастического
моделирования»

Студент группы ИУ9-81Б Окутин Д. А.

Преподаватель Домрачева А. Б.

Москва 2025

Цель работы

Целью данной работы является изучение принципов построения марковских моделей, а также изучение возможностей языка GPSS (General Purpose Simulation System). Применение данных видов моделей для демонстрации процесса сдачи домашнего задания студентами. Сравнение результатов этих моделей между собой.

Постановка задачи

Студенты группы получили домашнюю работу. Вероятность того, что студент не приступал к выполнению $p_1 = 0,3$, приступил к выполнению работы, но выполнил с ошибками $p_2 = 0,4$. После сдачи работа проверяется преподавателем, и, в случае обнаружения ошибок, возвращается студенту для исправлений.

Вероятность выполнить работу успешно после возврата (возвраты могут оказаться множественными) постоянна и определяется как $p_3 = 0,8$. Необходимо оценить количество дней, требующихся на успешную сдачу домашнего задания с момента выдачи (количество дней, реально затраченных студентом на выполнение задания, а преподавателем на проверку работы, не учитывать; считается, что действия происходят в течение дня).

Для генераций значений в диапазоне $[0,1]$ нужно использовать генератор равномерно распределенных в этом диапазоне случайных величин. Построить имитационную модель систем массового обслуживания и реализовать в среде GPSS. Сравнить полученный результат с оценкой количества дней, требующихся на успешную сдачу домашнего задания, полученной по марковской модели.

Теоретические сведения

Система массового обслуживания (СМО) — система, предназначенная для многократно повторяющегося (многоразового) использования при решении однотипных задач. Это модели систем, в которые в случайные моменты времени извне или изнутри поступают заявки (требования), которые должны тем или иным образом быть обслужены системой. Длительность обслуживания чаще всего случайна.

Имитационное моделирование является эффективным инструментом для изучения систем массового обслуживания, в которых случайные события и вероятностные распределения играют ключевую роль. Этот метод позволяет воспроизводить поведение сложных процессов, а также оценивать эффективность системы. С помощью имитационного моделирования можно проводить эксперименты с различными сценариями, что особенно актуально для систем, имеющих циклические или повторяющиеся этапы.

Язык GPSS (General Purpose Systems Simulator) является одним из специализированных средств для построения имитационных моделей систем массового обслуживания. Описание на языке GPSS есть совокупность операторов (блоков), характеризующих процессы обработки заявок. Имеются операторы и для отображения возникновения заявок, задержки их в ОА, занятия памяти, выхода из СМО, изменения параметров заявок (например, приоритетов), вывода на печать накопленной информации, характеризующей загрузку устройств и заполненность очередей. Существуют операторы, с помощью которых можно изменять значения любых параметров транзактов, и операторы, характер исполнения которых зависит от значений того или иного параметра обслуживаемого транзакта.

Стохастическое моделирование — это метод, при котором в модели учитываются случайные факторы. Такие модели используются для прогнозирования результатов событий на основе вероятностных расчётов и случайных величин. Они основаны на моделировании случайных событий с помощью различных алгоритмов и статистических методов. Стохастическое моделирование осуществляется методами математической статистики.

Марковская модель представляет собой стохастическую модель, описывающую динамику системы, в которой будущее состояние зависит только от текущего, а не от предыдущей истории. Это свойство, называемое *марковским свойством* или свойством отсутствия памяти. В марковской модели переходы между состояниями описываются матрицей переходных вероятностей $P = (p_{ij})$, где элемент p_{ij} означает вероятность перехода из состояния i в состояние j .

Практическая реализация

Построим модель Маркова для нашей задачи, где время измеряется в днях, а каждый день соответствует одной попытке выполнения задания.

Состояния системы:

- S_1 — начальное состояние, в котором студент приступает к выполнению задания.
- S_2 — состояние исправления работы, выполненной с ошибками.
- S_3 — состояние успешной сдачи задания (поглощающее состояние).

Вероятности переходов: Из S_1 :

- С вероятностью $p_1 = 0.3$ студент не приступает к выполнению задания, то есть остаётся в состоянии S_1 .
- С вероятностью $p_2 = 0.4$ студент выполняет работу с ошибками и переходит в состояние S_2 .
- С вероятностью $p_3 = 1 - p_1 - p_2 = 0.3$ студент выполняет задание без ошибок, значит, работа сдана успешно и переходит сразу в состояние S_3 .

Вероятности переходов: Из S_2 :

- С вероятностью $p_4 = 0.8$ студент исправляет ошибки и успешно сдаёт работу, т.е. переходит в состояние S_3 .
- С вероятностью $p_5 = 1 - p_4 = 0.2$ студент не исправляет работу и остаётся в состоянии S_2 .

Представление модели в виде графа указано на рисунке 1.

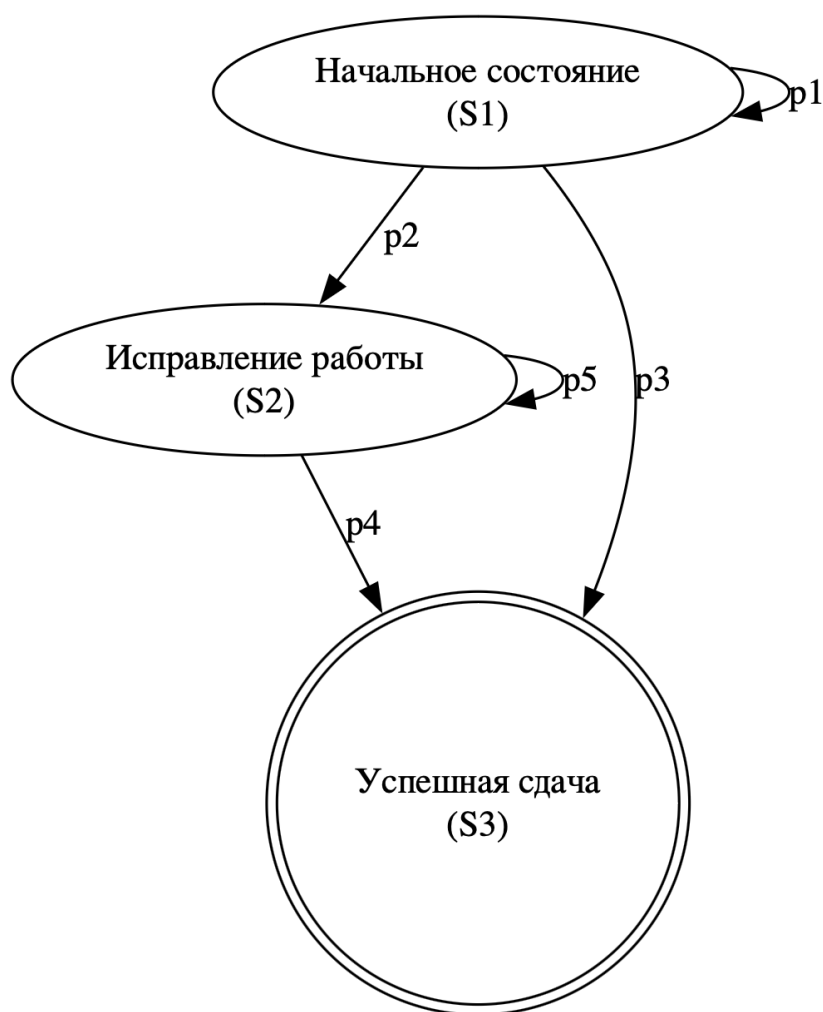


Рис. 1 — Визуализация Марковской модели

Таким образом, таблица переходов P имеет вид:

S_1	S_2	S_3
0.3	0.4	0.3
0	0.2	0.8
0	0	1

Для расчетов среднего количества дней с помощью Марковской модели проведем некоторое количество экспериментов и изобразим их графически.

Для этого был написан код на языке Python, который проходит по нашему автомату, учитывая вероятности событий и генерирует некоторую цепочку состояний, моделирующую процесс сдачи домашнего задания студентом.

Исходный код программы представлен в листинге 1.

Листинг 1: Исходный код

```
1 import numpy as np
2
3 def generate_sequence(p1, p2, p3, p4, p5, max_steps=100):
4     state = 'S1'
5     sequence = [state]
6
7     while state != 'S3':
8         if state == 'S1':
9             transitions = np.random.choice(['S1', 'S2', 'S3'], p=[p1, p2
10 , p3])
11         elif state == 'S2':
12             transitions = np.random.choice(['S2', 'S3'], p=[p5, p4])
13
14         state = transitions
15         sequence.append(state)
16
17         if len(sequence) > max_steps:
18             break
19
20     return sequence
21
22 def generate_sequences(num_sequences, p1, p2, p3, p4, p5):
23     sequences = [generate_sequence(p1, p2, p3, p4, p5) for _ in range(
24 num_sequences)]
25     return sequences
26
27 p1, p2, p3 = 0.3, 0.4, 0.3
28 p4, p5 = 0.8, 0.2
29
30 num_sequences = 100
31 sequences = generate_sequences(num_sequences, p1, p2, p3, p4, p5)
```

Эксперимент 1: S1 → S2 → S3
 Эксперимент 2: S1 → S2 → S3
 Эксперимент 3: S1 → S1 → S1 → S1 → S1 → S1 → S2 → S3
 Эксперимент 4: S1 → S3
 Эксперимент 5: S1 → S2 → S3
 Эксперимент 6: S1 → S2 → S2 → S3
 Эксперимент 7: S1 → S2 → S2 → S3
 Эксперимент 8: S1 → S2 → S2 → S3
 Эксперимент 9: S1 → S2 → S2 → S2 → S3
 Эксперимент 10: S1 → S1 → S3

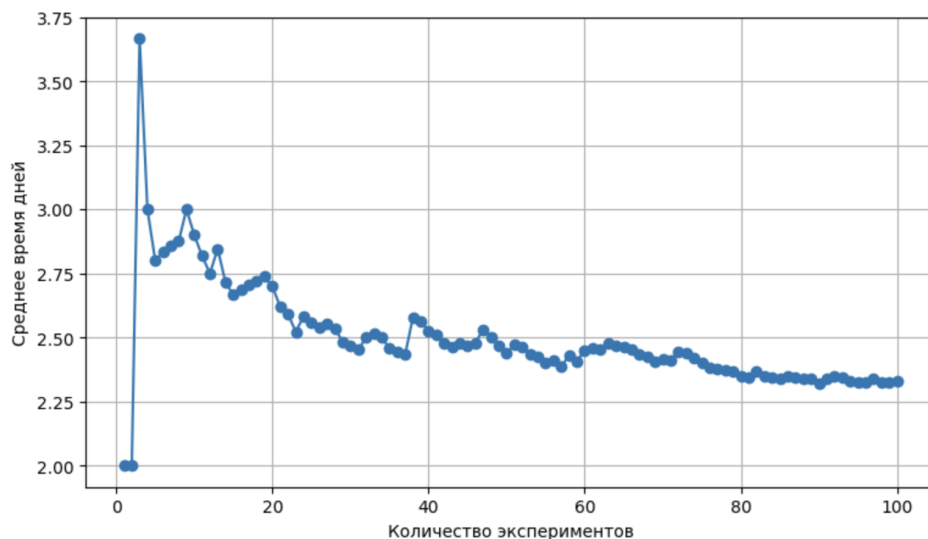


Рис. 2 — Результаты моделирования

Вывод: Из графика видно, что в среднем для успешной сдачи домашнего задания, требуется в районе 2.2 дня (т.к значения начинают стабилизироваться примерно у этой отметки). Следовательно отсюда можно сделать вывод, что среднее количество потраченных дней на сдачу работы составляет 3 дня.

Перейдём к реализации имитационной модели. Исходный код представляет собой реализацию имитационной модели на языке GPSS, которая моделирует процесс сдачи домашнего задания студентом.

В модели описываются переходы между состояниями, который совершает студент в зависимости он вероятности событий, который описывались в постановке задачи. При переходе студента на новый день сдачи модель увеличивает целочисленный счётчик дней для конкретного транзакта, а при успешной сдаче сохраняет этот счётчик в таблицу.

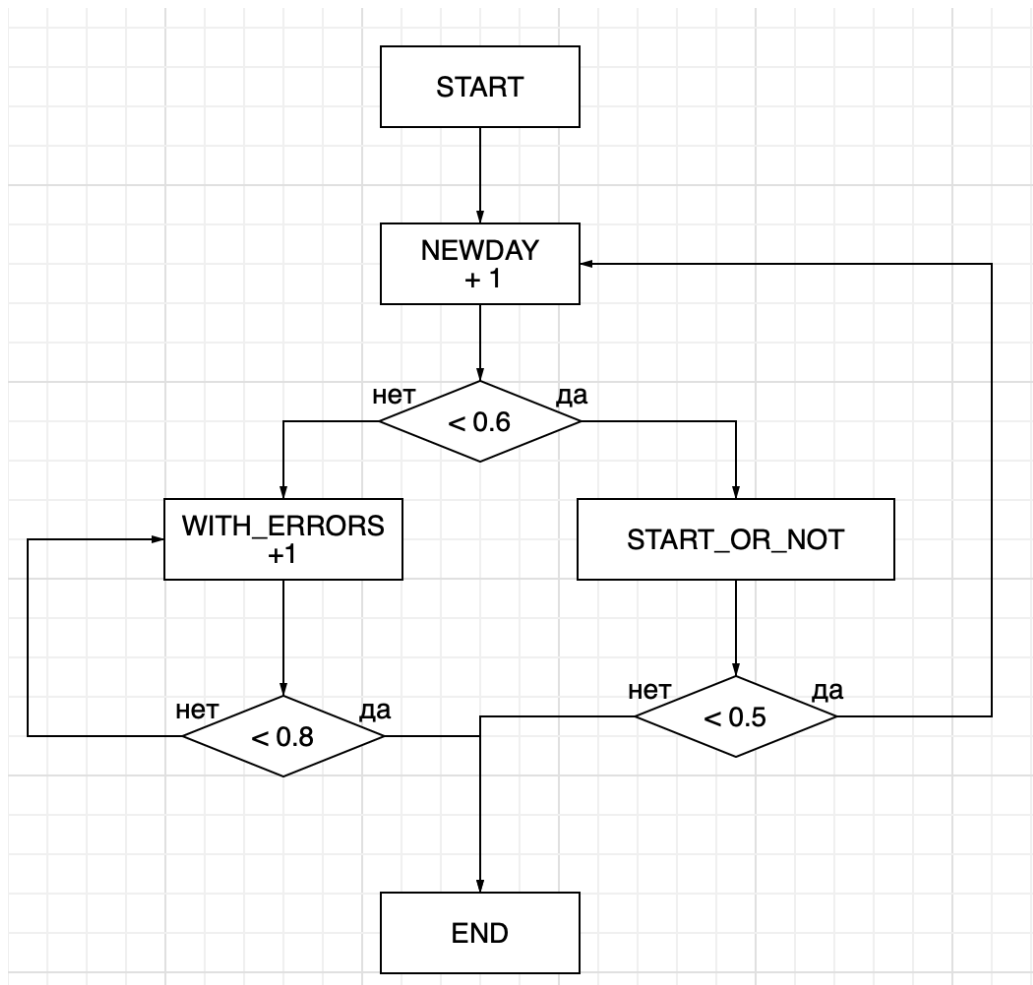


Рис. 3 — Схема GPSS модели

Исходный код программы представлен в листинге 2.

Листинг 2: Исходный код

```

1
2 SIMULATE
3
4 DAYS TABLE P1,0,1,23
5 GENERATE 1 ASSIGN 1,0
6
7 NEWDAY ASSIGN 1+,1
8     TRANSFER .6,WITH_ERRORS,START_OR_NOT
9
10 WITH_ERRORS ASSIGN 1+,1
11     TRANSFER .8,WITH_ERRORS,SUCCESS
12
13 START_OR_NOT TRANSFER .5,NEWDAY,SUCCESS
14 SUCCESS TABULATE DAYS,P1
15     TERMINATE 1
16

```


В начале создаем таблицу "DAYS где $P1$ — переменная, а значения 0, 1, 23 обозначают начальные значения столбцов. В данном случае, $P1$ будет отслеживать количество дней:

```
DAYS TABLE P1,0,1,23
```

Начинаем генерировать 1 случайный путь сдачи задания, устанавливая значение $P1$ равным 0.

```
GENERATE 1 ASSIGN 1,0
```

Увеличиваем значение $P1$ на 1, обозначая начало нового дня. И с вероятностью 0.6 перенаправляем в состояние WITH_ERRORS, иначе — в состояние START_OR_NOT.

```
NEWDAY ASSIGN 1+,1  
TRANSFER .6, WITH\_ERRORS, START\_OR\_NOT
```

Увеличиваем значение $P1$ на 1 в состоянии WITH_ERRORS. С вероятностью 0.8 остаемся в состоянии WITH_ERRORS, иначе переходим в состояние SUCCESS.

```
WITH\_ERRORS ASSIGN 1+,1  
TRANSFER .8, WITH\_ERRORS, SUCCESS
```

С вероятностью 0.5 возвращаемся в состояние NEWDAY, иначе переходим в состояние SUCCESS.

```
START\_OR\_NOT TRANSFER .5, NEWDAY, SUCCESS
```

Записываем текущее значение $P1$ в таблицу "DAYS".

```
SUCCESS TABULATE DAYS, P1
```

Листинг 3: Вывод результатов моделирования GPSS

1										
2	START TIME		END TIME		BLOCKS	FACILITIES		STORAGES		
3	0.000		1000.000		9	0		0		
4	NAME		VALUE							
5	DAYS		10000.000							
6	NEWDAY		3.000							
7	START_OR_NOT		7.000							
8	SUCCESS		8.000							
9	WITH_ERRORS		5.000							
10										
11										
12										
13	LABEL	LOC	BLOCK	TYPE	ENTRY COUNT	CURRENT COUNT	RETRY			
			1	GENERATE		1000	0			
	0									
14		2	ASSIGN		1000		0	0		
15	NEWDAY	3	ASSIGN		1412		0	0		
16		4	TRANSFER		1412		0	0		
17	WITH_ERRORS	5	ASSIGN		714		0	0		
18		6	TRANSFER		714		0	0		
19	START_OR_NOT	7	TRANSFER		824		0	0		
20	SUCCESS	8	TABULATE		1000		0	0		
21		9	TERMINATE		1000		0	0		
22										
23										
24	TABLE	MEAN	STD.DEV.	RANGE		RETRY FREQUENCY	CUM.%			
25	DAYS	2.677	1.252			0				
26			0.000	-	1.000	306	14.39			
27			1.000	-	2.000	846	54.19			
28			2.000	-	3.000	489	77.19			
29			3.000	-	4.000	260	89.42			
30			4.000	-	5.000	170	97.41			
31			5.000	-	6.000	48	99.67			
32			6.000	-	7.000	7	100.00			
33										
34	FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE		
35	1001	0	1001.000	1001	0	1				

Результаты работы

В результате моделирования было проведено несколько прогонов для оценки среднего количества дней, необходимого для успешной сдачи домашней работы. Полученные результаты продемонстрировали хорошее согласование между результатами имитационного моделирования и расчётом по марковской модели. При заданных параметрах:

$$p_1 = 0.3, \quad p_2 = 0.4, \quad p_3 = 0.3, \quad p_4 = 0.8, \quad p_5 = 0.2,$$

среднее количество дней, рассчитанное по стохастической модели, составило примерно 2.2 дней, а имитационная модель дала средний результат, близкий к этому значению 2.67 дней. Т.е можно сделать вывод, что в среднем уходит 3 дня.

Заключение

В рамках домашней работы была разработана имитационная модель процесса сдачи домашнего задания, реализованная в среде GPSS, с использованием равномерного генератора случайных чисел для определения исходов каждого этапа.

Анализ марковской модели позволил получить оценку среднего количества дней, необходимых для успешной сдачи задания. Сравнение результатов показало, что оба подхода дают сопоставимые оценки, что подтверждает корректность построенных моделей и выбранных вероятностных характеристик. Однако расхождения в моделях всё же были получены, так как в Марковской модели не учли каких-то факторов или было осуществлено недостаточное количество прогонов.

Таким образом, проведённое исследование продемонстрировало эффективность применения имитационных моделей для анализа процессов массового обслуживания с повторными попытками и позволило усовершенствовать навыки моделирования сложных систем.