О.А. Петухов
А.В. Морозов
Е.О. Петухова

МОДЕЛИРОВАНИЕ

СИСТЕМНОЕ ИМИТАЦИОННОЕ АНАЛИТИЧЕСКОЕ

УЧЕБНОЕ ПОСОБИЕ

Санкт-Петербург
Издательство СЗТУ
2008

Федеральное агентство по образованию

Государственное образовательное учреждение высшего профессионального образования

Северо-Западный государственный заочный технический университет

О.А. Петухов
А.В. Морозов
Е.О. Петухова

МОДЕЛИРОВАНИЕ

СИСТЕМНОЕ ИМИТАЦИОННОЕ АНАЛИТИЧЕСКОЕ

УЧЕБНОЕ ПОСОБИЕ

Допущено учебно-методическим объединением вузов по университетскому политехническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальности 230101 «Вычислительные машины, комплексы, системы и сети»

Санкт-Петербург

Издательство СЗТУ
2008

Утверждено редакционно-издательским советом университета

УДК 681.3

Петухов, О.А. Моделирование: системное, имитационное, аналитическое: учеб. пособие / О.А. Петухов, А.В. Морозов, Е.О. Петухова. – 2-е изд., испр. и доп. – СПб.: Изд-во СЗТУ, 2008. – 288 с.

Пособие написано в соответствии с требованиями государственных образовательных стандартов высшего профессионального образования по специальности 230101.65 – «Вычислительные машины, комплексы, системы и сети» и направлению подготовки бакалавра 230100.62 – «Информатика и вычислительная техника».

Излагаются принципы системного моделирования, построения имитационных и аналитических моделей. Приводится большое количество примеров. Может быть использовано студентами смежных специальностей и специалистами, интересующимися проблемами моделирования.

Под общей редакцией О.А. Петухова, канд. техн. наук, профессора.

Рецензенты: кафедра вычислительных машин, комплексов, систем и сетей СЗТУ (Г.И.Анкудинов, д-р техн. наук, проф.); М.М.Воронина, д-р физ.-мат. наук, проф. кафедры высшей математики ПГУПС, Ю.Н.Сидоров, канд. техн. наук, доц. кафедры математического обеспечения и применения ЭВМ СПбГЭТУ (ЛЭТИ).

Предисловие

Данное учебное пособие является исправленным и дополненным переизданием учебного пособия «Моделирование: системное, имитационное, аналитическое»,

ДИПЛОМ

ФОНД РАЗВИТИЯ ОТЕЧЕСТВЕННОГО ОБРАЗОВАНИЯ

ЛАУРЕАТОВ КОНКУРСА

на лучшую научную книгу 2006 года

НАГРАЖДАЕТСЯ КОЛЛЕКТИВ АВТОРОВ

Петухов Олег Александрович

Морозов А.В.

Петухова Е.О.

(г. Санкт-Петербург)

за книгу «Моделирование системное имитационное аналитическое»

Президент Фонда развития отечественного образования, ректор Университета Российской академии образования, академик Государственной академии наук, доктор педагогических наук, профессор

М.Н. Берулава

2007 год

Сочи

которое было отмечено Фондом развития отечественного образования и получило Диплом лауреатов конкурса на лучшую научную книгу 2006 года.

Компьютерное моделирование приме-няется в настоящее время во всех сферах деятельности человека. Отсюда целесообразность включения дисцип-лины «Моделирование» в блок специ-альных дисциплин при подготовке инженеров многих направлений, в том числе и специальности 230101.65.

Когда возникает потребность осуществить моделирование процесса функционирования системы? Эта потребность, как правило, появляется в тех случаях, когда человеку необходимо принять какое-то решение. Особенно, если это решение ответственное, может повлечь за собой большие финансовые или временные расходы или последствия могут оказать вредное воздействие на окружающих.

Можно долго перечислять сферы возможного применения теории моделирования: промышленность, транспорт, связь, торговля, здравоохранение, служба быта, наука и образование, деятельность органов внутренних дел и юстиции и, конечно, область обороны страны. Всё названное требует создания, а соответственно, возможно, и моделирования процессов функционирования соответствующих систем и систем управления, вычислительных систем и сетей, автоматизированных систем управления, баз данных и баз знаний.

Под моделированием понимается замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала с помощью объекта-модели. В общем виде моделирование определяется как представление исследуемого объекта моделью для получения информации путем проведения экспериментов. Теория замещения одних объектов (оригиналов) другими объектами (моделями) и исследование свойств объектов на их моделях называется теорией моделирования.

Учебное пособие посвящено изложению основ теории моделирования, приёмов построения имитационных и аналитических моделей, реализации моделей на современных программных средствах.

Введение

Моделирование разделяют на физическое и математическое. Преимущество физического моделирования перед натурным экспериментом очевидно, благодаря экономии времени и средств, но оно имеет более ограниченное применение по сравнению с математическим моделированием.

Под математическим моделированием понимают способ исследования различных процессов путем изучения явлений, имеющих различное физическое содержание, но описываемых одинаковыми математическими соотношениями. Важным является то обстоятельство, что при изучении любого процесса методом математического моделирования необходимо в первую очередь построить его математическое описание, то есть математическую модель. Под математической моделью реальной системы понимают совокупность соотношений (например, формул, уравнений, неравенств, логических условий, операторов и т. п.), определяющих характеристики состояний системы (а через них и выходные характеристики) в зависимости от ее параметров, входных характеристик, начальных условий и времени.

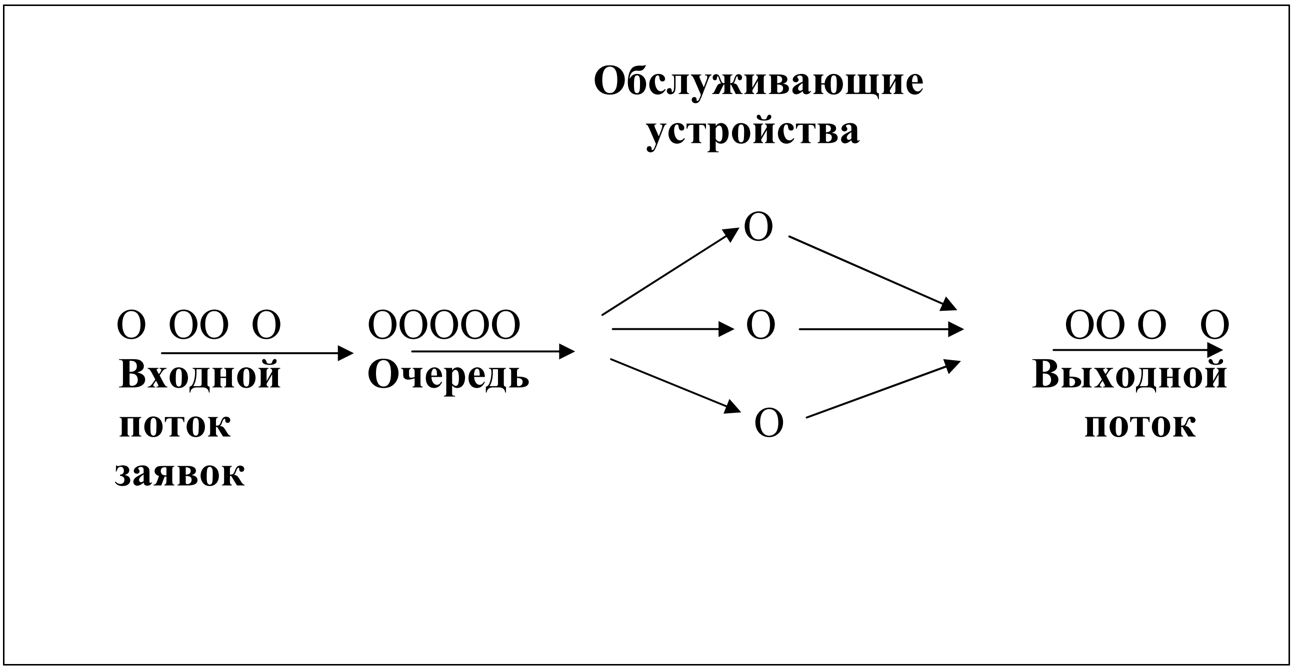
Математическая модель реальной системы является тем абстрактным формально описанным объектом, изучение которого возможно математическими методами, в том числе и с помощью математического моделирования.

Сказанное определило структуру данного учебного пособия, состоящего из трёх частей, посвященных основам системного, имитационного и аналитического моделирования, а также методикам реализации моделей с использованием различных программных средств.

Необходимость издания пособия обусловлена тем обстоятельством, что список рекомендуемой литературы по дисциплине «Моделирование» постоянно растет, с одной стороны, а с другой – рекомендуемая литература очень быстро становится библиографической редкостью. Все литературные источники, которые могли бы заинтересовать читателя, приведены в библиографическом списке.

Часть I

НАУКА И ИСКУССТВО МОДЕЛИРОВАНИЯ ПРОБЛЕМЫ РАЗРАБОТКИ МОДЕЛЕЙ



Глава 1. Основы терминологии моделирования

Глава 2. Технология моделирования

Глава 3. Классификация математических моделей

Глава 4. Модели систем массового обслуживания

Создание сложных систем, производственных комплексов, систем управления этими комплексами часто требует использования знаний о количественных и качественных закономерностях, свойственных рассматриваемым системам. Главными проблемами при этом являются общесистемные вопросы, включающие определение структуры, организацию взаимосвязи между элементами, взаимодействие с внешней средой, управление функционированием как всей системы, так и ее отдельных элементов.

Приведенный перечень проблем, являющийся далеко не полным, относится к задачам системного анализа и ***системного моделирования.*** Основные понятия этого подхода приводятся в первой части данного учебного пособия [1, 2, 7–9, 22, 31, 41].

Глава 1
ОСНОВЫ ТЕРМИНОЛОГИИ МОДЕЛИРОВАНИЯ

Центральными понятиями системного моделирования являются: ***система, системный подход, модель, теория моделирования***.

Система

Под **системой** обычно понимается совокупность объектов, элементов или составляющих произвольного характера, образующих в некотором смысле определенную целостность. Кроме того, часто используется термин *сложная (большая) система.* В этом случае предполагается, что в системе имеется большое количество связанных и взаимодействующих между собой элементов, обеспечивающих выполнение системой определенной достаточно сложной задачи.

Примеры сложных систем, встречающиеся в окружающем мире, можно перечислять достаточно долго. Это и компьютер, и автомашина, и телефонный аппарат, и автоматизированная система, и вычислительная система, и деканат, и производственный комплекс, и так далее.

Понятие элемента системы и расчленение ее на отдельные компоненты является относительным. Формально элементом можно считать объект, не подлежащий дальнейшему расчленению на части. Например, рассматривая в качестве сложной системы вычислительную сеть, можно считать ее элементами отдельные компьютеры. Если же сложной системой служит компьютер, то элементами можно считать процессор, память, монитор и так далее.

Системный подход

Системный подход заключается в том, что исследователь пытается изучать поведение системы в целом, а не концентрировать свое внимание на отдельных ее частях. Такой подход основывается на признании того, что если даже каждый элемент или подсистема имеет оптимальные конструктивные или функциональные характеристики, то результирующее поведение системы в целом может оказаться лишь субоптимальным вследствие взаимодействия между ее отдельными частями.

Возрастающая сложность организационных систем и потребность преодолеть эту сложность привели к тому, что системный подход становится все более и более необходимым методом исследования [31].

Так, например, в книге Р. Шеннон «Имитационное моделирование – искусство и наука» (М., 1975) отмечается: «Архитектор может рассматривать дом вместе с его электрической, отопительной и водопроводной системами как одну большую систему. Но инженер-теплотехник вправе рассматривать отопительную систему как законченную систему, а дом как окружающую среду. Для социопсихолога дом может рассматриваться как среда, окружающая семью, а последняя – как система, исследованием которой он занимается. Для него связь между отопительной и электрической системами может не иметь никакого значения, но для архитектора эта взаимосвязь может быть очень важной». Таким образом, необходимо проявлять значительную осторожность при определении изучаемой системы и границ между ней и окружающей её средой.

Определенная совокупность элементов рассматриваемой системы может представляться как ее подсистема. Считается, что к подсистемам относят некоторые самостоятельно функционирующие части системы. Поэтому для упрощения процедуры исследования первоначально необходимо грамотно выделить подсистемы сложной системы, то есть - определить ее структуру.

Структура системы

Структура системы - это устойчивая во времени совокупность взаимосвязей между ее компонентами (подсистемами). И при системном подходе важным этапом является определение структуры изучаемой, описываемой системы. Это связано с тем, что грамотное выделение подсистем способствует упрощению процедур исследования и анализа результатов.

В исследовании структуры системы существует ряд направлений, основными из которых являются структурный и функциональный подходы. Названия подходов уже говорят о смысле и содержании используемых методик. Структурный подход подразумевает описание состава элементов системы и связи между ними. Функциональный подход заключается в исследовании отдельных алгоритмов ее поведения.

В сложных системах одной из главных является задача управления, которая представляет собой процесс сбора, передачи и переработки информации, осуществляемый специальными средствами. От элементов системы к управляющим устройствам поступает осведомительная информация, характеризующая состояния элементов системы. Кроме того, средства управления могут получать информацию извне в виде управляющих команд от вышестоящих органов управления или воздействий внешней среды. Управляющие устройства перерабатывают всю поступающую к ним информацию. В результате этой переработки синтезируются управляющие команды, которые изменяют состояния и режимы функционирования элементов системы.

Реальные сложные системы обычно функционируют в условиях воздействия на них большого количества случайных факторов, в качестве источников которых могут быть внешние (соседние) системы, стоящие на других уровнях иерархии. Кроме того, случайными факторами воздействия могут быть ошибки, шумы и отклонения различных величин внутри системы. Внешние и внутренние случайные воздействия, естественно, влияют на режимы работы элементов системы и могут существенно менять характер ее функционирования.

В монографии Н.П. Бусленко [7] показано, что для сложных систем, встречающихся на практике, как правило, ***действие случайных факторов приводит к смещению средних значений результатов их функционирования***. Это обстоятельство требует особого внимания к учету случайных факторов при исследовании сложных систем.

На основании сказанного можно перечислить основные отличительные признаки сложных систем:

1. Наличие большого количества взаимно связанных и взаимодействующих между собой элементов.

2. Сложность функции, выполняемой системой и направленной на достижение заданной цели функционирования.

3. Возможность разбиения системы на подсистемы, цели функционирования которых подчинены общей цели функционирования всей системы.

4. Наличие управления (часто имеющего иерархическую структуру), разветвленной информационной сети и интенсивных потоков информации.

5. Наличие взаимодействия с внешней средой и функционирование в условиях воздействия случайных факторов.

Показатель эффективности

В соответствии с пунктами 1 – 3 перечисленных признаков считают, что сложная система, рассматриваемая как совокупность объектов (элементов, подсистем), предназначена для выполнения определенных работ или решения некоторого класса задач. Поэтому процесс функционирования сложной системы является совокупностью действий ее элементов, подчиненных единой цели.

Если цели и задачи системы определены, можно ставить вопрос об определении качества ее функционирования, которое оценивается при помощи ***показателя эффективности***. Эта числовая характеристика, с одной стороны, оценивает степень приспособленности системы к выполнению поставленных перед нею задач, а с другой – существенно влияет на результаты ее исследования.

Для пояснения сказанного можно рассмотреть некоторый производственный процесс. При описании целей и задач необходимо указать перечень изделий, для выпуска которых предназначена данная система. Но, приведя только этот перечень, невозможно получить сведения для определения оценки качества функционирования. Если выбрать в качестве показателя эффективности рассматриваемого производственного процесса ***производительность***, измеряемую количеством изделий, выпускаемых в течение фиксированного интервала времени (смена, неделя, месяц), то может оказаться, что формально предпочтение будет отдаваться факторам, способствующим достижению максимальной производительности. Это может привести к ухудшению других характеристик производственного процесса (экономия сырья, износ оборудования, расход энергии). Если же в качестве показателя эффективности использовать ***себестоимость*** продукции, то, наоборот, экономия сырья, износ оборудования, расход энергии и фонд заработной платы будут иметь важное значение, в то же время окажется, что производительность может вообще не учитываться.

В итоге для производственного процесса необходимо выбирать такие показатели эффективности, которые учитывают как себестоимость продукции, так и производительность оборудования, например величину прибыли, рентабельность.

Для того чтобы показатель эффективности полно характеризовал качество работы системы, должны учитываться все основные особенности и свойства системы, а также условия ее функционирования и взаимодействия с внешней средой. Таким образом, показатель эффективности должен зависеть от структуры системы, значений ее параметров, характера воздействия внешней среды, внешних и внутренних случайных факторов.

На основании сказанного показатель эффективности определяется ***процессом функционирования системы***. Он отражает поведение системы во времени и представляется как последовательное изменение ее состояний. Если система изменяет свое состояние, то говорят, что система ***переходит*** из одного состояния в другое.

Поэтому можно представить ***множество*** возможных процессов функционирования системы, элементы которого отличаются друг от друга за счет различных условий и режимов работы системы. Каждому элементу этого множества ставится в соответствие элемент другого множества, а именно множества значений показателя эффективности системы. Так как значения показателя представляют собой действительные числа, то можно говорить об отображении множества процессов функционирования системы на множество действительных чисел, заключенных внутри некоторого интервала (в пределах изменения значений показателя эффективности). На основании сказанного показатель эффективности можно считать ***функционалом*** от процесса функционирования системы.

В связи с тем, что сложные системы работают в условиях действия случайных факторов, значения функционалов оказываются ***случайными*** величинами. Это создает некоторые сложности и поэтому при выборе показателя эффективности пользуются средними значениями соответствующих функционалов. Примерами таких средних значений функционалов служат средние количества изделий, выпускаемых за смену, средняя себестоимость продукции, средняя прибыль (для производственных процессов), средняя длительность поездки, средняя стоимость перевозки (для транспорта), среднее время ожидания в очереди (для систем массового обслуживания).

Иногда в качестве показателя эффективности используются ***вероятности*** некоторых случайных событий, например вероятность успешной посадки самолета (для системы слепой посадки), вероятность застать абонентскую линию занятой (для систем телефонной связи), вероятность попасть в очередной автобус (для пассажира, находящегося в очереди) и, в общем случае, вероятность того, что заявка, пришедшая в систему, будет обслужена.

Модель

Модель (фр. modèle – образец) является представлением объекта, системы или понятия (идеи) в некоторой форме, отличной от формы их реального существования. Модель обычно служит средством, помогающим в объяснении, понимании или совершенствовании системы. Можно привести несколько определений этого понятия.

Модель – это используемый для предсказания и сравнения инструмент, позволяющий логическим путем спрогнозировать последствия альтернативных действий и достаточно уверенно указать, какому из них отдать предпочтение [44].

Модель – это некоторое представление о системе, отражающее наиболее существенные закономерности ее структуры и процесса функционирования и зафиксированное на некотором языке или в некоторой форме [13].

Модель – это объект-заместитель объекта-оригинала, обеспечивающий изучение некоторых свойств оригинала [41].

Таким образом, модель есть материально или теоретически созданная система, предназначенная заменить или представлять объект исследования в процессе познания. Модель должна быть более удобной для исследования. Изучение модели и реализация с её помощью различных задач позволяет получить информацию о реальном объекте исследования.

Моделирование

О моделировании естественно говорить лишь при использовании модели для познания оригинала.

Замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала с помощью объекта-модели называется ***моделированием***. Это один из наиболее распространенных способов изучения различных процессов и явлений.

Существует много подходов к классификации методов и приемов моделирования, но основным является подразделение на ***физическое*** и ***математическое*** моделирование.

При физическом моделировании модель воспроизводит изучаемый процесс (оригинал) с сохранением его физической природы и поэтому, конечно, имеет ограниченное применение.

Математическое моделирование - это способ исследования различных процессов путем изучения явлений, имеющих различное физическое содержание, но описываемых одинаковыми математическими соотношениями.

Под ***математической моделью*** реальной системы понимают ***совокупность соотношений*** (например, формул, уравнений, неравенств, логических условий, операторов), ***определяющих характеристики состояний системы (а через них и выходные сигналы) в зависимости от параметров системы, входных сигналов, начальных условий и времени*** [7]. То есть математическая модель концентрирует в себе написанную на определённом языке (естественном, математическом, алгоритмическом) совокупность знаний, представлений и гипотез о соответствующем объекте или явлении. Обычно модель только приближённо описывает поведение реальной системы, являясь её абстракцией, так как знания о реальной системе никогда не бывают абсолютными, а гипотезы часто вынужденно или намеренно не учитывают некоторые факторы.

Зачем разрабатываются модели и реализуется моделирование? В большинстве случаев невозможно повторить эксперимент на оригинале в одних и тех же условиях (например, изменяются погодные условия – при исследовании причин возникновения наводнений в Невской губе Финского залива), или эти эксперименты чрезвычайно опасны для исследователей (например, эксперименты на взорвавшемся блоке Чернобыльской АЭС), или просто эксперименты с системой являются дорогостоящим и трудоёмким делом.

Моделирование целесообразно, когда у модели отсутствуют те признаки оригинала, которые препятствуют его исследованию, или имеются отличные от оригинала признаки, способствующие фиксации или изучению свойств модели. Из сказанного можно сделать вывод, что польза от моделирования может быть получена только при соблюдении двух условий: модель обеспечивает адекватное отображение существенных свойств оригинала и модель позволяет устранить существующие проблемы, присущие проведению экспериментов на реальном объекте.

Теория моделирования

Теория моделирования представляет собой взаимосвязанную совокупность положений, определений, методов и средств создания и изучения моделей. Эти положения, определения, методы и средства, как и сами модели, являются предметом теории моделирования.

Основная задача теории моделирования заключается в том, чтобы вооружить исследователя технологией создания таких моделей, которые достаточно точно и полно фиксируют интересующие свойства оригинала, проще и быстрее поддаются исследованию и допускают перенесение его результатов на оригинал.

Формально модель объекта моделирования S представляют в виде множества величин [41], описывающих процесс функционирования реальной системы и образующих следующие подмножества:

– совокупность ***входных воздействий*** на систему

$$x_i \in X, \quad i = 1, 2, ..., n_X;$$

– совокупность ***воздействий внешней среды***

$$v_l \in V, \quad l = 1, 2, ..., n_V;$$

– совокупность ***внутренних параметров*** системы

$$h_k \in H, \quad k = 1, 2, ..., n_H;$$

– совокупность ***выходных характеристик*** системы

$$y_j \in Y, \quad j = 1, 2, ..., n_Y.$$

В перечисленных подмножествах можно выделить управляемые и неуправляемые переменные. В общем случае x_i, v_l, h_k, y_j являются элементами непересекающихся подмножеств и содержат как детерминированные, так и стохастические составляющие.

При моделировании системы S входные воздействия, воздействия внешней среды и внутренние параметры системы являются ***независимыми переменными***, которые в векторной форме записываются так:

$$\mathbf{x}(t) = \left(x_1(t), x_2(t), ..., x_{n_X}(t)\right);$$

$$\mathbf{v}(t) = \left(v_1(t), v_2(t), ..., v_{n_V}(t)\right);$$

$$\mathbf{h}(t) = \left(h_1(t), h_2(t), ..., h_{n_H}(t)\right),$$

а выходные характеристики системы являются ***зависимыми переменными*** и в векторной форме имеют вид

$$\mathbf{y}(t) = \left(y_1(t), y_2(t), ..., y_{n_Y}(t)\right).$$

Процесс функционирования системы S описывается во времени оператором $\mathbf{F}_S$, который в общем случае преобразует независимые переменные в зависимые в соответствии с соотношениями вида

$$\mathbf{y}(t) = \mathbf{F}_S\left(\mathbf{y}(t), \mathbf{v}(t), \mathbf{h}(t), t\right). \tag{1.1}$$

Совокупность зависимостей выходных характеристик системы от времени $y_j(t)$ для всех $j = 1, 2, ..., n_Y$ называется ***выходной траекторией*** $\mathbf{y}(t)$. Зависимость (1.1) называется ***законом функционирования системы*** S и обозначается $\mathbf{F}_S$. В общем случае закон функционирования системы $\mathbf{F}_S$ может быть задан в виде функции, функционала, логических условий, в алгоритмической и табличной формах или в виде словесного правила соответствия.

Для описания и исследования системы S важным является понятие ***алгоритма функционирования*** A_S, под которым понимается метод получения выходных характеристик с учетом входных воздействий $\mathbf{x}(t)$, воздействий внешней среды $\mathbf{v}(t)$ и собственных параметров системы $\mathbf{h}(t)$. Один и тот же закон функционирования $\mathbf{F}_S$ системы S может быть реализован различными способами, т.е. с помощью множества различных алгоритмов функционирования A_S.

В заключение данной главы можно перечислить пять главных принципов, на которых базируется теория моделирования и которые положены в основу технологии моделирования [12, 13]:

1. ***Принцип информационной достаточности***. Моделирование системы бессмысленно, если имеется исчерпывающая информация о её функционировании. Противоположностью данному тезису является следующее: если нет никакой информации о системе, то осуществить моделирование такой системы невозможно. Промежуточным вариантом является требование наличия определённого порогового уровня априорных знаний о системе (именно это является принципом информационной достаточности), когда существуют условия построения модели, адекватной исследуемой системе.

2. ***Принцип осуществимости***. Осуществимость заключается в том, что разрабатываемая модель должна достигнуть реализации цели исследования с отличной от нуля вероятностью за определённое (конечное) время. То есть должно выполняться соотношение

$$P(t_0) \geq p_0,$$

где $P(t_0)$ – вероятность достижения цели моделирования за приемлемое время t_0; p_0 – заданная (желаемая) вероятность достижения цели моделирования.

3. ***Принцип множественности моделей***. В разрабатываемой модели необходимо реализовывать только те свойства системы или явления, которые оказывают существенное влияние на показатель эффективности. Из этого следует, что использование полученной модели отражает только определённые (учтённые) стороны (характеристики) реального процесса. Поэтому для исчерпывающего исследования моделируемого процесса, возможно, потребуется построение набора моделей, которые бы позволили с разных сторон и с разной степенью детализации анализировать характеристики реального процесса.

4. ***Принцип агрегирования***. Любая сложная система может быть

представлена набором некоторых подсистем (агрегатов), а для их математического описания можно использовать определённые математические схемы. Этот принцип даёт возможность довольно легко перестраивать модель в зависимости от возникающих проблем и задач исследования.

5. ***Принцип параметризации***. Часто встречаются системы, в структуру которых включены достаточно изолированные компоненты (подсистемы). Если эти подсистемы характеризуются некоторым параметром, то представляется возможным заменить их в модели соответствующими числовыми значениями (или графиками, таблицами или формулами) и не описывать их функционирование. Использование этого принципа сокращает объём и время моделирования, но следует учитывать, что при этом, естественно, снижается адекватность модели.

Глава 2
ТЕХНОЛОГИЯ МОДЕЛИРОВАНИЯ

Расширению сфер применения моделирования способствовало, конечно, развитие и применение вычислительных машин и систем. С уверенностью можно сказать, что не существует областей человеческой деятельности, где бы ни использовалось моделирование. Это и функционирование атомных реакторов, и исследование макроэкономических процессов, и производство машин и агрегатов, и развитие биологических систем, и анализ последствий использования природных ресурсов и экологических катастроф [1].

Сами вычислительные комплексы стали не только инструментом исследования, но и объектами моделирования. Вычислительные машины, комплексы, системы и сети благодаря своей сложности и дороговизне, естественно, являются объектами моделирования. При этом моделирование целесообразно как на этапах проектирования вычислительных комплексов, так и для анализа функционирования действующих систем в экстремальных условиях или при изменении их состава, структуры, способов управления или рабочей нагрузки.

Применение моделирования на этапе проектирования позволяет анализировать варианты проектных решений, определять работоспособность и производительность, выявлять дефицитные и малозагруженные ресурсы, вычислять ожидаемые времена реакции и принимать решения по рациональному изменению состава и структуры комплекса или по способу организации вычислительного процесса.

Целесообразно использование моделирования для действующих вычислительных комплексов, поскольку можно опытным путём проверить адекватность модели и оригинала и более точно определить те параметры системы и внешние воздействия на неё, которые служат исходными данными для моделирования. Моделирование реальных вычислительных комплексов позволяет выявить его резервы и прогнозировать качество функционирования при любых изменениях, поэтому полезно иметь модели всех развивающихся систем.

Осуществление моделирования предусматривает конкретизацию цели моделирования, создание модели, проведение её исследования и анализ полученных результатов. Даже процесс создания модели состоит из нескольких этапов. Он начинается с изучения системы и внешних воздействий, а завершается разработкой или выбором математической модели и программного обеспечения. Естественно, что некоторые классы математических моделей могут быть исследованы и без применения компьютерной техники.

Таким образом, моделирование предполагает наличие следующих укрупнённых этапов:

– постановка цели моделирования,
– разработка концептуальной модели,
– подготовка исходных данных,
– разработка математической модели,
– выбор метода моделирования,
– выбор средств моделирования,
– разработка программного обеспечения,
– проверка адекватности и корректировка модели,
– планирование машинных экспериментов,
– моделирование на вычислительном комплексе,
– анализ результатов моделирования.

Приведённый перечень, конечно, не догма. Различные исследователи в соответствии со своим опытом и пристрастиями изменяют содержание и количество этапов, но логика последовательности должна сохраняться.

2.1. Постановка цели моделирования

Любая система может представляться некоторым набором, отличающихся друг от друга, моделей. Отличия могут содержаться в степени детализации и учёте различных особенностей и режимов функционирования. Могут отражаться некоторые грани сущности системы, можно ориентироваться на анализ некоторых наборов свойств. Поэтому разработке модели, естественно, предшествует постановка (формулировка) цели моделирования.

Как подойти к решению этой задачи? Так как создание модели обычно осуществляется при проектировании или модернизации системы, то естественно возникают задачи определения её (системы) эффективности, решаемые до моделирования. Этот этап заканчивается принятием решения о целесообразности или, наоборот, нецелесообразности проведения моделирования.

Необходимо помнить, что подобие процесса, протекающего в модели, реальному процессу не может быть целью. Это только условие правильного функционирования модели, да и то не всегда.

Определение цели моделирования – это расчёт значений выбранного показателя эффективности для различных вариантов реализации проектируемой или исследуемой системы. Например, при определении варианта построения компьютерной сети, которая бы обладала минимальной стоимостью при соблюдении требований по производительности и по надёжности, целью моделирования может быть отыскание параметров сети, обеспечивающих минимальное значение показателя эффективности. Одним из этих параметров может быть стоимость.

Задача может формулироваться иначе. Из нескольких вариантов конфигурации компьютерной сети необходимо выбрать наиболее надёжный. В данном случае показателем эффективности можно выбрать один из показателей надёжности (средняя наработка на отказ, вероятность безотказной работы и пр.), а целью моделирования является сравнительная оценка вариантов сети по этому показателю.

Одновременно с этим рассмотренная компьютерная сеть вуза может использоваться в системах обучения и управления набором студентов, в системе научно-технических разработок и пр. Эта же сеть может быть элементом систем энергоснабжения, технического обслуживания и снабжения. В этом случае показатель эффективности компьютерной сети невозможно оценить с помощью единиц СИ.

Тогда используют показатель технико-экономической эффективности E, который учитывает не только затраты, но и некоторые измеряемые выходные характеристики системы

$$E = E(W), \tag{1.2}$$

где элементы множества характеристик $w_s \in W$ ($s = 1, 2, \ldots, n_W$) – частные показатели качества системы (производительность, надёжность, стоимость, масса, габариты и пр.).

Если известны аналитические соотношения (1.1) и (1.2), то показатель эффективности E вычисляется элементарно по совокупности внутренних параметров системы H при определённых внешних воздействиях V. Или, наоборот, по заданному показателю эффективности вычисляются требуемые параметры системы.

При отсутствии аналитических зависимостей (1.1) и (1.2) используют подходы, называемые однокритериальной или многокритериальной оценками [31].

Однокритериальная оценка. Производится оценка эффективности по одному частному показателю качества w°, а на допустимые значения прочих элементов множества W накладываются ограничения:

$$E = E(w^{\circ}), \tag{1.3}$$

$$w_{s_{\min}} \le w_s \le w_{s_{\max}}, \tag{1.4}$$

где $w_{s_{\min}}$ и $w_{s_{\max}}$ – граничные значения s – го частного показателя качества.

Особенностью такого подхода к оценке эффективности является возможность получения нескольких вариантов системы для определённого значения w° при различных значениях прочих частных показателей качества, удовлетворяющих неравенству (1.4).

Многокритериальная оценка. В этом случае наиболее распространённым подходом является представление неизвестной функции (1.2) при помощи нормированного аддитивного критерия:

$$E = \sum_{s=1}^{s_w} g_s \gamma(w_s), \qquad (1.5)$$

где $\gamma_s(w_s)$ – значения оценки частного показателя качества, обеспечивающие исключение размерности и подобранные так, чтобы $\gamma_s(w_s) \in [0,\ 1]$; g_s – весовые коэффициенты, которые учитывают важность $s-$го показателя и удовлетворяют условию

$$\sum_{s=1}^{s_w} g_s = 1, \quad g_s > 0.$$

Уточнение цели моделирования. Таким образом, на основании анализа взаимодействия системы с окружающей средой и предварительного изучения внутренних структурных и функциональных особенностей определяется цель моделирования путём определения вида критерия эффективности системы. Затем исключаются те характеристики, которые можно определить без моделирования. Итоговое множество учитываемых внешних воздействий должно включать только способствующие (полезные воздействия) или препятствующие (возмущающие воздействия) функционированию системы. Но, естественно, при моделировании не удаётся, да и нецелесообразно включать всё многообразие возмущающих факторов. И это часто приводит к идеализации условий и, возможно, к искажению результатов моделирования.

Если моделирование преследует цель не только фиксации свойств, но и оптимизации характеристик, то необходимо выявить те параметры, которые можно изменять в процессе моделирования. Когда ставится задача определения зависимости характеристик от некоторых параметров системы (1.1), то эти характеристики и параметры должны быть точно зафиксированы.

Оценка целесообразности осуществления моделирования является завершающим моментом этапа определения цели моделирования. Проведение моделирования требует некоторых затрат: чем система сложнее и чем выше требования к точности результатов, тем затраты больше. Если целью является обоснование работоспособности системы, то экономический эффект от эксплуатации данной системы должен превосходить расходы на моделирование. При оптимизации системы затраты на моделирование должны окупаться за счёт разницы между наилучшим и наихудшим вариантами системы.

Если результат моделирования показывает, что исследуемый вариант системы является приемлемым (оптимальным), то нецелесообразно принимать решение о напрасных затратах на моделирование. Важно, что моделирование подтвердило правильность принятых проектных решений.

2.2. Разработка концептуальной модели

Концептуальная (содержательная) модель – это абстрактная модель, определяющая состав и структуру системы, свойства её элементов и причинно-следственные связи, присущие исследуемой системе и существенные для достижения цели моделирования.

Как видно, это всего-навсего словесное описание природы, параметров и условий взаимодействия отдельных компонентов системы. Это первоначальное представление модели, возможно, в воображении исследователя. В то же время концептуальная модель – это в некотором смысле субстрат системы, обеспечивающий достижение цели моделирования.

Необходимость обоснования включения в модель определённых важных элементов и свойств, а также исключения из модели несущественных свойств, требует глубоких знаний о самой системе. Это противоречие (включение одних элементов и исключение других) приводит к определённым трудностям, так как выявление влияния исключения того или иного фактора на степень искажения результатов требует создания уже двух моделей: с учётом и без учёта этого фактора. А так как часто количество элементов бывает чрезвычайно велико, то и количество моделей может увеличиваться лавинообразно, приводя к значительному возрастанию затрат.

Простота модели и её адекватность исследуемой системе – вот тот редко и с трудом находимый компромисс, который должен быть достигнут при моделировании.

Ответственность ложится на разработчика. Именно он на основании своих знаний и опыта принимает решение об исключении из модели некоторого элемента или набора элементов без полной уверенности в том, что эти действия не приведут к получению значительной погрешности при моделировании. Поэтому часто говорят, что моделирование является не только наукой, но и искусством [17, 44].

Разработка статической составляющей концептуальной модели

Создание разработчиком концептуальной модели можно формализовать, т.е. разбить на два следующих действия: разработка статической и динамической моделей. На этапе создания статической модели осуществляются стратификация, детализация, локализация и структуризация.

Стратификация. Обязательными свойствами системы (и вычислительной в том числе) являются, с одной стороны, её целостность, а с другой – способность к разбиению системы на совокупность элементов. В свою очередь и модель – это совокупность частей, обеспечивающих сохранение целостности.

Разделение системы на совокупность элементов (членимость системы и её элементов) может привести к построению иерархической последовательности моделей. В итоге система представляется набором моделей, отображающих её поведение на различных уровнях декомпозиции (эти уровни называются стратами). Каждый уровень учитывает присущие ему свойства, переменные и зависимости, а процесс выделения уровней называется стратификацией.

В модель, как правило, включают элементы одного уровня детализации – K-го страта. Иногда, в случае появления затруднений в описании элементов, включают в модель их детализированное представление из нижнего $(K-1)$-го страта.

Построение стратифицированной концептуальной модели предусматривает включение в неё параметров, с помощью которых возможно варьирование свойств при моделировании, которые обеспечивают определение необходимых характеристик при конкретных внешних воздействиях на заданном временном интервале функционирования системы. Остальные параметры без сожаления исключаются из модели.

Детализация. Функционирование любой системы – есть реализация определённого количества технологических процессов преобразования вещества, энергии или информации. Эти процессы складываются из последовательности элементарных операций, а выполнение каждой элементарной операции обеспечивается определённым ресурсом. Отсюда следует, что в модели должны присутствовать составляющие элементы, реализующие выполнение всех технологических процессов. Помимо того в модель могут включаться элементы, служащие управлению ресурсами и процессами, а также элементы, предназначенные для хранения информации, необходимой для управления.

Глубина детализации осуществляется до такого уровня, чтобы имелись зависимости параметров выходных воздействий для каждого элемента. Эти параметры должны быть существенны для функционирования системы и определения её выходных характеристик от параметров входных воздействий.

Повышение уровня детализации описания системы приводит к получению более точной её модели. Но это усложняет процесс моделирования и ведёт к росту временных затрат на его осуществление.

Из казанного вытекает правило: *в модель должны быть включены все параметры, обеспечивающие определение интересующих исследователя характеристик системы на заданном временном интервале её функционирования; остальные параметры, по возможности, исключаются из модели.*

Локализация. Это представление внешней среды в виде генераторов внешних воздействий, включаемых в состав модели в качестве элементов (рис. 1.1). Их можно подразделить на:

– генераторы рабочей нагрузки, поставляющие на вход системы вещество, энергию или данные;

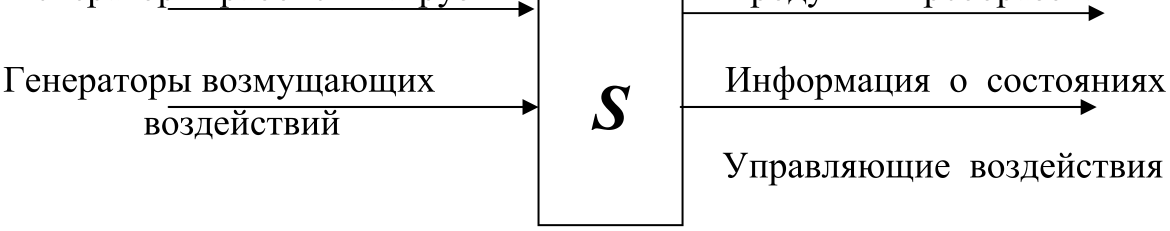
– генераторы управляющих и возмущающих воздействий.

Результаты функционирования системы подразделяют на:

– продукты преобразования,

– информацию о состоянии системы,

– управляющие воздействия на другие системы.



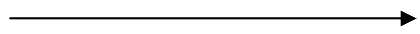


Рис. 1.1. Представление внешней среды системы в виде генераторов внешних воздействий

Структуризация. Это перечисление связей между элементами модели. Связи бывают вещественные и информационные. Вещественные связи показывают пути возможного перемещения продуктов преобразования от элемента к элементу. С помощью информационных связей осуществляется передача между элементами управляющих воздействий и информации о состояниях. Эти связи могут и не представляться в системе каким-то материальным каналом.

В случае, если система состоит из однофункциональных элементов с одной выходной вещественной связью, информационные связи, естественно, отсутствуют, а управление функционированием производится самой структурой. Это принцип структурного управления. В качестве примера подобных систем обычно приводят логические элементы вычислительных комплексов.

Если система включает элементы с несколькими выходными вещественными связями и имеются управляющие средства и информационные связи, то она функционируют в соответствии с программным или алгоритмическим управлением. В данном случае управление требуется для определения самого элемента, какую операцию преобразования выполнить и куда передать.

Концептуальная модель конкретно содержит перечисление и содержание всех правил и алгоритмов управления рабочей нагрузкой, элементами и процессами.

Разработка динамической составляющей концептуальной модели

Всё вышеперечисленное направлено на разработку статической модели, то есть отражается её состав и структура. А для описания динамики функционирования системы необходимо учитывать алгоритмы управления и параметры входных воздействий и элементов.

Как уже упоминалось, в процессе функционирования системы реализуется технологический процесс преобразования вещества, энергии и информации. В вычислительных машинах реализуется мультипрограммный режим работы. А технологический процесс в них реализует некоторую последовательность элементарных операций, часть из которых выполняется параллельно разными элементами (ресурсами).

В общем случае существует два подхода к реализации процедур управления технологическим процессом. Это программный и структурный подходы.

Технологический процесс, задаваемый программой, реализуется некоторым аналогом алгоритма управления. Алгоритм определяет, какие ресурсы системы, в какой последовательности и какие операции должны выполняться для достижения цели функционирования.

Существует интуитивное представление об алгоритме как о формальной совокупности операций и порядке их выполнения для решения задач какого-либо типа [3].

Под алгоритмом обычно понимается точное предписание, определяющее процесс переработки исходных данных в требуемый результат. Любой алгоритм характеризуется массовостью, детерминированностью и результативностью.

В случае использования структурного принципа управления для каждого элемента системы выбираются параметры, которые изменяются во времени и, соответственно, отражают ход технологического процесса. Множество этих параметров отражают состояние системы, а функционирование системы представляется в виде последовательной смены состояний. Множество возможных состояний системы называют пространством состояний (1.1).

На завершающем этапе создания концептуальной модели необходима проверка её адекватности исходной системе. Но так как самой модели в физическом смысле пока не существует, такую проверку должен осуществить эксперт, но, ни в коем случае, не разработчик модели.

2.3. Подготовка исходных данных

Нужно помнить, что разработчик в процессе создания модели взаимодействует со многими специалистами, знакомыми с самой системой. А так как требования к модели могут изменяться в процессе изучения системы, то разработчику модели может понадобиться постоянная консультация со специалистами по исследуемым вопросам. Главное, что при создании модели разработчик должен использовать всю доступную информацию, которая может быть получена разными методами [18]. Эти методы таковы:

– консультации со многими специалистами по исследуемому вопросу;

– наблюдение за системой (если система, подобная исследуемой, уже существует, то собранные данные будут полезны при моделировании); однако при использовании таких данных могут возникнуть некоторые неудобства: а) данные не представляют то, что в действительности необходимо моделировать, б) данные не имеют подходящего типа или формата, в) в данных могут быть ошибки измерения, записи или округления, г) данные могут быть необъективными для получения определённой выгоды, д) данные могут быть представлены в несовместимых единицах;

– результаты, полученные в ходе моделирования подобных систем;

– опыт и интуиция разработчика – особенно, если система, подобная моделируемой, на данный момент не существует.

Таким образом, сбор, подготовка, первоначальная обработка исходных данных (качественных и количественных) производится на протяжении нескольких исходных этапов процесса разработки модели. Данные могут постоянно корректироваться и уточняться.

Количественные параметры системы должны иметь конкретные значения, так как они решающим образом влияют на успех моделирования. Точность и полнота исходных данных во многом определяют достоверность результатов моделирования.

Параметры модели (исходные данные) в общем случае могут быть детерминированными или стохастическими. С другой стороны, они часто оказываются нестационарными. И так как в большинстве случаев эти данные должны характеризовать (описывать) проектируемую (не существующую) или модернизируемую систему, то процедура сбора исходных данных превращается в очень сложную и очень важную проблему.

Большинство параметров, по своей природе являясь случайными величинами, при моделировании часто представляются детерминированными средними значениями. Это допустимо в тех случаях, когда случайные величины имеют малый разброс, или когда допустимо учитывать в модели только средние значения. Но это всегда ведёт к появлению определённой погрешности моделирования, так как воздействие случайных факторов может привести не только к рассеиванию, но и к смещению средних значений результатов [7, 16].

Иногда возможен и обратный подход, когда детерминированные характеристики заменяются случайными величинами. Обычно это направлено на сокращение объёмов исходных данных. Пусть, например, моделируется обработка данных вычислительной системой. В этом случае для многократных процессов реализации программ с разными объёмами исходных данных можно задавать всю совокупность количеств этих данных. А можно их заменить случайной величиной с заданным законом распределения (методика реализации случайных величин показана во второй части учебного пособия).

Подбор закона распределения. Для систематизации значений случайных параметров необходим сбор статистических данных и их обработка для определения возможности представления параметров каким-либо теоретическим законом распределения.

Процедура подбора вида закона распределения заключается в следующем [1, 5, 10, 16, 44]. По совокупности численных значений параметра строится гистограмма относительных частот – эмпирическая плотность распределения. Гистограмма аппроксимируется плавной кривой. Полученная кривая последовательно сравнивается с кривыми плотности распределения различных теоретических законов распределения. Выбирается один из законов по наилучшему совпадению вида сравниваемых кривых. По эмпирическим значениям вычисляют параметры этого распределения. Затем выполняют количественную оценку степени совпадения эмпирического и теоретического распределения по тому или другому критерию согласия (например, Пирсона (хи-квадрат), Колмогорова, Смирнова и пр.). Перечисленные этапы достаточно детально разработаны и описаны в математической статистике.

С другой стороны, исходные данные могут быть представлены уже в обработанном виде. При этом тоже возникают проблемы: распределение построено по малой выборке, информация представлена в качественной форме, имеются только агрегированные оценки, данные устарели или получены в другом месте или при других условиях, в выборке отсутствует часть данных.

Аппроксимация функций и выдвижение гипотез. Каждому элементу системы в любой момент времени можно поставить в соответствие функциональную связь между параметрами входных воздействий и его выходными характеристиками. Эта функциональная зависимость часто очевидна, или иногда легко определяется в результате анализа природы функционирования системы.

Но для некоторых элементов удаётся только получить набор экспериментальных данных о количественных значениях выходных характеристик при различных значениях параметров. В этом случае вводится некоторая гипотеза о характере функциональной зависимости. То есть осуществляется аппроксимация этой зависимости определённым математическим уравнением. Для этого используются методы регрессионного, корреляционного или дисперсионного анализов.

Вид уравнения может задаваться разработчиком. Наиболее просто это осуществляется для двух переменных по результатам сравнения графика с экспериментальными точками и графиков распространённых аппроксимирующих функций (прямой, параболы, гиперболы, экспоненты и пр.). Затем, для обеспечения наилучшего приближения кривой к экспериментальным данным, используя регрессионный анализ, вычисляются значения констант выбранного уравнения. Обычно приближения оцениваются по критерию наименьших квадратов.

Если параметры отражают новые элементы создаваемой системы или новые условия её функционирования, то нет возможности собрать фактические данные. В этом случае привлекаются специалисты, хорошо представляющие обсуждаемую проблему, которые способны выдвинуть гипотезы о возможных значениях таких параметров. Степень субъективности может быть уменьшена, если воспользоваться методиками экспертных оценок.

Этап сбора и обработки исходных данных завершается классификацией их на внешние и внутренние, постоянные и переменные, непрерывные и дискретные, линейные и нелинейные, стационарные и нестационарные, детерминированные и стохастические. Кроме того, определяются границы изменения для переменных количественных параметров, а для дискретных величин – их возможные значения.

2.4. Разработка математической модели

В первой главе подробно рассматривалось понятие математической модели. **Математическая модель** – это совокупность математических объектов и отношений, которые отображают объекты и отношения некоторой предметной области (области реального мира). Известно, что одними и теми же математическими моделями могут описываться совершенно различные по природе процессы.

Основой для разработки математической модели служат концептуальная модель и количественные исходные данные. Преследуются две цели: получить однозначное формализованное описание структуры и процесса функционирования системы, а также представить процесс функционирования с использованием одной из возможных обобщённых формализованных схем. Этими наиболее распространёнными схемами являются, например, непрерывные детерминированные системы, автоматы, агрегативные системы, системы массового обслуживания и пр.

Непрерывные детерминированные системы. Простейшим видом моделей таких систем является линейное соотношение (прямая пропорциональная зависимость) между двумя числовыми переменными $y = kx$, где k – коэффициент (числовой параметр), отражающий свойства модели. Использование такой зависимости позволяет описывать многие процессы в реальных системах (это и закон Ньютона в механике, и закон Ома в электротехнике).

В более общем случае, если не учитывается воздействие случайных факторов, а малые изменения входных воздействий приводят к такого же порядка малым изменениям выходного воздействия и состояниям системы, соотношения (1.1) можно представить в виде векторного дифференциального уравнения

$$\frac{d\mathbf{y}}{dt} = \mathbf{F}(\mathbf{x}(t), \mathbf{v}(t), \mathbf{h}(t), t), \qquad (1.6)$$

где $\mathbf{F}$ – вектор-функция закона функционирования системы; $\mathbf{x}, \mathbf{v}, \mathbf{h}, \mathbf{y}$ – векторы входных, внутренних и выходных воздействий соответственно.

В случае линейности систем, когда переменные обладают свойством однородности и аддитивности, вид уравнений (1.6) упрощается, что позволяет решать их аналитическими или численными (приближёнными) методами [6].

Автоматы. Автоматные модели используются для анализа функционирования и проектирования отдельных узлов вычислительных систем, для разработки программного обеспечения, а также могут быть использованы при создании моделей в других областях техники, биологии, медицины и пр. [4].

Автомат представляют как некоторое устройство (чёрный ящик), на которое поступают внешние воздействия (входные сигналы), изменяющие внутреннее состояние автомата, причём реакция автомата на каждое воздействие (выходной сигнал автомата) зависит, как от конкретного значения воздействия, так и от состояния автомата.

Множества дискретных воздействий, реакций и состояний представляются в теории абстрактных автоматов с помощью соответствующих алфавитов: алфавита (множества) входных символов $V = \{v\}$, алфавита (множества) выходных символов $W = \{w\}$, а также алфавита (множества) состояний автомата $S = \{s\}$. Каждый входной символ $v \in V$ отображает входное воздействие, а выходной символ $w \in W$ – реакцию автомата.

Предполагается, что множества V и W конечны. Если множество S конечно, то автомат называется конечным, а если S бесконечно, то автомат называется автоматом с бесконечным числом состояний. Если автомат имеет всего лишь одно состояние, то он называется тривиальным или автоматом без памяти.

В процессе функционирования в результате очередного входного воздействия автомат с памятью изменяет своё состояние и вырабатывает выходную реакцию в дискретные моменты времени.

Поскольку конкретные значения времени в модели абстрактного автомата несущественны, используют дискретное автоматное время t, принимающее последовательно целочисленные значения 0, 1, 2, …, причём $t = 0$ соответствует начальному моменту, с которого рассматривается функционирование автомата.

Сети Петри предназначены для моделирования дискретных асинхронных процессов. Их основные отличия от автоматной модели – возможность отображать параллелизм, асинхронность и иерархичность моделируемых объектов.

Агрегативные системы. Это широко распространённый подход, позволяющий описывать функционирование непрерывных и дискретных, детерминированных и стохастических систем. Агрегат представляется основным элементом любой системы (это агрегатное представление системы) [7]. Общая схема агрегата изображена на рис. 1.2.

Свойства и функционирование любого агрегата определяют следующие факторы:

1. Входная информация, обрабатываемая агрегатом. Эта информация представляет собой сообщения вида a_j^k ($k = 1, 2, \ldots, k^*$), поступающие в моменты времени t_j ($j = 1, 2, \ldots, j^*$). В общем случае входная информация может быть функцией времени и представлена ($k^* + 1$) – мерным вектором

$$\bar{a}_j^k = \bar{a}_j^k (t_j, a_j^1, a_j^2, \ldots, a_j^{k^*}).$$

Каждое из сообщений $\left\{ \bar{a}_j^k \right\}$ в зависимости от вида агрегата может характеризоваться законом распределения моментов времени поступления, типом, приоритетом, длиной, параметром обслуживания и пр.

2. Входная информация, управляющая функционированием агрегата. Эта информация представлена сообщениями вида g_i^n ($n = 1, 2, \ldots, n^*$), поступающими в моменты времени τ_i $(i = 1, 2, \ldots i^*)$, и определяет изменения алгоритма G. Данные сообщения также являются функциями времени и могут быть заданы ($n^* + 1$) – мерным вектором

$$\bar{g}_i^n = \bar{g}_i^n(\tau_i, g_i^1, g_i^2, ..., g_i^{n^*}).$$

Сообщения, управляющие функционированием агрегата, могут характеризоваться законом распределения моментов поступления, длительностью, типом и другими факторами.

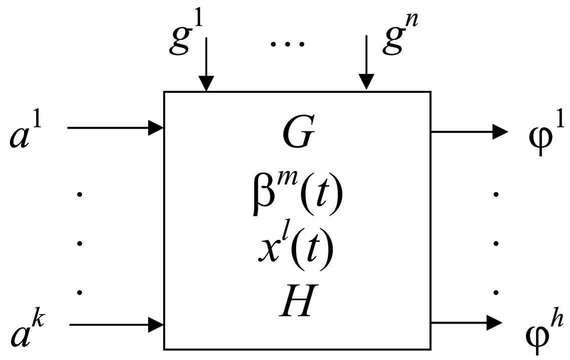


Рис. 1.2. Общая схема агрегата

3. Координаты агрегата, то есть совокупность некоторых величин x^l $(l = 1, 2, ..., l^*)$, определяющих состояние агрегата в каждый момент времени t. Координаты агрегата также можно представить в виде $(l^* + 1)$ – мерного вектора

$$\bar{x}_i^l = \bar{x}_i^l(t_i, g_i^1, g_i^2, ..., g_i^{l^*}).$$

В зависимости от вида агрегата координаты также могут быть разбиты на ряд групп в зависимости от их типа.

4. Выходная информация агрегата, выдаваемая им в другие агрегаты или внешнюю среду. Эта информация представляется сообщениями вида φ_r^h $(h = 1, 2, ..., h^*)$, выдаваемыми агрегатом в моменты времени t_r $(r = 1, 2, ..., r^*)$. Она также является функцией времени и может быть представлена $(h^* + 1)$ – мерным вектором

$$\bar{\varphi}_r^h = \bar{\varphi}_r^h(t_r, \varphi_r^1, \varphi_r^2, ..., \varphi_r^{h^*}).$$

Сообщения выходной информации также могут характеризоваться типом, приоритетом, длиной и другими факторами.

5. Параметры агрегата, представляющие собой совокупность величин $\beta^m(t)$, которые являются выходной информацией агрегата и определяют значения его вектора $\overline{x}^l$.

6. Алгоритм G, реализуемый агрегатом, который в зависимости от t_j, $\overline{a}_j^k$ и $\overline{x}_i^l(t)$ определяет $\overline{\varphi}_r^h$ и новые значения параметров $\beta^m(t)$ (алгоритм выходов).

7. Алгоритм H, позволяющий по известным параметрам $\beta^m(t)$, а также значениям величин t_j и $\overline{a}_j^k$ определить координаты агрегата $\overline{x}^l(t)$ для любого момента времени t (алгоритм переходов).

Таким образом, агрегат можно определить как объект

$$A = A\left\{\overline{a}^k, \overline{\varphi}^h, \overline{x}^l, G\left[\overline{g}^n\right], H\left[\overline{\beta}^m\right], t\right\},$$

состоящий из непустых множеств $\left\{\overline{a}^k\right\}, \left\{\overline{\varphi}^h\right\}, \left\{\overline{x}^l\right\}, \left\{\overline{g}^n\right\}, \left\{\overline{\beta}^m\right\}$ и алгоритмов G и H, определённых соответственно на множествах $\left\{\overline{a}^k\right\}, \left\{\overline{x}^l\right\}, \left\{\overline{g}^n\right\}$ и $\left\{\overline{a}^k\right\}, \left\{\overline{\beta}^m\right\}$.

Здесь $\left\{\overline{a}^k\right\}$ – множество входных сигналов, $\left\{\overline{g}^n\right\}$ – множество управляющих сигналов, $\left\{\overline{x}^l\right\}$ – множество состояний агрегата, $\left\{\overline{\beta}^m\right\}$ – множество его параметров, а $\left\{\overline{\varphi}^h\right\}$ – множество выходных сигналов.

Изложенное описательное определение агрегата является общим. На практике часто приходится иметь дело с агрегатами более простого типа. Их функционирование и свойства описываются лишь частью перечисленных факторов, то есть они будут частными случаями описанного агрегата.

Известны работы, направленные на оптимизацию конкретных систем агрегатным представлением их элементов [24].

Системы массового обслуживания. В основу построения моделей систем массового обслуживания (СМО) положена формализованная схема, которая подразумевает наличие обслуживающих приборов (каналов обслуживания), входного потока заявок на обслуживание, возможно очереди из заявок, ожидающих начала обслуживания, и выходного потока обслуженных заявок или заявок, получивших отказ [7, 11, 28, 33, 43]. Разработана и широко используется в повседневной практике для создания и исследования математических моделей соответствующая теория – теория массового обслуживания. Это позволяет во многих случаях широко использовать СМО для построения моделей.

Таким образом, для построения математической модели может использоваться одна из выше описанных формализованных схем. Выбор той или иной схемы осуществляется в результате анализа концептуальной модели и исходных данных.

Глава 3
КЛАССИФИКАЦИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

Модели, используемые в задачах математического моделирования, можно очень условно разделить на три основных класса: аналитические, имитационные и нечёткие (семиотические). В первом случае устанавливаются формульные, аналитические зависимости между параметрами системы. Для описания этих зависимостей разработан язык алгебраических, дифференциальных, интегральных и др. уравнений. В терминах аналитических моделей поставлены и решены достаточно простые управленческие задачи, в основном планирования на макроинтервалах времени. Эти модели можно получить, например, в рамках математического программирования (линейное, целочисленное, нелинейное, динамическое, стохастическое) и теории массового обслуживания.

Для задач, требующих учета большого количества факторов, в том числе и случайных или нечётких (неопределённых), разработаны методы имитационного и нечёткого моделирования [30].

Практика решения сложных прикладных задач показала, что наилучших результатов добиваются при использовании двухуровневых моделей. Аналитические модели за счет огрубления действительности позволяют сосредоточить внимание на существе явления, его основных закономерностях, а уточнение и конкретизация решений выполняется на статистических моделях (см. части 2 и 3 данного учебного пособия).

Однако в целом ряде случаев (например, при моделировании работы диспетчера в больших системах, когда требуется учитывать не только аналитические связи и случайные факторы, но и широкий спектр пространственных, временных, причинно-следственных и других отношений, а также нечеткие понятия естественного языка) рассмотренные выше методы оказываются малопригодными. Для автоматизации задач этого класса интенсивно развивается аппарат нечёткого (семиотического) моделирования, основанный на близком к естественным языкам представлении знаний и процедурах логического вывода решений с использованием временных, пространственных и других логик [30].

Каким образом принимаются решения при поиске тех или иных вариантов поведения системы, устройства или человека? Принятые решения обычно могут быть хорошими, плохими, удовлетворительными. То есть всегда существует потребность оценить качество принятого решения. Это приводит к необходимости использования понятия **критерия.**

Обычно задачи, встающие перед человеком, рано или поздно решаются. Сам процесс принятия решения может быть неформализованным и формализованным [23]. Принятие неформализованных решений – это творчество, искусство. Очень часто решения принимаются вообще без всяких обоснований. При этом руководствуются опытом, здравым смыслом, интуицией и пр.

Формализованные же решения принимаются по определенным правилам и рекомендациям. Различные люди, используя эти правила, принимают **одинаковые решения.** Потому что принятие формализованных решений – это наука. Ее можно изучить и использовать в дальнейшем.

Принятие формализованных решений базируется на двух основных подходах – логическом моделировании и оптимизации.

В основе логического моделирования лежит использование так называемых правил продукции вида: ЕСЛИ …, ТО … и некоторых логических функций И, ИЛИ, НЕ. Пример, одного из правил:

ЕСЛИ животное имеет крылья

И животное умеет летать,
ТО название животного – птица.

Использование набора подобных правил позволяет давать достаточно конкретные рекомендации по принятию решений в весьма сложных ситуациях. Реализация логического моделирования стала довольно простой и перспективной после создания специального алгоритмического языка PROLOG (PROgramming in LOGic) [34, 35].

Оптимизация или принятие оптимальных решений базируется на использовании математической модели, решении задач на компьютере и на исходных данных. О математическом моделировании и о моделях достаточно подробно сказано в главах 1 и 2.

Дополнительно к сказанному, к существенным преимуществам математического моделирования относят возможность:

1) получать быстрый ответ на поставленный вопрос (в реальной жизни на это можно затратить годы);

2) проводить эксперименты, которые на реальном объекте просто невозможны.

Для успешного моделирования необходимо выполнение трех правил:

– уметь отделить главные свойства от второстепенных,
– учитывать главные свойства моделируемого объекта,
– пренебрегать его второстепенными свойствами.

С этапами составления математической модели и методикой моделирования можно ознакомиться в предыдущей главе.

Как выглядит постановка задачи оптимизации в общем виде? В нее входят три составляющие: целевая функция, ограничения и граничные условия [22].

Целевая функция (ЦФ) или критерий оптимальности показывает в каком смысле решение должно быть оптимальным, т.е. наилучшим. При этом возможны три вида назначения целевой функции f: максимизация, минимизация и достижение заданного значения, т.е.

$$f = f(\mathbf{x}) \to \max\ (\min, cons),$$

где $\mathbf{x} = (x_1, x_2, ..., x_n)$ – n-вектор, f – функция, аргументы которой – допустимые значения переменных, а её значения – числа, которые описывают меру достижения поставленной цели. ЦФ показывает, в каком смысле решение должно быть оптимальным.

Ограничения устанавливают зависимости между переменными. Они могут быть как односторонними, например

$$g_i(\mathbf{x}) \le b_i \ \ (i = 1, 2, \ldots, m),$$

так и двусторонними

$$a_i \le g_i(\mathbf{x}) \le b_i \ \ (i = 1, 2, \ldots, m).$$

Граничные условия показывают, в каких пределах могут быть значения искомых переменных в оптимальном решении

$$d_j \le x_j \le D_j; \quad i = 1, 2, ..., m.$$

Описанная постановка задачи оптимизации, может быть записана в компактной форме:

$$\begin{aligned} & f = f(\mathbf{x}) \to \max, \\ & g_i(\mathbf{x}) \le b_i, \\ & d_j \le x_j \le D_j, \\ & i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \end{aligned}$$

Решение задачи оптимизации, удовлетворяющее всем ограничениям и граничным условиям, называется **допустимым.** Если математическая модель задачи оптимизации составлена правильно, то задача будет иметь целый ряд решений. Чтобы из всех возможных решений выбрать только одно – наилучшее (оптимальное) решение, надо четко представлять в чем заключается оптимальность решения, т.е. по какому критерию принимаемое решение должно быть оптимальным.

Задача принятия решения сводится к нахождению максимального (или минимального) значения целевой функции (критерия), а также к нахождению того конкретного решения (совокупности аргументов), при котором это значение достигается, т.е. является оптимальным. Итак, задача имеет оптимальное решение, если она удовлетворяет двум требованиям: есть реальная возможность иметь более одного решения, т.е. существуют допустимые решения, и имеется критерий, показывающий, в каком смысле принимаемое решение должно быть оптимальным, т.е. наилучшим из допустимых.

Как осуществить классификацию математических моделей? Например, можно производить классификацию по виду математических моделей, учитывая используемые ими исходные данные, искомые переменные и зависимости.

Исходными данными для математической модели являются: целевая функция $F(\mathbf{x})$, левые части ограничений $g_i(\mathbf{x})$ и их правые части b_i. Эти исходные данные могут быть детерминированными и случайными. **Детерминированными** называются такие исходные данные, когда при составлении модели их точные значения известны. И наоборот, **случайными -** когда их значения неизвестны.

Искомые переменные могут быть непрерывными и дискретными. **Непрерывные** величины в заданных граничных условиях могут принимать любые значения. **Дискретными** называются такие переменные, которые могут принимать только заданные значения. **Целочисленными** называются такие дискретные переменные, которые могут принимать только целые значения.

Зависимости между переменными (как целевые функции, так и ограничения) могут быть линейными и нелинейными. **Линейными** называются такие зависимости, в которых переменные входят в первой степени и с ними выполняются только действия сложения и вычитания. Если же переменные входят не в первой степени или с ними выполняются другие действия, то зависимости являются **нелинейными.** Необходимо иметь в виду, что, если хотя бы одна зависимость нелинейная, то вся задача является нелинейной.

Сочетание различных элементов модели образует различные классы задач оптимизации, которые требуют разных методов решения. Основные классы задач оптимизации приведены в табл. 1.1.

Все классы задач, приведенные в табл. 1.1, являются частными случаями общей задачи оптимизации. Остановимся на краткой характеристике классов математических моделей и методов.

Если известно аналитическое выражение математической модели объекта, а также формально записан критерий оценки качества управления, то решение задачи оптимизации не вызывает принципиальных трудностей. В частности, постановка задачи линейного программирования [20, 22] сводится к поиску при заданных ограничениях набора переменных (допустимое решение), которые доставляют экстремум целевой функции (оптимальное решение). При этом как ограничения, так и целевая функция должны быть заданы линейными алгебраическими уравнениями (см. табл. 1.1). Ниже приведен канонический вариант формулировки линейной модели – задачи линейного программирования.

Таблица 1.1

Исходные данные	Искомые переменные	Зависимости	Классы задач
Детерминированные	Непрерывные	Линейные	Линейного программирования
Детерминированные	Целочисленные	Линейные	Целочисленного программирования
Детерминированные	Непрерывные, целочисленные	Нелинейные	Нелинейного программирования
Случайные	Непрерывные	Линейные	Стохастического программирования

Найти вектор $\mathbf{x} = (x_1,\ x_2,\ ...,\ x_n)$, удовлетворяющий условиям:

$$x_j \geq 0, \quad j = 1,2,...,n;$$

$$g_i(\mathbf{x}) = \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1,2,...,m$$

и максимизирующий целевую функцию

$$f(\mathbf{x}) = \sum_{j=1}^{n} c_j x_j \ .$$

Среди методов нахождения оптимального решения наиболее распространен метод последовательного улучшения допустимого решения, имеющий большое число вычислительных реализаций (симплекс-метод, метод обратной матрицы, мультипликативный метод, метод потенциалов и др.). В задачах с большим количеством переменных используются специальные методы проверки оптимальности решения, например, динамическое программирование.

Если на искомые переменные наложено дополнительное условие целочисленности, то приходим к постановке задачи целочисленного (диофантного, дискретного) программирования. В случае, когда не на все переменные наложено это условие, имеем постановку задачи частично целочисленного программирования. Дискретность и целочисленность возникают и в задачах с логическими условиями.

Со значительными оговорками методы решения задач целочисленного программирования можно разделить на методы отсечения: три алгоритма Гомори и их модификации [22], комбинаторные и приближенные (случайный поиск, эвристические приемы).

В связи с постановкой и решением линейных задач часто встречается термин «блочное программирование». По существу, это раздел линейного программирования, охватывающий ряд приемов, сводящих решения линейных задач большого объема к решению линейных задач меньшего объема. Если целевая функция и (или) ограничения нелинейные, а процесс одноэтапный, приходим к постановке задач нелинейного программирования.

Основная задача нелинейного программирования формулируется следующим образом: при заданных функциях $f(\mathbf{x})$, $g_1(\mathbf{x})$, $g_2(\mathbf{x})$, ..., $g_m(\mathbf{x})$ от n действительных переменных ($\mathbf{x} - n$- мерный вектор) определить такой вектор $\mathbf{x} = (x_1, x_2, ..., x_n)$, удовлетворяющий условиям:

$$x_j \geq 0, \ \ j = 1,2,...,n;$$

$$g_i(\mathbf{x}) \leq 0, \ \ i = 1,2,...,m$$

и доставляющий глобальный минимум целевой функции $f(\mathbf{x})$.

В нелинейном программировании, в общем случае, допускаются любые соотношения между n и m, т.е. $n > m$, $n = m$, $n < m$.

Теоретически нелинейное программирование исследовано только для выпуклых функций $f(\mathbf{x})$ и $g_i(\mathbf{x})$. Этот раздел называется выпуклым программированием, в котором наиболее развито квадратичное программирование. Методы решения задач квадратичного программирования включают три группы: алгоритмы, использующие симплекс-метод (например, метод Баранкина-Дорфмана, градиентные и специальные методы).

Одним из гибких методов поиска оптимального решения является динамическое программирование, в отличие от линейного программирования, не требующее линейности исходных зависимостей. Однако метод основан на принципе оптимальности Беллмана и применим только к функциям, для которых заведомо этот принцип выполним (например, сепарабельным). В случае динамического программирования задача оптимального управления решается разбиением непрерывного процесса управления на дискретные участки и построением поэтапного управления путем поэтапного решения соответствующего функционального управления. На каждом этапе, начиная с конца, делают перебор управлений из класса допустимых и выбирают оптимальное.

Очень часто в моделях математического программирования некоторые переменные могут оказаться неопределенными или случайными. В одних случаях удается установить те или иные вероятностные характеристики параметров условий задачи. В других – нет оснований для суждений о статистической особенности явлений, способных изменить значения параметров. Ситуации первого типа называются ситуациями, связанными с риском, а ситуации второго типа – неопределенными. И те и другие – предмет исследования стохастического программирования.

Рассмотрим постановку одноэтапной задачи стохастического программирования:

$$\sum_{j=1}^{n} a_{ij}(\theta)x_j \le b_i(\theta),\ \ x_j \ge 0;$$

$$\sum_{j=1}^{n} c_j(\theta)x_j \to \max; \quad j=1,2,..,n; \quad i=1,2,..,m.$$

Величины $a_{ij},\ b_i,\ c_j$ могут принимать случайные значения в зависимости от значения параметра θ.

Развитие методов решения задач стохастического программирования идет по двум направлениям. Для одних классов стохастических задач строят эквивалентные детерминированные задачи математического программирования и применяют известные алгоритмы. Для других – разрабатывают специальные методы, использующие теоретико-множественную интерпретацию теоретико-вероятностных задач. Более подробно с теорией вопроса можно познакомиться в [8, 10].

Качественно новым направлением развития являются нечёткие (семиотические) модели, базирующиеся на двух основных гипотезах:

1) все сведения об объекте управления (структура, состояние, законы поведения, цель существования, критерий оценки качества функционирования) могут быть сообщены компьютеру на естественном языке [30, 34];

2) невозможность создания замкнутой управляющей системы, т.е. процесс формирования модели не может быть закончен в какой-либо из моментов времени (непрерывность обучения модели).

Нечёткую (семиотическую) модель можно задать в виде расширения обычной формальной дедуктивной системы

$$S = < T, \Omega, A, E >,$$

где T – термины формальной системы; Ω – правила построения правильных формул (ППФ); A – аксиомы (истинно интерпретированные ППФ); E – правила вывода.

При заданной начальной интерпретации в виде аксиом правила E позволяют породить все истинные правила в S формулы.

Семиотические модели в отличие от S допускают возможность договорной интерпретации, что достигается изменением системы аксиом (подсистема F) или системы правил вывода (подсистема L)

$$M = < T, \Omega, A, E, F, L >.$$

Для описания процессов, протекающих в M, применяются языки традиционной логики (например, исчисление предикатов первого порядка [3, 43]), языки сетевого типа (например, семиотические сети), а также реляционные языки (например, алгебра Кодда, язык ситуационного управления) [22].

Для реализации многих современных задач, связанных с решением проблем «человек и компьютер», «человек и техника», необходимо, чтобы, с одной стороны, компьютеры понимали людей, с другой - каждый мог пользоваться их возможностями. К сожалению, современные компьютеры понимают только специальные чёткие языки, строго математически описывающие решаемые проблемы и знания, вкладываемые человеком в компьютер. Но почти все понятия в естественном языке нечётки, также как нечётки наши знания в большинстве областей. Подобное различие между естественным и машинными языками сдерживает процесс информатизации общества, в частности ограничивает круг пользователей компьютеров.

Математическая теория нечётких множеств, предложенная Л. Заде, позволяет описывать нечёткие понятия и знания, оперировать этими знаниями и делать нечёткие выводы [30]. Основанная на этой теории новая методология построения компьютерных систем, а именно нечётких систем, существенно расширяет области применения компьютеров. Такие системы уже создаются во многих областях человеческой деятельности.

Почему нечёткое моделирование (применение нечётких моделей) находит широкое применение? В настоящее время теория нечетких систем – это единственная теория, которая математически оперирует со смысловым содержанием слов человека. Можно привести примеры человеко-машинных систем, которые предназначены для обработки нечётких знаний. В технических областях – автоматическое управление, автоматический перевод, интеллектуальные роботы, системы поддержания целостности баз данных и системы обеспечения безопасности, распознавание изображений и речи, автоматическое проектирование, поиск информации, базы знаний, интеллектуальные терминалы, автоматизация домашних работ и др. В сфере бизнеса – помощь в принятии экономических решений, маркетинг, советы по вложению капитала, различного рода управление и планирование, помощь в подготовке контрактов и др. Кроме того, оценка состояния окружающей среды, анализ риска, предсказание землетрясений, прогнозы погоды, системы самообучения, дегустации, обработка данных анализа и т. д.

Теперь настало время сделать одно очень важное замечание (допущение). В связи с тем, что в рамках одного учебного пособия невозможно описать всё многообразие существующих подходов к построению и исследованию математических моделей систем, в дальнейшем изложении будет сделан акцент на осуществлении моделирования только систем массового обслуживания. Этим вопросам посвящены все остальные семь глав данного учебного пособия.

Глава 4
МОДЕЛИ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

Формальное описание процессов в виде системы массового обслуживания широко применяется в самых различных областях науки и практики. Теоретической базой построения и исследования СМО является теория массового обслуживания (ТМО).

Теория массового обслуживания – это область прикладной математики, занимающаяся анализом процессов в СМО, в которых события повторяются многократно. С помощью этой теории разрабатываются методы решения типовых задач массового обслуживания, строятся модели СМО и определяются их количественные характеристики.

Методами ТМО анализируют функционирование объекта, а затем решают вопрос о синтезе обслуживающих устройств и о выборе оптимальных параметров системы [8, 9, 11, 14, 20, 33].

Структурная схема СМО и её компоненты

Можно рассмотреть работу некоторого склада, в котором есть несколько (S) разгрузочных терминалов (пусть $S = 10$). Используя теорию массового обслуживания, можно сказать, что имеется СМО с S каналами обслуживания. Длительность обслуживания транспортных средств (длительность разгрузки) в зависимости от типа средства и его груза колеблется от 1 до 2 часов. Когда все терминалы заняты, то прибывший транспорт становится в очередь на разгрузку. Каждое транспортное средство имеет график прибытия, но из-за тяжёлой обстановки на дорогах (непредвиденные обстоятельства) этот график, обычно, нарушается. Поэтому считается, что прибытие транспорта является случайным событием с какими-то значениями математического ожидания и дисперсии.

В случае, когда весь транспорт одинаков (с точки зрения приоритетности обслуживания) считают, что в СМО поступает однородный поток заявок, т.е. все заявки на разгрузку равноценны. Но часто возникают и некоторые неоднородности. Например, в случае поступления скоропортящегося груза заявке на обслуживание присваивается некоторый приоритет (пришедший последним обслуживается первым). Простейшая схема СМО изображена на рис. 1.3. В общем случае интервалы между поступающими заявками неодинаковы: это случайные величины, определяемые вероятностным законом распределения входного потока. Заявки, вставшие в очередь, ожидают начала обслуживания в соответствии с дисциплиной очереди (дисциплиной обслуживания). Они могут быть различными: первый пришёл – первый идёт на обслуживание или, наоборот, последний пришёл – первый обслуживается.

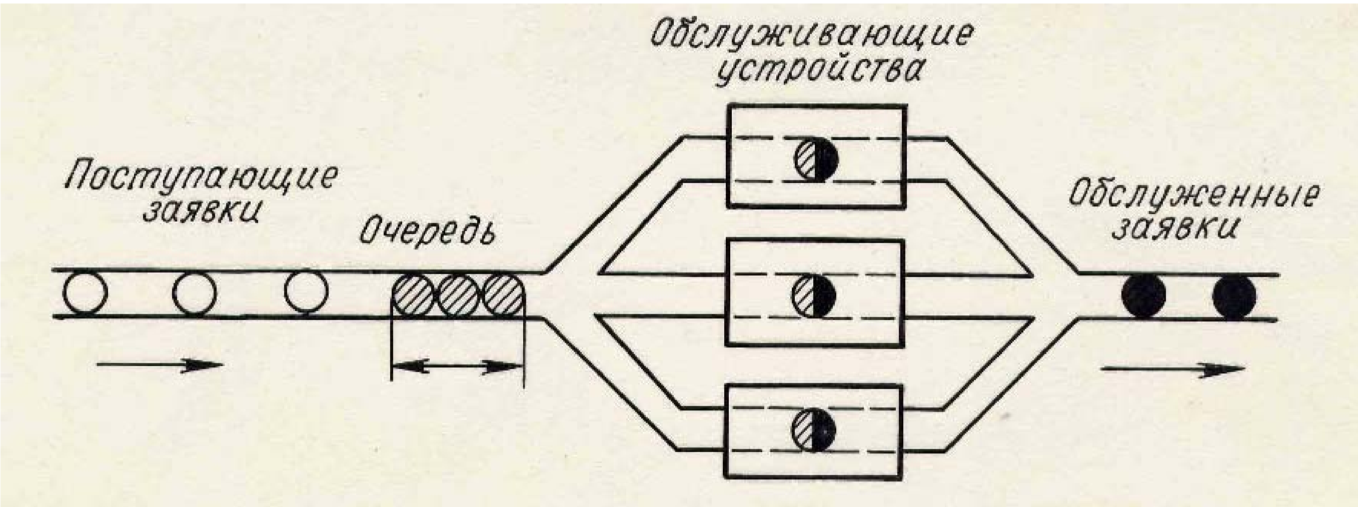


Рис. 1.3. Структура простейшей СМО

Обслуживающие устройства (приборы, каналы) производят обслуживание своей заявки (поступившей на вход) в соответствии с заданным детерминированным или случайным законом распределения. Обслуженная заявка поступает в выходной поток, который отличается от входного и зависит от дисциплин очереди и обслуживания.

Характерная особенность СМО – все явления описываются с помощью событий, появляющихся в те или иные моменты времени. На временной оси входной поток заявок отмечается как последовательность событий, выборка заявок из очереди или окончание обслуживания – это тоже события в соответствующие моменты времени. Важным обстоятельством здесь является абстрагирование от всех прочих несущественных свойств реальной системы, не вписывающихся в схему последовательности событий.

4.1. Классификация систем массового обслуживания

Можно использовать большое разнообразие моделей СМО и подходов к их классификации. Первоначально модели разделяют на марковские и немарковские. Это направление связано с классом некоторых марковских случайных процессов. Модели СМО, в которых протекают именно марковские случайные процессы описываются системой дифференциальных уравнений, или, в предельном случае, системой линейных алгебраических уравнений. Решение этих уравнений даёт возможность легко получить набор «красивых», легко запоминающихся, выражений, определяющих показатели эффективности функционирования СМО (см. главы 8 и 9 данного пособия). При наличии немарковских процессов в СМО аналитическому исследованию поддаются лишь некоторые частные случаи и они в пособии не рассматриваются.

Обычно используются следующие классификационные признаки: организация потока заявок, характер образования очереди, ограничения очереди, количество обслуживающих каналов, дисциплина очереди (рис.1.4).

По числу каналов СМО подразделяются на одноканальные и многоканальные системы. Многоканальные СМО, в свою очередь, делятся на системы с одинаковыми параметрами каналов обслуживания (равноценными каналами) и системы с различными параметрами каналов обслуживания (неравноценными каналами).

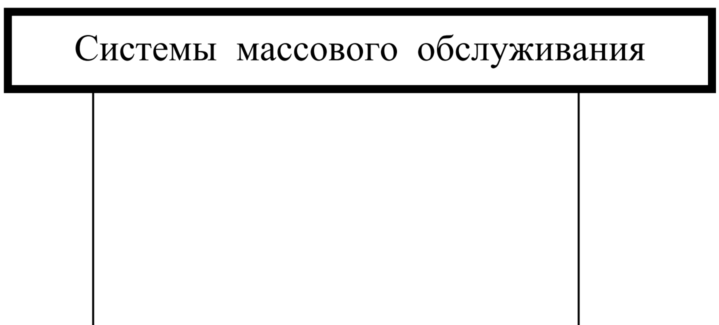


Рис. 1.4. Классификация систем массового обслуживания

В зависимости от взаимного расположения каналов СМО подразделяются на системы с последовательными и параллельными каналами. В СМО с параллельными каналами входной поток является общим, и заявка может обслуживаться любым свободным каналом. А в СМО с последовательным расположением каналов каждый канал можно рассматривать как одноканальную СМО, и тогда выходной поток одного из них одновременно является входным для последующего канала. Такие системы называются соответственно многофазными и однофазными. В общем случае с некоторыми допущениями многофазная СМО является несколькими последовательно соединёнными однофазными системами. В дальнейшем будут рассматриваться только однофазные СМО и, если они многоканальные, то – с равноценными каналами.

В зависимости от характера образования очереди СМО можно разделить на два типа: системы с отказами в обслуживании и системы с ожиданием обслуживания, т.е. с очередью. В первом случае заявка, поступившая в СМО и заставшая канал (для одноканальной системы) занятым или – каналы (для многоканальной системы) занятыми, получает отказ и уходит из СМО не обслуженной.

Особенностью СМО с ожиданием является характеристика самой очереди (ограничения на очередь). Такие системы подразделяются на СМО с неограниченным ожиданием (неограниченным количеством мест в очереди) и СМО с ограниченным ожиданием (ограниченным количеством мест в очереди). Первый вариант ограничения заключается в том, что заявка всегда ставится в очередь и когда-то будет обслужена. Второй вариант для постановки заявки в очередь требует наличия свободных мест в ней. И только в этом случае заявка ставится в очередь. Но возможен и другой вид ограничения – по времени ожидания заявки начала обслуживания. В этом случае заявка покидает систему не обслуженной, если длительность её возможного пребывания в очереди превышает заданный ресурс времени ожидания (это так называемые «нетерпеливые» заявки).

По дисциплине очереди (т.е. по критерию выбора из очереди очередной заявки) СМО подразделяются на системы с приоритетом и без приоритета. В первом случае заявке, пришедшей в СМО, присваивается некоторый приоритет, в соответствии с которым на обслуживание отправляется заявка с наивысшим приоритетом, несмотря на то, что, возможно, она пришла в систему после прихода прочих заявок. В этом случае на самом деле образуется несколько очередей с разным уровнем приоритета. При реализации дисциплины очереди – без приоритета возможны различные правила отбора заявок на обслуживание: первый пришел – первый обслужен, последний пришел – первый обслужен или в случайном порядке (например, с заданными вероятностями выборки из очереди).

Системы с приоритетом, в свою очередь, могут подразделяться на СМО с прерыванием и без прерывания. В системах с прерыванием заявка с высшим приоритетом прерывает обслуживание заявки с более низким приоритетом. Прерванная заявка может быть поставлена в очередь и, возможно, когда-нибудь будет обслужена в соответствии со своим приоритетом.

Удобно пользоваться формальным классификационным описанием СМО как абстрактной системой. При этом под абстрактной СМО понимают математический объект $\alpha|\beta|\gamma|\delta$, характеризующийся следующими четырьмя элементами: α – входящим потоком требований, β – потоком обслуживания, γ – структурой системы (количеством обслуживающих каналов), δ – дисциплиной обслуживания (характером образования очереди и наличием ограничения на очередь).

Характеристики первых двух элементов системы α и β будут достаточно подробно рассмотрены в § 4.3. Абстрактная система, например, вида M|M|3|5 – это СМО с марковскими входным потоком и потоком обслуживания, числом обслуживающих приборов, равным 3, числом мест в очереди, равным 5.

Показатели эффективности работы СМО

Качество функционирования любой системы определяется количественными показателями эффективности её работы. В зависимости от целей исследования эти показатели могут быть различными. Например, для оценки работы некоторой СМО можно использовать среднюю величину потерь C в единицу времени, происходящих в процессе эксплуатации системы:

$$C = k_1 c_1 + k_2 c_2,$$

где k_1 – среднее число заявок в очереди на обслуживание, c_1 – стоимость ожидания в очереди в единицу времени, k_2 – среднее число свободных каналов, c_2 – стоимость простоя одного канала в единицу времени.

Принято оценивать эффективность СМО по таким показателям, как:

– средняя длительность пребывания заявки в СМО – клиент выберет для своего обслуживания ту систему, в которой, при одинаковом качестве обслуживания, он проведёт меньше времени;

– вероятность наступления заданного события, например, вероятность того, что заявка, пришедшая в СМО, будет обслужена;

– среднее количество занятых каналов – если, например, среднее количество занятых каналов значительно меньше общего количества каналов в системе, то это говорит о том, что некоторые каналы постоянно простаивают, а это уже экономические потери;

– среднее количество занятых мест в очереди – при большом среднем количестве занятых мест в очереди велика вероятность отказа в обслуживании, а в этом случае возникает потребность в увеличении количества каналов или увеличении интенсивности обслуживания заявок имеющимся количеством каналов и пр.

Приведённый небольшой перечень характеристик уже свидетельствует о тесной их взаимосвязи и противоречивости. Улучшение показателей одних из них обычно влечёт ухудшение других. Отсюда следует, что при проектировании СМО желательно установить разумное равновесие между различными показателями эффективности системы.

Для решения этой задачи желательно выбрать обобщённый показатель (C) эффективности СМО, например критерий экономической эффективности, который включал бы и издержки обращения ($C_{\text{ио}}$), и издержки потребления ($C_{\text{ип}}$):

$$C = (C_{\text{ио}} + C_{\text{ип}}) \rightarrow \min.$$

Первое слагаемое обобщённого показателя включает затраты, например, на эксплуатацию системы и простой обслуживающих каналов. Второе слагаемое – потери, связанные с уходом необслуженных заявок и с пребыванием в очереди.

Методика расчёта отдельных характеристик СМО будет освещена ниже при построении моделей конкретных СМО.

4.2. Виды моделирования СМО

Аппарат ТМО развит для ситуаций, когда случайный процесс, протекающий в рассматриваемой системе, марковский или близок к марковскому. Целый ряд ограничений на характер входящего потока позволяет для реальных систем рассматривать лишь частные задачи (см. часть III данного учебного пособия). Математический же аппарат ТМО не всегда дает соответствующие аналитические зависимости. В этих случаях применяют метод имитационного моделирования (метод статистических испытаний, метод Монте-Карло). Общая схема применения метода, подробно рассматриваемого в части II данного учебного пособия, состоит в формировании и запоминании возможных значений некоторой случайной величины, зависящей от траектории случайного процесса. Ее среднее значение, полученное в результате выполнения большого числа реализаций процесса, и оказывается искомым решением.

Таким образом, наиболее распространёнными подходами к исследованию СМО являются аналитическое и имитационное моделирование. Ниже дана краткая вводная характеристика этих методов, а в частях II и III соответственно – более или менее подробное изложение.

Аналитические модели

Аналитическая модель – это математическое описание структуры и процесса функционирования системы, а также методика определения показателей её эффективности. Такая модель позволяет быстро и с высокой точностью характеризовать поведение системы.

Наиболее эффективны и наглядны аналитические модели при описании функционирования систем массового обслуживания. Под системой массового обслуживания (СМО) обычно понимают динамическую систему, предназначенную для эффективного обслуживания случайного потока заявок (требований на обслуживание) при ограничениях на ресурсы системы. Другими словами можно сказать, что особенностью массового обслуживания является наличие потока однородных (идентичных) событий (требований, заявок), которые подвергаются обслуживанию. Кроме того, для полного описания СМО и постановки задачи исследования необходимо определить структуру системы и дисциплину (правила) обслуживания, а также показатели качества обслуживания, т.е. некоторые числовые показатели, по значению которых можно было бы судить о качестве функционирования исследуемой СМО. Структурная схема системы массового обслуживания представлена на рис. 1.3.

Решением перечисленных проблем и задач занимается теория массового обслуживания, являющаяся разделом теории вероятностей. Вот несколько задач и примеров, относящихся к проблематике теории массового обслуживания:

1. В компьютер, используемый для управления технологическим процессом, время от времени поступают сигналы от датчиков,

связанных с управляемым объектом. Каждый сигнал требует обработки в течение некоторого случайного времени (зависящего от содержания сигнала). Таким образом, работу компьютера можно рассматривать как операцию массового обслуживания, состоящую из элементарных составляющих – обработки отдельных сигналов. Требуется решить задачу: способен ли компьютер с данным объёмом памяти и быстродействием справиться с обработкой всех поступающих сигналов?

2. В грузовой морской порт, имеющий несколько причалов, прибывают суда на разгрузку-погрузку в случайные моменты времени. Время разгрузки-погрузки (обслуживания) судов колеблется в некотором интервале в зависимости от типа судна и груза. Если все причалы заняты, то прибывшее судно становится на рейде и ждёт своей очереди. Каждый корабль имеет расписание прибытия, но из-за непредвиденных обстоятельств оно, как правило, нарушается. Поэтому считается, что прибытие судна является случайным событием с каким-то математическим ожиданием, дисперсией и т.д. Спрашивается: чему равна средняя длительность разгрузки-погрузки?

3. В сборочном цехе осуществляется сборка изделий различных видов. При нехватке хотя бы одного вида деталей производство останавливается. Избыточные детали поступают в бункеры определённой вместимости. На процесс поступления деталей, как и на время сборки изделия, влияют случайные факторы. Какова вероятность простоя производственной линии? Чему равна вероятность переполнения бункеров? Элементарной операцией в данном случае является сборка одного изделия из готового комплекта деталей.

Искомыми характеристиками СМО считаются показатели эффективности, под которыми понимаются количественные показатели, частично или полностью характеризующие уровень выполнения системой массового обслуживания возложенных на неё функций. Например, такие характеристики: вероятность обслуживания заявки (это вероятность того, что заявка, пришедшая в СМО, будет обслужена), вероятность отказа в обслуживании, среднее время ожидания заявки начала обслуживания, среднее время пребывания заявки в СМО, средняя длина очереди и пр.

На первый взгляд перечисленные характеристики свидетельствуют только о факте их вычисления. Но на самом деле это далеко не так.

Пусть вычисленная величина вероятности обслуживания заявки равна 0.65. О чём это говорит? Это значит, что из ста поступивших заявок на обслуживание только 65 из них будет обслужено. А это, в свою очередь, говорит о том, что 35 заявок получит отказ, т.е. треть из числа поступивших. Каждая третья заявка (каждый третий клиент) получит отказ в обслуживании. А это упущенная прибыль, упущенный клиент, упущенная возможность расширения сферы услуг.

Или другая характеристика: среднее количество занятых каналов. Пусть при количестве обслуживающих каналов равном трём, среднее количество занятых равно 1.75. Это свидетельствует о том, что, по крайней мере, один канал постоянно простаивает и, возможно, нужно задуматься о целесообразности его существования. Или противоположный случай: при трёх каналах среднее количество занятых равно 2.98. Это может свидетельствовать о том, что вероятность отказа в обслуживании велика (опять потери потенциальных клиентов) и, возможно, необходимо задуматься об увеличении количества обслуживающих каналов или об увеличении интенсивности обслуживания заявок каналами (например, покупка нового оборудования).

Часто бывает очень трудно получить искомые характеристики показателей эффективности СМО из-за сложных аналитических расчётов. Но если в СМО протекает марковский случайный процесс, то аналитическая модель (аналитические зависимости) записываются (запоминаются) просто и красиво.

Имитационные модели

Иногда аналитическое исследование поведения сложной системы оказывается весьма затруднительным вследствие сложности её математического описания. А в случае воздействия на систему случайных факторов трудности анализа часто становятся совершенно непреодолимыми. Тогда экспериментальное исследование поведения системы в реальных условиях её функционирования принципиально могло бы позволить получить наиболее полную и достоверную информацию о свойственных ей количественных и качественных закономерностях. Однако часто оно неосуществимо в случае вредности или удалённости системы от экспериментатора. Или экспериментальное исследование также невозможно, например, на стадии разработки системы (когда она ещё не создана). В этих случаях создание имитационной модели и проведение экспериментов на ней остаётся часто единственно возможным путём получения желаемых характеристик процесса функционирования системы.

Сущность имитационного моделирования сложной системы заключается в следующем. На основании описания функционирования системы и численных методов [6] разрабатывается моделирующий алгоритм, имитирующий внешние воздействия на систему, поведение её элементов, их взаимодействие и последовательное изменение состояний всей системы во времени. Разработанный моделирующий алгоритм реализуется затем на компьютере. Такой подход лишён всех недостатков приведённых выше приёмов исследования систем. Полнота и достоверность полученной путём имитационного моделирования информации о свойствах исследуемой системы зависят от точности математического описания системы, взятого в основу моделирования, и точности вычислительных методов, использованных при разработке моделирующего алгоритма.

Пример 1.1. Можно рассмотреть простейшую СМО, а именно очередь покупателей к контролёру (кассиру) в магазине. Пусть интервалы времени между последовательными появлениями покупателей распределяются равномерно в интервале [1, 10] мин, а длительность обслуживания – в интервале [1, 6] мин.

Необходимо определить среднее время t_0, которое покупатель проводит в системе (ожидание и обслуживание), и процент непроизводительного времени (простоя) кассира t_k (табл. 1.2).

Для моделирования системы искусственно создаётся эксперимент, отражающий условия задачи. Нужно создать процедуру имитации искусственной последовательности прибытия покупателей и времени для обслуживания каждого из них. Для этого потребуется десять фишек с номерами от 1 до 10 и обычный игральный кубик с шестью гранями. Поместив фишки в урну и доставая после перемешивания одну из них, считывают выпавшее число, которое и определит интервал времени между приходом очередных покупателей. Бросая кубик, можно искусственно получать длительность обслуживания клиентов. Возвратив фишку на место и встряхнув урну, а также повторяя другие операции, получают временные ряды, имитирующие временные интервалы между прибытием клиентов, а также соответствующие времена длительности обслуживания.

Таким образом, задача сводится к регистрации и обработке результатов искусственного (имитационного) эксперимента. Вопросы определения размеров выборки для получения статистически значимых результатов будут рассмотрены ниже (см. § 5.6).

Таблица 1.2

Покупатель	Интервал между моментами прибытия покупателей, мин	Длительность обслуживания, мин	Модельное время в моменты прибытия покупателей, мин	Начало обслуживания, мин	Конец обслуживания, мин	Длительности пребывания покупателей на обслуживании, мин	Длительности простоя кассира, мин
1	-	1	0	0	1	1	0
2	3	4	3	3	7	4	2
3	7	4	10	10	14	4	3
4	3	2	13	14	16	2	0
5	9	1	22	22	23	1	6
6	10	5	32	32	37	5	9
7	6	4	38	38	42	4	1
8	8	6	46	46	52	6	4
9	8	1	54	54	55	1	2
10	8	3	62	62	65	3	7
					Всего	31	34

$$t_0 = 31/10 = 3.1 \text{ мин}; \qquad t_k = (34/35)\cdot 100 = 52.3\%.$$

Можно подвести итог сказанному. Имитационное моделирование, являясь разновидностью численных приёмов решения различных задач, получило в последнее время очень широкое распространение. Это объясняется рядом его преимуществ:

– метод позволяет получить ответы на многие вопросы, возникающие на стадии замысла и предварительного проектирования будущей системы без дорогостоящих проб и ошибок;

– имитация на компьютере позволяет исследовать особенности функционирования системы в любых возможных условиях, при этом параметрами системы и окружающей среды можно варьировать для получения любой обстановки, в том числе, и нереализуемой в натурных экспериментах;

– применение компьютера сокращает продолжительность испытаний системы, занимающих в реальных условиях дни или месяцы, до долей минут и секунд;

– так как анализ некоторых очень сложных систем и их оценки не могут быть выполнены ни с помощью лабораторных, или натурных экспериментов, ни аналитическими методами, то во многих случаях реализация имитационных моделей на компьютерах является единственным реализуемым способом решения этих задач.

4.3. Потоки событий

Переходы СМО из одного состояния в другое происходят под воздействием вполне определённых событий – поступление заявок и их обслуживание.

Потоком событий называется последовательность однородных событий, следующих одно за другим в некоторые моменты времени. Например, это поток отказов или поток заявок на обслуживание вычислительного комплекса. События потока происходят в заранее неизвестные (обычно случайные) моменты времени t_1, t_2, ..., поэтому поток событий удобно изображать рядом точек на оси времени t (рис.1.5).

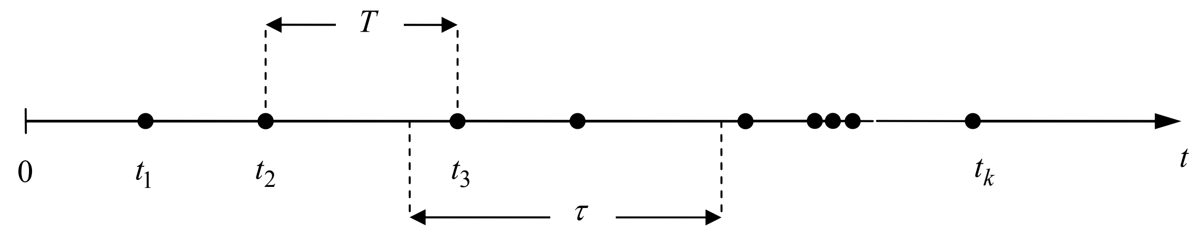


Рис. 1.5. Ось времени t

Регулярным (или детерминированным) потоком событий называется поток, в котором события следуют одно за другим через одинаковые промежутки времени.

Важной характеристикой любого потока является число m событий, происшедших за время τ. Если m случайная величина с возможными значениями $x \in \{0, 1, 2, ...\}$, то рассматривают или вероятности $p(m=x)$, или математические ожидания m_x и дисперсии σ_x^2, зависящие от τ.

Поток событий называется **стационарным,** если при любом x величина $p(m=x)$ зависит только от длины участка τ и не зависит от его расположения на оси времени.

Ординарным потоком событий называется такой поток, для которого вероятность $p(m \geq 2)$ попадания на отрезок τ пренебрежимо мала по сравнению с вероятностью $p(m=x)$. То есть практически невозможно попадание двух и более событий на достаточно малый интервал τ.

Потоком без последействия называется поток событий, в котором для любых двух непересекающихся интервалов τ_1 и τ_2 оси t число событий m_1, попадающих на τ_1, не зависит от того, сколько событий m_2 попало на τ_2. Это говорит о том, что события, образующие поток, появляются в последовательные моменты времени независимо друг от друга.

Поток событий называется **простейшим** (или стационарным пуассоновским), если он обладает одновременно тремя свойствами: стационарности, ординарности и отсутствием последействия. Интерес к этому потоку вызван простотой формального описания его статистических характеристик. Кроме того, при сложении нескольких независимых случайных потоков, стационарных и ординарных, образуется суммарный, который по своим характеристикам приближается к простейшему. Поэтому, при исследовании реальных потоков стремятся свести их к простейшим. Для этого нестационарные потоки представляются как стационарные на ограниченном отрезке времени. Неординарные потоки сводят к ординарным, рассматривая несколько одновременно наступающих событий как одно событие, и т. д.

Нестационарным пуассоновским потоком является поток событий, который не имеет последействия, ординарен, но не стационарен.

В простейшем потоке (как стационарном, так и нестационарном) величина m подчиняется распределению Пуассона:

$$p(m=x) = \frac{(\lambda\tau)^x}{x!} e^{-\lambda x}.$$

Это вероятность того, что в промежуток времени τ попадет ровно x событий, λ – интенсивность потока (среднее число событий в единицу времени).

Если a – среднее число событий, приходящееся на промежуток τ, то для стационарного простейшего потока $a = \lambda\tau$, а для нестационарного пуассоновского потока $\lambda = \lambda(\tau)$ и

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

Важной характеристикой потока событий является закон распределения длины промежутка между соседними событиями. Для простейшего потока с интенсивностью λ длина этого промежутка T (рис.1.5) распределена по показательному (экспоненциальному) закону с плотностью:

$$f(t) = \begin{cases} 0, \ t < 0, \\ \lambda e^{-\lambda t}, \ t \ge 0. \end{cases}$$

Числовые характеристики случайной величины, распределенной по показательному закону, а именно, математическое ожидание и дисперсия соответственно равны: $m_t = \dfrac{1}{\lambda}, \quad D_t = \sigma_t^2 = \dfrac{1}{\lambda^2}.$

Потоком Пальма (или потоком с ограниченным последействием) называется поток событий, для которого промежутки времени между последовательными событиями $T_1, T_2, \ldots, T_i, \ldots.$ представляют собой независимые, одинаково распределенные случайные величины.

Очевидно, что простейший поток представляет собой частный случай потока Пальма, когда интервалы между событиями имеют показательное распределение. Так, пусть имеется система, состоящая из какого-то количества элементов, которые независимо друг от друга выходят из строя. Неисправный элемент тут же заменяется новым. Поток неисправностей образует поток Пальма.

Особый класс потоков Пальма образуют **потоки Эрланга,** которые получаются путем прореживания простейших потоков, т.е. отбрасыванием некоторых событий как несостоявшихся. Если в простейшем потоке сохраняется каждое k – е событие (считая от условного первого), а остальные просто не учитываются, то возникает поток Эрланга k – го порядка (k = 1, 2, ...), обозначаемый Э$_k$. На рис. 1.6 приведен поток Эрланга 3-го порядка (два события простейшего потока пропускаются, а третье учитывается).

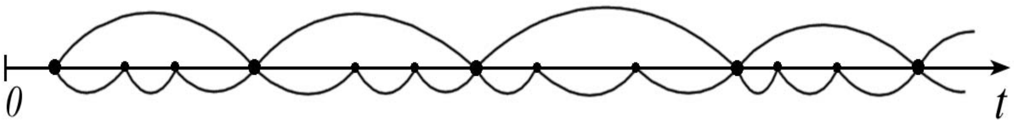


Рис. 1.6. Поток Эрланга 3-го порядка

Очевидно, что интервал времени T_k между любыми соседними событиями в потоке Э$_k$ – есть сумма k независимых случайных величин, а именно расстояний между соответствующими событиями простейшего потока. Эти независимые случайные величины распределены по показательному закону:

$$f_k(t) = \frac{\lambda(\lambda t)^{k-1}}{(k-1)!} e^{-\lambda t}, \quad t > 0.$$

При k = 1 получают $f_1(t) = \lambda e^{-\lambda t}$, $t > 0$, т. е. простейший поток является Э$_1$. Числовые характеристики величины T_k (математическое ожидание и дисперсия) следующие:

$$m_k = \frac{k}{\lambda} \qquad \text{и} \qquad \sigma_k^2 = \frac{k}{\lambda^2}.$$

Следует обратить внимание на то, что потоки Эрланга являются стационарными и ординарными, но обладают последействием, усиливающимся с увеличением k при $\lambda_k = \dfrac{\lambda}{k}$, где λ_k – интенсивность потока Э$_k$.

4.4. Планирование и организация компьютерного эксперимента*

Компьютерный эксперимент (КЭ) с моделью системы при её исследовании и проектировании проводится с целью получения информации о характеристиках процесса функционирования рассматриваемой системы. В нашем случае КЭ предполагает в большей степени наблюдение за поведением имитационной модели (см. Часть II данного учебного пособия). Причем каждому прогону модели соответствует определенная комбинация значений её параметров и длительность интервала времени, обеспечивающая набор необходимой статистики, при условии обеспечения заданной точности модели.

Целью планирования экспериментов является достижение минимума затрат на проведение всего комплекса экспериментов при ограничениях на объем и статистическое качество собираемой информации.**

Теория планирования экспериментов использует методы математической статистики и требует от исследователя обеспечения возможности управления экспериментом. При этом особенное значение приобретают лёгкость повторения условий КЭ с моделью системы и возможность управления экспериментом с моделью, включая его прерывание, возобновление и лёгкость варьирования воздействий внешней среды [1, 5, 16, 36, 41].

Преимуществами КЭ является возможность полного воспроизведения условий эксперимента с моделью исследуемой системы и простота прерывания и возобновления экспериментов. А прерывание эксперимента на время часто необходимо для анализа уже полученных результатов и принятия решений о его дальнейшем ходе.

---

* Этот параграф написан при авторском участии канд. техн. наук, доцента кафедры информатики СЗТУ, Петуховой Н.М.

** По логике этот параграф должен быть расположен после параграфа 2.4, но так как данный материал включён в пособие позже, то, чтобы не менять рубрикацию в Части I, он оказался здесь.

Планирование эксперимента

В основу методологии планирования экспериментов положена абстрактная схема, называемая «чёрным ящиком» (black box).

Различают входные переменные (факторы входного воздействия) x_1, x_2, …, x_n и выходные переменные (реакция, отклик) $y_1, y_2, …, y_k$. Каждый фактор x_i, для $i = 1, 2, …, x_n$ обычно принимает одно или несколько значений (уровней). Если исследуется влияние двух факторов с числом уровней каждого, равным тоже двум, то требуемое количество опытов равно четырём. Если же число факторов $n = 5$, а число уровней v каждого из факторов одинаково и $v = 3$, то количество экспериментов $N = v^n$, которые надо провести, получается равным уже $v^n = 3^5 = 243$. А при десяти факторах и пяти уровнях каждого, количество экспериментов уже приближается к 10 млн. Отсюда появилась необходимость выбора определённых сочетаний факторов входного воздействия и последовательности проведения экспериментов.

Анализ ситуаций, зависящих от различных, одновременно действующих факторов, выбор наиболее важных факторов и оценка их влияния освещаются в математической статистике. В зависимости от количества факторов, включённых в анализ, различают классификацию по одному признаку – *однофакторный анализ*, по нескольким признакам – *многофакторный анализ*.

Планирование эксперимента начинается с выбора объекта исследования, изучаемого с определённой целью ради отыскания определённых условий протекания процесса функционирования (см. главу 2 данного учебного пособия). Цель исследования определяется целевой функцией (параметром оптимизации, критерием оптимизации). Для прогнозирования значений целевой функции параметр оптимизации связывают с факторами входного воздействия некоторой функциональной зависимостью $Y = f(x_1, x_2, …, x_n)$. Эта зависимость называется функцией реакции, или моделью объекта исследования. А геометрический образ, соответствующий функции реакции, – поверхностью реакции.

Обычно вид зависимости f заранее неизвестен, поэтому используют приближённое соотношение $f(x_1, x_2, …, x_n) \approx \varphi(x_1, x_2, …, x_n)$. Функция φ находится по данным эксперимента. А эксперимент необходимо реализовать при минимальных затратах ресурсов, например при минимальном числе испытаний.

Объект исследования может иметь несколько выходных переменных (уровней реакции, отклика), которые обычно пытаются сократить до минимума. Если удаётся ограничиться одной выходной переменной, то вектор Y превращается в скалярную величину y. Если переменных состояний несколько, эксперимент проводится по каждой из них, а затем решается компромиссная задача.

При выборе факторов должны выполняться следующие требования – факторы должны быть регулируемыми и управляемыми. Регулируемость даёт возможность изменять значение фактора от некоторого минимального до максимального значения. Управляемость фактора – это возможность установки и поддержания выбранного значения фактора постоянным или изменяющимся в заданных границах. Другим требованием к выбору фактора является его непосредственное воздействие на величину выходных переменных, так как трудно управлять фактором, если он является функцией других факторов.

При использовании нескольких факторов они должны быть совместимы и независимы – то есть все их комбинации должны быть осуществимы и установление значения некоторого фактора независимо от значений других.

Для повышения возможности получения положительного результата от КЭ к выходным переменным предъявляют следующие требования: они должны иметь количественную характеристику, то есть измеряться; однозначно измерять эффективность объекта исследования (это требование эквивалентно корректной постановке задачи); должны быть статистически эффективными, то есть обладать возможно меньшей дисперсией. Реализация этих требований повышает шансы положительного завершения КЭ.

При проведении КЭ с моделью необходимо создать такие условия, которые способствуют выявлению влияния факторов, находящихся в функциональной связи с искомой характеристикой. Для этого подбирают факторы x_i, $i = 1, 2, \ldots, x_n$, влияющие на искомую характеристику, и описывают функциональную зависимость f или φ; устанавливают диапазон изменения факторов $x_{i\,\min} \div x_{i\,\max}$; определяют координаты точек факторного пространства $\{x_1, x_2, \ldots, x_n\}$, в которых будет проводиться моделирование.

Выделяют две составляющие планирования КЭ: стратегическое и тактическое планирования. При стратегическом планировании решается задача получения требуемой информации о системе с помощью компьютерного моделирования с учётом ограничений на вычислительные и временные ресурсы. При тактическом планировании реализуется совокупность методов уменьшения длительности компьютерного эксперимента при обеспечении статистической достоверности результатов. На длительность одного эксперимента влияет степень стационарности системы, взаимозависимости характеристик и значения начальных условий моделирования.

При тактическом планировании КЭ решают целый ряд проблем:

1. Определяют начальные условия и их влияние на достижение установившегося результата.

2. Определяют точность и достоверность результатов моделирования (см. § 5.6 данного учебного пособия).

3. Уменьшают дисперсию оценок характеристик процесса функционирования моделируемой системы.

4. Выбирают правила автоматической остановки КЭ.

Подводя итог обсуждению проблем планирования КЭ, необходимо отметить, что тактическое планирование реализуется непосредственно перед моделированием. И вообще, планирование является процессом итерационным, так как при уточнении некоторых свойств моделируемой системы этапы стратегического и тактического планирований могут повторяться.

Рассмотренные вопросы стратегического и тактического планирований тесно связаны с эффективной обработкой и представлением результатов КЭ.

Обработка результатов эксперимента

На реализацию методов обработки существенно влияют особенности КЭ, а именно – возможность получать в процессе моделирования большие выборки количественных характеристик функционирования системы. Но тут же появляются проблемы хранения промежуточных результатов моделирования, которые решаются путём использования рекуррентных алгоритмов обработки.

Другой особенностью является блочная конструкция компьютерной модели и возможность раздельного исследования блоков. Это часто требует реализации программной имитации входных переменных какого-то из частных блоков по оценкам выходных переменных другого (предшествующего) блока.

Часто при обработке результатов эксперимента возникает необходимость в подборе аналитического выражения для описания функциональной зависимости между величинами (переменными), полученными в результате эксперимента.

Пусть известно, что две случайные величины X и Y (X независимая, а Y зависимая переменные) связаны функциональной зависимостью $y = f(x)$. В результате КЭ получено n экспериментальных точек (x_1, y_1), (x_2, y_2), …, (x_n, y_n). Эта задача решается в два этапа: сначала выбор зависимости и затем уточнение коэффициентов этой зависимости.

Для определения вида функциональной зависимости $y = f(x)$ можно использовать теоретические соображения и опыт предыдущих аналогичных исследований.

Можно использовать графический метод, который предполагает построение корреляционного поля. Корреляционное поле – это точечный график в системе координат (x, y), каждая точка которого соответствует одному наблюдению (x_i, y_i). Затем осуществляется подбор эмпирической линии регрессии, связывающей переменные Y и X. К ним «подбирается» функциональная зависимость. После перебора нескольких функций для каждой из них строится уравнение регрессии и выбирается лучшее из них по показателям качества.

Например, в табл. 1.3 представлены некоторые эксперимен-тальные данные, а по ним построено корреляционное поле (рис. 1.7). Положение каждой точки на графике определяется величиной двух признаков: факторного x_i и результативного y_i. Рис. 1.8 … 1.10 иллюстрируют «подбор» средствами *Excel* функциональных зависимостей – линейной, экспоненциальной и полиномиальной.

Чем ближе точки корреляционного поля подходят к линии рег-рессии, тем лучше данное уравнение подходит к исходным данным. Из рис. 1.8 … 1.10 видно, что X и Y скорее всего связаны поли-номиальной зависимостью (полиномом второй степени – рис. 1.10).

Если уравнение регрессии проходит через все точки корреляционного поля, то оно полностью описывает исходные данные. Как правило, в практических исследованиях такого результата получить не удаётся. Обычно имеется некоторое рассеивание точек корреляционного поля относительно линии регрессии.

Таблица 1.3

x	y
0,8	2,6
1,6	1,4
2,1	0,9
2,5	1,3
3,3	2,3
3,8	4,2

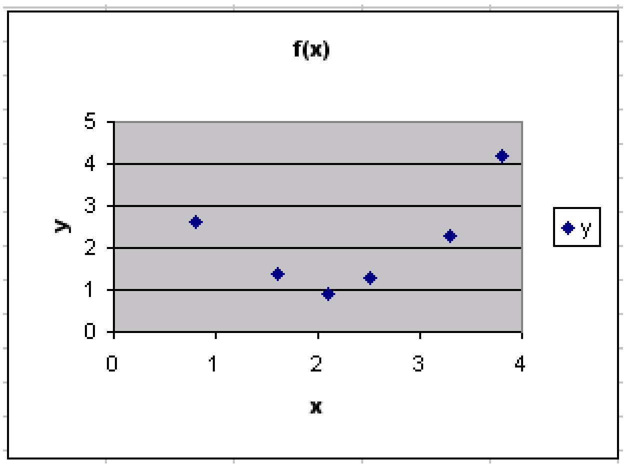


Рис. 1.7. Корреляционное поле

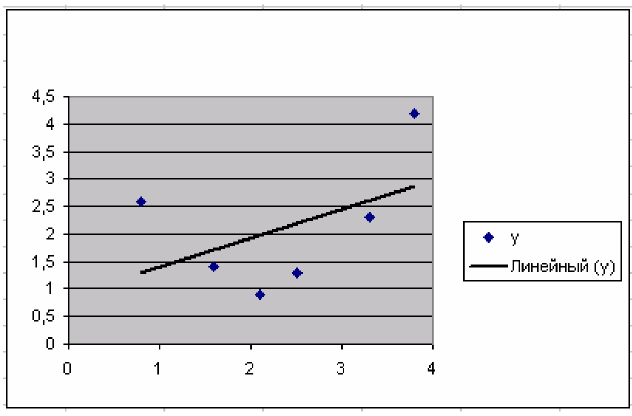


Рис.1.8. Линейная модель

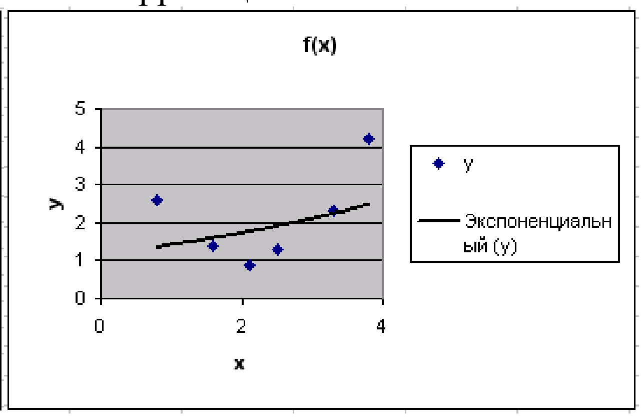


Рис.1.9. Экспоненциальная модель

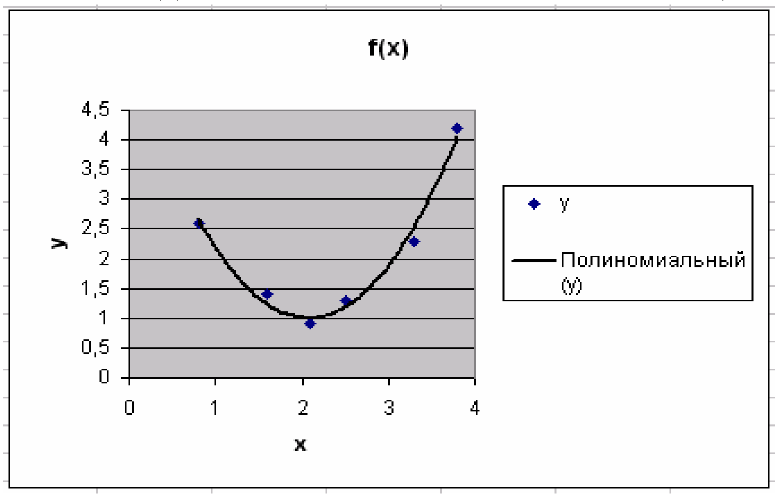


Рис. 1.10. Полиномиальная (параболическая) модель

Чаще всего зависимости между X и Y описываются следующими формулами: $y = a + bx$ (линейная зависимость), $y = a + b/x$ (гиперболическая), $y = a + bx + cx^2$ (параболическая), $y = ax^b$

(степенная), $y = ab^x$ (показательная), $y = a + bx + cx^2 + dx^3$ (полиномиальная) и др.

Определить вид функциональной зависимости можно и экспериментальным путём. Он заключается в построении уравнений регрессии разного вида для описания зависимости $Y = f(X)$ (например, линейной, экспоненциальной или полиномиальной) и выборе лучшей из них по показателям качества, например путём сравнения величины остаточной дисперсии $\sigma^2_{\text{ост}}$, рассчитанной для разных подобранных функций.

Чем меньше величина остаточной дисперсии, тем лучше уравнение регрессии подходит к исходным данным. Если $\sigma^2_{\text{ост}} = 0$, то подобранная линия регрессии полностью совпадает с истинной (но неизвестной) зависимостью $y = f(x)$, и уравнение регрессии проходит через все точки корреляционного поля. Если остаточная дисперсия оказывается примерно одинаковой для нескольких видов функций, то предпочтение отдаётся более простым видам функций, так как они в большей степени поддаются интерпретации и требуют меньшего объёма наблюдений.

Результаты исследований показывают, что число наблюдений должно в 6 – 7 раз превышать число рассчитываемых параметров при переменной x. Поэтому искать линию регрессии, если имеется число наблюдений меньше 7, бессмысленно. Если вид функции усложняется, требуется увеличение объёма наблюдений. Если выбирается парабола второй степени, требуется не менее 14 наблюдений.

Парная регрессионная модель

Парная регрессия – это уравнение связи между значениями Y и X ($y = f(x)$), где $Y = \{y_1, y_2, \ldots, y_n\}$ – результирующий признак, $X = \{x_1, x_2, \ldots, x_n\}$ – факторный признак. Данное уравнение называется «истинным» (фактическим) уравнением однофакторной регрессии. После определения вида функциональной зависимости $y = f(x)$ требуется оценить параметры модели. Для определения «наилучших» параметров модели, то есть модели, в которой точки корреляционного поля ближе всего подходят к линии регрессии, в большинстве случаев используют метод наименьших квадратов (МНК).

Метод наименьших квадратов позволяет получить такие оценки параметров модели, при которых сумма квадратов отклонений фактических значений результативного признака y (заданных таблично) от теоретических $\hat{y}$ (вычисленных с помощью уравнения регрессии) минимальна:

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \rightarrow \min.$$

То есть из всего множества линий линия регрессии выбирается так, чтобы сумма квадратов расстояний по вертикали между точками (x_i, y_i) и этой линией (то есть точкой $(x_i, \hat{y}_i)$) была минимальна (рис. 1.11).

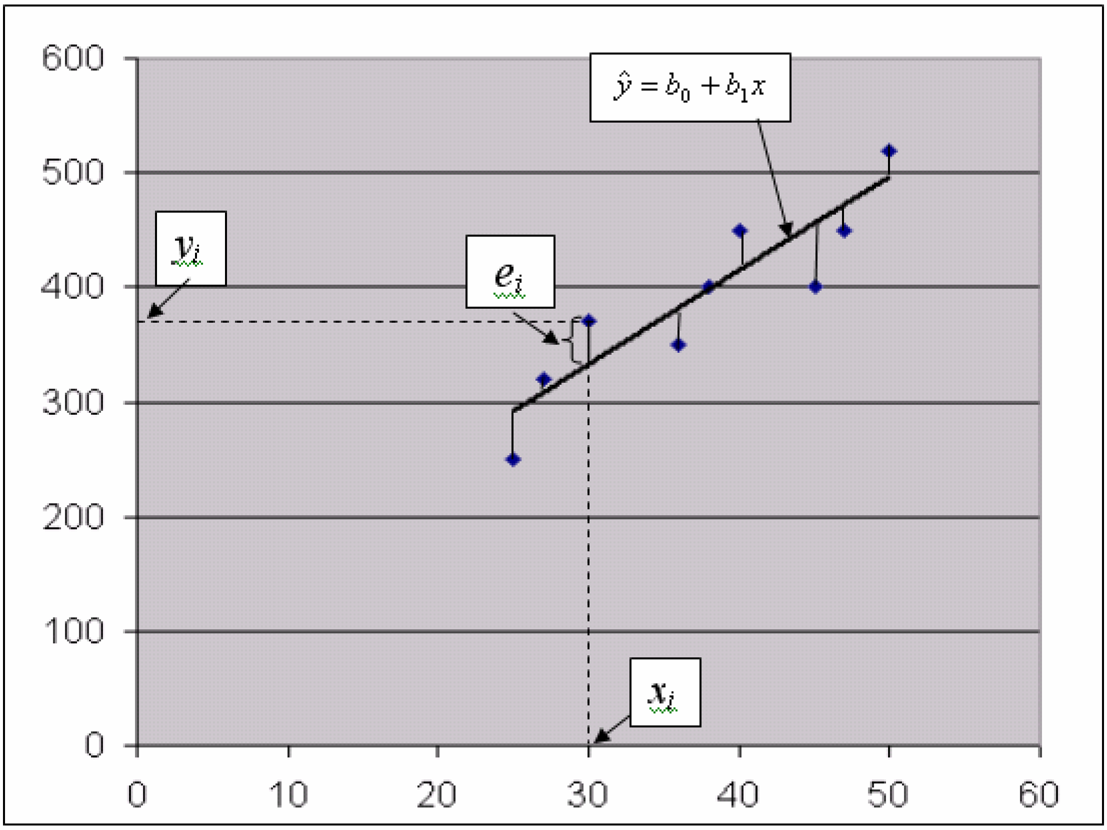


Рис. 1.11. Пример определения отклонения фактических значений y от теоретических $\hat{y}$

Отклонение фактического значения y_i от теоретического $\hat{y}_i$ обозначают через $e_i = y_i - \hat{y}_i$ и называют остатком:

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

Пусть предполагается, что связь между значениями X и Y является линейной, то есть «истинное» уравнение регрессии имеет вид $y = a_0 + a_1 x$, где a_0 и a_1 – неизвестные параметры модели. А теоретическое уравнение регрессии определяют в виде $\hat{y} = b_0 + b_1 x$, где b_0 и b_1 – оценки параметров a_0 и a_1 при использовании МНК. Тогда

$$F(b_0, b_1) = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2.$$

Известно, что необходимым условием существования экстремума функции является равенство нулю всех частных производных. Для рассматриваемой функции этот экстремум – минимум.

Для нахождения минимума функции двух переменных $F(b_0, b_1)$ необходимо вычислить частные производные по каждому из аргументов b_0 и b_1 и приравнять их к нулю: $dF/db_0 = 0$ и $dF/db_1 = 0$. Будет получена система двух уравнений:

$$\begin{cases} dF/db_0 = 2\sum_{i=1}^{n} [y_i - (b_0 + b_1 x_i)](-1) = 0 \\ dF/db_1 = 2\sum_{i=1}^{n} [y_i - (b_0 + b_1 x_i)](-x_i) = 0 \end{cases}$$

$$\begin{cases} \sum_{i=1}^{n}(-y_i + b_0 + b_1 x_i) = 0, \\ \sum_{i=1}^{n}(-x_i y_i + b_0 x_i + b_1 x_i^2) = 0. \end{cases} \quad \begin{cases} -\sum_{i=1}^{n} y_i + b_0 \sum_{i=1}^{n} 1 + b_1 \sum_{i=1}^{n} x_i = 0, \\ -\sum_{i=1}^{n} x_i y_i + b_0 \sum_{i=1}^{n} x_i + b_1 \sum_{i=1}^{n} x_i^2 = 0. \end{cases} \quad \begin{cases} b_0 n + b_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i, \\ b_0 \sum_{i=1}^{n} x_i + b_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i. \end{cases} \quad (1.7)$$

Решив систему уравнений (1.7), можно получить значения параметров b_0 и b_1:

$$b_1 = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}, \quad b_0 = \frac{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}, \quad b_0 = \frac{1}{n}\sum_{i=1}^{n} y_i - b_1 \frac{1}{n}\sum_{i=1}^{n} x_i.$$

Производятся некоторые преобразования. Числитель и знаменатель делится на n^2. Вводятся обозначения:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}, \qquad \overline{y} = \frac{\sum_{i=1}^{n} y_i}{n}, \qquad \overline{xy} = \frac{\sum_{i=1}^{n} x_i y_i}{n}.$$

Тогда $$b_1 = \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \left(\overline{x}\right)^2}, \qquad b_0 = \overline{y} - b_1 \overline{x}.$$

Систему уравнений (1.7) можно записать в матричной форме:

$$X_k \cdot B = Y_k, \tag{1.8}$$

где

$$X_k = \begin{vmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{vmatrix}, \qquad Y_k = \begin{vmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{vmatrix}, \qquad B = \begin{vmatrix} b_0 \\ b_1 \end{vmatrix}.$$

Элементы вектора B – искомые значения параметров b_0 и b_1. Для решения уравнения (1.8), представленного в матричной форме, следует выполнить преобразования:

$$X_k^{-1} \cdot (X_k \cdot B) = X_k^{-1} \cdot Y_k, \quad (X_k^{-1} \cdot X_k) \cdot B = X_k^{-1} \cdot Y_k, \quad E \cdot B = X_k^{-1} \cdot Y_k,$$

$$B = X_k^{-1} \cdot Y_k. \tag{1.9}$$

Тогда, решив приведённое уравнение, получают коэффициенты уравнения парной регрессии.

Если предположить, что связь между значениями Y и X является параболической, то есть $y = a_0 + a_1x + a_2x^2$, где a_0, a_1 и a_2 – параметры модели, то теоретическое уравнение регрессии можно представить в виде $\hat{y} = \boldsymbol{y = b_0 + b_1x + b_2x^2}$, где b_0, b_1 и b_2 – оценки параметров a_0, a_1 и a_2. Используя МНК, получают:

$$F(b_0,b_1,b_2) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - b_0 - b_1x_i - b_2x_i^2)^2 \to \min.$$

Условие существования экстремума примет следующий вид: $dF/db_0 = 0$, $dF/db_1 = 0$ и $dF/db_2 = 0$. В результате дифференцирования и приравнивания к нулю всех частных производных получают систему трёх линейных алгебраических уравнений:

$$\begin{cases} \dfrac{dF}{db_0} = 2\sum\limits_{i=1}^{n} (y_i - b_0 - b_1x_i - b_2x_i^2)(-1) = 0\,, \\ \dfrac{dF}{db_1} = 2\sum\limits_{i=1}^{n} (y_i - b_0 - b_1x_i - b_2x_i^2)(-x_i) = 0\,, \\ \dfrac{dF}{db_2} = 2\sum\limits_{i=1}^{n} (y_i - b_0 - b_1x_i - b_2x_i^2)(-x_i^2) = 0\,. \end{cases}$$

$$\begin{cases} b_0 n + b_1\sum\limits_{i=1}^{n} x_i + b_2\sum\limits_{i=1}^{n} x_i^2 = \sum\limits_{i=1}^{n} y_i\,, \\ b_0\sum\limits_{i=1}^{n} x_i + b_1\sum\limits_{i=1}^{n} x_i^2 + b_2\sum\limits_{i=1}^{n} x_i^3 = \sum\limits_{i=1}^{n} x_i y_i\,, \\ b_0\sum\limits_{i=1}^{n} x_i^2 + b_1\sum\limits_{i=1}^{n} x_i^3 + b_2\sum\limits_{i=1}^{n} x_i^4 = \sum\limits_{i=1}^{n} x_i^2 y_i\,, \end{cases}$$

или в матричной форме $X \cdot B = Y$, где

$$X=\begin{vmatrix} n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^4 \end{vmatrix}, \quad Y=\begin{vmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i \cdot x_i \\ \sum y_i \cdot x_i^2 \end{vmatrix}, \quad B=\begin{vmatrix} b_0 \\ b_1 \\ b_2 \end{vmatrix}.$$

Решив систему уравнений, получают элементы вектора B (1.9), то есть искомые значения параметров модели b_0, b_1 и b_2.

Множественная регрессия

Множественная регрессия представляет собой уравнение связи результативного признака Y с несколькими независимыми переменными – факторами (X_1, X_2,…,X_n). Если предположить, что между факторами существует линейная связь, то искомое уравнение регрессии имеет вид: $\hat{y} = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n$, где $b_0, b_1, b_2, \ldots, b_n$ – оценки неизвестных параметров $a_0, a_1, a_2, \ldots, a_n$ истинного уравнения регрессии. Для построения уравнения множественной регрессии используют и другие функции, приводимые к линейному виду.

Используя МНК, получают систему уравнений, решение которой позволяет получить оценки параметров регрессии. Если $y = f(x_1, x_2)$, то регрессионная модель является двухфакторной вида: $\hat{y} = b_0 + b_1x_1 + b_2x_2$. Применяя МНК, получают следующую систему уравнений:

$$\begin{cases} b_0 n + b_1\sum_{i=1}^{n} x_{1i} + b_2\sum_{i=1}^{n} x_{2i}^2 = \sum_{i=1}^{n} y_i \ , \\ b_0\sum_{i=1}^{n} x_{1i} + b_1\sum_{i=1}^{n} x_{2i}^2 + b_2\sum_{i=1}^{n} x_{1i}x_{2i} = \sum_{i=1}^{n} y_i x_{1i} \ , \\ b_0\sum_{i=1}^{n} x_{2i} + b_1\sum_{i=1}^{n} x_{1i}x_{2i} + b_2\sum_{i=1}^{n} x_{2i}^2 = \sum_{i=1}^{n} y_i x_{2i} \ , \end{cases}$$

или в матричной форме $X \cdot B = Y$, где

$$X = \begin{vmatrix} n & \sum_{i=1}^{n} x_{1i} & \sum_{i=1}^{n} x_{2i} \\ \sum_{i=1}^{n} x_{1i} & \sum_{i=1}^{n} x_{1i}^2 & \sum_{i=1}^{n} x_{2i} \cdot x_{1i} \\ \sum_{i=1}^{n} x_{2i} & \sum_{i=1}^{n} x_{1i} \cdot x_{2i} & \sum_{i=1}^{n} x_{2i}^2 \end{vmatrix}, \qquad Y = \begin{vmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i \cdot x_{1i} \\ \sum y_i \cdot x_{2i} \end{vmatrix}, \qquad B = \begin{vmatrix} b_0 \\ b_1 \\ b_2 \end{vmatrix}.$$

Решив систему уравнений, получают значения параметров модели b_0, b_1 и b_2.

Получив значения параметров модели, следует оценить её качество. Для этого используются показатели качества уравнения регрессии [5].

Проверка качества уравнения регрессии

Анализ качества уравнения регрессии – это проверка адекватности (соответствия) модели регрессии фактическим данным. Проверку адекватности можно производить на основе анализа остатков $e_i = y_i - \hat{y}_i$ $(i = \overline{1,n})$, то есть отклонений фактических значений y_i от значений, полученных расчётным путём, то есть вычисленных с использованием уравнения $\hat{y}_i$.

Если $e_i = 0$ $(i = \overline{1,n})$, то для всех наблюдений фактические значения совпадают с теоретическими (расчётными) – $y_i = \hat{y}_i$. Графически это означает, что теоретическая линия регрессии проходит через все точки корреляционного поля, а это возможно, если между Y и X имеется строгая функциональная зависимость (рис. 1.12).

На практике имеет место некоторое рассеивание точек корреляционного поля относительно теоретической линии регрессии (см. рис. 1.8 – 1.10), то есть – отклонение наблюдаемых значений от теоретических ($e_i \neq 0$) (рис. 1.13). Величина этих отклонений лежит в основе расчёта адекватности уравнений исходным данным.

При анализе качества регрессионной модели используется теорема о разложении дисперсии. Согласно этой теореме общая дисперсия результативного признака может быть разложена на две

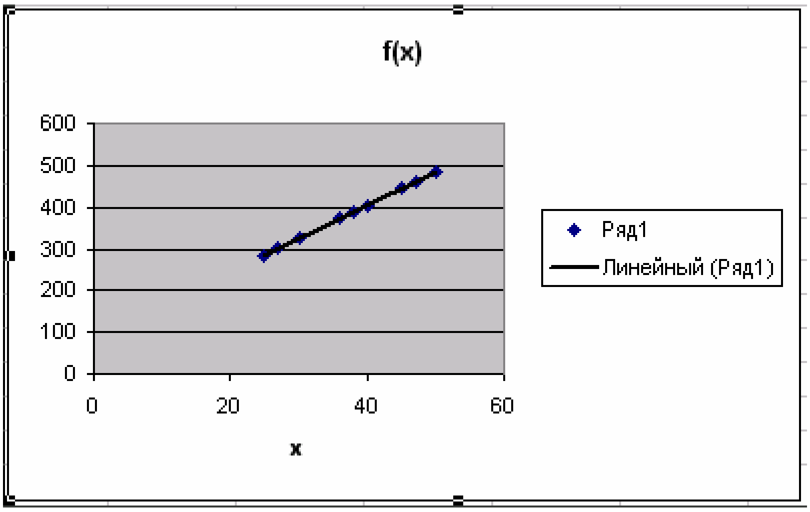


Рис. 1.12. Все фактические значения $\hat{y}$ совпадают с теоретическими ($e_i = 0,\ i = \overline{1, n}$)

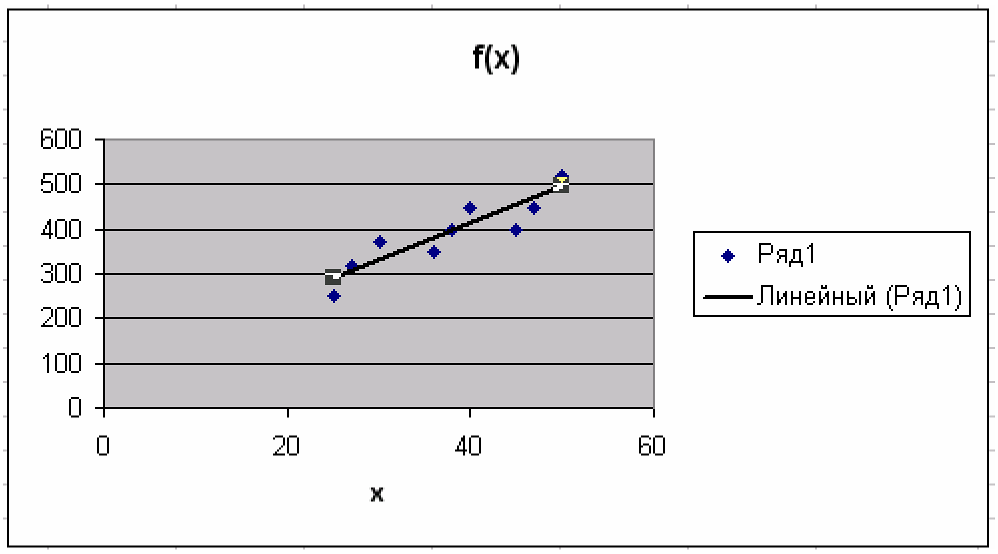


Рис. 1.13. Фактические значения $\hat{y}$ не совпадают с теоретическими ($e_i \neq 0,\ i = \overline{1, n}$)

составляющие – объяснённую уравнением регрессии (факторную) и не объяснённую уравнением регрессии (остаточную):

$$\sigma^2_{\text{îáù}} = \sigma^2_{\text{ôàêò}} + \sigma^2_{\text{îñò}} .$$

$\sigma^2_{\text{îáù}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y}_i)^2$ – общая дисперсия результативного признака.

$\sigma^2_{\text{ôàêò}} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - \overline{y}_i)^2$ – объяснённая уравнением регрессии, дисперсия результативного признака (факторная).

$\sigma^2_{\text{îñò}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ – не объяснённая уравнением регрессии, дисперсия результативного признака (остаточная).

На основании использования этой теоремы рассчитывают показатели качества модели.

1. **Стандартная ошибка *S*.** Она представляет собой среднее квадратическое отклонение наблюдаемых значений результативного признака от теоретических значений, рассчитанных по модели:

$$S = \sqrt{S^2},$$

где S^2 – несмещенная оценка дисперсии:

$$S^2 = \frac{1}{n-h}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n-h}\sum_{i=1}^{n}e_i^2, \qquad (1.10)$$

где $n - h$ – число степеней свободы (h – число параметров модели). Для парной регрессии $h = 2$, так как модель имеет два параметра b_0 и b_1.

Величину стандартной ошибки S сравнивают со среднеквадратическим отклонением результативного признака Y (σ_y). Если $S < \sigma_y$, то использование модели возможно.

2. Средняя ошибка аппроксимации

Это среднее отклонение расчётных значений от фактических:

$$\overline{A} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y}\right| = \frac{1}{n}\sum\left|\frac{e_i}{y_i}\right| \cdot 100\ \%.$$

Чем меньше рассеяние точек корреляционного поля вокруг теоретической линии регрессии, тем меньше средняя ошибка аппроксимации. Ошибка аппроксимации меньше 7 % свидетельствует о хорошем подборе модели к исходным данным. Допустимое предельное отклонение 10 %.

3. **Коэффициент линейной парной корреляции** r_{xy}

Это показатель тесноты линейной связи между результативным признаком и фактором:

$$r_{xy} = \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\sigma_x \sigma_y},$$

где
$$\overline{xy} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i, \qquad \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i,$$

σ_x и σ_y – среднеквадратические отклонения фактора x и результирующего признака y соответственно.

Область допустимых значений коэффициента линейной парной корреляции: $-1 \le r_{xy} \le 1$.

Если $b_1 > 0$, то $0 < r_{xy} \le 1$.

Если $b_1 < 0$, то $-1 \le r_{xy} < 0$.

Знак «+» коэффициента корреляции означает, что с увеличением ***x*** значения ***y*** увеличиваются ($b_1 > 0$); знак «–» означает, что с ростом ***x*** значения ***y*** уменьшаются ($b_1 < 0$).

При $r_{xy} = \pm 1$ корреляционная связь представляет полную линейную функциональную зависимость (все точки корреляционного поля располагаются на одной линии регрессии) (рис. 1.14, 1.15).

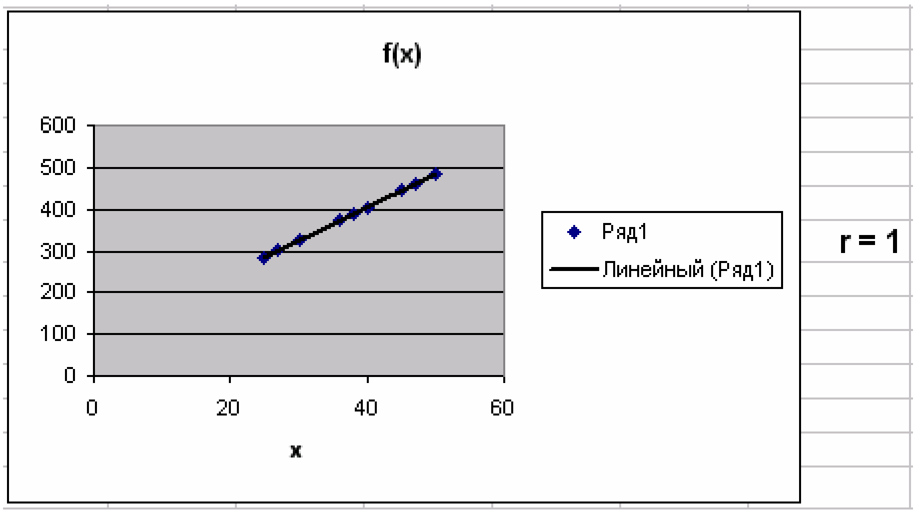


Рис. 1.14. $b_1 > 0$, $r_{xy} = 1$.

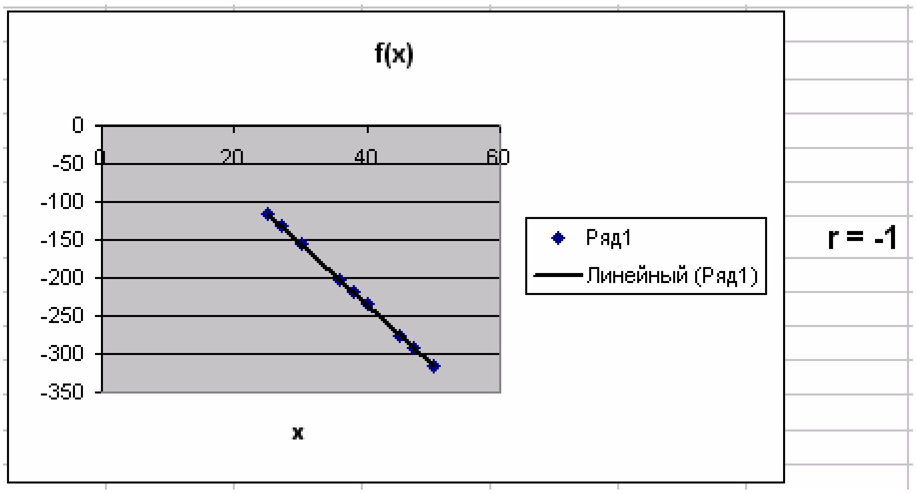


Рис. 1.15. $b_1 < 0$, $r_{xy} = -1$

При $r_{xy} = 0$ линейная корреляционная связь отсутствует. Близость $|r_{xy}|$ к нулю не означает отсутствие связи между x и y. При выборе регрессионной модели другого вида (например, параболической, степенной, …) связь между ними может оказаться достаточно тесной. Чем ближе $|r_{xy}|$ к единице, тем сильнее линейная связь x и y.

Считают, что при $r_{xy} \le 0{,}3$ наблюдается слабая линейная связь; при $r_{xy} = 0{,}3 \div 0{,}7$ – средняя линейная связь;

$r_{xy} = 0{,}7 \div 0{,}9$ – сильная линейная связь;

$r_{xy} > 0{,}9$ – весьма сильная линейная связь;

$r_{xy} = 1$ – полная линейная связь.

Для парной линейной регрессии коэффициент корреляции может быть рассчитан по формуле

$$r_{xy} = b_1 \frac{\sigma_x}{\sigma_y}.$$

Для нелинейной регрессии тесноту связи оценивает индекс корреляции

$$\rho_{xy} = \sqrt{1 - \frac{\sigma^2_{îñò}}{\sigma^2_y}}.$$

4. **Коэффициент (индекс) детерминации**

Коэффициент детерминации R^2 – мера качества подгонки регрессионной модели к исходным данным. R^2 характеризует долю дисперсии результативного признака *Y*, объяснённую регрессией в общей дисперсии результативного признака. ($1 - R^2$) – характеризует долю дисперсии результативного признака *Y*, необъяснённую уравнением регрессии, а, следовательно, вызванную влиянием неучтённых факторов. Чем больше доля объяснённой дисперсии, тем лучше модель аппроксимирует исходные данные.

Для вычисления R^2 можно пользоваться формулами:

$$R^2 = \frac{\sigma^2_{îáÿñí}}{\sigma^2_{îáù}} = 1 - \frac{\sigma^2_{îñò}}{\sigma^2_{îáù}};$$

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}.$$

R^2 – не убывающая функция от числа факторов. Включение в модель любого дополнительного фактора не приведет к снижению коэффициента детерминации. Поэтому не всегда ясно, за счет чего возрос R^2 – за счет реального влияния на результат дополнительно включенного фактора или за счет увеличения числа факторов.

Чтобы R^2 были сравнимы по разным моделям, необходимо учесть число факторов в модели. Для этого используется $R^2_{\text{скор}}$ (нормированный R^2), который определяется через дисперсии, вычисленные на одну степень свободы.

Существует равенство между числом степеней свободы общей дисперсии и сумм факторной и остаточной дисперсий.

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - \bar{y}_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

$$\sigma^2_{îáù} \quad = \quad \sigma^2_{ôàêò} \quad + \quad \sigma^2_{îñò}$$

$$(n-1) \quad = \quad 1 \quad + \quad (n-2).$$

Определение дисперсии на одну степень свободы приводит дисперсии к сравнимому виду.

$$R^2_{\text{скор}} = 1 - \frac{\sum_{i=1}^{n} e_i^2 / (n-h)}{\sum_{i=1}^{n} (y_i - \overline{y}_i)^2 / (n-1)},$$

или

$$R^2_{\text{скор}} = 1 - (1 - R^2) \cdot \frac{n-1}{n-h}.$$

Чем больше $R^2_{\text{скор}}$, тем лучше качество модели по сравнению с другими вариантами.

Корень квадратный из коэффициента детерминации есть коэффициент множественной корреляции:

$$R_{y(x)} = \sqrt{R^2}.$$

Для парной линейной регрессии $R_{y(x)} = r_{xy}$.

Если все точки корреляционного поля лежат на теоретической линии регрессии, то $R_{y(x)} = 1$; следовательно, связь между y и x функциональная, и уравнение регрессии очень хорошо описывает фактические данные. Если $R_{y(x)} = 0$, то уравнение регрессии плохо описывает исходные данные.

Примечание, относящееся к многофакторному анализу.

Использование многофакторных моделей приводит к возникновению новых проблем, связанных с необходимостью оценки не только влияния каждой независимой переменной на зависимую, но и разграничением их воздействия на результат. Следует определить, какие независимые переменные нужно включить в уравнение регрессии, а какие исключить.

Факторы не должны быть коррелированны между собой. Если же факторы имеют высокую корреляцию между собой, это может привести к ненадёжности коэффициентов регрессии. Поэтому отбор факторов происходит в два этапа. Сначала подбирают факторы из сущности задачи, а затем на основе показателей корреляции между объясняющими переменными исключают из модели дублирующие факторы. Считают, что два фактора явно коллинеарны между собой, если коэффициент корреляции меду ними $r_{x_i y_i} \geq 0.7.$ Оставляют в модели тот фактор, который при достаточно тесной связи с результирующим признаком имеет наименьшую тесноту связи с другими факторами.

Факторы, включаемые в модель, должны быть количественно измеримы. Если же в модель необходимо включить качественный фактор, не имеющий количественного измерения, ему надо присвоить количественную определённость.

В математической статистике разработан аппарат проверки статистической значимости (надёжности) уравнения регрессии в целом, статистической значимости коэффициентов регрессии и корреляции, оценки целесообразности включения дополнительного фактора в уравнение регрессии, расчётов доверительных интервалов для коэффициентов регрессии и прочее [5]. Но перечисленное не входит в данный курс из-за ограниченности объёма.

Выводы по первой части учебного пособия

Предметом рассмотрения данной дисциплины является моделирование систем. Этим вопросам посвящён весь материал этого учебного пособия.

Система – это совокупность объектов, функционирующих и взаимодействующих друг с другом для достижения определённой цели. На практике понятие системы зависит от задач конкретного исследования [18].

Состояние системы определяется как совокупность переменных, необходимых для описания системы на определённый момент времени в соответствии с задачами исследования. Обычно исследователь решает следующие вопросы (рис. 1.16):

– Осуществлять эксперимент с реальной системой или с моделью системы?

– Исследовать физическую модель или создавать математическую модель?

– Производить аналитическое решение или имитационное моделирование? И ещё многие вопросы.

Подводя итог рассмотрению общих вопросов системного моделирования, можно прийти к выводу о том, что требуется приложить много усилий, направленных на разработку и анализ сложной системы посредством моделирования. На рис. 1.17 показаны обобщённые типичные этапы последовательности исследования системы посредством моделирования.

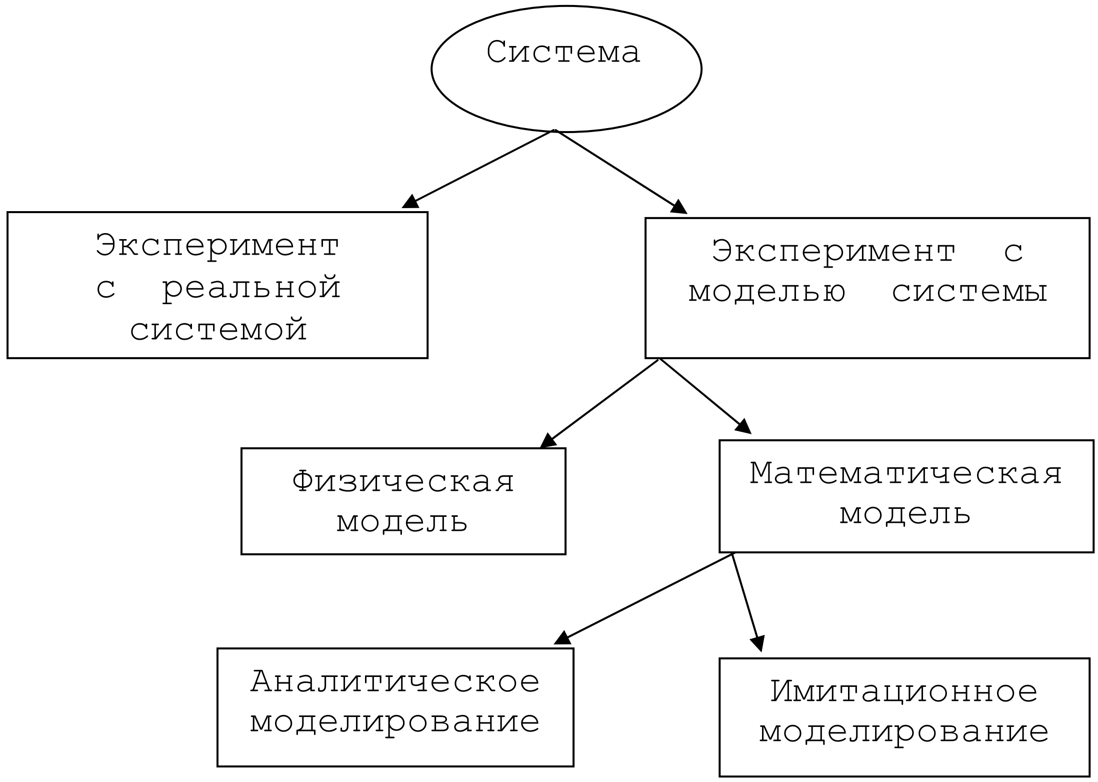


Рис. 1.16. Общее представление о способах исследования систем

По сравнению с перечнем этапов моделирования, приведённых на с.17, эта схема несколько отличается последовательностью и степенью детализации. Этапы 4 – 10 (рис. 1.17) в различной степени детализации будут освещены в Частях II и III.

Главный момент, о котором в процессе моделирования необходимо всё время помнить, – это проблема адекватности модели, метода, подхода. На рис. 1.18 приведена схема коррелированной проверки созданной модели.

Популярности моделирования способствуют следующие его преимущества: моделирование позволяет оценить эксплуатационные показатели существующей системы при некоторых проектных условиях эксплуатации или сравнить предлагаемые альтернативные варианты проектов системы, или обеспечить более эффективный контроль условий эксплуатации, нежели при экспериментировании с самой системой.

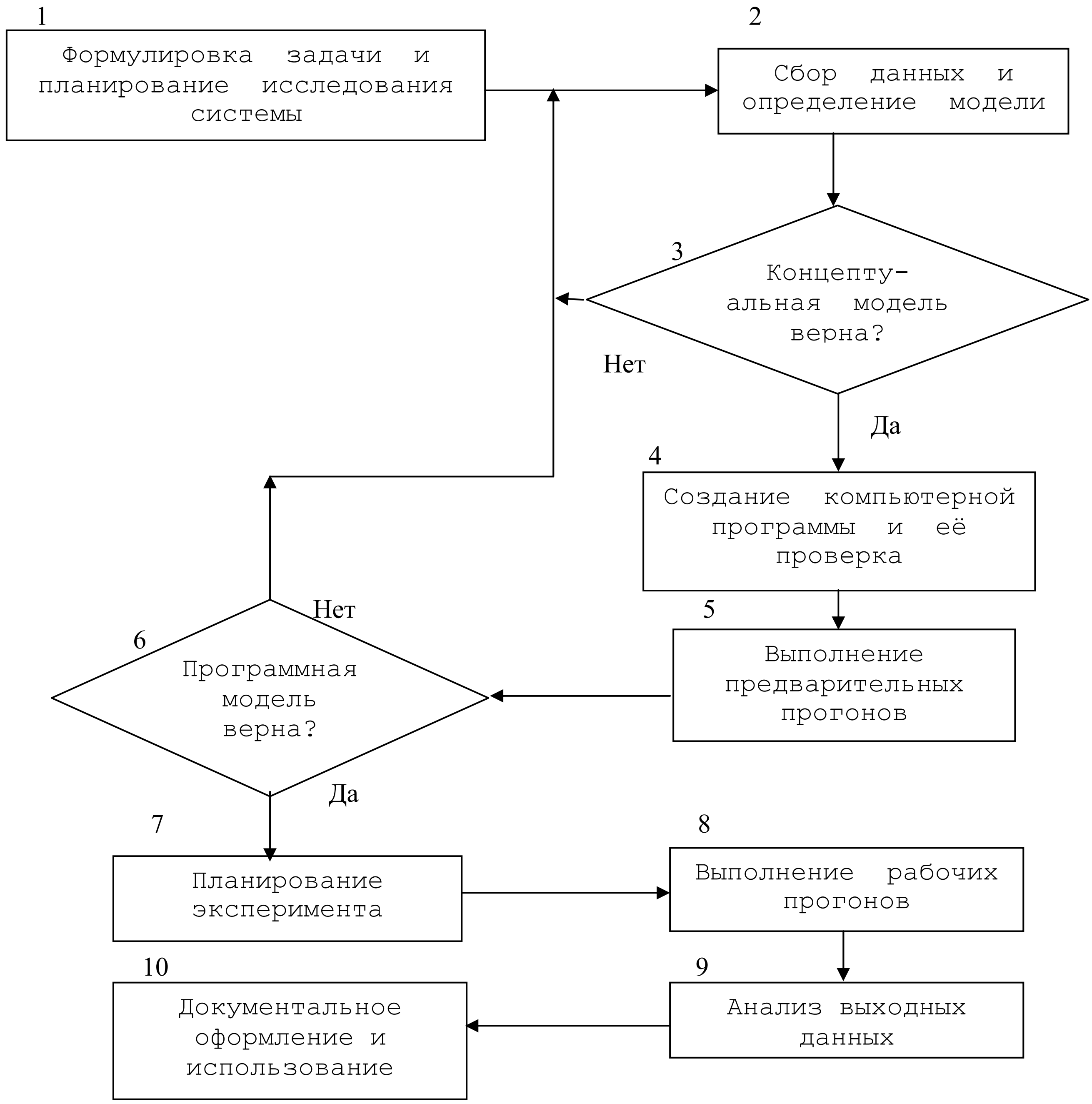


Рис. 1.17. Этапы моделирования

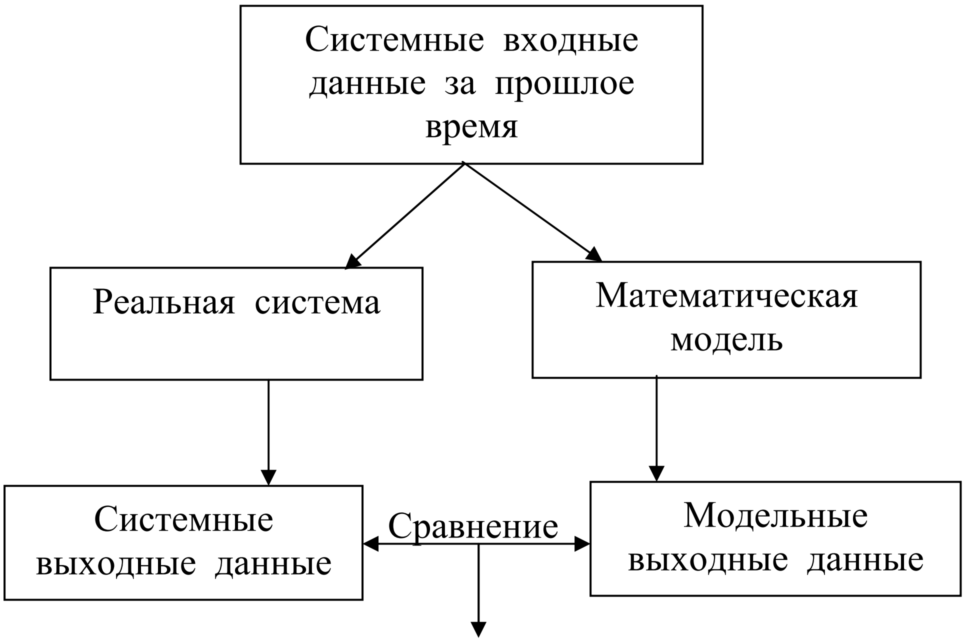
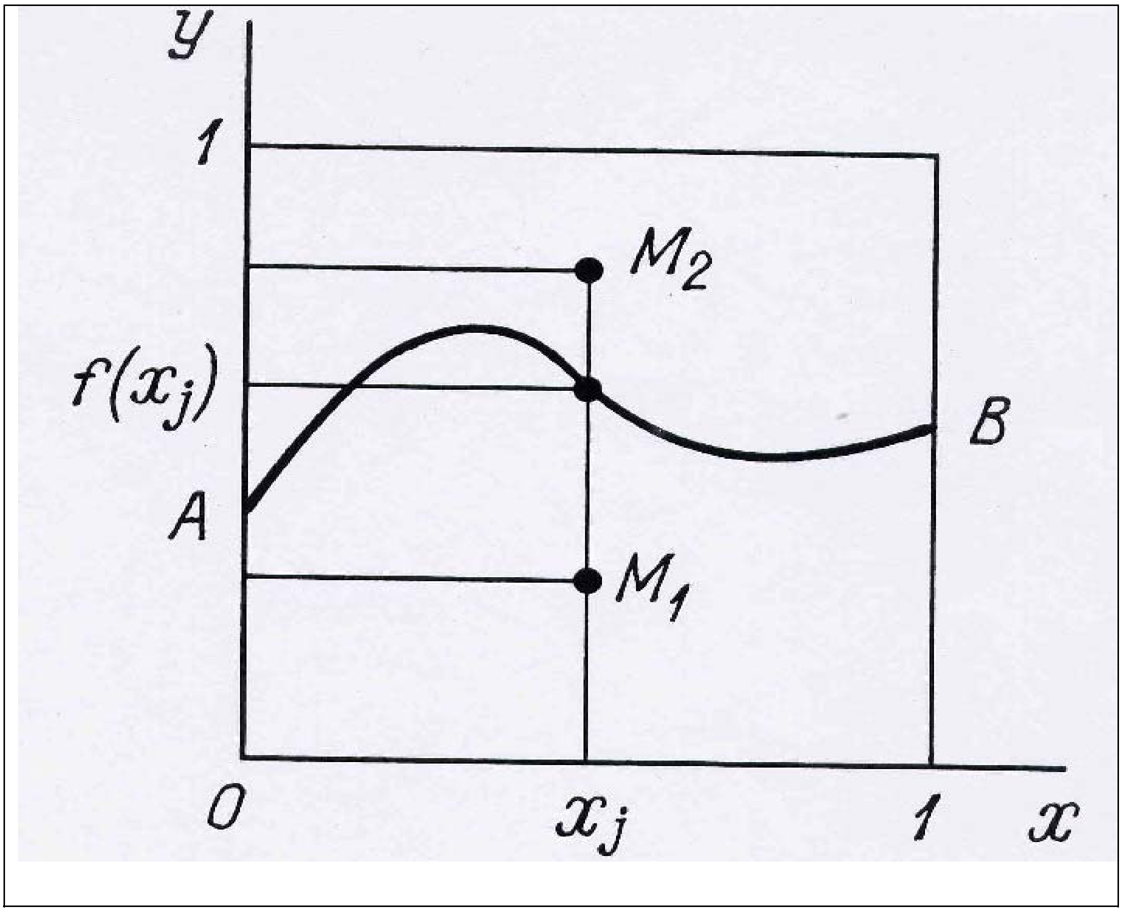


Рис. 1.18. Схема коррелированной проверки созданной модели

Что может помешать проведению моделирования? Это следующие немаловажные факторы: нечёткая постановка задачи, недостаточный уровень проработки модели, подход к исследованию системы посредством моделирования как к простому упражнению в программировании, неподходящее программное обеспечение, использование неправильных критериев оценки работы.

Часть II

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ



Глава 5. Теоретические основы метода имитационного моделирования

Глава 6. Моделирование СМО методом имитационного моделирования

Глава 7. Моделирование на GPSS

Имитационное моделирование – один из наиболее распространённых методов, а возможно, и самый распространённый метод исследования операций и теории управления. Предположим, что имеется имитационная модель, которую требуется исследовать. Сначала должно быть принято решение о средствах её исследования, то есть необходимо классифицировать имитационные модели по нескольким признакам:

– *Статическая или динамическая?* Статическая – это система с зафиксированным временем, или система, в которой время не имеет значения. Динамическая – это система, изменяющаяся во времени.

– *Детерминированная или стохастическая?* Модель, не содержащая вероятностных (случайных) компонентов, называется детерминированной. И наоборот, стохастическая имитационная модель характеризуется случайными входными данными.

– *Непрерывная или дискретная?* В данном случае разговор идёт о состояниях системы. В первой из них переменные состояния меняются непрерывно во времени, а в дискретной системе переход из состояния в состояние происходит мгновенно.

Имитационные модели, методика построения которых рассматривается ниже, будут дискретными, динамическими и стохастическими. Такие модели называются дискретно-событийными имитационными моделями.

Глава 5
ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МЕТОДА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Прежде чем описывать методики построения и функционирования имитационных моделей СМО необходимо ознакомиться с теоретическими основами метода. Это реализуется в настоящей главе 5. Здесь излагается сущность метода, методология построения генераторов случайных величин и проверки их качества, подходы к получению последовательностей случайных величин, распределенных по заданным законам, методики оценки точности характеристик, получаемых при использовании метода имитационного моделирования, фиксации и обработки результатов моделирования [7, 12, 13, 17, 18, 20, 26, 33, 39, 40].

5.1. Сущность метода и области его применения

Зарождение метода связано с работой фон Неймана и Улама, в 1940-е годы применивших его к решению задач экранирования ядерных излучений. Основной принцип метода имитационного моделирования (его часто называют методом Монте-Карло, или методом статистического моделирования) состоит в построении искусственного вероятностного процесса, параметры которого давали бы решение поставленной задачи, причем сама задача может и не быть вероятностной. При этом можно сформулировать две задачи: прямую и обратную.

Прямая задача. Необходимо реализовать операцию, исход которой определяется случайными факторами. Поэтому эта операция представляет собой случайный процесс. Путем проведения серии экспериментов с последующей статистической обработкой результатов отдельных наблюдений можно найти приближенное значение вероятностных характеристик этого процесса. Для этого осуществляется многократное проведение этой операции при фиксированных условиях. Каждое наблюдение представляет собой реализацию случайного процесса, а каждый результат - случайную величину, подчиненную тому закону распределения, который и отыскивается при проведении экспериментов.

Обратная задача. Известны вероятностные характеристики процесса и необходимо воспроизвести одну из его реализаций. Процесс этого воспроизведения называют статистическим испытанием. При многократном испытании и последующей обработке можно найти статистические оценки исходных вероятностных характеристик.

Пусть известно, что вероятность совершения какого-то события равна 0.6 (например, это вероятность безотказной работы вновь установленного оборудования без соответствующей настройки). Элементарной статистической моделью процесса работы оборудования может являться обычная традиционная урна с 10 шарами (6 белых и 4 черных). Реализация одного испытания состоит в извлечении шара, последующем его возвращении в урну и перемешивании шаров. При большом числе испытаний статистическая оценка вероятности равна 0,6, т.е. совпадает с вероятностной характеристикой процесса. Но, по-видимому, не целесообразно строить модель с целью получения заранее известного результата. Но так как сложный процесс может складываться из комплекса простых процессов и при этом связь между ними может быть неизвестной, кроме их вероятностных характеристик, то оказывается возможным нахождение с помощью имитационного моделирования вероятностных характеристик всего процесса.

Пусть, например, строятся модели функционирования систем массового обслуживания. Известно, что сложность построения вероятностных моделей, в которых необходимо учитывать влияние ряда случайных факторов, резко возрастает с увеличением их числа. Поэтому построение аналитических моделей систем или процессов, основанных на математическом аппарате теории вероятностей, удается только при значительных упрощениях или даже при отказе от тех или иных факторов. Аналитические методы теории массового обслуживания, ограниченные рамками марковских случайных процессов, могут быть использованы только для исследования СМО простых конструкций (см. Часть III данного учебного пособия).

Идея метода имитационного моделирования проста и состоит в следующем. Вместо описания случайного явления аналитическими зависимостями производится моделирование (розыгрыш) этого случайного явления с помощью некоторой процедуры, дающей случайный результат. В результате получается одна реализация случайного явления. После проведения большого количества таких реализаций этого явления производится обработка результатов моделирования методами математической статистики. Этими розыгрышами можно решать любые вероятностные задачи, но оправданными (выгодными) они становятся тогда, когда процедура моделирования проще применения аналитических, вычислительных методов.

И еще раз, пусть исследуется многоканальная СМО, но процесс, протекающий в ней, не марковский [8, 9, 26, 33]. Закон распреде-ления интервалов между моментами появления заявок не показательный, а произвольный. Длительность времени обслуживания заявок тоже не показательная, а имеет какое-то другое распределение. Кроме того, каналы могут иногда выходить из строя, а время безотказной работы имеет произвольное распределение. Требуется найти вероятности состояний СМО и показатели ее эффективности. Создание аналитической модели такой СМО может быть затруднительным. Поэтому нужно (легче) построить имитационную модель или использовать уже готовую.

Имитационное моделирование базируется на использовании генераторов случайных величин в сочетании с интегральной функцией распределения вероятностей для исследуемого процесса. Генератором может быть просто колесо рулетки, таблица, программа компьютера или другой известный источник равномерно распределенных случайных чисел (см. § 2.2). Разыгрываемое распределение вероятностей основано на эмпирических данных или является известным теоретическим законом.

В основе общей схемы метода имитационного моделирования лежит центральная предельная теорема теории вероятностей.

Теорема 2.1. Если $X_1, X_2, ..., X_N$ – независимые случайные величины, имеющие один и тот же закон распределения с математическим ожиданием m_x и дисперсией σ_x^2, то при неограниченном увеличении N закон распределения суммы $\sum_{k=1}^{N} X_k$ неограниченно приближается к нормальному.

В силу теоремы 2.1 имеет место соотношение

$$P\left\{\left|\frac{1}{N}\sum_{k=1}^{N} X_k - m_x\right| < \frac{3\sigma_x}{\sqrt{N}}\right\} \approx 0{,}997,$$

где $X_k,\ k = 1,2,...,N -$ значения случайной величины X, получаемые в каждом из N испытаний. Это соотношение определяет неизвестное число m_x и одновременно дает оценку погрешности (о погрешности подробнее см. в § 5.6).

При таком подходе общая схема процедуры рассматриваемого метода выглядит очень просто:

1. На основе эмпирического ряда чисел построить график или таблицу интегральной функции распределения. Значения случайной переменной откладываются по оси абсцисс, а значения вероятностей (от 0 до 1) – по оси ординат.
2. Получить случайное число (СЧ), равномерно распределённое в интервале [0,1].
3. Провести горизонтальную прямую от точки на оси ординат, соответствующей выбранному случайному числу, до пересечения с кривой распределения вероятностей. Из этой точки опустить перпендикуляр на ось абсцисс.
4. Принять полученное значение x за выборочное значение.
5. Повторить пп. 2 – 4 требуемое количество раз.

Пример 2.1. Пусть имеется система, в которой за каждый десятиминутный период число заявок, поступающих на обслуживание, подчиняется распределению из табл. 2.1.

Таблица 2.1

Число заявок	Вероятность	Интегральная вероятность
0	0.40	0.40
1	0.25	0.65
2	0.20	0.85
3	0.15	1.00

Искусственный имитационный эксперимент проводится для шести периодов времени. Построенный график функции распределения вероятностей представлен на рис. 2.1.

Из любой таблицы случайных величин выбирается шесть чисел. Тогда в соответствии с процедурой имитационного моделирования получают результаты, представленные в табл. 2.2.

Если генерируемые числа распределены равномерно, то значение искомой величины в процессе имитации будет появляться с такой же относительной частотой, как и в реальной системе.

Таблица 2.2

Период времени	Случайное число	Число заявок
1	0.35	0
2	0.13	0
3	0.28	0
4	0.83	2
5	0.56	1
6	0.75	2

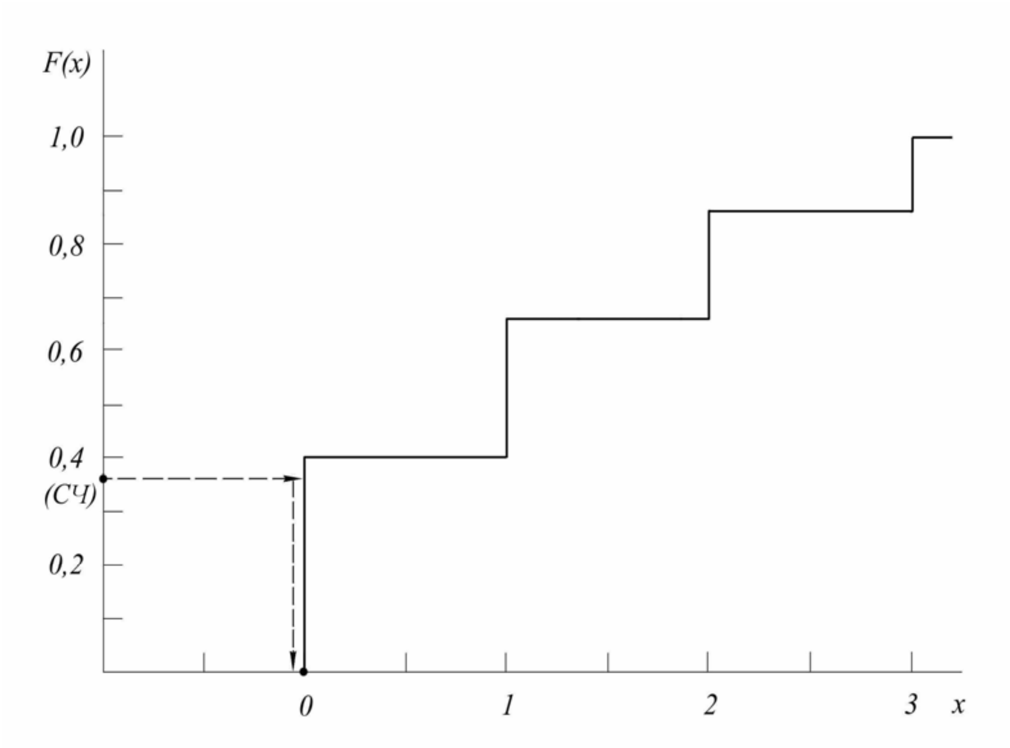


Рис. 2.1. Эмпирическая функция распределения вероятностей

Пример 2.2. Рассматривается классическая задача о случайном блуждании. Пусть прохожий, находясь на углу улицы, решает совершить прогулку, плохо зная город. Вероятности того, что, достигнув очередного перекрестка, он пойдет на север, юг, запад или восток – одинаковы. Какова вероятность того, что, пройдя 10 кварталов, прохожий окажется не далее двух кварталов от места начала прогулки?

Координаты прохожего на перекрестке обозначаются двумерным вектором (x, y), где x – направление «восток – запад» и y – направление «север – юг». Перемещение на один квартал к востоку дает приращение $x = x + 1$, а к западу – $x = x - 1$. По аналогии, передвижению на квартал к северу соответствует приращение $y = y + 1$, а к югу – $y = y - 1$. Исходное положение обозначают (0,0).

Таким образом, если в конце прогулки протяженностью 10 кварталов окажется, что $|x| + |y| > 2$, то прохожий ушел от точки (0,0) больше, чем на два квартала.

Поскольку на перекрестке (включая исходную точку) вероятность дальнейшего перемещения в любом направлении одинакова, то она равна 0.25. Необходимо закрепить направления перемещения за интервалами значений равномерно распределенных случайных чисел: на восток – (0.00 – 0.24), на запад – (0.25 – 0.49), на север – (0.50 – 0.74) и на юг – (0.75 – 0.99). Алгоритм имитационного моделирования задачи приведен на рис. 2.2, а результаты даны в табл. 2.3.

Для получения достоверных оценок проведенного числа опытов явно недостаточно. Для проведения испытания в требуемом объеме можно описываемый алгоритм запрограммировать и реализовать на компьютере.

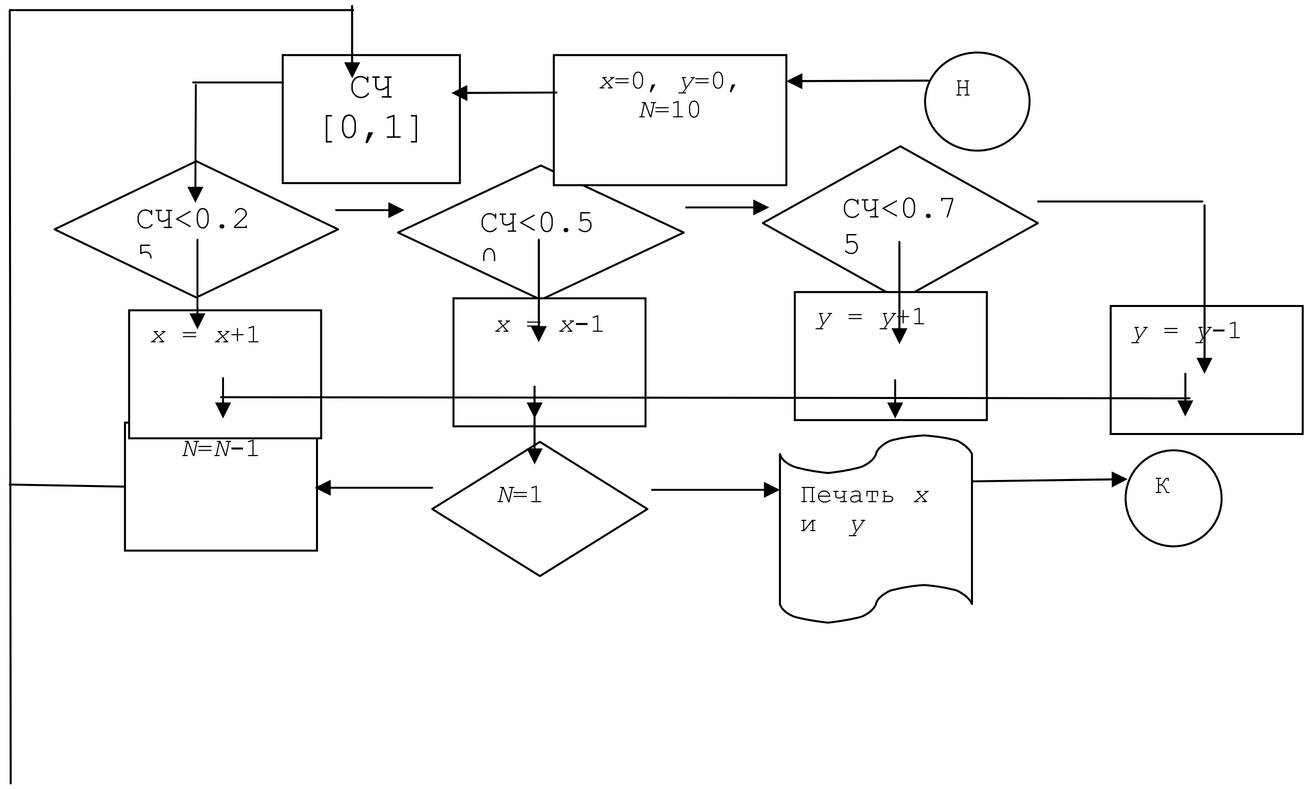


Рис. 2.2. Алгоритм решения задачи о случайном блуждании

Итак, краткая последовательность этапов имитационного моделирования:

– разрабатывается и реализуется на ЭВМ детерминированная математическая модель системы, отображающая связь значений выходных параметров системы со значениями входных воздействий, начальных условий и структуры системы;

– обеспечивается получение на ЭВМ отдельных реализаций случайных событий, величин, функций, т. е. моделируется случайное явление с заданными характеристиками, соответствующими характеристикам случайных явлений, сопровождающих функционирование реальной исследуемой системы;

– производится многократная реализация детерминированного процесса, где в каждой из реализаций учитывается влияние случайных явлений;

– производится статистическая обработка полученных результатов в соответствии с характером имитируемого процесса.

Таблица 2.3

Пройдено Кварталов	Опыт 1		Опыт 2		Опыт 3	
	Случайное число	Местоположение	Случайное число	Местоположение	Случайное число	Местоположение
1	0.73	0,1	0.10	1,0	0.06	1,0
2	0.21	1,1	0.89	1,-1	0.95	1,-1
3	0.45	0,1	0.14	2,-1	0.04	2,-1
4	0.76	0,0	0.81	2,-2	0.67	2,0
5	0.96	0,-1	0.30	1,-2	0.51	2,1
6	0.94	0,-2	0.91	1,-3	0.95	2,0
7	0.53	0,-1	0.06	2,-3	0.73	2,1
8	0.57	0,0	0.38	1,-3	0.10	3,1
9	0.96	0,-1	0.79	1,-4	0.76	3,0
10	0.43	-1,-1	0.43	0,-4	0.30	2,0
Успешно?	$\|x\|+\|y\|=2$ → Да		$\|x\|+\|y\|=4$ → Нет		$\|x\|+\|y\|=2$ → Да	

Практическая реализация метода имитационного моделирования предполагает последовательное (в некоторых случаях совместное) выполнение следующих этапов: постановка задачи исследования, разработка методики моделирования и обработки результатов эксперимента, построение модели, представляемой в виде блок-схемы моделирующего алгоритма, разработка программного обеспечения, которое может быть выполнено либо на универсальном алгоритмическом языке, либо на специализированном языке моделирования (например GPSS [26, 33, 37, 42, 44]), реализация имитационного эксперимента, анализ и обобщение результатов моделирования.

Широкое использование имитационного моделирования, получившего большое распространение как в нашей стране, так и за рубежом, объясняется рядом преимуществ, присущих этому методу. Во-первых, метод позволяет получить ответы на многие вопросы, возникающие на этапах эскизного проектирования системы, без применения дорогостоящего метода проб и ошибок. Во-вторых, имитация на ЭВМ позволяет исследовать особенности функционирования системы в любых возможных условиях. При этом параметры системы могут изменяться в любых заданных пределах. Поэтому уменьшается потребность в эксплуатационных испытаниях системы. Наконец, применение ЭВМ сокращает продолжительность испытаний системы от месяцев до долей минут и секунд, а также сокращает сроки получения и обработки экспериментальных данных о характере поведения исследуемой системы.

На популярном примере вычисления определенного интеграла

$$I = \int_0^1 f(x)dx$$

можно проследить реализацию основных этапов имитационного моделирования. При этом для простоты принимается $0 \le f(x) \le 1$. Вычисление интеграла (рис. 2.3) сводится к определению площади криволинейной трапеции 0*AB*1.

Соответственно можно сформулировать следующую задачу: определить вероятность p того, что случайная точка с координатами (x_j, y_j) окажется в области 0*AB1*, т.е. ниже кривой $f(x)$. Причем числа x_j и y_j являются случайными величинами, равномерно распределенными в интервале [0,1].

Поставленная задача реализуется путем формирования координат *N* случайных точек в единичном квадрате и определения количества *m* тех точек, которые попали под кривую или на саму кривую. В этом случае искомый интеграл $I = p \approx \frac{m}{N}$. Если *N* достаточно велико (методика определения количества реализаций процесса моделирования изложена в § 5.6), то полученное значение искомого интеграла становится приемлемым.

Координаты точек можно сформировать путем использования последовательности случайных чисел k_i, равномерно распределенных в интервале [0, 1]. Эти числа получаются с помощью генератора случайных величин. Принципы построения этих генераторов подробно изложены в разделе 4.2 учебника [41], а методики проверки качества этих генераторов – в разделе 4.3 того же учебника.

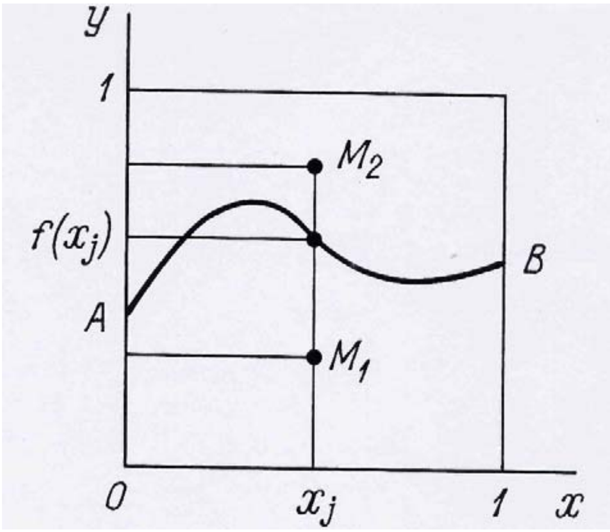


Рис. 2.3. Вычисление интеграла методом имитационного моделирования

Блок-схема алгоритма рассмотренной задачи вычисления значения интеграла представлена на рис. 2.4. Прокомментируем смысл операций, реализуемых в этом алгоритме.

Блок **1 –** с генератора случайных величин (ГСВ) получаются два числа k_i и k_{i+1}. Это случайные величины, равномерно распределенные в интервале [0, 1].

Блок **2 –** формируются координаты случайной точки *M* (см. рис. 2.3 (x_j, y_j).

Блок **3 –** вычисляется значение функции $f(x)$ в точке x_j.

Блок **4 –** выясняется, где находится точка *M*: под кривой или над кривой

$f(x)$. Если под кривой (точка M_1), то осуществляется переход к блоку **5**, а если над кривой (точка M_2), то – к блоку **6**.

Блок **5 –** к содержимому счетчика m количества точек, попавших под кривую или на саму кривую, прибавляется единица: $m = m+1$.

Блок **6 –** к содержимому счетчика N общего количества точек (количества реализаций процесса моделирования) прибавляется единица: $N = N + 1$.

Блок **7 –** проверяется условие $N < N1$, где $N1$ — заданное количество реализаций, необходимое для обеспечения требуемой точности расчета.

Блок **8 –** вычисляется приближенное значение искомого интеграла I.

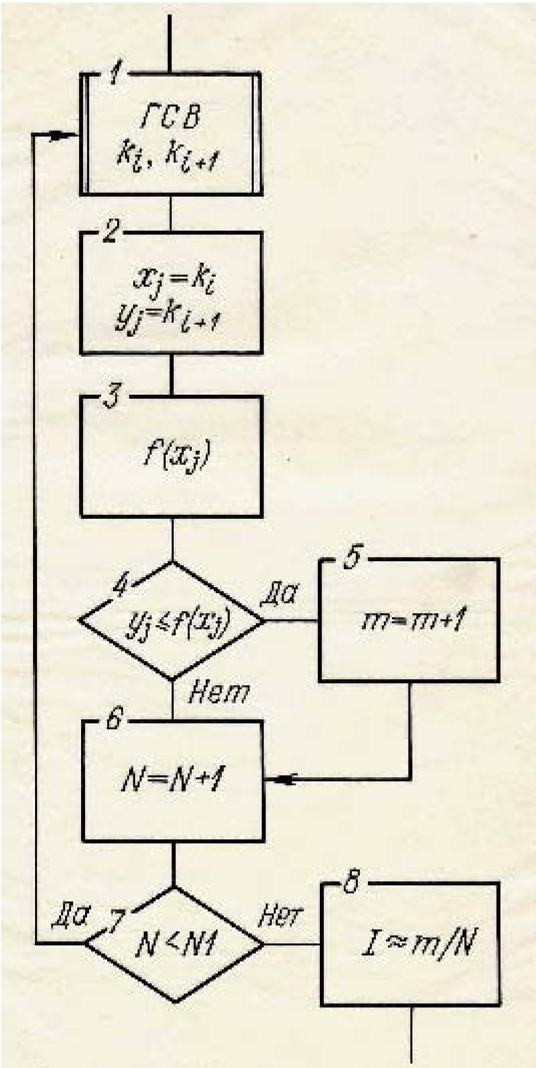


Рис. 2.4. Алгоритм задачи вычисления значения интеграла

Рассмотренный пример иллюстрирует еще одно преимущество метода имитационного моделирования: при реализации метода не требуется запоминания всех случайных чисел, вырабатываемых в процессе моделирования. В памяти компьютера хранятся только значения m и N. Кроме этого, метод устойчив к случайным сбоям в работе ЭВМ, т.е. искажение нескольких результатов отдельных реализаций при большом их общем числе не ведет к существенным ошибкам.

Методом имитационного моделирования в настоящее время решается широкий круг практических задач, среди которых можно выделить две: во-первых, это применение для изучения процессов в так называемых вероятностных системах управления (массовое обслуживание, прохождение ядерных частиц через вещество, надежность сложных систем, распознавание образов); во-вторых, этот метод используется в детерминированных задачах расчета определенных одномерных и многомерных интегралов, при решении алгебраических уравнений, в операциях с матрицами (особенно обращение матриц), при оптимизации дискретных процессов управления. Так, при решении алгебраических уравнений методом имитационного моделирования количество вычислительных операций пропорционально числу уравнений, а при использовании детерминированных вычислительных методов оно пропорционально кубу числа уравнений.

Большое теоретическое и практическое значение получили исследования с помощью имитационного моделирования процессов передачи сообщений, кинетики химических реакций, а также функционирования сложных производственных, информационных и автоматизированных систем управления, экономических и биологических объектов.

5.2. Случайные числа

Случайные числа необходимы в целом ряде прикладных областей: имитационное моделирование, численный анализ, программирование на компьютере, принятие решений и др. В этой связи важно грамотно выбирать способы получения последовательностей случайных чисел, реализуемые с наименьшими по возможности затратами и обеспечивающими простоту и удобство последующих преобразований.

Этим требованиям удовлетворяют **случайные числа с равномерным распределением в интервале [0,1].** Можно напомнить, что непрерывная случайная величина имеет равномерное распределение в интервале $[a,b]$, если ее функции плотности $f(x)$ и распределения $F(x)$, а также математическое ожидание m_x, дисперсия σ_x^2 и среднее квадратическое отклонение σ_x соответственно равны:

$$f(x)=\begin{cases}0, & при \quad x<a,\\ 1/(b-a), & при \quad a\le x\le b,\\ 0, & при \quad x>b.\end{cases}$$

$$F(x)=\begin{cases}0, & при \quad x<a,\\ \dfrac{x-a}{b-a}, & при \quad a\le x\le b,\\ 1, & при \quad x>b.\end{cases}$$

$$m_x=\int_a^b xf(x)dx=\frac{1}{b-a}\int_a^b xdx=\frac{1}{b-a}\frac{x^2}{2}\Big|_a^b=\frac{b+a}{2},$$

$$\sigma_x^2=\int_a^b (x-m_x)^2 f(x)dx=\frac{(b-a)^2}{12},$$

$$\sigma_x = \sqrt{\sigma_x^2} = \frac{b-a}{2\sqrt{3}}\ .$$

Случайная величина называется равномерно распределенной в интервале [0,1], если ее плотность распределения $f(x)$ имеет вид

$$f(x) = \begin{cases} 1, & x \in [0,1], \\ 0, & x \notin [0,1]. \end{cases}$$

Функция распределения в этом случае имеет вид:

$$F(x) = \begin{cases} 0, & при \quad x < 0, \\ x, & при \quad 0 \le x \le 1, \\ 1, & при \quad x > 1. \end{cases}$$

Математическое ожидание m_x и дисперсия σ_x^2 соответственно равны:

$$m_x = \int_0^1 x dx = \frac{1}{2};$$

$$\sigma_x^2 = \int_0^1 \left(x - \frac{1}{2}\right)^2 dx = \frac{1}{12}.$$

До появления компьютеров широкое распространение имели физические генераторы случайных величин, в основе которых лежит естественный механизм, основанный на организации выборки из стандартного флуктуационного процесса. Простейшим устройством, основанном на этом принципе, является механическая рулетка. Получаемые от физических генераторов числа по самой сути «случайные», поскольку основываются на флуктуационных процессах, стохастичность которых доказана практикой.

Развитие компьютерных технологий дало мощный толчок совершенствованию математических методов генерирования случайных величин. Но эти величины, в основе получения которых лежат соответствующие детерминированные алгоритмы, по сути своей не являются случайными. Поэтому более правильно было бы называть их псевдослучайными. Хотя они и имеют соответствующие статистические характеристики. Наилучшие из известных сегодня генераторов случайных величин реализованы на основе конгруэнтных методов – линейного, смешанного и аддитивного [18, 19, 25].

В основу построения всех программных генераторов случайных величин положено использование следующего рекуррентного соотношения

$$\xi_{i+1} = f(\xi_i)\ ,$$

т.е. каждое очередное случайное число ξ_{i+1} получается в результате некоторого преобразования f над предыдущим числом ξ_i или группой предыдущих случайных чисел. Преобразование f – это некоторое перемешивание предыдущих чисел и выполнение каких-нибудь арифметических операций.

Например, пусть ξ_0 = 2061, возводя его в квадрат получают ξ_0 = 04247721. Выделяя из последовательности цифр, например, четыре средних разряда 2477, говорят, что ξ_1 = 2477. Затем $\xi_1^{\ 2}$ = 06(1355)29 и ξ_2 = 1355 и т.д. Если к итоговой последовательности чисел 2061, 2477, 1355, … приписать ноль с запятой (с точкой), то получаемые числа 0.2061; 0.2477; 0.1355; … действительно находятся в интервале [0,1]. О проверке равномерности распределения этой последователь-ности будет сказано в следующем параграфе 5.3.

Можно привести один пример известного алгоритма получения последовательности случайных величин, хотя на практике этих алгоритмов используются сотни или даже тысячи. Блок-схема алгоритма этого генератора изображена на рис. 2.5. Исходными данными в алгоритме рисунка 2.5 являются величины U_0 = 0.5421 и U_1 = 3.1416. ξ – это получаемые случайные величины, I – текущий номер величины ξ, U – заданное количество получаемых величин.

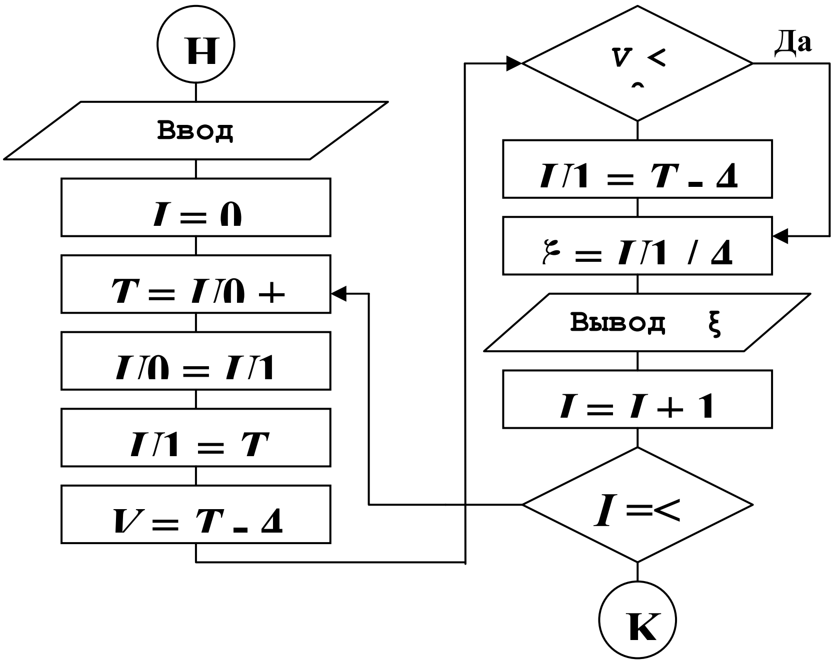


Рис. 2.5. Пример алгоритма программного генератора

При программировании на различных алгоритмических языках (Бейсике, Паскале, Си, Прологе и т.д.) всегда можно воспользоваться стандартными функциями, реализующими получение последова-тельностей случайных величин, равномерно распределенных в интервале [0,1]. Это функция ***random.*** Алгоритм ее работы, как правило, неизвестен. Поэтому особенно тщательно нужно проверять качество последовательности величин, получаемых с помощью этого генератора (об этом следующий параграф).

Фрагмент программы на Паскале использования генератора ***random*** и двадцать чисел, полученных при реализации этого генератора, приведены ниже.

```
program demoran1;
uses crt;
var
  i: integer;   {i - количество генерируемых чисел.}
```

```
  n: real;      {n - генерируемое число.}
begin
   clrscr;
   writeln ('  Программа генерирует двадцать');
   writeln ('  случайных чисел от 0 до 1');
   writeln ('  и выводит их на экран в четыре строчки');
   writeln ('  по 5 чисел в каждой.');
   writeln;
   randomize;
   for i:=1 to 20 do
      begin
         n:=random;
         write(n:8:3);
         if i mod 5 = 0 then writeln;
      end;
   readln;
end.
```

```
      Случайные величины, полученные при использовании стандартной функции  random:
                   0.123      0.005     0.976     0.588     0.342
                   0.496      0.903     0.005     0.427     0.349
                   0.427      0.526     0.637     0.923     0.437
                   0.846      0.457     0.439     0.752     0.376
```

5.3. Проверка качества последовательностей случайных величин

Чтобы быть уверенным в результатах имитационного эксперимента, предварительно необходимо убедиться в случайности используемых последовательностей случайных величин. А качество случайных чисел, получаемых программным способом на компьютере, зависит от того, насколько удачно построен алгоритм, или подобраны коэффициенты и начальные значения параметров генераторов.

Статистическая теория предлагает целый ряд количественных критериев случайности. Однако, если последовательность удовлетворяет относительно тестов $T_1, T_2, ..., T_n$, все же нет уверенности, что и тест T_{n+1} она выдержит столь же успешно. И все-таки чем больше тестов прошла последовательность, тем надежнее получаются результаты. Как правило, случайные числа проходят проверку с помощью следующих тестов: универсальных тестов (критерий χ^2 («хи-квадрат»), критерий Колмогорова – Смирнова (КС-критерий)) и эмпирических тестов (проверка по моментам распределения, проверка на равномерность, проверка серий, проверка интервалов, проверка комбинаций, проверка с помощью вычисления какой-нибудь общеизвестной константы).

Проверка по моментам распределения. Как отмечалось в § 5.2, математическое ожидание и дисперсия равномерно распределенной случайной последовательности в интервале [0,1] равны $\frac{1}{2}$ и $\frac{1}{12}$ соответственно. Пусть имеется последовательность чисел $\xi_1, \xi_2, ..., \xi_N$, полученная с использованием какого-нибудь программного генератора. Для этих чисел

$$m_\xi \approx \frac{1}{N}\sum_{i=1}^{N}\xi_i;$$

$$\sigma_\xi^2 \approx \frac{1}{N}\sum_{i=1}^{N}\xi_i^2 - \frac{1}{N^2}\left(\sum_{i=1}^{N}\xi_i\right)^2.$$

Если генерируемые числа близки к равномерной случайной последовательности в интервале [0,1], то при достаточно больших N

$$m_x \approx \frac{1}{2}, \quad \sigma_x^2 \approx \frac{1}{12}.$$

Проверка на равномерность. Интервал [0,1] разбивается на n равных подынтервалов и фиксируется, в какой из подынтервалов попадают числа ξ_i. Пусть m_1 – количество случайных чисел, попавших в первый интервал; m_2 – во второй и т.д. Ясно, что

$$m_1 + m_2 + \ldots + m_n = N.$$

Затем вычисляются частоты попадания случайных чисел в каждый из подынтервалов

$$\mu_i = \frac{m_i}{N}, \quad i = 1, 2, .., n.$$

Если случайная последовательность чисел равномерная, то при больших N гистограмма (ломаная линия) должна приближаться к теоретической прямой $y = \frac{1}{n}$.

Если число разбиений $n = 10$, то $y = \frac{1}{10}$. На рис. 2.6 приведены результаты исследования последовательности случайных чисел (для $N = 1000$), полученных с использованием стандартной функции ***random*** языка Паскаль.

Проверка по критерию χ^2. Применение этого метода основано на следующей процедуре:

1. Отрезок [0,1] разбивается на n подынтервалов (n принимают равным от 10 до 20).

2. По совокупности N чисел подсчитывается количество m_i, попавших в i-й подынтервал ($i = 1, 2, \ldots, n$).

3. Определяется эмпирическое значение

$$\chi_э^2 = \sum_{i=1}^{n}\frac{(m_i - Np)^2}{Np}, \qquad \text{где} \quad p = \frac{1}{n}.$$

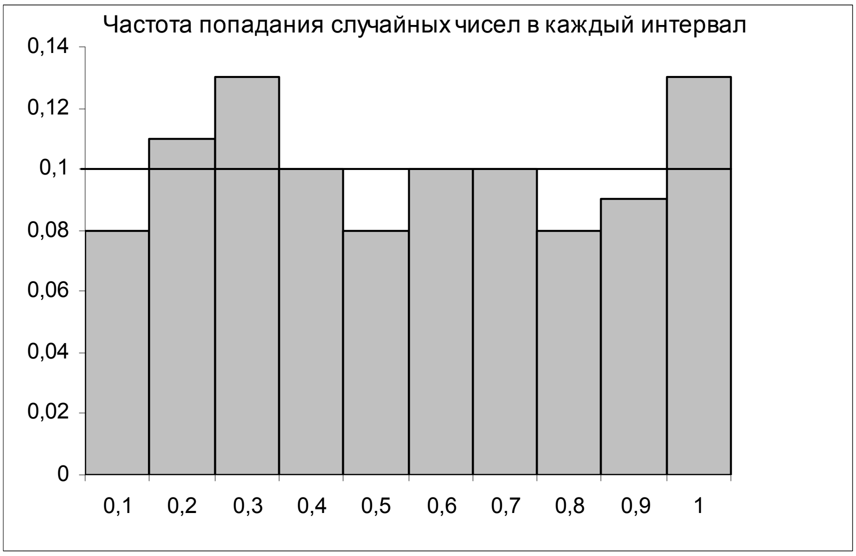


Рис. 2.6. Результат исследования последовательности чисел на равномерность

4. Для доверительной вероятности α и числа степеней свободы $l = n - 1$ определяется теоретическое значение критерия

$$\chi_m^2 = \chi_m^2(l, \alpha).$$

5. При выполнении условия $\chi_э^2 \le \chi_m^2$ гипотеза о равномерности принимается, в противном случае – признается несостоятельной.

Проверка с помощью вычисления какой-нибудь общеизвестной константы. Для реализации этой проверки выбирается, например, вычисление числа π. Значение его известно и определено с большой точностью: π = 3.1416... . Нужно выбрать какую-нибудь формулу, в записи которой участвует число π. Например, площадь круга: $S = \pi R^2$ и, если $R = 1$, то $S = \pi$. То есть, вычислив методом имитационного моделирования площадь круга с радиусом, равным единице, и проверив, на сколько эта площадь отличатся от величины 3.1416, можно судить о качестве генератора, выдающего последовательность случайных величин, равномерно распределенных в интервале [0,1].

Для вычисления *S* необходимо знать уравнение окружности, а оно известно: $x^2 + y^2 = 1$. Вычисление произвольной площади (вычисление интеграла) методом имитационного моделирования описано в § 5.1. Для этого нужно иметь аналитическое выражение подынтегральной функции, которое для рассматриваемого примера записывается очень просто $y = \pm\sqrt{1 - x^2}$. Так как все действия производятся в первом координатном углу, то $y = \sqrt{1 - x^2}$, $x \in [0,1]$, и так как вычисляется площадь только четверти круга (см. § 5.1 и рис. 2.3), то искомая величина четверти круга $S \approx \dfrac{m}{N}$. А площадь всего круга *S* и, следовательно, число $\pi \approx 4S = \frac{4m}{N}$. Здесь $m-$ число точек, попавших при имитационном эксперименте под кривую $y = \sqrt{1-x^2}$ или на саму кривую, а $N-$ общее количество точек, брошенных в единичный квадрат $[0,1]\times[0,1]$.

Чем меньше отличается вычисленное значение π от точного (3.1416), тем с большей уверенностью можно считать, что статистические характеристики генератора случайных величин, который использовался при решении этой задачи, являются удовлетворительными. При этом необходимо учитывать, что точность вычисления зависит, конечно, от количества точек, брошенных в единичный квадрат, т.е. от количества реализаций процесса моделирования. Но это отдельный вопрос и он освещается в § 5.6.

5.4. Моделирование случайных процессов

При моделировании процессов функционирования систем массового обслуживания часто возникает необходимость имитировать случайные события. При этом учитывается, что выполнение любого события (любой поставленной задачи) характеризуется определенным значением вероятности желательного (или нежелательного) исхода. Такими событиями могут быть: явка студента на занятия, поступление запроса в систему, состояние технических средств системы управления и пр. Как моделировать такие и подобные события? Например, как отразить в модели тот факт, что к началу функционирования (к моменту включения) объект будет находиться в исправном состоянии с вероятностью, равной, допустим, 0.99?

В этом параграфе рассматриваются типовые приемы моделирования простых и сложных случайных событий, вероятности свершения которых известны.

Считается, что в распоряжении экспериментатора (исследователя, субъекта), реализующего процесс имитационного моделирования, имеется возможность получать последовательность случайных величин, равномерно распределенных в интервале [0,1] (см. § 5.2).

Реализация события А. Это единичное случайное событие A (поступление заявки в СМО, отказ аппаратуры и т.п.), наступающее с вероятностью $P(A)$. Моделью такого события является попадание значения случайной величины ξ, равномерно распределенной в интервале [0,1], в интервал [0, $P(A)$]:

$$P(0 \le \xi \le P(A)) = \int_0^{P(A)} f(x)dx = \int_0^{P(A)} 1dx = P(A).$$

Следовательно, о том свершилось или не свершилось событие A, судят по значению равномерно распределенной случайной величины ξ_i. Если значение случайной величины удовлетворяет условию

$$\xi_i \le P(A) \tag{2.1}$$

то считают, что событие A свершилось. Если же

$$\xi_i > P(A),$$

то свершившимся считается противоположное событие $\overline{A}$.

Например, пусть вероятность $P(A)$ поступления заявки в СМО равна 0.75, случайное число $\xi_i = 0.95$. Это говорит о том, в *i*-й реализации заявка в систему не поступила (наступило событие $\overline{A}$ – рис. 2.7). Если же $\xi_i = 0.53$, то – считают, что событие *A* наступило.

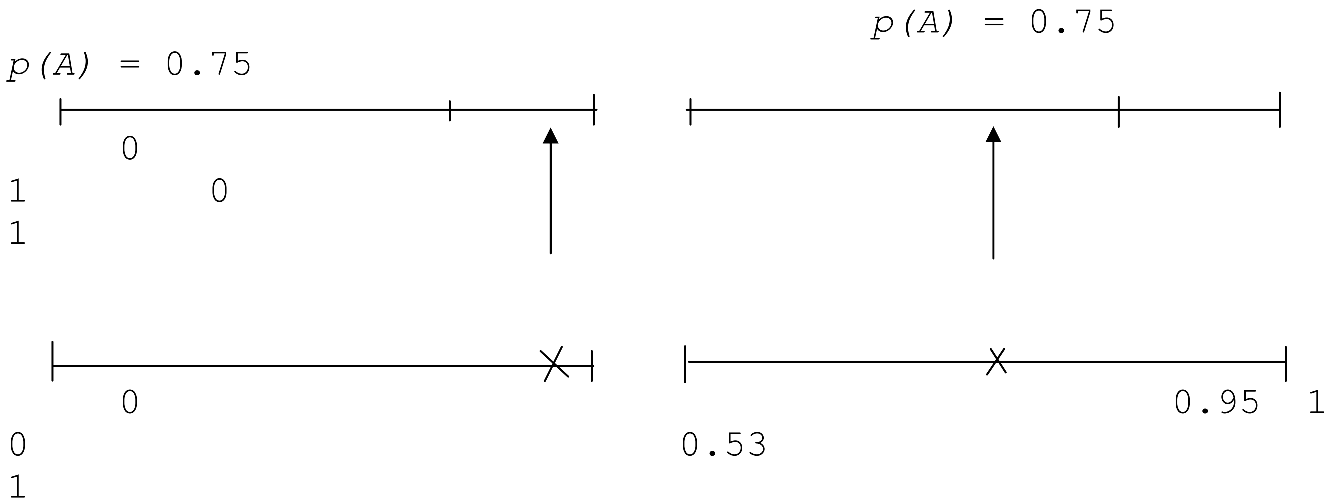


Рис. 2.7. Реализация события *A*: событие не произошло (слева), событие произошло (справа)

Реализация группы событий. Пусть $A_1, A_2, ..., A_n$ – полная группа несовместных событий, наступающих с вероятностями $p_1, p_2, ..., p_n$; $\sum_{i=1}^{n} p_i = 1$. Реализация таких событий производится следующим образом:

1. Отрезок [0,1] разбивается на *n* частей с длинами $p_1,\ p_2, ..., p_n$. Точки деления отрезка имеют такие координаты:

$$l_0 = 0,\quad l_1 = p_1,\quad l_2 = p_1 + p_2,\quad ...\quad , l_s = p_1 + p_2 + ... + p_s,\ ...$$

$$l_n = \sum_{i=1}^{n} p_i = 1.$$

2. Пусть ξ_i – *i*-я случайная величина, равномерно распределенная в интервале [0,1]. Если

$$l_{k-1} < \xi_i \le l_k\ , \qquad (2.2)$$

то считают, что произошло событие A_k. Действительно, при выполнении соотношения (2.2)

$$P(A_k) = P(l_{k-1} \le \xi_i < l_k) = l_k - l_{k-1} = p_k.$$

Такая процедура моделирования называется **схемой испытаний по жребию.**

Пример 2.3. Канал передачи данных может находиться в четырех состояниях с соответствующими вероятностями:

1. Исправен и свободен, $p_1 = 0.20$.
2. Исправен и занят, $p_2 = 0.35$.
3. Неисправен, $p_3 = 0.15$.
4. Подавлен помехами, $p_4 = 0.30$.

Расчет l_k: $l_1 = 0.20$; $l_2 = 0.20 + 0.35 = 0.55$; $l_3 = 0.55 + 0.15 = 0.70$; $l_4 = 1.00$.

Пусть при первой реализации $\xi_1 = 0.95$. Так как $l_3 < \xi_1 \le l_4$, то считают, что произошло событие 4 – канал подавлен помехами. Если же $\xi_2 = 0.05$, то – событие 1 – канал исправен и свободен (рис. 2.8).

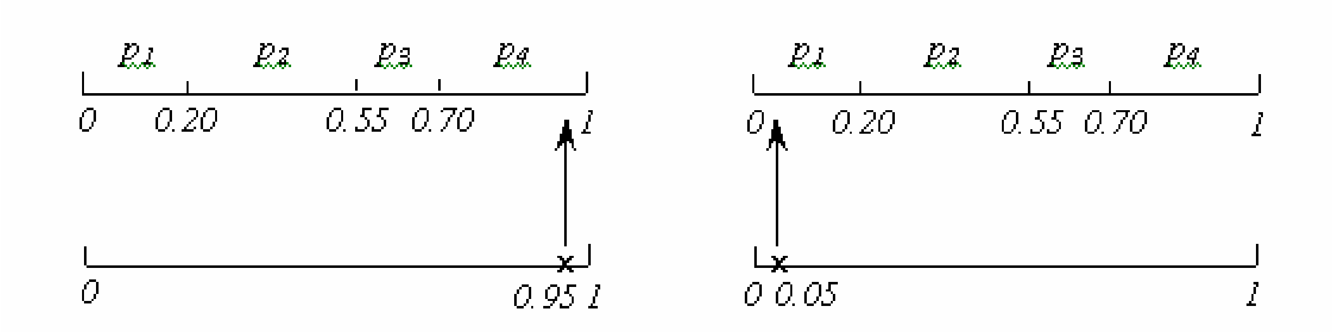


Рис. 2.8. Моделирование группы из четырех несовместных событий

Реализация сложного события, состоящего из двух независимых событий A и B. Пусть независимые события *A* и *B* наступают с вероятностями p_A и p_B соответственно. Если последовательно пользоваться соотношением (2.1) относительно событий *A* и *B*, то получается **алгоритм 1**, блок-схема которого представлена на рис. 2.9. То есть для реализации наступления одного из событий требуется использование двух чисел ξ_i и ξ_{i+1} и две проверки соотношения (2.1).

Если же пользоваться соотношением (2.2), то реализация сложного события производится по **алгоритму 2**. Для его осуществления необходимо рассчитать вероятности наступления сложных событий $AB, A\overline{B}, \overline{A}B, \overline{A}\overline{B}$:

$$p_A p_B, \quad p_A(1-p_B), \quad (1-p_A)p_B, \quad (1-p_A)(1-p_B).$$

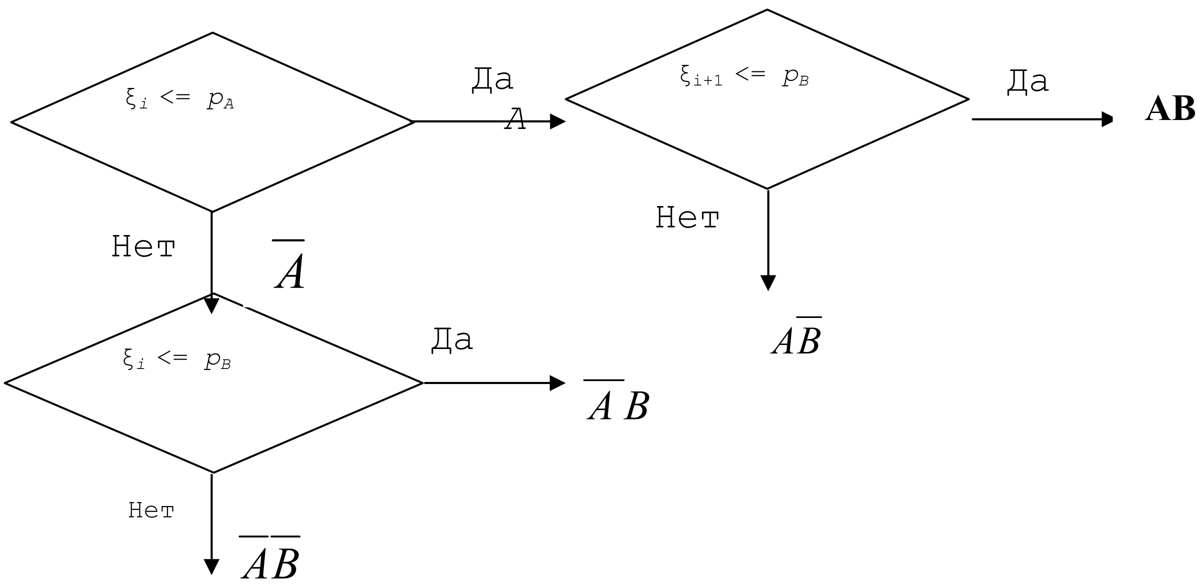


Рис. 2.9. Блок-схема алгоритма 1 при реализации сложного события, состоящего из двух независимых событий

Когда конкретные значения вероятностей событий $AB, A\overline{B}, \overline{A}B, \overline{A}\overline{B}$ будут рассчитаны, то их можно обозначить соответственно: p_1, p_2, p_3, p_4. Затем, как уже говорилось, используется соотношение (2.2).

Пример 2.4. Осуществить розыгрыш наступления сложного события, состоящего из независимых событий A и B. Вероятности событий $p_A = 0.3$ и $p_B = 0.4$.

Рассчитываются соответствующие вероятности сложного события:

$p_1 = p_A\, p_B = 0.3 \cdot 0.4 = 0.12$,
$p_2 = p_A(1 - p_B) = 0.3 \cdot 0.6 = 0.18$,
$p_3 = (1 - p_A)\, p_B = 0.7 \cdot 0.4 = 0.28$,
$p_4 = (1 - p_A)(1 - p_B) = 0.7 \cdot 0.6 = 0.42$.

Если $\sum_{i=1}^{n} p_i = 1$, а это, действительно, так, то для использования соотношения (2.2) рассчитываются величины l_s

$$l_0 = 0, \quad l_1 = p_1 = 0.12, \quad l_2 = p_1 + p_2 = 0.12 + 0.18 = 0.30,$$
$$l_3 = 0.3 + 0.28 = 0.58, \quad l_4 = 0.58 + 0.42 = 1.$$

Если $\xi_1 = 0.95$, то в соответствии с соотношением

(2.2) наступает событие $\overline{A}\overline{B}$, если же ξ_2 = 0.05, то событие AB и т.д.

Реализация сложного события, состоящего из двух зависимых событий A и B. При этом, пусть заданы вероятности событий p_A, p_B и $p(B/A)$. Величина вероятности $p(B/\overline{A})$, необходимая для дальнейших расчётов, вычисляется по формуле полной вероятности

$$p_B = p_A p(B/A) + p_{\overline{A}} p(B/\overline{A}).$$

Откуда

$$p(B/\overline{A}) = \frac{p_B - p_A p(B/A)}{1 - p_A}. \tag{2.3}$$

Как и в случае независимых событий, имеются два варианта построения алгоритмов реализации сложного события, причем оба варианта аналогичны рассмотренным выше.

В случае использования соотношения (2.1) **алгоритм 1** аналогичен процедуре, показанной на рис. 2.9.

В случае же использования соотношения (2.2) (**алгоритм 2**), как и прежде, для его осуществления необходимо рассчитать вероятности наступления сложных событий $AB, A\overline{B}, \overline{A}B, \overline{A}\overline{B}$. Но так как события зависимые, то

$$p_1 = p_A p(B/A),$$
$$p_2 = p_A(1 - p(B/A)),$$
$$p_3 = (1 - p_A) p(B/\overline{A}),$$
$$p_4 = (1 - p_A)(1 - p(B/\overline{A})).$$

Когда конкретные значения вероятностей событий $AB, A\overline{B}, \overline{A}B, \overline{A}\overline{B}$ будут рассчитаны, их можно, как это сделано выше, обозначить соответственно: p_1, p_2, p_3, p_4. Затем, как уже говорилось, используется соотношение (2.2).

Пример 2.5. Реализовать сложное событие, состоящее из двух зависимых событий *A* и *B*. Заданы вероятности событий $p_A = 0.3$, $p_B = 0.4$ и $p(B/A) = 0.8$.

Величина вероятности $p(B/\overline{A})$ вычисляется по формуле (2.3):

$$p(B/\overline{A}) = \frac{0.4 - 0.3 \cdot 0.8}{0.7} = \frac{0.16}{0.7} \approx 0.23.$$

Рассчитываются вероятности сложных событий $AB, A\overline{B}, \overline{A}B, \overline{A}\overline{B}$:

$$p_1 = p_A p(B/A) = 0.3 \cdot 0.8 = 0.24,$$
$$p_2 = p_A(1 - p(B/A)) = 0.3 \cdot 0.2 = 0.06,$$
$$p_3 = (1 - p_A) p(B/\overline{A}) = 0.161,$$
$$p_4 = (1 - p_A)(1 - p(B/\overline{A})) = 0.539.$$

Для проверки правильности математических выкладок вычисляется сумма вероятностей несовместных событий p_i, которая должна равняться единице:

$$\sum_{i=1}^{4} p_i = 0.24 + 0.06 + 0.161 + 0.539 = 1.0\,.$$

А теперь так же, как в предыдущем примере, рассчитываются l_s, чтобы можно было пользоваться соотношением (2.2) и т.д., как в примере 2.4.

Реализация однородной цепи Маркова. Прежде всего необходимо определиться с терминологией. **Случайная последовательность событий называется марковской цепью, если для каждого шага вероятность перехода из состояния S_i в любое состояние S_j не зависит от того, когда и как система пришла в состояние S_i** [22, 41]. Пусть характер смены состояний определяется матрицей переходов:

$$\Pi = \begin{Vmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{Vmatrix},$$

где любой элемент матрицы p_{ij} – условная вероятность наступления события (состояния) S_j при условии, что предыдущим было событие S_i. Система может находиться в состояниях $S_1, S_2, \ldots, S_n$. Начальные состояния могут задаваться распределением вероятностей $p_{01}, p_{02}, \ldots, p_{0n}$. Или, если распределение не задано, то

$$p_{0i} = \frac{1}{n}, \quad \text{где} \quad i = 1, 2, \ldots, n\,.$$

Для реализации однородной марковской цепи используется соотношение (2.2), которое применяется к каждой строке матрицы переходных вероятностей.

Необходимо помнить, что сумма элементов любой строки матрицы Π должна равняться единице, как сумма вероятностей полной группы несовместных событий.

Пример 2.6. Для матрицы переходов системы $S = \{S_1, S_2, S_3\}$

$$\Pi = \begin{Vmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.4 & 0.3 \\ 0.7 & 0.2 & 0.1 \end{Vmatrix}$$

реализовать однородную марковскую цепь. Начальные состояния системы равновероятны.

Как было сказано выше, однородная марковская цепь реализуется при использовании соотношения (2.2). Так как начальные состояния равновероятны, а количество состояний в системе S всего три, то $p_{0i} = 1 / 3$, т.е. $p_{01} = 0.33$; $p_{02} = 0.33$ и $p_{03} = 0.33$.

Теперь, необходимо произвести подготовку к дальнейшему использованию соотношения (2.2), т.е. нужно рассчитать величины l_s для начальных состояний и для каждой строки матрицы переходных вероятностей. Для определения начального состояния:

$$l_{00} = 0;\ l_{01} = p_{01} = 0.33;\ l_{02} = p_{01} + p_{02} = 0.33 + 0.33 = 0.66;$$

$$l_{03} = p_{01} + p_{02} + p_{03} = 0.33 + 0.33 + 0.33 = 1.$$

Для переходов из первого состояния (в соответствии с первой строкой матрицы Π):

$$l_{10} = 0;\ l_{11} = p_{11} = 0.2;\ l_{12} = p_{11} + p_{12} = 0.2 + 0.3 = 0.5;$$
$$l_{13} = p_{11} + p_{12} + p_{13} = 0.2 + 0.3 + 0.5 = 1.$$

Для второй строки:

$$l_{20} = 0;\ l_{21} = p_{21} = 0.3;\ l_{22} = p_{21} + p_{22} = 0.3 + 0.4 = 0.7;$$
$$l_{23} = p_{21} + p_{22} + p_{23} = 0.3 + 0.4 + 0.3 = 1.0.$$

И, наконец, для третьей строки:

$$l_{30} = 0;\ l_{31} = 0.7;\ l_{32} = 0.9;\ l_{33} = 1.0.$$

Теперь, обращаясь с генератору случайных величин, получают последовательность ξ_i и, сравнивая каждую из них с соответствующим набором величин l_s (то есть пользуясь соотношением (2.2)), получают реализацию однородной марковской цепи (получают последовательность переходов системы S из состояния в состояние). Можно воспользоваться последовательностью случайных величин, приведенных на с. 80. Пусть $\xi_1 = 0.123$, тогда $l_{00} < \xi_1 < l_{01}$, следовательно, в начальный момент времени система S находилась в состоянии S_1. Для продолжения эксперимента пусть $\xi_2 = 0.005$, а набор величин l_s должны соответствовать первой строке матрицы переходов П, тогда в соответствии с соотношением (2.2) $l_{10} < \xi_2 < l_{11}$, а это свидетельствует о том, что система S, находясь в состоянии S_1, в нем же S_1 и останется. Затем пусть $\xi_3 = 0.976$, тогда $l_{12} < \xi_3 < l_{13}$, а это говорит о том, что система S из состояния S_1 переходит в S_3. Далее нужно пользоваться величинами l_s, рассчитанными для третьей строки матрицы П, и т.д. В итоге получается следующая последовательность реализаций (переходов)

$$\boldsymbol{S_1 \rightarrow S_1 \rightarrow S_3 \rightarrow S_1 \rightarrow S_2 \rightarrow S_2 \rightarrow} \quad \text{и т.д.}$$

5.5. Получение случайных величин с заданным законом распределения

Задача получения случайных величин с заданным законом распределения может быть сформулирована так: **преобразовать возможные значения случайной величины ξ_i, имеющей равномерное распределение в интервале [0,1], в возможные значения x_i случайной величины с заданным законом распределения.**

Существует всего два пути решения этой задачи. Первый называется прямым и предусматривает реализацию некоторой операции над числом ξ_i, формирующей число x_i, подчиняющееся заданному закону распределения. Второй способ основан на моделировании условий предельных теорем теории вероятностей.

Существо первого подхода вытекает из следующей теоремы.

Теорема 2.2. Если случайная величина X имеет плотность распределения $f(x)$, то распределение случайной величины

$$\xi = \int_{-\infty}^{X} f(x)dx$$

является равномерным в интервале [0,1].

Отсюда, чтобы получить число из совокупности $\{x_i\}$, имеющих функцию плотности $f(x)$, необходимо разрешить относительно x_i уравнение

$$\int_{-\infty}^{x_i} f(x)dx = \xi_i$$
(2.4)

Пример 2.7. Получить выражение для вычисления значений случайных величин x_i, равномерно распределенных в интервале [*a*,*b*], т.е. подчиняющихся равномерному закону распределения

$$f(x) = \begin{cases} \frac{1}{b-a}, a < x < b; \\ 0, \ x < a, x > b. \end{cases}$$

Используя соотношение (2.4), получают

$$\int_{a}^{x_i} \frac{1}{b-a} dx = \xi_i$$

и далее

$$\frac{x_i - a}{b - a} = \xi_i,$$

$$x_i = a + (b-a)\xi_i.$$
(2.5)

Пусть необходимо получить последовательность случайных величин, равномерно распределенных в интервале [-10, +20]. Тогда, подставляют пределы искомого интервала в полученное соотношение (2.5):

$$x_i - 10 + (-(-10))\xi_i = -10 + 30\xi_i.$$

Для получения равномерной последовательности в интервале [-30, 0] опять пользуются соотношением (2.5):

$$x_i - 30 + (0 - (-30))\xi_i = -30(1 - \xi_i).$$

В полученном выражении разность $1 - \xi_i$ является тоже случайной величиной, равномерно распределенной в интервале [0, 1], и эту разность можно условно заменить на некоторую величину ξ_i' : $\xi_i' = 1 - \xi_i$.

В итоге получается $x_i = -30\xi_i$.

Пример 2.8. Получить выражение для вычисления значений x_i случайной величины, подчиняющейся экспоненциальному закону распределения

$$f(x) = \lambda e^{-\lambda x}, \quad x > 0 .$$

В силу соотношения (2.4) можно записать

$$\lambda \int_0^{x_i} e^{-\lambda x} dx = \xi_i.$$

После вычисления интеграла

$$1 - e^{-\lambda x_i} = \xi_i.$$

Это уравнение разрешается относительно x_i

$$x_i = -\frac{1}{\lambda} \ln(1 - \xi_i).$$

Если учесть, как и в предыдущем примере, что величина $1 - \xi_i$ также равномерно распределена на интервале [0, 1], то искомое соотношение примет следующий вид:

$$x_i = -\frac{1}{\lambda} \ln \xi_i.$$

Заканчивая рассмотрение прямого метода, следует подчеркнуть, что в большинстве случаев (например, для нормального распределения) уравнение (2.4) не решается точно относительно x_i. В этом случае целесообразно воспользоваться вторым способом, который основан на использовании предельных теорем теории вероятностей.

В силу центральной предельной теоремы (см. теорему 2.1 на с.67) сумма большого числа одинаково распределенных независимых случайных величин при весьма общих условиях имеет приближенно нормальное распределение.

Пусть $\xi_1, \xi_2, ..., \xi_n$ случайные числа, равномерно распределенные на интервале [0,1] и имеющие $m_\xi = \frac{1}{2}$ и $\sigma_\xi^2 = \frac{1}{12}$. Тогда сумма $X = \xi_1 + \xi_2 + ... + \xi_n$ оказывается случайной величиной с распределением, близким к нормальному, а $m_X = \frac{n}{2}$ и $\sigma_X^2 = \frac{n}{12}$. Таким образом, при n = 12 дисперсия равна $\sigma_X^2 = 1$. Если из суммы вычесть число 6, то получают $m_X = 0$. Отсюда, если $\xi_i -$ равномерно распределенные на интервале [0,1] числа, то можно получить x_i - значения случайной величины, распределенной нормально с $m_X = 0$ и $\sigma_X^2 = 1$ по формуле

$$x_i = \sum_{i=1}^{12} \xi_i - 6. \tag{2.6}$$

При исследовании реализаций выражения (2.6) было обнаружено, что формула работает плохо при моделировании «хвостов» нормального распределения, а за пределами $m_X = 2\sigma_X$ имеет место сильная погрешность. В этой связи можно использовать следующую формулу:

$$x_i = ((((C_1 R^2 + C_2)R^2 + C_3)R^2 + C_4)R^2 + C_5)R^2,$$

где

$$R = \frac{\sum_{i=1}^{12} \xi_i - 6}{4};$$

$$C_1 = 0.029899776; \quad C_2 = 0.008355968; \quad C_3 = 0.076542912;$$
$$C_4 = 0.252408784; \quad C_5 = 3.949846138.$$

Эта формула расширяет пределы метода до $\pm 3\sigma_X$.

Все вышеизложенные методы относились к задаче получения значений непрерывных случайных величин с заданным законом распределения. А как формируются значения дискретных величин?

Пусть дискретная случайная величина η принимает счетное множество значений η_i с вероятностями p_i (i = 1, 2, …), а p_i задается соотношением

$$p_i = P(\eta = \eta_i).$$

Ясно также, что $\sum_{i=1}^{\infty} p_i = 1$.

Генерируется очередное ξ_α, имеющее равномерное распределение на интервале [0,1]. Методом последовательных приближений определяют такое η_α^*, которое достаточно точно удовлетворяло бы равенству

$$\sum_{i=1}^{\alpha} p(\eta_i) = \xi_\alpha.$$

Если η_α^* совпадает с одним из η_i, то оно и принимается в качестве значения дискретной случайной величины η. Иначе имеет место неравенство

$$\eta_{i-1} < \eta_\alpha^* < \eta_i,$$

и можно принять за очередное значение η величину η_i.

Пример 2.9. Как имитировать на компьютере (получить последовательность) случайную величину η, имеющую распределение Пуассона

$$p_n = \frac{\lambda^n}{n!} e^{-\lambda}, \quad n = 0, 1, \ \ldots$$

Для этого получаются ξ_i и проверяется справедливость неравенства

$$L_{n-1} < \xi_i < L_n, \tag{2.7}$$

где $$L_\alpha = e^{-\lambda} \sum_{i=0}^{\alpha} \frac{\lambda^i}{i!} \quad (\alpha = 0, 1, 2, \ldots; \quad L_{-1} = 0).$$

Если соотношение (2.7) выполнено, то очередное случайное число η_i принимается равным n. В этом случае η имеет распределение, близкое к пуассоновскому.

Главный недостаток метода – медлительность. В этой связи известен алгоритм, работающий значительно быстрее, в основе которого лежит генерирование ξ_i до тех пор, пока не станет справедливым соотношение

$$\prod_{i=0}^{n} \xi_i \geq e^{-\lambda} > \prod_{i=0}^{n-1} \xi_i \ .$$

Если прямые методы применить не удается или это не выгодно с вычислительной точки зрения, можно воспользоваться удобным и универсальным приемом преобразования случайных чисел, основанным на кусочной аппроксимации функции плотности.

5.6. Оценка точности характеристик. Необходимое число реализаций

Величины, получаемые в результате исследования различных процессов методом имитационного моделирования, обычно являются случайными, потому, что в основу метода положено использование предельных теорем теории вероятностей [6, 29, 32]:

Теорема 2.3. При неограниченном увеличении числа опытов *N* частота события *A* сходится по вероятности к его вероятности.

Теорема 2.4. При достаточно большом числе независимых опытов *N* среднее арифметическое наблюдаемых значений случайной величины X сходится по вероятности к ее математическому ожиданию.

Таким образом, для обеспечения статистической устойчивости соответствующие оценки вычисляются по большому количеству реализаций. Справедливо спросить, насколько велика ошибка от замены вероятности события его частотой, а математического ожидания – средним арифметическим? И каково должно быть число реализаций *N* для того, чтобы эта ошибка с практической достоверностью не вышла за данные пределы? То есть ставится вопрос об оценке точности характеристик случайного процесса.

Пусть в качестве оценки для некоторого параметра a, оцениваемого по результатам моделирования x_i, выбирается величина $\bar{X}$, являющаяся функцией от x_i. В силу случайных причин $\bar{X}$ будет в общем случае отличаться от a. Это отличие характеризуется следующим образом.

Величину ε, когда

$$\left|a - \bar{X}\right| < \varepsilon, \tag{2.8}$$

называют ***точностью*** оценки $\bar{X}$, а вероятность α того, что неравенство выполняется, ее ***достоверностью,*** т.е.

$$P(\left|a - \bar{X}\right| < \varepsilon) = \alpha\ . \tag{2.9}$$

Таким образом, если для оценки параметра *a* систематически используется величина $\bar{X}$, с точностью ε и достоверностью α, то в среднем на каждые 100 случаев применения этого правила, в 100α случаях, $\bar{X}$ будет отличаться от *a* меньше, чем на ε, и только в (1 – α)*100 случаях разница между ними может превосходить ε.

При ответе на все поставленные вопросы пользуются центральной предельной теоремой теории вероятностей (см. теорему 2.1). Согласно этой теореме принимается, что при большом числе опытов *N* их средний результат (частота события *A* или среднее арифметическое наблюдаемых значений случайной величины X) распределяется приближенно по нормальному закону.

Закон распределения частоты события

при большом числе опытов

Пусть моделируется событие *A*, имеющее вероятность появления *p*. Обозначим через x_i величину, равную единице, если в *i*-м испытании произошло событие *A*, и равную нулю, если событие *A* не произошло. Обозначим через m — число испытаний, в результате которых событие *A* произошло:

$$m = \sum_{i=1}^{N} x_i \ .$$

Здесь *N* — общее число испытаний. Отдельные испытания считаются независимыми друг от друга.

Можно определить математическое ожидание Mx_i и дисперсию $\sigma_{x_i}^2$ случайной величины x_i:

$$Mx_i = 0 \cdot (1-p) + 1 \cdot p = p,$$

$$\sigma_{x_i}^2 = M(x_i^2) - \left(Mx_i\right)^2 = 0 \cdot (1-p) + 1 \cdot p - p^2 =$$

$$= p - p^2 = p(1-p).$$

Частота появления события *A* равна $\dfrac{m}{N}$ и является случайной величиной, имеющей математическое ожидание:

$$M\left(\frac{m}{N}\right) = \frac{1}{N} M(m) = \frac{1}{N} \sum_{i=1}^{N} Mx_i = \frac{Np}{N} = p \qquad (2.10)$$

и дисперсию

$$\sigma_{m/N}^2 = \frac{1}{N^2} \sigma_m^2 = \frac{1}{N^2} \sum_{i=1}^{N} \sigma_{x_i}^2 = \frac{Np(1-p)}{N^2} = \frac{p(1-p)}{N}. \qquad (2.11)$$

Теперь можно поставить и решить следующие задачи.

Задача 1. Произведено *N* независимых реализаций, в каждой из которых событие *A* появляется с вероятностью *p*. В результате этих опытов получена частота $\dfrac{m}{N}$ события *A*. Найти вероятность того, что частота $\dfrac{m}{N}$ отличается от вероятности p не больше, чем на заданную величину ε > 0.

Решение. Считая число *N* достаточно большим для того чтобы полагать частоту $\dfrac{m}{N}$ распределенной по нормальному закону с характеристиками (2.10), (2.11), получают:

$$P\left(\left|\frac{m}{N}-p\right|<\varepsilon\right)=2\Phi\left(\frac{\varepsilon}{\sigma_{m/N}}\right)=2\Phi\left(\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}}\right), \qquad (2.12)$$

где $\Phi=\Phi(x)$ – функция Лапласа.

Итак, если вероятность p события *A* известна, можно оценить точность определения этой вероятности по частоте $\frac{m}{N}$ и зависимость этой точности от числа реализаций *N*. В практике моделирования вероятность p обычно неизвестна, поэтому выбирают $N_0=50...100$, по результатам N_0 реализаций определяют m/N_0, а затем принимают, что $p \approx m/N_0$.

Задача 2. Производится ряд независимых опытов, в каждом из которых событие *A* появляется с вероятностью *p*. Каково должно быть число опытов (реализаций) для того, чтобы с заданной, достаточно высокой вероятностью *Q* можно было ожидать, что частота $\frac{m}{N}$ события *A* отклонится от его вероятности p меньше, чем на ε?

Решение. Необходимо задаться каким-нибудь достаточно близким к единице значением вероятности *Q*, которая называется уровнем доверия. Если вероятность того, что частота и вероятность расходятся меньше, чем на ε, будет *Q* или больше, считают задачу решенной. На практике уровень доверия *Q* выбирается значением, близким к единице, например, 0.95, 0.99 или 0.995 и т.д., в зависимости от важности задачи, которая преследуется. Предположим, что вероятность *Q* задана. Приравняем этому значению *Q* правую часть равенства (2.12):

$$2\Phi\left(\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}}\right)=Q, \qquad (2.13)$$

и разрешим уравнение (2.13) относительно *N*:

$$\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}}\Phi^{-1}\left(\frac{1}{2}Q\right),$$

где Φ^{-1} – функция, обратная функции Лапласа.

Отсюда получается формула для числа опытов *N*:

$$N=\frac{p(1-p)}{\varepsilon^2}\left[\Phi^{-1}\left(\frac{1}{2}Q\right)\right]^2. \qquad (2.14)$$

Если по формуле (2.14) *N* оказывается не целым, его округляют в большую сторону до ближайшего целого.

Для вычислений по формуле (2.14) удобно иметь в распоряжении таблицу значений функции $\left(\Phi^{-1}(Q/2)\right)^2$. В табл. 2.4 приведены значения этой функции для некоторых величин уровня доверия *Q*.

Таблица 2.4

Q	0.80	0.85	0.90	0.95	0.97	0.99	0.999	0.9995
$\left(\Phi^{-1}(Q/2)\right)^2$	1.64	2.08	2.71	3.84	4.49	6.61	10,9	12.25

Закон распределения среднего арифметического при большом числе опытов

Пусть моделируется процесс, в котором при каждом из *N* независимых испытаний, случайная величина *k* принимает значения:

$$x_1, x_2, \ldots, x_i, \ldots, x_n .$$

Предположим, что эта величина обладает конечным математическим ожиданием Mx_i и дисперсией $\sigma_{x_i}^2$. Тогда среднее арифметическое

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

распределяется приближенно по нормальному закону с математическим ожиданием

$$M\bar{X} = Mx_i \tag{2.15}$$

и средним квадратическим отклонением

$$\sigma_{\bar{X}} = \frac{\sigma_{x_i}}{\sqrt{N}}. \tag{2.16}$$

На основании сказанного можно решить еще две задачи.

Задача 3. Производится *N* независимых опытов, в каждом из которых наблюдается значение случайной величины *k*, имеющей математическое ожидание Mx_i и среднее квадратическое отклонение σ_{x_i}. Вычисляется среднее арифметическое наблюдаемых значений случайной величины *k*:

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} x_i .$$

Найти вероятность того, что среднее арифметическое $\bar{X}$ отклоняется от математического ожидания Mx_i меньше, чем на заданную величину ε:

$$P\left(\left|\bar{X} - Mx_i\right| < \varepsilon\right).$$

Решение. На основании центральной предельной теоремы, считая число опытов большим, можно утверждать, что случайная величина $\bar{X}$ распределена нормально, с характеристиками (2.15) и (2.16). Отсюда

$$P\left(\left|\bar{X} - Mx_i\right| < \varepsilon\right) = 2\Phi\left(\frac{\varepsilon}{\sigma_{\bar{X}}}\right) = 2\Phi\left(\frac{\varepsilon\sqrt{N}}{\sigma_{x_i}}\right). \tag{2.17}$$

По формуле (2.17) может быть оценена точность определения математического ожидания по среднему арифметическому.

Необходимо отметить, что для оценки точности определения математического ожидания Mx_i методом имитационного моделирования не требуется заранее знать самого математического ожидания случайной величины, зато важно знать ее среднее квадратическое отклонение σ_{x_i}, которое входит в правую часть формулы (2.17).

Обычно на практике, приступая к моделированию случайного явления не известно ни математическое ожидание, ни среднее квадратическое отклонение случайной величины. Однако для приближенной оценки точности моделирования можно в первом приближении вместо σ_{x_i} воспользоваться ее статистической оценкой, полученной в самой серии из *N* реализаций:

$$\sigma_{x_i} \approx \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2 - \overline{X}^2},$$

где $\overline{X}$ – среднее арифметическое. Если точность окажется недостаточной, следует продолжить испытания, внося в среднее квадратическое соответствующие поправки по мере увеличения числа реализаций.

Задача 4. Производится ряд независимых опытов над случайной величиной *k*. Сколько надо сделать опытов, чтобы с заданной вероятностью (уровнем доверия) *Q* ожидать, что среднее арифметическое наблюдающихся значений случайной величины отклонится от ее математического ожидания не более, чем на ε?

Решение. Положим правую часть формулы (2.17) равной уровню доверия *Q* :

$$2\Phi\left(\frac{\varepsilon\sqrt{N}}{\sigma_{x_i}}\right) = Q \qquad (2.18)$$

и разрешим уравнение (2.18) относительно *N*. Получим

$$N = \frac{\sigma_{x_i}^2}{\varepsilon^2}\left[\Phi^{-1}\left(\frac{1}{2}Q\right)\right]^2. \qquad (2.19)$$

где $\left(\Phi^{-1}(Q/2)\right)^2$ – функция, приведенная в табл. 2.4.

5.7. Фиксация и обработка результатов моделирования

При моделировании сложных систем для удовлетворения требований точности проводится большое количество испытаний (см. § 5.6), что сопряжено со значительными затратами компьютерного времени для организации хранения информации о состоянии системы, усложняет последующую обработку и анализ результатов. Отсюда следует необходимость такой организации вычислительного процесса, при которой оценки интересующих параметров могли бы быть получены в ходе моделирования и не требовали бы запоминания и последующей переработки больших объемов данных.

Ниже приводятся основные приемы формирования оценок для искомых величин.

Оценка вероятности состояния системы. Пусть требуется определить вероятность некоторого состояния *А*. Тогда оценкой для искомой величины вероятности будет частота наступления этого состояния A при некотором числе испытаний. Пусть в результате разыгрывания N выборок процесса получено m случаев появления состояния A. Тогда оценкой $\overline{p}(A)$ для вероятности $p(A)$ состояния A служит величина

$$\overline{p}(A) = \frac{m}{N}.$$

Оценка вероятности возможных значений случайной величины. Область значений случайной величины разбивается на n интервалов. Вводятся переменные m_k (k = 1,2, …, n), принимающие значения числа попаданий в эти интервалы. Тогда оценкой для вероятности попадания случайной величины в k-й интервал служит величина $\overline{p}_k = \dfrac{m_k}{N}$.

Оценка среднего значения случайной величины. Для этого необходимо накапливать сумму $\sum x_k$ возможных значений x_k, случайной величины, которые она принимает при различных реализациях процесса. Тогда можно определить математическое ожидание, которое приближенно равно среднему значению $\overline{X}$ при большом числе N реализаций процесса

$$m_X \approx \overline{X} = \frac{1}{N}\sum_{k=1}^{N} x_k \quad . \tag{2.20}$$

Оценка дисперсии случайной величины. Как видно, оценку

$$\sigma_X^2 = \frac{1}{N-1}\sum_{k=1}^{N}(x_k - \overline{X})^2 \tag{2.21}$$

для дисперсии случайной величины непосредственно по формуле (2.21) получать неудобно, поскольку в процессе вычислений постоянно изменяется $\overline{X}$, что требует запоминания N значений x_k.

В этой связи можно воспользоваться формулой, полученной в результате преобразования выражения (2.21),

$$S^2 = \frac{1}{N-1}\sum_{k=1}^{N} x_k^2 - \frac{1}{N(N-1)}\left(\sum_{k=1}^{N} x_k\right)^2,$$

что потребует накапливать суммы $\sum\limits_{k=1}^{N} x_k^2$ и $\sum\limits_{k=1}^{N} x_k$.

Пример 2.10. Получено десять реализаций случайной величины X (табл. 2.5). Найти оценки среднего значения и дисперсии величины X.

Таблица 2.5

Номера реализаций	1	2	3	4	5	6	7	8	9	10
x_k	28	27	30	33	32	29	32	31	27	31

По формуле (2.20) имеем

$$m_X = \frac{1}{10} \cdot 300 = 30.$$

По формуле (2.21) получаем

$$\sigma^2 = \frac{1}{10-1} \cdot 9042 - \frac{1}{10(10-1)} \cdot 90000 = 1004{,}7 - 1000 = 4{,}7.$$

Оценка корреляционного момента случайной величины. Корреляционный момент $K_{\xi\eta}$ для случайных величин ξ и η с возможными значениями x_k и y_k можно рассчитать по формуле

$$\bar{K} = \frac{1}{N-1} \sum_{k=1}^{N} (x_k - \bar{X})(y_k - \bar{Y}), \tag{2.22}$$

а после преобразования выражения (2.22) получают:

$$\bar{K} = \frac{1}{N-1} \sum_{k=1}^{N} x_k y_k - \frac{1}{N(N-1)} \sum_{k=1}^{N} x_k \sum_{k=1}^{N} y_k . \tag{2.23}$$

Пример 2.11. Получено четыре реализации случайных величин *X* и *Y*. Произвести предварительные вычисления для определения корреляционного момента (табл. 2.6) по формуле (2.23).

Таблица 2.6

K	x_k	y_k	$x_k\ y_k$
1	0	1	0
2	2	4	8
3	2	3	6
4	3	5	15
	$\sum_{k=1}^{4} x_k = 7$	$\sum_{k=1}^{4} y_k = 7$	$\sum_{k=1}^{4} x_k y_k = 29$

Глава 6
МОДЕЛИРОВАНИЕ СМО МЕТОДОМ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

В главе 5 рассмотрены принципы, положенные в основу имитационного моделирования. Теперь необходимо ознакомиться с методологией построения моделирующих алгоритмов систем массового обслуживания [7, 12, 13, 17, 18, 20, 24, 26, 33, 37, 39, 40].

6.1. Моделирование потоков заявок

СМО разделяются на однофазные и многофазные, одноканальные и многоканальные, системы с отказами и системы с ожиданием, которые в свою очередь делятся на системы с неограниченным ожиданием заявки в очереди и системы с ограниченным ожиданием. В системах с ограниченным ожиданием на пребывание заявок в очереди накладываются те или иные ограничения. Эти ограничения могут касаться длины очереди (числа заявок, одновременно находящихся в очереди), времени пребывания заявки в очереди (после какого-то срока пребывания в очереди заявка покидает очередь и уходит не обслуженной), общего времени пребывания заявки в СМО и т. д. Обслуживание заявок в СМО может быть упорядоченное (в порядке поступления), неупорядоченное (в случайном порядке) и с приоритетами (см. рис. 1.4).

Основными характеристиками СМО, определяющими особенности ее функционирования, являются характер входящего потока заявок (т. е. последовательность событий, специальным образом упорядоченных во времени), а также дисциплины ожидания и обслуживания.

Входной поток заявок на обслуживание. Если по отношению к обслуживанию все заявки потока равноправны, то в данный момент времени важен, лишь сам факт наступления события (поступления заявки). Такие потоки называются потоками однородных событий. Каждое событие потока характеризуется моментом времени t_j, в который оно наступает. В случае вполне детерминированного потока однородных событий последовательность t_j можно получить, перечислив их (например, сформировав одномерный массив в памяти компьютера), задав функцию $t_j = f(t)$ или, наконец, использовав

рекуррентные соотношения, позволяющие получить текущее значение t_j по предыдущему.

Для описания случайных потоков однородных событий задается закон распределения, характеризующий последовательность случайных величин $t_1, t_2, ..., t_m, ...$. Обычно t_j целесообразно заменять величинами ξ_j, определяющими длину интервалов времени между моментами поступления заявок t_j:

$$t_1 = \xi_1,$$
$$t_2 = \xi_1 + \xi_2, \quad ...$$
$$t_m = \xi_1 + \xi_2 + ... + \xi_m.$$

Совокупность ξ_j задается совместным законом распределения. Случайный поток однородных событий называют *потоком с ограниченным последействием*, если ξ_j являются независимыми случайными величинами. В этом случае интервал ξ_j может быть задан своей функцией плотности $f_j(z)$. Многочисленные применения имеют стационарные ординарные потоки с ограниченным последействием (потоки Пальма). Поток однородных событий называется стационарным, если вероятность $p_k(t_0,t)$ появления k событий на интервале времени (t_0,t_0+t) не зависит от t_0, а зависит только от t и k (т.е. вероятностный режим потока не зависит от времени). Поток называется *ординарным*, если вероятность $\varphi(t_0,t)$ появления двух и более заявок за промежуток времени (t_0,t_0+t) для любого t_0 мала по сравнению с t:

$$\lim_{t\to 0}\frac{\varphi(t_0,t)}{t}=0.$$

Для стационарных потоков с ограниченным последействием справедливо следующее соотношение: $f_2(z)=f_3(z)=...=f_k(z)=f(z)$, означающее, что при $j>0$ интервалы ξ_j распределены одинаково. Математическое ожидание m_ξ случайной величины ξ_j при $j>0$ равно

$$m_\xi=\int_0^\infty zf(z)dz.$$

Величина m_ξ, по существу, есть средняя длина интервала между последовательными заявками.

Тогда для стационарных потоков с ограниченным последействием величина

$$\lambda=\frac{1}{m_\xi},$$

определяющая среднее количество событий в единицу времени, называется плотностью или интенсивностью потока. Для рассматриваемых потоков справедлива формула Пальма, связывающая $f_1(z_1)$ и $f(z)$:

$$f_1(z_1)=\lambda\left[1-\int_0^{z_1}f(u)du\right] \tag{2.24}$$

Это соотношение позволяет получить функцию плотности $f_1(z_1)$ по известной плотности $f(z)$.

Пример 2.12. Получить функцию плотности $f_1(z_1)$ для потока с равномерным распределением интервалов.

Функция плотности $f(z)$ в этом случае имеет вид

$$f(z) = 1/b, \quad 0 \le z \le b.$$

Поскольку $m_\xi = b/2$, то $\lambda = 2 / b$. Тогда (2.24) будет иметь следующий вид:

$$f_1(z_1) = \frac{2}{b}\left[1 - \int_0^{z_1} \frac{du}{b}\right] = \frac{2}{b}\left(1 - \frac{z_1}{b}\right) = \frac{2(b - z_1)}{b^2} = \lambda\left(1 - \frac{\lambda}{2} z_1\right).$$

Случайный поток однородных событий с ограниченным последействием называется потоком без последействия, если вероятность $p_k(t_0, t)$ поступления k событий за промежуток времени $(t_0, t_0 + t)$ не зависит от чередования событий до момента t_0.

При моделировании СМО важную роль играет простейший (стационарный, ординарный, без последействия) поток, для которого вероятность $p_k(t)$ наступления k событий за интервал времени t выражается законом распределения Пуассона

$$p(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}. \tag{2.25}$$

Отсюда простейший поток иногда называют пуассоновским.

Пример 2.13. Получить функцию плотности $f_1(z_1)$ для простейшего потока.

Из (2.25) следует, что функция плотности $f(z)$

случайной величины ξ_j при $j>0$ в этом случае имеет вид

$$f(z)=\lambda e^{-\lambda z}, z>0 .$$

В соответствии с (2.24) получаем

$$f_1(z_1)=\lambda\left(1-\int_0^{z_1}\lambda e^{-\lambda u}du\right)=\lambda\left(1+\int_0^{z_1}e^{-\lambda u}d(-\lambda u)\right)=\lambda\left(1+e^{-\lambda z_1}-1\right)=\lambda e^{-\lambda z_1} .$$

Таким образом, в отличие от потока Пальма, при пуассоновском потоке $f_1(z_1)=f(z)$.

Кроме рассмотренных потоков, практически интересны потоки Эрланга. В отдельных случаях рассматривают нестационарные входящие потоки.

Дисциплина ожидания. В общем случае СМО имеет n каналов (линий), параллельно обслуживающих входной поток заявок. Каждый канал может быть либо свободен, либо занят. Если в СМО имеются свободные каналы, заявка обслуживается, иначе она ожидает в очереди время τ, после чего получает отказ и покидает систему.

В зависимости от значения τ различают: системы с ожиданием ($\tau=\infty$), системы с отказами ($\tau=0$) и смешанные системы ($0\le\tau\le\hbar$).

В ряде случаев длительность ожидания или вообще возможность встать в очередь зависят от ограничений на длину очереди. Отсюда различают СМО без ограничений по длине очереди и с ограничением по длине очереди.

По характеру «рассасывания» очереди различают СМО с упорядоченной и абсолютно неупорядоченной очередью. В первых –

дисциплина очереди подчиняется правилу «раньше пришел - раньше поступил на обслуживание», в других - любая из заявок, стоящих в очереди, при освобождении канала с равной вероятностью может поступить на обслуживание.

Наконец, если заявки, поступающие в СМО, неравнозначные, т.е. имеют различные приоритеты, говорят о системах без приоритета на обслуживание и с приоритетами. Необходимо отметить, что система приоритетов и соответствующие ей дисциплины ожидания могут быть достаточно сложными.

Дисциплина обслуживания. По числу каналов СМО подразделяются на одно - и многоканальные. Последние, в свою очередь, делятся на системы с равноценными и неравноценными каналами.

Длительность обслуживания является случайной величиной с показательным законом распределения и реже – эрланговским.

По времени обслуживания различают СМО с неограниченной длительностью обслуживания и с ограничением по длительности обслуживания.

Кроме того, часто учитывается надежность каналов обслуживания. В этой связи различают системы с выходом из строя каналов. Наконец, последние подразделяются на СМО без восстановления вышедших из строя каналов обслуживания и СМО с восстановлением каналов.

Моделирование потоков заявок. Пусть в СМО поступает стационарный поток однородных событий с ограниченным последействием (поток типа Пальма).

Поток типа Пальма задается функцией плотности $f(z)$ случайных интервалов $z_j,\ j=2,3,...$ между последовательными моментами $t_j = \quad = t_{j-1} + z_j$ поступления заявок. Для его имитации на компьютере требуется получить необходимое число реализаций, т.е. таких неслучайных последовательностей $t_1^*, t_2^*, \ldots$ моментов поступления заявок в СМО, интервалы между которыми $t_j^* - t_{j-1}^*$ были бы значениями случайной величины z_j, описываемыми функцией плотности $f(z)$.

Рассмотрим процедуру формирования последовательности $t_1^*, t_2^*, \ldots$:

1. Формируем $t_1 = z_1$. Функция плотности $f_1(z_1)$ определяется по
(2.24).
2. Получим случайное число ξ_j с равномерным распределением на
интервале [0,1].
3. Формируем z_1, для чего одним из описанных в § 5.5 методов
преобразуем ξ_j в случайную величину z_1^*, имеющую функцию плотности $f_1(z_1)$.
4. Полагаем $t_1^* = z_1^*$.
5. Формируем $t_2 = t_1 + z_2$. Для чего, получив ξ_{j+1}, преобразуем
его в z_2^*, имеющее функцию плотности $f(z)$.

6. Полагаем $t_2^* = t_1^* + z_2^*$.

7. Очередное t_j^* вычисляется по формуле $t_j^* = t_{j-1}^* + z_j^*$.

Пример 2.14. Получить последовательность моментов поступления заявок t_j^* для простейшего входного потока.

На основании вывода в примере 2.13 было получено соотношение $f_1(z_1) = f(z)$. Отсюда для моделирования реализаций простейшего потока необходимо иметь генератор случайных величин, распределенных по закону Пуассона (см. пример 2.9): η_1, η_2, … , η_k. Тогда последовательность моментов поступления заявок будет иметь вид:

$$t_1^* = \eta_1; \quad t_2^* = \eta_1 + \eta_2; \ \ldots; \quad t_k^* = \eta_1 + \eta_2 + \ldots + \eta_k. \qquad (2.26)$$

Пример 2.15. Получить последовательность моментов поступления заявок t_j^* для потока с равномерным распределением z_j.

Решение будем основывать на результатах примера 2.12.

Формирование z_j, $j = 2, 3, \ldots$ сводится к получению ξ_i, равномерно распределенных на интервале $[0, b]$ и соответствующих

$$f(z) = \begin{cases} 1/b, & 0 < z < b; \\ 0, & z <= 0, \quad z >= b. \end{cases}$$

Для этого достаточно ξ_i, равномерно распределенные на интервале [0, 1], привести к интервалу $[0, b]$, т.е.

$$z_i = b\ \xi_i.$$

Тогда последовательность моментов поступления заявок равномерного потока задается системой уравнений (2.26).

Для некоторых других потоков интеграл в формуле Пальма не может быть получен в конечном виде и необходимо использовать специальные численные методы [6].

6.2. Обработка результатов моделирования СМО

При изучении начального, нестационарного периода функционирования СМО моделирование представляет, по существу, разыгрывание множества реализаций процесса, а нужные характеристики (вероятность отказа в обслуживании, среднее число заявок в очереди) получаются обработкой эмпирических данных как статистическое среднее по множеству реализаций.

При моделировании стационарных процессов можно воспользоваться только одной, но достаточно длинной реализацией, а интересующие характеристики рассчитать как средние по времени.

Допустим имеется двухканальная ($n = 2$) СМО с очередью, максимальная длина которой три заявки. Если все три места в очереди заняты, вновь поступившая заявка получает отказ и покидает систему. Поток заявок - пальмовский. Время обслуживания одной заявки – случайная величина, распределенная по показательному закону.

Определим по одной, но достаточно длинной реализации следующие характеристики: вероятность занятости 0, 1, 2 каналов – p_0, p_1, p_2; вероятности того, что в очереди будет 0, 1, 2, 3 заявки – $\overline{p_0}$, $\overline{p_1}$, $\overline{p_2}$, $\overline{p_3}$; среднее число занятых каналов – $\bar{z}$; среднее время ожидания заявки в очереди – $\overline{t_{ож}}$; среднее время обслуживания заявок – $\overline{t_{об}}$; вероятность отказа в обслуживании – $p_{отк}$.

Временная диаграмма функционирования СМО изображена на рис. 2.10. Вверху дана ось времени (0) с отмеченными моментами поступления заявок (t_1, …, t_{14}). На осях 1 и 2 показаны состояния 1-го и 2-го каналов соответственно (тонкая линия – «свободен», жирная – «занят»). На осях 3 – 5 фиксируются состояния 1 – 3 мест в очереди соответственно.

Перед появлением первой заявки t_1 все каналы и места в очереди свободны. В момент t_1 поступает заявка и занимает первый канал. Время обслуживания τ_1 формируется в результате работы генератора, допустим, из примера 2.8. В момент t_2 прихода следующей заявки первый канал занят и она поступает на второй.

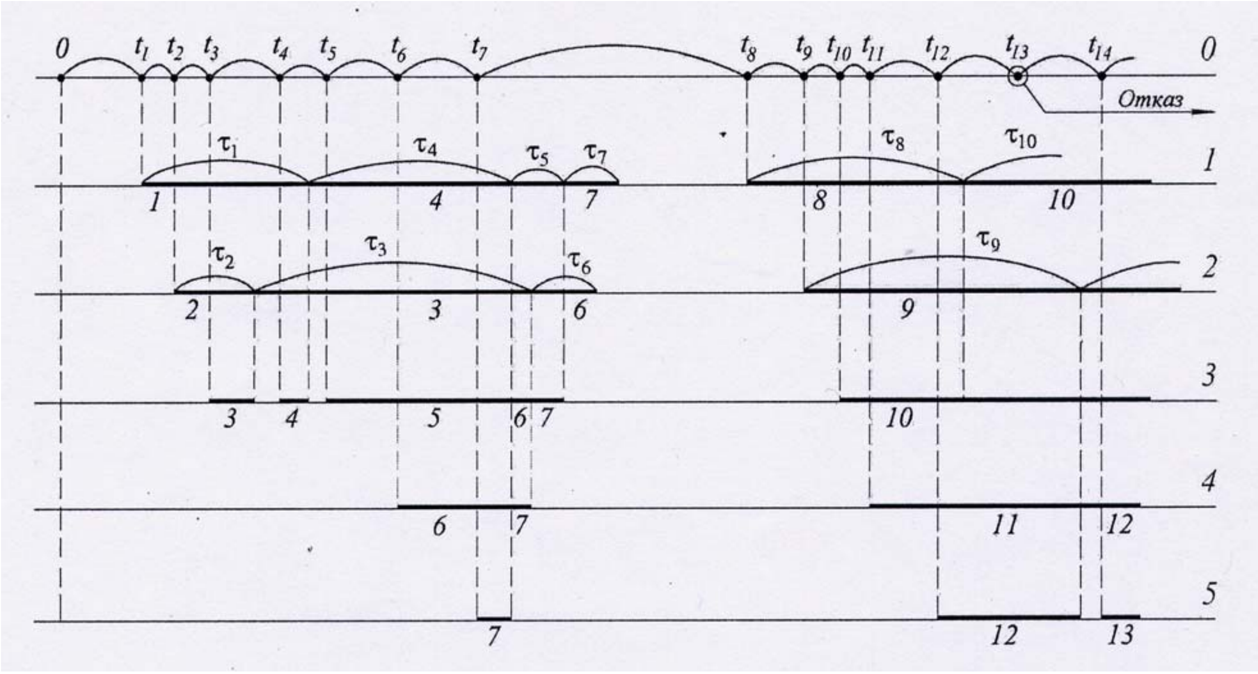


Рис. 2.10. Временная диаграмма функционирования двухканальной СМО с отказами

Генерируется еще одно значение времени обслуживания τ_2. Заявка t_3 приходит в систему, когда оба канала заняты, поступает в очередь и ждет освобождения одного из каналов. Раньше заканчивает обслуживание канал 2 и третья заявка начинает им обрабатываться (время обслуживания τ_3 получается аналогично τ_1 и τ_2). Далее процесс повторяется. Для наглядности на рис. 2.10 под каждым участком занятости канала проставлен номер заявки, что позволяет проследить ее «путь» в СМО.

Обработаем результат моделирования. В первую очередь найдем p_0, p_1, p_2. Для этого разделим временную ось на участки и обозначим их цифрой 0, если не занят ни один канал, 1 – если занят один и 2 – если заняты два канала.

Выберем большой участок времени моделирования T (рекомендуется брать не с самого начала процесса, где сказывается влияние начальных условий), на котором просуммируем длины участков, помеченных нулем, единицей и

двойкой. Очевидно, что $T_0 + T_1 + T_2 = T$.
Тогда $p_0 \approx T_0 / T$; $p_1 \approx T_1 / T$; $p_2 \approx T_2 / T$.

Аналогично поступим для определения характеристик $\overline{p}_0, \overline{p}_1, \overline{p}_2, \overline{p}_3$:

$$\overline{p}_0 \approx \frac{\overline{T}_0}{T}, \quad \overline{p}_1 \approx \frac{\overline{T}_1}{T}, \quad \overline{p}_2 \approx \frac{\overline{T}_2}{T}, \quad \overline{p}_3 \approx \frac{\overline{T}_3}{T},$$

где $\overline{T}_i$ – сумма длин участков, помеченных i-й цифрой.

Среднее число занятых каналов $\overline{z}$ определится как

$$\overline{z} = 1p_1 + 2p_2.$$

Среднее время ожидания заявки в очереди $\overline{t}_{ож}$ определится так: выберем большой участок T и рассмотрим заявки, поступившие в моменты t_k, t_{k+1}, …, t_{k+i}, …, t_{k+N}, для каждой из которых по эмпири-ческим данным подсчитаем время ожидания в очереди $t_{ож}^{k+i}$, равное нулю, если $k+i$ - я заявка была сразу обслужена (или получила отказ), или сумме времен ожидания этой заявки на осях 3 – 5. Тогда

$$\overline{t}_{ож} = \frac{1}{N} \sum_{i=0}^{N} t_{ож}^{k+i}.$$

Среднее время обслуживания заявки по аналогии с $\overline{t}_{ож}$ можно найти по формуле

$$\overline{t}_{обс} = \frac{1}{N}\sum_{i=0}^{N} t_{обс}^{k+i}.$$

где $t_{обс}^{k+i}$ – время обслуживания $(k + i)$-й заявки.

Вероятность отказа можно рассчитать по формуле

$$p_{отк} = \frac{N^*}{N},$$

где N^* – число заявок, получивших отказ;

N – общее число заявок за время T.

6.3. Простейшие модели одноканальной и многоканальной СМО

Как уже было сказано, из всего многообразия имитационных моделей СМО будут рассматриваться только дискретно-событийные модели. Это такие модели, которые развиваются во времени, а состояния системы изменяются мгновенно в конкретные моменты времени.

При моделировании исследователь имеет дело с различными представлениями времени. Во-первых, требуется учитывать реальное время, в котором функционирует имитируемая система. Во-вторых, работа модели осуществляется в определённом масштабе – это модельное (системное) время. И наконец, в-третьих, – затраты машинного времени (времени работы компьютера) на реализацию самой имитации. Первые два измерения времени имеют важнейшее значение при построении модели. С учетом этих временных факторов реализуются переходы системы из состояния в состояние, осуществляется синхронизация функционирования элементов системы, задаётся масштаб времени функционирования («жизни») имитируемой системы, осуществляется управление технологическим процессом модельного эксперимента.

Если не учитывать различные варианты и

тонкости подходов к реализации времени при имитационном моделировании, можно назвать всего три метода реализации моделей – ***принцип Δt*** (постоянный шаг изменения времени), ***принцип особых состояний*** и ***принцип последовательной проводки заявок*** [7, 12, 13].

При использовании первого принципа в моделирующем алгоритме определяются последовательности состояний системы через некоторые фиксированные интервалы времени Δt. Этот подход эффективен, когда число событий в моделируемой системе велико, интервалы между моментами их появления близки и события появляются регулярно (их распределение во времени достаточно равномерно) и, что самое главное, невозможно заранее определить моменты совершения событий.

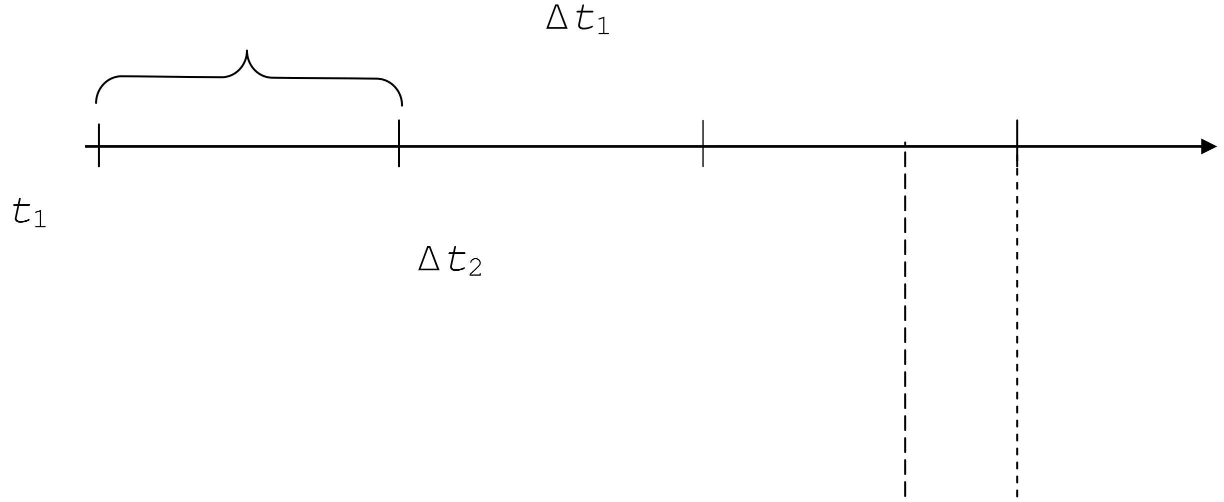
Использование принципа Δt часто малоэффективно. Так при малых шагах (малых Δt) тратится компьютерное время на бесполезное фиксирование множества не особых состояний. А при увеличении Δt можно пропустить важные особые состояния, что приведёт к получению недостоверных результатов (см. рис. 2.11). На этом рисунке:

t – ось реального времени,

t_1 или t_2 – оси модельного времени при шаге Δt_1 или Δt_2,

T – реальный момент наступления события,

T_1 или T_2 – фиксируемые моменты наступления события, соответственно для шага Δt_1 или Δt_2.



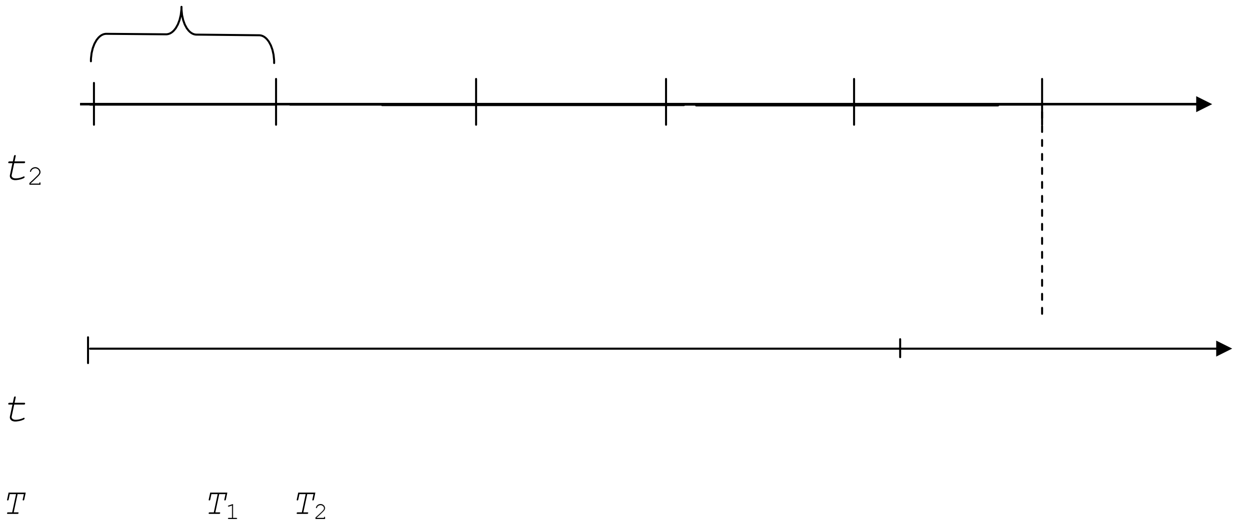


Рис. 2.11. Пример использования различных величин шага Δt

Пример рис. 2.11 показывает, что погрешность фиксации момента наступления событий слишком велика и для шага Δt_1 равна $T_1 - T$, а для шага Δt_2 – $T_2 - T$. Для выхода из создавшегося положения, когда получаются очень грубые результаты, используют возврат к предыдущему моменту времени для осуществления повторного прогона грубого участка с меньшим шагом Δt.

Принцип особых состояний отличается от принципа Δt тем, что включает процедуру определения момента времени, соответствующего следующему особому состоянию по известным характеристикам данного или предыдущего состояний.

Принцип последовательной проводки заявок получил широкое распространение именно при моделировании функционирования СМО. Осуществляется формирование момента поступления заявки в систему, и затем эта заявка «протягивается» по всем этапам обслуживания и так поступают с каждой заявкой. Этот принцип положен в основу построения всех моделирующих алгоритмов СМО, рассмотренных в этой главе. Следует отметить также, что принцип последовательной проводки заявок в сочетании с принципом Δt лежит в основе всех языков моделирования (см. гл. 7).

В этом параграфе сначала рассматривается имитационная модель одноканальной СМО с ограниченным временем ожидания заявок начала обслуживания. Описание этой модели, а также моделей в последующих параграфах, и изложение принципов их работы осуществляется в предположении, что читатель ознакомился с методикой моделирования потоков событий, изложенной в параграфе 6.1 данного учебного пособия, или в разделах 5.1 и 5.2 монографии [7], или разделе 4.4 учебника [41].

Процесс функционирования системы рассматривается за период времени $[0, T]$. На вход в случайные моменты времени поступают заявки. Если в момент поступления очередной заявки канал свободен, то заявка принимается к немедленному обслуживанию. В противном случае заявка становится в очередь на обслуживание. Время пребывания заявки в очереди ограничено. Длительность обслуживания заявок также является случайной величиной. Показателями эффективности СМО можно считать величины вероятности того, что заявка, пришедшая в систему, будет обслужена, или производительности рассматриваемой СМО.

Суть подхода к имитации работы СМО проста – случайный процесс работы системы в интервале $[0, T]$ рассматривается как детерминированный, но со случайно выбранными параметрами. Для обеспечения статистической устойчивости искомых результатов процесс моделирования повторяется заданное количество раз $N1$ (см. § 5.6). Показатели эффективности формируются на основе статистической обработки результатов отдельных реализаций процесса моделирования.

Итак, рассматривается одноканальная СМО, в которую поступают заявки, образующие ординарный поток однородных событий с заданным законом распределения $f(t)$ интервалов между моментами появления заявок. Заявки поступают в случайные моменты времени t_j, они обслуживаются в течение времени τ^*_j, и обслуживание заканчивается в момент t^*_j. Случайная величина длительности обслуживания τ^* определяется в соответствии с заданным законом распределения $f(\tau^*)$.

Судьба последующей заявки зависит (рис. 2.12, а) от момента ее появления t_j относительно момента окончания обслуживания предыдущей t^*_{j-1}. Так, если бы рассматривалась СМО с отказами и очередная заявка поступила бы раньше наступления момента окончания обслуживания предыдущей, т.е. $t_j < t^*_{j-1}$, то такая заявка получила бы отказ (рис. 2.12, б), если же $t_j >= t^*_{j-1}$, то заявка была бы обслужена (см. рис. 2.12, а).

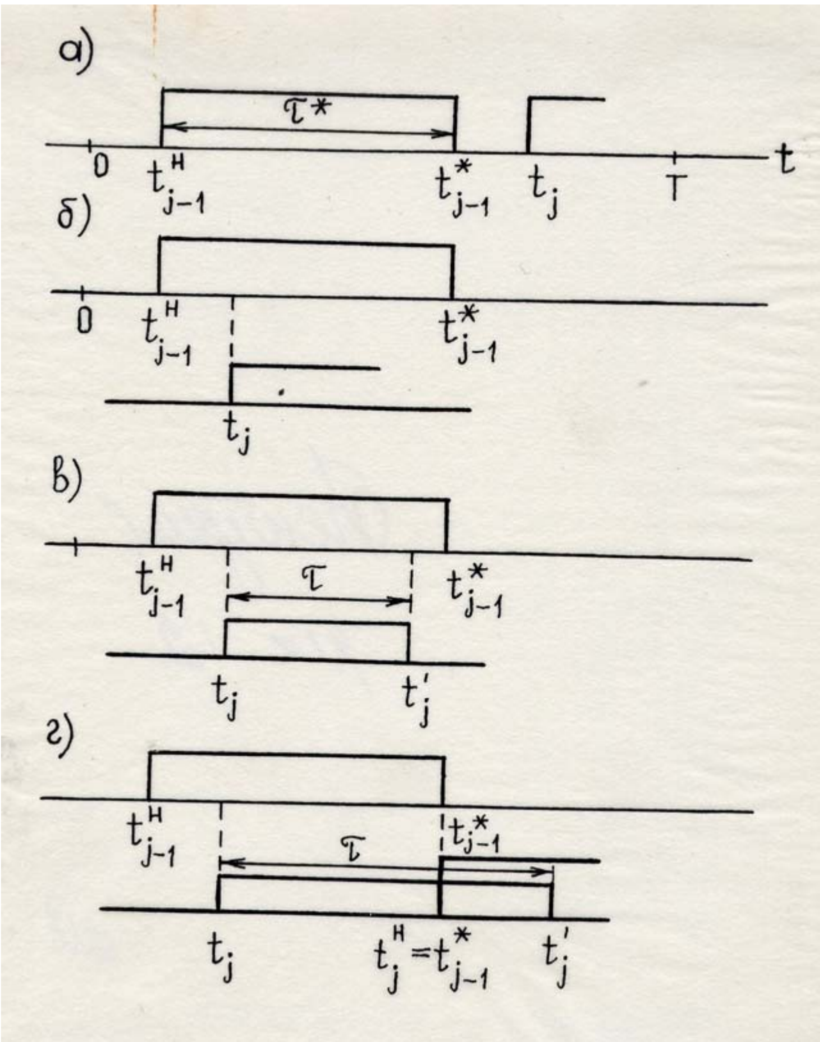


Рис. 2.12. Детали временной диаграммы

В связи с тем, что в рассматриваемой СМО должно учитываться ограничение времени возможного ожидания, необходимо формировать τ_j – случайную величину длительности ожидания j-й заявки начала обслуживания с соответствующим законом распределения $f(\tau)$. Когда $t_j < t^*_{j-1}$, то j-я заявка получает отказ (рис. 2.12, в), если $t'_j < t^*_{j-1}$, где $t'_j = t_j + \tau_j$ – момент окончания ожидания j-й заявки начала обслуживания. Если же $t'_j >= t^*_{j-1}$ (рис. 2.12, г), то обслуживание j-й заявки начинается в момент времени $t^н_j = t^*_{j-1}$.

Исходные данные для моделирования: законы распределения $f(t)$, $f(\tau)$ и $f(\tau^*)$, величины T и

$N1$, а также состояние системы при $t = 0$, например, $t^*_{j-1} = t_j = 0$, $m = n = N = 0$. Считается, что заявки, для которых момент появления t_j **>=** T, в систему не попадают и не обслуживаются. Заявки, для которых время окончания обслуживания $t^*_j > T$, считаются получившими отказ, а для которых $t^*_j <= T$, являются обслуженными.

Фрагмент блок-схемы моделирующего алгоритма одноканальной СМО представлен на рис. 2.13, а ниже дается описание большинства используемых блоков:

1 – формирование очередного момента t_j поступления заявки в систему;

2 – проверка условия $t_j<T$ принадлежности момента поступления очередной заявки t_j интервалу исследования системы $[0,T]$;

3 – проверка условия $t_j < t^*_{j-1}$, где t^*_{j-1} - момент окончания обслуживания предыдущей заявки;

4 – формирование случайного значения длительности ожидания τ_j в соответствии с законом распределения $f(\tau)$;

5 – вычисление момента $t'_j = t_j + \tau_j$ (в момент t'_j заявка покидает систему, если она не будет принята к обслуживанию);

6 – проверка условия $t'_j < t^*_{j-1}$ (заявка покидает систему раньше, чем освободится канал);

7 – выбор в качестве момента начала обслуживания j-й заявки момента окончания обслуживания $(j -1)$-й заявки $t^н_j = t^*_{j-1}$;

8 – выбор в качестве момента начала обслуживания j-й заявки момента ее поступления в систему $t^н_j = t_j$;

9 – формирование длительности обслуживания τ^*_j заявки (времени занятости канала) в соответствии с законом $f(\tau^*)$;

10 – вычисление момента окончания обслуживания j-й заявки (момента освобождения канала) $t^*_j = t^{н}_j + \tau^*_j$;

11 – проверка условия $t^*_j <= T$;

12 – подсчет количества обслуженных заявок m;

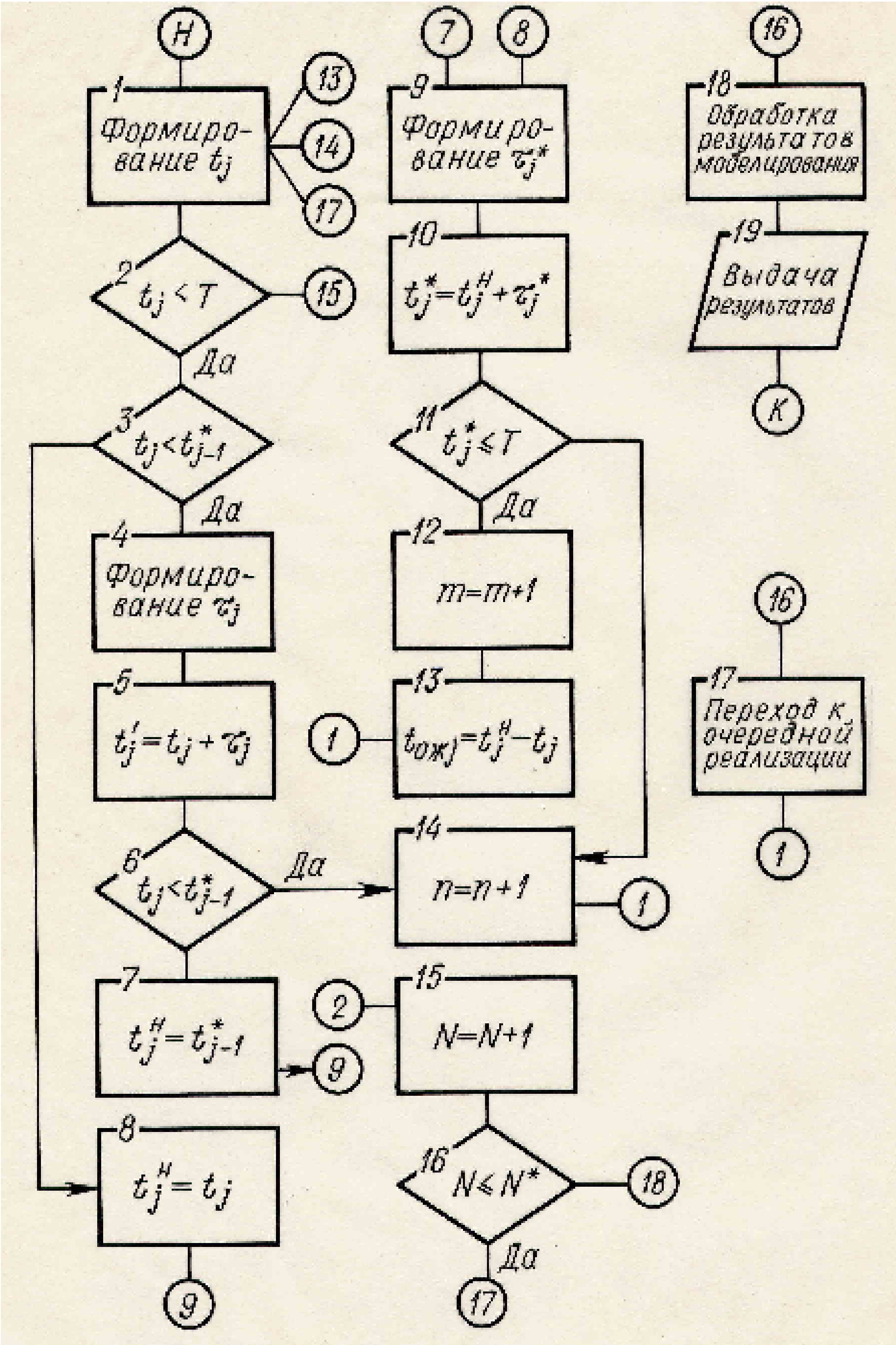


Рис. 2.13. Модель одноканальной СМО

13 – вычисление длительности пребывания в очереди j-й заявки, $t_{ожj} = t^{н}_{j} - t_{j}$;

14 – подсчет количества заявок n, получивших отказ;

15 – подсчет количества реализаций N;

16 – проверка условия $N < N1$, где $N1$ - количество реализаций, необходимое для обеспечения заданной точности.

Как превратить описанную модель СМО в модель с отказами, в которой заявка, заставшая канал занятым, получает отказ? Для этого нужно в модели (см. рис. 2.13) ликвидировать блоки **4 ...7** и из блока **3** по признаку «Да» переходить в блок **14**.

Как превратить описанную модель СМО в модель с неограни-ченным ожиданием заявки начала обслуживания? А для этого нужно в модели ликвидировать блоки **4 ...6** и из блока **3** по признаку «Да» переходить в блок **7.**

Пусть многоканальная СМО имеет K каналов, вполне идентичных (с точки зрения длительности обслуживания) и равноправных (в смысле возможности использования для обслуживания данной заявки любого свободного канала). В систему по-прежнему поступает ординарный поток заявок с заданным законом распределения $f(t)$. Время возможного ожидания до начала обслуживания τ – случайная величина с законом распределения $f(\tau)$. Длительность обслуживания τ^* имеет закон распределения $f(\tau^*)$. Процесс функционирования системы рассматривается в интервале времени $[0, T]$.

Заявки принимаются к обслуживанию в порядке очереди (в порядке поступления в систему). Если для обслуживания данной заявки имеется несколько свободных каналов, то каналы привлекаются к обслуживанию в порядке очереди (в первую очередь привлекается тот канал, который ранее других освободился от обслуживания).

Описанная здесь система близка по характеру к одноканальной СМО. Поэтому многие блоки могут быть использованы для построения моделирующего алгоритма в этом случае. Здесь также сохраняются все использованные выше обозначения.

Блок-схема алгоритма многоканальной СМО представлена на рис. 2.14. В ней дополнительно использованы следующие блоки:

3 – проверка условия $t_j < \min t^*_{lk}$,

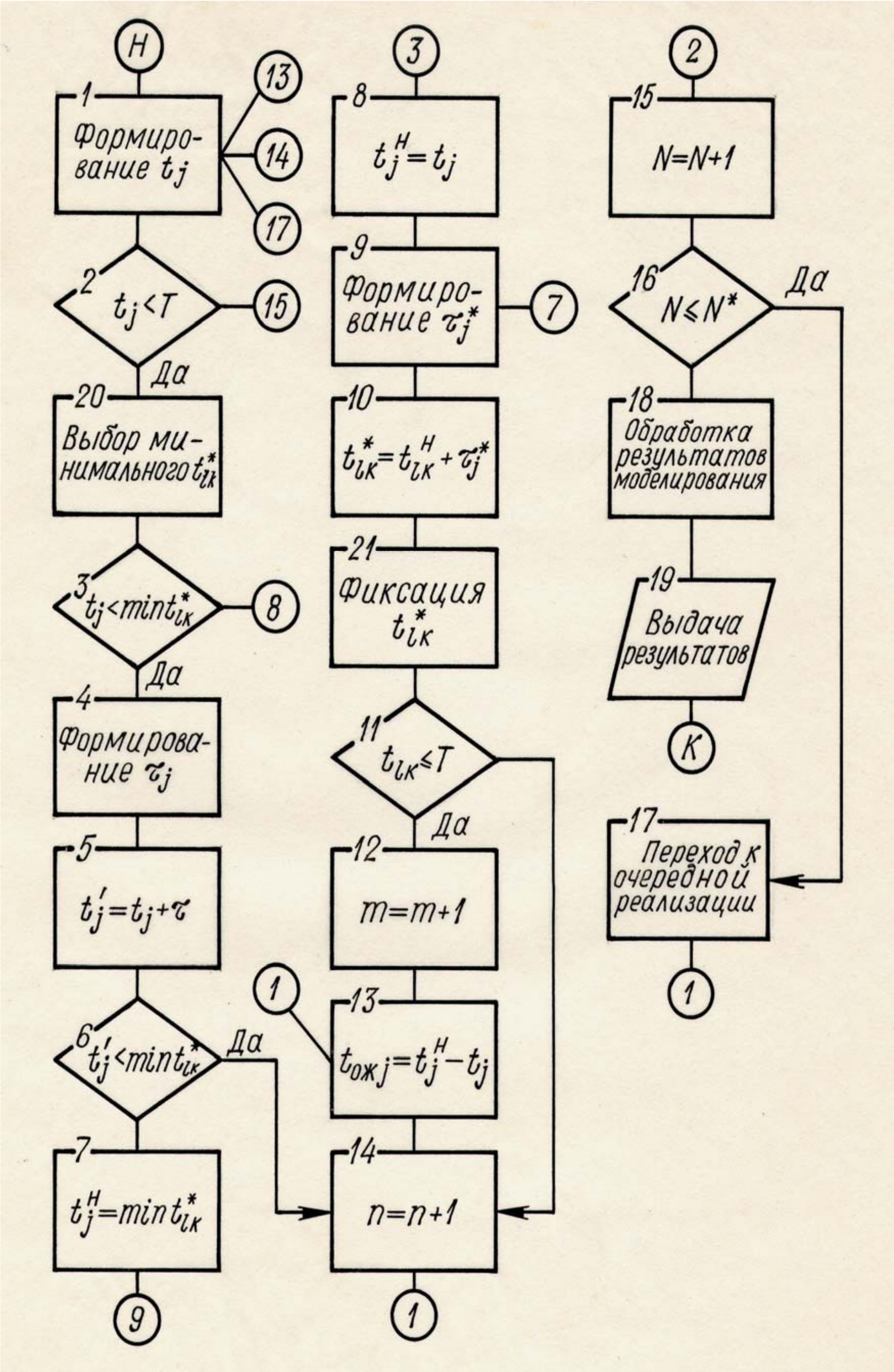


Рис. 2.14. Модель многоканальной СМО

6 – проверка $t'_j < \min t^*_{lk}$,

7 – выбор в качестве момента начала обслуживания $t^н_j$, величины $\min t^*_{lk}$,

10 – вычисление t^*_{lk} - момента окончания обслуживания l-й заявки k-м каналом (момента освобождения k-го канала, $k = \overline{1,K}$),

$$t^*_{lk} = t^н_{lk} + \tau^*_j,$$

где $t^н_{lk}$ - момент начала обслуживания l-й заявки k-м каналом.

6.4. Модификации моделей СМО

Встречающиеся на практике разновидности СМО не исчерпываются рассмотренными выше простейшими моделями. Можно выделить особенности моделирования некоторых других классов СМО. Приводимые ниже фрагменты алгоритмов следует разбирать совместно с простейшими моделями.

Системы с ограничением по длине очереди. В моделируемой СМО может устанавливаться ограничение по длине очереди. При этом в момент поступления заявки необходимо проверять условие $l' < l$, где l – количество мест в очереди, l' – количество занятых мест в очереди в момент t_j. Если проверяемое условие выполняется, то заявка ставится в очередь и ждет освобождения канала, а если не выполняется, то получает отказ. Такая проверка осуществляется только в том случае, если все каналы заняты. А если хотя бы один канал свободен, это свидетельствует о том, что очереди нет и проверка не нужна, а заявка идет сразу на обслуживание.

Системы с ограничениями по длительности обслуживания и длительности пребывания. Для обеспечения моделирования этих ограничений в модель вводятся и реализуются соответствующие проверки. Так, при ограничении по длительности обслуживания проверяется условие $\tau^*_j < \tau_1$, где τ^*_j – расчетная длительность обслуживания j-й заявки, а τ_1 - предельная длительность обслуживания заявок. При выполнении этого условия обслуживание такой заявки продолжается, а при невыполнении обслуживание прерывается и заявка получает отказ.

Для реализации ограничения по длительности пребывания заявки в СМО необходимо проверять условие $\tau_{jсист} <= \tau_2$, где $\tau_{jсист} = t^*_j - t_j$ - длительность пребывания заявки в системе, а τ_2 - предельная длительность пребывания заявок в СМО. При выполнении этого условия обслуживание j-й заявки продолжается, а при невыполнении заявка получает отказ.

Модель СМО с предпочтительным обслуживанием заявок, ранее других покидающих систему. В рассмотренных примерах одно- и многоканальной СМО заявки обслуживаются в порядке поступления. Очень часто заявки, имеющие право (в порядке очереди) на немедленное обслуживание, могут пребывать в системе довольно долго и обслуживаться после заявок с малым τ и быстро уходящих из системы.

Рассмотрим модификацию алгоритма многоканальной СМО (см. рис. 2.14). Блок-схема алгоритма, изображенная на рис. 2.15, позволяет ввести приоритет для заявок, ранее других покидающих систему.

Если очередь состоит из k заявок, то необходимо найти такую, которая имеет наименьший момент окончания ожидания $t'_{\min}$. После определения $t'_{\min}$ управление передается блоку **3**, устанавливающему возможность обслуживания этой заявки. При невыполнении условия управление передается оператору, формирующему $t^{н}$ –

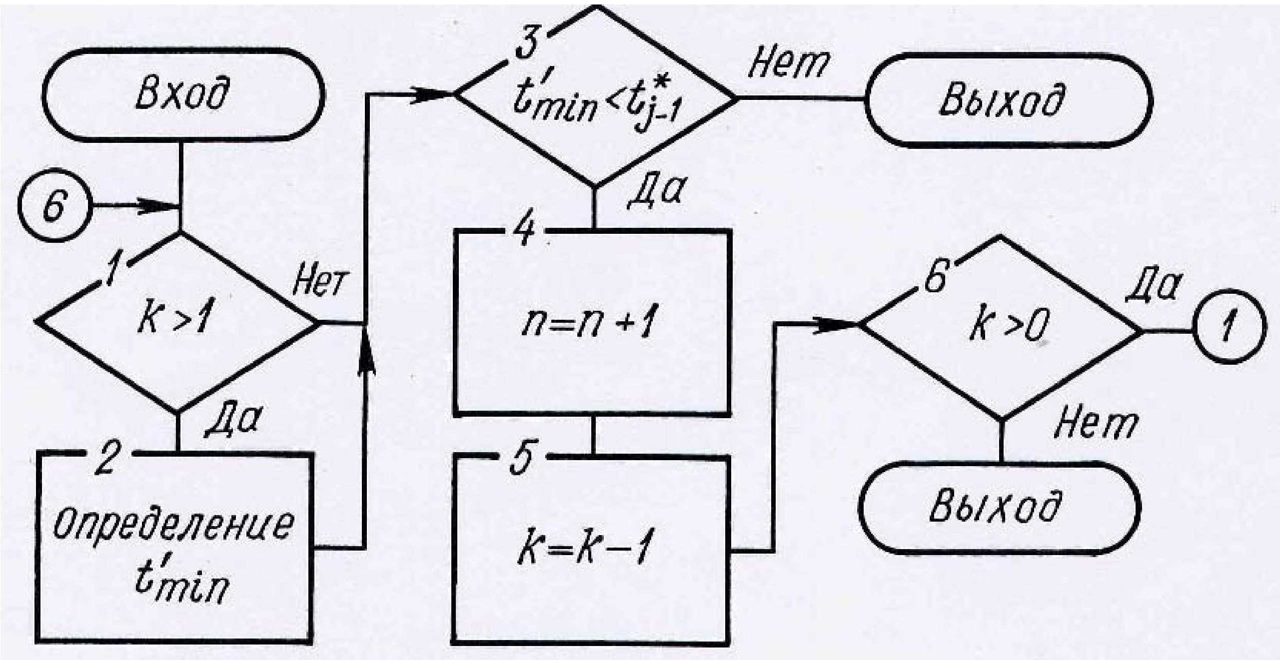


Рис. 2.15. Фрагмент модели СМО с предпочтительным обслуживанием заявок

момент начала обслуживания в основном алгоритме. В случае отказа (т. е. заявка обслужена не будет) вычисляется число отказов n, а очередь уменьшается на единицу: $k - 1$. Если очередь еще не исчерпана, т.е. $k > 0$, процедура повторяется, иначе управление передается оператору основного алгоритма, формирующему t_{j+1} – момент прихода очередной заявки.

Модель СМО со случайным порядком обслуживания. На рис. 2.16 изображена блок-схема моделирующего алгоритма.

Группа операторов **1** … **7** выбирает заявки, которые могут быть обслужены, т. е. для них $t'_j > t^*_{j-1}$. Моменты времени t'_j формируют массив, из которого затем в блоке **8** осуществляется выбор заявки. После просмотра всех заявок, стоящих в очереди, блок **6** передает управление блоку **7** и, если массив t'_j не пуст ($a > 0$), то блок **8** выбирает заявки в соответствии с заданным законом распределения. Для этой заявки блок **9** формирует время начала обслуживания. Если регистр пуст (т.е. ни одна заявка из очереди не может быть обслужена), то управление от блока **7** передается в основной алгоритм оператору, формирующему время поступления очередной заявки.

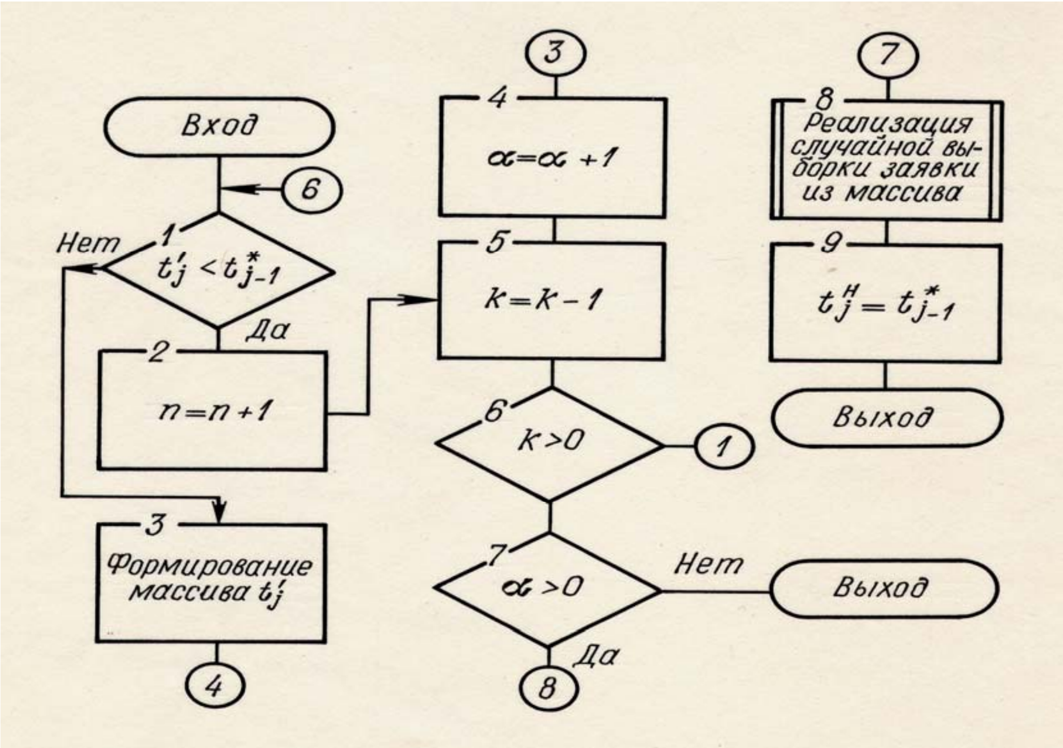


Рис. 2.16. Фрагмент модели СМО со случайным порядком обслуживания

Если соответствующим образом изменить случайную схему выбора заявки (блок **8**), например, жесткой детерминированной системой приоритетов, то приходим к очередной модификации СМО

Модель СМО с заданным порядком выбора обслуживающих каналов. На рис. 2.17 изображена блок-схема моделирующего алгоритма.

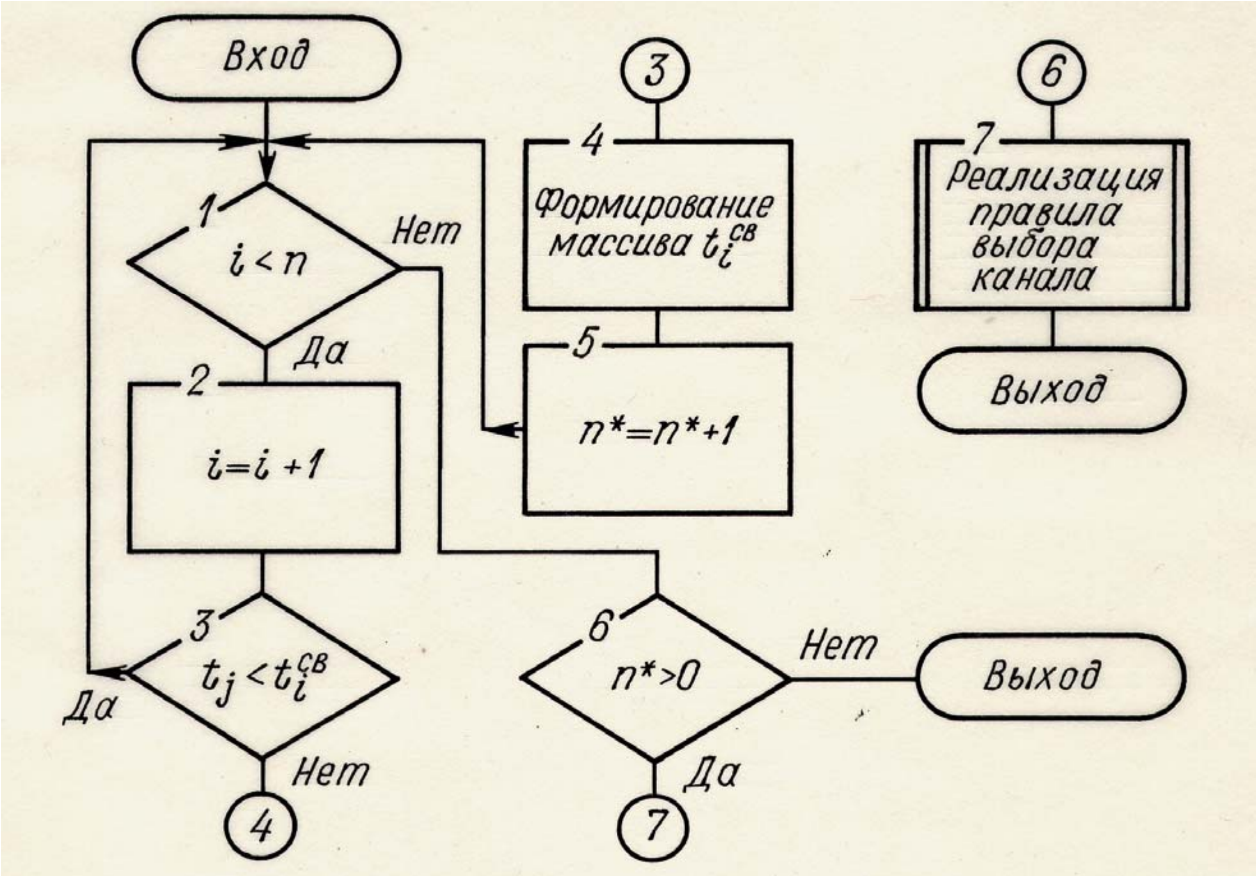


Рис. 2.17. Фрагмент модели СМО с заданным порядком выбора обслуживающих каналов

Для каждой заявки, поступившей в t_j момент времени, просматриваются все каналы и определяются свободные. Проверяется условие $i < n$ (i – текущее, n – число каналов в СМО). Занятость канала проверяется блоком **3**. Незанятые каналы сформируют массив $t_i^{св}$ (блок **4**), а их количество накапливается в счетчике $n^* = = n^* + 1$. После того как свободные к моменту t_j каналы отобраны, управление передается блоку **7**, реализующему правило выбора канала. Если $n^* = 0$, т. е. нет ни одного свободного к моменту t_j канала, переходим в основной алгоритм к блоку, формирующему новые значения t_j, и повторяем отбор каналов для момента t_{j+1}.

Модель СМО с выходом из строя каналов. Особенностью этого класса СМО является возможность сбоя в работе канала. Канал может быть восстановлен за время $\tau^р$ и вступить в строй в момент $t = t^{сб} + \tau^р$, где $t^{сб}$ – момент сбоя. Надежность канала задается функцией распределения интервала безотказной работы $P(t)$, а $\tau^р$ определяется как случайная величина с некоторым законом распределения.

Сбой и ремонт канала можно интерпретировать как обслуживание фиктивной заявки с моментом начала обслуживания $t^{сб}$, длительностью обслуживания $\tau^р$ и первым приоритетом. Если заявка приходит в момент сбоя, то она получает отказ или может быть обслужена после ремонта канала.

Рассмотрим этот класс СМО на примере одноканальной системы со следующими свойствами: канал выходит из строя в моменты $t_j^{сб}$, определяемые законом $P(t)$; канал ремонтируется случайное время $\tau_j^р$, заданное законом распределения $f(\tau^р)$; заявка, при обслуживании которой произошел сбой, получает отказ.

На рис. 2.18 изображена блок-схема моделирующего алгоритма. Блок **1** проверяет условие $\gamma>0$, т. е. определяет сформировано и занесено ли в массив значение $t^{сб}$. Если это так, то управление передается блоку **3**, иначе в блоке **2** формируется $t^{сб}$. Далее проверяется условие $t^{сб} < t^{св}$. Если оно не выполнено, т. е. сбой произошел после окончания обслуживания, то заявка может быть обслужена. В этом случае $t^{сб}$ записывается в массив, а $\gamma = 1$.

Если же $t^{сб} > t^{св}$, управление передается блоку **6**. Если $t^{сб} <= t_j^н$ (сбой произошел после начала обслуживания), то заявка получает отказ и управление передается в основной алгоритм блоку, фиксирующему количество отказов. Иначе, т.е. если сбой произойдет до начала обслуживания, управление передается блоку **8**, где формируется время ремонта, и далее блоку **9**, где вычисляется момент окончания ремонта. Теперь, если $t^p < t_j'$, т. е. ремонт закончится раньше, чем истечет время ожидания, управление передается в основной алгоритм блокам, занимающимся обслужива-нием заявки, иначе – блоку, подсчитывающему отказы.

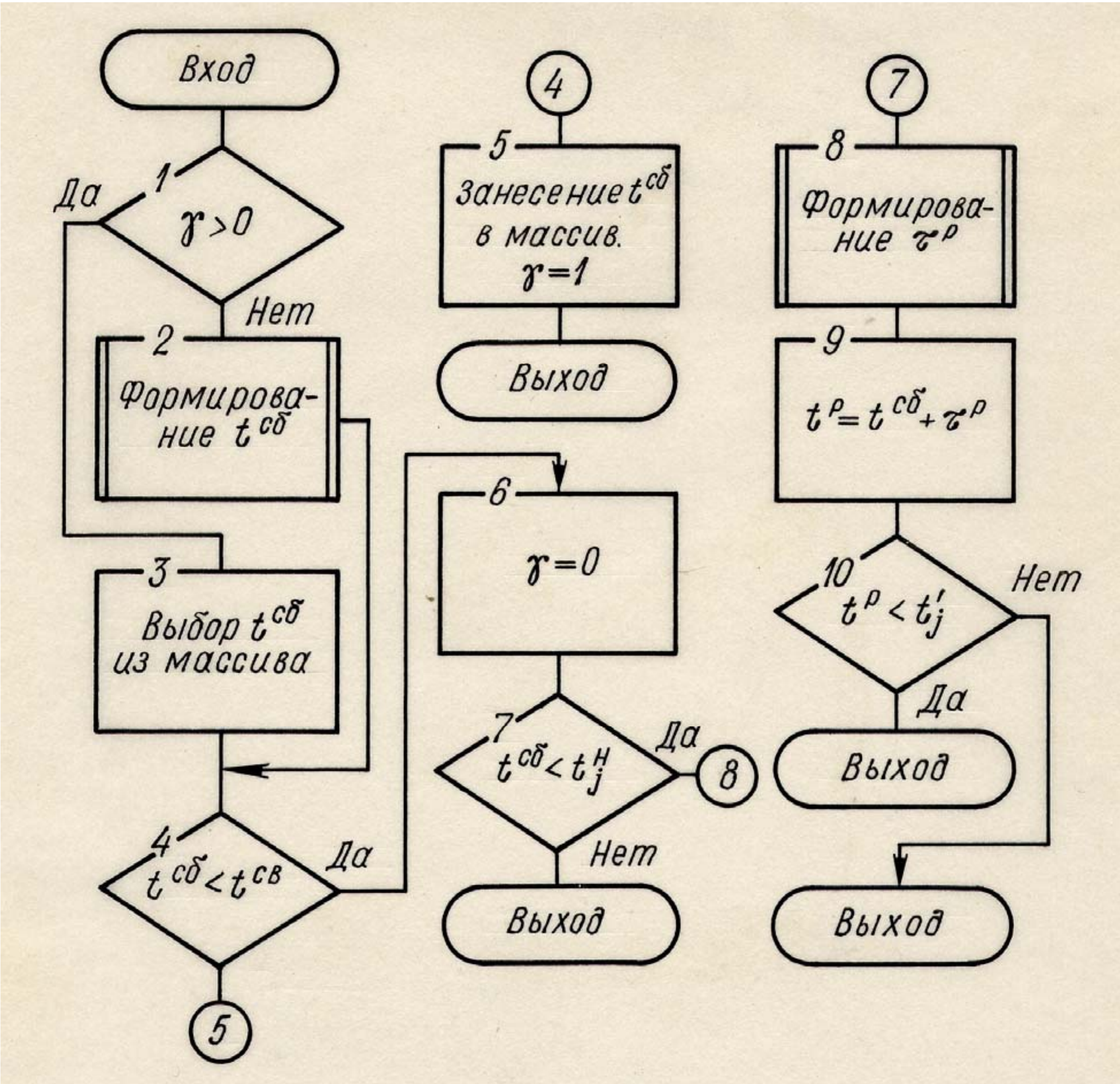


Рис. 2.18. Фрагмент модели СМО с выходом из строя каналов

6.5. Модель многоканальной СМО с ограниченной длиной очереди и ограниченным временем ожидания

заявок начала обслуживания

Алгоритм многоканальной СМО (см. рис. 2.14) дает только общее представление о работе СМО подобного типа. В этом алгоритме нет конкретного количества каналов, определяется в общем виде t^*_{lk} – время окончания обслуживания l-й заявки на каком-то k-м канале и т. д. Поэтому для получения конкретного представления о сложности построения имитационной модели ниже описывается модель многоканальной СМО (рис. 2.19) (количество каналов $I1$) с очередью (длина очереди ограничена и равна $Z1$) и ограниченным временем пребывания заявок в очереди.

В модели приняты следующие обозначения:

T – время моделирования,

$T1$ – момент появления очередной заявки,

$T2$ – расчетная длительность ожидания заявки начала обслуживания,

$T3$ – момент окончания ожидания,

$T4$ – действительная длительность ожидания заявки начала обслуживания,

$T5$ – суммарная длительность ожидания заявок начала обслужи-вания,

$T6$ – длительность обслуживания заявки,

$T7$ – суммарная длительность обслуживания заявок,

I - номер канала,

$I1$ - количество каналов в системе,

$I2$ – количество занятых каналов,

$I3$ – номер свободного канала,

F – минимальное время освобождения канала с номером $I3$,

Z – номер места в очереди,

$Z1$ – количество мест в очереди,

$Z2$ – количество занятых мест в очереди,

M – количество обслуженных заявок,

$M1$ – количество заявок, получивших отказ,

$M2$ – количество заявок, ожидавших начало обслуживания,

N – количество реализаций процесса моделирования,

$N1$ – заданное количество реализаций.

Для обеспечения работы модели в различных ситуациях, возникающих в процессе моделирования, введены следующие признаки:

$X = 1$ – признак того, что очереди нет, не все каналы заняты и нужно сразу ставить заявку на обслуживание на тот канал, который раньше всех освободится от обслуживания предыдущих заявок, в противном случае $X = 0$;

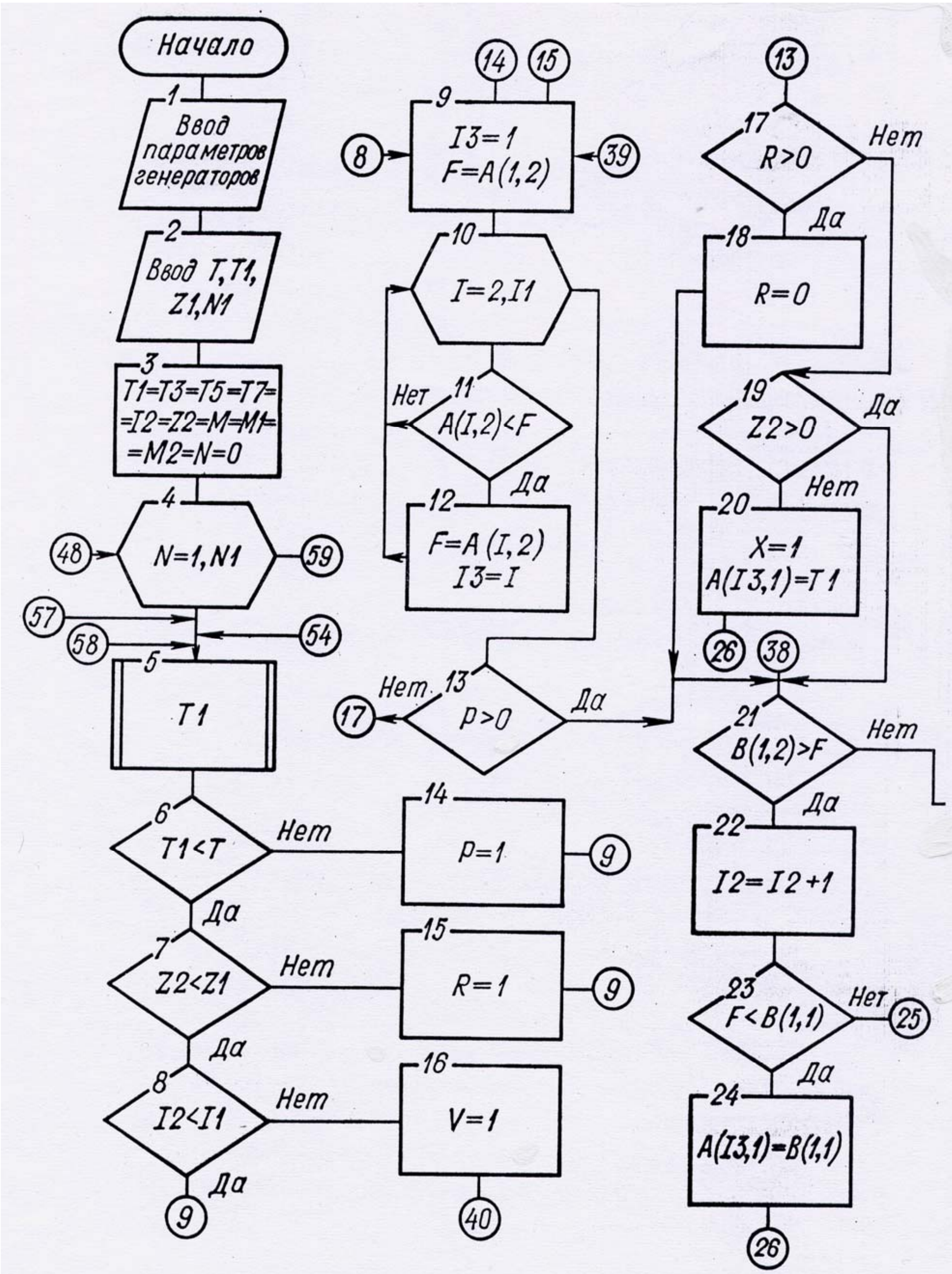


Рис. 2.19. Модель многоканальной СМО (см. также с. 151 и 152)

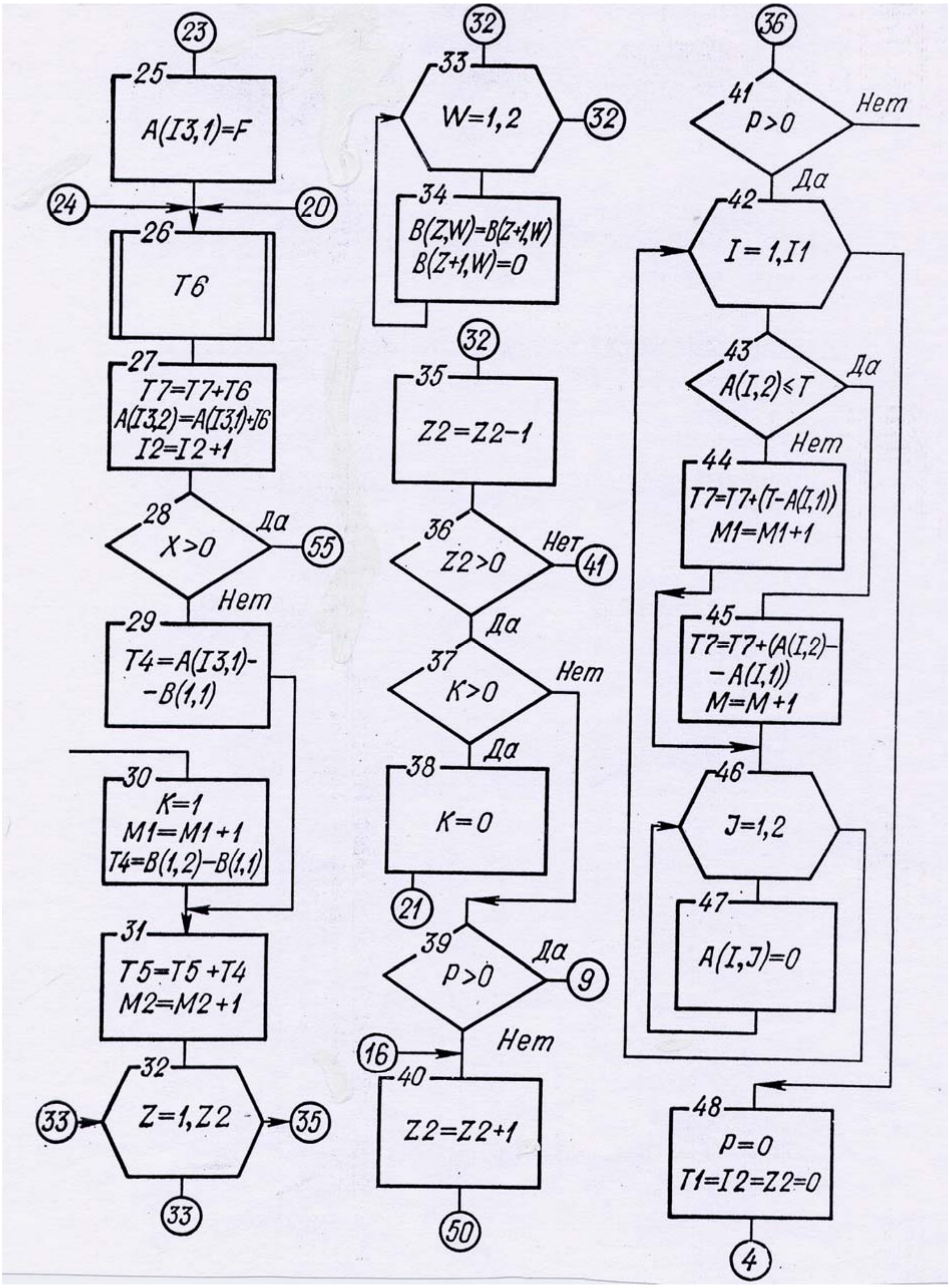


Рис. 2.19. Продолжение

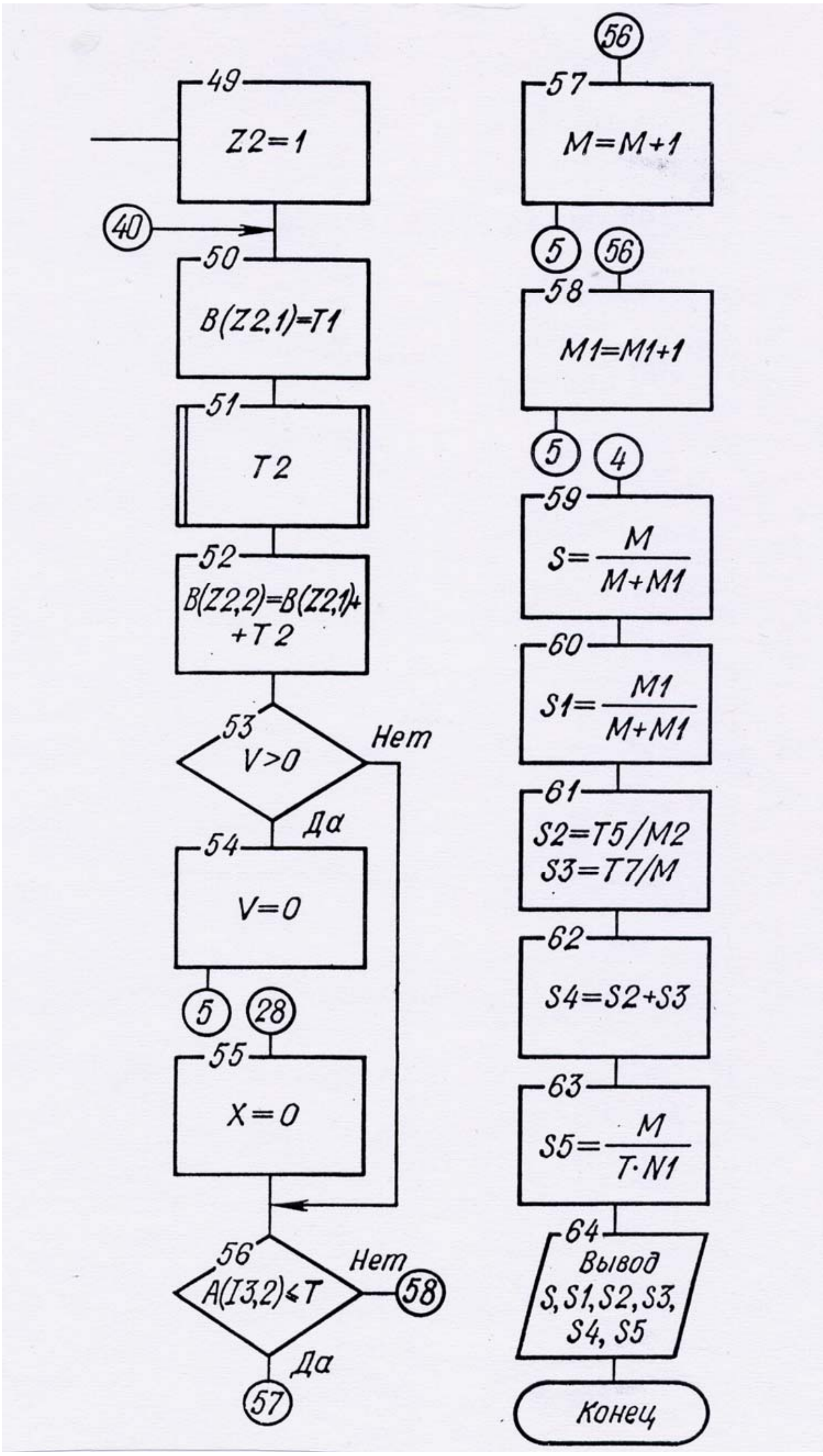


Рис. 2.19. Окончание

V = 1 – все каналы заняты и нужно ставить заявку в очередь, в противном случае V = 0;

R = 1 – все места в очереди заняты, в противном случае *R* = 0;

K = 1 – первая заявка в очереди не дождется начала обслуживания;

P = 1 – *N*-я реализация закончена, больше заявок в этой реализации поступать не будет и необходимо дообработать заявки, еще находящиеся на обслуживании и в очереди.

Для фиксации моментов появления заявок (*T*1), постановки их на обслуживание на конкретный канал или на конкретное место в очереди, моментов окончания ожидания (*T*3) и окончания обслужива-
ния используется два двумерных массива.

A(*I*,*J*) – массив состояния каналов, где *I* $=\overline{1,I1}$ – номер канала (номер строки массива), *J* = 1, 2 – номер столбца. Первый столбец отводится для фиксации момента начала обслуживания заявки *I* - м каналом, второй – момента окончания обслуживания.

B(*Z*,*W*) – массив состояния очереди, *Z* = $\overline{1,Z1}$ – номер места в очереди (номер строки массива), *W* = $\overline{1,2}$ – номер столбца. Первый столбец отводится для фиксации заявки (момента *T*1 ее прихода) в очереди (место в очереди номер *Z*), второй – момента *T*3 окончания расчетного времени ожидания этой заявки начала обслуживания.

В последних блоках модели (59 ... 63) производится обработка результатов моделирования. При этом определяются следующие величины: *S* – вероятность обслуживания заявки, *S*1 – вероятность отказа, *S*2 – среднее время пребывания заявки в очереди, *S*3 – среднее время пребывания заявки на обслуживании, *S*4 – среднее время пребывания заявки в системе, *S*5 – производительность системы.

Подробно с методикой получения и обработки результатов моделирования можно ознакомиться в параграфах 5.7 и 6.2 данного учебного пособия и в разделе 3.4 учебника [41].

6.6. Методика определения приоритетов обслуживания заявок

Потребность построения эффективного алгоритма организации обслуживания заявок приводит к необходимости присвоения им некоторых признаков, которые обычно называются приоритетами.

Как уже было сказано, заявки на обслуживание поступают в СМО в случайные моменты времени и становятся в очередь. При этом возникает вопрос, в какой последовательности их обслуживать? Как раз здесь вступает в действие система приоритетов. Но остаётся неясным принцип присвоения приоритетов. Эти принципы могут быть совершенно различными. Вот один из возможных алгоритмов реализации подобной задачи:

– на основании анализа характеристик заявок на обслуживание (обслуживаемых задач) устанавливается множество параметров этих заявок (задач) и элементам множества придаются некоторые значения;

– в соответствии со значениями элементов множества параметров выбираются методы упорядочения выполнения заявок (задач);

– сравниваются варианты упорядочения заявок (задач) по различным элемента множества параметров;

– разрабатывается правило использования выбранного метода и оценивается его эффективность.

Предположим, удалось выделить несколько элементов x_i множества признаков $X = \{x_1, x_2, \ldots\}$. Пусть это будут, например, следующие характеристики: потребный ресурс на выполнение данной заявки, отношения с заказчиком (с источником заявки), характеристика срочности выполнения заявки, ожидаемая прибыль и так далее. Порядок учёта признаков пусть соответствует номерам индексов признаков x_i (i = 1, 2, …). Элементам множества X придаются некоторые значения, например, соответствующие смыслу, или важности, или весу этих элементов. Принцип учёта элементов множества X заключается в придании элементам значений y_i, равных 0 или 1. Пусть значение 1 соответствует случаю, когда отдаётся предпочтение большему по абсолютному значению элементу множества X, а 0 – меньшему.

Методику использования элементов множества признаков можно пояснить на примере. Пусть используется три (i = 3) элемента множества X: x_1, x_2, x_3. Причём y_1 = 1, y_2 = 0, y_3 = 1. В очереди на обслуживание находится, например, 5 заявок и приняты следующие значения элементов x_i множества X для каждой заявки:

$x_{1(1)} = 0.4$,	$x_{2(1)} = 0$,	$x_{3(1)} = 1$,
$x_{1(2)} = 0.4$,	$x_{2(2)} = 0$,	$x_{3(2)} = 1$,
$x_{1(3)} = 0.2$,	$x_{2(3)} = 2$,	$x_{3(3)} = 3$,
$x_{1(4)} = 0.4$,	$x_{2(4)} = 1$,	$x_{3(4)} = 4$,
$x_{1(5)} = 0.2$,	$x_{2(5)} = 2$,	$x_{3(5)} = 5$.

Индекс в скобках относит значение параметров к одной из рассматриваемых заявок.

Выбор последовательности заявок для реализации происходит следующим образом. При анализе по первому элементу x_1 множества X будут выбраны 1, 2 и 4 заявки. При анализе по второму элементу x_2 множества X – 1 и 2 заявки. А при анализе по третьему элементу окончательно будет выбрана 2-я заявка. На следующем этапе (осталось 4 заявки), уже после анализа по второму элементу x_2, будет выбрана 1-я заявка. После этого уже на первом этапе будет выбрана 4-я заявка. Остаётся выбрать последовательность выполнения 3-й и 5-й заявок. При анализе по x_1 и x_2 не удаётся ничего выбрать, так как $x_{1(3)} = x_{1(5)}$ и $x_{2(3)} = x_{2(5)}$ и только при анализе по элементу x_3 первой будет выбрана 5-я заявка. И последней выполняется оставшаяся 3-я заявка. Таким образом, установлена следующая последовательность выполнения заявок: 2, 1, 4, 5, 3.

Что можно сказать о таком подходе к выбору приоритетов обслуживания? Во-первых, в очереди может находиться в разное время различное количество заявок, ожидающих присвоения приоритета и начала обслуживания. Во-вторых, сам алгоритм выбора последовательности выполнения заявок (алгоритм присвоения приоритетов) довольно громоздкий и на свою реализацию может потребовать некоторый вычислительный ресурс, возможно, сравнимый с получаемым выигрышем. И в третьих, для реализации подобного алгоритма присвоения приоритетов нужен предварительный модельный эксперимент для подбора элементов множества X, выбора порядка их упорядочения, значения этих элементов и принципа их учёта. Если же все эти препятствия преодолены, то в некоторых случаях он может дать значительный эффект.

Возможен и иной подход к определению приоритетов на обслуживание заявок. Он основан на учёте некоторых рассчитываемых случайных факторов [39, 40]. Пусть при формировании очереди заявок на обслуживание учитываются два фактора z_1 и z_2, принимающие значения 0 или 1. Можно сказать, что формируется случайный вектор приоритетов Z, включающий две компоненты z_1 и z_2, то есть вектор $Z(z_1, z_2)$. Первая компонента принимает два значения $z_{11} = 1$ (например, существует возможность предоплаты) и

$z_{12} = 0$ (возможность предоплаты отсутствует). Компонента z_2 также может принимать два значения $z_{21} = 1$ (например, существует договор на обслуживание заявок заказчика) и $z_{22} = 0$ (договора нет).

Закон распределения вектора Z удобно задать в виде табл. 2.7.

Таблица 2.7

z_1 \ z_2	z_{21}	z_{22}
z_{11}	$P(z_{11}\ z_{21})$	$P(z_{11}\ z_{22})$
z_{12}	$P(z_{12}\ z_{21})$	$P(z_{12}\ z_{22})$

Здесь $P(z_{11}\ z_{21})$ – вероятность совместного появления компонент z_{11} и z_{21} (например, есть предоплата и договор);

$P(z_{11}\ z_{22})$ – вероятность совместного появления компонент z_{11} и z_{22} (например, есть предоплата, но нет договора);

$P(z_{12}\ z_{21})$ – вероятность совместного появления компонент z_{12} и z_{21} (например, нет предоплаты, но есть договор);

$P(z_{12}\ z_{22})$ – вероятность совместного появления компонент z_{12} и z_{22} (например, нет предоплаты, и нет договора).

Вероятности $P(z_{11}\ z_{21})$, $P(z_{11}\ z_{22})$, $P(z_{12}\ z_{21})$, $P(z_{12}\ z_{22})$ могут вычисляться на основе анализа статистического материала (табл. 2.8). Здесь кроме вышеперечисленных факторов учитывается отказ в обслуживании.

Таблица 2.8

Данные об обслуживании по кварталам					
Квартал	Общее число заявок	По договору	По предоплате	Получили отказ	Прочие заявки
1	1856	920	400	37	499
2	2004	883	671	49	401
3	1639	764	528	23	324
4	1922	952	594	34	342
Итого за год	7421	3519	2193	143	1566

Считается, что заявки получают отказ, например, в случаях:

– предлагается не предоплата, а бартер;

– предоплата составляет менее 50 %, а в базе надёжности у заказчика неважная характеристика (или вообще отсутствует в базе);

– нет достаточного вычислительного ресурса для качественного обслуживания заявки.

Можно рассчитать вероятности событий с учётом сведений из табл. 2.8. Для этого сначала рассчитываются вероятности для компоненты z_1:

$P(z_1 = z_{11} = 1) = 3519 / 7421 = 0.47$,

$P(z_1 = z_{12} = 0) = 1 - 0.47 = 0.53$.

Получена таблица распределения первой компоненты z_1 (табл. 2.9).

Таблица 2.9

Z_1	$z_{11} = 1$	$z_{12} = 0$
P	0.47	0.53

Затем рассчитываются вероятности для компоненты z_2 (из табл. 2.8).

$P(z_2 = z_{21} = 1) = 2193 / 7421 = 0.30$,

$P(z_2 = z_{22} = 0) = 1 - 0.30 = 0.70$.

Получена таблица распределения второй компоненты z_2 (табл. 2.10).

Таблица 2.10

Z_2	$Z_{21} = 1$	$Z_{22} = 0$
P	0.30	0.70

Теперь можно рассчитать вероятности событий для табл. 2.8.

$P(z_{11}\ z_{21}) = P(z_{11}) * P(z_{21}) = 0.47 * 0.30 = 0.14$,

$P(z_{11}\ z_{22}) = P(z_{11}) * P(z_{22}) = 0.47 * 0.70 = 0.33$,

$P(z_{12}\ z_{21}) = P(z_{12}) * P(z_{21}) = 0.53 * 0.30 = 0.16$,

$P(z_{12}\ z_{22}) = P(z_{12}) * P(z_{22}) = 0.53 * 0.70 = 0.37$.

Таким образом, распределение вектора Z для приведённых статистических данных можно свести в табл. 2.11.

Таблица 2.11

z_2 / z_1	$z_{21} = 1$	$z_{22} = 0$
$z_{11} = 1$	0.14	0.33
$z_{12} = 0$	0.16	0.37

Исходя из полученных результатов при имитационном моделировании реальной СМО, заявке присваивается наивысший (нулевой) приоритет, если $z_1 = 1$ и $z_2 = 1$; первый приоритет – $z_1 = 1$ и $z_2 = 0$, второй приоритет – $z_1 = 0$ и $z_2 = 1$ и третий приоритет – $z_1 = 0$ и $z_2 = 0$. В итоге заявки, приходящие на вход с одномерным распределением интервалов времени поступления, выстраиваются в очередь, длительность пребывания в которой определяется двумерным законом распределения (табл. 2.11).

Количество учитываемых параметров при моделировании приоритетов может быть и больше. Если, например, переменные z_1 и z_2 принимают по три возможных значения, то число приоритетов будет равно 9. Если учитываются три фактора ($Z(z_1, z_2, z_3)$) при двух значениях каждого из них, то – 12. Если 3 при трёх значениях каждого, то – 27 и так далее. Число моделируемых приоритетов может увеличиваться, но методика их задания, усложняясь, сохраняется.

Глава 7

МОДЕЛИРОВАНИЕ НА GPSS

Язык *GPSS* (*General Purpose Simulation System* – система имитационного моделирования общего назначения) предназначен для построения дискретных моделей и проведения имитационного моделирования. Он создан в фирме IBM в начале 1960-х годов и является наиболее популярным в мире инструментом имитации. Основная ориентация языка – системы массового обслуживания произвольной структуры [21, 26, 33, 37, 42, 45].

За время своего существования язык прошел бурный путь развития и в настоящее время является популярным и наиболее распространённым языком имитационного моделирования. Есть трансляторы для операционных систем *DOS* – *GPSS/PC*, для *OS/2* и *DOS* – *GPSS/H* и для *Windows* – *GPSS World*. В настоящей главе в рамках доступного объёма описана методика проведения имитационных экспериментов на «студенческой» версии *GPSS World* 4.3.5, далее именуемой просто *GPSS*. Она бесплатно распространяется через Интернет (электронный адрес *www.minutemansoftware.com*).

В отличие от традиционных методов программирования разработка имитационной модели требует перестройки принципов моделирования. Недаром принципы, положенные в основу имитационного моделирования, дали толчок к развитию объектного программирования.

7.1. Введение в язык GPSS

Система *GPSS* представляет собой язык и транслятор. Как каждый язык он содержит словарь и грамматику, с помощью которых могут быть разработаны модели систем определённого типа. Транслятор языка работает в две фазы. На первой фазе компиляции проверяется синтаксис и семантика написания строк *GPSS*-программы или всей программы в целом, а на второй (интерпретирующей) осуществляется продвижение транзактов по модели от блока к блоку.

Использование языка *GPSS* приводит к достаточно высокой степени автоматизации процесса имитационного моделирования благодаря тому, что логика моделирования исследуемых дискретных процессов встроена в интерпретатор *GPSS*, а также благодаря тому, что автоматически производится сбор статистических данных, характеризующих поведение модели, автоматически по окончании моделирования выводятся результаты исследования.

Модели СМО, написанные на языке *GPSS*, более компактны, чем модели, написанные на универсальных языках программирования: Паскаль, Си, Бейсик и др. Язык *GPSS* получил в последние годы наибольшее распространение по сравнению с языками моделирования *SIMULA, SIMSCRIRT, SOL* и др.

Любая модель исследуемой системы может быть представлена описанием абстрактных элементов – объектов и логических правил их взаимодействия – стандартных операций. Комбинируя объекты и стандартные операции над ними, можно осуществлять эксперименты над интересующей исследователя системой (моделью). Комплекс программ, описывающих функционирование объектов и реализующих логические операции, положен в основу создания программного комплекса системы моделирования *GPSS.*

Объекты делятся на четыре класса: динамические, оборудование, статистические и операционные. Динамические объекты называются транзактами. **Транзакт** – это абстракция, которой разработчик модели придает определенный смысл, т.е. устанавливает его соответствие с объектами реальной системы. Транзакты создаются и уничтожаются в соответствии с логикой моделирования. С транзактами связываются некоторые характеризующие их параметры.

Объекты, называемые оборудованием, – тоже абстрактные элементы, управляемые транзактами: устройство, память и логические ключи. Устройство обслуживает одновременно один транзакт, а накопитель – несколько транзактов. Логический переключатель под влиянием прошедшего транзакта изменяет путь движения других транзактов в соответствии с логикой работы модели.

Для оценки поведения системы используются статистические объекты двух видов: очереди и таблицы.

Логика работы системы формируется операционными объектами, называемыми блоками. Они определяют пути движения транзактов, давая им указание, куда идти и что делать дальше.

Схему, описывающую структуру системы и закономерности ее функционирования, можно представить, состоящей из последовательности блоков и соединяющих их линий. При моделировании производится генерация транзактов, продвижение их через блоки и выполнение действий, соответствующих этим блокам. Продвижение транзактов осуществляется в определенные моменты времени. Для этого моделируются часы, показывающие текущее время в исследуемой системе. Время в модели – только целочисленные величины. Единица времени (секунда, месяц и т.п.) выбирается пользователем и должна быть единой в модели. При моделировании не анализируется состояние системы в каждый последующий момент времени (значение текущего момента плюс единица). Оно изменяется сразу до времени наступления ближайшего из рассматриваемых событий.

Алфавит языка включает латинские буквы, цифры и специальные символы. Служебные слова набираются заглавными буквами, а индивидуальные имена объектов – строчными, начиная с заглавной.

7.2. Моделирование случайных величин

Содержательный смысл транзактов определяется разработчиком модели. Им устанавливается аналогия между транзактами и реальными динамическими элементами моделируемой системы. Эта аналогия нигде не указывается и остаётся только в мыслях разработчика модели. Это могут быть покупатели, клиенты, сигналы, заявки и пр.

Транзакты нумеруются по мере их появления в модели, а их параметры отображают свойства моделируемого динамического объекта. Например, если моделируется система передачи данных, то параметрами транзакта (сообщения) в зависимости от целей моделирования могут быть скорость передачи, объём сообщения и др. Транзакт имеет прямую аналогию с заявкой в СМО.

При моделировании транзакт начинает движение, выходя из некоторого истока, проходит определенные блоки и заканчивает перемещение в стоке, т.е. исчезает из модели. Истокам и стокам соответствуют блоки **GENERATE** (генерировать) и **TERMINATE** (завершить) соответственно.

В момент входа транзакта в блок вызывается соответствующая этому блоку подпрограмма. Далее транзакт пытается войти в следующий блок. Его перемещение продолжается до тех пор, пока не выполнится одно из следующих возможных условий:

- транзакт входит в блок, функцией которого является задержка транзакта на определённое время;
- транзакт входит в блок, функцией которого является удаление транзакта из модели;
- в соответствии с логикой модели транзакт пытается войти в следующий блок, но блок не принимает его. В этом случае транзакт остаётся в блоке, в котором находится в данное время, но позже будет повторять попытки войти в следующий блок. При изменении условий модели одна из таких попыток может оказаться успешной. После этого транзакт продолжит своё движение по модели.

Когда транзакт входит через блок **GENERATE**, интерпретатор планирует время прихода следующего разыгрыванием случайного числа в соответствии с распределением интервала поступления и прибавляет его к текущему значению модельного времени. При достижении этого времени следующий транзакт вводится в модель и т.д.

При моделировании входящих потоков с равномерным законом распределения общий вид блока:

GENERATE ***A***, ***B***, ***C***, ***D***, ***E***,

т.е. генерируются транзакты через ***A*** единиц времени, ***B*** – половиной поля допуска равномерно распределенного интервала, с задержкой ***C***, ***D*** транзактов, с приоритетом ***E***. Операнд ***E*** устанавливает для транзакта уровень приоритета, задаваемого числами от 0 до 127. Чем больше число, тем выше приоритет.

Пример 2.16. Имитировать входящий поток из 100 заявок 30-го уровня приоритета вычислительной системы с равномерным распределением интервалов между заявками на отрезке [10, 20]. Для этого достаточно записать: **GENERATE 15, 5, , 100, 30.**

Для удаления транзакта из модели, т.е. для имитации выходящих потоков, используется блок **TERMINATE.** Он всегда разрешает транзакту выйти, а количество этих блоков в модели не ограничено.

Общий вид блока:

TERMINATE ***A*.**

Операнд ***A*** задает величину, вычитаемую из счетчика завершений каждый раз при входе транзакта в блок **TERMINATE.**

В счетчик завершений интерпретатор заносит целое положительное число из поля операндов оператора **START** (начать). При моделировании транзакты попадают в блоки **TERMINATE,** и содержимое счетчика уменьшается. При достижении им нулевого значения моделирование завершается.

Пример 2.17. Выполнить моделирование поведения вычислительной системы для 50 заявок. Для этого записывается:

. . .

TERMINATE 1

START 50

В стандартной версии *GPSS* имеется восемь различных источников равномерно распределенных случайных чисел со следующими символическими именами: ***RN*1, *RN*2, ..., *RN*8.** Все они (***RN**$_j$*) выдают идентичные последовательности случайных чисел в интервале 0 … 0.999. В *GPSS World* количество генераторов не ограничено, а выдаваемые значения 0 … 0.999999.

Равномерно распределенные случайные числа применяются для последующего преобразования в случайные числа с интересующим пользователя распределением: пуассоновским, экспоненциальным и др. В этом случае интерпретатор использует имя одного из генераторов в качестве аргумента соответствующей функции, обрабатывая которую, он обратится к генератору, получит равномерно распределенное случайное число и преобразует его в величину с заданным законом распределения.

Значения генерируемых чисел зависят от контекста: если имя генератора – аргумент функции, то выдаются шестизначные числа в интервале [0.000000; 0.999999], иначе формируется трехзначное число от 000 до 999.

Пример 2.18. Как формируется последовательность случайных величин равномерно распределенных в интервале [10, 20] в примере 2.16. Для планирования прихода первого транзакта разыгрывается случайное число в интервале [0, 10]. Для этого интерпретатор обращается к ***RN*1**, генерирующему значение 0.000573. Случайная величина x_i = 10 + (20 -10)***RN*1** имеет равномерное распределение на интервале [10, 20]. Тогда x_1 = 10 + 10*0,000573, а с учетом целочисленности x_1 = 10 и т. д.

Для получения чисел с отличным от равномерного распределением используются специальные объекты – функции, которые бывают дискретными, непрерывными и табличными.

Дискретная функция представляет собой кусочно-постоянную функцию, графическое представление которой состоит из горизонтальных ступеней. Непрерывная функция – это кусочно-непрерывная функция. Она в *GPSS* состоит из соединённых между собой прямых отрезков и представляет собой ломаную линию.

Для задания дискретной функции необходимо задать координаты крайних правых точек горизонтальных отрезков. Для непрерывной – координаты всех точек, являющихся концами отрезков.

Для определения дискретной функции указываются: символическое или числовое имя; аргумент (RN_j); число значений, принимаемое случайной величиной; значения переменной и функции распределения.

Эта информация вводится оператором **FUNCTION** (функция). Его общий вид:

ИМЯ

FUNCTION ***A, B***

$x_1, y_1 / x_2, y_2 / \ldots /x_n, y_n$

Операнд ***A*** – это ссылка на генератор, в операнде ***B*** первым стоит символ ***D*** или ***C*** (дискретная или непрерывная функция), за которым указано число значений n случайной величины x. Далее (в следующей строке, начиная с первой позиции поля имен) записываются x_1, x_2, ..., x_n и соответствующие им значения функции распределения y_1, y_2, ..., y_n.

Пример 2.19. Фрагмент *GPSS*-модели для имитации дискретного входящего потока представлен ниже:

```
DRAS FUNCTION RN3, D5
.15, 2 / .35, 5 / .6, 8 / .82, 9 / 1, 1.2
GENERATE RN3 DRAS
```

Когда имеет место пуассоновский входящий поток, интервалы между приходами заявок в СМО распределены по экспоненциаль-ному закону. Для расчета значений случайной экспоненциально распределенной величины x_i используется функция $x_i = \Theta\ (-\ lnk_i) =$

$= \Theta f(k_i)$, где Θ – математическое ожидание, а k_i – случайное равномерно распределенное на интервале [0, 1] число. В *GPSS* нет подпрограммы расчета значений натурального логарифма, поэтому $f(k_i)$ аппроксимирована 23 отрезками, чего вполне достаточно для практических целей.

Пример 2.20. Функция $f(k_i)$ задаётся как непрерывная функция следующим образом:

```
EXPON    FUNCTION    RN1,  C24
0, 0 /.1, .104 / .2, .222 / .3, .355 / .4, .509 / .5, .69 /
.6, .915 / .7, 1.2 / .75, 1.38 /.8, 1.6 / .84, 1.83 / .88,
2.12/
.9 ,2.3 / .92, 2.52 / .94, 2.81 / .95, 2.99 / .96, 3.2 /.97,
3.5 /.98,  3.9 /
.99, 4.6 / .995, 5.3 / .998, 6.2 / .999, 7 / .9998, 8 /
GENERATE 10, FN3 EXPON,
```

т.е. для определения значений случайной величины, распределенной по экспоненциальному закону, для числа с ***RN1***, например 0.45, отыскивается значение 0.599 (среднее между 0.509 и 0.69) и вычисляется x_i = 10 * 0.599 = =5.99 ≈ 6.

~~Пример 2.21. *Содать функцию для моделирования случайной величины,*~~ *равномерно распределённой на интервале [2, 5]. Эта величина может быть смоделирована функцией:*

```
                                                  INN
FUNCTION  RN2,C2

0,2/1,6
```

Современная система – GPSS, *кроме описанных приёмов, позволяет использовать библиотеку процедур, включающую 24 вероятностных распределения: нормальное (*Normal*), Пуассона (*Poisson*), равномерное (*Uniform*), экспоненциальное (*Exponential*) и пр.*

Пример 2.22. Для генерации потока транзактов может быть использована библиотечная процедура экспоненциального распределения с параметром λ = =0.25 и использованием генератора случайных величин ***RN1***:

GENERATE

(***Exponential***(1,0,(1/0.25))).

7.3. Модель одноканальной СМО

Для составления модели одноканальной СМО необходимо прежде всего освоить использование блоков **GENERATE** и **TERMINATE** .

Статические элементы СМО моделируются специальными объектами *GPSS* – приборами. Прибор может обслуживать лишь одну заявку, при этом она задерживается на время обслуживания. Занятие и освобождение прибора моделируются в *GPSS* блоками **SEIZE** (занять) и **RELEASE** (освободить) соответственно. Общий вид блоков: **SEIZE *A*** и **RELEASE *A***.

Операнд ***A*** обозначает символическое или числовое имя занимаемого или освобождаемого прибора. Если прибор не занят, то транзакт входит в **SEIZE**. Если прибор уже занят, то транзакту не разрешается вход в блок **SEIZE** и он ожидает очереди. Блок **RELEASE** всегда разрешает вход транзакту.

Если требуемый прибор оказывается занятым, то транзакт ожидает освобождения, находясь в предшествующем **SEIZE** блоке, и становится элементом очереди. Если в ней более одного транзакта, то первым поступает на обслуживание (после освобождения прибора) пришедший раньше.

Для сбора статистики по очереди (общее и среднее число транзактов, находившихся в очереди, среднее время ожидания и др.) в модель, включается регистратор очереди. Их может быть несколько, различающихся символическими или числовыми именами.

Регистратор очереди задается парой взаимодополняющих блоков **QUEUE** (встать в очередь) и **DEPART** (покинуть очередь), имитирующих события «присоединение к очереди» и «уход из очереди» соответственно. Общий вид блоков:

QUEUE *A, B* и **DEPART *A, B*.**

Операнд ***A*** указывает имя регистратора (по умолчанию – ошибка), а ***B*** – число элементов, на которое изменяется значение счетчика содержимого очереди (по умолчанию – единица).

При входе транзакта в блок **QUEUE** значение счетчика входов для очереди ***A*** увеличивается на единицу, а счетчика содержимого на величину ***B*,** транзакту приписывается имя очереди и запоминается время вхождения в нее. Транзакт перестает быть элементом очереди только после перехода в **DEPART,** выполняющего действия, обратные **QUEUE.** Транзакт может быть одновременно элементом не более пяти очередей.

Теперь можно рассмотреть модель простейшей одноканальной СМО. Пусть система обрабатывает заявки на обслуживание с показательно распределенными интервалами между смежными заявками и длительностями обслуживания. Используется принцип «первый пришел – первым обслужен».

Моделирование выполняется в следующем порядке: разработка схемы СМО, формирование таблицы определений, выбор единицы модельного времени, составление программы и обсуждение результатов. Вместо таблицы определений ниже перечисляются основные обозначения:

- заявка – транзакт первого сегмента,
- таймер – транзакт второго сегмента,
- очередь – регистратор очереди с символическим именем **OCH,**
- система – прибор с символическим именем **PROCE.**

Программа, состоящая из двух сегментов, представлена ниже.

Запуск системы – *GPSS* осуществляется одним из способов запуска приложений *Windows*, например, щёлкнув левой кнопкой мыши по иконке *GPSS* на рабочем столе, или – **Пуск – Программы –** ***GPSS World.*** Появляется рабочее окно системы (фрагмент которого представлен на рис. 2.20), содержащее, как обычно, строку меню и панель инструментов.

Рис. 2.20. Фрагмент рабочего окна системы

Реализовав команду ***File – New,*** в открывшемся диалоговом окне **Новый документ** (рис. 2.21) выбрать команду **Создать Model.**

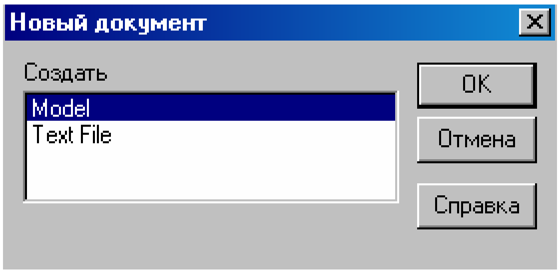


Рис. 2.21. Диалоговое окно **Новый документ**

Откроется окно **Untitled Model 1** (Модель 1 без имени), в котором можно уже набирать текст программы. После успешной отладки можно сохранить модель под именем **Model OSM01.gps** (рис. 2.22).

```
Model OSM01.gps
*****************************
*   Модель   m/m/1/         *
*****************************
; Основной  сегмент
GENERATE    (Exponential(1,0,1))
QUEUE       OCH
SEIZE       PROCE
DEPART      OCH
ADVANCE     (Exponential(2,0,0.9))
RELEASE     PROCE
TERMINATE
; Сегмент   таймера
GENERATE    5000,,,1
TERMINATE   1
START       1
```

Рис. 2.22. Программа модели одноканальной СМО

Как видно из рис. 2.22 допустимо использование комментариев с буквами кириллицы. Они начинаются и заканчиваются звёздочкой или начинаются после точки с запятой.

Модель состоит из двух сегментов. В каждом сегменте модели имеется отдельная пара блоков **GENERATE – TERMINATE,** но только один из **TERMINATE** может задаваться с операндом, вычитаемым из счётчика завершений.

Первый сегмент описывает «биографию» направляемой на обслуживание заявки. Во втором генерируется единственный (см. четвёртый операнд **GENERATE)** транзакт, который обеспечивает завершение моделирования при значении таймера модели 50000 (первый операнд). Отдельный сегмент таймера убыстряет работу модели, так как не требует проверки числа обслуженных заявок для каждого транзакта. Кроме того, этот вариант позволяет при необходимости организовать заключительную обработку, выполняемую однократно после завершения сбора статистики и программируемую непосредственно перед **TERMINATE 1.**

Теперь обсудим первый сегмент. В нём **GENERATE** формирует неограниченное число заявок через экспоненциально распределённые интервалы времени, доставляемые встроенной функцией **Exponential.** Первый операнд функции указывает номер используемого генератора (1), второй – смещение экспоненты (нулевое), третий – её среднее значение (единица).

Вошедшая заявка отмечается в очереди **QUEUE** по имени ***OCH*** и пытается захватить (**SEIZE**) устройство по имени ***PROCE.*** Если устройство занято, транзакт помещается в связанную с ним цепь задержки, которая просматривается в моменты освобождения устройства.

Если устройство свободно, то блок **DEPERT** фиксирует конец пребывания в очереди **OCH.** Соответственно накапливается статистика ожидания, связанная с этой очередью. Блоки **QUEUE** и **DEPERT** определяют только необходимость сбора статистики по данной очереди: их отсутствие не влияет на реальное существование и динамику очереди.

Захватившая устройство заявка задерживается в нём (блок **ADVANCE**) на интервал с показательно распределённой длительностью и средней задержкой 0.9. Для его формирования используется другой (второй) генератор. По истечении задержки блок **RELEASE** освобождает канал ***PROCE,*** и траектория транзакта заканчивается без воздействия на счётчик завершений.

По результатам прогона программы автоматически формируется стандартный отчёт. После заголовка отчёта в нём указывается системное время начала и окончания моделирования и состав системы. Далее следует таблица внутренних адресов, поставленных в соответствие меткам. Затем печатается таблица с количеством входов в блоки (***ENTRIES***), коэффициентом использования (***UTIL***), средним временем пребывания транзакта в устройстве (***AVE.TIME***) и пр. Две последующие выдачи содержат информацию об устройствах и очередях.

Моделирование начинается после того, как интерпретатор обнаружит оператор **START** и поместит в счетчик завершений единицу (**START** обычно находится в конце программы). Порядок следования сегментов произволен.

7.4. Модель многоканальной СМО

При моделировании многоканальных СМО с однородными по свойствам и параллельно работающими приборами используется специальное средство – многоканальное устройство. В модели может быть несколько многоканальных устройств, различающихся символическими или числовыми именами. Число приборов, моделирующих многоканальные устройства, определяется пользователем и называется емкостью многоканального устройства.

Многоканальное устройство вводится парой взаимодополняющих друг друга блоков: **ENTER** (войти) и **LEAVE** (выйти). Когда транзакт входит в первый из них, моделируется событие «Занятие одного из группы параллельно работающих приборов», а когда он попадает во второй – «Освобождение параллельно работающего прибора».

Общий вид блоков:

ENTER ***A, B*** и **LEAVE** ***A, B.***

Операнд ***A*** указывает имя многоканального устройства (по умолчанию – ошибка), а ***B*** – число приборов, занимаемых при входе одного транзакта (по умолчанию – единица).

В конце прогона интерпретатор автоматически распечатывает статистику по многоканальным устройствам. Емкость многоканаль-ного устройства задается оператором **STORAGE** (многоканальное устройство). Общий вид оператора:

ИМЯ

STORAGE ***A***,

где **ИМЯ** – соответствует символическому или числовому наименованию многоканального устройства (операнд ***A*** в блоках **ENTER** и **LEAVE**). Операнд ***A*** определяет емкость многоканального устройства.

Рассмотрим модель многоканальной системы. Пусть СМО состоит из 30 равноценных, параллельно подключенных каналов, каждый из которых функционирует по 24 часа в сутки пять дней в неделю. Время их наработки на отказ 100 ± 15 часов. Возможен ремонт одного или одновременно двух вышедших из строя каналов за 4 ± 2 часа. А так как каналы могут выходить из строя, то имеется резерв – 3 канала. Необходимо выполнить моделирование за 12 месяцев (в месяце 4 недели). Определить загрузку ремонтных мест и среднее число каналов в резерве.

В качестве такой СМО может быть рассмотрено техническое устройство, состоящее из равноценных блоков, подключенных параллельно. Ремонт осуществляется в мастерской двумя рабочими.

Моделируемая система изображена на рис. 2.23. Ремонтный участок, где два рабочих места, и систему, где работают параллельно 30 блоков, удобно моделировать многоканальными устройствами, а сами блоки – транзактами. Пусть все каналы исправны, тогда резервные блоки не могут войти в многоканальное устройство и ожидают в очереди. Как только канал (блок) нарабатывает на отказ, он поступает в очередь на ремонт, а его место занимает резервный блок. После ремонта блоки поступают в резерв, т.е. в очередь. Таким образом, в модели циркулирует 33 блока – транзакта.

Соответствие объектов системы и GPSS: блок – транзакт первого сегмента, таймер – транзакт второго сегмента, канал – многоканальное устройство с символическим именем **KANAL**, ремонтная мастерская – многоканальное устройство с символическим именем **REMM**, резерв – очередь с символическим именем **REZ**.

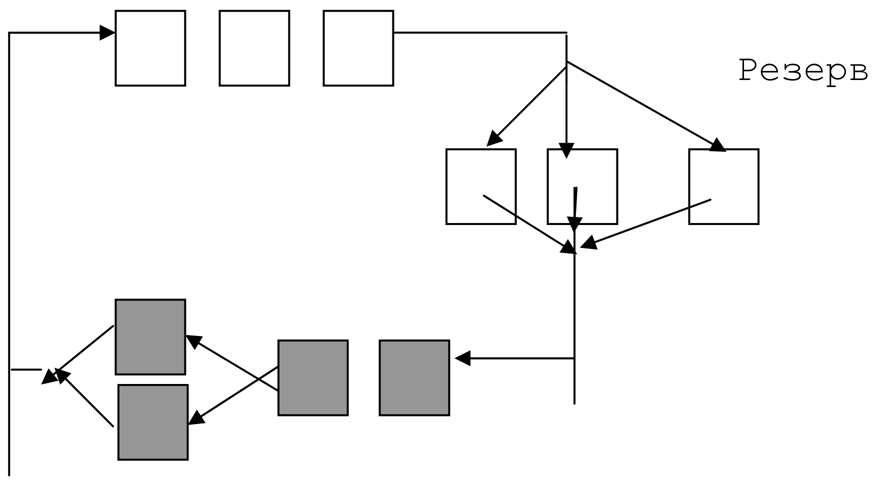


Рис. 2.23. Схема моделируемой многоканальной СМО

В модели необходимо использовать блок **TRANSFER** (передать), предназначенный для безусловной передачи транзакта в блок ***B***, отличный от следующего: **TRANSFER , *B.*** Операнд ***B*** указывает символическое или числовое имя блока, которому передается транзакт. В режиме безусловной передачи **TRANSFER** не отказывает транзакту во входе, а сразу же пытается продвинуть его в блок ***B.*** Если этот блок отказывает, то транзакт остается в **TRANSFER** и повторяет попытки в следующие моменты времени. Программа модели приводится ниже:

```
  KANAL     STORAGE          30
    REMM      STORAGE          2
    ; Сегмент 1
                          GENERATE        ,,, 33
    NNN          QUEUE            REZ
                          ENTER           KANAL
                          DEPART          REZ
                          ADVANCE         100,15
                          LEAVE           KANAL
                          ENTER          REMM
                          ADVANCE         4,2
                          LEAVESFER       REMM
                          TRANSFER        , NNN
     ; Сегмент 2
                          GENERATE        5760
                          TERMINATE       1
                          START           I
```

7.5. Модель многоканальной СМО с ограниченной длиной очереди

Во время моделирования интерпретатор регистрирует и корректирует информацию по объектам GPSS: приборам, очередям, многоканальным устройствам и т.п. Поэтому процессом моделирования можно управлять динамически – воздействием со стороны атрибутов модели. Например, в реальных СМО время обслуживания иногда зависит от длины очереди. В пособии [11] приведены некоторые часто используемые стандартные числовые атрибуты (СЧА). Наименование **СЧА**, как правило, состоит из группового имени (**F** – прибор, **S** – устройство или **Q** – очередь), типа информации (например, счетчик занятий, максимальное значение длины очереди и др.) и имени числового – *j* или символического.

Рассмотрим модель описанной ниже системы. В информационной системе работает пять операторов, они, привлекая технические средства системы, отвечают на запросы, поступающие по каналам связи. Если у каждого оператора по 2 запроса в очереди (т.е. ее общая длина 10 заявок), очередной запрос получает отказ. Заявки поступают через 5 ± 3 секунды, а обслуживаются за 30 ± 20 секунд. Необходимо выполнить моделирование работы системы по обслуживанию 150 заявок. Определить число отказов и загрузку операторов. Схема СМО приведена на рис. 2.24. Соответствие объектов системы и GPSS: заявка – транзакт, очередь – регистратор очереди с символическим именем **TLF**, операторы – многоканальное устройство с символическим именем **KOM**.

Для проверки значений **СЧА** будет использован блок **TEST** (проверить). Его общий вид:

TEST ***X, A, B, C.***

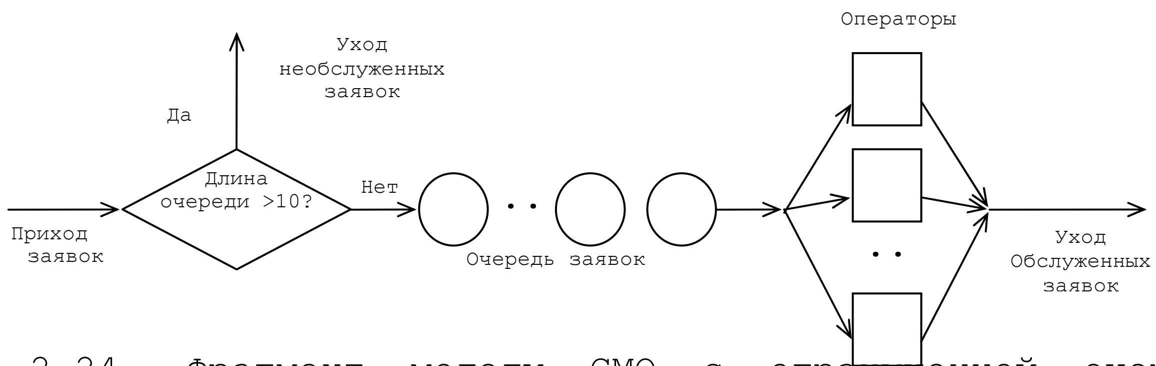


Рис. 2.24. Фрагмент модели СМО с ограниченной очередью

Вспомогательный оператор ***X*** указывает отношение сравнения двух **СЧА,** имена которых определены в ***A*** и ***B***:

G – A > B?,

GE – A >= B?,
E – A = B?,
NE – A <> B?,
LE – A <= B?,
L – A < B?

Операнд ***C*** не обязателен и содержит имя блока, в который переходит проверяющий транзакт, если ответ на вопрос, содержащийся в ***X,*** отрицателен.

Если ***C*** отсутствует, то **TEST** работает в режиме отказа и, когда транзакт пытается войти, вход запрещается, если ответ на вопрос отрицателен. Тогда транзакт задерживается в предыдущем блоке. При очередном просмотре цепи текущих событий попытка повторяется. Если использован операнд ***C,*** то транзакт проходит в следующий за **TEST** блок, если ответ на вопрос положительный, или в блок ***C –*** если отрицательный.

Программа модели приведена ниже:

```
KOM   STORAGE          5
            GENERATE      5,3
            TEST G           Q$TLF,10,NNN
            TERMINATE
 NNN  QUEUE               TLF
            ENTER              KOM
            DEPART           TLF
            ADVANCE       30,20
            LEAVE              KOM
            TERMINATE  1
            START              150
```

Особенностью модели является отсутствие сегмента – таймера. По условию требуется отработать 150 заявок. В этом случае в счетчик завершений оператором START заносится число 150.

Результаты моделирования: число отказов равно 10, а загрузка операторов – 0,981.

7.6. Модель СМО с приоритетами

При входе транзакта в модель уровень его приоритета определяется блоком **GENERATE** (операнд ***E***). Для динамического изменения приоритета в ходе моделирования используется блок **PRIORITY** (назначить приоритет):

PRIORITY *A*.

Операнд ***A*** определяет уровень назначаемого приоритета, являющегося в GPSS **СЧА** с именем **PR.**

Рассмотрим модель товарной станции, к которой с интервалом 8 ± 3 часа подходят поезда с некоторым сырьем. Среди них 20 % составов со скоропортящимся сырьем имеют первый приоритет, а остальные – нулевой. Станция обрабатывает (разгружает) составы за 6 ± 2 часа. Если состав ожидает в очереди более 10 часов, то разгрузка теряет смысл и он уходит на другую станцию.

Необходимо выполнить моделирование системы в течение 5 суток: определить число необслуженных составов, среднее время пребывания в очереди и загрузку станции. Схема системы с приоритетами изображена на рис. 2.25, а соответствие объектов системы и GPSS: состав – транзакт первого сегмента, таймер –транзакт второго сегмента, очередь – регистратор очереди с символическим именем **OCH,** а станция – прибор с символическим именем **STAN.**

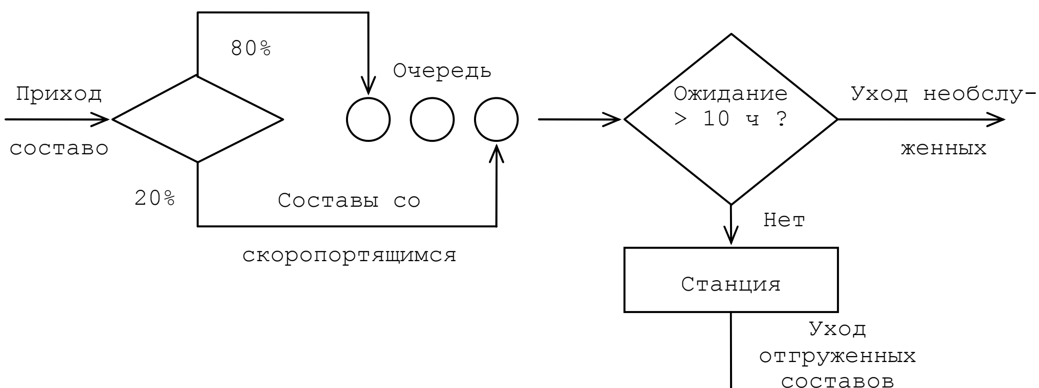


Рис. 2.25. Фрагмент модели СМО с приоритетами

Особенностью модели является использование **СЧА M1,** определяющего число единиц времени пребывания транзакта в модели (резидентное время). Кроме того, применяется блок **TRANSFER** в режиме статистической передачи:

TRANSFER *A, B, C.*

Операнд ***A*** задает частоту передачи транзакта в блок ***C.*** В остальных случаях транзакт переходит в блок ***B.*** Таким образом, в 80 % случаев транзакт переходит в блок **MMM,** а в 20 % – в **NNN.**

GPSS - программа приводится ниже:

```
; Сегмент 1
            GENERATE         8,3
            TRANSFER        .8, NNN, MMM
 NNN    PRIORITY         1
 MMM QUEUE            OCH
            TEST L           MI, 10, TTT
            SEIZE            STAN
            DEPART           OCH
            ADVANCE          6,2
            RELEASE          STAN
            TERMINATE
 TTT     TERMINATE
; Сегмент 2
             GENERATE         96
             TERMINATE        I
             START            I
```

По итогам моделирования все составы разгружены, среднее время пребывания в очереди – 0.083 часа, а загрузка станции – 0.656.

7.7. Модель СМО с выходом из строя и

ремонтом каналов

Транзакты имеют до 100 параметров, которые рассматриваются как принадлежащие им **СЧА.** Значения параметров устанавливаются и модифицируются по желанию пользователя, а также используются как операнды блоков. Число параметров задается в операнде ***F*** блока **GENERATE** (по умолчанию 12). Имя параметра содержит символ и порядковый номер (например, ***P*1, *P*2, ..., *P*100**), а сами они принимают только целочисленные значения со знаком. При входе транзакта в модель все его параметры равны нулю, а назначаются и изменяются они в блоке **ASSIGN** (назначить):

ASSIGN *A, B.*

Данные из операнда ***B*** фиксируются в параметре, имя которого задается в ***A.*** В результате предыдущее значение параметра заменяется на новое.

Рассмотрим некоторые блоки **ASSIGN.** Например, запись **1 ASSIGN 3, 15** означает, что при входе транзакта в блок **1** значение его третьего параметра становится равным **15,** а запись **2 ASSIGN *P*2, 4** – блок **2** записывает **4** как значение параметра, номер которого указан в ***P*2**, т.е., если в ***P*2** число **6,** то шестой параметр примет значение **4.** В блоке

3 ASSIGN *P*1, *FR $ TRAN*

величина нагрузки прибора ***TRAN*** запишется как значение параметра, номер которого в ***P*1.** Четвертый и пятый блоки

4 ASSIGN *2+, Q $ OTH*

5 ASSIGN

***P*3-, 6**

работают в режиме приращения. В четвертом значение параметра **2** увеличивается на текущее содержимое очереди ОТН. В блоке **5** из значения параметра, номер которого указан в ***P*3**, вычитается **6**. Если, например, ***P*3 = 4**, то из значения четвертого параметра вычитается **6**.

Транзакты для занятия прибора ждут, как правило, своей очереди. В некоторых СМО вновь прибывающему разрешается заместить транзакт на занятом приборе. В этом случае прибор отбирается (захватывается) у еще не закончившего обслуживание транзакта, а вновь приходящему разрешается немедленно занять прибор. Замещенный транзакт либо ждет освобождения прибора, либо перемещается на другой. Такая ситуация часто встречается в СМО с ненадежными каналами и заданным временем наработки на отказ.

Захват и возвращение прибора в *GPSS* моделируется блоками **PREEMPT** (захватить) и **RETURN** (вернуть). Формат блока:

PREEMPT

A, B.

В операнде ***A*** указывается имя захватываемого прибора, а в ***B*** – условия возникновения захвата. Ниже рассмотрены лишь простейшие из них.

Если ***B*** опущен, захват происходит, когда обслуживаемый транзакт сам не является захватчиком. При использовании ***B*** (комбинация ***PR***) захват разрешен, когда у захватчика более высокий уровень приоритета, чем у обслуживаемого транзакта. Как в том, так и в другом случаях транзакт, прервавший обслуживание, ожидает конца захвата.

Транзакт-захватчик освобождает прибор, т.е. возвращает его ожидающему транзакту, только войдя в блок **RETURN:**

RETURN ***A.***

Операнд ***A*** указывает имя возвращаемого прибора.

Для того чтобы позволить транзактам обмениваться информацией друг с другом, в *GPSS* используются постоянные ячейки памяти, называемые сохраняемыми. Их значения изменяются только по указанию пользователя блоком **SAVEVALUE** (сохранить величину):

SAVEVALUE ***A, B.***

Операнд ***A*** – номер или символическое имя сохраняемой величины, а операнд ***B*** – величина модификации.

Когда транзакт входит в блок **SAVEVALUE,** величина ***B*** становится значением сохраняемой величины, имя которой определено в ***A***. Например, при входе транзакта в блок

1

SAVEVALUE 1, 500

сохраняемая величина ***N*1** примет значение 500, а при входе в блок

2

SAVEVALUE 2, *P*3

сохраняемая величина ***N*2** станет равной значению второго параметра транзакта. Блок

3 SAVEVALUE

***MAR,Q$OCH*1**

определяет сохраняемую величину ***MAR*** как текущее содержимое очереди ***OCH*1**. Блок

4 SAVEVALUE *P2, V$WAR*

присваивает сохраняемой величине с номером, записанным в ***P2***, значение арифметической переменной ***WAR***. Пятый и шестой блоки

5 SAVE VALUE *1+,20,*

6 SAVEVALUE *5-,FN$ GAMMA*

работают в режиме приращения и уменьшения. При входе транзакта в пятый блок сохраняемая величина ***N*1** увеличивается на 20, а в шестой – уменьшается на величину, определяемую функцией ***GAMMA***, значение сохраняемой величины ***N*5**.

Пусть необходимо выполнить моделирование работы системы, в которую приходят заявки на обслуживание с интервалом 5 ± 2 мин и обрабатываются за 4 ± 3 мин. Время наработки системы на отказ 150 ± 20 часов. Ремонтируется система в среднем 5 ± 2 часа. Заявка, обслуживание которой прервано, ожидает устранения отказа. Выполнить моделирование за 3 месяца. Определить загрузку системы и потери, если час простоя стоит 8 000 руб.

Условная схема работы СМО показана на рис. 2.26, а соответствие объектов *GPSS* и системы такое: заявка – транзакт первого сегмента, отказ – транзакт второго сегмента, таймер – транзакт третьего сегмента, система – прибор с символическим именем ***PROZ***, время простоя – сохраняемая величина, равная 1. За единицу модельного времени принята 1 мин.

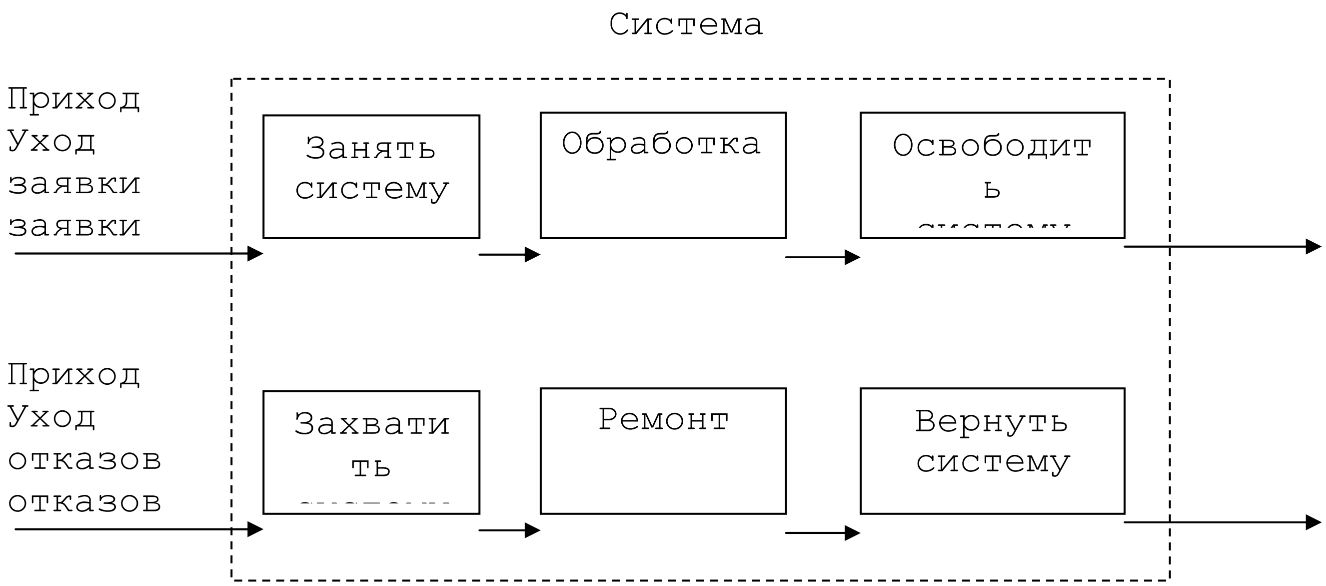


Рис. 2.26. Условная схема СМО с отказами

GPSS-программа приводится ниже:

```
; Сегмент 1
    GENERATE       5,2
    SEIZE                 PROZ
    ADVANCE         4,3
    RELEASE          RROZ
    TERMINATE
; Сегмент 2
    GENERATE       900,120
    PREEMPT         PROZ
    MARK                1
    ADVANCE         300,120
    RETURN            PROZ
    ASSIGN              2, MP1
    SAVEVALUE     1+,P2
                          TERMINATE
; Сегмент 3
    GENERATE        4320
    TERMINATE      1
    START                 1
```

Первый сегмент имитирует обработку заявок, а второй – выход из строя системы, её ремонт и накопление времен простоя. Блок ***MARK*** при входе в него транзакта переносит в параметр с номером, указанным в операнде ***A***, значение таймера абсолютного времени, т.е. делает отметку времени. Когда транзакт достигнет блока

```
ASSIGN  2, MP1,
```

интерпретатор вычисляет значение ***СЧА MPj***, где ***j*** – номер параметра транзакта с отметкой времени. Для этого он вычитает значение ***j***-го параметра из текущего значения таймера абсолютного времени. В итоге в ***P*2** заносится время простоя системы из-за отказа.

В результате моделирования загрузка системы – 0,827, а потери от простоя – 172 400 руб. Они подсчитываются умножением значения ***X*1** на 133 руб./мин.

Выводы по второй части учебного пособия

Имитационная модель – это описание логики функционирования исследуемой системы и взаимодействия отдельных её элементов во времени, выполненное на некотором формальном языке, учитывающее наиболее существенные причинно-следственные связи и обеспечивающее проведение статистических экспериментов [11, 12]. Принципы, на которых базируется вся теория моделирования (принципы информационной достаточности, осуществимости, множественности моделей, агрегирования и параметризации) (см. с. 15–16), остаются действенными и при имитационном моделировании.

Имитационная модель считается реализуемой и имеющей практическую ценность только в том случае, если в ней отражены только те свойства реальной системы, которые влияют на значение выбранного показателя эффективности.

Применение имитационного моделирования целесообразно в следующих случаях:

– если не существует законченной постановки задачи на исследование, и идёт процесс познания объекта моделирования;

– если характер протекающих в системе процессов не позволяет описать эти процессы в аналитической форме;

– если необходимо наблюдать за поведением системы в течение определённого периода, в том числе с изменением скорости протекания процессов;

– при изучении новых ситуаций в системе либо при оценке функционирования её в новых условиях;

– если исследуемая система является элементом более сложной системы, другие элементы которой имеют реальное воплощение;

– когда необходимо исследовать поведение системы при введении в неё новых компонентов;

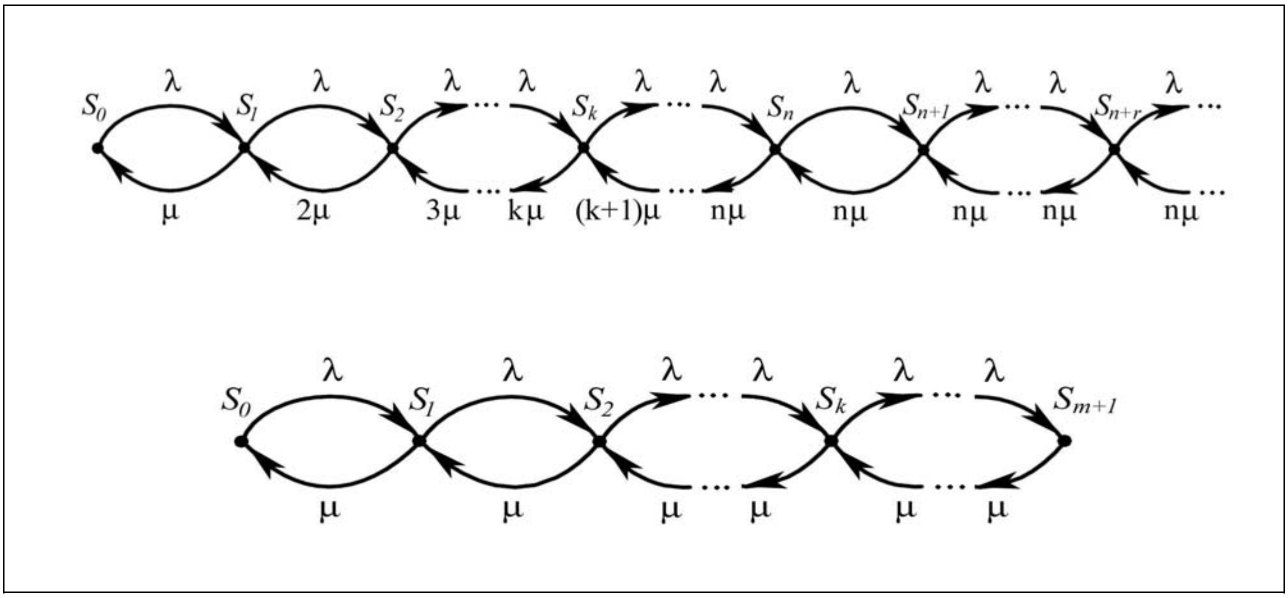
– при подготовке специалистов и освоении новой техники (в качестве тренажеров).

Имитационные модели имеют ряд недостатков. На их разработку требуются значительные затраты времени и сил. Кроме того, любая имитационная модель системы значительно менее объективна, чем аналитическая модель, так как в ней отражаются субъективные представления разработчика о моделируемой системе. Бывает достаточно затруднительно обосновать адекватность созданной имитационной модели, особенно если речь идёт о проектируемой системе. Результаты имитационного моделирования, как и при любом численном методе, всегда носят частный характер. Для получения обоснованных выводов необходимо проведение серии модельных экспериментов, а обработка результатов требует применения специальных статистических процедур.

Имитационное моделирование многими часто воспринимается как «метод последней надежды», и в этом видится толика правды. Но в большинстве ситуаций разработчик быстро осознаёт необходимость прибегнуть именно к этому средству, поскольку исследуемые системы и модели достаточно сложны и их нужно представить доступным способом.

Часть III

АНАЛИТИЧЕСКОЕ МОДЕЛИРОВАНИЕ



Глава 8. Марковские случайные процессы

Глава 9. Моделирование СМО аналитическими методами теории массового обслуживания

Глава 10. Методика выполнения инженерных расчётов в среде Maple

Многие задачи, рассматриваемые в повседневной практике, описываются как **случайные процессы** в связи с тем, что факторы, влияющие на параметры этих задач, являются случайными величинами или случайными функциями. Это приводит к тому, что в большинстве случаев становится трудно или даже невозможно построить точную математическую модель рассматриваемого процесса. Но найти характеристики эффективности рассматриваемых процессов и задач иногда все же удается. Происходит это тогда, когда исследуемая система, процесс или задача могут быть представлены в терминах **марковских случайных процессов.**

В этой части учебного пособия, включающей три главы, даны основные понятия и определения марковских случайных процессов, приёмы моделирования СМО аналитическими методами теории массового обслуживания и методика реализации инженерных расчётов в среде *Maple* [8, 9, 11, 14, 15, 25, 28, 33, 36, 43].

Глава 8
МАРКОВСКИЕ СЛУЧАЙНЫЕ ПРОЦЕССЫ

Случайный процесс, протекающий в системе S, называется **марковским** (или процессом «без последействия»), если для любого момента времени t_0 вероятностные характеристики процесса в будущем (при $t > t_0$) зависят только от его состояния в данный момент времени t_0 и не зависят от того, когда и как система пришла в это состояние. Из приведенного определения очевиден вывод: если рассматриваемый процесс случайный, т. е. непредсказуемый, то поведение системы в будущем точно описать нельзя. Однако некоторые вероятностные характеристики поведения системы можно определить легко, если удается свести случайный процесс к марковскому.

8.1. Процессы с дискретными состояниями и непрерывным временем

Случайные процессы можно разделить на процессы с дискретными и непрерывными состояниями. При рассмотрении процесса с дискретными состояниями всегда возможно все состояния описать (перенумеровать, перечислить), а переход из состояния в состояние осуществляется скачком (мгновенно). Принципиально отличаются от них процессы с непрерывными состояниями, в которых происходит плавный переход из состояния в состояние. При дальнейшем изложении материала предполагается, что описываются и анализируются только процессы с дискретными состояниями.

Для наглядности изображения случайных процессов с дискретными состояниями удобно пользоваться **графами состояний.** Вершины графа – это состояния системы, а дуги – возможные переходы из состояния в состояние. На графе изображаются только непосредственные переходы из состояния в состояние. Если S_1, S_2, …, S_n - дискретные состояния системы S и если переход из S_1 в S_2 возможен лишь через S_3, то стрелками (дугами) на графе отмечаются следующие переходы: $S_1 \to S_3 \to S_2$. При этом нужно еще раз подчеркнуть, что переход происходит скачком (мгновенно).

Пример 3.1. Система S (например, какое-то техническое устройство) может находиться в одном из состояний: S_1 - исправна, S_2 – неисправна, S_3 – осматривается, S_4 – ремонтируется. Граф состояний такой системы представлен на рис. 3.1.

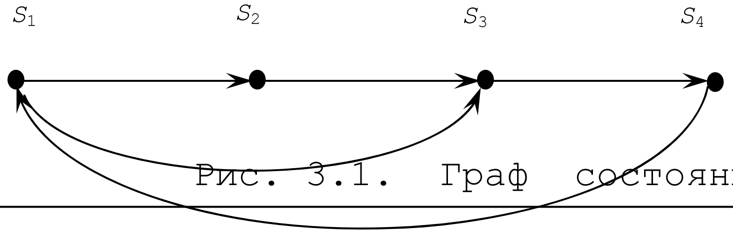


Рис. 3.1. Граф состояний системы S

Пусть система S переходит из состояния S_1 в состояние S_m и из S_m в S_n. Время между этими двумя последовательными переходами, т. е. время нахождения системы в состоянии S_m может быть дискретной или непрерывной переменной величиной. Отсюда идет разбиение систем и случайных процессов на процессы с дискретным временем или на процессы с непрерывным временем.

Случайный процесс, в котором возможные переходы из одного состояния в другое осуществляются лишь в заранее определенные (фиксированные) моменты времени, называется **процессом с дискретным временем**. И соответственно, если переходы возможны в любой случайный момент времени, то это **процесс с непрерывным временем** (см. § 8.2).

В чем суть процесса, происходящего в системе S, которая может, например, находиться в состояниях S_1, S_2, S_3, S_4 (рис. 3.2)?

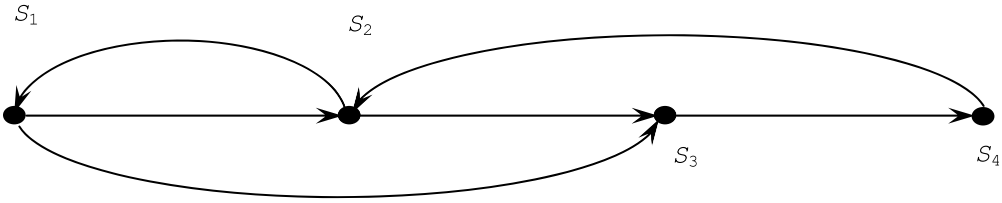


Рис. 3.2. Граф состояний системы S'

Эта система в моменты времени t_k ($k = 0,1,2,3,\ldots$) может находиться, например, в следующих состояниях:

$$S_1 \to S_2 \to S_1 \to S_3 \to S_4 \,,$$

или

$$S_2 \to S_3 \to S_4 \to S_2 \,.$$

При этом нужно учитывать, что система S' в моменты t_k может оставаться (задерживаться) в прежнем состоянии:

$$S_2 \to S_3 \to S_4 \to S_4 \to S_2 \,.$$

Таким образом, при рассмотрении дискретного случайного процесса с дискретным временем в системе S с возможными состояниями S_i $(i = \overline{1,n})$ и моментами t_0, t_1, t_2, ..., t_k, ... перехода (называемыми шагами, или этапами перехода) из состояния в состояние обозначают через S_i^k событие, состоящее в том, что после k шагов система находится в состоянии S_i. Тогда, осуществив привязку к моментам времени (этапам переходов), последовательность смены состояний можно представить так:

$$S_2^0 \to S_3^1 \to S_4^3 \to S_4^4 \to S_2^5 \ldots$$

Марковская цепь. Случайная последовательность событий называется **марковской цепью,** если для каждого шага вероятность перехода из состояния S_i в любое состояние S_j не зависит от того, когда и как система пришла в состояние S_i. Марковская цепь описывается с помощью **вероятностей состояний.** Для $k = 1$

$$p_1(1) = P(S_1^1),\quad p_2(1) = P(S_2^1),\ \ldots,\quad p_n(1) = P(S_n^1),$$

для $k = 2$

$$p_1(2) = P(S_1^2),\quad p_2(2) = P(S_2^2),\ \ldots,\quad p_n(2) = P(S_n^2)$$

и в общем виде

$$p_i(k) = P(S_i^k).$$

То есть $p_i(k) = P(S_i^k)$ - есть вероятность i-го состояния системы S после k-го шага. Необходимо помнить, что на k-м шаге осуществляется одно из полной группы несовместных событий S_i^k $(i=\overline{1,n})$ и что для каждого k

$$p_1 + p_2 + \ldots + p_n = 1,$$

так как $p_i(k)$ - вероятности несовместных событий, образующих полную группу.

Как найти вероятности состояний $p_i(k)$ **для любого** $\boldsymbol{k}$ **?** Для этого пользуются графом состояний системы S. Марковская цепь представляется точкой, перемещающейся по графу и перескаки-вающей из состояния в состояние в моменты t_0, t_1, … или задерживающейся в некоторых состояниях. Очевидно, что для любого момента t_k существуют вероятности перехода системы из состояния S_i в S_j (некоторые из них могут равняться нулю при невозможности перехода), а также могут существовать вероятности задержки системы в определенных состояниях. Они называются переходными вероятностями марковской цепи.

В зависимости от того, изменяются или нет переходные вероятности при увеличении номера шага $\boldsymbol{k}$, рассматривают однородные или неоднородные марковские цепи.

Однородная марковская цепь. Если $p_i(k)$ не зависят от k, такая цепь называется **однородной марковской.** А если $p_i(k)$ зависят от номера шага k – **неоднородной.**

Пусть известны вероятности p_{ij} перехода за один шаг системы S из состояния S_i в состояние S_j $(i,j=\overline{1,n})$. Причем p_{ii} есть вероятность задержки системы в состоянии S_i. Эти вероятности удобно записывать в виде квадратной матрицы Π переходных вероятностей

$$\Pi = \begin{Vmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{Vmatrix}$$

Некоторые элементы p_{ij} этой матрицы могут равняться нулю, если невозможен переход системы за один шаг из состояния S_i в состояние S_j.

Переходные вероятности p_{ij} можно записать как условные вероятности:

$$p_{ij} = P(S_j^k / S_i^{k-1}),$$

т.е. p_{ij} есть условная вероятность того, что система S будет в состоянии S_j на k-м шаге при условии, что была в состоянии S_i на (k-1)-м шаге. Из этого следует, что для всех $i = \overline{1,n}$

$$\sum_{j=1}^{n} p_{ij} = 1,$$

так как для любого k события (состояния) $S_i^{(k)}$ $(i = \overline{1,n})$ несовместны и образуют полную группу.

При использовании графа состояний марковской цепи на нем у стрелок обычно ставят соответствующие переходные вероятности p_{ij}. Такой граф (рис. 3.3) называется **размеченным**.

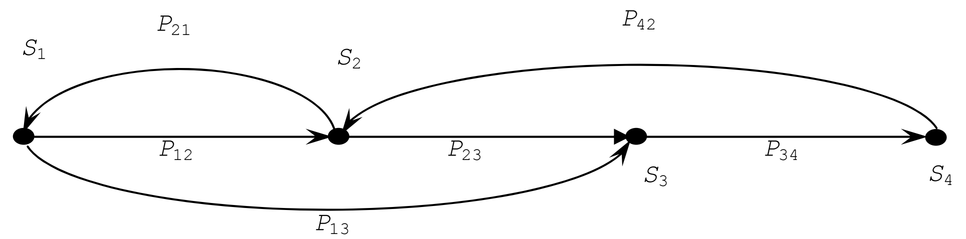


Рис. 3.3. Размеченный граф состояний

На графе (рис. 3.3), чтобы не загромождать рисунок, не проставлены вероятности p_{ii} $(i = \overline{1,n})$, но они легко вычисляются, так как любой диагональный элемент матрицы переходных вероятностей дополняет до единицы сумму элементов соответствующей строки:

$$p_{11} = 1 - (p_{12} + p_{13}), \qquad p_{22} = 1 - (p_{21} + p_{23}),$$
$$p_{33} = 1 - p_{34}, \qquad p_{44} = 1 - p_{42}.$$

Если же из i-й вершины графа не выходит ни одной стрелки, то это свидетельствует о том, что $p_{ii} = 1$.

Как определить вероятности состояний $p_i(k)$ $(i=\overline{1,n})$ после любого k-го шага, если известна матрица переходных вероятностей и известно начальное состояние системы S?

Пусть известно, что перед первым шагом (в начальный момент времени) система S находится в состоянии S_m. Тогда для момента времени $t_0 = k = 0$ можно записать $p_m(0) = 1$,

а $p_1(0) = p_2(0) = ... = p_{m-1}(0) = p_{m+1}(0) = ... = p_n(0) = 0$.

Теперь можно определить $p_i(1)$ $(i=\overline{1,n})$. Известно, что перед первым шагом система была в состоянии S_m, поэтому за 1-й шаг она перейдет в состояние S_i с вероятностями p_{mi}, которые находятся в m-й строке матрицы $\Pi = \left\| p_{ij} \right\|_1^n$, т.е. $p_i(1) = p_{mi}$ $(i=\overline{1,n})$, или иначе

$$p_1(1) = p_{m1}, \quad p_2(1) = p_{m2}, \ ..., \ p_n(1) = p_{mn}.$$

Вероятности состояний после второго шага $p_i(2)$ нужно рассчитывать по формуле полной вероятности с учетом того, что после первого шага система S могла быть или в состоянии S_1, или S_2, или S_3 и т.д. Поэтому по формуле полной вероятности для состояния S_i можно записать:

$$p_i(2) = p_1(1)\cdot p_{1i} + p_2(1)\cdot p_{2i} + ... + p_n(1)\cdot p_{ni}$$

или в сокращенном виде:

$$p_i(2) = \sum_{j=1}^{n} p_j(1)\cdot p_{ji} \qquad (i=\overline{1,n}).$$

После k-го шага

$$p_i(k) = \sum_{j=1}^{n} p_j(k-1)\cdot p_{ji} \quad (i=\overline{1,n}). \tag{3.1}$$

Как читается это выражение? Вероятность i-го состояния на k-м шаге равна сумме по всем j от 1-го состояния до n-го произведений вероятностей j-х состояний на предыдущем $(k-1)$-м шаге на условные вероятности переходов из j-х состояний в i-е.

Таким образом, получена рекуррентная формула (3.1) определения вероятностей состояний системы S после k-го шага по рассчитанным вероятностям состояний для $(k-1)$-го шага.

Пример 3.2. Для размеченного графа состояний системы S (рис. 3.4) рассчитать вероятности состояний $p_i(k)$ для k = 0, 1, 2, 3 при условии, что при $k=0$ система находилась в состоянии S_2.

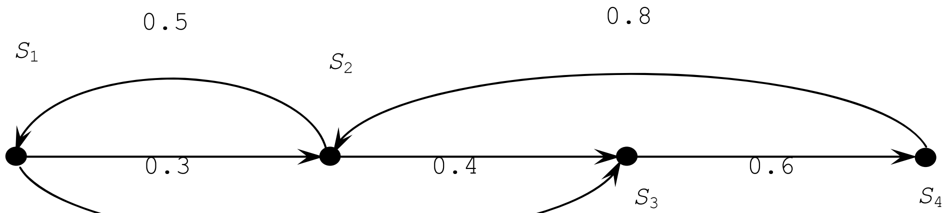


Рис. 3.4. Размеченный граф состояний примера 3.2

Для удобства расчетов переходные вероятности p_{ij} (рис. 3.4) представляются в виде матрицы

$$\Pi = \begin{Vmatrix} 0.5 & 0.3 & 0.2 & 0 \\ 0.5 & 0.1 & 0.4 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0.8 & 0 & 0.2 \end{Vmatrix}$$

В начальный момент времени для $k=0$ $p_2(0)=1$, т.е. система S находилась в состоянии S_2, а для остальных состояний вероятности равны нулю:

$$p_1(0) = p_3(0) = p_4(0) = 0 .$$

Для k =1 используются переходные вероятности, взятые из второй строки матрицы, т.е. считается, что предыдущее состояние было строго определено:

$$p_1(1) = p_{21} = 0.5,$$
$$p_2(1) = p_{22} = 0.1,$$
$$p_3(1) = p_{23} = 0.4,$$
$$p_4(1) = p_{24} = 0.$$

Для последующих значений k вероятности состояний вычис-ляются с учетом того, что в предыдущий момент (для предыдущего значения k) система S могла находиться в любом из состояний S_i $(i=1,2,3,4)$ с соответствующими вероятностями $p_i(1)$. Поэтому используется соотношение (3.1), а переходные вероятности берутся
из соответствующего столбца матрицы Π.

$$p_1(2) = p_1(1)p_{11} + p_2(1)p_{21} + p_3(1)p_{31} + p_4(1)p_{41} =$$
$$= 0.5 \cdot 0.5 + 0.1 \cdot 0.5 + 0.4 \cdot 0 + 0 \cdot 0 = 0.30;$$
$$p_2(2) = p_1(1)p_{12} + p_2(1)p_{22} + p_3(1)p_{32} + p_4(1)p_{42} =$$
$$= 0.5 \cdot 0.3 + 0.1 \cdot 0.1 = 0.16;$$
$$p_3(2) = p_1(1)p_{13} + p_2(1)p_{23} + p_3(1)p_{33} + p_4(1)p_{43} =$$
$$= 0.5 \cdot 0.2 + 0.1 \cdot 0.4 + 0.4 \cdot 0.4 = 0.30;$$

$$p_4(2) = p_1(1)p_{14} + p_2(1)p_{24} + p_3(1)p_{34} + p_4(1)p_{44} =$$
$$= 0.4 \cdot 0.6 = 0.24;$$
$$p_1(3) = p_1(2)p_{11} + p_2(2)p_{21} + p_3(2)p_{31} + p_4(2)p_{41} =$$
$$= 0.3 \cdot 0.5 + 0.16 \cdot 0.5 = 0.23;$$
$$p_2(3) = p_1(2)p_{12} + p_2(2)p_{22} + p_3(2)p_{32} + p_4(2)p_{42} =$$
$$= 0.3 \cdot 0.3 + 0.16 \cdot 0.1 + 0.24 \cdot 0.8 = 0.29;$$
$$p_3(3) = p_1(2)p_{13} + p_2(2)p_{23} + p_3(2)p_{33} + p_4(2)p_{43} =$$
$$= 0.3 \cdot 0.2 + 0.16 \cdot 0.4 + 0.3 \cdot 0.4 = 0.244;$$
$$p_4(3) = p_1(2)p_{14} + p_2(2)p_{24} + p_3(2)p_{34} + p_4(2)p_{44} =$$
$$= 0.3 \cdot 0.6 + 0.24 \cdot 0.2 = 0.228.$$

Выполняя вычисления $p_i(k)$, необходимо для каждого k проверять правильность выкладок, помня, что

$$\sum_{i=1}^{4} p_i(k) = 1 .$$

Неоднородная марковская цепь. Для неоднородной марковской цепи вероятности переходов p_{ij} системы S меняются от шага к шагу. Если $p_{ij}(k)$ – вероятность перехода системы из состояния S_i в состояние S_j на k-м шаге, то ее можно записать как условную вероятность

$$p_{ij}(k) = P(S_j(k) / S_i(k-1)) .$$

Пусть заданы матрицы вероятностей перехода на каждом шаге. Тогда на основании соотношения (3.1) можно получить выражение для вычисления вероятности того, что система S после k-го шага окажется в состоянии S_i:

$$p_i(k) = \sum_{j=1}^{n} p_j(k-1) \cdot p_{ji}(k), \qquad (i = \overline{1,n}) . \tag{3.2}$$

Выражение (3.2) отличается от (3.1) тем, что в нём участвуют вероятности переходов $p_{ij}(k)$ из состояния в состояние, зависящие от номера шага k. То есть для каждого k должна быть задана своя матрица переходных вероятностей $\Pi(k)$ для всех k от 1 до n.

8.2. Непрерывная цепь Маркова

Для описания таких процессов, когда в марковской цепи происходят переходы из состояния в состояние в случайные моменты времени, используется схема марковского случайного процесса с дискретными состояниями и непрерывным временем, называемая **непрерывной цепью Маркова.**

Как определить вероятности состояний $p_i(t)$ для любого t? Для этого нужно знать характеристики процесса, аналогичные переходным вероятностям для марковской цепи. Вместо переходных вероятностей p_{ij} в данном случае используются плотности вероятностей перехода λ_{ij}.

Если система S в момент t находится в состоянии S_i и если Δt – элементарный промежуток времени, примыкающий к моменту t, то **плотностью вероятности перехода** λ_{ij} (соотношение (3.3)) называется предел отношения вероятности перехода $p_{ij}(\Delta t)$ системы за время Δt из состояния S_i в состояние S_j к длине промежутка Δt:

$$\lambda_{ij} = \lim_{\Delta t \to 0} \frac{p_{ij}(\Delta t)}{\Delta t} . \qquad (3.3)$$

Из выражения (3.3) следует, что для малых Δt вероятность перехода $p_{ij}(\Delta t)$ (с точностью до бесконечно малых высшего порядка) равна $\lambda_{ij}\,\Delta t$:

$$p_{ij}(\Delta t) \approx \lambda_{ij}\,\Delta t . \qquad (3.4)$$

Если все плотности вероятностей перехода λ_{ij} не зависят от t, марковский процесс называется **однородным**, в противном случае – **неоднородным**.

Уравнения Колмогорова для вероятностей состояний. *Если известны плотности вероятностей перехода λ_{ij} для всех пар состояний S_i и S_j, то, построив граф состояний системы* S, *удобно против каждой дуги графа ставить соответствующее значение λ_{ij}. Получится размеченный граф состояний (рис. 3.5). Имея в распоряжении размеченный граф состояний системы, легко построить математическую модель данного процесса, а по модели определить вероятности состояний $p_i(t)$. Для этого составляются и решаются так называемые уравнения Колмогорова. Методика их составления может быть легко продемонстрирована на конкретном примере.*

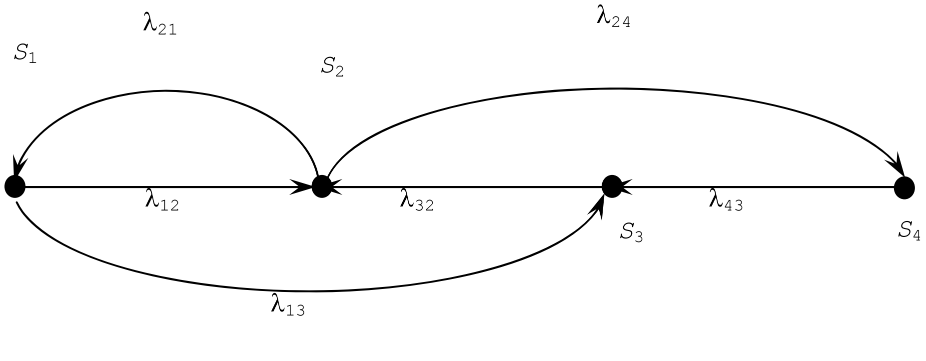


Рис. 3.5. Граф состояний системы S

Пусть система S имеет четыре возможных состояния: S_1, S_2, S_3 и S_4 (рис. 3.5). Как найти одну из вероятностей состояний, например, $p_1(t)$, т.е. какова вероятность того, что в момент t система будет в состоянии S_1? Для этого времени t дается малое приращение $t+\Delta t$ и находится вероятность $p_1(t+\Delta t)$ того, что в момент $t+\Delta t$ система будет в состоянии S_1. Это может произойти двумя способами (в соответствии с рис. 3.5):

1) если в момент t система была в состоянии S_1, и за Δt она не вышла из него;

2) если в момент t система находилась в состоянии S_2, и за Δt она перешла из S_2 в S_1.

Других путей перехода системы S в состояние S_1 за промежуток Δt нет (в соответствии с графом рис. 3.5). Для первого варианта перехода искомая вероятность равна произведению вероятности $p_1(t)$ на условную вероятность того, что из состояния S_1 система за время Δt не перейдет ни в S_2, ни в S_3. Вероятность того, что за Δt система выйдет из состояния S_1 в соответствии с выражением (3.4), равна

$$(\lambda_{12} + \lambda_{13})\Delta t,$$

а вероятность того, что не выйдет:

$$1 - (\lambda_{12} + \lambda_{13})\Delta t.$$

Тогда для первого случая вероятность равна:

$$p_1(t)\ (1 - (\lambda_{12} + \lambda_{13})\Delta t).$$

Для второго случая – равна вероятности того, что в момент времени t система была в S_2, а за Δt она перешла в S_1, т.е. она равна:

$$p_2(t)\lambda_{21}\,\Delta t.$$

По правилу сложения вероятность

$$p_1(t+\Delta t) = p_1(t)(1 - (\lambda_{12} + \lambda_{13})\Delta t) + p_2(t)\lambda_{21}\Delta t.$$

Перенеся $p_1(t)$ в левую часть полученного уравнения и разделив обе части его на Δt, можно записать:

$$\frac{p_1(t+\Delta t) - p_1(t)}{\Delta t} = \lambda_{21} p_2(t) - (\lambda_{12} + \lambda_{13}) p_1(t).$$

Переходя к пределу $(\Delta t \to 0)$, получаем дифференциальное уравнение:

$$\frac{dp_1}{dt} = \lambda_{21} p_2 - (\lambda_{12} + \lambda_{13}) p_1. \qquad (3.5)$$

Подобным образом можно вывести еще одно уравнение для вероятности состояния $p_2(t)$. Для этого рассматривают состояние S_2 и находят вероятность $p_2(t+\Delta t)$ одним из следующих способов:

1) в момент t система была в состоянии S_2 и за Δt она не перешла из этого состояния ни в S_1, ни в S_4;

2) в момент t система была в S_1 и за Δt она перешла в S_2;

3) в момент t она была в состоянии S_3 и за Δt перешла в S_2.

Вероятность первого варианта:

$$p_2(t)(1 - (\lambda_{21} + \lambda_{24})\ \Delta t),$$

вероятность второго варианта:

$$p_1(t)\lambda_{12}\Delta t,$$

вероятность третьего варианта:

$$p_3(t)\lambda_{32}\Delta t.$$

Складывая перечисленные вероятности и приравнивая их к $p_2(t+\Delta t)$, получают:

$$p_2(t+\Delta t) = p_2(t)(1 - (\lambda_{21} + \lambda_{24})\ \Delta t) + p_1(t)\lambda_{12}\Delta t + p_3(t)\lambda_{32}\Delta t.$$

Выделим в этом равенстве разностное отношение

$$\frac{p_2(t+\Delta t) - p_2(t)}{\Delta t} = -(\lambda_{21} + \lambda_{24})p_2 + p_1\lambda_{12} + p_3\lambda_{32}.$$

Переходя к пределу при $\Delta t \to 0$ получим дифференциальное уравнение:

$$\frac{dp_2}{dt} = \lambda_{12}p_1 + \lambda_{32}p_3 - (\lambda_{21} + \lambda_{24})p_2. \qquad (3.6)$$

Получая по изложенной методике уравнения для состояний S_3 и S_4 и присоединяя к ним уравнения (3.5) и (3.6), можно записать систему дифференциальных уравнений для вероятностей состояний:

$$\frac{dp_1}{dt} = \lambda_{21}p_2 - (\lambda_{12} + \lambda_{13})p_1,$$

$$\frac{dp_2}{dt} = \lambda_{12}p_1 + \lambda_{32}p_3 - (\lambda_{21} + \lambda_{24})p_2, \qquad (3.7)$$

$$\frac{dp_3}{dt} = \lambda_{12}p_1 + \lambda_{43}p_4 - \lambda_{32}p_3,$$

$$\frac{dp_4}{dt} = \lambda_{24}p_2 - \lambda_{43}p_4$$

Уравнения (3.7) называются **уравнениями Колмогорова.** Это система четырех линейных дифференциальных уравнений с четырьмя неизвестными функциями $p_1,\ p_2, p_3,\ p_4$. Начальные условия задаются в зависимости от того, каково было начальное состояние системы S. Например, если в начальный момент времени (при t=0) система была в состоянии S_i, то $p_i(0)=1$, а все остальные вероятности равны нулю:

$$p_j(0)=0 \quad (i \neq j).$$

Так, например, уравнения (3.7) можно решать при начальных условиях:

$$p_1(0)=1, \quad p_2(0)=p_3(0)=p_4(0)=0.$$

Нужно отметить, что все четыре уравнения системы (3.7) можно не записывать, так как можно воспользоваться условием $p_1+p_2+p_3+p_4=1$. Поэтому любую из вероятностей можно выразить через остальные и получить систему трех уравнений.

Анализируя систему уравнений (3.7), можно вывести общее правило формирования таких уравнений непосредственно по размеченному графу состояний, пригодное для любой непрерывной марковской цепи.

В левой части каждого уравнения находится производная вероятности состояния, а правая часть содержит столько слагаемых, сколько стрелок в графе связано с данным состоянием. Если стрелка направлена из состояния, соответствующее слагаемое имеет знак «минус», если в состояние – знак «плюс». Каждое слагаемое равно произведению плотности вероятности перехода, соответствующей данной стрелке, на вероятность того состояния, из которого исходит стрелка.

Пример 3.3. Для графа состояний системы S, представленного на рис. 3.6, составить систему дифференциальных уравнений Колмогорова и написать начальные условия, учитывая, что при t = 0 система была в состоянии S_2.

Искомая система уравнений:

$$\frac{dp_1}{dt}=-\lambda_{12}p_1+\lambda_{21}p_2+\lambda_{41}p_4,$$

$$\frac{dp_2}{dt}=-(\lambda_{21}+\lambda_{23})p_2+\lambda_{12}p_1+\lambda_{32}p_3+\lambda_{42}p_4,$$

$$\frac{dp_3}{dt}=-(\lambda_{32}+\lambda_{34})p_3+\lambda_{23}p_2,$$

$$\frac{dp_4}{dt}=-(\lambda_{41}+\lambda_{42})p_4+\lambda_{34}p_3.$$

Начальные условия: при t = 0: p_2 = 1, $p_1 = p_3 = p_4 = 0$.

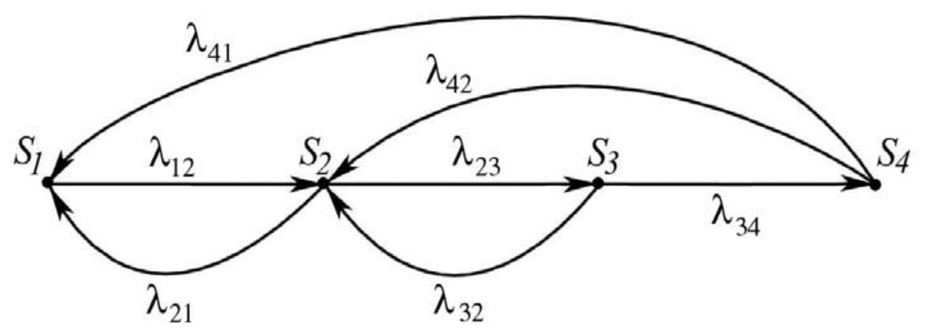


Рис. 3.6. Граф состояний системы *S* примера 3.3

8.3. Предельные вероятности состояний

Если в системе *S* протекает марковский случайный процесс с непрерывным временем (см. рис. 3.5.), то считается, что реализуется непрерывная цепь Маркова. При λ_{ij} = const считается, что потоки событий в системе простейшие (стационарные пуассоновские).

Пусть имеется непрерывная цепь Маркова. Используя методику составления системы дифференциальных уравнений Колмогорова и решая эту систему при заданных начальных условиях, можно найти вероятности состояний как функции времени

$p_1(t)$, $p_2(t)$, ..., $p_n(t)$, и $\sum_{i=1}^{n} p_i(t) = 1$ при любом *t*.

При этом возникает вопрос, что будет происходить с системой *S* при *t* → ∞? Имеют ли при этом функции $p_i(t)$ пределы? Эти пределы, если они существуют, называются **предельными** («финальными») **вероятностями состояний.** Их существование устанавливается эргодической теоремой Маркова и теоремой о предельных вероятностях состояний. Теорема Маркова справедлива для транзитивных процессов.

Марковский случайный процесс называется **транзитивным,** если существует такой промежуток времени *t*, в течение которого возможен переход системы *S* из любого состояния в любое другое. Таким образом, для транзитивного процесса существует такое *t* > 0, что переходные вероятности $p_{ij}(t) > 0$ ($i, j = 1, 2, ..., n$), *t* – промежуток времени.

Показанный на рис. 3.6, граф состояний системы удовлетворяет условию транзитивности: из любого состояния можно рано или поздно перейти в любое другое, а на рис. 3.7 – не удовлетворяет условию транзитивности.

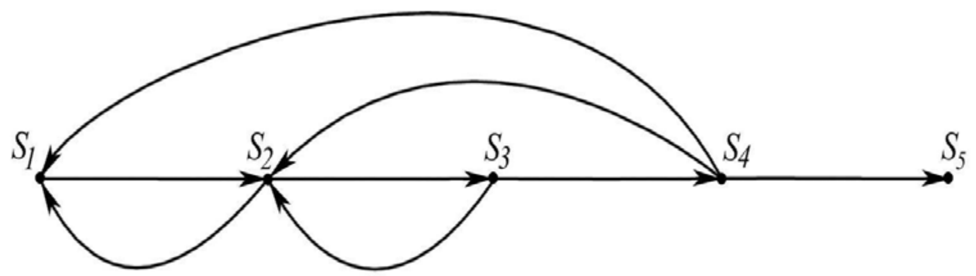


Рис. 3.7. Граф, не удовлетворяющий условию транзитивности

<u>Эргодическая теорема Маркова.</u> Для любого транзитивного марковского процесса

$$\lim_{t\to 0} p_{ij}(t) \quad (i,j=\overline{1,n})$$

существует и не зависит от «*i*».

Физический смысл теоремы следующий: вероятность того, что система *S* находится в каком-либо состоянии S_j, практически не зависит от того, в каком состоянии система находилась в далеком прошлом.

<u>Теорема о предельных вероятностях состояний.</u> Для того чтобы вероятности $p_i(t)$ $(i=\overline{1,n})$ **при** $t\to\infty$ **стремились к независящим от начальных данных числам** p_j $(j=\overline{1,n})$, **необходимо и достаточно, чтобы к тем же пределам стремились соответственно переходные вероятности** $p_{ij}(t)$ **при любом значении** i.

В системе *S* при $t\to\infty$ устанавливается предельный стационарный режим, заключающийся в том, что система случайным образом меняет свои состояния, но вероятность каждого из них уже не зависит от времени. Физический смысл этой вероятности следующий: она представляет собой среднее относительное время пребывания системы *S* в данном состоянии.

Как вычислить предельные вероятности состояний $p_i\ (i=\overline{1,n})$**?** В предельном (установившемся) режиме все вероятности постоянны. Следовательно, их производные равны нулю. Поэтому, если в уравнениях Колмогорова, описывающих вероятности состояний системы, положить все производные равными нулю, то эта система превратится в систему алгебраических уравнений. Совместно с нормировочным условием $\sum_{i=1}^{n} p_i = 1$ эти уравнения дают возможность вычислить все предельные вероятности: $p_1,\ p_2, \ldots,\ p_n$.

Пример 3.4. Граф системы *S* с тремя возможными состояниями S_1, S_2, S_3 представлен на рис. 3.8.

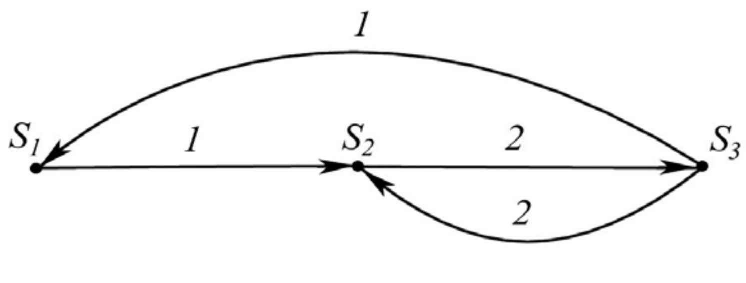


Рис. 3.8. Граф системы примера 3.4

Необходимо составить уравнения Колмогорова, затем записать систему линейных алгебраических уравнений и решить их, найдя значения предельных вероятностей состояний $p_i\ (i=1,2,3)$.

Уравнения Колмогорова:

$$\begin{cases} \dfrac{dp_1}{dt} = -p_1 + p_3, \\ \dfrac{dp_2}{dt} = -2p_2 + p_1 + 2p_3, \\ \dfrac{dp_3}{dt} = -(1+2)p_3 + 2p_2. \end{cases} \tag{3.8}$$

Система линейных алгебраических уравнений записывается приравниванием нулю левых частей уравнений (3.8):

$$\begin{cases} 0 = -p_1 + p_3, \\ 0 = -2p_2 + p_1 + 2p_3, \\ 0 = -3p_3 + 2p_2. \end{cases} \tag{3.9}$$

Выразим из 1-го уравнения $p_1 = p_3$, а из 3-го уравнения $p_2 = \dfrac{2}{3}p_3$ и подставим их в нормировочное уравнение $p_1 + p_2 + p_3 = 1$. Откуда окончательно получим: $p_1 = \dfrac{2}{7}, \quad p_2 = \dfrac{3}{7}, \quad p_3 = \dfrac{2}{7}$

При решении системы (3.9) не использовалось второе уравнение. Оно является следствием остальных (1-го и 3-го): если сложить все три уравнения, то получится тождественный нуль.

Чтобы в дальнейшем сразу записывать такие уравнения, полезно пользоваться следующим мнемоническим правилом. **Для каждого состояния сумма слагаемых, соответствующих входящим стрелкам, равна сумме слагаемых, соответствующих выходящим стрелкам («что втекает, то и вытекает»); каждое слагаемое равно интенсивности потока событий, переводящего систему по данной стрелке, умноженной на вероятность того события, из которого выходит стрелка.**

Пример 3.5. Граф состояний системы *S* представлен на рис. 3.9. Составить систему алгебраических уравнений для нахождения предельных вероятностей состояний.

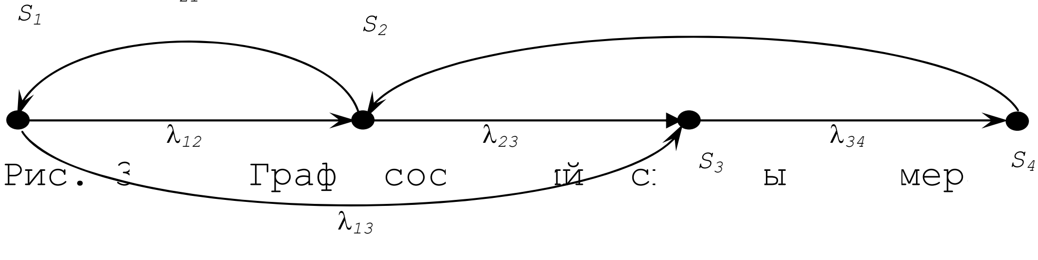


Рис. 3 Граф сос ий с ы мер 5

$$\lambda_{21}p_2 = (\lambda_{12} + \lambda_{13})p_1,$$
$$\lambda_{12}p_1 + \lambda_{42}p_4 = (\lambda_{21} + \lambda_{23})p_2,$$
$$\lambda_{13}p_1 + \lambda_{23}p_2 = \lambda_{34}p_3,$$
$$\lambda_{34}p_3 = \lambda_{42}p_4.$$

8.4. Процесс «размножения и гибели»

На практике часто встречаются ситуации, когда некоторые случайные процессы, относящиеся, например, к различным областям знаний, имеют одинаковые графы состояний. Тогда, если рассматриваемые процессы являются непрерывными цепями Маркова и имеют одинаковые графы состояний, можно найти аналитические выражения для предельных вероятностей состояний типовых процессов в общем виде, а затем подставить вместо λ_{ij} соответствующие значения.

Пусть некоторый марковский случайный процесс, протекающий в системе S, может быть представлен графом состояний, как на рис. 3.10, т.е. все состояния можно вытянуть в одну цепочку, в которой каждое из средних состояний $(S_1, S_2, ..., S_{n-1})$ связано прямой и обратной связью (стрелкой) с каждым из соседних состояний, а крайние состояния (S_0 и S_n) – только с одним соседним состоянием.

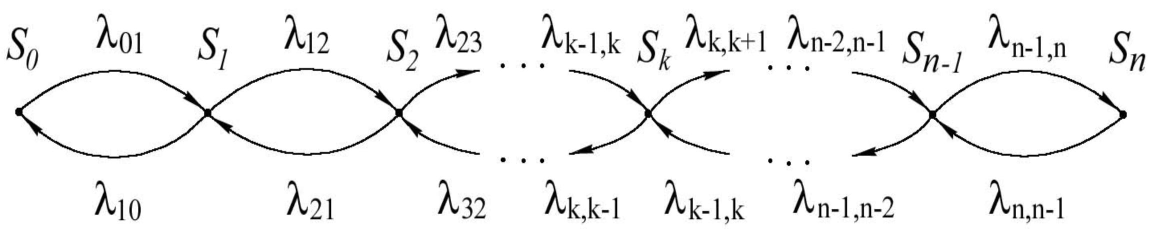


Рис. 3.10. Граф процесса «размножения и гибели»

Процессы такого рода описывают, в частности, изменения численности биологической популяции, что и послужило основанием для появления термина – **процесс «размножения и гибели»**.

Схема «размножения и гибели» встречается во многих практических задачах, поэтому целесообразно определить аналитические выражения предельных вероятностей состояний системы, граф которой приведен на рис. 3.10, и в дальнейшем пользоваться готовыми формулами.

Согласно правилу «что втекает, то и вытекает», записывается система уравнений для нахождения предельных вероятностей состояний процесса «размножения и гибели».

Для состояния S_0 получаем:

$$\lambda_{01}p_0 = \lambda_{10}p_1, \tag{3.10}$$

а для состояния S_1:

$$\lambda_{12}p_1 + \lambda_{10}p_1 = \lambda_{01}p_0 + \lambda_{21}p_2. \quad (3.11)$$

Учитывая (3.10), из выражения (3.11) получают для S_1:

$$\lambda_{21}p_2 = \lambda_{12}\, p_1\,, \quad (3.12)$$

а далее аналогично:

– для S_2 $$\lambda_{32}p_3 = \lambda_{23}p_2, \quad (3.13)$$

– для S_{k-1} $$\lambda_{k,k-1}p_k = \lambda_{k-1,k}p_{k-1}, \quad (3.14)$$

– для S_{n-1} $$\lambda_{n,n-1}p_n = \lambda_{n-1,n}p_{n-1}, \quad (3.15)$$

Запишем нормировочное условие:

$$p_0 + p_1 + \ldots + p_n = 1\,. \quad (3.16)$$

Итак, получена система уравнений (3.10), (3.12) – (3.15) и нормировочное условие (3.16). Из уравнения (3.10) находим p_1:

$$p_1 = \frac{\lambda_{01}}{\lambda_{10}}p_0, \quad (3.17)$$

из уравнения (3.12) с учетом (3.17) находится p_2:

$$p_2 = \frac{\lambda_{01}\lambda_{12}}{\lambda_{21}\lambda_{10}}p_0, \quad (3.18)$$

из (3.13) с учетом (3.18) получают p_3:

$$p_3 = \frac{\lambda_{01}\lambda_{12}\lambda_{23}}{\lambda_{32}\lambda_{21}\lambda_{10}}p_0$$

и так далее.

Таким образом, для любого k имеет место формула:

$$p_k = \frac{\lambda_{01}\;\lambda_{12}\ldots\lambda_{k-2,k-1}\;\lambda_{k-1,k}}{\lambda_{k,k-1}\;\lambda_{k-1,k-2}\ldots\lambda_{21}\;\lambda_{10}}\,p_0, \qquad k = \overline{1,n}.$$

(3.19)

Структура формулы (3.19) следующая. В числителе находится произведение всех λ_{ij}, стоящих у стрелок, направленных слева направо, от S_0 и до той, которая идет в S_k, а в знаменателе –произведение всех λ_{ij}, идущих в обратном направлении – справа налево от S_k до S_0.

Если все p_i (кроме случая, когда i=0) подставить в выражение (3.16), то можно получить выражение для p_0:

$$p_0+\frac{\lambda_{01}}{\lambda_{10}}p_0+\frac{\lambda_{01}\lambda_{12}}{\lambda_{21}\lambda_{10}}p_0+\frac{\lambda_{01}\lambda_{12}\lambda_{23}}{\lambda_{32}\lambda_{21}\lambda_{10}}p_0+\frac{\lambda_{01}\lambda_{12}\ldots\lambda_{n-1,n}}{\lambda_{n,n-1}\ldots\lambda_{21}\lambda_{10}}p_0=1,$$

$$p_0=\left(1+\frac{\lambda_{01}}{\lambda_{10}}+\frac{\lambda_{01}\lambda_{12}}{\lambda_{21}\lambda_{10}}+\ldots+\frac{\lambda_{01}\lambda_{12}\ldots\lambda_{n-1,n}}{\lambda_{n,n-1}\lambda_{n-1,n-2}\ldots\lambda_{10}}\right)^{-1}. \quad (3.20)$$

Формулы (3.19) и (3.20) дают общее решение задачи «размножения и гибели». Они будут широко использоваться в следующей главе при рассмотрении конкретных моделей СМО.

Пример 3.6. Граф состояний системы *S* представлен на рис. 3.11. Определить предельные вероятности состояний.

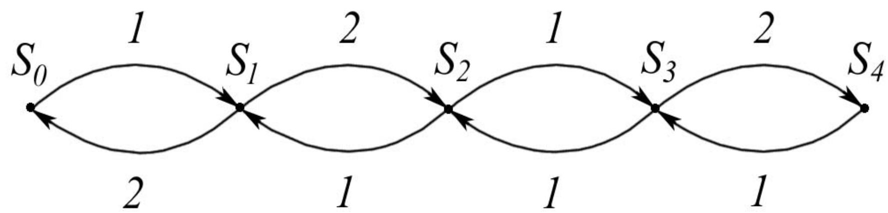


Рис. 3.11. Граф состояний системы примера 3.6

Процесс, описанный графом на рис. 3.11, транзитивный, значит, предельные вероятности существуют. В то же время структура графа показывает, что это процесс «размножения и гибели».

Выпишем по графу интенсивности

$$\lambda_{01}=1,\quad \lambda_{12}=2,\quad \lambda_{23}=1,\quad \lambda_{34}=2,$$
$$\lambda_{10}=2,\quad \lambda_{21}=1,\quad \lambda_{32}=1,\quad \lambda_{43}=1.$$

Пользуясь формулами (3.19) и (3.20) можно рассчитать p_i (i=1, 2, 3, 4):

$$p_0 = \left(1 + \frac{1}{2} + \frac{1\cdot 2}{1\cdot 2} + \frac{1\cdot 2\cdot 1}{1\cdot 1\cdot 2} + \frac{1\cdot 2\cdot 1\cdot 2}{1\cdot 1\cdot 2\cdot 1}\right)^{-1} = \frac{2}{11},$$

$$p_1 = \frac{1}{2} p_0 = \frac{1}{11},$$

$$p_2 = \frac{1\cdot 2}{2\cdot 1} p_0 = \frac{2}{11},$$

$$p_3 = \frac{1\cdot 2\cdot 1}{2\cdot 1\cdot 1} p_0 = \frac{2}{11},$$

$$p_4 = \frac{1\cdot 2\cdot 1\cdot 2}{2\cdot 1\cdot 1\cdot 1} p_0 = \frac{4}{11}.$$

Проверка показывает, что $\sum_{i=1}^{4} p_i = 1$.

Глава 9

МОДЕЛИРОВАНИЕ СМО АНАЛИТИЧЕСКИМИ МЕТОДАМИ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ

Основная задача, стоящая перед теорией массового обслуживания, заключается в установлении зависимости между характером потока заявок, числом каналов, их производительностью, правилами работы СМО и успешностью (эффективностью) обслуживания. Решение задач анализа СМО связано с привлечением различных методов оптимизации: линейного или нелинейного программирования, динамического программирования, теории игр, метода имитационного моделирования и т. д.

Математический анализ работы СМО значительно упрощается, если случайный процесс, протекающий в системе, является марковским. В этом случае работу СМО довольно просто удается описать с помощью дифференциальных уравнений, а в предельном случае – линейных алгебраических, и выразить в явном виде основные характеристики эффективности обслуживания через параметры СМО и потока заявок.

В качестве характеристик эффективности обслуживания могут применяться различные величины и функции, например: среднее количество заявок, обслуживаемых в единицу времени; средний процент заявок, получающих отказ и покидающих систему не обслуженными; вероятность того, что поступившая заявка немедленно будет принята к обслуживанию; среднее время ожидания в очереди; закон распределения времени ожидания; среднее количество заявок, находящихся в очереди и т. д.

9.1. Многоканальная СМО с отказами

Система массового обслуживания, описываемая схемой $M|M|n|m=0$, является простейшей из всех СМО – это многоканальная система с отказами. Конечно, одноканальная СМО вида $M|M|1|m=0$, являясь частным случаем многоканальной, имеет более простые характеристики, но они могут быть получены из характеристик многоканальной СМО при $n = 1$.

В систему поступает простейший, т. е. пуассоновский поток заявок на обслуживание с интенсивностью λ, где λ – число заявок, поступающих на обслуживание в единицу времени. Интервалы между моментами поступления заявок предполагаются случайными и распределенными по показательному закону

$$f(t) = \lambda e^{-\lambda t}, \quad t > 0 .$$

Заявка, заставшая каналы занятыми, покидает систему не обслуженной.

Обслуживание заявки продолжается в течение случайного времени $T_{обс}$, распределенного по показательному закону с интенсивностью обслуживания

$$\mu = \frac{1}{T_{обс}} : \qquad f(\tau) = \mu e^{-\mu \tau}, \quad \tau > 0 .$$

Из этого распределения времени обслуживания следует, что поток обслуживания простейший.

Основными характеристиками такой СМО являются: абсолютная пропускная способность СМО A, относительная пропускная способность СМО q и вероятность отказа $P_{отк}$. Для нахождения этих характеристик СМО рассматривается как физическая система S, которая находится поочередно в одном из n+1 состояний, а именно:

S_0 – все каналы свободны;

S_1 – занят один канал, остальные $n - 1$ свободны;

...

S_k – заняты k каналов, остальные $n - k$ свободны;

...

S_n – заняты все n каналов.

Размеченный граф состояний исследуемой системы представлен на рис. 3.12. Всего в графе $n + 1$ состояние. Стрелки направо переводят систему из состояния S_j в другое состояние, которое определяется началом обслуживания пришедшей заявки очередным (j+1)-м каналом. При этом плотность переходной вероятности из одного состояния в другое (последующее) определяется интенсивностью λ поступления заявок в систему, то есть каждая поступающая заявка переводит СМО из состояния S_j в состояние S_{j+1} с интенсивностью λ.

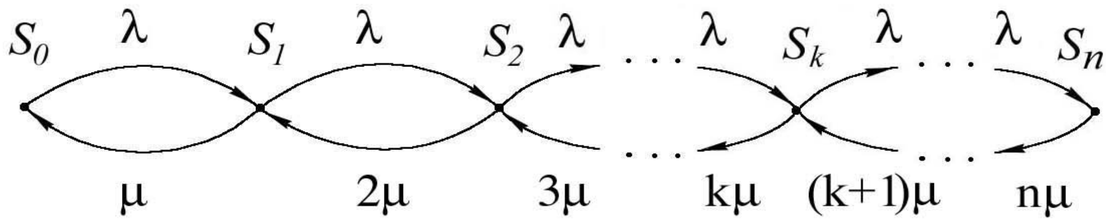


Рис. 3.12. Граф состояний многоканальной СМО с отказами

Как размечаются переходы системы справа налево? Если система находится в состоянии S_1, то, как только закончится обслуживание заявки, занимающей этот канал, система перейдет в состояние S_0. Этот переход будет определяться интенсивностью μ обслуживания каналов. Если система находится в состоянии S_2, то есть в системе обслуживанием заявок занято два канала, то интенсивность перехода из S_2 в S_1 вырастает вдвое (2μ). Очевидно, если обслуживанием занято k каналов, то интенсивность этого обслуживания будет в k раз выше, чем при обслуживании одним каналом ($k\,\mu$).

Итак, процесс, протекающий в данной СМО, является частным случаем рассмотренного ранее процесса «размножения и гибели». Пользуясь формулами (3.19) и (3.20), можно записать выражения для расчёта вероятностей состояний:

$$p = \left(1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{\mu^2 2!} + \frac{\lambda^3}{\mu^3 3!} + \ldots + \frac{\lambda^n}{\mu^n n!}\right)^{-1},$$

$$p_1 = \frac{\lambda}{\mu} p_0, \;\; p_2 = \frac{\lambda^2}{\mu^2 2!} p_0, \;\; p_3 = \frac{\lambda^3}{\mu^3 3!} p_0, \;\; \ldots, \;\; p_n = \frac{\lambda^n}{\mu^n n!} p_0.$$

Вводя величину $\rho = \dfrac{\lambda}{\mu}$, называемую приведенной интенсивностью потока заявок, то есть среднее число заявок, приходящих в СМО за среднее время обслуживания одной заявки, можно получить формулы для расчета предельных вероятностей состояний (при $k = \overline{1,n}$):

$$p_0 = \left(1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} + \ldots + \frac{\rho^n}{n!}\right)^{-1}.$$

Или в более компактной записи:

$$p_0 = \left(\sum_{\alpha=0}^{n} \frac{\rho^n}{\alpha!}\right)^{-1} \qquad (3.21)$$

$$p_k = \frac{\rho^k}{k!} p_0, \quad k = \overline{1,n}.$$

Формулы (3.21) носят название формул Эрланга. Они выражают предельные вероятности всех состояний системы в зависимости от параметров n, λ и μ. Пользуясь ими, можно найти характеристики эффективности СМО: $P_{отк}$, q и A.

Вычисление $P_{отк}$ сводится к определению вероятности занятости всех n каналов системы, т.е. к определению вероятности n-го состояния:

$$P_{отк} = p_n = \frac{\rho^n}{n!} p_0. \tag{3.22}$$

Относительная пропускная способность q – это вероятность того, что заявка будет принята к обслуживанию. Так как обслуживание и отказ это полная группа несовместных событий, то

$$q = 1 - P_{отк} = 1 - p_n. \tag{3.23}$$

Абсолютная пропускная способность A – это среднее число заявок, обслуженных системой в единицу времени:

$$A = \lambda q = \lambda(1 - p_n). \tag{3.24}$$

Кроме перечисленных характеристик (3.22) ... (3.24), можно определить еще целый ряд важных величин. Среднее количество занятых каналов $\bar{k}$ может быть вычислено как математическое ожидание $M(k)$ случайной величины k. Дискретная случайная величина $k = \{0, 1, 2, \ldots, n\}$ (количество занятых каналов) наступает с вероятностями p_i (табл. 3.1).

Таблица 3.1

k	0	1	2	…	k	…	n
p_i	p_0	p_1	p_2	…	p_k	…	p_n

Тогда математическое ожидание случайной величины

$$M(k) = 0\ p_0 + 1\ p_1 + 2\ p_2 + 3\ p_3 + \ldots + k\ p_k + \ldots + n\ p_n$$

и, если уже известны все p_i $(i = \overline{0,n})$, то $\bar{k}$ легко вычисляется.

Но среднее количество занятых каналов $\bar{k}$ можно выразить и через абсолютную пропускную способность A, где A есть среднее число заявок, обслуживаемых в единицу времени. Один занятый канал обслуживает в среднем за единицу времени μ заявок, а среднее число занятых каналов

$$\overline{k} = \frac{A}{\mu} = \rho(1 - p_n). \qquad (3.25)$$

Пример 3.7. Задана четырёхканальная СМО с отказами (M | M | 4 | 0), λ = 2 (c^{-1}), μ = 1 (c^{-1}). Вычислить p_i (i= 0, 1, 2, 3, 4), $P_{отк}$, q, A, $\overline{k}$.

В учебных целях для решения этой задачи (и впредь в данном учебном пособии в последующих примерах) необходимо последовательно реализовывать четыре этапа:

1. Построение графа состояний.
2. Разметка графа.
3. Вычисление p_i .
4. Вычисление показателей эффективности (в данной задаче - $P_{отк}$, q, A, $\overline{k}$) .

Для правильного построения графа состояний необходимо перечислить возможные состояния:

S_0 - все каналы свободны;
S_1 - 1 канал занят, остальные 3 свободны;
S_2 - 2 канала заняты, остальные 2 свободны;
S_3 - 3 канала заняты, 1 свободен;
S_4 - все каналы заняты.

Размеченный граф, в котором n + 1 = 5 состояний, представлен на рис. 3.13. Это говорит о том, что уже реализованы первые два этапа.

Приведенная интенсивность обслуживания $\rho = \frac{\lambda}{\mu} = 2$.

Результаты реализации последних двух этапов пердствлены ниже.

$$p_0 = (1 + 2 + \frac{4}{2} + \frac{8}{6} + \frac{16}{24})^{-1} = \frac{1}{7},$$

$$p_1 = \frac{2}{7}, p_2 = \frac{2}{7}, p_3 = \frac{4}{21}, p_4 = \frac{2}{21},$$

$$P_{\text{îòê}} = p_4 = \frac{2}{21} = 0.095,$$

$$q = 1 - P_{\text{îòê}} = 0.905,$$

$$A = \lambda q = 1.81(c)^{-1},$$

$$\overline{k} = \rho(1 - p_4) = 1.81.$$

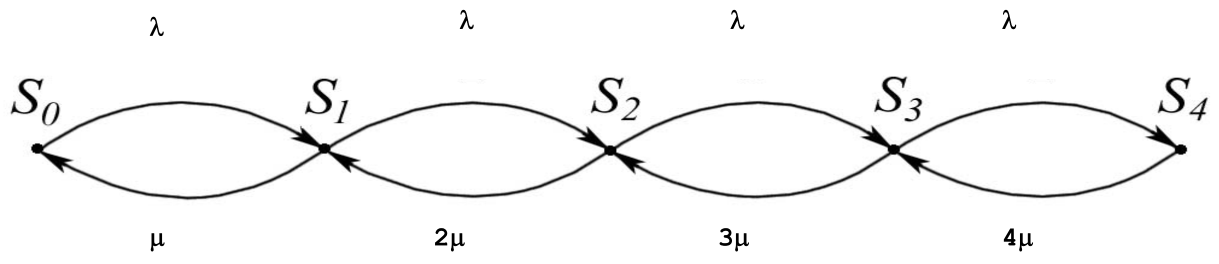


Рис. 3.13. Граф состояний системы примера 3.7

Анализ результатов показывает, что примерно 90 % заявок будет обслуживаться и, сответственно, только 10 % получат отказ. Это получается за счет сравнительно высокой интенсивности обсуживания заявок μ. Вследствие этого в среднем обслуживанием заявок будет занято всего примерно 2 канала, $\bar{k} = 1.81$, а соответственно 2 канала - свободны. Эти результаты приводят к мысли о том, что, возможно, выгодно было бы сократить число обслуживающих каналов. Следовательно, возникает вопрос об оптимизации структуры системы (количества обслуживающих каналов).

В выражениях (3.22) ... (3.25) основным параметром является приведённая интенсивность обслуживания $\rho = \frac{\lambda}{\mu}$. Поэтому любая СМО рассматриваемого типа с n каналами работает одинаково хорошо (или одинаково плохо) при любых λ и μ, для которых неизменна величина ρ. Например, если СМО с каким-то количеством каналов и с производительностью $\mu = 5$ подвергается воздействию потока заявок интенсивности $\lambda = 5$, то такая же СМО с производительностью $\mu = 0.1$, взаимодействующая с потоком $\lambda = 0.1$, имеет те же величины $p_0,\ p_k,\ P_{отк},\ q,\ \bar{k}$ и только A изменяется пропорционально λ.

Как зависят от *n* характеристики эффективности СМО при ρ = const?

При увеличении n вероятности p_0 и p_k, при $k = \overline{1,n}$, уменьшаются, так как

для $n = 1$ $p_0 = (1 + \rho)^{-1}$,
для $n = 2$ $p_0 = (1 + \rho + \rho^2/2!)^{-1}$,
для $n = 3$ $p_0 = (1 + \rho + \rho^2/2! + \rho^3/3!)^{-1}$ и т. д.,

а p_k в соответствии с (3.21) пропорциональны p_0. То есть, при увеличении числа состояний n системы S, осуществляется перераспределение вероятностей состояний при общем ограничении $p_0 + \sum_{k=1}^{n} p_k = 1$.

Таким образом, рост n при $\rho = const$ приводит к улучшению всех характеристик СМО. А именно

$$\text{при } n \to \infty: \quad P_{отк} \to 0, \quad q \to 1, \quad A \to \lambda, \quad \bar{k} \to \rho.$$

При задании верхних пределов для $P_{отк}$ или q можно выбрать конкретное значение n. Пусть, например, $\rho = 1$ и $P_{отк} \le 0.03$. Тогда $\frac{p_0}{n!} \le 0.03$. Отсюда находим $\frac{n!}{p_0} \ge 33$, т. е.

$$n!\left(1+\frac{1}{1!}+\frac{1}{2!}+\frac{1}{3!}+..+\frac{1}{n!}\right)>33.$$

Последнее неравенство начинает выполняться при $n = 4$.

Как зависят от ρ характеристики эффективности СМО при заданном *n*? С увеличением ρ заявки поступают в среднем чаще, чем осуществляется их обслуживание, уменьшается вероятность p_0, сумма вероятностей $p_k \ (k=\overline{1,n})$ соответственно увеличивается, так как

$$p_0+\sum_{k=1}^{n} p_k = 1\,.$$

Таким образом, при возрастании ρ значения $P_{отк} \to 1,$ $q \to 0,$ $\overline{k} \to n\,.$ Поэтому необходимо уменьшать ρ за счет увеличения μ для улучшения показателей эффективности рассматриваемой СМО.

Используя размеченный граф состояний, представленный на рис. 3.14, и методики, описанные в предыдущих параграфах, можно

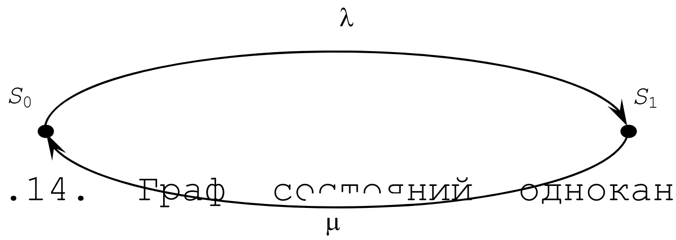


Рис. 3.14. Граф состояний одноканальной СМО

получить характеристики СМО, описы схемой $M \mid M \mid 1 \mid 0$.

Система дифференциальных уравнений Колмогорова записывается по методике, изложенной в § 8.2:

$$\frac{dp_0(t)}{dt} = \mu p_1(t) - \lambda p_0(t),$$

$$\frac{dp_1(t)}{dt} = \lambda p_0(t) - \mu p_1(t),$$

$$p_0(t) + p_1(t) = 1\,.$$

Эту систему можно решить при следующих начальных условиях: в момент времени $t = 0$ система находилась в состоянии S_0, то есть в начальный момент времени канал свободен, тогда $p_0(0) = 1$, $p_1(0) = 0$.

Выражая одну переменную через другую, из системы уравнений Колмогорова можно получить дифференциальное уравнение для определения вероятности $p_0(t)$ состояния S_0:

$$\frac{dp_0(t)}{dt} = -(\lambda+\mu)p_0(t)+\mu.$$

Это линейное уравнение первого порядка. Учитывая начальное условие $p_0(t) = 1$, можно получить его решение в следующем виде:

$$p_0(t)=\frac{\mu}{\lambda+\mu}+\frac{\lambda}{\lambda+\mu}e^{-(\mu+\lambda)t}.$$

Тогда

$$p_1(t)=1-p_0(t)=\frac{\lambda}{\lambda+\mu}-\frac{\lambda}{\lambda+\mu}e^{-(\mu+\lambda)t}.$$

Графики зависимости $p_0(t)$ и $p_1(t)$ от t представлены на рис. 3.15 для различных соотношений λ и μ: а) для $\lambda>\mu$, б) для $\lambda<\mu$, и в) для $\lambda=\mu$. Кривая $p_0(t)$ определяет уменьшающуюся вероятность того, что канал свободен: при $t = 0$ $p_0=1$, и при $t \to \infty$

$$p_0(t)\to\frac{\mu}{\lambda+\mu}. \tag{3.26}$$

Кривая $p_1(t)$, дополняет кривую $p_0(t)$ до единицы, причем $p_1(0)=0$ и при $t \to \infty$

$$p_1(t)\to\frac{\lambda}{\lambda+\mu}. \tag{3.27}$$

Соотношения (3.26) и (3.27) являются предельными вероятностями состояний системы $M \mid M \mid 1 \mid 0$.

Для рассматриваемой одноканальной СМО с отказами показатели эффективности рассчитываются по следующим формулам:

$$\begin{aligned} q &= p_0=\frac{\mu}{\lambda+\mu},\\ A &= \lambda q, \\ P_{отк} &= 1-q=\frac{\lambda}{\lambda+\mu}. \end{aligned} \tag{3.28}$$

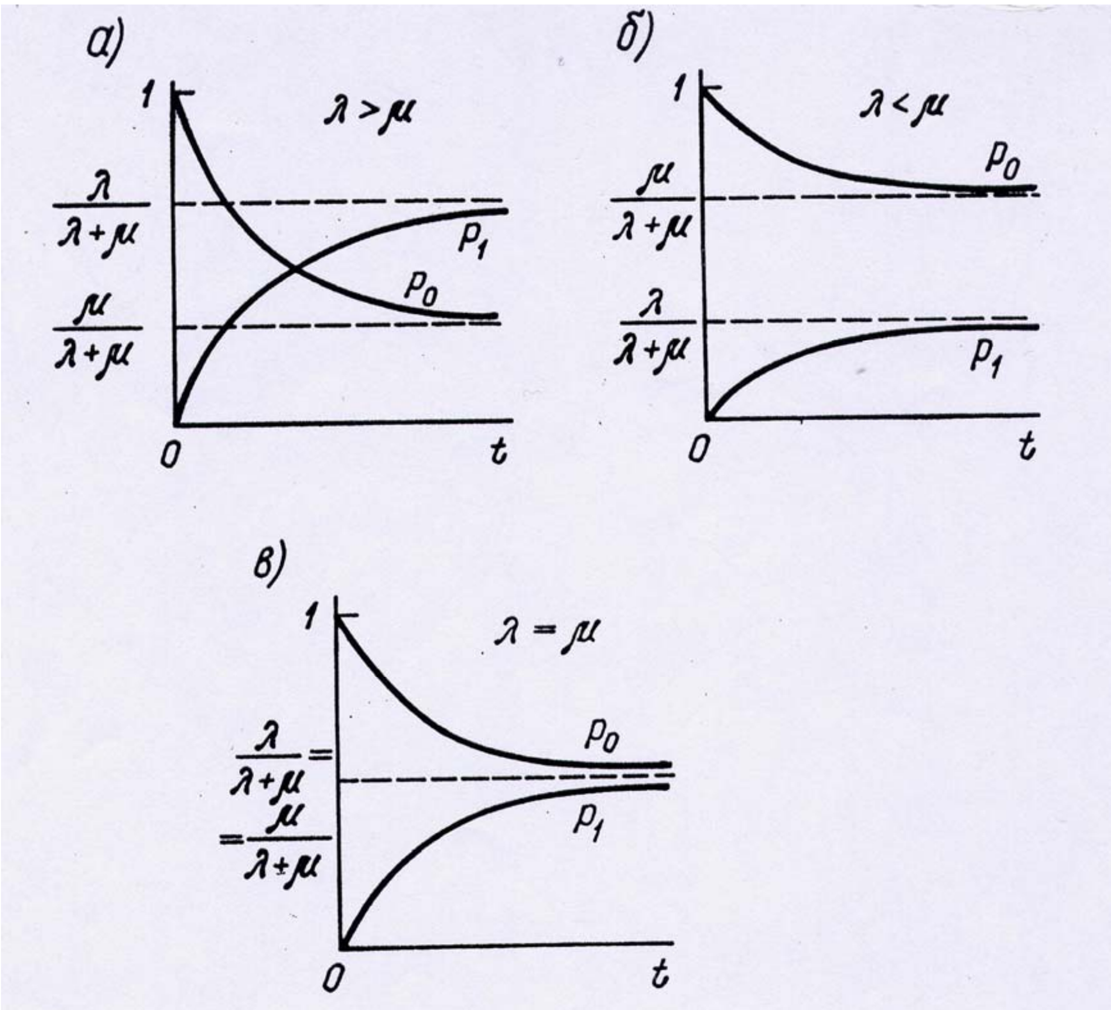


Рис. 3.15. Графики зависимости p_0 и p_1 от t

Абсолютная пропускная способность *A* имеет размерность 1/с и определяет интенсивность потока, образовавшегося на входе канала обслуживания с учетом частоты поступления в канал заявок и интенсивности их обслуживания μ.
Если сравнить формулы (3.21) ... (3.24) с формулами (3.26) ... (3.28), то можно увидеть, что при $n = 1$ первые выражения сводятся ко вторым, что естественно, так как СМО вида $M \mid M \mid 1 \mid 0$ является частным случаем системы $M \mid M \mid n \mid 0$.

Пример 3.8. Система обслуживает каждую заявку в течение 30 минут. Заявки приходят в среднем 1 раз в 40 минут. Если система занята (она одноканальная вида $M \mid M \mid 1 \mid 0$). Определить характеристики этой СМО.

Интенсивность поступления заявок в систему

$$\lambda = 60 \text{ мин} / 40 \text{ мин} = 1.5.$$

Интенсивность обслуживания заявок

$$\mu = 60 \text{ мин} / 30 \text{ мин} = 2.$$

Вероятность того, что заявка, пришедшая в СМО, будет обслужена

$$p_0 = \frac{\mu}{\lambda + \mu} = \frac{2}{3.5} = 0.57,$$

то есть 57 % заявок будет обслуживаться. А вероятность отказа $p_1 = 1 - p_0 = 1 - 0.57 = 0.43$, то есть 43 % заявок получат отказ.

Можно ещё определить долю времени работы СМО, называемую коэффициентом загрузки системы

$$k_{загр} = \frac{A}{\mu} = \frac{\lambda}{\mu} p_0 = \frac{1.5}{2} \cdot \frac{4}{7} = \frac{3}{7} 0.429.$$

В заключение данного параграфа проанализируем выражения (3.28). Как взаимодействует СМО с потоком заявок? Так, при $\lambda >> \mu$ $P_{отк} \to 1$, $q \to 0$. При $\lambda << \mu$ $P_{отк} \to 0$, $q \to 1$, $A \to \lambda$. Очевидно, что нельзя найти какие-либо предпочтительные соотношения между λ и μ для оптимизации показателей эффективности рассматриваемой СМО, но выбрать выгодные значения μ при каком-то λ = const можно, задаваясь какими-либо пределами увеличения A и q или уменьшения $P_{отк}$. Например, для получения $P_{отк} \le 0.1$ необходимо, чтобы $\mu \ge 9\lambda$, а для $q \ge 0.8$ – $\mu \ge 4\lambda$ и т.д.

9.2. Модель одноканальной СМО с очередью

Рассмотрим простейшую задачу: исследуемая СМО одноканальная ($n = 1$) с ограниченным числом m мест в очереди, поток заявок стационарный пуассоновский с интенсивностью λ, время (длительность) обслуживания распределено по показательному закону с параметром μ. Заявка, поступившая в момент, когда канал занят, становится в очередь, если ее длина (количество заявок в очереди) не превышает m. В противном случае заявка получает отказ и покидает систему не обслуженной, то есть рассматривается СМО вида $M \mid M \mid 1 \mid m$.

Система в произвольный момент времени может оказаться в одном из состояний: S_0 (канал свободен, очереди нет), S_1 (канал занят, очереди нет), S_2 (канал занят, в очереди одна заявка), ..., S_k (канал занят, k-1 заявка в очереди), ..., S_{m+1} (канал занят, m заявок в очереди).

Размеченный граф состояний системы представлен на рис. 3.16. В графе всего m+2 состояния. Система переходит из состояния S_i в состояние S_{i+1} под воздействием потока заявок с интенсивностью λ, и, наоборот, из состояния S_{i+1} она переходит в состояние S_i под воздействием потока обслуженных заявок (потока обслуживания) с интенсивностью μ.

Как найти предельные вероятности состояний системы, то есть вероятности состояний для установившегося (стационарного) режима функционирования СМО? В этом случае, как указано в параграфе 8.3, в уравнениях Колмогорова $\frac{dp_i}{dt} = 0$, поскольку p_i для всех состояний являются постоянными величинами.

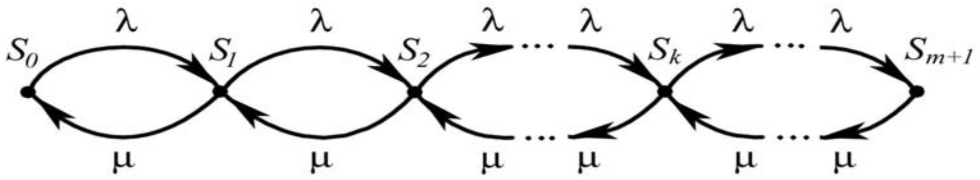


Рис. 3.16. Граф состояний одноканальной СМО с очередью

Можно напомнить правило составления этих уравнений:

– для всех потоков (для всех стрелок размеченного графа состояний) интенсивность потока умножается на вероятность того состояния, из которого выходит поток (стрелка);

– в левой части равенства суммируются произведения, соответствующие входящим потокам;

– в правой части равенства суммируются произведения выходящих потоков.

Или, пользуясь общим решением для схемы «размножения и гибели» (см. выражения (3.19) и (3.20) параграфа 8.4), можно записать выражения для предельных вероятностей состояний:

$$p_0 = \left(1 + \rho + \rho^2 + \rho^3 + \ldots + \rho^{m+1}\right)^{-1},$$

$$p_1 = \rho p_0,$$

$$p_2 = \rho^2 p_0,$$

$$\ldots \quad \ldots,$$

$$p_{m+1} = \rho^{m+1} p_0.$$

Или в более компактной записи:

$$p_0 = \left(\sum_{\alpha=0}^{m+1} \rho^{\alpha}\right)^{-1},$$
$$p_k = \rho^k p_0, \quad k = \overline{1, m+1}. \tag{3.29}$$

В выражении для p_0 формул (3.29) суммируются члены геометрической прогрессии $1, \rho, \rho^2, \ldots, \rho^{m+1}$. Используя формулу суммы членов прогрессии, можно записать:

$$p_0 = \frac{1 - \rho}{1 - \rho^{m+2}}.$$

Или, что тоже самое

$$p_0 = (1 - \rho)\left(1 - \rho^{m+2}\right)^{-1}.$$

Тогда

$$p_k = \rho^k (1-\rho)\left(1-\rho^{m+2}\right)^{-1}.$$

(3.30)

Выражения для p_0 и p_k справедливы для всех ρ, отличных от 1, если же $\rho = 1$, то для всех i :

$$p_i = \frac{1}{m+2}. \tag{3.30а}$$

Если $m = 0$, то происходит переход к одноканальной СМО с отказами.

Теперь можно последовательно определить основные характеристики эффективности СМО: $P_{отк}$, q, A, $\bar{r}$ – среднее число заявок, находящихся в очереди, $\bar{k}$ – среднее число заявок, связанных с системой (стоящих в очереди и обслуживаемых), $\bar{t}_{ож}$ – среднее время ожидания заявки в очереди и $\bar{t}_{сист}$ – среднее время пребывания заявки в системе.

Вероятность отказа в обслуживании (канал занят и все места в очереди заняты)

$$P_{отк} = p_{m+1} = \rho^{m+1} p_0 = \rho^{m+1}(1-\rho)\left(1-\rho^{m+2}\right)^{-1}. \tag{3.31}$$

так как p_{m+1} – вероятность того, что канал занят и все m мест в очереди заняты.

Относительная пропускная способность системы (вероятность того, что заявка, пришедшая в СМО, будет обслужена)

$$q = 1 - P_{отк} = 1 - \rho^{m+1} p_0. \tag{3.32}$$

Абсолютная пропускная способность (число заявок, обслуживаемых в единицу времени):

$$A = \lambda q\,. \tag{3.33}$$

Среднее число заявок, находящихся в очереди – $\bar{r}$. Для определения этой величины необходимо рассмотреть случайную величину R – число заявок в очереди, закон распределения которой следующий: с вероятностью p_2 в очереди находится одна заявка ($R =$ = 1), с вероятностью p_3 - две заявки (R = 2), с вероятностью p_{m+1} - m заявок ($R = m$). Тогда математическое ожидание этой случайной величины равно среднему количеству заявок, находящихся в очереди

$$\overline{r} = M(R) = 1 \cdot p_2 + 2 \cdot p_3 + ... + m \cdot p_{m+1} =$$

$$= \rho^2 p_0 \left(1 + 2\rho + ... + (k-1)\rho^{k-2} + ... + m\rho^{m+1}\right).$$

В полученном выражении в скобках находится сумма некоторых слагаемых. Нужно вывести формулу для этой суммы, которая представляет собой производную по ρ следующего выражения:

$$\sum_{k=1}^{m} \rho^k = \rho + \rho^2 + ... + \rho^k + ... + \rho^m,$$

а уже для полученного выражения можно использовать формулу суммы геометрической прогрессии:

$$\sum_{k=1}^{m} \rho^k = \frac{\rho - \rho^{m+1}}{1-\rho}.$$

Дифференцируя по ρ правую и левую части этого равенства получим:

$$1 + 2\rho + ... + (k-1)\rho^{k-2} + .. + m\rho^{m-1} =$$

$$= \frac{1 - \rho^m(m+1-m\rho)}{(1-\rho)^2}$$

В итоге выражение для среднего числа заявок, находящихся в очереди, записывается в следующем виде:

$$\overline{r} = \frac{\rho^2(1-\rho^m(m+1-m\rho))}{(1-\rho^{m+2})(1-\rho)} \quad . \tag{3.34}$$

Выражение (3.34) справедливо для $\rho \neq 1$, если же $\rho = 1$, то, используя (3.30а) и формулу для суммы ряда, можно получить:

$$\overline{r} = p_0(1+2+3+...+m) = \frac{1}{m+2} \cdot \frac{m(m+1)}{2} = \frac{m(m+1)}{2(m+2)}. \tag{3.34а}$$

Обозначим среднее число заявок, связанных с системой (стоящих в очереди и обслуживаемых) $\overline{k}$. Рассмотрим случайную величину $K = R + \Omega$, где $K -$ число заявок, связанных с системой, $R -$ число заявок в очереди, $\Omega -$ число заявок, находящихся на обслуживании. Учитывая, что математическое ожидание от суммы случайных величин есть сумма математических ожиданий, получим

$$\bar{k} = M[K] = M[R] + M[\Omega] = \bar{r} + \bar{\omega},$$

где $\bar{r}$ вычисляется по формуле (3.34), а выражение для $\bar{\omega}$ необходимо вывести.

В исследуемой системе используется один канал, поэтому случайная величина Ω принимает лишь два значения: 0 – с вероятностью $p_0 = (1-\rho)\left(1-\rho^{m+2}\right)^{-1}$, когда канал свободен и 1 – с вероятностью $1-p_0$, когда канал занят. Поэтому

$$\bar{\omega} = M[\Omega] = 0 \cdot p_0 + 1 \cdot (1-p_0) = (\rho - \rho^{m+2})(1-\rho^{m+2})^{-1}. \qquad (3.35)$$

Среднее время ожидания заявки в очереди $\bar{t}_{ож}$. Если учесть, что среднее время обслуживания одной заявки равно $\frac{1}{\mu}$, то, рассматривая случайную величину $T_{ож}$ – время ожидания в очереди, можно определить закон распределения этой случайной величины: с вероятностью p_1 заявка придет в систему во время обслуживания предыдущей заявки, но перед ней не будет очереди, и она будет ждать начала обслуживания в течение времени $T_{ож} = \frac{1}{\mu}$; с вероятностью p_2 в очереди перед рассматриваемой заявкой будет стоять еще одна, и время ожидания $T_{ож} = \frac{2}{\mu}$ и, наконец, с вероятностью p_k пришедшая заявка застанет в системе k заявок и будет ждать $T_{ож} = \frac{k}{\mu}$ единиц времени. Поэтому

$$\bar{t}_{ож} = M[T_{ож}] = p_1 \cdot \frac{1}{\mu} + p_2 \cdot \frac{2}{\mu} + \ldots + p_m \cdot \frac{m}{\mu}.$$

Подставляя в полученное выражение p_k $(k = \overline{1,m})$ и преобразуя его, получим:

$$\bar{t}_{ож} = M[T_{ож}] = \frac{\rho(1-\rho^m(m+1-m\rho))}{\mu(1-\rho^{m+2})(1-\rho)}. \qquad (3.36)$$

Выражение (3.36) с учетом (3.34) можно записать в упрощенном виде:

$$\overline{t}_{ож} = \frac{\overline{r}}{\rho\mu} = \frac{\overline{r}}{\lambda}, \qquad (3.37)$$

т. е. среднее время ожидания заявки начала обслуживания равно среднему числу заявок в очереди, делённому на интенсивность входного потока заявок.

Среднее время пребывания заявки в системе $\overline{t}_{сист.}$. Для определения $\overline{t}_{сист.}$ введем случайную величину $T_{сист.}$– время пребывания заявки в СМО:

$$T_{сист.} = T_{ож} + \theta,$$

где $T_{ож}$ – время ожидания заявки в очереди; θ — случайная величина времени обслуживания. Она равна времени обслуживания $T_{обсл.}$, если заявка обслуживается, и нулю, если она не обслуживается (получает отказ). Таким образом,

$$\overline{t}_{сист.} = M\left[T_{сист.}\right] = M\left[T_{ож}\right] + M\left[\theta\right] =$$

$$= \overline{t}_{ож} + q\overline{t}_{обс} = \overline{t}_{ож} + \frac{q}{\mu}$$

или иначе, с учетом выражения (3.37):

$$\overline{t}_{сист.} = \frac{\overline{r}}{\lambda} + \frac{q}{\mu}. \qquad (3.38)$$

Пример 3.9. В СМО поток заявок является простейшим с интенсивностью $\lambda = 2$ заявки в минуту. Работает один обслуживающий прибор с интенсивностью обслуживания $\mu = 2$ заявки в минуту. Определить характеристики СМО при условии, что длина очереди на входе системы ограничена и $m = 5$.

Исходные данные: Рассматривается СМО М | М | 1 | 5, $\lambda = \mu = 2$, $\rho = 1$. Граф системы содержит $m + 2 = 7$ состояний.

Решение. Согласно выражению (3.30а) для $\rho = 1$

$$p_0 = p_1 = p_2 = ... = p_6 = \frac{1}{7} \approx 0.143.$$

Вероятность отказа в обслуживании $P_{отк} = p_6 = 0.143.$

Вероятность обслуживания $q = 1 - P_{отк} = 0.857.$

Производительность системы $A = \lambda q = 2 \cdot 0.857 = 1.7$ заявок / мин.

Остальные характеристики:

$$\bar{r} = \frac{m(m+1)}{2(m+2)} = \frac{5 \cdot 6}{2 \cdot 7} = \frac{15}{7} \approx 2.1 \text{ заявок},$$

$$\bar{t}_{ож} = \frac{\bar{r}}{\lambda} = \frac{2.1}{2} = 1.05 \text{ мин},$$

$$\bar{t}_{обсл} = \frac{q}{\mu} = \frac{0.857}{2} = 0.428 \text{ мин},$$

$$\bar{t}_{сист} = 1.05 + 0.428 \approx 1.5 \text{ мин}.$$

Пример 3.10. Как и в предыдущем примере 3.9, рассматривается СМО *M* | *M* | 1 | 5, λ = 6 заявок /мин, $\bar{t}_{обс} = 20$ с. Определить характеристики СМО.

Интенсивность обслуживания $\mu = \dfrac{1}{\bar{t}_{обс}} = \dfrac{60}{20} = 3$ заявки /мин.

Приведённая интенсивность обслуживания $\rho = \dfrac{\lambda}{\mu} = \dfrac{6}{3} = 2.$

Вероятность нулевого состояния – это доля времени простоя обслуживающего канала:

$$p_0 = \frac{1-\rho}{1-\rho^{m+2}} = \frac{1-2}{1-2^7} \approx 0.008.$$

Вероятность отказа в обслуживании:

$$P_{отк} = \rho^{m+1} p_0 = 2^6 \cdot 0.008 = 0.51.$$

Вероятность обслуживания:

$$q = 1 - P_{отк} = 0.49.$$

Производительность СМО:

$$A = \lambda q = 6 \cdot 0.49 = 2.94 \text{ заявки /мин}.$$

Среднее количество заявок в очереди:

$$\bar{r} = \frac{\rho^2 (1-\rho^m (m+1-m\rho))}{(1-\rho^{m+2})(1-\rho)} = \frac{2^2 (1-2^5 (5+1-5\cdot 2))}{(1-2^7)(1-2)} = 4.03.$$

Средняя длительность ожидания заявки в очереди:

$$\bar{t}_{ож} = \frac{\bar{r}}{\lambda} = \frac{4.03}{6} = 0.67 \text{ мин.}$$

Средняя длительность обслуживания:

$$\bar{t}_{обсл} = \frac{q}{\mu} = \frac{0.49}{2} = 0.245 \text{ мин.}$$

Общее время пребывания заявки в системе:

$$\bar{t}_{сист} = 0.67 + 0.245 \approx 0.91 \text{ мин.}$$

Анализ показателей эффективности двух СМО в этом примере и предыдущем показывает, что системы отличаются только интенсивностью входного потока заявок: у второй СМО λ в три раза больше, чем у первой, а отсюда следует, что приведённая интенсивность обслуживания ρ стала во втором случае в два раза больше. Это обстоятельство во второй СМО увеличивает $P_{отк}$ приблизительно в 3.6 раза, а q увеличивает почти в два раза. Во второй СМО половина заявок уходят из системы не обслуженными ($P_{отк} = 0.51$). Эти обстоятельства требуют кардинального изменения характеристик второй СМО: увеличения количества обслуживающих каналов или увеличения интенсивности обслуживания, или изменения того и другого, или снятия ограничения на количество мест в очереди.

Подтверждением того, что необходимы некоторые структурные изменения в СМО примера 3.10, служит следующий элементарный расчёт: пусть СМО работает в сутки 12 часов, прибыль системы от обслуживания одной заявки – 100 руб., $\mu = 3$ заявки / мин, $P_{отк} = 0.51$.
Тогда за 12 часов будет потеряно

$$12 \cdot 60 \cdot \mu \cdot P_{отк} = 720 \cdot 3 \cdot 0.51 \approx 1102 \text{ заявки.}$$

Потеря выручки составит 1102 · 100 = 110 тыс. руб., что, конечно, нежелательно.

Теперь можно рассмотреть работу одноканальной СМО (n = 1) с ожиданием для случая, когда число мест в очереди не ограничено (m = = ∞). Процесс функционирования такой системы описывается путем предельного перехода от системы M | M | 1 | m при $m \to \infty$, то есть будет рассматриваться система M | M | 1 | ∞ (рис. 3.17). Этот предельный режим может существовать только при $\rho < 1$ ($\mu > \lambda$),

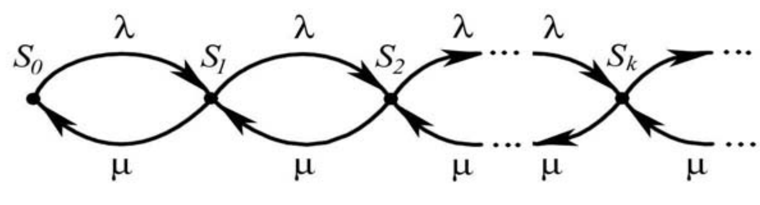


Рис. 3.17. Граф состояний одноканальной СМО с неограниченной очередью

так как в первой формуле (3.29) представлена сумма бесконечного числа членов геометрической прогрессии. При $\rho > 1$ очередь растет до бесконечности. Поэтому в дальнейшем полагают, что $\rho < 1$, тогда из формул (3.29) при $m \to \infty$ получают, что сумма прогрессии

$$1 + \rho + \rho^2 + ... + \rho^k + ... = \frac{1}{1-\rho},$$

откуда

$$p_0 = 1 - \rho, \quad p_1 = \rho(1-\rho), \quad p_2 = \rho^2(1-\rho), \quad ...,$$
$$p_k = \rho^k(1-\rho). \tag{3.39}$$

Вероятности $p_0, p_1, p_2, ..., p_k, ..$ образуют геометрическую прогрессию со знаменателем ρ. Максимальная из них p_0 – вероятность того, что канал будет вообще свободен. Это свидетельствует о том, что как бы ни была нагружена система с очередью, если она только вообще справляется с потоком заявок ($\rho < 1$), самое вероятное число заявок в системе будет равно нулю.

Основные показатели эффективности для системы M | M | 1 | ∞ определяются по следующим формулам (3.40), полученным путем предельного перехода при $m \to \infty$ из соответствующих формул для системы M | M | 1 | m. Так как при отсутствии ограничений по длине очереди каждая заявка, пришедшая в систему, будет обслужена, то:

$$P_{отк} = 0,$$

$$q = 1,$$

$$A = \lambda q = \lambda,$$

$$\bar{r} = \frac{\rho^2}{1-\rho},$$

$$\bar{k} = \bar{r} + \rho = \frac{\rho^2}{1-\rho} + \rho = \frac{\rho}{1-\rho}, \tag{3.40}$$

$$\bar{t}_{ож} = \frac{1}{\mu} \cdot \frac{\rho^2}{\lambda(1-\rho)},$$

$$\bar{t}_{сист} = \frac{1}{\mu} \cdot \frac{\rho}{1-\rho} + \frac{1}{\mu} = \frac{1}{\mu} \cdot \frac{1}{1-\rho}.$$

Пример 3.11. Рассматривается одноканальная СМО, в которую поступает простейший поток заявок с интенсивностью $\lambda = 2$ заявки / час. Обслуживание производится со средним временем $\overline{t}_{обс} = 20$ мин. В системе есть внутренний бункер с двумя местами, на которых могут ожидать обслуживания прибывающие заявки. Если оба места заняты, то заявки вынуждены ждать на внешнем накопителе. Требуется найти для стационарного режима характеристики системы; среднее число заявок, ожидающих на внешнем накопителе, вычислить среднесуточную величину штрафа за простой во внешнем накопителе, если за 1 час ожидания одной заявки система платит a ед.

Решение. По условию задачи имеем: $\mu = 3$ заявки / час, $\lambda = 2$ заявки / час, $\rho = \dfrac{2}{3}$, ρ < 1, *q* = 1.

Среднее число заявок в системе:

$$\overline{k} = \frac{\rho}{1-\rho} = 2 \text{ заявки.}$$

Средняя длительность пребывания заявки в системе:

$$\overline{t}_{сист} = \frac{1}{\mu} \cdot \frac{1}{1-\rho} = 1 \text{ час.}$$

Среднее число заявок в очереди:

$$\overline{r} = \frac{\rho^2}{1-\rho} = \frac{4/9}{1/3} = \frac{4}{3} \text{ заявки}$$

Средняя длительность пребывания заявки в очереди:

$$\overline{t}_{ож} = \frac{\rho^2}{\lambda(1-\rho)} = \frac{4/9}{2(1-2/3)} = \frac{2}{3} \text{ час.}$$

Обозначим $L_{вн.нак.}$ – длину очереди во внешнем накопителе. Тогда для вычисления средней длины очереди необходимо учитывать, что состояние S_4 – это, когда канал занят, две заявки находятся во внутреннем бункере, и пришла ещё одна заявка, размещаемая во внешнем накопителе, и так далее:

$$\overline{L}_{вн.нак.} = 1 \cdot p_4 + 2 \cdot p_5 + 3 \cdot p_6 + ... =$$
$$= \rho^4(1-\rho) + 2\rho^5(1-\rho) + 3\rho^6(1-\rho) + .. =$$
$$= \rho^4(1-\rho)(1 + 2\rho + 3\rho^2 + ...) =$$
$$= \rho^4(1-\rho)(1 + \rho + \rho^2 + \rho^3 + ...)'_\rho = \rho^4(1-\rho)\left(\frac{1}{1-\rho}\right)'_\rho =$$
$$= \rho^4(1-\rho)\frac{1}{(1-\rho)^2} = \frac{\rho^4}{1-\rho} = \frac{(2/3)^4}{1-2/3} = 3 \cdot \frac{16}{81} = \frac{16}{27}\,заявки$$

Средняя длительность пребывания заявки во внешнем накопителе:

$$\overline{T}_{вн.нак.} = \frac{\overline{L}_{вн.нак.}}{\lambda} = \frac{16/27}{2} = \frac{8}{27} \text{ час.}$$

И, наконец, сумма штрафа:

$$S_{штрафа} = \lambda_{сутки} \cdot T_{вн.очередь} \cdot a = 2 \cdot 24 \cdot \frac{8}{27} \cdot a \approx 14.2 \cdot a$$

9.3. Многоканальная СМО с очередью

Схема рассматриваемой СМО имеет вид: M | M | n | m, то есть число каналов ограничено и равно n (n >1), количество мест в очереди тоже ограничено и равно m (m >1). На вход системы поступает поток заявок с интенсивностью λ, а интенсивность обслуживания каждого канала μ.

Для правильного построения графа состояний необходимо перечислить возможные состояния СМО:

S_0 – все каналы свободны, очереди нет;

S_1 – занят один канал, очереди нет;

S_2 – заняты два канала, очереди нет;

…

S_n – заняты все n каналов, очереди нет;

S_{n+1} – заняты все каналы и одна заявка находится в очереди;

S_{n+2} – каналы заняты и две заявки в очереди;

…

S_{n+m} – заняты все n каналов и все m мест в очереди.

Размеченный граф, в котором $n + m + 1$ состояние, представлен на рис. 3.18. Как размечался этот граф?

У всех верхних стрелок поставлена интенсивность входного потока заявок λ. С такой интенсивностью входящие заявки переводят систему из любого состояния в соседнее правое. Теперь относительно нижних стрелок. Если СМО находится в состоянии S_1, то занят один канал и он обслуживает свою заявку с интенсивностью μ. Поэтому у стрелки в графе на рис. 3.18, соединяющей состояния S_1 и S_0 поставлена интенсивность обслуживания μ. Если же система находится в состоянии S_2, то заняты два канала и каждый из них обслуживает свою заявку с интенсивностью μ, поэтому суммарная интенсивность будет 2μ и у стрелки, соединяющей состояния S_2 и S_1 поставлена интенсивность обслуживания 2μ и так далее: 3μ, 4μ, ... Когда же СМО находится в состоянии S_n, то заняты все n каналов, каждый из них обслуживает свою заявку с интенсивностью μ и поэтому суммарная интенсивность максимальна и равна $n\mu$. Она поставлена у стрелки, соединяющей состояния S_n и S_{n-1}. Далее интенсивность обслуживания расти не будет, так как будут заниматься только свободные места очереди, поэтому у всех остальных нижних стрелок в графе поставлена интенсивность $n\mu$.

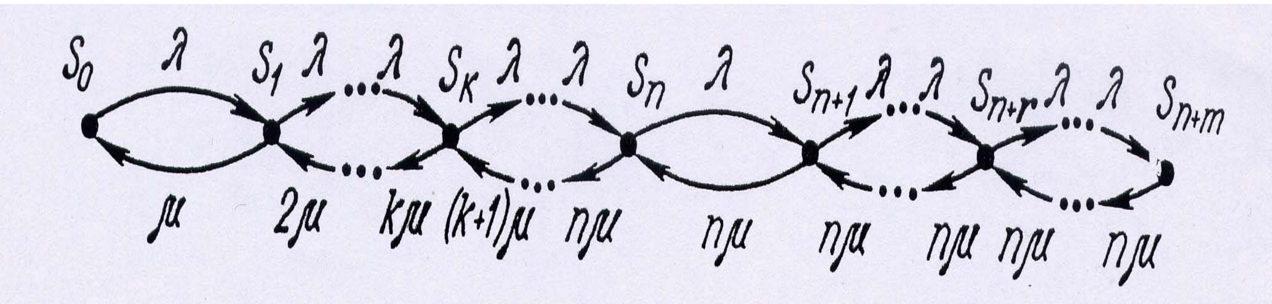


Рис. 3.18. Граф многоканальной СМО с ограниченной очередью

Пользуясь введённым обозначением $\rho = \dfrac{\lambda}{\mu}$ (это приведённая интенсивность обслуживания) и готовыми формулами процесса «размножения и гибели» (3.19) и (3.20), можно записать:

$$p_1 = \rho p_0, \quad p_2 = \frac{\rho^2}{2!} p_0, \quad \dots , \quad p_n = \frac{\rho^n}{n!} p_0,$$

$$p_{n+1} = \frac{\rho^{n+1}}{n \cdot n!} p_0, \quad p_{n+2} = \frac{\rho^{n+2}}{n^2 \cdot 2!} p_0, \quad \dots, \quad p_{n+m} = \frac{\rho^{n+m}}{n^m \cdot n!} p_0,$$

$$p_0 = \left(1 + \rho + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} + \dots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{nn!} + \frac{\rho^{n+2}}{n^2 n!} + \dots + \frac{\rho^{n+m}}{n^m n!}\right)^{-1}.$$

Приведённые выражения расчёта вероятностей состояний СМО можно записать компактно:

$$p_\alpha = \frac{\rho^k}{\alpha!} p_0, \quad \alpha = \overline{1,n},$$

$$p_{n+\beta} = \frac{\rho^{n+\beta}}{n^\beta n!} p_0, \quad \beta = \overline{1,m},$$

(3.41)

$$p_0 = \left(\sum_{\alpha=1}^{n} \frac{\rho^\alpha}{\alpha!} + \frac{\rho^n}{n!} \sum_{\beta=1}^{m} \frac{\rho^\beta}{n^\beta} \right)^{-1}.$$

Последние m слагаемых в выражении для p_0 (3.41) являются членами геометрической прогрессии со знаменателем $\frac{\rho}{n}$, поэтому

$$p_0 = \left(\sum_{\alpha=0}^{n} \frac{\rho^\alpha}{\alpha!} + \frac{\rho^n}{n!} \cdot \frac{\frac{\rho}{n} - \left(\frac{\rho}{n}\right)^{m+1}}{1 - \frac{\rho}{n}} \right)^{-1}.$$

(3.42)

Определим характеристики эффективности рассматриваемой СМО. Когда заявка, пришедшая в систему, получает отказ? Когда всё занято: все n каналов и все m мест в очереди:

$$P_{отк} = p_{n+m} = \frac{\rho^{n+m}}{n^m n!} p_0 , \tag{3.43}$$

$$q = 1 - P_{отк} = 1 - \frac{\rho^{n+m}}{n^m n!} p_0 , \tag{3.44}$$

$$A = \lambda q = \lambda \left(1 - \frac{\rho^{n+m}}{n^m n!} p_0\right). \tag{3.45}$$

Вероятность образования очереди равна сумме вероятностей того, что все каналы заняты и в системе будет находиться n, $n+1$, $n+2$, …, $n+m-1$ заявок:

$$p_{оч} = \sum_{k=n}^{n+m+1} p_k = \frac{\rho^n}{n!} \cdot \frac{1-\left(\frac{\rho}{n}\right)^m}{1-\frac{\rho}{n}} \cdot p_0.$$

Для нахождения среднего времени ожидания заявки в очереди $\overline{t}_{ож}$ нужно рассмотреть случайную величину G, которая принимает значения $\dfrac{r+1}{n\mu}$ с вероятностями p_{n+r}, для всех r, изменяющихся от 0 до $m-1$:

$$\bar{t}_{ож} = \frac{1}{n\mu} p_n + \frac{2}{n\mu} p_{n+1} + \ldots + \frac{m}{n\mu} p_{n+m-1} =$$

$$= \frac{\rho^n p_0}{nn!\mu}\left(1 + \frac{2\rho}{n} + \frac{3\rho^2}{n^2} + \ldots + \frac{m\rho^{m-1}}{n^{m-1}}\right). \tag{3.46}$$

Для нахождения среднего числа заявок, ожидающих начала обслуживания $\overline{r}$, рассматривается дискретная случайная величина R, принимающая значения r с вероятностью p_{n+r} для всех r, изменяющихся от 1 до m:

$$\overline{r} = M(R) = 1 \cdot p_{n+1} + 2 \cdot p_{n+2} + \ldots + m \cdot p_{n+m} = \sum_{r=1}^{m} r \cdot p_{n+r} =$$

$$= \frac{\rho^{n+1} p_0}{nn!}\left(1 + \frac{2\rho}{n} + \frac{3\rho^2}{n^2} + \ldots + \frac{m\rho^{m-1}}{n^{m-1}}\right). \tag{3.47}$$

Выражение (3.46) с учетом (3.47) можно переписать:

$$\bar{t}_{ож} = \frac{\overline{r}}{\lambda}. \tag{3.48}$$

В выражениях (3.46) и (3.47) в скобках находится производная по $\dfrac{\rho}{n}$ суммы членов геометрической прогрессии

$$\sum_{i=1}^{m}\left(\frac{\rho}{n}\right)^{i}=\frac{\frac{\rho}{n}-\left(\frac{\rho}{n}\right)^{m+1}}{1-\frac{\rho}{n}}.,$$

дифференцируя которую по $\frac{\rho}{n}$ и делая некоторые преобразования, получают:

$$\bar{r}=\frac{\rho p_0}{n\cdot n!}\cdot\frac{1-(m+1)\left(\frac{\rho}{n}\right)^{m}+m\left(\frac{\rho}{n}\right)^{m+1}}{\left(1-\frac{\rho}{n}\right)^{2}}. \qquad (3.49)$$

Аналогичным образом можно получить выражение для $\bar{t}_{ож}$.

Среднее число занятых каналов $\bar{z}$ определяется с учетом того, что каждый занятый канал обслуживает в среднем μ заявок в единицу времени, а вся СМО обслуживает *A* заявок, тогда

$$\bar{z}=\frac{A}{\mu}=\rho\left(1-\frac{\rho^{n+m}}{n^{m}\cdot n!}p_0\right). \qquad (3.50)$$

Среднее число заявок $\bar{k}$, связанных с системой, определяется с учетом выражений (3.49) и (3.50):

$$\bar{k}=\bar{r}+\bar{z}. \qquad (3.51)$$

Для нахождения среднего времени пребывания заявки в системе можно, используя выражение (3.48), записать:

$$\bar{t}_{сист}=\bar{t}_{ож}+\bar{t}_{обс}=\frac{\bar{r}}{\lambda}+\frac{q}{\lambda}. \qquad (3.52)$$

Таким образом, рассмотрены основные показатели эффективности *n*-канальной СМО с ожиданием, причем длина очереди ограничена и равна *т*.

Пример 3.12. В СМО в среднем через каждые 30 мин прибывают заявки на обслуживание. Средняя длительность обслуживания одной заявки составляет 1.5 ч. Обработку заявок производят два канала. В системе есть накопительный бункер, в котором могут находиться в очереди в ожидании обслуживания не более 4 заявок. Определить основные характеристики СМО.

Решение. Рассматривается СМО вида M | M | 2 | 4, то есть *n* = 2, *m* = 4,

$\lambda = 2$ заявки/час, $\mu = \dfrac{2}{3}$ заявки/час, $\rho = \dfrac{\lambda}{\mu} = 3$, $\dfrac{\rho}{n} = \dfrac{3}{2}.$

В соответствии с выражением (3.42) вероятность нулевого сос-тояния вычисляется так: $p_0 = \left(1 + 3 + \dfrac{3^2}{2!} + \dfrac{3}{2!} \cdot \dfrac{\frac{3}{2} - \left(\frac{3}{2}\right)^5}{1 - \frac{3}{2}} \right)^{-1} = 0.158.$

Вероятности остальных состояний в соответствии с первыми двумя формулами выражения (3.41):

$$p_1 = 3 \cdot 0.0158 = 0.0474, \quad p_2 = 3^2\!\big/_2 \cdot 0.0158 = 0.0711,$$

$$p_3 = \frac{3^3}{2 \cdot 2} \cdot 0.0158 = 0.107, \quad p_4 = \frac{3^4}{2^2 \cdot 2!} \cdot 0.0158 = 0.160,$$

$$p_5 = \frac{3^5}{2^3 \cdot 2!} \cdot 0.0158 = 0.240, \quad p_6 = \frac{3^6}{2^4 \cdot 2!} \cdot 0.0158 = 0.360.$$

Можно проверить правильность проведённых вычислений, так как сумма p_i должна равняться единице или с учётом округлений – приблизительно единице:

0.0158 + 0.0474 + 0.0711 + 0.107 + 0.160 + 0.240 + 0.360 = 1.001 ≈ 1.

Вероятность того, что заявка, пришедшая в СМО, получит отказ (все каналы заняты и все места в очереди тоже заняты):

$$P_{отк} = p_6 = 0.36.$$

Вероятность образования очереди:

$$p_{очер} = \sum_{k=n}^{n+m+1} p_k = \frac{\rho^n}{n} \cdot \frac{1 - \left(\frac{\rho}{n}\right)^m}{1 - \frac{\rho}{n}} \cdot p_0 = \frac{3^2}{2} \cdot \frac{1 - \left(\frac{3}{2}\right)^4}{1 - \frac{3}{2}} \cdot 0.0158 \approx 0.58,$$

но эту величину (вероятность образования очереди) можно вычислить, просто просуммировав уже полученные значения вероятностей p_2, p_3, p_4 и p_5.

Вероятность того, что заявка, пришедшая в СМО, будет обслужена вычисляется по формуле (3.44):

$$q = 1 - P_{отк} = 0.64.$$

Производительность СМО (3.45): $A = 2 \cdot 0.64 = 1.28$ заявок / ч.

Среднее количество заявок, находящихся в очереди, (3.49):

$$\bar{r} = \frac{3^3}{2 \cdot 2} 0.0158 \cdot \frac{1 - \left(\frac{3}{2}\right)^4 \cdot \left(5 - 4 \cdot \left(\frac{3}{2}\right)\right)}{\left(1 - \left(\frac{3}{2}\right)\right)^2} \approx 2.6 \text{ заявки.}$$

Средняя длительность ожидания заявки в очереди (3.48):

$$\bar{t}_{ож} = \frac{2.6}{2} = 1.3 \text{ час.}$$

Средняя длительность обслуживания заявки:

$$\bar{t}_{обсл} = \frac{q}{\lambda} = \frac{0.64}{2/3} = 0.96 \text{ час.} \approx 58 \text{ мин.}$$

Средняя длительность пребывания заявки в системе (3.52):

$$\bar{t}_{сист} = 1.3 + 0.96 = 2.26 \text{ час.}$$

Среднее число занятых каналов (3.50):

$$\bar{z} = \frac{1.28}{2/3} = 1.92 \text{ каналов.}$$

Коэффициент занятости каналов:

$$z_{зан} = \frac{\bar{z}}{n} = 1.92/2 = 0.96.$$

Полученные характеристики СМО M | M | 2 | 4 свидетельствуют о том, что доля простоя каналов очень мала и составляет всего 1.58 % рабочего времени, а вероятность отказа в обслуживании очень большая: 36 % заявок получат отказ. Коэффициент загрузки каналов – 0.96, а относительная пропускная способность маленькая (только 64 % заявок будет обслужено). Из всего этого можно сделать вывод: СМО не справляется с потоком заявок, и необходимо увеличить число каналов или интенсивность обслуживания и, возможно, увеличить число мест в очереди.

По аналогии с подходом, используемым в § 9.2, можно перейти к системе с неограниченной очередью. Пусть в СМО, состоящую из n каналов, поступает простейший поток заявок с интенсивностью λ, время обслуживания одной заявки подчинено показательному закону распределения с параметром μ. Размеченный граф состояний таких систем M | M | n | ∞ имеет бесконечное число состояний (рис. 3.19).

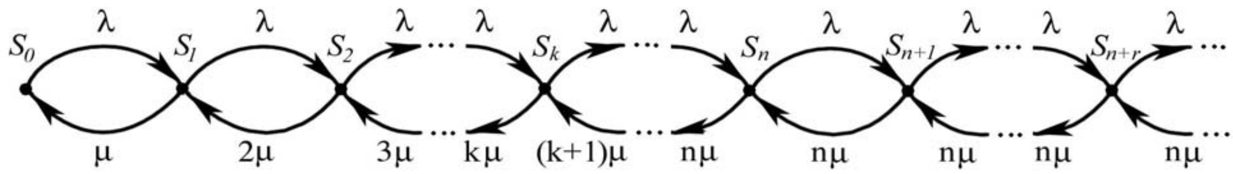


Рис. 3.19. Граф многоканальной СМО с неограниченной очередью

Можно перечислить возможные состояния в этом графе:

S_0 – все каналы свободны, очереди нет;

S_1 – занят один канал, очереди нет;

S_2 – заняты два канала, очереди нет;

...

S_n – заняты все n каналов, очереди нет;

S_{n+1} – заняты все каналы, в очереди одна заявка;

S_{n+2} – заняты все каналы, в очереди две заявки;

…

S_{n+i} – заняты все каналы, в очереди i заявок;

…

Используя формулы (3.41) и (3.42), получают вероятности состояний для рассматриваемой системы M | M | n | ∞, осуществляя предельный переход ($m \to \infty$). Другим подходом может являться использование формул (3.19) и (3.20) процесса «размножения и гибели» (см. параграф 8.4). В любом случае выражения для вероятностей состояний системы M | M | n | ∞ записаны ниже.

$$p_1 = \rho \cdot p_0, \quad p_2 = \frac{\rho^2}{2!} \cdot p_0, \; ..., \; p_n = \frac{\rho^n}{n!} \cdot p_0,$$

$$p_{n+1} = \frac{\rho^{n+1}}{n \cdot n!} \cdot p_0, \quad p_{n+2} = \frac{\rho^{n+2}}{n^2 \cdot n!} \cdot p_0, \quad ..., \; p_{n+i} = \frac{\rho^{n+i}}{n^i \cdot n!}, \quad ...,$$

$$p_0 = \left(1 + \rho + \frac{\rho^2}{2!} + ... + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n! \cdot (n - \rho)} \right)^{-1}.$$

Полученные выражения можно записать более компактно:

$$p_\alpha = \frac{\rho^\alpha}{\alpha!} p_0, \quad \alpha = \overline{1,n}$$

$$p_{n+1} = \frac{\rho^{n+1}}{nn!} p_0, \quad \ldots, \quad p_{n+r} = \frac{\rho^{n+r}}{n^r n!} p_0, \quad \ldots \qquad (3.53)$$

$$p_0 = \left(\sum_{\alpha=0}^{n} \frac{\rho^\alpha}{\alpha!} + \frac{\rho^{n+1}}{n(n-\rho)} \right)^{-1}.$$

Выражение для p_0 в формулах (3.53) получено в предположении, что $\frac{\rho}{n} < 1$, так как только при выполнении этого условия существует установившийся режим, а при $\frac{\rho}{n} \geq 1$ очередь бесконечно возрастает.

Определим основные зависимости для вычисления параметров эффективности рассматриваемой СМО.

Вероятность отказа в обслуживании $P_{отк} = 0$, так как количество мест в очереди не ограничено и при занятых каналах пришедшая

заявка всегда будет поставлена в очередь и когда-нибудь будет обслужена.

Вероятность образования очереди вычисляется как вероятность того, что все каналы заняты:

$$P_{очер} = \frac{\rho^{n+1}}{n!\,(n-\rho)} \cdot p_0.$$

Вероятность обслуживания $q = 1$, а абсолютная пропускная способность соответственно $A = \lambda$.

Средняя длина очереди $\bar{r}$ находится из (3.49) при $T \to \infty$:

$$\bar{r} = \frac{\rho^{n+1}}{n \cdot n! \left(1 - \frac{\rho}{n}\right)^2} p_0 = \frac{n}{n-\rho} P_{очер} \quad . \qquad (3.54)$$

Из (3.46) и (3.48) или из (3.54) средняя длительность ожидания заявки начала обслуживания

$$\bar{t}_{ож} = \frac{\bar{r}}{\lambda} = \frac{\rho^n}{n \cdot \mu \cdot n! \cdot \left(1 - \frac{\rho}{n}\right)^2} p_0. \tag{3.55}$$

Среднее число занятых каналов $\bar{z} = \frac{A}{\mu} = \rho.$ Как и прежде среднее число заявок, связанных с системой, $\bar{k} = \bar{r} + \bar{z}$, а среднее время пребывания заявок в СМО $\bar{t}_{сист} = \bar{t}_{ож} + \frac{1}{\mu}$.

Пример 3.13. Рассматривается СМО типа M | M | 3 | ∞, то есть это трёхканальная система с неограниченным числом мест в очереди. Интенсивность входного потока заявок составляет 5 заявок в единицу времени. Интенсивность обслуживания заявок каждым каналом составляет 2 заявки в единицу времени. Определить характеристики СМО.

Таким образом, $n = 3, \quad \lambda = 5, \quad \mu = 2, \quad \rho = \frac{\lambda}{\mu} = 2.5, \quad \frac{\rho}{n} = \frac{2.5}{3} < 1.$

Согласно формулам (3.53):

$$p_0 = \left(1 + 2.5 + \frac{2.5^2}{2!} + \frac{2.5^3}{3!} + \frac{2.5^4}{3! \cdot (3 - 2.5)}\right)^{-1} \approx 0.045,$$

$$p_1 = 2.5 \cdot 0.045 \approx 0.112, \quad p_2 = \frac{2.5^2}{2} \cdot 0.045 \approx 0.141, \quad p_3 = \frac{2.5^3}{6} \cdot 0.045 \approx 0.117,$$

$$p_4 = \frac{2.5^4}{3 \cdot 3!} \cdot 0.045 \approx 0.098, \quad p_5 = \frac{2.5^5}{3^2 \cdot 3!} \cdot 0.045 \approx 0.081, \quad p_6 = \frac{2.5^6}{3^3 \cdot 3!} \cdot 0.045 \approx 0.068,$$

$$p_7 = \frac{2.5^7}{3^4 \cdot 3!} \cdot 0.045 \approx 0.057, \quad p_8 = \frac{2.5^8}{3^5 \cdot 3!} \cdot 0.045 \approx 0.047, \quad p_9 = \frac{2.5^9}{3^6 \cdot 3!} \approx 0.039, \quad \ldots$$

Анализ результатов расчёта вероятностей состояний показывает, что вероятность того, что обслуживающие каналы свободны (p_0), значительно меньше величин p_1, p_2 или p_3 (то есть вероятностей занятости одного, двух или трёх каналов). Затем, с ростом очереди вероятности состояний уменьшаются и при пяти занятых местах в очереди p_0 и p_8 почти равны. Это говорит о том, что в данной СМО маловероятны как простаивание всех обслуживающих каналов, так и очередь из пяти и более заявок.

Вероятность возникновения очереди

$$P_{очер} = \frac{\rho^{n+1}}{n!(n - \rho)} \cdot p_0 = \frac{2.5^4}{3!(3 - 2.5)} \cdot 0.045 \approx 0.586.$$

Эту величину можно вычислить ещё и путём вычитания из единицы суммы вероятностей тех состояний, в которых очередь отсутствует:

$$P_{очер} = 1 - (p_0 + p_1 + p_2 + p_3) =$$
$$= 1 - (0.045 + 0.112 + 0.141 + 0.117) \approx 0.586.$$

Вероятность обслуживания, как уже говорилось, всегда равна единице (q = 1), а производительность данной СМО $A = \lambda q = 5$ (заявок в единицу времени).

Прочие характеристики:

$$\bar{r} = \frac{n}{n-\rho} \cdot P_{очер} = \frac{3}{3-2.5} \cdot 0.045 \approx 3.5,$$

$$\bar{t}_{ож} \frac{\bar{r}}{\lambda} = 3.5/5 = 0.7,$$

$$\bar{t}_{обсл} = \frac{1}{\mu} = 0.5,$$

$$\bar{t}_{сист} = 0.7 + 0.5 = 1.2,$$
$$\bar{z} = \rho = 2.5,$$

$$\bar{k} = \bar{r} + \bar{z} = 3.5 + 2.5 = 6.$$

Таким образом, доля простоя канала равна 4.5 % от длительности интервала функционирования СМО, вероятность возникновения очереди довольно большая – 0.586, но длина очереди небольшая (3.5 заявки) и время ожидания обслуживания небольшое (0.7 единиц времени). Работу данной системы следует признать удовлетворительной.

9.4. СМО с ограниченным временем ожидания

Рассматривается n-канальная СМО с ожиданием и неограни-ченным числом мест в очереди. Однако время ожидания заявки начала обслуживания в этой системе ограничено некоторым случайным сроком со средним значением $\bar{t}_{очер}$ (то есть заявка считается нетерпеливой). Если до истечения этого времени заявка не будет принята к обслуживанию, то она покидает очередь и уходит из системы не обслуженной. В этом случае можно сказать, что на каждую заявку, стоящую в очереди, действует «поток уходов» интенсивности $\nu = \frac{1}{\bar{t}_{очер}}$.

Если этот поток пуассоновский, то процесс, протекающий в СМО, является марковским.

Размеченный граф состояний такой системы представлен на рис. 3.20. При этом у стрелок, направленных справа налево от S_n до S_0, может быть проставлена суммарная интенсивность потока обслуживания занятых каналов.

У остальных стрелок справа налево проставлены суммарные интенсивности потока обслуживаний всех n каналов $n\mu$ плюс соответствующая интенсивность потока уходов из очереди. Это опять схема «размножения и гибели». Поэтому, пользуясь формулами из параграфа 8.4 и обозначая дополнительно $\beta = \frac{\nu}{\mu}$, можно записать:

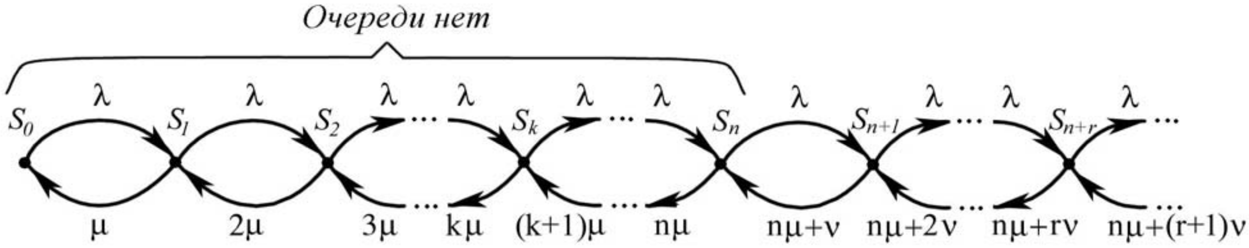


Рис. 3.20. Граф состояний СМО с ограниченным временем ожидания

$$p_0 = \left\{1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \ldots + \frac{\rho^n}{n!} + \right.$$

$$\left. + \frac{\rho^n}{n!}\left[\frac{\rho}{n+\beta} + \frac{\rho^2}{(n+\beta)(n+2\beta)} + \ldots + \frac{\rho^r}{(n+\beta)(n+2\beta)\ldots(n+r\beta)} + \ldots\right]\right\}^{-1} =$$

$$= \left(\sum_{\alpha=0}^{n} \frac{\rho^\alpha}{\alpha!} + \frac{\rho^n}{n!}\sum_{r=1}^{\infty} \frac{\rho^r}{\prod_{s=1}^{r}(n+s\beta)}\right)^{-1},$$

(3.56)

$$p_\alpha = \frac{\rho^\alpha}{\alpha!} p_0, \ \ \alpha = \overline{1,n},$$

$$p_{n+1}=\frac{\rho^n}{n!}\frac{\rho}{n+\beta}p_0,\;\; p_{n+2}=\frac{\rho^n}{n!}\frac{\rho^2}{(n+\beta)(n+2\beta)}p_0,\;\ldots$$

$$\ldots,\;\; p_{n+r}=\frac{\rho^n}{n!}\frac{\rho^r}{\prod\limits_{S=1}^{r}(n+s\beta)}p_0.$$

В формулах (3.56) в выражении для p_0 записана сумма бесконечного ряда, не являющегося прогрессией. При необходимости она может быть вычислена приближенно.

Если в формулах (3.56) перейти к пределу при $\gamma \to 0$ (или $\beta \to 0$), то при $\rho < n$ получаются выражения (3.41) и (3.42), то есть «нетерпеливые» заявки станут «терпеливыми».

Теперь можно определить основные характеристики рассмат-риваемой системы. Понятие вероятность отказа, по-видимому, не имеет смысла, так как каждая заявка, поступившая в систему, ставится в очередь. Но заявка может и не дождаться обслуживания, уйдя из очереди по истечении времени ожидания.

Среднее число заявок в очереди

$$\bar{r}=M[R]=\sum_{r=}^{\infty} r p_{n+r}=1\cdot p_{n+1}+2\cdot p_{n+2}+\ldots$$

Абсолютная пропускная способность СМО *A* (среднее число обслуженных заявок в единицу времени) легко определяется, если учесть, что из среднего числа заявок в очереди $\bar{r}$ под воздействием «потока ухода» будет уходить, не дождавшись обслуживания, $\gamma\bar{r}$ заявок в единицу времени. Поэтому

$$A=\lambda-\gamma\,\bar{r},$$

$$q=\frac{A}{\lambda}=1-\frac{\gamma}{\lambda}\bar{r}.$$

Можно найти среднее число занятых каналов:

$$\bar{z}=\frac{A}{\mu}=\rho-\beta\bar{r},$$

а отсюда

$$\bar{r}=\frac{\rho-\bar{z}}{\beta}.$$

Что касается среднего числа занятых каналов, то его можно найти как математическое ожидание величины Z, принимающей значения 0, 1, 2, ..., n с вероятностями $p_0,\ p_1, p_2,\ ..., p_{n-1},\ (1-(p_0+p_1+p_2+...+p_{n-1}))$

$$\bar{z} = 1p + 2p + ... + n\left(1-\sum_{k=0}^{n-1} p_k\right).$$

Формулы для среднего времени ожидания в очереди и среднего времени пребывания заявки в системе выводятся достаточно сложно и в пособии не приводятся [8, 9].

В заключение данной главы в табл. 3.2 приведены формулы для вычисления основных характеристик СМО, которые подробно описаны в параграфах 9.1 – 9.3 и которые наиболее широко распространены в повседневной практике.

Таблица 3.2

Вид СМО	Величины p_0, p_k	Величины $P_{отк}, q, A$
M|M|n|0	$p_0 = \left(\sum_{\alpha=0}^{n} \frac{\rho^\alpha}{\alpha!}\right)^{-1},$ $p_k = \frac{\rho^k}{k!} p_0,\ k=\overline{1,n}$	$P_{отк} = p_n = \frac{\rho^n}{n!} p_0,$ $q = 1 - P_{отк},$ $A = \lambda q = \lambda(1-P_{отк})$
M|M|1|0	$p = \frac{\mu}{\lambda+\mu},$ $p_1 = 1 - p_0 = \frac{\lambda}{\lambda+\mu}$	$P_{отк} = p_1 = \frac{\lambda}{\lambda+\mu},$ $q = p_0 = \frac{\mu}{\lambda+\mu},$ $A = \lambda q = \frac{\lambda\mu}{\lambda+\mu}$
M|M|1|m	$p_0 = \left(\sum_{\alpha=0}^{m+1} \rho^\alpha\right)^{-1},$ $p_k = \rho^k p_0,\ \ k=\overline{1,m+1}$	$P_{отк} = p_{m+1} = \rho^{m+1} p_0,$ $q = 1-\rho^{m+1} p_0,$ $A = \lambda q$

M|M|1|∞	$p_0 = 1 - \rho$ $p_k = \rho^k\,(1-\rho), \quad k = 1, 2, 3, \ldots$	$P_{отк} = 0,$ $q = 1,$ $A = \lambda$

Окончание таблицы 3.2

Вид СМО	Величины $p_0,\ p_k$	Величины $P_{отк}$, q, A
M|M|n|m	$p_0 = \left(\sum\limits_{\alpha=0}^{n} \dfrac{\rho^\alpha}{\alpha!} + \dfrac{\rho^n}{n!} \sum\limits_{\beta=1}^{m} \dfrac{\rho^\beta}{n^\beta} \right)^{-1},$ $p_\alpha = \dfrac{\rho^\alpha}{\alpha!} p_0, \quad \alpha = \overline{1,n},$ $p_{n+\beta} = \dfrac{\rho^{n+\beta}}{n^\beta n!} p_0, \quad \beta = \overline{1,m}$	$P_{отк} = p_{n+m} = \dfrac{\rho^{n+m}}{n^m n!} p_0,$ $q = 1 - \dfrac{\rho^{n+m}}{n^m m!} p,$ $A = \lambda q$
M|M|n|∞	$p_0 = \left(\sum\limits_{\alpha=0}^{n} \dfrac{\rho^\alpha}{\alpha!} + \dfrac{\rho^{n+1}}{n!(n-\rho)} \right)^{-1},$ $p_\alpha = \dfrac{\rho^\alpha}{\alpha!} p_0, \quad \alpha = \overline{1,n},$ $p_{n+\beta} = \dfrac{\rho^{n+\beta}}{n^\beta n!} p_0, \quad \beta = 1, 2, 3, \ldots$	$P_{отк} = 0,$ $q = 1,$ $A = \lambda$

Г л а в а 10
МЕТОДИКА ВЫПОЛНЕНИЯ ИНЖЕНЕРНЫХ РАСЧЁТОВ В СРЕДЕ СИСТЕМЫ *MAPLE*

Программа *Maple* 9 (*http: maplesoft.com*), называемая в дальнейшем просто *Maple*, выпускается корпорацией *Waterloo Maple Inc.* (Канада) и является одной из последних версий этой системы. Она является чрезвычайно удобным средством реализации аналитических вычислений. При этом необходимо учитывать, что можно пользоваться встроенными функциями, которых более 3000.

Пользовательский интерфейс *Maple* позволяет создавать доку-менты, состоящие из команд входного языка, результатов вычислений, графических данных и комментариев. Результаты вычислений могут быть представлены в числовом виде и в виде обычных математических формул. Управление системой возможно с помощью главного системного меню, панелей инструментов и палитр, а также горячих клавиш. Поддерживаются многие возможности мыши, характерные для *Windows*-приложений [15, 38].

10.1. Основы программирования в среде *Maple*

Запустить *Maple* можно одним из способов запуска приложений *Windows*, например, выполнив команду

Пуск – Программы – *Maple* 9.

Появится *Windows*–образное окно *Maple*. В нем верхняя строка **– Строка заголовка.** Ниже – **Панель главного меню**. Третья строка – **Панель инструментов**. Её вид зависит от исполняемой задачи. Четвертая строка – **Панель форматирования.** Под ней располагается **Контекстное меню.** И нижняя строка экрана – **Строка состояния.** Центральная часть экрана между панелью форматирования и строкой состояния представляет собой рабочее поле *Maple.* На рабочем поле расположено окно файла ***Introduction to Maple*** **9** (рис. 3.21). Если не планируется ознакомление с содержанием этого файла, то можно его закрыть, нажав кнопку **Закрыть.** Останется пустое окно *Maple.* Если щелкнуть по кнопке **Create a new worksheet** или выполнить команду **Файл – Открыть файл,** то появится окно файла *Maple,* которому по умолчанию присваивается имя **Untitled (1) - [Server1]** (рис. 3.22)**.**

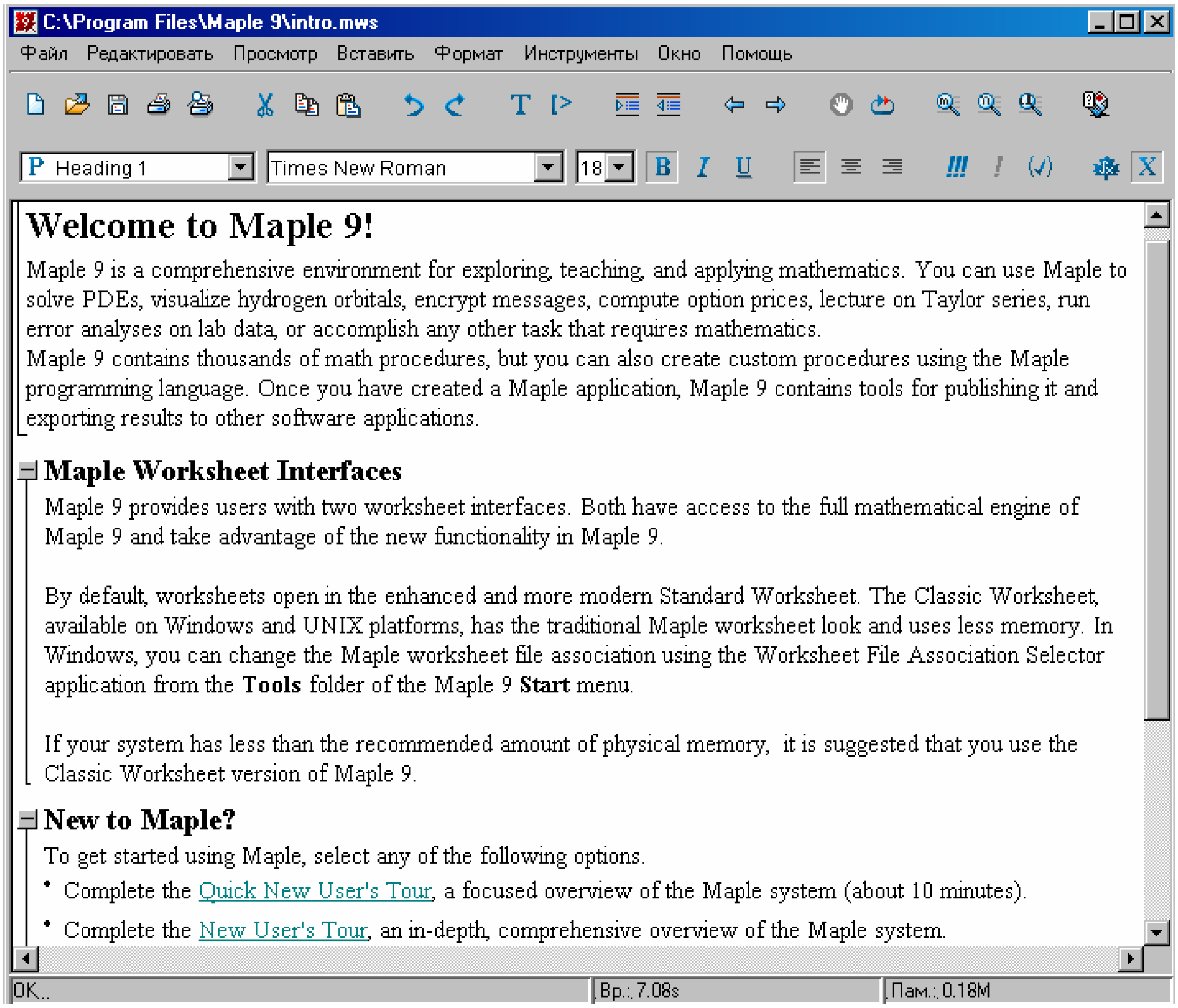


Рис. 3.21. Окно файла ***Introduction to Maple*** **9**

В первой строке слева находятся символы [(квадратная скобка), > (знак больше) и | (мерцающий курсор). Теперь можно начинать диалог с системой, предварительно убедившись, что клавиатура находится в режиме *Enter*. Левая квадратная скобка [обозначает вычислительную секцию, а знак > – приглашение к вводу. Мигающая вертикальная черточка – курсор ввода. Каждый вопрос (ввод) и получаемый ответ (вывод) занимает отдельную вычислительную секцию. Ввод осуществляется в командную строку, начинающуюся со знака >, и все вводимые символы имеют на экране красный цвет. Вывод (ответ) осуществляется в следующую за командной

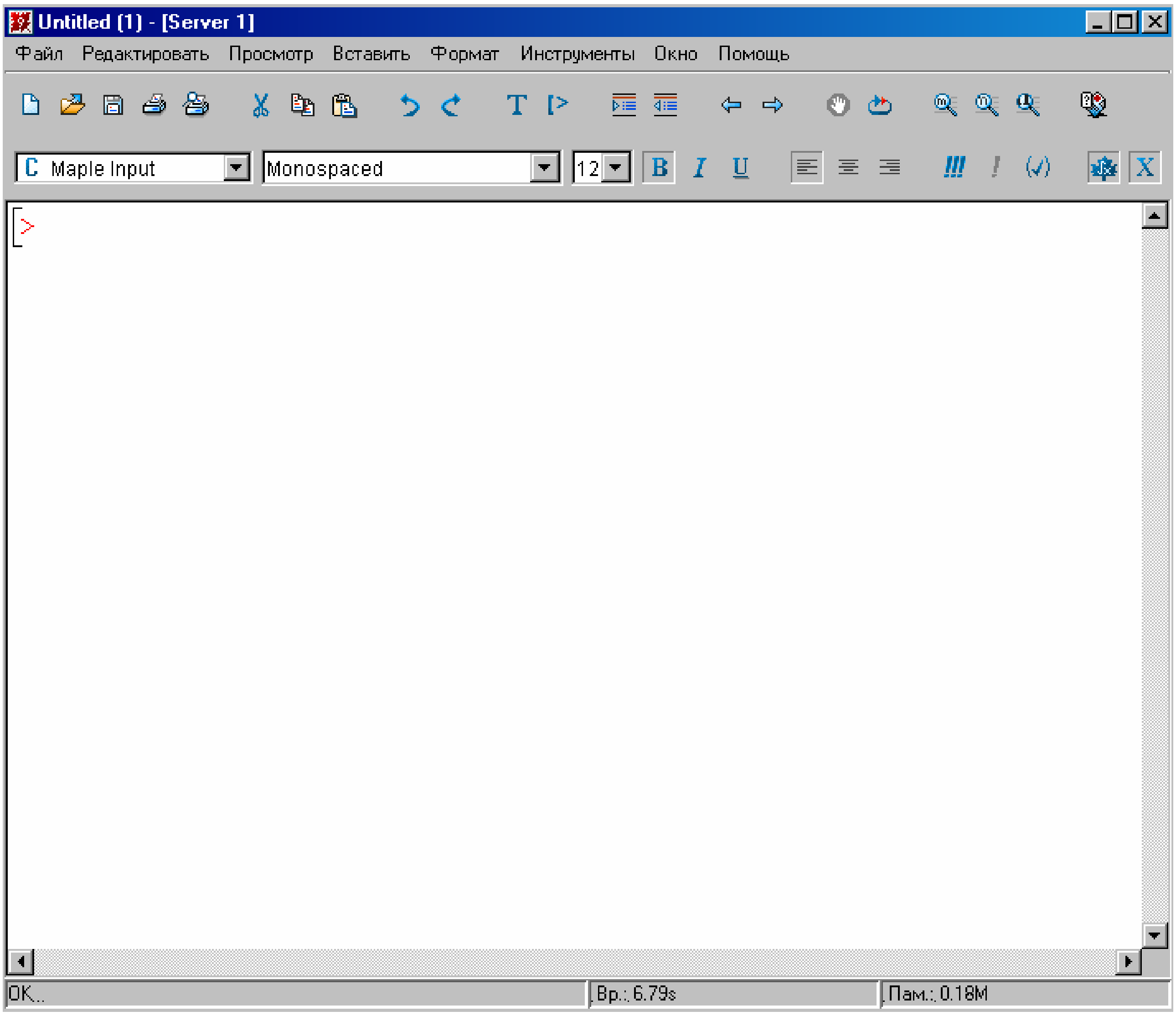


Рис. 3.22. Рабочее окно программы *Maple*

строку и автоматически выравнивается по центру. Все выводимые символы окрашены в синий цвет. Длина квадратной скобки [изменяется в зависимости от размера вводимого выражения и результата ответа. Выражение в *Maple* – это объект, соответствующий обычному математическому выражению. Чтобы правильно ввести выражение, необходимо ознакомиться с правилами его построения.

Элементы входного языка. Алфавит языка содержит 26 малых и 26 больших букв латинского алфавита (от *a* до *z* и от *A* до *Z*), 10 арабских цифр (от 0 до 9) и 32 специальных символа. Совокупность правил, по которым записываются определения всех объектов *Maple*-языка, называется синтаксисом. *Maple* – это система для манипулирования математическими выражениями. Выражения строятся из числовых констант, переменных, знаков операторов и круглых скобок. Выражения записываются в одну строку и вычисляются слева направо в порядке старшинства операций.

Допускается использование вещественных, целых и рациональных чисел. Требуется выполнение следующих правил: целая часть числа от дробной отделяется десятичной точкой; нулевая мантисса не отображается (число начинается с разделительной точки); мантисса отделяется от порядка пробелом; указание десятичной точки в любом месте числа позволяет считать число вещественным и переводит вычисление в режим работы с вещественными числами.

Ввод выражений производится в обычной для языков программирования строчной форме. Вывод результатов вычислений по умолчанию осуществляется в привычной математической записи. Для улучшения наглядности вводимых формул можно использовать текстовые комментарии. Для ввода выражений и комментариев используют командные строки различных типов. Как уже говорилось, строка ввода математических выражений имеет отличительный символ >, а строка ввода комментария такого признака не имеет. Переключение типа текущей строки осуществляется нажатием клавиши *F*5.

Переменные задаются своими идентификаторами (именами). Идентификаторы представляют собой последовательность букв латинского алфавита, цифр и некоторых знаков, начинающуюся с буквы. В идентификаторах различаются строчные и прописные буквы, так что $X_{\max}$ и $x_{\max}$ являются именами различных переменных. Ограничений на длину идентификатора практически нет (длина не должна превышать 524 275 символов). Нельзя в идентификаторы вставлять знаки арифметических операторов и в качестве идентификаторов использовать ключевые слова языка.

Ключевые слова используются для конструирования операторов, команд и процедур.

Имена переменных должны быть уникальными. Примеры имен переменных: X, Y, x, y, $pr1$, $pr2$, $X_{\max}$, x_{max}, sm_7.

Константы – это простейшие именованные объекты, несущие заранее предопределенные значения. Их имена заранее определены и не могут меняться.

Строка – это цепочка символов. Последовательность символов рассматривается как строка, если она заключается в обратные апострофы (` ). Например: \`строка\`, \`столбец\`, \`*X*\`, \`*Y*\`.

В строку может входить любое математическое выражение, но при этом оно не выполняется. Строки используются для создания комментариев. Для этого надо перейти в другой режим ввода, убрав из командной строки знак >. Как уже говорилось, это делается путем нажатия клавиши *F*5.

При необходимости ввода комментария в *Maple*-программу следует перед вводом комментария поставить знак #. Любой текст, даже если это математическое выражение, рассматривается как неисполняемый.

Выражения могут включаться в наборы. Наборы представляют собой совокупность выражений, отделенных друг

от друга запятой и заключенных в фигурные скобки {}. Примеры выражений приведены на рис. 3.23. *Maple* устраняет из множеств повторяющиеся значения и расставляет их в определенном порядке (числа – в порядке возрастания).

```
> {4,2,3,2,4,2,1};
                                   {1, 2, 3, 4}
> {3+2,4+5,6+1};
                                    {5, 7, 9}
> {b,a,d};
                                    {b, d, a}
> {'Peter','Ann','Kate'};
                               {Ann, Peter, Kate}
>
```

Рис. 3.23. Примеры выражений

Список – упорядоченный набор выражений, заключенных в квадратные скобки [ ] и отделенных друг от друга запятой. Примеры списков приведены на рис. 3.24.

```
> [4,2,3,2,4,2,1];
                               [4, 2, 3, 2, 4, 2, 1]
> [3+2,4+5,6+1];
                                    [5, 9, 7]
> [b,a,d];
                                    [b, a, d]
> ['Peter','Ann','Kate'];
                               [Peter, Ann, Kate]
>
```

Рис. 3.24. Примеры списков

Элементы списка преобразуются и выводятся в том порядке, в каком они были заданы. Список можно обозначить и при необходимости использовать его элемент, указав в квадратных скобках номер этого элемента (рис. 3.25).

```
> S:=[`Pete`, `Ann`, `Kate`];
```

$$S := [Pete, Ann, Kate]$$

```
> S[2];
```

$$Ann$$

```
>
```

Рис. 3.25. Пример элемента списка

Векторы и матрицы создаются с помощью функции *array*:

***array*(*a*...*b*, *s*1) – создает вектор с индексами от *a* до *b* и значениями в одномерном списке *s*1;**

***array*(*a*...*b*, *c*...*d*, *s*2) – создает матрицу, номера строк которой изменяются от *a* до *b*, номера столбцов – от *c* до *d*, со значениями в двумерном массиве списка *s*2.**

Например, функция *array*(1…3, [*x*, *y*, *x*+*y*]) создает вектор с элементами *x*, *y*, *x*+*y*; функция *array*(1…2, 1…2, [[*a*,*b*], [*c*,*d*]]) создает квадратную матрицу $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ (рис. 3.26).

```
> array(1..3, [x, y, x+y]);
```

$$[x, y, x + y]$$

```
> array(1..2, 1..2, [[a, b], [c, d]]);
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

```
>
```

Рис. 3.26. Пример создания векторов и массивов

В *Maple* имеется большое количество операторов, которые могут быть поделены на пять основных типов: бинарные (*binary*) (с двумя операндами), унарные (*unary*), нульарные (*nullary* – операторы без операндов – одна, две или три пары кавычек), операторы старшинства (*precedence*), функциональные операторы (*functional*).

Включено более тридцати бинарных операторов: сложения, вычитания, умножения, деления, возведения в степень, определения остатка числа, присваивания, логические операторы. Операторы используются для конструирования выражений. Они обеспечивают выполнение определенных операций над данными, представленными операндами.

Для записи арифметических выражений используются следующие операторы: + – сложения, - – вычитания, * – умножения, / – деления, ^ – возведения в степень

(альтернативный оператор **). Ввод выражения должен осуществляться в одну строку и заканчиваться точкой с запятой (;) или двоеточием (:).

Точка с запятой (;) – оператор вывода результатов. Если выражение заканчивается точкой с запятой, то после нажатия кнопки *Enter* будут произведены вычисления и выведен результат в центре второй строки данной вычислительной секции, в каком бы месте ни находился курсор. На следующей строке появятся атрибуты следующей вычислительной секции: [> |.

Двоеточие (:) – оператор отказа от вывода результата. Его используют, если результат вычислений является промежуточным, не требующим вывода на рабочий лист. Двоеточие может быть использовано при записи нескольких операторов в одной строке.

Пусть вводится выражение 3 + 5 * 6 – 1, после него ставится точка с запятой (ввод происходит красным цветом) и нажимается *Enter.* В центре следующей строки появится результат 32 (синим цветом). Левая скобка (квадратная), отмечающая первую вычислительную секцию, имеет длину, соответствующую размеру введенного выражения и размеру ответа (рис. 3.27). На следующей строке появятся знаки следующей вычислительной секции.

Рис. 3.27. Пример вычислений

Если после ввода выражения 3 + 5 * 6 – 1 поставить двоеточие, то результат вычислений не будет выведен (рис. 3.28).

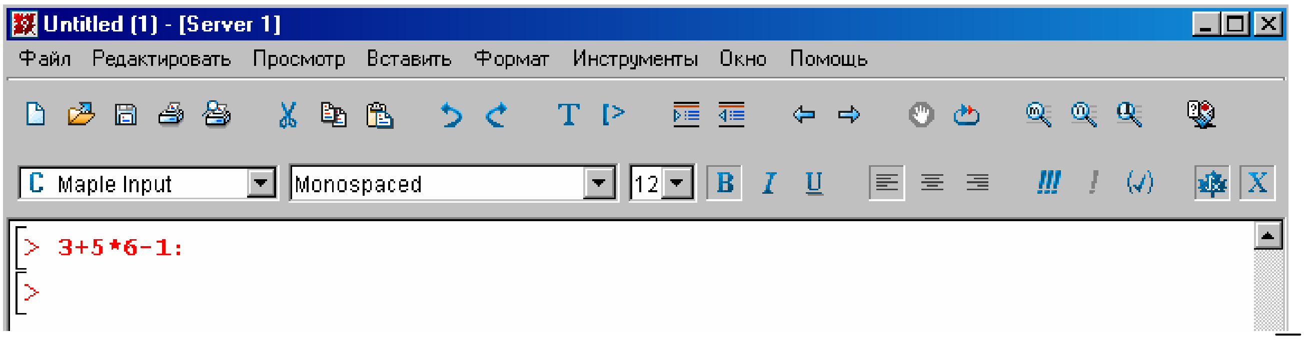


Рис. 3.28. Пример ввода выражения без выполнения вычислений

Оператор присваивания. ***Этот оператор используется для задания переменным определенного значения.***

Например, $I := 1$ – переменной I присваивается целочисленное значение, равное 1; $X := 12.36$ означает, что переменной X присваивается вещественное значение, равное 12,36; $f := 13 / 17$ – присваивает переменной f рациональное значение; $Y := X \wedge 2$ – переменной Y присваивается значение выражения X^2.

Правая часть выражения присваивает переменной, находящейся в левой части оператора присваивания, определенный тип. В рассматриваемых примерах – целочисленный, вещественный, рациональный.

На рис. 3.29 приведены примеры выполнения операторов присваивания. После каждого оператора стоит точка с запятой, поэтому результаты выполнения всех операторов выводятся. Результаты выполнения предыдущего оператора используются для выполнения следующего оператора.

```
> X:=2;
>
                                        X := 2
> Y:=X^3;
                                        Y := 8
> Z:=Y+1;
                                        Z := 9
>
```

Рис. 3.29. Примеры выполнения операторов присваивания

```
> X:=2: Y:=X^3: Z:=Y+1;
>
                                        Z := 9
>
```

Рис. 3.30. Другой вариант представления и выполнения операторов присваивания

На рис. 3.30 все операторы, которые были представлены на рис. 3.29, записаны в одной строке и разделены двоеточием и только в конце поставлена точка с запятой, поэтому выдается только окончательный результат. Все операторы занимают одну вычислительную секцию.

Операторы подстановки %, %%, %%%. Последний, пред-последний и предпредпоследний результаты вычислений сохра-няются и могут быть вызваны операторами %, %%, %%% для подстановки в строку ввода или в выражение. На рис. 3.31 демонстрируется использование оператора %: результат вычисления первого выражения, равный 32, не выведен, так как после него поставлено двоеточие. В этой же строке записано выражение, в соответствии с которым к последнему результату, получаемому с помощью оператора %, следует прибавить единицу. После него стоит точка с запятой, поэтому результат, равный 32 + 1 = 33, выведен на рабочий лист.

Используя оператор подстановки, вычисления, представленные на рис. 3.30, можно записать иначе (рис. 3.32).

```
> 3+5*6-1:%+1;
                                        33
>
```

Рис. 3.31. Вариант использования оператора подстановки

```
> x:=2: y:=%^3:Z:=%+1;
                                        Z := 9
>
```

Рис. 3.32. Другой вариант использования оператора подстановки

Отмена операции присваивания. *Maple* сохраняет в памяти все значения, присвоенные переменным на рабочем листе, поэтому результаты вычисления одного выражения могут зависеть от результатов присваивания значений переменным, используемых для вычислений другого выражения. Чтобы этого избежать при вычислении того или иного выражения, следует снять определения с тех переменных, имена которых использовались ранее. Для отмены присваивания значения переменной следует присвоить этой переменной её имя, заключенное в прямые апострофы (рис. 3.33).

```
> n:='n';
```

$$n := n$$

```
> m:='m';
```

$$m := m$$

```
> n!/m!/(n-m)!;
```

$$\frac{n!}{m!\,(n-m)!}$$

```
> n:=5: m:=2: n!/m!/(n-m)!;
```

$$10$$

```
> |
```

Рис. 3.33. Пример выполнения отмены операций присваивания

Для отмены операции присваивания сразу всем переменным рабочего листа можно использовать команду ***restart***.

Встроенные функции. В *Maple* имеется большое количество встроенных функций. В зависимости от частоты применения, функции разделены на внутренние, входящие в ядро *Maple* и загружаемые в оперативную память компьютера при запуске системы, и внешние, остающиеся на жестком диске.

Одни из них загружаются с помощью команды ***readlib(name)***, другие – вызовом всего пакета, в котором они находятся. В *Maple* имеется около семидесяти пакетов, которые вызываются командой ***with(name)***. Со списком пакетов можно ознакомиться командой ***?packages;***. Справку о функциях можно получить, задав вопрос ***?functions***.

Встроенные функции делятся на используемые и инертные. Используемые функции начинаются со строчной буквы и выдают результат. Инертные функции начинаются с прописной буквы. Они не вычисляются, а дают вывод математических выражений в естественной математической форме. Их можно применять для ввода формул в текстовые комментарии.

Функция задается вводом ее имени и списком параметров функции (одного или нескольких), заключенных в круглые скобки. Если параметров несколько, они отделяются друг от друга запятой.

На рис. 3.34 представлены выражения, в которые вставлены функции. Для представления функции, выведенной в символьном виде, в виде числа можно использовать функцию ***evalf( )*** (рис. 3.35).

```
> x:=3;
                                        x := 3
> sin(x)^2+cos(x)^2;
                                  sin(3)^2 + cos(3)^2
> x:=3.;
                                        x := 3.
> sin(x)^2+cos(x)^2;
                                     1.000000000
>
```

Рис.3.34. Примеры использования встроенных функций

```
> sin(3);
                                        sin(3)
> evalf(%);
                                     0.1411200081
>
```

Рис. 3.35. Другой пример использования встроенной функции

Ввод математических символов. Для ускорения ввода мате-матических символов в выражения удобно использовать палитру ВЫРАЖЕНИЕ. Она выводится на экран командой **Просмотр – Палитра – Выражение** (рис. 3.36). Появится данная палитра (рис. 3.37). Чтобы вставить в командную строку один из ее символов, надо щелкнуть по нему левой кнопкой мыши, и его шаблон появится на месте, занимаемом курсором ввода.

Пусть требуется ввести выражение для вычисления $\sqrt{x^2 + y^2}$. Для этого необходимо щелкнуть по кнопке $\sqrt{a}$. На месте курсора ввода появится шаблон с приглашением ввести величину подкоренного выражения (рис. 3.38). Введя подкоренное выражение и нажав кнопку *Enter*, получают результат (рис. 3.39).

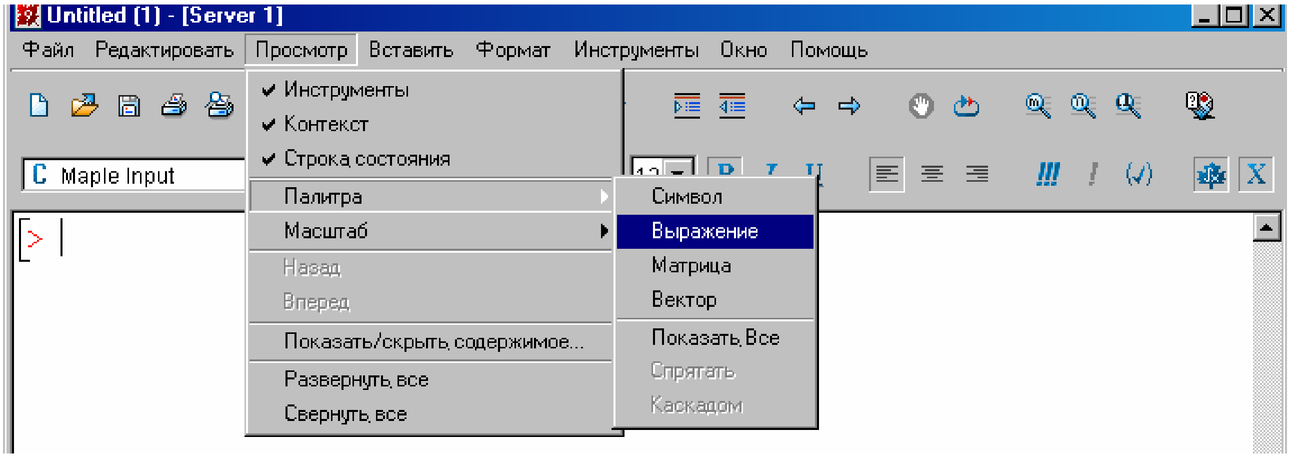


Рис. 3.36. Этапы вызова палитры ВЫРАЖЕНИЕ

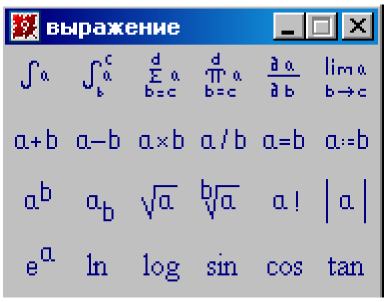


Рис. 3.37. Вид палитры ВЫРАЖЕНИЕ

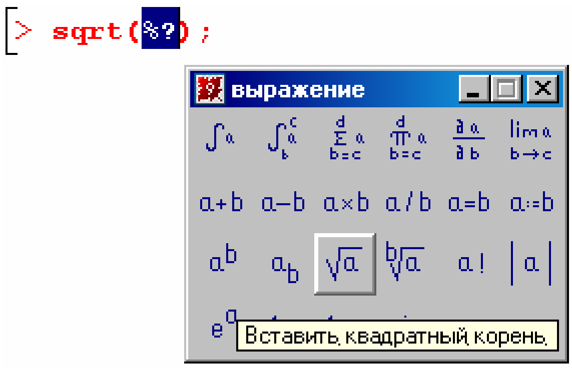


Рис. 3.38. Пример приглашения ввода подкоренного выражения

```
> sqrt(x^2+y^2);
```

$$\sqrt{x^2+y^2}$$

```
> x:=3:y:=4:
> sqrt(x^2+y^2);
```

$$5$$

```
>
```

Рис. 3.39. Примеры вычисления величины квадратного корня

Функции пользователя. Самым простым способом задания функции пользователя является присвоение некоторой переменной аналитического выражения:

имя : = выражение.

Значение функции вычисляет встроенная функция подстановки ***subs*(*x*=*a*, *f*)**. Например, определим функцию *f*(*x*) = *x*^2 + 5 и вычислим ее значения при *x* = 3 и *x* = 10 (рис. 3.40). При таком задании функции нет объявленного списка параметров, от которых зависит значение функции, в ней используются только глобальные переменные и значения переменных приходится задавать отдельно.

```
> f:=x^2+5;
```

$$f := x^2 + 5$$

```
> subs(x=3,f);
```

$$14$$

```
> subs(x=10,f);
```

$$105$$

Рис. 3.40. Пример задания функции пользователя без объявления списка параметров

Другой способ задания функции лишен этих недостатков и основан на применении функционального оператора. Используется следующая конструкция:

имя := (*x*, *y*, …) -> выражение,

где в круглых скобках записывается один или несколько формальных параметров определяемой функции. Переменные, указанные в списке формальных параметров, являются локальными. Они сохраняют свои значения только в теле функции.

Вызов функции осуществляется в следующем виде:

имя := (*x*1, *y*1, …),

где в круглых скобках указан список фактических параметров функции. При обращении к функции формальным параметрам присваиваются значения фактических параметров, и происходит вычисление функции. Определим функцию $f(x) = x$^2 + 5 этим способом и вычислим ее значения при $x = 3$ и $x = 10$ (рис. 3.41).

```
> f1:=(x)->x^2+5;
                                        f1 := x -> x^2 + 5
> f1(3);
                                        14
> f1(10);
                                        105
```

Рис. 3.41. Пример задания функции пользователя с объявлением формальных параметров

Функцию можно задать как процедуру программирования:

имя := *proc*(переменные)

выражение
end;

Например, составим такую процедуру для вычисления функции x^2+5x+1 (рис. 3.42).

```
> y:=proc(x)
> x^2+5*x+1
> end;
                    y := proc(x) x^2 + 5*x + 1 end proc
> y(3);
                    25
>
```

Рис. 3.42. Пример задания процедуры

Условные выражения. Для реализации разветвляющихся вычислений в *Maple*-языке имеются условные операторы. Чаще всего используются операторы двух видов:

if **<**условие **>** ***then*** **<** выражение **>** ***fi*** **–**

если условие выполняется, то будет выполнено выражение, записанное после ***then,*** в противном случае ничего не выполняется.

if **<**условие **>** ***then*** **<** выражение 1**>** ***else*** **<**выражение 2**>** ***fi*** **–**

если условие выполняется, то будет выполнено выражение, записанное после ***then,*** в противном случае – выражение, записанное после ***else.***

fi **–** признак конца оператора. Для записи условия используются любые логические выражения со знаками <, <=, >, >=, =, <> и логические операторы *and, or, not,* результатами выполнения которых являются логические значения *true* и *false.*

Примеры использования условных операторов приведены на рис. 3.43.

```
> x:=25: if x>=25 then print(`зачет`)fi;
                                                        зачет
> x:=10: if x>=25 then print(`зачет`)fi;
>
> x:=25: if x>=25 then print(`зачет`)else print(`незачет`)fi;
                                                        зачет
> x:=10: if x>=25 then print(`зачет`)else print(`незачет`)fi;
                                                       незачет
```

Рис. 3.43. Примеры использования условных операторов

Можно использовать и конструкцию следующего вида:

`` `if`(условие, выражение 1, выражение 2). ``

Если условие выполняется, то будет реализовано выражение 1, в противном случае – выражение 2. Примеры приведены на рис. 3.44.

```
> x:=25:
> `if`(x>=25, print(`зачет`), print(`незачет`));
                                                        зачет
> x:=10:
> `if`(x>=25, print(`зачет`), print(`незачет`));
                                                       незачет
```

Рис. 3.44. Примеры выполнения условных операторов

Условные операторы позволяют определять функции, заданные несколькими выражениями. Пусть требуется задать функцию вида:

$$f(x)=\begin{cases} 0, & \text{ïðè} \quad \tilde{o}<=0, \\ 3x, & \text{ïðè} \quad 0<x<=1, \\ 1 & \text{ïðè} \quad \tilde{o}>1. \end{cases}.$$

Один из возможных вариантов задания представлен на рис. 3.45.

```
> x:=0;
```

$$x := 0$$

```
> FV:=`if`(x<=0,0,`if`(x<=1,3*x,1));
>
```

$$FV := 0$$

```
> x:=0.5;
```

$$x := 0.5$$

```
> FV:=`if`(x<=0,0,`if`(x<=1,3*x,1));
```

$$FV := 1.5$$

```
> x:=3;
```

$$x := 3$$

```
> FV:=`if`(x<=0,0,`if`(x<=1,3*x,1));
```

$$FV := 1$$

Рис. 3.45. Определение функции, заданной несколькими выражениями

Операторы цикла. Для циклического повторения выполнения выражения используют операторы цикла. Оператор цикла может описывать повторение выполнения выражения заданное число раз или до тех пор, пока выполняется определенное условие. В первом случае оператор цикла имеет следующую структуру:

```
for <имя> from <нач. знач.> to <кон.знач.> by <шаг изменения>
          do <набор объектов> od;    (или end do).
```

Примеры выполнения операторов цикла представлены на рис. 3.46. Во втором случае оператор цикла имеет такую структуру:

```
for <имя> from <нач. знач.> by <шаг изменения> while  <условие>
          do <набор объектов> od;    (или end do).
```

На рис. 3.47 показан пример выполнения этих операторов.

```
> for i from 1 to 5 do i^2 end do;
                                        1
                                        4
                                        9
                                        16
                                        25
> for j from 1 by 2 to 5 do j^2 od;
                                        1
                                        9
                                        25
```

Рис. 3.46. Пример выполнения операторов цикла

```
> for i from 1 while i<=5 do i^2 od;
                                        1
                                        4
                                        9
                                        16
                                        25
> for i from 1 by 2 while i<=5 do i^2 od;
                                        1
                                        9
                                        25
```

Рис. 3.47. Другой пример выполнения операторов цикла

Имеется еще и такая структура оператора цикла:

for <имя> ***in*** <список значений> ***do*** <набор объектов> ***od;***

for <имя> ***in*** <список значений> ***while*** <условие> ***do*** <набор объектов> ***od;***

В операторах такой структуры задается список значений, которые будет принимать переменная. Цикл будет выполняться, пока не будет исчерпан список и пока выполняется заданное условие (рис. 3.48).

```
> for i in [5, 10, 15, 20] do i/5 od;
                                                    1
                                                    2
                                                    3
                                                    4
> for i in [5,10,-15,-20] while i>0 do i/5 od;
                                                    1
                                                    2
```

Рис. 3.48. Пример выполнения операторов цикла с заданным списком значений переменных

Циклы могут быть вложенными. Пусть требуется создать матрицу, размерностью 5 × 3, элемент которой равен произведению номера строки на номер столбца, на пересечении которых находится этот элемент (рис. 3.49).

```
> M:=array(1..5,1..3);
```

$$M := \mathrm{array}(1..5, 1..3, [\ ])$$

```
> for i from 1 to 5 do for j from 1 to 3 do M[i,j]:=i*j od od;
> evalm(M);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{bmatrix}$$

Рис. 3.49. Пример использования вложенного цикла

Построение графиков. В *Maple* имеются огромные возможности для построения математических графиков различных типов.

Двумерные графики отображают на плоскости зависимости типа *f*(*x*). Для их построения имеется функция ***plot,*** которая задается в следующем виде:

plot (f, x, o),
plot (f, x, o, s),

где ***f*** – функция (или функции), ***x*** – переменная с указанием области ее изменения, ***o*** – необязательная переменная, указывающая область изменения, ***s*** – набор параметров, задающих стиль построения графика (цвет, толщину, тип кривых и т.д.).

Самыми простыми формами задания этой функции являются:

plot(f, x=$x_{\min}$***.… x***$_{\max}$***),***

где f – имя функции, x_{min} ... x_{max} – диапазон изменения аргумента функции;

$$plot(f(x),\ x{=}x_{min} \ldots x_{max}),$$

где $f(x)$ – функциональная зависимость.

Обычно графики по умолчанию строятся в достаточно приемлемом виде, но можно задать управляющие параметры и в явном виде. Например, можно задать цвет кривых параметром ***color***, стиль построения графика ***style*** (***point*** – точечный, ***line*** – линиями), ***labels*** – надписи по координатным осям в виде **[*x*, *y*]**, где ***x*** и ***y*** – надписи по осям ***x*** и ***y*** графика.

Если для построения графика функции ***f*** предполагаете воспо-льзоваться функцией ***plot (f, x =*** x_{min} … x_{max}**)**, следует предварительно в командную строку ввести ее аналитическое выражение.

10.2. Расчет характеристик СМО

Процесс «размножения и гибели». В формуле (3.20) для расчета вероятности p_0 используются λ_{ij}.

В общем случае выражение для p_0 - есть единица, деленная на единицу, сложенную с суммой частного от деления произведения $\lambda_{k,k+1}$ при изменении переменной k от 0 до i на произведение $\lambda_{k+1,k}$ при изменении переменной k тоже от 0 до i. Поэтому для вычисления p_0 нужно реализовать операции многократного умножения и сложения.

Реализация элементарной операции сложения выглядит так:

```
> a:=2; b:=5; x:=a+b;
                                        a := 2
                                        b := 5
                                        x := 7
```

А для выполнения операции многократного сложения можно использовать стандартную функцию суммирования ***sum*** с параметрами:

```
> 1+Sum('k','k'=1..n);
```

$$1+\left(\sum_{k=1}^{n} k\right)$$

Кроме того, необходимо использовать стандартную функцию вычисления произведения ***product***:

```
> product(lambda[k,k+1],k=0..i);
```

$$\prod_{k=0}^{i} \lambda_{k,\,k+1}$$

В итоге вероятности p_0 и p_k рассчитываются приведенными ниже командными строками:

```
> p[0]:=(1+sum(product(lambda[k,k+1],k=0..i)/product(lambda[k+1,k],k=0..i),i=0..n-1))^(-1);
```

$$p_0 := \frac{1}{1+\left(\sum_{i=0}^{n-1} \frac{\prod_{k=0}^{i} \lambda_{k,k+1}}{\prod_{k=0}^{i} \lambda_{k+1,k}}\right)}$$

```
> p:=p[0]*product(lambda[k,k+1],k=0..i+1)/product(lambda[k+1,k],k=0..i+1);
```

$$p := \frac{\prod_{k=0}^{i+1} \lambda_{k,k+1}}{\left(1+\left(\sum_{i=0}^{n-1} \frac{\prod_{k=0}^{i} \lambda_{k,k+1}}{\prod_{k=0}^{i} \lambda_{k+1,k}}\right)\right)\left(\prod_{k=0}^{i+1} \lambda_{k+1,k}\right)}$$

Например, пусть:

$$\lambda_{0,1} = 1; \quad \lambda_{1,2} = 2; \quad \lambda_{2,3} = 3;$$
$$\lambda_{1,0} = 0.5; \quad \lambda_{2,1} = 1.5; \quad \lambda_{3,2} = 2.5 .$$

Можно подставить данные значения в исходные формулы и использовать функцию ***restart*** для реализации вычислений. В общем случае эта функция позволяет отменить все предыдущие присваивания.

```
> restart:lambda[0,1]:=1; lambda[1,2]:=2; lambda[2,3]:=3; lambda[1,0]:=0.5; lambda[2,1]:=1.5;
lambda[3,2]:=2.5; n:=3;
p[0]:=(1+sum(product(lambda[k,k+1],k=0..i)/product(lambda[k+1,k],k=0..i),i=0..n-1))^(-1);
```

$$\lambda_{0,1} := 1$$

$$\lambda_{1,2} := 2$$

$$\lambda_{2,3} := 3$$

$$\lambda_{1,0} := 0.5$$

$$\lambda_{2,1} := 1.5$$

$$\lambda_{3,2} := 2.5$$

$$n := 3$$

$$p_0 := 0.1127819549$$

В итоге для заданных значений λ_{ij} и n вычислено p_0 = 0.1127819549.

Затем, используя формулу (3.19), можно записать выражение для расчета остальных вероятностей состояний (p_n):

```
> p:=p[0]*product(lambda[k,k+1],k=0..i-1)/product(lambda[k+1,k],k=0..i-1);
```

$$p := \frac{0.1127819549 \left(\prod_{k=0}^{i-1} \lambda_{k,k+1} \right)}{\prod_{k=0}^{i-1} \lambda_{k+1,k}}$$

.

В приведенном фрагменте программы p это процедура с формальным параметром i.

Используя эту процедуру, можно записать выражения для вычисления p_i с i, принимающим значения 1, 2 и 3. Получаемые выражения, формируют список из трёх элементов.

```
> [subs(i=1,p),subs(i=2,p),subs(i=3,p)];
```

$$\left[\frac{0.1127819549\left(\prod_{k=0}^{0}\lambda_{k,k+1}\right)}{\prod_{k=0}^{0}\lambda_{k+1,k}}, \frac{0.1127819549\left(\prod_{k=0}^{1}\lambda_{k,k+1}\right)}{\prod_{k=0}^{1}\lambda_{k+1,k}}, \frac{0.1127819549\left(\prod_{k=0}^{2}\lambda_{k,k+1}\right)}{\prod_{k=0}^{2}\lambda_{k+1,k}}\right]$$

Вычислить числовые значения для p_i можно, используя встроенную функцию ***evalf***:

```
> evalf(%);
```

$$[0.2255639098, 0.3007518798, 0.3609022557]$$

.

В результате расчета получены следующие величины:

$$p_0 \approx 0.113,$$
$$p_1 \approx 0.225,$$
$$p_2 \approx 0.301,$$
$$p_3 \approx 0.361.$$

Многоканальная СМО с отказами. Система массового обслуживания, описываемая схемой М | М | n | 0, является простейшей из всех СМО – это многоканальная система с отказами.

Задание выражений (3.21) в форме процедуры программирования выглядит так:

```
> p:=proc(ro,n)
> local q;
> q[0]:=(sum(ro^i/i!,i=1..0))^(-1);
> [q[0],[q[0]*ro^i/i!,i=1..n]];
```

$$\left[q_0, \left[\frac{q_0\, ro^i}{i!}, i = 1..n\right]\right]$$

Соответственно можно найти характеристики эффективности этой СМО: $P_{отк}$, q и A.

Пример 3.14. В СМО, в которой два обслуживающих канала, приходит в среднем 8 заявок на обслуживание в час, причем если каналы заняты, то заявки уходят не обслуженными. Среднее время обслуживания одной заявки 1/6 часть часа. Содержание одного рабочего места (обслуживающего канала) составляет 100 руб в час, а доход от обслуживания одной заявки 50 руб.

1. Найти характеристики эффективности работы СМО.
2. Определить доход, полученный за час функционирования двух каналов.
3. Найти характеристики эффективности работы СМО, если будут функционировать три обслуживающих канала. Выгодно ли создавать третий канал с точки зрения общего дохода, получаемого за 1 час работы системы?

Решение. 1. По условию задачи $n = 2$, $\lambda = 8$, $\mu = 6$, $\rho = \lambda / \mu = 4/3$. С помощью введенной процедуры находим:

```
> p (4 / 3, 2);
```

$$\left[\frac{9}{29}, \left[\frac{12}{29}, \frac{8}{29}\right]\right]$$

```
> evalf (%);
```

$$[.3103448276,\ [.4137931034,\ .2758620690]]$$

Следовательно, вероятность отказа $P_{\text{отк}} = 0.276$, относительная пропускная способность $q = 0.724$, абсолютная пропускная способность $A = \lambda q = 5.792$, среднее число занятых рабочих мест $\bar{k} = A / \mu = 0.965$.

2. Средний доход за один час при двух каналах $D_2 = 50A = 289.6$ руб.
3. Замена в процедуре $n = 2$ на $n = 3$ даёт:

```
> p (4 / 3, 3);
```

$$\left[\frac{81}{293}, \left[\frac{108}{293}, \frac{72}{293} \frac{32}{293}\right]\right]$$

```
> evalf (%);
```

$$[.2764505119,\ .3686006826,\ .2457337884,\ .1092150171]].$$

В этом случае доход D_3, получаемый от одного канала за один час:

```
> 50 * 8 * (1 - p (4 / 3, 3) [2, 3]);
```

$$\frac{104400}{293}$$

```
> evalf (%);
```

$$356.31399332.$$

Вывод: $D_3 = 356.3$ руб. Создание третьего канала не выгодно, так как $D_3 - D_2 = 356.3 - 289.6 = 66.7 < 100$, где 100 руб. – это стоимость содержания одного канала в час.

Выводы по третьей части учебного пособия

В теории моделирования к наиболее изученным и исследованным относятся модели, у которых случайный процесс функционирования принадлежит к классу марковских процессов, то есть марковские модели. Случайный процесс, протекающий в системе, называется марковским, если для любого момента времени вероятностные характеристики процесса в будущем зависят только от его состояния в данный момент времени и не зависят от того, когда и как система пришла в это состояние (см. главу 8). Понятие состояния зависит от целей моделирования. В одном случае, например, оно может быть определено по состояниям элементов, каждый из которых может быть «свободен» или «занят»; в другом случае состояние системы определяется числом заявок, находящихся на обслуживании и в очереди.

При исследовании вычислительных систем аналитическим моделированием наибольшее значение имеют марковские случайные процессы с дискретными состояниями и непрерывным временем. Процесс называется процессом с дискретными состояниями, если его возможные состояния удаётся заранее перечислить, то есть состояния системы принадлежат конечному множеству, и переход системы из одного состояния в другое происходит мгновенно (см. § 8.1).

Процесс называется процессом с непрерывным временем, если смена состояний может произойти в любой случайный момент. Существует методика составления системы дифференциальных уравнений Колмогорова, решая которые можно вычислить вероятности состояний системы для любого момента времени (см. § 8.2).

В теории случайных процессов доказывается, что если число состояний системы конечно и из каждого состояния можно перейти в любое другое за конечное число шагов, то существуют предельные вероятности состояний. Предельную вероятность состояния трактуют как среднее относительное время пребывания системы в этом состоянииии. Для их вычисления все левые части уравнений Колмогорова приравниваются нулю и затем решается полученная система линейных алгебраических уравнений (см. § 8.3).

Разновидностью марковской модели с дискретным числом состояний и непрерывным временем является модель (процесс) «размножения и гибели» (см. § 8.4). Он характеризуется тем, что граф состояний имеет вид цепи (см. рис. 3.10). Особенность этого графа состоит в том, что каждое из средних состояний связано прямой и обратной стрелками с каждым из соседних состояний – правым и левым, а крайние состояния – только с одним соседним состоянием. В этой модели (процессе «размножения и гибели») формулы для определения вероятностей состояний (3.19) и (3.20), полученные в результате решения уравнений Колмогорова, широко используются при решении задач теории массового обслуживания.

Области применения методов теории массового обслуживания постоянно расширяются. Анализ функционирования СМО значительно упрощается, если случайный процесс, протекающий в системе, является марковским (см. гл. 9). В этой главе рассмотрен ряд простейших моделей СМО, начиная от одноканальной с отказами (M | M | 1 | 0) и заканчивая многоканальной с неограниченной очередью (M | M | *n* | ∞). Следует помнить, что все модели связаны с предположением о стационарности и ординарности потока заявок, а также отсутствии в нём последействия (простейший поток). Важную роль играет и показательный закон распределения длительности обслуживания заявок.

При необходимости проведения многочисленных повторяющихся расчётов характеристик СМО естественно возникает потребность автоматизации этого процесса. Решением этой проблемы может быть использование одного из многочисленных языков программирования: Си, Паскаль и пр. Особенно, если исследователь владеет одним из них. Если же разработчику необходимо быстро освоить методику компьютерных расчётов, то перспективным может быть использование одного из широко распространяемых пакетов для инженерных расчетов: *Mathcad, Maple* и пр. Глава 10 посвящена именно освоению методики инженерных расчётов в среде *Maple*. Приводятся примеры.

Если же аналитическое моделирование по каким-либо причинам невозможно, то приходится обращаться к методу исследования, рассмотренному во второй части пособия, – имитационному моделированию.

Заключение

Данное учебное пособие посвящено изложению методологии моделирования и использованию компьютерных технологий для реализации различных процессов в системах массового обслуживания или просто реальными устройствами. Устройство или процесс называют системой. Для научного исследования системы прибегают к некоторым допущениям, касающимся её функционирования. Описание процесса функционирования в виде математических или логических зависимостей формирует модель, которая дает представление о поведении системы. Набор процедур формирования модели и её исследования создаёт технологию моделирования, изложенную в первой части пособия.

Если зависимости, образующие модель, довольно просты для получения точной информации о характере функционирования системы, то используется аналитическое моделирование (см. Часть III данного пособия).

В большинстве практических случаев существующие системы являются очень сложными и для них не представляется возможным создать реальную модель, описанную аналитически. Такие модели обычно исследуют с помощью имитационного моделирования (см. Часть II пособия).

Компьютерные реализации моделей позволяют получать численные оценки характеристик систем (см. главы 7 и 10), а с помощью полученных данных можно рассчитать и реальные характеристики. Результаты компьютерных экспериментов позволяют развеять существующее впечатление о моделировании как упражнении в программировании, пусть и очень сложном.

Дальнейшее развитие методологии моделирования может быть получено при изучении студентами специальности 230101.65 следующих дисциплин учебного графика: введение в искусственный интеллект, математические основы теории систем, методы оптимизации, проектирование информационных систем, экспертные системы и пр.

Библиографический список

1. *Альянах И.Н.* Моделирование вычислительных систем / И.Н. Альянах. – Л.: Машиностроение. Ленинград. отд-ние, 1988. – 223 с.

2. *Андриевский Б.Р.* Элементы математического моделирования в программных средах MATLAB 5 и Scilab / Б.Р. Андриевский, А.Л. Фрадков. – СПб.: Наука, 2001. – 286 с.

3. *Анкудинов Г.И.* Математическая логика и теория алгоритмов: учеб. пособие / Г.И. Анкудинов, И.Г. Анкудинов, О.А. Петухов. – СПб.: СЗТУ, 2003. – 104 с.

4. *Анкудинов Г.И.* Теория автоматов: учеб. пособие / Г.И. Анкудинов, И.Г. Анкудинов, Р.Р. Хамидуллин. – СПб.: СЗТУ, 2003. – 112 с.

5. *Боброва Л.В.* Методы и средства обработки экономической информации: учебно-методический комплекс (учебное пособие) / Л.В. Боброва. – СПб.: Изд-во СЗТУ, 2008. – 217 с.

6. *Бригаднов И.А.* Методы вычислительной математики: учеб. пособие / И.А. Бригаднов. – СПб.: СЗТУ, 2001. – 83 с.

7. *Бусленко Н.П.* Моделирование сложных систем / Н.П. Бусленко. – М.: Наука, 1978. – 400 с.

8. *Вентцель Е.С.* Исследование операций / Е.С. Вентцель. – М.: Сов. радио, 1972. – 551 с.

9. *Вентцель Е.С.* Исследование операций: задачи, принципы, методология / Е.С. Вентцель. – М.: Наука, 1980. – 208 с.

10. Вероятностные методы в вычислительной технике / Под ред. А.Н. Лебедева и Е.А. Чернявского. – М.: Высшая школа, 1986. – 312 с.

11. *Готшальк О.А.* Теория массового обслуживания: письменные лекции / О.А. Готшальк. – СПб.: СЗТУ, 2000. – 42 с.

12. *Гультяев А.К.* Визуальное моделирование в среде MATLAB: учебный курс / А.К. Гультяев. – СПб.: Питер, 2000. – 432 с.

13. *Гультяев А.К.* Имитационное моделирование в среде Windows: практич. пособие / А.К. Гультяев. – СПб.: КОРОНА принт, 2001. – 400 с.

14. *Дейтел Л.* Введение в операционные системы: В 2 т. Т. 2. / Л. Дейтел. – М.: Мир, 1987. – 298 с.

15. *Дьяконов Д.* Maple 7: учебный курс / Д. Дьяконов. – СПб.: Питер, 2002. – 672 с.

16. *Калинина В.Н,* Математическая статистика: учебник / В.Н. Калинина, В.Ф. Панкин. - 3-е изд.– М.: Высш. шк., 2001. – 336 с.

17. *Карпов Ю.Г.* Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 / Ю.Г. Карпов. – СПб.: БХВ-Петербург, 2005. – 400 с.

18. *Кельтон В., Лоу А.* Имитационное моделирование: 3-е изд. Пер. с англ. / В. Кельтон, А. Лоу. – СПб.: Издательская группа BHV, 2004. – 847 с.

19. *Кнут Д.* Искусство программирования для ЭВМ: В 3 т. Т. 2. / Д. Кнут. – М.: Мир, 1977. – 724 с.

20. *Колесников А.В., Петухов О.А.* Моделирование систем: учеб. пособие / А.В. Колесников, О.А. Петухов. – Л.: СЗПИ, 1981. – 72 с.

21. *Кудрявцев Е.М.* GPSS World. Основы имитационного моделирования различных систем / Е.М. Кудрявцев. – М.: ДМК Пресс, 2004. – 320 с.

22. *Кузин Л.Т.* Основы кибернетики: В 2 т. Т. 2. Основы кибернетических моделей / Л.Т. Кузин. – М.: Энергия, 1979. – 584 с.

23. *Курицкий Б.Я.* Поиск оптимальных решений средствами Excel 7.0 / Б.Я. Курицкий. – СПб.: BHV – Санкт-Петербург, 1997. – 384 с.

24. *Лифшиц А.Л.* Статистическое моделирование систем массового обслуживания / А.Л. Лифшиц, Э.А. Мальц. – М.: Сов. радио, 1978. – 248с.

25. Моделирование. Аналитические модели: учеб. пособие / О.А. Петухов, И.А. Бригаднов, А.В. Морозов, Е.О. Петухова. – СПб.: Изд-во СЗТУ, 2005. – 112 с.

26. Моделирование. Имитационные модели: учеб. пособие / О.А. Петухов, И.А. Бригаднов, А.В. Морозов, Е.О. Петухова. – СПб.: Изд-во СЗТУ, 2005. – 133 с.

27. *Морозов А.В.* Математические основы теории систем: учеб. пособие / А.В. Морозов, И.А. Бригаднов, Р.Р. Хамидуллин. – СПб.: СЗТУ, 2004. – 179 с.

28. *Морозов А.В.* Элементы теории массового обслуживания: учебно-методическое пособие / А.В. Морозов, Н.В. Назарова. – СПб.: ВТУЖВРФ, 2005. – 56 с.

29. *Нейлор Т.* Машинные имитационные эксперименты с моделями экономических систем: Пер. с англ. / Т. Нейлор. – М.: Мир, 1975. – 500 с.

30. Нечёткие модели: учеб. пособие / О.А. Петухов, И.А. Бригаднов, Р.Р. Хамидуллин, Е.О. Петухова. – 2-е изд., исправл. – СПб.: СЗТУ, 2007. – 92 с.

31. *Николаев В.И.* Системотехника: методы и приложения / В.И. Николаев, В.М. Брук. – Л.: Машиностроение, 1985. – 199 с.

32. *Николаев В.И.* Дискретные структуры. Основы теории: учеб. пособие / В.И. Николаев, О.А. Петухов, Р.Р. Хамидуллин. – СПб.: СЗПИ, 1999. – 154 с.

33. *Петухов О.А.* Модели систем массового обслуживания: учеб. пособие / О.А. Петухов. – Л.: СЗПИ, 1989. – 86 с.

34. *Петухов О.А.* Экспертные системы. учеб. пособие / О.А. Петухов. – СПб.: СЗТУ, 2003. – 152 с.

35. *Петухов О.А.* PDC Prolog – язык систем искусственного интеллекта: учеб. пособие / О.А. Петухов. – 4-е изд., испр. – СПб.: СЗТУ, 2006. – 172 с.

36. *Петухова Н.М.* Информатика. *Microsoft Excel* 2000. Основные приёмы работы: учеб. пособие. – 4-е изд., исправл. и доп. / Н.М. Петухова, Е.О. Петухова. – СПб.: Изд-во СЗТУ, 2006. – 186 с.

37. *Рыжиков Ю.И.* Имитационное моделирование. Теория и технологии / Ю.И. Рыжиков. – СПб.: КОРОНА принт; М.: Альтекс-А, 2004. – 384 с.

38. *Сдвижников О.А.* Математика на компьютере: Maple 8 / О.А. Сдвижников. – М.: СОЛОН-Пресс, 2003. – 176 с.

39. *Смирнова Н.А.* О получении модельных реализаций случайных векторов / Н.А. Смирнова // Проблемы машиноведения и машиностроения: Межвуз. сб. Вып.21. – СПб.: СЗТУ, 2000. – С. 53 – 57.

40. *Смирнова Н.А.* Моделирование СМО с учётом приоритета заявок / Н.А. Смирнова, О.А. Петухов. // Проблемы машиноведения и машиностроения: Межвуз. сб. Вып. 27. – СПб.: СЗТУ, 2002. – С.152 – 158.

41. *Советов Б.Я.* Моделирование систем: учебник для вузов / Б.Я. Советов, С.А. Яковлев. – М.: Высш. школа, 2005. – 343 с.

42. *Томашевский В.Н.* Имитационное моделирование в среде GPSS / В.Н. Томашевский, Е.Г. Жданова. – М.: Бестселлер, 2003. – 416 с.

43. *Фомин Г.П.* Математические методы и модели в коммерческой деятельности: учебник / Г.П. Фомин. – М.: Финансы и статистика, 2001. – 544 с.

44. *Шеннон Р.* Имитационное моделирование систем - искусство и наука / Пер. с англ. / Р. Шеннон. – М.: Мир, 1975. – 420 с.

45. *Шрайбер Т.Д.* Моделирование на GPSS / Т.Д. Шрайбер. – М.: Машиностроение, 1980. – 592 с.

Предметный указатель

Оглавление

Олег Александрович Петухов
Алексей Валентинович Морозов
Елизавета Олеговна Петухова

Моделирование
системное
имитационное
аналитическое

Учебное пособие

Редактор Т.В. Шабанова
Сводный темплан 2008 г.
Лицензия ЛР № 020308 от 14.02.97

Санитарно-эпидемиологическое заключение № 78.01.07.953.П.005641.11.03 от 21.11.2003г.

Подписано в печать . .2008 г. Формат 60 х 84 1/16.
Б. кн.-журн. П. л. 18,00. Б.л. 9,00. Изд-во СЗТУ
Тираж 100. Заказ

Северо-Западный государственный заочный технический университет
Издательство СЗТУ, член Издательско-полиграфической ассоциации
университетов России
191186, Санкт-Петербург, ул. Миллионная, 5