Современная
Гуманитарная
Академия

Дистанционное образование

1233.01.01;2 Рабочий учебник

Фамилия, имя, отчество обучающегося _____
Направление подготовки _____
Номер контракта _____

МОДЕЛИРОВАНИЕ СИСТЕМ

ЮНИТА 1

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СИСТЕМ

МОСКВА 2005

Разработано Н.Б. Артемьевой
Под ред. А.П. Пятибратова, д-ра техн. наук, проф.

Рекомендовано Учебно-методическим советом в качестве учебного пособия для студентов СГА

КУРС: МОДЕЛИРОВАНИЕ СИСТЕМ

Юнита 1. Математическое моделирование систем.
Юнита 2. Имитационное моделирование систем.
Юнита 3. Анализ и интерпретация результатов моделирования систем на ЭВМ.

ЮНИТА 1

Определяются цели и принципы моделирования, дается классификация моделей. Общая постановка задачи и постановка задачи в условиях неопределенности отдельно рассматриваются многокритериальные модели. Основные математические методы моделирования и их приложение. Разобраны примеры моделей.

Для студентов Современной Гуманитарной Академии

1233.004.01.05.01;2/

ОГЛАВЛЕНИЕ

* Глоссарий расположен в середине учебного пособия и предназначен для самостоятельного заучивания новых понятий.

ДИДАКТИЧЕСКИЙ ПЛАН

Основные понятия и принципы моделирования систем, классификация моделей. Искусство и концепции моделирования. Общая постановка задачи. Общая постановка задачи в условиях неопределенности. Многокритериальные модели. Концептуальные модели и математические схемы моделирования систем. Сетевые модели. Линейные модели. Модели динамического программирования. Модели теории игр. Марковские модели. Модели массового обслуживания.

ЛИТЕРАТУРА

Основная

*1. Е.С.Вентцель Исследование операций. Задачи, принципы, методология. - М.: Высшая школа, 2001, 208 стр.

*2. Шикин Е.В.,Чхартишвили А.Г. Математические методы и модели в управлении. - М.: Дело, 2002, 440 стр.

Дополнительная

3. Вентцель Е.С. Динамическое программирование. - М.: Изд-во иностр. лит., 1980.

*4. Вагнер Г. Основы исследования операций, Том 1. - М.: Мир, 1972, 335 стр.

*5. П.Конюховский Математические методы исследования операций в экономике. - СПб.: Питер, 2000, 208 стр.

6. Адлер Ю.П. Статистические методы в имитационном моделирова-нии. - М.: Мир, 1990.

7. Советов Б.Я., Яковлев С.А. Моделирование систем. Учебник для вузов, - М.: Высшая школа, 1998.

*8. Карманов В.Г. Математическое программирование. - М.: Наука, 1975, 272 стр.

Примечание. Знаком (*) отмечены работы, на основе которых составлен тематический обзор.

ТЕМАТИЧЕСКИЙ ОБЗОР*

1. ВВЕДЕНИЕ

1.1. Модели систем в процессе принятия решений

Уровень развития науки и техники, достигнутый к настоящему времени, позволяет планировать и осуществлять мероприятия, в которые вовлекаются значительные ресурсы, мероприятия, масштабы, стоимость и последствия которых существенно превышают все, что было когда-либо ранее. Риски, связанные с реализацией подобных крупномасштабных мероприятий, в наше время усугубляются быстрой сменой поколений техники и технологии. В результате, требования со стороны техники и навыки обращения с ней должны сменяться уже несколько раз на протяжении одной человеческой жизни. Т.е. опытные люди, умеющие приводить эту технику в действие и разумно управлять ею, просто не успевают сформироваться. Поэтому в наши дни метод проб и ошибок становится уже неприемлемым: слишком катастрофическими могут быть последствия ошибок и слишком мало времени отпущено для проб. Становится все яснее, что сегодня меньше, чем когда-либо ранее, допустимы произвольные, чисто волевые решения.

На первый план выходит проблема организации и управления, причем управления не только (и не столько) машинами, но сложными человеко-машинными системами. А это означает, что ответственные решения должны приниматься на основе предварительных прикидок и расчетов. Не случайно, поэтому, в наше время наблюдается бурный рост математических методов во всех областях практики: вместо того чтобы пробовать и ошибаться по отношению к реальным объектам, предпочтительнее делать это на моделях. Формируется **исследование операций** (в англоязычной литературе — operations research/ management science или OR/MS) — наука о предварительном обосновании решений во всех областях целенаправленной деятельности, широко использующая математический аппарат, но не сводящаяся к нему.

Это означает, в частности, что помимо традиционных областей приложения — точных и опытных наук — математика начинает заниматься вопросами, которые от века изучались только на гуманитарном уровне: конфликтными ситуациями, иерархическими

---

* Жирным шрифтом выделены новые понятия, которые необходимо усвоить. Знание этих понятий будет проверяться при тестировании.

отношениями в коллективах, согласием, авторитетом, общественным мнением. Строятся и анализируются математические модели, применяются математические методы. Математические модели не только проникают в ранее чуждые для них области, но и трансформируется при этом, меняют свои методологические черты. Причина — в том, что явления, составляющие предмет гуманитарных наук, неизмеримо сложнее тех, которыми занимаются науки точные. Эти явления гораздо труднее поддаются формализации (если вообще поддаются), гораздо шире круг причин, от которых они зависит, и поэтому вербальный способ построения исследования, как это ни парадоксально, часто оказывается здесь полезнее формально — логического.

Какое же место отводится здесь математической составляющей? Прежде всего, математические методы можно рассматривать как достаточно эффективное средство структурированного, более компактного и обозримого представления имеющейся информации. Это особенно ясно в тех случаях, когда информация задается в виде числовых массивов, в графической форме и др. Анализ результатов математической обработки данных зачастую позволяет высказать некоторые рекомендации относительно тех или иных способов действия. При принятии решений в больших задачах с огромными объемами информации это играет очень важную роль. Кроме того, существует целый ряд типовых ситуаций принятия решений, допускающих формализацию по стандартным схемам, эффективность которых уже проверена на практике. Тогда математические подходы и соображения обоснованно становятся решающими.

Уже ранние работы (XVIII—XIX вв.) явились важным этапом развития и становления исследования операций. Пионерские разработки нового подхода к организации труда и производства, к учету человеческого фактора в промышленности, были предприняты А. Смитом (A. Smith), Ч. Бэббиджем (Ch. Babbage), Ф. Тейлором (F. Taylor), Г. Гэнтом (H. Gantt). В результате были получены эффективные решения целого ряда конкретных задач. Например, введение в Великобритании в 1840 г. почтовой оплаты в 1 пении, существенно упростившей процедуру обработки корреспонденции, явилось результатом анализа операций в почтовом ведомстве, предпринятого Бэббиджем. Он нашел, что большая часть стоимости письма приходится на его обработку при сортировке, а вовсе не на дальность путешествия от отправителя к получателю, как это считалось ранее.

Начало XX в. было отмечено первыми попытками составить математическую модель антагонистического конфликта (модель

Ф. Ланчестера исхода артиллерийской дуэли), создать теорию управления инвестициями (Ф. Харрис), теорию очередей (А. Эрланг). Однако, несмотря на заметные продвижения в разработке математических подходов к решению проблем управления, исследование операций как научное направление было признано лишь в 40—50-е годы XX в., когда существенный прорыв обозначился при решении проблем, возникших непосредственно перед и в ходе второй мировой войны. Наиболее известным примером здесь могут служить результаты работы британской группы экспертов, состоявшей из 11 человек, оказавшие заметное влияние на исход битвы за Англию и сражений в Северной Атлантике. В эту группу, которая стала потом известна под названием “Blackett’s Circus” (по имени руководителя П.М.С. Блэкетта), входили физиологи, математики, физики, геодезист, астрофизик и военный.

Специфика полученных результатов определенное время была сдерживающим фактором на пути их применения вне военной сферы. Однако заметное теоретическое продвижение в теории игр и теории полезности (Дж. фон Нейман) и линейном программировании (Дж. Данциг, Л.В.Канторович), а также создание электронных вычислительных машин обеспечили существенный прорыв в расширении области приложения операционного анализа. Многие задачи управления удалось достаточно хорошо формализовать, и сейчас они уже весьма широко и довольно успешно решаются стандартными методами исследования операций. Впрочем, зависимость методологии исследования операций от возможностей вычислительных средств не следует преувеличивать. Даже сегодня многие крупномасштабные задачи еще нельзя решить при помощи существующих высокоскоростных компьютеров.

Итак, в первой половине XX в. начали разрабатывать (и довольно успешно) элементы научного подхода к поиску решений при управлении, а схемы, хорошо показавшие себя при проведении естественнонаучных и инженерно-технических изысканий, стали пытаться приспосабливать к решению управленческих задач. Сравнительно быстро пришло понимание того, что для поиска путей перехода от фактически наблюдаемого состояния изучаемой системы к желаемому весьма существенно, насколько хорошо формализована решаемая задача.

Характер и степень формализации управленческой задачи во многом определяет и методику поиска ее решения. Различают хорошо структурированные, слабоструктурированные и неструктурированные задачи. Резкой грани между ними провести нельзя. Нередко оказывается, что (вначале) слабоструктурированная проблема

становится (потом) хорошо структурированной и даже стандартной, для решения которых построены хорошо зарекомендовавшие себя схемы. Именно об этих схемах по большей части и идет речь в настоящей юните.

Приведем некоторые данные об использовании математических подходов, методов и моделей в задачах управления 125 крупнейшими корпорациями США [из статьи: Guisseppi A. Forgionne. Corporate Management Science Activities: An Update, Interfaces, 13 (June 1983). P. 20—23], (табл.1).

Таблица 1

Метод, модель	Частота использования, % корпорапий		
	редко	умеренно	постоянно
Статистический анализ	2	38	60
Имитационное моделирование	13	53	34
Сетевое планирование	26	53	21
Линейное программирование	26	60	14
Теория массового обслуживания	40	50	10
Нелинейное программирование	53	39	8
Динамическое программирование	61	34	5
Теория игр	69	27	4

Исследование операций представляет собой применение математических инструментов для решения сложных проблем, связанных с функционированием больших систем людей, машин, материалов и денег в промышленной и деловой сфере, в государственном управлении, обороне и т.п. Его характерной особенностью является построение для соответствующей системы математической модели, которая обычно должна учитывать присутствующие факторы вероятности и риска. Модель позволяет рассчитать и сравнить результаты различных решений, стратегий и методов управления.

Основная задача исследования операций состоит в том, чтобы помочь менеджеру или иному лицу, принимающему решение, обосновать свою политику и избранный вариант действий среди всех возможных путей достижения поставленных целей. Коротко исследование операций можно назвать научным подходом к проблеме принятия решений. Проблема — это разрыв между желаемым и фактическим состояниями (прежде всего целями) той или иной системы. Решение — это средство преодоления такого рода разрыва, выбор одного из многих объективно существующих курсов действий, который позволил бы перейти от наблюдаемого состояния к желаемому.

В настоящее время под операцией понимается управляемое целенаправленное мероприятие (система действий), объединенных

общим замыслом, а под основной задачей исследования операций — разработка и исследование путей реализации этого замысла. Ясно, что такое весьма широкое понимание операции охватывает значительную часть деятельности людей. Однако наука о принятии решений, о поиске путей достижения цели и особенно ее математическая составляющая еще весьма далеки до завершения даже по основным вопросам.

Совокупность людей, организующих операцию и участвующих в ее проведении, принято называть оперирующей стороной. Следует иметь в виду, что на ход операции могут оказывать влияние лица и природные силы, далеко не всегда содействующие достижению цели в данной операции.

Во всякой операции существует лицо (группа лиц), облеченное полнотой власти и наиболее информированное о целях и возможностях оперирующей стороны и называемое руководителем операции или **лицом, принимающим решение** (ЛПР). ЛПР несет полную ответственность за результаты проведения операции.

Особое место занимает лицо (группа лиц), владеющее математическими методами и использующее их для анализа операции. Это лицо (**аналитик**) несет полную ответственность за соответствие модели требованиям технического задания на моделирование. Аналитик сам решений не принимает, а лишь помогает в этом оперирующей стороне. Степень его информированности определяется ЛПР. Так как аналитик, с одной стороны, не имеет всей информации об операции, которой обладает ЛПР, а с другой, как правило, более осведомлен в общих вопросах методологии принятия решений, то желательно, чтобы взаимоотношения между исследователем операции и оперирующей стороной имели характер творческого диалога. Результатом этого диалога должен быть выбор (или построение) математической модели операции, на основе которой формируется система объективных оценок конкурирующих способов действий, более четко обозначается окончательная цель операции и появляется понимание оптимальности выбора образа действий. Право оценки альтернативных курсов действий, выбора конкретного варианта проведения операции (принятие решения) принадлежит ЛПР. Это обусловлено еще и тем, что абсолютных критериев рационального выбора не существует — во всяком акте принятия решения неизбежно содержится элемент субъективизма. И только время — единственный объективный критерий — в конце концов может показать, насколько разумным было принятое решение.

Для того чтобы пояснить, какое место занимает математическая составляющая в исследовании операций, опишем коротко основные этапы разрешения проблемы принятия решения.

1. Формулировка проблемы. Это весьма существенный и нетривиальный шаг даже в том случае, когда формулировка проблемы звучит совсем просто. Определение реальной проблемы, а не описание ее симптомов требует понимания, интуиции, воображения и времени.

2. Выбор модели. В случае если проблема сформулирована корректно, появляется возможность выбора готовой модели (из числа моделей, описывающих стандартные ситуации), либо, если готовой модели нет, возникает необходимость создания такой модели, которая в достаточной степени отражала бы существенные стороны проблемы.

3. Поиск решения. Для поиска решения необходимы конкретные данные, сбор и подготовка которых требуют, как правило, значительных совокупных усилий. При этом часто, даже в случае, когда необходимые данные уже имеются, их часто приходится преобразовывать к виду, соответствующему выбранной модели.

4. Тестирование решения. Полученное решение обязательно должно быть проверено на приемлемость при помощи соответствующих тестов. Неудовлетворительность решения обычно означает, что модель не отражает истинную природу изучаемой проблемы. В этом случае она должна быть либо как-то усовершенствована, либо заменена на более подходящую модель.

5. Организация контроля. Если найденное решение оказывается приемлемым, и модель необходима не только для разрешения единичной сиюминутной ситуации, но и при рассмотрении сходных обстоятельств в будущем, возникает необходимость в реализации механизма контроля правильности использования модели. Основная задача контроля состоит в обеспечении систематического соблюдения предполагаемых моделью ограничений, качества входных данных и получаемого решения.

6. Создание режима благоприятствования. Этот шаг часто оказывается самым трудным — внедрение новаций нередко наталкивается на незаинтересованность и даже на сопротивление. Поэтому обучение персонала, реклама, качество сопроводительной документации и учет разнообразия поведенческих мотивов людей играют здесь решающую роль.

На схеме (рис.1) пунктирной линией отмечена та часть процесса принятия решения, где заметную роль играют различные соображения математического характера.

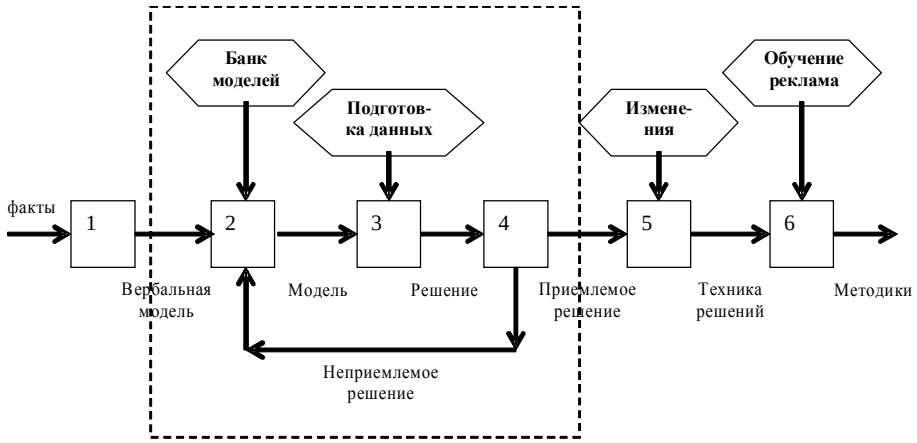

Рис. 1. Этапы принятия решений

Отметим, что саму цель операции можно понимать по-разному. Это и организация, в том числе и технологическая, той или иной осмысленной деятельности для достижения каких-либо целей (в качестве математического обеспечения здесь используются преимущественно детерминированные и стохастические модели), и изучение моделей и мотивов поведения взаимодействующих сторон (здесь применяются игровые модели).

В настоящее время к решению представляющих практический интерес сложных задач привлекаются большие коллективы людей (и, добавим, значительные вычислительные средства) с разной профессиональной подготовкой и ориентацией, с разной степенью осведомленности о задаче в целом и с разной степенью ответственности — от руководителя (ЛПР) до специалиста-разработчика (исследователя) и рядового исполнителя.

Для того чтобы подобный коллектив мог достаточно плодотворно функционировать, важно подготовить тех, кто был бы способен к действенному связыванию разных его блоков, кто осуществлял бы нетривиальные коммуникационные функции, был посредником как между ЛПР и специалистом-разработчиком, так и между разработчиком и исполнителем. Этому посреднику не обязательно знать в деталях всю техническую сторону вопроса (это задача для найденных при его посредстве специалистов), а достаточно ориентироваться в основных идеях. Иными словами, если касаться только математической части, у

него должны быть определенные представления о возможностях математических методов, об их идейных основаниях и о банке готовых математических моделей и ключевых методов.

Большинство идей используемых в моделировании допускает простое и вполне доступное объяснение, которое, разумеется, требует определенной математической культуры. Не следует также забывать о том, что для корректной постановки задачи моделирования, построения адекватной модели и интерпретации результатов моделирования необходимы глубокие знания в соответствующей предметной области. Аппарат, используемый при выработке и принятии решений с использованием моделей, должен быть настолько прост и нагляден, насколько это возможно. И насколько это возможно, он должен быть согласован с точностью и объемом доступной информации, а также — с вопросами, на которые требуется получить ответы с использованием моделей.

Необходимо отметить, что область, где методы математического моделирования работают достаточно эффективно, не совпадают с множеством всех актуальных прикладных задач. Последние часто слабо формализуемы и традиционно консервативны. Отсюда подозрительное отношение к рекомендациям, основанным на расчетах, требующих обширных и математических знаний.

Нужно признать, что определенные основания для подобных подозрений есть. Методы моделирования способны решать только те задачи, которые могут быть формулированы на языке математики. А это предполагает непременные упрощения в реальной сложной ситуации. Разделение же определяющих факторов задачи на существенные и второстепенные требует практического (а не математического) опыта и интуиции.

1.2. Классификация моделей

В качестве инструментов для решения научных и практических задач применяются весьма разнообразные модели.

Физическое моделирование — метод исследования объектов, при котором изучаемый объект (процесс, явление) воспроизводится с сохранением его физической природы или используется аналогичное ему другое физическое явление. Необходимыми условиями при этом являются: соблюдение геометрического подобия оригинала, модели или соответствующих масштабов для параметров исследуемого объекта. По характеру исследуемого объекта различают виды подобия, для

которых разработаны соответствующие виды критериев: электрические, гидравлические, аэродинамические и др.

Теоретическое (математическое) моделирование — метод исследования объектов, при котором изучаемый объект воспроизводится с помощью его формализованного математического описания (уравнений, алгоритмов и т.д.). Необходимым условием при этом является однозначная связь между параметрами объекта и его математического описания, то есть их аналогия.

В начале 60-х годов был разработан **квазианалоговый** метод моделирования, занимающий промежуточное положение по отношению к физическому и математическому моделированию. Этот метод состоит в экспериментальном изучении не самого исследуемого объекта, а объекта иной физической природы, который описывается математическими соотношениями эквивалентными относительно получаемых результатов (табл. 2).

Таблица 2

Пример квазианалогого метода моделирования

Механическая система	Аналогичная электрическая система	
	вариант 1	вариант 2
M – Масса X – Перемещение ν – Скорость $Q = m\frac{d\nu}{dt}$ – Сила $s = \frac{Q}{\nu}$ – Коэффициент скоростного трения	L – Индуктивность q – Заряд I – Ток $E = L\frac{dI}{dt}$ – ЭДС $R = \frac{E}{I}$ – Сопротивление	C – Емкость ψ – Потокосцепление U – Напряжение $I = C\frac{dU}{dt}$ – Ток $\gamma = \frac{I}{U}$ – Проводимость

Для того, чтобы разрабатывать, внедрять и использовать математические модели необходимо хорошо знать как математику, так и предметную область, т.е. объект моделирования. В определенном смысле в этом — недостаток математического моделирования. Достоинством математических моделей является относительная легкость их расширения и возможность использования в самых разных областях применения. Главным достоинством физических моделей является их наглядность, а главными недостатками — относительно высокая стоимость и очень низкая универсальность. Основными

достоинствами квазианалоговых моделей являются сравнительно невысокая стоимость, возможность наблюдения за экспериментом в реальном масштабе времени, а основной недостаток — в сравнительно высокой погрешности модели. Эти характеристики, главным образом, и определяют области применения физических моделей. Например: модель застройки микрорайона, предназначенная для проведения презентаций с целью привлечения инвесторов, должна быть физической (макет в масштабе), а модель, предназначенная для исследования и оптимизации транспортных потоков через тот же микрорайон, — математической, причем допускающей многократные испытания при различных условиях.

В рамках данного курса мы будем изучать только математические модели.

Существует множество математических моделей, описывающих различные ситуации, требующие принятия тех или иных решений. Выделим из них следующие три класса — детерминированные, стохастические ‘и игровые модели.

При разработке детерминированных моделей исходят из той предпосылки, что основные факторы, характеризующие ситуацию, вполне определенны и известны. Здесь обычно ставится задача оптимизации некоторой величины — критерия эффективности решений (например, минимизация затрат).

Стохастические модели применяются в тех случаях, когда некоторые факторы носят неопределенный, случайный характер.

Наконец, при учете наличия противников либо союзников с собственными интересами необходимо применение теоретико-игровых моделей.

В стохастических и игровых моделях ситуация с оценкой эффективности дополнительно и существенно усложняется. Зачастую выбор самого критерия зависит здесь от конкретной ситуации, и возможны различные критерии эффективности принимаемых решений.

В дальнейшем мы будем опираться на приведенную классификацию моделей, хотя возможны и другие, например, модели можно делить на статические и динамические, дискретные и непрерывные и т.д.

Степень сходства модели и объекта, который она отображает, характеризуется понятиями **изоморфизма** и **гомоморфизма**.

Модель изоморфна объекту, если можно установить взаимно однозначное соответствие между элементами модели и объекта, а также — между характеристиками взаимодействия элементов модели и элементами объекта.

Под гомоморфизмом понимается сходство по форме при различии основных структур. Гомоморфные модели являются результатом процессов упрощения и абстракции.

Большинство моделей скорее гомоморфны, чем изоморфны.

Модель, обычно, позволяет дать оценку (критерий) эффективности каждого принятого решения, если заданы само решение все необходимые входные параметры. Эту функцию модели иногда называют решением **прямой задачи**.

Гораздо чаще, однако, требуется решить **обратную задачу** — при заданных входных параметрах найти неизвестное решение, при котором достигается максимальная эффективность решения (например, минимум затрат).

При решении обратных задач широко используется аппарат и методы **математического программирования** — математический дисциплины, изучающей свойства и методы решения задач на экстремум линейных и нелинейных функций многих переменных, при наличии различного рода ограничений на переменные.

В современном математическом программировании, в зависимости от вида решаемых задач, выделяется несколько разделов: линейное, нелинейное, дискретное, динамическое, геометрическое, стохастическое программирование. Некоторые из них будут рассматриваться далее в настоящем рабочем учебнике.

При создании или разработке модели важно найти верный баланс между ее точностью и простотой. Построение и внедрение действующих моделей приходит с опытом и практикой, в процессе соотнесения конкретных ситуаций с математическим описанием наиболее существенных сторон рассматриваемого явления. Конечно, ни одна математическая модель не может охватить всех особенностей изучаемой проблемы. Поэтому создание математической модели, дающей описание цели, процесса и результатов проведения операции, всегда было, есть и, по-видимому, останется не только наукой, но и искусством.

1.3. Детерминированные модели

Как уже отмечалось, в детерминированных моделях обычно имеется некоторый критерий эффективности, который требуется оптимизировать за счет выбора управленческого решения.

Пусть имеется некоторый объект, функционированием которого можно управлять, выбирая подходящим способом соответствующие параметры управления. Предположим, что математическая модель

объекта построена и позволяет вычислить показатель эффективности W при любом принятом решении и для любой совокупности условий, в которых может функционировать объект управления.

Обозначим управляющие параметры (элементы решения), которые можно свободно менять в известных пределах как $x_1, x_2, \ldots, x_n$, а независимые от выбора решения условия функционирования объекта как $a_1, a_2, \ldots, a_m$.

Условия $a_1, a_2, \ldots, a_m$ могут быть заданы в самых различных формах. Для определенности здесь можно просто считать, что $a_1, a_2, \ldots, a_m$ — это набор числовых параметров, входящих совместно с $x_1, x_2, \ldots, x_n$ в какие–либо конечные или дифференциальные уравнения модели.

В детерминированных моделях независимые параметры $a_1, a_2, \ldots, a_m$ точно известны. Таким образом, все факторы, характеризующие условия функционирования объекта либо известны, либо устанавливаются по нашему выбору. Так как математическая модель построена, будем считать, что зависимость показателя (критерия) эффективности от всех вышеперечисленных параметров известна, и для любых $a_1, a_2, \ldots, a_m; x_1, x_2, \ldots, x_n$ можно найти соответствующее значение W.

Тогда задача выбора решения заключается в том, что требуется найти такую совокупность элементов решения $x_1, x_2, \ldots, x_n$, которая обращает в максимум показатель эффективности при любых заданных значениях $a_1, a_2, \ldots, a_m$:

$$W(a_1, a_2, \ldots, a_m; x_1, x_2, \ldots, x_n) \to max. \qquad (1)$$

Случай, когда W требуется обратить в минимум, сводится к предыдущему, если взять показатель $\overline{W} = -W$, и поэтому отдельно здесь не рассматривается.

Кроме того, практически всегда задаются дополнительные ограничения на допустимые пределы изменения элементов решения. В самом простом случае такие ограничения могут быть заданы в виде набора неравенств:

$$x_i^{min} \le x_i \le x_i^{max}, i = 1,2,\ldots,n.$$

Однако значительно чаще допустимые решения ограничиваются косвенно, путем задания некоторого числа функций $F_1,F_2,...,F_L$ элементов решения $x_1,x_2,...,x_n$ и независимых параметров $a_1,a_2,...,a_m$ и требования, чтобы на допустимых элементах решения эти функции принимали заданные значения:

$$\begin{aligned} &F_1(a_1,a_2,...;x_1,x_2,...)=0,\\ &F_2(a_1,a_2,...;x_1,x_2,...)=0,\\ &.....\\ &F_L(a_1,a_2,...;x_1,x_2,...)=0. \end{aligned} \qquad (2)$$

Таким образом, в детерминированном случае задача отыскания оптимального решения сводится к математической задаче отыскания экстремума функции (функционала) W. Эта задача может быть весьма сложной, особенно при многих переменных $a_1,a_2,...,a_m;x_1,x_2,...,x_n$.

Итак, общим для рассматриваемых детерминированных задач является то, что в них стоит проблема поиска наибольшего или наименьшего (оптимального) значения некоторой функции, отражающей цель функционирования системы, которую иначе называют также **целевой функцией** или **критерием качества** управления.

Универсальных математических методов нахождения оптимального значения функций любого вида при наличии произвольных ограничений не существует. Математическое исследование подобных нерешенных проблем привело в недавнем прошлом к появлению новых разделов математической науки — математического программирования и теории оптимального управления, которые сейчас являются основой теории и практики исследования операций.

Здесь, однако, математически неразрешимые проблемы рассматриваться не будут. Отметим также, что часто решения задачи (1)—(2) не существует из-за того, что задача поставлена некорректно или модель построена неверно; тогда анализ ситуации отсутствия решения позволяет устранить некорректные элементы постановки задачи и/или модели.

1.4. Многокритериальные модели

Далеко не всегда весь комплекс целей и задач, стоящих перед объектом моделирования, можно выразить в форме единственной целевой функции. Чаще эффективность решений может оцениваться

сразу по нескольким показателям $W_1, W_2, ..., W_q$. Причем одни из них желательно сделать больше, а другие — меньше. Пусть, например, рассматривается модель боевой операции. Тогда вероятность достижения поставленной задачи (W_1) желательно сделать максимальной, а потери (W_2) и материально-технические затраты (W_3) — минимальными.

Сразу заметим, что, в общем случае, таким образом поставленная задача решения не имеет. Пусть, например, заданы два критерия эффективности W_1, W_2 и задача формулируется как:

$$\begin{aligned} W_1(a_1, a_2, ..., a_m; x_1, x_2, ..., x_n) &\to max, \\ W_2(a_1, a_2, ..., a_m; x_1, x_2, ..., x_n) &\to max, \end{aligned} \qquad (3)$$

при известных значениях независимых параметров $a_1, a_2, ..., a_m$ и общих ограничениях на элементы решения (2), где функции $W_1(\cdot)$ и $W_2(\cdot)$ различны. Рассмотрим на плоскости (U,V) множество точек Ω с координатами $U = W_1(a_1, a_2, ..., a_m; x_1, x_2, ..., x_n)$, $V = W_2(a_1, a_2, ..., a_m; x_1, x_2, ..., x_n)$ соответствующее множеству всевозможных решений $x_1, x_2, ..., x_n$ (рис. 2).

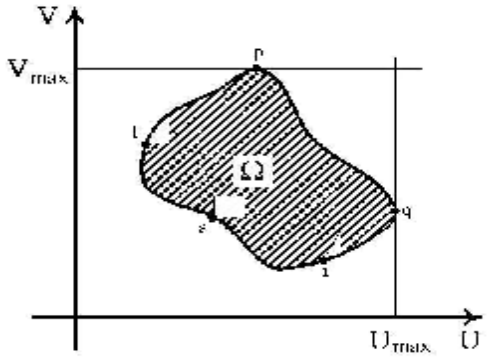

Рис. 2. Множество значений векторного критерия

Из рисунка видно, что наибольшее значение U, равное U_{max} и наибольшее значение V, равное V_{max}, достигаются в разных точках, а точка с координатами (U_{max}, V_{max}) лежит вне множества Ω.

Рассмотрим подробнее множество Ω. Пусть M — его произвольная точка, внутренняя или граничная. Поставим следующий вопрос: можно ли, оставаясь в множестве Ω, переместиться из точки M в некоторую другую, может быть — достаточно близкую, точку таким образом, чтобы увеличились обе ее координаты? Если M — внутренняя точка множества, то это, бесспорно, возможно[1]. Если же M — граничная точка, то такое возможно не всегда (рис.2). Из точек r,s,t это сделать можно, но из точек дуги pq уже нельзя: перемещение вдоль нее способно лишь увеличить одну из координат при одновременном уменьшении другой.

Точки дуги pq обладают тем свойством, что из них нельзя переместиться оставаясь в пределах Ω и, одновременно, увеличивая обе координаты — показатели эффективности решения. Множество точек, обладающих таким свойством называется **множеством (границей) Парето**. Отыскание и исследование множества Парето является одним из основных этапов решения многокритериальных задач.

Итак, многокритериальная задача (3), вообще говоря, неразрешима — удовлетворить обоим требованиям (3) одновременно невозможно. Следовательно, нужно искать компромиссное решение. Среди известных способов нахождения подобных решений стоит отметить два:

1) метод уступок;

2) метод идеальной точки.

Оба метода используют множество Парето задачи (3), состоящее из точек, которые нельзя улучшить сразу по всем критериям, оставаясь в пределах допустимого множества задачи: улучшая значения одного из критериев, мы неизбежно ухудшаем значения других.

Метод (последовательных) уступок заключается в том, что лицо, принимающее решение (ЛПР), работая в режиме диалога со специалистом, анализирует точки на границе Парето и в конце концов соглашается остановиться на некоторой компромиссной.

Метод идеальной точки состоит в отыскании на границе Парето точки, ближайшей к точке утопии, задаваемой ЛПР. Обычно ЛПР формулирует цель в виде желаемых значений показателей, а в качестве координат целевой точки выбирается сочетание наилучших значений

---

[1] Приведенные здесь рассуждения, конечно, нуждаются в более строгом обосновании. Как минимум, следовало бы ввести определения внутренней и граничной точки и осмыслить вытекающие из них следствия. Однако в данном рабочем учебнике эти математические вопросы не рассматриваются. Интересующиеся могут более глубоко познакомиться с ними в учебниках по современному математическому анализу.

всех критериев. Обычно эта целевая точка нереализуема при заданных ограничениях, поэтому ее и называют точкой утопии.

В качестве альтернативы комплексной оценке сразу по нескольким показателям на практике достаточно часто пытаются свести несколько показателей в один обобщенный (составной, комплексный) критерий эффективности. В дальнейшем этот обощенный показатель используется в качестве единственного критерия оптимальности при решении задачи.

Один из способов построения подобного обощенного показателя — составить дробь; в числителе которой стоят те частные показатели $W_1, W_2, ..., W_r$, $1 < r < q$, которые требуется увеличить, а в знаменателе, — те, которые требуется уменьшить:

$$U(a_1, a_2, ..., a_m; x_1, x_2, ..., x_n) = \frac{\prod_{i=1}^{r} W_i(a_1, a_2, ..., a_m; x_1, x_2, ..., x_n)}{\prod_{j=r+1}^{q} W_j(a_1, a_2, ..., a_m; x_1, x_2, ..., x_n)}. \quad (4)$$

Широко используется также составной критерий в виде взвешенной суммы частных критериев:

$$U(a_1, a_2, ..., a_m; x_1, x_2, ..., x_n) = \sum_{1}^{q} \omega_i W_i(a_1, a_2, ..., a_m; x_1, x_2, ..., x_n), \quad (5)$$

здесь ω_i — положительные или отрицательные коэффициенты. Положительные ставятся при тех показателях, которые требуется максимизировать; отрицательные – при тех, которые необходимо минимизировать, а абсолютные значения коэффициентов (веса) выбираются в соответствии со «степенью важности» показателей.

Общий недостаток «составных критериев» типа дроби (4) или взвешенной суммы (5) заключается в том, что снижение эффективности по одному показателю может компенсироваться за счет другого (например, малую вероятность выполнения боевой задачи – за счет малого расхода боеприпасов, и т. п.).

Следует отметить, что в случае когда частные критерии оптимальности W_i суть линейные функции компонент решения, существует тесная связь между решениями задачи на экстремум составного критерия типа (5) и множеством Парето исходной многокритериальной задачи. Можно доказать, что существуют такие весовые коэффициенты

ω_i, при которых решения задачи на максимум составного критерия (5) совпадают с точками множества Парето. При этом, однако, значения ω_i уже нельзя выбирать произвольно, но они могут быть вычислены по ходу решения задачи.

Задачу с несколькими показателями можно, разумеется, свести к однокритериальной, если выделить только один основной показатель эффективности W_1 и потребовать, чтобы он обратился в максимум, а на остальные показатели $W_2, W_3, \ldots$ наложить дополнительные ограничения типа:

$$W_2 \geq W_2^{min}, \ldots, W_r \geq W_r^{min}, W_{r+1} \leq W_{r+1}^{max}, \ldots, W_q \leq W_q^{max}. \quad (6)$$

После чего эти ограничения, разумеется, уже можно будет отнести к числу заданных условий $a_1, a_2, \ldots$. Например, при оптимизации плана работы промышленного предприятия можно потребовать, чтобы прибыль была максимальна, а себестоимость продукции – не выше заданной.

Полученные по результатам решения рекомендации, очевидно, будут зависеть от того, как выбраны ограничения (6) для вспомогательных показателей. Чтобы определить, в какой степени эти ограничения влияют на окончательные рекомендации по выбору решения, необходимо сравнить решения, получающиеся при различных ограничениях на вспомогательные показатели.

1.5. Поиск решений в условиях неопределенности

Очень многие реальные проблемы исследования операций не укладываются в рамки изложенной выше схемы детерминированной оптимизационной задачи с одним или несколькими критериями оптимальности. Основная причина — в том, что только часть условий функционирования объекта моделирования точно известны, а остальные содержат элемент неопределенности.

В подобных случаях эффективность зависит уже не от двух, а от трех категорий факторов:

- заранее известные условия $a_1, a_2, \ldots$, которые, однако, изменены быть не могут;
- неопределенные условия или факторы $\xi_1, \xi_2, \ldots$;
- элементы решения $x_1, x_2, \ldots$, которые необходимо выбрать.

Пусть, по–прежнему, эффективность характеризуется некоторым показателем W, зависящим теперь от всех трех групп факторов:

$$W(a_1,a_2,...;x_1,x_2,...;\xi_1,\xi_2,...). \quad (7)$$

Если бы условия $\xi_1,\xi_2,...$ были известны, можно было бы заранее подсчитать показатель W и выбрать такое решение $x_1,x_2,...$, при котором он достигает максимума. Однако, параметры $\xi_1,\xi_2,...$ неизвестны, а значит, неизвестно при каких условиях должна решаться задача на максимум зависящего от них показателя эффективности W. Иначе говоря, в отличие от детерминированного случая, выбор наиболее эффективных решений уже не сводится к решению некоторой оптимизационной задачи, даже если абсолютно точно известна «формула» для критерия эффективности.

Отметим, что **задача выбора решений в условиях неопределенности** (и не только в этом случае) стоит не перед модельером, а перед ЛПР. При этом основное назначение модели состоит не столько в том, чтобы найти какое-то ранее неизвестное решение, сколько в том, чтобы в интересах ЛПР получить оценки характеристик для ряда решений, на основании чего некоторые из них будут приняты (или не приняты) ЛПР. Само же принятие решений – это другая, не математическая задача.

Таким образом, появление неизвестных факторов $\xi_1,\xi_2,...$ переводит задачу, которая должна решаться методами моделирования в другую категорию – она превращается в задачу о выборе решения в условиях неопределенности.

Применяемые при этом методы существенно зависят от того, какова природа факторов $\xi_1,\xi_2,...$ и какими (неполными и ориентировочными) сведениями о них мы располагаем.

Наиболее простая в методологическом отношении и благоприятная для расчетов ситуация возникает, когда неизвестные факторы $\xi_1,\xi_2,...$ можно рассматривать как случайные величины (или случайные функции), относительно которых имеются необходимые статистические данные.

Пусть, например, мы рассматриваем работу железнодорожной сортировочной станции, стремясь оптимизировать процесс обслуживания прибывающих на эту станцию грузовых поездов. Заранее неизвестны ни точные моменты прибытия поездов, ни количество вагонов в каждом поезде, ни адреса, по которым направляются вагоны. Однако, все эти характеристики можно рассматривать как случайные величины, закон распределения которых можно оценить по имеющимся данным обычными методами математической статистики.

Если присутствующие в модели неопределенные факторы $\xi_1,\xi_2,...$ можно рассматривать как случайные величины или функции, распределение которых, хотя бы ориентировочно, известно, то схему поиска оптимального решения удается сохранить. При этом для оптимизации решения может быть применен один из двух приемов:

– **искусственное сведение к детерминированной схеме**;

– **оптимизация в среднем**.

Первый прием заключается в том, что неопределенная, вероятностная картина явления приближенно заменяется детерминированной. Для этого все участвующие в задаче случайные факторы $\xi_1,\xi_2,...$ приближенно заменяются неслучайными. Как правило, для замены в этом случае используются соответствующие математические ожидания, т.е. исходная задача на максимум (7) заменяется на задачу:

$$W = W(a_1,a_2,...,\mathbf{E}\xi_1,\mathbf{E}\xi_2,...,x_1,x_2,....) \to \max_{x_1,x_2,...},$$

обычно — при дополнительных ограничениях на элементы решения:

$$F_1(a_1,a_2,...,\mathbf{E}\xi_1,\mathbf{E}\xi_2,...,x_1,x_2,....) = 0,$$
$$F_2(a_1,a_2,...,\mathbf{E}\xi_1,\mathbf{E}\xi_2,...,x_1,x_2,....) = 0,$$
$$.....$$
$$F_m(a_1,a_2,...,\mathbf{E}\xi_1,\mathbf{E}\xi_2,...,x_1,x_2,....) = 0.$$

Здесь и в дальнейшем символ **E** будет использоваться для обозначения оператора математического ожидания.

Прием сведения к детерминированной схеме применяется по преимуществу в грубых, ориентировочных расчетах, когда диапазон случайных изменений величин $\xi_1,\xi_2,...$ сравнительно мал, т.е. они без большой натяжки могут рассматриваться как неслучайные. Заметим, что тот же прием замены случайных величин их математическими ожиданиями может успешно применяться и в случаях, когда величины $\xi_1,\xi_2,...$ обладают большим разбросом, но зависимость показателя эффективности W от них линейна или мало отличается от линейной.

Второй прием (оптимизация в среднем) применяется, когда существенно, что величины $\xi_1,\xi_2,...$ являются случайными и замена каждой из них ее математическим ожиданием может привести к большим ошибкам.

Рассмотрим оптимизацию в среднем более подробно. Пусть показатель эффективности W существенно зависит от случайных

факторов (будем для простоты считать их случайными величинами) $\xi_1,\xi_2,...$; допустим, что нам известно совместное распределение этих факторов, заданное, например, в виде плотности распределения $f(\xi_1,\xi_2,...)$. Предположим, что моделируемый объект реализует свою функцию много раз, причем условия $\xi_1,\xi_2,...$ меняются от раза к разу случайным образом. Какое решение $x_1,x_2,...$ следует выбрать? Рецепт метода оптимизации в среднем: следует выбрать то решение, при котором функционирование объекта в среднем будет наиболее эффективно, т.е. математическое ожидание показателя эффективности W будет максимально:

$$\overline{W} = \underset{\xi_1,\xi_2,...}{\mathbf{E}}[W] = \iiint W(a_1,a_2,...;\xi_1,\xi_2,...;x_1,x_1,...)\cdot f(\xi_1,\xi_2,...)\cdot d\xi_1 d\xi_2 ... \to \underset{x_1,x_2,...}{max} \quad (8)$$

при ограничениях:

$$\underset{\xi_1,\xi_2,...}{\mathbf{E}} F_1(a_1,a_2,...,\xi_1,\xi_2,...,x_1,x_2,....) = 0,$$
$$\underset{\xi_1,\xi_2,...}{\mathbf{E}} F_2(a_1,a_2,...,\xi_1,\xi_2,...,x_1,x_2,....) = 0,$$
$$.....$$
$$\underset{\xi_1,\xi_2,...}{\mathbf{E}} F_m(a_1,a_2,...,\xi_1,\xi_2,...,x_1,x_2,....) = 0.$$

Это и есть «оптимизация в среднем».

Элемент неопределенности в результате, в известном смысле, все же сохраняется. Дело в том, что эффективность функционирования объекта при случайных, заранее неизвестных значениях $\xi_1,\xi_2,...$ может сильно отличаться от ожидаемого среднего, как в большую, так и, к сожалению, в меньшую сторону. При многократной реализации объектом своей функции эти различия, в среднем, сглаживаются; однако, нередко данный способ оптимизации решения, за неимением лучшего, применяется и тогда, когда объект осуществляет свою функцию всего несколько раз или даже однажды. Тогда надо считаться с возможностью неприятных неожиданностей в каждом отдельном случае. Применяя «оптимизацию в среднем» к многочисленным (хотя бы и различным) объектам, все же мы в среднем выигрываем больше, чем если бы совсем не пользовались расчетом.

Для того, чтобы иметь представление о том, чем мы рискуем в каждом отдельном случае, желательно, кроме математического ожидания показателя эффективности, оценивать также и его дисперсию (или среднеквадратическое отклонение).

Наиболее сложным для исследования является тот случай неопределенности, когда неизвестные факторы $\xi_1,\xi_2,...$ не могут быть изучены и описаны методами математической статистики: их законы распределения или не могут быть получены из-за отсутствия соответствующих статистических данных, или вовсе не существуют, когда явление, о котором идет речь, не обладает свойством статистической устойчивости. Например, мы знаем, что на Марсе возможно наличие органической жизни, и некоторые ученые даже считают это весьма вероятным, но не существует каких-либо статистических данных, позволяющих подсчитать эту вероятность.

В подобных случаях, вместо произвольного и субъективного назначения вероятностей с дальнейшей «оптимизацией в среднем», рекомендуется рассмотреть весь диапазон возможных условий $\xi_1,\xi_2,...$ и составить представление о том, какова эффективность решений в этом диапазоне и как на нее влияют неизвестные условия. При этом задача приобретает новые методологические особенности.

Действительно, рассмотрим случай, когда эффективность W зависит, помимо заданных условий $a_1,a_2,...$ и элементов решения $x_1,x_2,...$, еще и от ряда факторов $\xi_1,\xi_2,...$, относительно которых никаких определенных сведений нет, а можно делать только предположения. Зафиксируем мысленно параметры $\xi_1,\xi_2,...$, придадим им вполне определенные значения $\xi_1 = y_1,\xi_2 = y_2,...$, и переведем их тем самым в категорию заданных условий $a_1,a_2,...$. Для этих условий мы в принципе можем решить задачу и найти соответствующее оптимальное решение $x_1^o,x_2^o,...$:

$$\left(x_1^o,x_2^o,...\right): W\left(a_1,a_2,...,y_1,y_2,...,x_1^o,x_2^o,...\right) = \max_{x_1,x_2,...} W\left(a_1,a_2,...,\xi_1 = y_1,\xi_2 = y_2,...,x_1,x_2,...\right).$$

Его элементы, кроме заданных условий $a_1,a_2,...$, очевидно, будут зависеть еще и от того, какие частные значения $y_1,y_2,...$ мы придали неопределенным условиям $\xi_1,\xi_2,...$, т.е.:

$$x_{11}^o = x_1^o\left(a_1,a_2,...;y_1,y_2,...\right).$$
$$x_2^o = x_2^o\left(a_1,a_2,...;y_1,y_2,...\right).$$

L L L L L

Это решение является **локально-оптимальным**, т.е. оно оптимально только для данной совокупности условий $y_1, y_2, \ldots$ и, как правило, для других значений $y_1, y_2, \ldots$ уже не оптимально. Совокупность локально-оптимальных решений для всего диапазона возможных условий $y_1, y_2, \ldots$ может дать представление о том, как следовало бы поступать, если бы неизвестные условия $y_1, y_2, \ldots$ были известны точно.

Отсюда можно сделать вывод, что локально-оптимальное решение, на получение которого зачастую тратится много усилий, имеет в случае неопределенности ограниченную ценность. Вместо того, чтобы после скрупулезных расчетов однозначно указать единственное, в точности оптимальное (в каком-то смысле) решение, при наличии неопределенности лучше выделить область приемлемых решений, которые оказываются несущественно хуже других, какой бы точкой зрения мы ни пользовались. Иначе говоря, предпочтительным является, скорее, не решение, строго оптимальное для каких-то определенных условий, а компромиссное решение, которое, не будучи строго оптимальным ни для каких условий, является приемлемым в целом диапазоне условий.

В настоящее время полноценной математической теории для такого рода компромисса еще не существует, хотя в **теории решений** и имеются определенные результаты в этом направлении. Обычно окончательный выбор компромиссного решения делает человек. Опираясь на расчеты, он может оценить и сопоставить сильные и слабые стороны каждого варианта решения в разных условиях и на основе этого сделать окончательный выбор. При этом необязательно (хотя иногда и любопытно) знать точный локальный оптимум для каждой совокупности условий $\xi_1, \xi_2, \ldots$.

Разработаны специальные математические методы, предназначенные для обоснования решений в условиях неопределенности. В некоторых, наиболее простых случаях эти методы дают возможность фактически найти и выбрать оптимальное решение. В более сложных случаях эти методы доставляют вспомогательный материал, позволяющий глубже разобраться в сложной ситуации и оценить каждое из возможных решений с различных (иногда противоречивых) точек зрения, взвесить его преимущества и недостатки и в конечном счете принять решение, если не единственно правильное, то, по крайней мере, до конца продуманное.

Необходимо учитывать, что при выборе решения в условиях неопределенности всегда неизбежен элемент произвола и, значит, риска. Недостаточность информации всегда опасна, и за нее приходится

платить. Однако в условиях сложной ситуации всегда полезно представить варианты решения и их возможные последствия в такой форме, чтобы сделать произвол выбора менее грубым, а риск – минимальным.

Разумеется, когда речь идет о неопределенной в каком-то смысле ситуации, рекомендации, вытекающие из научного исследования, не могут быть столь же четкими и однозначными, как в случаях полной определенности. Однако и при отсутствии полной определенности количественный анализ ситуации все же может принести пользу и помочь при выборе решения.

Наконец рассмотрим случай так называемых конфликтных ситуаций, когда неопределенные параметры $\xi_1,\xi_2,...$ зависят не от объективных обстоятельств, а от активно противодействующего противника. Такие ситуации характерны для боевых действий, отчасти для спортивных соревнований, для конкурентной борьбы и т.п.

Модели выбора решений в конфликтных ситуациях рассматриваются в **теории игр** – математической теории конфликтных ситуаций. Модели теории игр, основаны на предположении, что в конфликте мы имеем дело с разумным и дальновидным противником, всегда выбирающим свое поведение наихудшим для нас (и наилучшим для себя) образом. Такая идеализация конфликтной ситуации часто приводит к наименее рискованным, «перестраховочным» решениям, которое не обязательно принимать, но во всяком случае целесообразно иметь в виду.

2. КОНЦЕПТУАЛЬНЫЕ МОДЕЛИ И МАТЕМАТИЧЕСКИЕ СХЕМЫ МОДЕЛИРОВАНИЯ СИСТЕМ

2.1. Линейные модели

2.1.1. Задачи линейного программирования

Во многих случаях задачи принятия решений можно сформулировать таким образом, что показатель эффективности W представляет собой линейную функцию элементов решения $x_1,x_2,...,x_n$, а дополнительные ограничения, наложенные на элементы решения, можно представить в виде линейных равенств и/или неравенств. В этом случае модель называется линейной. Задача отыскания оптимального решения в результате сводится к математической задаче поиска экстремума линейной функции W при линейных ограничениях на

переменные. В математике такие задачи называются задачами линейного программирования.

2.1.2. Задача о пищевом рационе

Одной из первых задач линейного программирования является задача о пищевом рационе. Имеется четыре вида продуктов питания F_1, F_2, F_3, F_4. Известна цена каждого из продуктов c_1, c_2, c_3, c_4, а также содержание белков, жиров и углеводов в каждом из продуктов (табл.3).

Таблица 3

Продукт	Элемент		
	белки	углеводы	жиры
F_1	a_{11}	a_{12}	A_{13}
F_2	a_{21}	a_{22}	A_{23}
F_3	a_{31}	a_{32}	A_{33}
F_4	a_{41}	a_{42}	a_{43}

Из этих продуктов необходимо составить пищевой рацион, содержащий белков —не менее b_1 единиц, углеводов — не менее b_2 единиц, жиров — не менее b_3 единиц.

Требуется так составить пищевой рацион, чтобы обеспечить указанные условия при минимальной стоимости рациона.

Запишем условия задачи в виде формул. Пусть x_1, x_2, x_3, x_4 — количества продуктов F_1, F_2, F_3, F_4 в рационе. Тогда стоимость рациона будет:

$$W(x_1, x_2, x_3, x_4) = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4. \quad (9)$$

Согласно вышеприведенной таблице количество белков в рационе есть $a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4$ и по условию задачи должно быть:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 \geq b_1, \quad (10)$$

и, аналогично, для жиров и углеводов

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 \geq b_2, \quad (11)$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 \geq b_3, \quad (12)$$

Кроме того, для того чтобы решение имело смысл, его компоненты должны быть, очевидно, неотрицательны:

$$x_1, x_2, x_3, x_4 \geq 0. \qquad (13)$$

Таким образом, задача о пищевом рационе формулируется следующим образом: требуется найти такие неотрицательные значения переменных x_1, x_2, x_3, x_4, удовлетворяющие линейным неравенствам (10)—(13), при которых линейная функция этих переменных (9) обращается в минимум.

2.1.3. Задача о загрузке станков

Ткацкая фабрика располагает N_1 станками типа 1 и N_2 станками типа 2. Станки могут производить четыре вида тканей T_1, T_2, T_3, T_4. Каждый тип станка может производить любой из видов тканей, но с разной производительностью. Производительность станков при производстве каждого вида ткани даны в табл. 4.

Таблица 4

Тип станка	Вид ткани			
	T_1	T_2	T_3	T_4
1	a_{11}	a_{12}	a_{13}	a_{14}
2	a_{21}	a_{22}	a_{23}	a_{24}

Каждый метр ткани T_1 приносит фабрике доход c_1, ткани T_2 – доход c_2, ткани T_3 – доход c_3 и ткани T_4 – доход c_4.

Фабрике предписан план, согласно которому объемы производства тканей T_1, T_2, T_3, T_4 за месяц должны быть, соответственно, не менее b_1, b_2, b_3, b_4, т.е. плановое задание по объемам производства и ассортименту выражается числами b_1, b_2, b_3, b_4.

Требуется распределить загрузку станков производством тканей различного вида таким образом, чтобы план был выполнен и при этом месячная прибыль была максимальна.

Обозначим через x_{ij} — число станков типа i, занятых производством ткани T_j. Поскольку первый индекс соответствует типу станка,

а второй — виду ткани, постольку i = 1,2: j = 1,2,3,4. Вычислим прибыль W. Очевидно:

$$W = \sum_{j=1}^{4} c_j (x_{1j} + x_{2j}). \qquad (14)$$

Требуется выбрать такие неотрицательные значения переменных x_{ij}, чтобы линейная функция от них (14) обращалась в максимум.

При этом должны быть выполннены следующие ограничения:

ε - Суммарное число станков каждого типа, занятых производством всех видов тканей, не должна превышать наличного числа станков:

$$\sum_{j=1}^{4} x_{ij} \le N_i, \quad i = 1,2. \qquad (15)$$

- Плановые задания должны быть выполнены (или перевыполнены). С учетом данных по производительности станков эти условия запишутся в виде неравенств:

$$a_{1j}x_{1j} + a_{2j}x_{2j} \ge b_j, \quad j = 1,...,4. \qquad (16)$$

Итак, задача о загрузке станков формулируется следующим образом. Требуется выбрать такие неотрицательные значения переменных $x_{11}, x_{12}, \ldots, x_{24}$, удовлетворяющие линейным неравенствам (15) и (16), при которых линейная функция этих переменных (14) обращается в максимум.

2.1.4. Задача о распределении ресурсов

Имеется m видов ресурсов $R_1, R_2, \ldots, R_m$ (сырье, рабочая сила, оборудование и т.д.), которые можно использовать для производства n видов товаров $T_1, T_2, \ldots, T_n$.

Доступный объем каждого вида ресурсов ограничен. Ресурсные ограничения выражаются числами $b_1, b_2, .., b_m$.

Для производства единицы товара T_j, необходимо a_{ij} единиц ресурса R_i (i = 1,2,..., m; j = ..., n). Цена единицы ресурса R_i есть d_i, рублей (i = 1,2,…,m). Каждая единица товара T_j может быть реализована по цене c_j (j = 1,2, ..., n).

Объем производства каждого вида товара ограничивается спросом: известно, что на рынке нельзя реализовать более, чем k_j единиц товара T_j, (j = 1,2, ..., n).

Требуется определить план производства в ассортименте (какое количество единиц, каждого товара надо произвести), при котором прибыль максимальна. При этом должны быть учетены ограничения на спрос и объемы доступных ресурсов.

Запишем формально условия задачи. Пусть $x_1, x_2, \ldots, x_n$ плановые объемы производства товаров $T_1, T_2, \ldots, T_n$.

Тогда объемы ресурсов R_i, необходимые для реализации плана в ассортименте суть $\sum_{j=1}^{n} a_{ij} x_j$, $i = 1,2,\ldots,m$. Поэтому ресурсные ограничения запишутся как:

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = 1,2,\ldots,m. \tag{17}$$

Ограничения по спросу можно, очевидно представить как:

$$x_j \le k_j, \quad j = 1,2,\ldots,n. \tag{18}$$

Выразим прибыль W в зависимости от элементов решения. Себестоимость S_j единицы товара T_j, равна:

$$S_j = \sum_{i=1}^{m} a_{ij} d_i, \quad j = 1,2,\ldots,n.$$

Чистая прибыль q_i, получаемая от реализации одной единицы товара T_j, равна разнице между ее продажной ценой c_j и себестоимостью S_j:

$$q_j = c_j - S_j, \quad j = 1,2,\ldots,n.$$

Общая чистая прибыль от реализации всех товаров будет:

$$L = \sum_{j=1}^{n} q_j x_j. \tag{19}$$

Задача, таким образом, сводится к следующему: необходимо выбрать такие неотрицательные значения переменных $x_1, x_2, \ldots, x_n$, которые удовлетворяли бы линейным неравенствам (17), (18) и, одновременно, обращали бы в максимум линейную функцию этих переменных (19).

2.1.5. Задача о перевозках

Имеются m складов $C_1,C_2,...,C_m$ и n пунктов потребления $П_1,П_2,...,П_n$. Планируются перевозки товара со складов $C_1,C_2,...,C_m$ в пункты $П_1,П_2,...,П_n$. На складах $C_1,C_2,...,C_т$ имеются запасы товара в количестве $a_1,a_2,...,a_m$ единиц, а в пункты потребления $П_1,П_2,...,П_n$ — заявки соответственно на поставку $b_1,b_2...,b_n$ единиц товара. Заявки выполнимы, т. е. сумма всех заявок не превосходит суммы всех имеющихся запасов:

$$\sum_{j=1}^{n} b_j \le \sum_{i=1}^{m} a_i .$$

Склады $C_1,C_2,...,C_m$ связаны с пунктами потребления $П_1$ ..., $П_n$ транспортной сетью с заданными тарифами на перевозки: стоимость перевозки единицы товара со склада C_i в пункт $П_j$ равна c_{ij}, (i = 1, 2, ..., m; j = 1, 2, ..., n).

Требуется составить план перевозок так, чтобы все заявки были выполнены, а общие расходы на все перевозки были минимальны.

Пусть x_{ij} – число единиц товара, направляемое со склада C_i; в пункт $П_j$, (если с этого склада в этот пункт товары не направляются, $x_{ij} = 0$).

Решение (план перевозок) состоит из m×n чисел, которые можно записывать в виде прямоугольной таблицы, строки которой соответствуют складам, а столбцы — пунктам потребления[1]. Требуется выбрать такие неотрицательные значения переменных x_{ij} (i = l,2,...,m, j = 1,2, ..., n), чтобы:

– общее количество товара, взятое с каждого склада, не должно превышать имеющихся на нем запасов:

$$\sum_{j=1}^{n} x_{ij} \le a_i , \quad i = 1,2,...,m . \tag{20}$$

– Все поданные заявки были выполнены:

$$\sum_{i=1}^{m} x_{ij} = b_j , \quad j = 1,2,...,n . \tag{21}$$

---

[1] В математике такие таблицы называются матрицами и обозначаются следующим образом: таблица с элементами x_{ij} обозначается как $\|x_{ij}\|$. Теория матриц изучает свойства таблиц и операций над ними как математических объектов.

Если первозки осуществляются в объемах x_{ij}, то их общая стоимость L, очевидно, будет равна:

$$L = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}; \qquad (22)$$

ее необходимо свести к минимуму.

Таким образом, снова возникает задача, аналогичная рассмотренным ранее: необходимо выбрать неотрицательные значения переменных x_{ij} (i =1,2,...,m; j =1,2,...,n) так, чтобы линейная функция этих переменных (22) достигла минимума при выполнении условий (20),(21).

Особенность этой задачи, по сравнению с ранее рассмотренными, состоит в том, что не все ограничения, наложенные на переменные, являются линейными неравенствами — условия (21) записаны в виде линейных равенств. В дальнейшем будет рассматриваться метод, позволяющий переходить при решении задач линейного программирования от ограничений–неравенств к ограничениям–равенствам и обратно.

Заметим, что если с каждого склада будет вывезено все, что на нем имеется, то неравенства (20), так же как и (21), превратятся в равенства. В этом случае сумма всех заявок равна сумме всех запасов:

$$\sum_{j=1}^{n} b_j = \sum_{i=1}^{m} a_i .$$

В такой постановке задача о перевозках называется **транспортной задачей**. Она будет более подробно рассматриваться далее.

2.1.6. Задача о производстве сложного оборудования

Планируется производство сложного оборудования, каждый комплект которого состоит из n элементов или деталей $Э_1, Э_2, ..., Э_n$. Заказы на производство деталей могут быть размещены на m разных предприятиях $П_1$, $П_2$, ..., $П_m$.

Сдаче подлежат только полные комплекты оборудования, включающие все необходимые детали $Э_1$, $Э_2$, ..., $Э_n$.

В течение заданного времени T на предприятии $П_m$ можно изготовить a_{ij} деталей типа $Э_j$ (i = 1, ..., m; j = 1, ..., n).

Требуется распределить заказы по предприятиям так, чтобы число полных комплектов оборудования, изготовленных за время T, было

максимально. Планируя производство оборудования, необходимо для каждого предприятия $П_i$; указать, какую часть имеющегося в его распоряжении времени оно должно отдать на производство деталей $Э_j$ (j = 1, ..., m; i = 1, ..., n).

Обозначим x_{ij} долю времени Т, которую предприятие $П_i$ будет тратить на производство детали $Э_j$ (если эта деталь на данном предприятии вообще не производится, $x_{ij} = 0$).

Время, которое каждое предприятие тратит на производство всевозможных деталей не должно, очевидно, превышать продолжительность всего горизонта планирования Т. Так как x_{ij} есть доля общего времени Т, это ограничение можно представить как:

$$\sum_{j=1}^{n} x_{ij} \le 1, \quad i = 1,2,...,m. \tag{23}$$

Определим число полных комплектов оборудования, которое за время Т поставят все предприятия вместе. Общее количество деталей Эj, которое произведут все предприятия, будет равно:

$$N_j = \sum_{i=1}^{m} a_{ij} x_{ij}, \quad j = 1,2,...,n. \tag{24}$$

Таким образом, при заданном плане распределения заказов, т. е. при заданных x_{ij} (i = 1, ... , m; j = l, ... , n) будет произведено N_j экземпляров деталей $Э_j$.

Сколько же полных комплектов оборудования можно собрать из этих элементов? Очевидно, что полное число комплектов оборудования равно минимальному из чисел N_1, N_2, …, N_n. Действительно, если, например, элементов типа $Э_1$ произведено 100 шт., а элементов типа $Э_2$ – всего 10 шт., то мы никак не сможем собрать из этих элементов более 10 полных комплектов.

Обозначим Z – число полных комплектов оборудования, которое можно собрать при данном плане размещения заказов x_{ij}. Из предыдущего очевидно, что Z —минимальное из чисел N_j для всевозможных j, или подробнее, с учетом (25), это условие можно записать в виде:

$$Z = \min_{j} \sum_{i=1}^{m} a_{ij} x_{ij}. \tag{25}$$

В результате приходим к следующей постановке задачи: найти такие неотрицательные значения переменных x_{ij}, чтобы выполнялись

неравенства (24) и при этом обращалась в максимум функция этих переменных (25).

Так как функция (25) не является линейной по переменным x_{ij}, то на первый взгляд кажется, что рассматриваемая задача не является задачей линейного программирования. Однако, на самом деле, эту задачу легко свести к задаче линейного программирования. Действительно, поскольку величина Z является минимальной из всех величин (24), постольку должны выполняться неравенства

$$\sum_{i=1}^{m} a_{ij}x_{ij} \geq Z, \quad j = 1,2,...,n. \tag{26}$$

Рассмотрим теперь Z как новую неотрицательную переменную и поставим задачу: найти такие неотрицательные переменных $x_{11}, x_{12}, ..., x_{mn}$ и Z, чтобы они удовлетворяли линейным неравенствам (23) и (26) и при этом линейная функция:

$$L(x_{11}, x_{12}, ..., x_{mn}, Z) = 0 \cdot x_{11} + 0 \cdot x_{12} + ... + 0 \cdot x_{mn} + 1 \cdot Z;$$

обращалась в максимум.

Тем самым, путем введения дополнительной переменной Z, наша задача сведена к обычной задаче линейного программирования.

2.1.7. Геометрическая интерпретация решения задачи линейного программирования

Выше мы рассмотрели целый ряд задач исследования операций, которые имеют существенные общие черты: в каждой из них элементы решения представляют собой ряд неотрицательных переменных $x_1, x_2, \ldots$, а задача состоит в том, чтобы найти значения этих переменных, при которых заданная линейная целевая функция $x_1, x_2, \ldots$ обращается в максимум или минимум, и, кроме того, выполняются дополнительные ограничения относительно тех же переменных x_1, x_2, ..., заданные в виде линейных равенств или неравенств. Задачи этого типа называются **задачами линейного программирования** (ЗЛП).

Условимся, относительно терминологии, которая используется в дальнейшем и является общепринятой в линейном программировании.

Планом ЗЛП называется всякий набор переменных $x_1, x_2, ..., x_n$, который обычно рассматривается также как вектор линейного пространства R^n.

Допустимым планом называется такой план, который удовлетворяет всем ограничениям на переменные задачи, т.е. всем

имеющимся ограничениям—равенствам, ограничениям—неравенствам и условиям неотрицательности переменных.

Оптимальным планом $x_1^*, x_2^*, ..., x_n^*$ называется такой допустимый план, при котором целевая функция достигает желаемого оптимального (например— максимального) значения:

$$W^* = W\left(x_1^*, x_2^*, ..., x_n^*\right) = \max_{x_1, x_2, ..., x_n} W\left(x_1, x_2, ..., x_n\right).$$

Решением задачи называется набор $x_1^*, x_2^*, ..., x_n^*; W^*$, включающий оптимальный плана и оптимальное значение целевой функции; его нахождение является результатом вычислительного процесса решения задачи.

Рассмотрим более детально задачу ЛП с двумя переменными X_1, X_2:

$$W = c_1 X_1 + c_2 X_2 \to max, \quad c_1 = c_2 = 1; \tag{27}$$

при следующих ограничениях:

– условиях неотрицательности переменных:

$$X_1, X_2 \geq 0;$$

– дополнительных ограничениях — неравенствах:

$$X_1 + 2X_2 \leq 5, \tag{28}$$

$$3X_1 + X_2 \leq 8. \tag{29}$$

Так как число переменных равно двум, процесс решения ЗЛП можно проиллюстрировать графически на координатной плоскости $\{X_1, X_2\}$, (рис. 3).

Заменим, временно, в любом из ограничений—неравенств или условий неотрицательности знак неравенства на знак равенства. Получим уравнение прямой на плоскости $\{X_1, X_2\}$. Эта прямая делит всю плоскость на две полуплоскости. В одной полуплоскости данное неравенство выполняется, а в другой – нет. Иначе говоря, область значений наших двух переменных «допустимых» по каждому из взятых по отдельности ограничений геометрически представляет собой полуплоскость.

Следовательно, область или **множество допустимых решений** (допустимое множество) задачи, в котором все ограничения должны выполняться одновременно, представляет собой пересечение всех полуплоскостей, «допустимых» по каждому ограничению в отдельности.

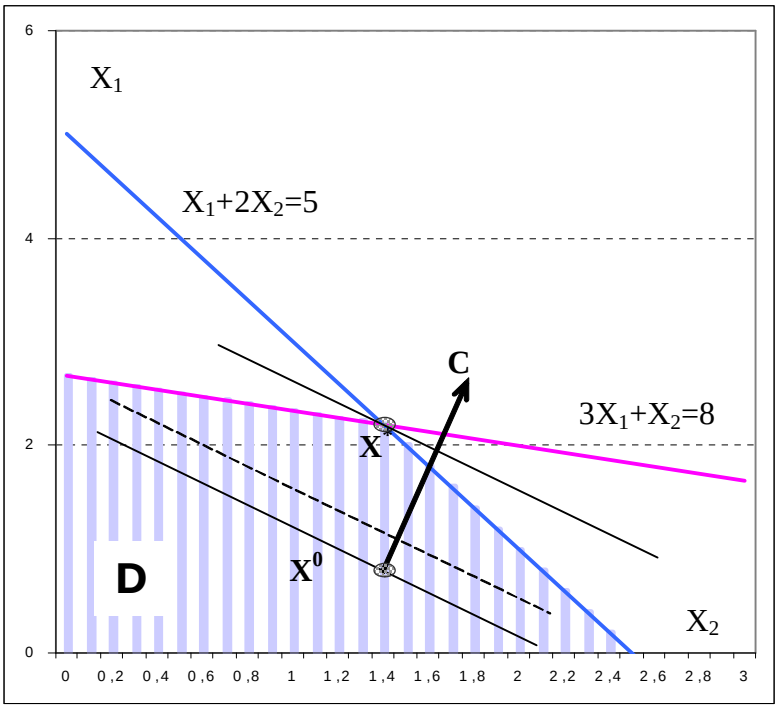

Рис. 3. Геометрическая интерпретация решения ЗЛП с двумя переменными

Итак, множество допустимых планов есть множество точек, которые одновременно принадлежат каждой из полуплоскостей, заданных ограничениями задачи. На рис. 3 это — заштрихованный четырехугольник **D** плоскости $\{X_1, X_2\}$. Четырехугольник **D** получается как пересечение первого квадранта (часть плоскости на которой выполняются ограничения неотрицательности) и двух полуплоскостей: полуплоскости под прямой $X_1 + 2X_2 = 5$, где выполняется неравенство (28) и полуплоскости под прямой $3X_1 + X_2 = 8$, где выполнено неравенство (29).

Наша цель теперь состоит в том, чтобы среди всех допустимых планов (точек множества D) найти план (точку), в которой значение целевой функции максимально.

Будем искать оптимальный план, отправляясь от некоторой точки допустимого множества. На плоскости $\{X_1, X_2\}$ возьмем некоторую точку $\mathbf{X}_0 = (X_{10}, X_{20})$ внутри допустимого множества. Значение целевой функции в точке $\mathbf{X}_0$, очевидно, будет $W_0 = c_1 X_{10} + c_2 X_{20}$.

Геометрическое место точек плоскости, в которых целевая функция (27) принимает то же самое значение W_0 (это множество называется **линией уровня** целевой функции), есть, очевидно, прямая с уравнением $c_1 X_1 + c_2 X_2 = W_0 = \text{const}$. Рассмотрим семейство линий уровня функции (27) с числовым параметром λ:

$$c_1 X_1 + c_2 X_2 = \lambda = const. \qquad (30)$$

Ясно, что это — параллельные прямые, и что заданное значение параметра λ равно значению целевой функции на соответствующей прямой.

Поэтому, с геометрической точки зрения, задачу максимизации целевой функции (27) можно рассматривать как задачу нахождения такой точки допустимого множества, через которую проходит линия уровня, соответствующая наибольшему значению λ.

Тогда процесс решения нашей задачи можно построить следующим образом:

1. Взять произвольную линию уровня, проходящую через какую-нибудь точку допустимого множества.

2. Перемещать линию уровня в направлении возрастания целевой функции (параметра λ) до тех пор, пока она имеет общие точки с допустимой областью.

Из математического анализа известно, что направление скорейшего возрастания целевой функции задается вектором ее **c** градиента с координатами:

$$\mathbf{c} = \left(\frac{\partial W}{\partial X_1}, \frac{\partial W}{\partial X_2} \right).$$

В данном случае целевая функция линейна, ее градиент — постоянный вектор с координатами $\mathbf{c} = (c_1, c_2)$, ортогональный линиям уровня.

Будем перемещать прямую (30) в направлении **c** (см. рис.3) до тех пор, пока она еще имеет общие точки с допустимым множеством. При этом значение целевой функции возрастает. В конце концов придем в точку $\mathbf{X}^*$, где значение целевой функции максимально. Дальше двигаться нельзя, потому что прямая (30) и четырехугольник **D** уже не будет иметь общих точек. Конечной точкой оказалась вершина четырехугольника **D**, в которой пересекаются прямые, соответствующие ограничениям (28),(29). В этой точке оба ограничения выполняются как равенства:

$$X_1 + 2X_2 = 5,$$
$$3X_1 + X_2 = 8,$$

откуда нетрудно вычислить координаты оптимального решения и значение целевой функции в точке оптимума:

$$X_1^* = 2.2, \quad X_2^* = 1.4, \quad W^* = 3.6.$$

Подведем итоги. В случае, изображенном на рисунке, множество D — ограниченный многоугольник в первом квадранте плоскости $\{X_1, X_2\}$, причем решение ЗЛП существует, достигается в вершине (угловой точке) допустимого множества и единственно.

Нетрудно, однако, представить и другие случаи. Они изображены на рис.4. Возможны четыре типа аномалий (рис.4).

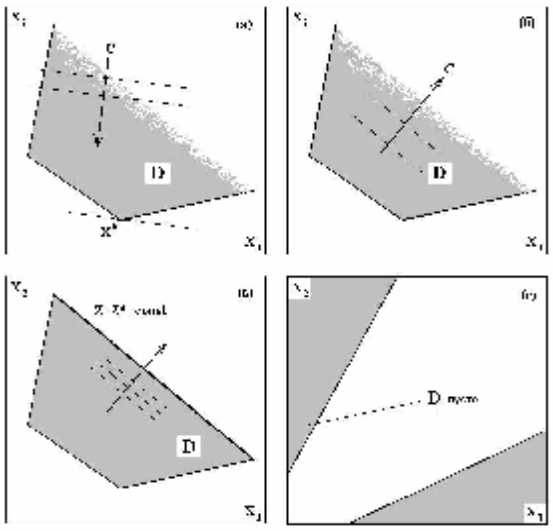

Рис. 4. Типовые особенности ЗЛП

а — допустимое множество неограниченно, но ограниченное и единственное решение задачи существует и достигается в угловой точке X^* множества D;

б — допустимое множество неограниченно и ограниченного решения не существует: сколько бы мы ни перемещались по линиям уровня в направлении вектора $\mathbf{c}$ целевая функция будет возрастать;

в — линия уровня, соответствующая максимальному значению целевой функции (параметра λ), касается одной из сторон многоугольника $\mathbf{D}$; соответственно, все точки этой стороны являются оптимальными планам, т.е. решение не единственно;

г — допустимое множество пусто, поскольку система ограничений—неравенств несовместна (полуплоскости не пересекаются) и допустимых планов не существует.

Таким образом, задача линейного программирования может иметь единственное решение, множество решений и ни одного решения. Последнее часто свидетельствует о том, что при постановке задачи упущены те или иные существенные факторы.

В общем виде, при $n > 2$, ЗЛП с ограничениями—неравенствами формулируется следующим образом. Требуется найти n—мерный вектор $X = (x_1, x_2, ..., x_n)$ с неотрицательными компонентами $x_i \geq 0, i = 1,2,...,n$, доставляющий максимум заданной линейной функции:

$$L(X) = \sum_{1}^{n} c_i x_i \rightarrow max;$$

(c_i — заданные коэффициенты) и, одновременно, удовлетворяющий системе линейных неравенств:

$$\sum_{1}^{n} a_{ij} x_i \leq b_j; \qquad (31)$$

где $j = 1,2,...,m$ — номер неравенства, a_{ij} — его числовые коэффициенты, и b_j — число в правой части неравенства.

При $n > 2$ геометрический способ решения ЗЛП практически бесполезен. Однако, как доказывается в теории линейного программирования, с увеличением числа переменных характер допустимого множества и решения задачи не меняется: каждому ограничению (31) соответствует гипервлоскость:

$$\sum_{1}^{n} a_{ij}x_i - b_j = 0, \quad j = 1,2,...,m;$$

которая делит все n—мерное пространство переменных на два полупространства, допустимое множество представляет собой выпуклый n—мерный многогранник, который образуется как пересечение указанных полупространств, а решение задачи, если оно существует, находится в одной из вершин многогранника.

2.1.8. Основная задача линейного программирования

В общем случае среди дополнительных (к условиям неотрицательности решения) ограничений ЗЛП могут быть как равенства так и неравенства. В связи с этим на практике, перед тем как решать ЗЛП, ее всегда приводят к какой-либо стандартной канонической форме. Мы сейчас рассмотрим такую каноническую форму ЗЛП, которая называется **основной задачей линейного программирования** (ОЗЛП), а затем покажем как перейти к ней от ЗЛП с ограничениями—неравенствами и обратно.

Основная задача линейного программирования ставится следующим образом.

Требуется найти неотрицательные значения переменных $x_1, x_2, ..., x_n$, которые удовлетворяли бы системе линейных уравнений:

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i = 1,2,...,m; \tag{32}$$

и, кроме того, обращали бы в минимум линейную функцию

$$L(x_1, x_2, ..., x_n) = \sum_{j=1}^{n} c_j x_j. \tag{33}$$

Случай, когда линейную функцию нужно обратить не в минимум, а в максимум, сводится к предыдущему, если изменить знак функции и рассмотреть вместо нее функцию $L'(x_1, x_2, ..., x_n) = -L(x_1, x_2, ..., x_n)$.

Допустимым решением или допустимым планом ОЗЛП называют любую совокупность неотрицательных переменных $x_1, x_2, ..., x_n$, удовлетворяющую уравнениям (32). Оптимальным решением (планом)

называется то из допустимых решений, при котором линейная функция (33) обращается в минимум.

Основная задача линейного программирования, как другие ЗЛП, необязательно должна иметь решение. Может оказаться, что уравнения (32) противоречат друг другу; может оказаться, что они имеют решение, но не в области неотрицательных значений переменных. Тогда ОЗЛП не имеет допустимых решений. Наконец, может оказаться, что допустимые решения ОЗЛП существуют, но среди них нет оптимального: функция L в области допустимых решений не ограничена снизу.

Покажем, как можно перейти от задачи с ограничениями-неравенствами к основной задаче линейного программирования.

Пусть имеется задача линейного программирования с n переменными $x_1,x_2,\ldots,x_n$, в которой наложенные на переменные ограничения имеют вид линейных неравенств. В некоторых из них знак неравенства может быть $\geq$, а других $\leq$. Так как второй вид сводится к первому переменой знака обеих частей неравенства, все ограничения-неравенства можно записать в стандартной форме:

$$\sum_{j=1}^{n} a_{ij}x_j + b_i \geq 0, \quad i = 1,2,\ldots,m. \tag{34}$$

Требуется найти такую совокупность неотрицательных значений $x_1,x_2,\ldots,x_n$, которая удовлетворяла бы неравенствам (34), и, кроме того, обращала бы в минимум линейную функцию:

$$L = \sum_{j=1}^{n} c_j x_j. \tag{35}$$

Введем обозначения:

$$y_i = \sum_{j=1}^{n} a_{ij}x_j + b_i, \quad i = 1,2,\ldots,m, \tag{36}$$

где $y_1,y_2,\ldots,y_m$ – некоторые новые переменные, которые мы будем называть «добавочными». Согласно условиям (34), эти добавочные переменные так же, как и $x_1,x_2,\ldots,x_n$, должны быть неотрицательными.

Таким образом, возникает задача в следующей постановке: найти такие неотрицательные значения n+m переменных $x_1,x_2,\ldots,x_n,y_1,y_2,\ldots,y_m$, чтобы они удовлетворяли системе уравнений (36) и одновременно обращали в минимум линейную функцию этих переменных (35).

Тем самым, задача линейного программирования с ограничениями-неравенствами сведена нами к основной задаче линейного программирования, но с большим числом переменных, чем было в исходной задаче. Общее число переменных теперь равно n+m, из них n «первоначальных» и m «добавочных». Функция L выражена только через «первоначальные» переменные (коэффициенты при «добавочных» переменных в ней равны нулю). Суть дела от этого, конечно, не меняется.

Обратно. Пусть дана ОЗЛП относительно n переменных $x_1,x_2,...,x_n$ и требуется минимизировать линейную целевую функцию (33), при условии, что переменные неотрицательны и удовлетворяют m уравнениям (32). Преобразуем эту задачу к задаче с ограничениями — неравенствами.

Любую систему линейных уравнений можно представить в виде некоторой совокупности линейных неравенств с помощью одного дополнительного ограничения. С целью пояснения этого утверждения заметим, что равенство $x=9$ эквивалентно комбинации неравенств $x\le 9$ и $x\ge 9$, которая в свою очередь может быть записана в виде пары неравенств $x\le 9$ и $-x\le -9$. Как нетрудно проверить графически, система уравнений $x=1,y=2$ эквивалентна комбинации неравенств $x\le 1,y\le 2,x+y\ge 3$, которую в свою очередь можно представить в виде $x\le 1,y\le 2,-x-y\le 3$. Изложенные соображения допускают следующее обобщение: систему уравнений:

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

можно записать в виде:

$$\sum_{j=1}^{n} a_{ij}x_j \le b_i,\quad i=1,2,...m,\quad \sum_{j=1}^{n} \alpha_j x_j \le \beta,$$

где

$$\alpha_j = -\sum_{i=1}^{m} a_{ij},\quad \beta = -\sum_{i=1}^{m} b_i.$$

Таким образом, от ОЗЛП относительно n переменных и m ограничений типа равенства мы перешли к ЗЛП относительно n переменных с m+1 ограничениями—неравенствами.

2.1.9. Симплекс-метод решения задач линейного программирования

Отметим, что задачи (32)—(33) и соответствующая ей ОЗЛП (35)—(36) получаются одна из другой с помощью элементарных алгебраических действий умножения и сложения линейных уравнений. То есть несмотря на внешнее различие этих задач, обе они представляют собой разные формулировки одной и той же проблемы. Поэтому обе они, одновременно, либо имеют, либо не имеют решения, и если оно существует, то значения соответствующих компонент решений обеих задач одинаковы.

Поэтому мы проиллюстрируем здесь знаменитый симплекс-метод решения задач линейного программирования[1] на примере ОЗЛП, при том понимании, что на самом деле этим методом может быть решена любая задача ЛП.

Симплекс-методом на современных персональных компьютерах эффективно решаются задачи ЛП с числом ограничений до порядка 10000[2].

Итак, пусть имеется m ограничений типа равенства (32) и требуется найти неотрицательные значения $x_1,x_2,\ldots,x_n$, при которых линейная функция (33) будет (для определенности) максимальна. Предположим, что число уравнений меньше числа переменных n, а решение существует и конечно[3]. В этом случае вычислительная процедура симплекс–метода будет выглядеть следующим образом:

1. Выберем m переменных, задающих допустимое пробное (называемое базисным) решение, такое что только эти переменные принимают положительные значения, а остальные переменные равны нулю. Исключим базисные переменные из выражения для целевой функции.

2. Проверим, нельзя ли за счет одной из переменных, приравненных вначале к нулю, улучшить (увеличить) значение целевой функции, придавая этой переменной отличные от нуля положительные значения. Если это возможно, перейдем к шагу 3. В противном случае прекратим вычисления.

---

[1] Первая программа для ЭВМ, реализующая симплекс-метод была разработана в 60-х годах прошлого века фирмой IBM и, через короткое время, стала стандартом де-факто при решении задач ЛП.

[2] В настоящее время в коммерческих программах наряду с симплекс-методом используются также новые методы, получившие название методов "внутренней точки". Эти методы, по сравнению с симплекс-методом, характеризуются на порядок большей эффективностью.

[3] Симплекс-алгоритм позволяет установить отсутствие решений по ходу вычислительной процедуры, поэтому допущение о наличии решения на самом деле несущественно и используется здесь только для иллюстрации.

3. Найдем предельное значение переменной, за счет которой можно улучшить значение целевой функции. Увеличение значения этой переменной допустимо до тех пор, пока одна из m переменных, вошедших в пробное решение, не обратится в нуль. Исключим из выражения для целевой функции только что упомянутую переменную и введем в пробное решение ту переменную, за счет которой результат может быть улучшен.

4. Разрешим систему m уравнений относительно переменных, вошедших в новое базисное решение. Исключим эти переменные из выражения для целевой функции. Вернемся к шагу 2.

В теории линейного программирования **доказано**, что предложенный алгоритм действительно приводит к оптимальному решению для любой модели линейного программирования, причем за конечное число шагов. Здесь мы проиллюстрируем его отдельные шаги в формульной записи.

Предположим, что $n > m$ и что ограничения—равенства (32) линейно независимы[1]. Если это не так, и среди ограничений есть некоторое число $k < m$ линейно зависимых от остальных $m-k$ ограничений, то линейно зависимые ограничения будут выполняться автоматически, как только выполняются линейно независимые. Следовательно, если среди наших n ограничений есть линейно зависимые, то они несущественны и их можно просто отбросить.

Итак, имеется m линейно—независимых ограничений типа равенства (32) и дано допустимое по ограничениям пробное решение, в котором некоторые m переменных строго больше нуля (эти переменные называются **базисными**), а остальные $n-m$ (называемые **внебазисными** или **свободными** переменными) равны нулю. Так как ограничения линейно—независимы, то, как известно из линейной алгебры, систему линейных уравнений (32) можно разрешить относительно базисных переменных, выразив их через свободные переменных. Пусть базисными переменными будут $x_1, x_2, ..., x_m$. Тогда вместо m уравнений (32) мы будет иметь тоже m уравнений, но записанных в другой форме:

---

[1] Требование линейной независимости ограничений используется здесь только для простоты иллюстрации. В общем случае линейной независимости ограничений не требуется.

$$
\begin{aligned}
x_1 &= \alpha_{m+1,1}x_{m+1} + \alpha_{m+2,1}x_{m+2} + \ldots + \alpha_{n,1}x_{m+1} + \beta_1,\\
x_2 &= \alpha_{m+1,2}x_{m+1} + \alpha_{m+2,2}x_{m+2} + \ldots + \alpha_{n,2}x_{m+1} + \beta_2,\\
&\mathbf{L}\ \mathbf{L}\\
x_m &= \alpha_{m+1,m}x_{m+1} + \alpha_{m+2,m}x_{m+2} + \ldots + \alpha_{n,m}x_{m+1} + \beta_m.
\end{aligned}
\qquad (37)
$$

Подставим уравнения (37) в (33) для того, чтобы исключить из целевой функции (33) базисные переменные. Получим следующее новое выражение для целевой функции:

$$
L = \sum_{j=m+1}^{n} \epsilon_j x_j + \lambda, \qquad (38)
$$

где $\epsilon_j, j = m+1,\ldots,n$ – преобразованные после подстановки (37) коэффициенты:

$$
\epsilon_j = c_j + \sum_{1}^{m} c_i \alpha_{j,i}, \quad j = m+1,\ldots,n,
$$

а λ – соответствующее пробному решению значение целевой функции (свободные переменные равны нулю),

$$
\lambda = \sum_{1}^{m} c_j \beta_j .
$$

Заметим, что, поскольку свободные переменные пробного решения равны нулю, постольку простое перечисление базисных переменных в силу (37) однозначно определяет их значения:

$$
\begin{aligned}
x_1 &= \beta_1,\\
x_2 &= \beta_2,\\
&\mathbf{L}\ \mathbf{L}\\
x_m &= \beta_m.
\end{aligned}
\qquad (39)
$$

Поставим теперь вопрос, нельзя ли улучшить (увеличить) значение целевой функции придавая положительные значения одной из свободных переменных? Предположим, что среди преобразованных коэффициентов ϵ_j в (38) есть положительные. Тогда это сделать можно, так как значение (38) будет увеличиваться, как только увеличится значение соответствующей свободной переменной. Однако неограниченно увеличивать значение выбранной свободной

переменной нельзя, так как при этом должны, в соответствии с ограничениями (32), изменяться и базисные переменные. Для определения допустимых пределов роста свободной переменной удобно воспользоваться системой уравнений (37), которая есть просто другая форма записи ограничений (32). Пусть, например, увеличивается свободная переменная номера k, $m+1 \le k \le n$. Тогда, согласно (37), базисные переменные должны меняться как:

$$x_1 = \alpha_{k,1} x_k + \beta_1,$$
$$x_2 = \alpha_{k,2} x_k + \beta_2,$$
$$\mathbf{L}\ \mathbf{L}$$
$$x_m = \alpha_{k,m} x_k + \beta_m.$$

И если среди $\alpha_{k,i}$ есть отрицательные, то наибольшее возможное значение k-той свободной переменной из условия равенства нулю какой-либо i-той базисной переменной, очевидно, будет:

$$x_k^{max} = \min_{1 \le i \le m} \left| \frac{\beta_i}{\alpha_{k,i}} \right|,$$

где минимум берется по тем значениям индекса i, для которых $\alpha_{k,i}$ отрицательны.

В результате будем иметь увеличенное значение целевой функции $\lambda + \epsilon_k x_k^{max}$ и новый набор базисных переменных, из которого исключена бывшая i-тая базисная переменная (она стала равна нулю в результате увеличения k-той свободной) и добавлена k-тая свободная переменной, которая стала положительной.

Таким образом, мы получили новое допустимое пробное решение, в котором имеется m положительных переменных, остальные переменные равны нулю, а целевая функция увеличилась. В соответствии с симплекс-методом (см. выше), теперь надо повторить вышеописанные вычисления для новых базисных переменных и т.д.

Вычислительная процедура симплекс-метода легко интерпретируется геометрически в пространстве решений. При этом каждый пробный базис соответствует вершине выпуклого полиэдрального множества допустимых решений (см. геометрическую интерпретацию решения задачи ЛП выше в подразделе 2.1.7). Переход от одного базиса к другому геометрически выглядит как переход от одной угловой точки к другой (причем смежной) угловой точке. Таким образом, можно утверждать, что поиск оптимального решения симплексным методом

заключается в последовательном восхождении вдоль ребер упомянутого многогранника от одной его вершины к соседней.

Строгое доказательство того, что полученное итеративным симплекс-методом решение действительно является оптимальным см. в (математических) учебниках по линейному программированию. Оно длинно, но в принципе несложно и базируется на том основном факте, что соотношения (37) получаются из исходных уравнений (32) с помощью простых алгебраических преобразований и поэтому, системы уравнений (32) и (37), несмотря на внешнее различие, представляют собой математическую формулировку одной и той же проблемы.

Хорошо известно, также что симплекс-метод позволяет построить допустимое пробное решение или — убедиться в невозможности его построения из-за несовместности системы уравнений (32).

Таким образом, справедлива следующая важная теорема о базисе.

Теорема о базисе. Если задача линейного программирования имеет конечное (ограниченное) оптимальное решение, то для нее существует базисное оптимальное решение.

При этом для линейной задачи с m ограничениями базисное решение представляет собой набор m переменных с однозначно определенными положительными значениями, удовлетворяющими всем ограничениям задачи, когда значения всех остальных переменных равны нулю.

Чтобы убедиться в том, что данная теорема действительно играет исключительно важную роль, рассмотрим следующую ситуацию. Предположим, что требуется найти решение задачи линейного программирования с 50-ю линейно независимыми ограничениями и 300-ми неизвестными. На основании теоремы о базисе можно утверждать, что в данном случае существует оптимальное решение, содержащее не более чем 50 переменных с положительными значениями. Введение в рассмотрение других переменных может улучшить оптимальное значение целевой функции, однако число переменных, требуемых для получения оптимального решения, не должно при этом превышать 50. Обратим внимание на то, что увеличение числа линейно независимых ограничений в той или иной задаче приводит к расширению базиса. Следовательно, количество переменных, входящих в оптимальное решение, тем больше, чем больше ограничений содержит модель.

2.1.10. Транспортная задача линейного программирования

В предыдущих разделах рабочего учебника были рассмотрены некоторые общие подходы к решению задач линейного программирования. Известны частные типы ЗЛП, которые, ввиду особенно простой структуры, допускают решение более простыми методами. В качестве одного из таких примеров рассмотрим решение т.н. **транспортной задачи**. Она ставится следующим образом.

Имеются m пунктов отправления (ПО) $A_1, A_2, ..., A_m$, в которых имеются запасы какого-то однородного товара (груза) в количестве соответственно $a_1, a_2, ..., a_m$ единиц. Кроме того, имеются n пунктов назначения (ПН) $B_1, B_2, ..., B_n$, подавших заявки соответственно на $b_1, b_2, ..., b_n$ единиц товара. Предполагается, что сумма всех заявок равна сумме всех запасов:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \ . \qquad (40)$$

Известна цена c_{ij} перевозки единицы товара от каждого пункта отправления A_i, до каждого пункта назначения B_j. Все числа в таблице (матрице) стоимостей перевозки $\|c_{ij}\|, \ i=1,...,m; \ j=1,...,n$ заданы. Считается, что стоимость перевозки нескольких единиц товара пропорциональна их числу и цене.

Требуется составить такой план перевозок (откуда, куда и сколько единиц везти), при котором все заявки были бы выполнены, а общая стоимость всех перевозок была бы минимальна.

Дадим этой задаче математическую формулировку. Обозначим x_{ij}, – количество груза, отправляемого из пункта отправления A_i в пункт назначения B_j (i=l,2,...,m; j=1,2, ...,n). Неотрицательные переменные x_{11}, x_{12}, ..., x_{mn} (число которых, очевидно, равно $m \times n$) должны удовлетворять следующим условиям:

– суммарное количество груза, направляемое из каждого пункта отправления во все пункты назначения, должно быть равно запасу груза в данном пункте (m условий—равенств):

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1,2,...,m,$$

– суммарное количество груза, доставляемое в каждый пункт назначения изо всех пунктов отправления, должно быть равно заявке данного пункта (n условий—равенств):

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,2,...,n,$$

– суммарная стоимость всех перевозок должна быть минимальной:

$$L = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to min.$$

2.1.11. Решение транспортной задачи

Простота формулировки транспортной задачи позволяет применить для ее решения упрощенный вариант симплексного метода, который рассматривается ниже.

Рассмотрим транспортную задачу для 4-х пунктов отправления A_1, A_2, A_3, A_4 с запасами $a_1 = 30, a_2 = 48, a_3 = 20, a_4 = 30$ и 5-ти пунктов назначения B_1, B_2, B_3, B_4, B_5 с заявками $b_1 = 18, b_2 = 27, b_3 = 42, b_4 = 15, b_5 = 26$. Представим исходные данные в виде стандартной транспортной таблицы (табл. 5).

Таблица 5

О ПН ПО **О**	B_1	B_2	B_3	B_4	B_5	Запасы
A_1	13	7	14	7	5	30
A_2	11	8	12	6	8	48
A_3	6	10	10	8	11	20
A_4	14	8	10	10	15	30
Заявки	18	27	42	15	26	128

В правом верхнем углу каждой клетки транспортной таблицы указана стоимость перевозок c_{ij}, $i = 1,...,4; j = 1,...,5$ между соответствующими пунктами отправления и назначения. Требуется найти (и

заполнить клетки) неотрицательные объемы перевозок x_{ij}, из пункта отправления A_i в пункт назначения B_j (i=l,...,4; j=1,...,5) при которых общая стоимость перевозок $L = \sum_{ij} c_{ij} x_{ij}$ будет минимальна.

Начнем заполнение транспортной таблицы с левого верхнего («северо-западного») угла. Пункт B_1 подал заявку на 18 единиц груза; удовлетворим ее из запасов пункта A_1. После этого в нем остается еще 30 — 18 = 12 единиц груза; отдадим их пункту B_2. Но заявка этого пункта еще не удовлетворена; выделим недостающие 15 единиц из запасов пункта A_2 и т. д. Рассуждая точно таким же образом, заполним до конца перевозками x_{ij} транспортную таблицу (таблица 6). Проверим, является ли этот план допустимым. Поскольку сумма перевозок по строке равна запасу соответствующего пункта отправления, а сумма перевозок по столбцу — заявке соответствующего пункта назначения; постольку все заявки удовлетворены и все запасы израсходованы (сумма запасов равна сумме заявок и выражается числом 128, стоящим в правом нижнем углу табл. 6). Следовательно, план допустимый.

Таблица 6

О ПН ПО **О**	B_1	B_2	B_3	B_4	B_5	Запасы
A_1	13 18	7 12	14	7	5	30
A_2	И	8 15	12 33	6	8	48
A_3	6	10	10 9	8 11	11	20
A_4	14	8	10	10 4	15 26	30
Заявки	18	27	42	15	26	128

Здесь и в дальнейшем в таблице проставляются только отличные от нуля перевозки, а клетки, соответствующие нулевым перевозкам, остаются «свободными».

Теперь необходимо сформулировать алгоритм оптимизации плана. Непосредственно из таблицы видно, что опорный план табл. 6 можно

улучшить, если произвести в нем «циклическую перестановку» перевозок между клетками таблицы, уменьшив перевозки в «дорогой» клетке (2.3) со стоимостью 12 и, соответствующим образом увеличив перевозки в «дешевой» клетке (2.4) со стоимостью 6 (таблица 7).

Таблица 7

О ПН ПО О	B_1	B_2	B_3	B_4	B_5	Запасы
A_1	13 18	7 12	14	7	5	30
A_2	И	8 15	12 33	6	8	48
A_3	6	10	10 9	8 11	11	20
A_4	14	8	10	10 4	15 26	30
Заявки	18	27	42	15	26	128

Чтобы план оставался допустимым, мы должны при этом сделать одну из свободных клеток базисной, а одну из базисных — свободной. Сколько единиц груза можем мы перенести по циклу (2.4) → (3.4) → (3.3) → (2.3), увеличивая перевозки в нечетных вершинах цикла и уменьшая — в четных? Очевидно, не больше, чем 11 единиц (иначе перевозки в клетке (3.4) стали бы отрицательными). В результате циклического переноса допустимый план остается допустимым — баланс запасов и заявок не нарушается. Произведем перенос и запишем новый, улучшенный план в таблице 8.

Общая стоимость опорного плана, показанного в таблице 8, равна:

$$L_{оп} = 18\times13+12\times7+15\times8+33\times12+9\times10+11\times8+4\times10+26\times15 = 1442;$$

общая стоимость нового плана, показанного в таблице 8, равна:

$$L_2 = 18\times13+12\times7+15\times8+22\times12+11\times6+20\times10+4\times10+26\times15 = 1398.$$

Таблица 8

О ПН ПО **О**	B_1	B_2	B_3	B_4	B_5	Запасы
A_1	13 18	7 12	14	7	5	30
A_2	И	8 15	12 22	6 11	8	48
A_3	6	10	10 20	8 0	11	20
A_4	14	8	10	10 4	15 26	30
Заявки	18	27	42	15	26	128

Таким образом, удалось уменьшить стоимость перевозок на 1442–1398 = 44 единицы. Это, впрочем, можно было предсказать и не подсчитывая полную стоимость плана. Действительно, алгебраическая сумма стоимостей, стоящих в вершинах цикла, со знаком плюс, если перевозки в этой вершине увеличиваются, и со знаком минус — если уменьшаются (так называемая «цена цикла»), в данном случае равна: 6–8+10–12 = –4. Значит, при переносе одной единицы груза по этому циклу стоимость перевозок уменьшается на четыре. А мы перенесли 11 единиц; значит, стоимость должна была уменьшиться на 44 единицы, что и произошло.

В теории линейного программирования доказывается, что при опорном плане для каждой свободной клетки транспортной таблицы существует цикл, и притом единственный, одна вершина которого (первая) лежит в данной свободной клетке, а остальные — в базисных клетках. Поэтому, отыскивая «выгодные» циклы с отрицательной ценой, мы должны проверять — нет ли в таблице «дешевых» свободных клеток. Если такая клетка есть, нужно для нее найти цикл, вычислить его цену и, если она будет отрицательной, перенести по этому циклу столько единиц груза, сколько будет возможно (без того, чтобы какие-то перевозки сделать отрицательными). При этом данная свободная клетка становится базисной, а какая-то из бывших базисных — свободной.

Попробуем еще раз улучшить план, приведенный в таблице 8. Нельзя ли улучшить план, увеличив перевозки «дешевой» клетка (1.5) со стоимостью 5, уменьшив в других (при этом, конечно, придется кое-где перевозки тоже увеличить)? Рассмотрим внимательно таблицу 8 и найдем в ней цикл, первая вершина которого лежит в свободной клетке (1.5), а остальные — все в базисных клетках. Это последовательность клеток (1.5) → (4.5) → (4.4) → (2.4) → (2.2) → (1.2) (и замыкая цикл, снова возвращаемся в (1.5)). Нечетные вершины цикла отмечены плюсом — это значит, что перевозки в этих клетках увеличиваются: четные — знаком минус (перевозки уменьшаются). Цикл показан стрелками в таблице 9.

Таблица 9

О ПН ПО О	B_1	B_2	B_3	B_4	B_5	Запасы
A_1	13 18	7 12	14	7	5	30
A_2	И	8 15	12 22	6 11	8	48
A_3	6	10	10 20	8 0	11	20
A_4	14	8	10	10 4	15 26	30
Заявки	18	27	42	15	26	128

Подсчитаем цену этого цикла. Она равна 5–15+10–6+8–7 = –5. Так как цена цикла отрицательна, то переброска перевозок по этому циклу выгодна. Число единиц груза, которые можно перебросить по этому циклу, определяется наименьшей перевозкой, стоящей в отрицательной вершине цикла. В данном случае это — 11. Умножая 11 на цену цикла — 5 получим, что за счет переброски 11 единиц груза по данному циклу мы уменьшим стоимость перевозок на 55 единиц.

Таким образом, разыскивая в транспортной таблице свободные клетки с отрицательной ценой цикла и перебрасывая по этому циклу

наибольшее возможное количество груза, мы будем последовательно уменьшать стоимость перевозок.

Но стоимость перевозок, очевидно, ограничена снизу (нулем), и, следовательно, рано или поздно оптимальный план будет найден. Для такого плана уже не остается ни одной свободной клетки с отрицательной ценой цикла. Это — признак того, что решение найдено.

В теории линейного программирования предлагаются способы «автоматического» выделения свободных клеток с отрицательной ценой цикла (например, т.н. «метод потенциалов»), с которыми можно ознакомиться по специальным руководствам.

В заключение — рассмотрим несколько вариантов транспортной задачи, в которых она может быть поставлена на практике.

2.1.12. Транспортная задача с неправильным балансом

До сих пор мы рассматривали транспортную задачу, в которой сумма запасов равна сумме заявок. Это – классическая задача, иначе называемая «транспортной задачей с правильным балансом».

Существуют также варианты транспортной задачи, в которых условие баланса заявок и запасов (40) нарушено. В подобных случаях говорят о ТЗ с **неправильным балансом**.

Баланс ТЗ может нарушаться в двух направлениях: либо сумма запасов в пунктах отправления превышает сумму заявок, либо сумма заявок превышает наличные запасы. Первый случай называется «Транспортной задачей с избытком запасов», второй – «Транспортной задачей с избытком заявок». Рассмотрим эти случаи последовательно.

ТЗ с избытком запасов. В пунктах A_1, A_2, ..., A_m имеются запасы груза a_1, a_2, ..., a_m, пункты B_1, B_2, ..., B_n подали заявки b_1, b_2, ..., b_n, причем $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$. Требуется найти такой план перевозок x_{ij}, при котором все заявки будут выполнены, а общая стоимость перевозок минимальна, т.е.:

$$L = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to min.$$

Очевидно, при этой постановке задачи некоторые условия-равенства ТЗ превращаются в условия-неравенства, а некоторые – остаются равенствами.

Мы умеем решать задачу линейного программирования, в какой бы форме – равенств или неравенств – ни были заданы ее условия. Поэтому поставленная задача может быть решена, например, симплекс–методом. Однако, задачу можно решить проще, если искусственным приемом свести ее к ранее рассмотренной транспортной задаче с правильным балансом.

Для этого, сверх имеющихся n пунктов назначения B_1, B_2, ... , B_n введем еще один, фиктивный, пункт назначения B_{n+1} которому припишем фиктивную заявку, равную избытку запасов над заявками:

$$b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j , \qquad (41)$$

и положим стоимости перевозок из всех ПО в фиктивный ПН B_{n+1} равными нулю, т.е. $c_{in+1} = 0$ для всех i=1,2,…,m.

Таким образом, отправление какого-то количества груза x_{in+1} из пункта A_i, в пункт B_{n+1} просто будет означать, что в пункте A_i остались неотправленными x_{in+1} единиц груза. Введением фиктивного ПН B_{n+1}, с его заявкой b_{n+1} мы сравняли баланс ТЗ, и теперь ее можно решать как обычную ТЗ с правильным балансом.

ТЗ с избытком заявок. В пунктах A_1,A_2,...,A_m имеются запасы груза a_1,...,a_m пункты B_1,...,B_n подали заявки b_1,...,b_n причем $\sum_{j=1}^{n} b_j > \sum_{i=1}^{m} a_i$, т.е. имеющихся запасов недостаточно для удовлетворения всех заявок. Требуется составить такой план перевозок, при котором все запасы окажутся вывезенными, а стоимость перевозок – минимальной. Очевидно, эту задачу также можно свести к обычной ТЗ с правильным балансом, если ввести в рассмотрение фиктивный пункт отправления A_{m+1} с запасом a_{m+1}, равным недостающему запасу:

$$a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i ,$$

и положить стоимости перевозок из ПО A_{m+1}, в любой ПН равными нулю, т.е. $c_{m+1j} = 0$ для всех j=1,2,...,n. При этом какая-то часть заявок на каждом пункте останется неудовлетворенной; будем считать, что она покрывается за счет фиктивного ПО A_{m+1}.

Таким образом, мы свели транспортную задачу с избытком заявок к задаче с правильным балансом. Заметим, что при этом мы вовсе не заботились о «справедливости» удовлетворения заявок, не налагая никаких условий на то, какую долю своей заявки должен получить каждый ПН — нас интересовали лишь расходы, которые нужно минимизировать.

Если поставить задачу по-иному, например, потребовать, чтобы все ПН были удовлетворены в равной доле, задача снова сводится к транспортной задаче с правильным балансом. А именно, нужно поданные заявки «исправить», умножив каждую из них на коэффициент $k = \sum_{1}^{m} a_i \Big/ \sum_{1}^{n} b_j$, после чего решать ТЗ с правильным балансом.

2.1.13. Транспортная задача по критерию времени

До сих пор критерием оптимальности решения транспортной задачи была общая стоимость перевозок, и требовалось эту стоимость минимизировать.

В большинстве случаев именно критерий стоимости является основным, определяющим эффективность плана перевозок. Однако в некоторых случаях на первый план выдвигается не стоимость перевозок L, а время T, в течение которого все перевозки будут закончены. Так, например, бывает, когда речь идет о перевозках скоропортящихся продуктов или же — о подвозе боеприпасов к месту боевых действий. Наилучшим планом перевозок (x_{ij}) будет считаться тот план, при котором время окончания всех перевозок T минимально:

$$T \to min.$$

Транспортная задача, где оптимальным считается план с минимальным временем перевозок, называется транспортной задачей по критерию времени. Она ставится следующим образом.

Имеется m пунктов отправления $A_1, A_2 ..., A_m$ с запасами a_1, ..., a_m и n пунктов назначения B_1, ..., B_n с заявками b_1, ..., b_n ; сумма запасов равна сумме заявок:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j .$$

Заданы времена перевозок t_{ij} из каждого ПО A_i в каждый ПН B_j, предполагается, что они не зависят от количества перевозимого груза x_{ij}, т. е. количество транспортных средств всегда достаточно для осуществления любого объема перевозок.

Требуется выбрать перевозки $x_{ij} \geq 0$ таким образом, чтобы удовлетворялись балансовые условия:

$$\sum_{j=1}^{n} x_{ij} = a_i \quad i = 1,2,...,m,$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad j = 1,2,...,n.$$

и, кроме того, обращалось в минимум время окончания всех перевозок T.

Выразим время T через времена t_{ij} и перевозки x_{ij}. Так как все перевозки заканчиваются в момент, когда кончается самая длительная из них, то время T есть максимальное из всех времен t_{ij}, стоящих, в ячейках с ненулевми перевозками. Запишем это в виде формулы:

$$T = \max_{x_{ij}>0} t_{ij}, \tag{42}$$

где знак $x_{ij} > 0$ показывает, что берется максимальное не из всех t_{ij}, а только из тех, для которых перевозки отличны от нуля.

Мы хотим найти такой план перевозок x_{ij}, для которого время T обращается в минимум:

$$T = \max_{x_{ij}>0} t_{ij} \to \min.$$

Поставленная задача не является задачей линейного программирования, так как величина T, в соответствии с (42) – нелинейная функция x_{ij}. Ее решение, однако, можно свести к решению задач линейного программирования, но не одной, а нескольких.

2.2. Сетевые модели. Планирование комплекса работ

На практике часто возникают задачи рационального планирования сложных, комплексных проектов, примерами которых могут служить: строительство большого промышленного объекта, перевооружение армии или отдельных видов вооруженных сил, развертывание системы медицинских или профилактических мероприятий, выполнение какого-либо комплексного проекта с участием ряда организаций и т. д.

Характерным для таких проектов является то, что в их состав входит ряд отдельных, элементарных работ или мероприятий, которые должны выполняться в определенной последовательности, так что выполнение некоторых работ не может быть начато до того момента, как будут завершены некоторые другие. Например, при строительстве

промышленного предприятия рытье котлована не может быть начато до того как будут доставлены и смонтированы соответствующие агрегаты; укладка фундамента не может быть начата раньше, чем будут доставлены необходимые материалы, для чего, в свою очередь, требуется завершение строительства подъездных путей; для всех этапов строительства необходимо наличие соответствующей технической документации, и т. д.

Существенными для планирования любого комплексного проекта является оценка:

- времени, необходимого для выполнения проекта в целом и входящих в него работ; сроки начала и окончания работ;
- стоимости реализации проекта и отдельных работ;
- объемов сырьевых, энергетических, финансовых и др. ресурсов, которые необходимо привлечь для реализации проекта.

Соответственно, планирование проектов осуществляется с целью разработки календарного плана работ, сметы работ и плана распределения ресурсов между работами. При этом необходимо получить представление о том, какие могут возникнуть препятствия к своевременному завершению комплекса работ и как их устранять.

Одним из методов, широко применяемых при решении такого рода задач, является **метод сетевого планирования**.

Основным материалом для сетевого планирования является список работ проекта, с указанием их продолжительности и взаимной обусловленности по срокам — окончание каких работ требуется для начала выполнения каждой работы. Такой список называется **структурной таблицей** комплекса работ. В структурной таблице для каждой работы указывается время ее выполнения и перечень предшествующих ей работ, которые должны быть уже выполнены для того, чтобы можно было бы приступить к выполнению данной работы. Если работе A должны предшествовать некоторые другие работы B,C,D,… то говорят также, что работа A **опирается** на работы B,C,D,…

Рассмотрим в качестве примера проект, состоящий из 12 работ (A,B,...,L), структурная таблица которого представлена в табл. 10.

Таблица 10

Структурная таблица комплекса работ

Работа	Время выполнения, дней	Опирается на работы	Обозначение
A	30	—	(s,2)
B	7	—	(s,1)
C	10	A	(2,3)
D	14	B, C	(3,4)
E	10	D	(4,5)
F	7	E	(5,6)
G	21	B, C	(3,6)
H	7	G, F	(6,8)
I	12	G, F	(6,9)
J	15	I, H	(9,f)
K	30	B	(1,7)
L	15	F, G, K	(7,f)

Если число работ не слишком велико, информацию о проекте, заданную в форме табл. 10 удобно представить графически — в виде так называемого **сетевого графика**. Сетевой график (рис.5) представляет собой ориентированный граф, ребра которого соответствуют работам, а вершины — событиям, состоящим в завершении одних работ и возможности начать другие работы.

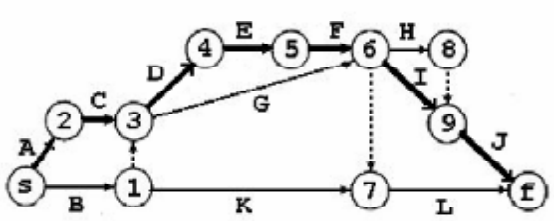

Рис. 5. Сетевой график комплекса работ

При построении сетевого графика обычно принимается следующая схема:

1. Вершины соответствуют событиям окончания одних работ и начала других работ; вводятся дополнительные вершины s (start) и f (finish), соответствующие событиям начала и окончания проекта в целом; остальные вершины помечаются произвольно.

2. Ребра, входящие в вершину (событие), соответствуют работам, которые должны быть закончены к моменту свершения события.

3. Ребра, выходящие из вершины (события), отображают работы, которые могут быть начаты в момент свершения события.

4. В случаях, когда две или более работы начинаются и/или заканчиваются одними и теми же событиями, в сетевой график могут быть введены фиктивные события и фиктивные работы с нулевым временем выполнения, таким образом чтобы любую работу можно было однозначно сопоставить с парой событий (i,j), ее начала и окончания.

5. Работа (i,j) может быть начата, как только закончатся все работы, представленные ребрами, входящими в вершину i.

На рис.5 имеются три фиктивные работы: (1,3), (6,7) и (8,9). Работа (1,3) необходима потому, что работы B и C должны непосредственно предшествовать D и G, но работе E непосредственно предшествует только D. Работа (6,7) вводится потому, что работы F, G и K предшествуют L, но работам H и I предшествуют только F и G. Третья работа (8,9) вводится для того, чтобы исключить ситуацию, когда две или более работ начинаются и заканчиваются одними и теми же событиями. Так как конечное событие работ F и G является начальным событием работ H и I, которые совместно и непосредственно предшествуют работе L, нужно ввести фиктивную операцию. Если не выполнить перечисленные условия, то однозначно определить работы событиями их начала и окончания не удастся.

Итак, мы имеем сетевой график с вершинами, представляющими события проекта, включая события s “начало проекта” и f “окончание проекта”, и ребрами (i,j), представляющими множество работ. Требуется проанализировать проект с точки зрения его продолжительности и сроков выполнения работ. Введем следующие определения.

Путь – последовательность взаимосвязанных работ, ведущая из одной вершины сетевого графика проекта в другую. Например (см. рис. 5), {A, C, G} и {C, F} – два различных пути.

Длина пути – суммарная продолжительность выполнения всех работ пути.

Критический путь – путь, длина которого является наибольшей.

Из последнего определения следует, что минимальное время выполнения проекта T равно длине критического пути. Формально, для того, чтобы найти эти времена, достаточно было бы перебрать в сетевом графике всевозможные пути из вершины s в вершину f и выбрать тот, или те из них, которые имеют наибольшую суммарную продолжительность. Однако для больших проектов простой перебор может быть связан с существенными вычислительными трудностями.

2.2.1. Поиск критического пути

Метод (CPM или Critical Path Method, т.е. метод критического пути), который рассматривается ниже, позволяет проанализировать сетевой график с точки зрения сроков выполнения работ без использования формального перебора. Метод CPM состоит из двух шагов.

Первый шаг. Вычислим ранние сроки наступления событий проекта. Обозначим через t_i ранний срок наступления события i. Пусть, проект начинается в нулевой момент времени, т.е. $t_S = 0$. Продолжительность работы (i,j) обозначим t_{ij} (для фиктивных работ $t_{ij} = 0$). Тогда значения ранних сроков наступления событий определяются по формуле:

$$t_i = \max_k [t_k + t_{ki}], \tag{43}$$

где k пробегает все события, для которых существуют работы (k,i).

В результате будет вычислен также ранний срок события f окончания всего комплекса работ. Очевидно, ранний срок наступления события f и есть минимальное время выполнения проекта в целом.

Второй шаг. Обозначим поздние сроки свершения событий проекта через T_i. Положим, что для события f окончания проекта $T_f = t_f$, т.е. поздний срок совпадает с ранним, который становится известен по окончании расчетов на первом шаге. Тогда наиболее поздние сроки завершения событий последовательно вычисляются по формуле:

$$T_i = \min_j [T_j - t_{ij}], \tag{44}$$

где j пробегает все события, для которых существуют работы (j,i).

Проделаем вычисления метода CPM для рассматриваемого примера.

В соответствии с формулой (43) для того, чтобы подсчитать ранние сроки наступления событий на первом шаге CPM достаточно «пройти» сетевой график рис.5 «по стрелкам», начиная с вершины S. На первом

этапе можно вычислить ранние сроки наступления только для событий 1 и 2, поскольку ранний срок наступления (0) известен только для события S:

$$t_2 = t_S + t_{S2} = 30, \quad t_1 = t_S + t_{S1} = 7.$$

После этого можно вычислить ранний срок наступления события 3:

$$t_3 = max[t_2 + t_{23}, \quad t_1 + t_{13}] = max[30+10, \quad 7+0] = 40.$$

Далее,

$$t_4 = t_3 + t_{34} = 40+14 = 54, \quad t_5 = t_4 + t_{45} = 54+10 = 64.$$

Для событий 6,7 и 8:

$$t_6 = max[t_5 + t_{56}, \quad t_3 + t_{36}] = max[64+7{,}40+21] = 71,$$

$$t_7 = max[t_6 + t_{67}, \quad t_1 + t_{17}] = max[71+0, \quad 7+30] = 71,$$

$$t_8 = t_6 + t_{68} = 71+7 = 78.$$

Наконец,

$$t_9 = max[t_6 + t_{69}, \quad t_8 + t_{89}] = max[71+12, \quad 78+0] = 83,$$

$$t_f = max[t_9 + t_{9f}, \quad t_7 + t_{7f}] = max[83+15, \quad 71+15] = 98.$$

Первый шаг закончен. Минимальное время, необходимое для выполнения проекта и длина критического пути нам известны. Теперь необходимо определить критические работы. Положим

$$T_f = t_f = 98,$$

и перейдем ко второму шагу.

Согласно формуле (44) для того, чтобы на втором шаге подсчитать поздние сроки наступления событий необходимо «пройти» сетевой график рис.5 «против стрелок», начиная с вершины f. На первом этапе, поскольку задан лишь поздний срок наступления события F, можно вычислить только поздние сроки наступления событий 7,8 и 9:

$$T_7 = T_f - t_{7f} = 98-15 = 83, \quad T_9 = T_f - t_{9f} = 98-15 = 83, \quad T_8 = T_9.$$

Далее,

$$T_6 = min[T_7 - t_{67}, \quad T_8 - t_{68}, \quad T_9 - t_{69}] = min[83-0, \quad 83-7, \quad 83-12] = 71,$$

$$T_5 = T_6 - t_{56} = 71-7 = 64, \quad T_4 = T_5 - t_{45} = 64-10 = 54,$$

$$T_3 = min[T_4 - t_{34}, \quad T_6 - t_{36}] = min[54-14, \quad 71-21] = 40,$$

$$T_2 = T_3 - t_{23} = 40-10 = 30.$$

И наконец,

$$T_1 = min[T_7 - t_{17},\ \ T_3 - t_{13}] = min[83-30,\ \ 40-0] = 40,$$

$$T_S = min[T_1 - t_{S1},\ \ T_2 - t_{S2}] = min[40-7,\ \ 30-30] = 0.$$

Представим полученные результаты в виде таблицы сроков наступления событий проекта (табл. 11):

Таблица 11

Событие	t_i	T_i	$R_i = T_i - t_i$
S	0	0	0
1	7	40	33
2	30	30	00
3	40	40	0
4	54	54	0
5	64	64	0
6	71	71	0
7	71	83	12
8	78	83	5
9	83	83	0
F	98	98	0

В соответствии с вышесказанным события, для которых $t_i = T_i$ лежат на критическом пути. В табл.11 критические события выделены жирным шрифтом. Работы (i, j), начало и окончание которых суть критические события, образуют критический путь.

Заметим, что на сетевом графике вообще говоря может быть не один критический путь, а два или более; естественно, сумма времен критических работ на каждом из них должна быть одна и та же.

Обратим внимание на следующее обстоятельство: время T представляет собой сумму времен исполнения не всех работ, а только некоторых из них:

$$T = t_{S2} + t_{23} + t_{34} + t_{45} + t_{56} + t_{69} + t_{9F} = 98.$$

Работы A,C,D,E,F,I,J, из длительностей которых составлено время T, называются **критическими работами**, соответствующий им путь на сетевом графике является критическим путем. На рис.5 критический путь показан толстыми стрелками.

Характерная особенность критических работ состоит в следующем. Если такая работа будет отложена на некоторое время, то время окончания проекта будет отложено на то же время. Для того, чтобы выполнить проект в целом за минимальное время, каждая из критических работ должна быть начата точно в момент, когда закончена предыдущая критическая работа, и продолжаться не более того времени, которое ей отведено по плану. Малейшее опоздание в выполнении любой из критических работ приводит к соответствующей задержке выполнения плана в целом. Таким образом, критический путь на сетевом графике – это перечень «слабых мест» плана, которые должны укладываться во временной график с наибольшей четкостью. Если необходимо сократить время выполнения проекта, то в первую очередь нужно сокращать время выполнения хотя бы одной работы на критическом пути.

Что касается остальных, «некритических» работ (в нашем случае это B,G,H,K), то каждая из них имеет известные временные резервы и может быть начата и закончена с некоторым опозданием без изменения сроков выполнения проекта в целом. Точнее говоря, время начала некритической работы (i,j) можно отложить на срок, равный разности между наиболее поздним и наиболее ранним сроками события i начала этой работы.

В состав реальных проектов могут входить сотни и тысячи работ, сложным образом опирающиеся друг на друга. Поэтому графическое представление проекта в виде сетевого графика обычно не используется, а анализ проектов осуществляется с помощью ЭВМ.

Метод CPM достаточно прост для программирования. Обычно при машинном анализе проекта заданную структурную таблицу упорядочивают таким образом, чтобы каждая работа опиралась бы только на работы с меньшими порядковыми номерами. Для этого вводится понятие **ранга** работы. Для начала работ первого ранга не требуется окончания никаких других работ. Работа второго ранга опирается на одну или несколько работ первого ранга и т.д. Вообще, работа называется работой k-го ранга, если она опирается на одну или несколько работ ранга не выше (k–1), среди которых есть хотя бы одна работа (k–1)-го ранга. После того как вычислены ранги всех работ, для упорядочения структурной таблицы достаточно перенумеровать (т.е. переставить) в ней работы в порядке возрастания ранга, нумеруя

работы одного ранга произвольном порядке. В результате в новой упорядоченной структурной таблице каждая работа будет опираться только на работы с меньшим порядковым номером. Упорядоченная структурная таблица при расчете критического пути на ЭВМ может сканироваться последовательно, в порядке возрастания номера работы.

Знание критического пути полезно в двух отношениях. Во-первых, оно позволяет выделить перечень наиболее «угрожаемых» работ проекта, следить за ними и, при необходимости, форсировать их выполнение. Во-вторых, оно дает возможность ускорить выполнение комплекса работ за счет привлечения резервов, скрытых в некритических работах.

Рассмотренный метод построения сетевого графика (CPM) исходит из предположения, что заранее точно известны длительности t_{ij} каждой работы. На самом деле длительности работ зависят от множества факторов и фактически являются случайными величинами. Неучет этого обстоятель-ства может привести к значительным ошибкам в определении общего срока выполнения проекта. Существуют методы анализа проектов[1], которые позволяют учесть случайный характер времени выполнения работ проекта.

В качестве примера рассмотрим здесь кратко один из таких методов — известный метод PERT (Program Evaluation and Review Technique – метод оценки и обзора проекта).

Для того, чтобы использовать метод PERT, для каждой работы ij, время выполнения которой является случайной величиной, необходимо определить следующие три оценки:

- оптимистическое время a_{ij} - время выполнения работы ij в наиболее благоприятных условиях;
- наиболее вероятное время m_{ij} - время выполнения работы ij в нормальных условиях;
- пессимистическое время b_{ij}- время выполнения работы ij в неблагоприятных условиях.

Учитывая, что время выполнения работ, обычно, хорошо описывается бета – распределением вероятностей, среднее или ожидаемое время t_{ij} выполнения работы ij может быть определено по формуле:

$$t_{ij} = \frac{a_{ij} + 4m_{ij} + b_{ij}}{6}.$$

---

[1] См. кн. Афанасьев М.Ю., Суворов Б.П. Исследование операций в конкретных ситуациях. - М.: МГУ, 1999

Если для некоторой работы ij время ее выполнения известно точно и равно d_{ij}, то:

$$t_{ij} = a_{ij} = m_{ij} = b_{ij} = d_{ij}.$$

Располагая указанными выше тремя оценками времени выполнения работы, мы можем также рассчитать общепринятую статистическую меру неопределенности – дисперсию σ_{ij}^2 или вариацию Var_{ij} времени выполнения работы (i,j):

$$\sigma_{ij}^2 = Var_{ij} = \left(\frac{b_{ij} - a_{ij}}{6} \right)^2.$$

Если время выполнения работы ij известно точно, то $\sigma_{ij} = Var_{ij} = 0$.

Пусть T - время, необходимое для выполнения проекта. Если в проекте есть работы с неопределенным временем выполнения, то время T является случайной величиной. Математическое ожидание (ожидаемое значение) времени выполнения проекта $E(T)$ равно сумме ожидаемых значений времени выполнения работ, лежащих на критическом пути. Для определения критического пути проекта может быть использован метод CPM. На этом этапе анализа проекта время выполнения работы полагается равным ожидаемому времени t_{ij}. Вариация (дисперсия) общего времени, требуемого для завершения проекта, в предположении о независимости времен выполнения работ равна сумме вариаций работ критического пути. Если же две или более работы взаимозависимы, то указанная сумма дает приближенное представление о вариации времени завершения проекта.

Распределение времени T завершения проекта является ассимптотически нормальным со средним $E(T)$ и дисперсией $\sigma^2(T)$. С учетом этого можно рассчитать вероятность завершения проекта в установленный срок T_0. Для определения вероятности того, что $T \leq T_0$, используется таблица стандартного нормального распределения величины:

$$Z = \frac{T - E(T)}{\sigma(T)}.$$

2.2.2. Задачи оптимизации плана комплекса работ

Как уже говорилось, сетевой график (или заменяющий его алгоритм анализа комплекса работ) может быть использован для улучшения (оптимизации) плана с различными целями.

Если окажется, что общее время выполнения комплекса работ нас не устраивает; возникает вопрос о том, как нужно форсировать работы для того, чтобы общее время не превосходило заданного срока. Очевидно, для этого имеет смысл форсировать именно критические работы, снижение длительности которых непосредственно скажется на времени выполнения комплекса работ. Однако при этом может оказаться, что критический путь изменится, и наиболее слабыми местами по времени окажутся другие работы. Естественно предположить, что форсирование работ требует вложения каких-то средств. Возникает задача интенсификации работ: какие дополнительные средства следует распределить между работами, чтобы общий срок выполнения комплекса работ был не больше заданной величины, а расход дополнительных средств был минимальным?

Другая задача оптимизации возникает в связи с возможностью перераспределения ресурсов между отдельными работами. Мы убедились, что все работы, кроме критических, имеют временные резервы. Если за счет переброски ресурсов с некритических участков плана на критические можно добиться уменьшения общего времени выполнения работ, то можно ставить задачу о том, какие ресурсы надо перебросить с одних работ на другие для того, чтобы время выполнения комплекса работ стало минимальным.

Наконец, возможна еще одна постановка задачи оптимизации плана. Пусть после построения сетевого графика стало известно, что минимальное время выполнения всего комплекса работ укладывается в заданный срок с избытком, т.е. существует известный резерв времени, которым можно распоряжаться. Растянув работы в пределах установленного резерва, можно сэкономить некоторые средства. Возникает вопрос: до каких пределов можно увеличивать время выполнения работ и каких работ, чтобы полученная от этого экономия средств была максимальной? В такой постановке может ставиться задача оптимизации необязательно всего плана, а только отдельных некритических работ, для которых выявлены временные резервы.

Дадим постановку каждой из этих трех задач в формульной записи. Для простоты будем предполагать, что критический путь – единственный.

Задача 1: Проект состоит из работ $a_1,a_2,\ldots,a_n$ с временами выполнения $t_1,t_2,\ldots,t_n$; известен критический путь, причем время выполнения комплекса равно:

$$T = \sum_{(ккр} t_i > T_0,$$

где суммирование распространяется только на критические работы, T_0 – заданный срок выполнения комплекса работ.

Известно, что вложение определенных дополнительных средств x_i в работу a_i сокращает время ее выполнения с t_i до $t_i' = t_i - f_i(x_i) < t_i$.

Спрашивается, какие минимальные дополнительные средства $x_1,x_2\ldots,x_n$ следует вложить в каждую из работ, чтобы срок выполнения комплекса стал не выше заданного T_0. Иначе говоря, требуется определить неотрицательные значения переменных x_1, x_2, …, x_n (дополнительные вложения) так, чтобы выполнялось условие:

$$T' = \sum_{(кр)} t_i - f(x_i) \le T_0, \qquad (45)$$

где суммирование распространяется по всем критическим работам нового критического пути (полученного после перераспределения средств и изменения времен), и при этом общая сумма дополнительных вложений была минимальна:

$$x = \sum_{i=1}^{n} x_i \to min. \qquad (46)$$

Входящие в ограничения (45) функции $f_i(x)$ нелинейны, поскольку сокращение времени выполнения работы обычно не пропорционально вложенным средствам. Таким образом, мы имеем задачу на минимум линейной функции вложений (46) при нелинейных ограничениях на переменные, т.е. задачу нелинейного программирования.

Задача 2: Имеется проект, включающий работы $a_1,a_2,\ldots,a_n$ с временами выполнения $t_1,t_2,\ldots,t_n$. Время выполнения проекта есть:

$$T = \sum_{(кр)} t_i,$$

где суммирование распространяется только на критические работы, а на некритических работах имеются некоторые резервы времени. Пользуясь этими резервами, т.е. перебрасывая какие-то средства с некритических работ на критические, можно уменьшить времена

выполнения критических работ и тем самым — общее время выполнения работ.

Имеется некоторый фиксированный запас подвижных средств B, который распределен между работами $a_1, a_2, \ldots, a_n$ так, что каждой работе выделены средства из запаса в объеме b_1, b_2, .. b_n, где:

$$\sum_{i=1}^{n} b_i = B .$$

Известно, что средства x>0, снятые с работы a_j, увеличивает время ее выполнения до $t_i' > t_i$, а средства, вложенное дополнительно в работу a_i уменьшает время ее выполнения до $t_i'' < t_i$.

Требуется так перераспределить имеющиеся подвижные средства B между работами, чтобы срок выполнения комплекса был минимальным.

Покажем, как может быть решена подобная задача. Обозначим x_i — количество подвижных средств, перебрасываемых на работу a_i; (x_i отрицательно, если с работы a_i снимается какое-то количество средств). Величины x_i должны удовлетворять ограничениям:

$$x_i \geq -b_i, \quad i = 1,2,\ldots,n . \tag{47}$$

Естественно, что сумма средств, снимаемых с каких-то работ, должна быть равна сумме средств, добавляемых другим работам, так что:

$$x_1 + x_2 + \ldots + x_n = 0 . \tag{48}$$

После переброски средств для тех работ, на которые они перебрасываются, новые времена будут равны:

$$t_i'' = \varphi_i(x) < t_i ,$$

для тех же работ, с которых средства снимаются:

$$t_i' = f_i(x) > t_i ,$$

общий срок выполнения комплекса работ будет:

$$T' = \sum_{(\text{кр})_i} \varphi(x_i) + \sum_{(\text{кр})_j} f_j(|x_j|) \quad , \tag{49}$$

где первая сумма распространяется на все работы, на которые переносятся средства, если они входят в критический путь; вторая – на все работы, с которых снимаются средства, если они входят в критический путь.

Естественно, кажется, считать, что перенос средств имеет смысл только с некритических работ на критические; однако не надо забывать, что при этом некритические работы могут переходить в критические, и наоборот; поэтому в формуле (49) в общем случае присутствуют обе суммы.

Таким образом, требуется найти значения переменных x_i (i = 1, ... , n), доставляющие минимум функции (49) при ограничениях (47),(48).

Эта задача также представляет собой задачу нелинейного программирования, поскольку, даже если (с некоторой натяжкой) допустить что функции f_j, и φ_i линейны, значения i, j – номеров работ, на которые распространяется сумма (т. е. критических работ), сами зависят от x_i.

Задача 3: Имеется проект, включающий работы a_1, a_2, ..., a_n с временами выполнения t_1, t_2, ... , t_n. Для этого проекта найден критический путь и установлено, что минимальное время его выполнения $T < T_0$, где T_0 – заданный нормативный срок выполнения проекта. Предполагается, что можно получить экономию средств за счет снижения темпов выполнения некоторых работ так, чтобы срок выполнения комплекса не превысил заданного значения T_0.

Увеличение времени выполнения работы a_i на τ, т.е. до $t_i' = t_i + \tau$ высвобождает некоторые средства x_i, которые зависят от задержки τ:

$$x_i = f_i(\tau).$$

Требуется так распределить задержки между работами, чтобы экономия средств была максимальна при сохранении заданного общего срока выполнения работ T_0.

Обозначим τ_i, время задержки работы a_i. Сумма времен выполнения работ, лежащих на критическом пути, не должна превосходить T_0:

$$\sum_{(кр)} (t_i + \tau_i) \le T_0 ,$$

где сумма, как и ранее, распространяется только на критические работы. Требуется выбрать такие неотрицательные значения переменных τ_i чтобы сумма высвободившихся средств достигала максимума:

$$\sum_i f_i(\tau_i) \to max .$$

Поставленная задача снова относится к классу задач нелинейного программирования. В случае, когда речь идет только о незначительных задержках τ_i, иногда удается свести ее к задаче линейного

программирования (а именно, если функции f_i близки к линейным в диапазоне возможных значений τ_i, а критический путь при задержках не меняется).

2.3. Модели динамического программирования

2.3.1. Основные идеи метода динамического программирования

Некоторые задачи математического программирования обладают специфическими особенностями, которые позволяют свести их решение к рассмотрению некоторого множества более простых «подзадач». В результате вопрос о глобальной оптимизации целевой функции сводится к поэтапной оптимизации некоторых промежуточных целевых функций. В **динамическом программировании**[1] рассматриваются методы, позволяющие путем поэтапной (многошаговой) оптимизации получить общий (результирующий) оптимум.

Обычно методами динамического программирования оптимизируют работу **управляемых систем**, эффект которой оценивается **аддитивной**, или **мультипликативной**, целевой функцией (функцией выигрыша). **Аддитивной** называется такая функция нескольких переменных $f(x_1,x_2,...,x_n)$, которую можно представить как сумму функций $f_j(x_j)$, зависящих только от одной переменной x_j:

$$f(x_1,x_2,...,x_n)=\sum_{1}^{n} f(x_j). \qquad (50)$$

Слагаемые аддитивной целевой функции соответствуют эффекту решений, принимаемых на отдельных этапах управляемого процесса. По аналогии, мультипликативная функция распадается на произведение положительных функций различных переменных:

$$f(x_1,x_2,...,x_n)=\prod_{1}^{n} f_j(x_j). \qquad (51)$$

Поскольку логарифм функции типа (51) является аддитивной функцией, достаточно ограничиться рассмотрением функций вида (50).

---

[1] Динамическое программирование как научное направление возникло и сформировалось в 1951-1953 гг. благодаря работам Р. Беллмана и его сотрудников.

Изложим сущность метода динамического программирования на примере задачи оптимизации на максимум (или минимум) аддитивной функции:

$$f(x_1,x_2,...,x_n)=\sum_1^n f(x_j)\to max, \tag{52}$$

при ограничениях

$$\sum_1^n a_j x_j \le b, \quad a_j>0, \quad x_j \ge 0. \tag{53}$$

Отметим, что в отличие от задач, рассмотренных в предыдущих главах, о линейности или дифференцируемости функций $f_j(x_j)$ не делается никаких предположений, поэтому применение классических методов оптимизации (например, метода множителей Лагранжа) для решения задачи (52)—(53) либо проблематично, либо просто невозможно.

Предварительно рассмотрим несколько примеров.

2.3.2. Задача о наборе высоты и скорости летательным аппаратом

Пусть самолет (или другой летательный аппарат), находящийся на высоте H_0 и имеющий скорость V_0, должен быть поднят на заданную высоту H_ω, а скорость его доведена до заданного значения V_ω (буквой ω мы будем отмечать конец процесса). Известен расход горючего, потребный для подъема аппарата с любой высоты H на любую другую $H' > H$ при неизменной скорости V, известен также расход горючего, потребный для увеличения скорости от любого значения V до $V' > V$, при неизменной высоте H.

Требуется найти оптимальный режим набора высоты и скорости, при котором общий расход горючего будет минимальным.

Решение будем строить следующим образом. Предположим, что весь процесс набора высоты и скорости можно разделить на ряд последовательных этапов или шагов, причем за один шаг самолет может увеличить либо только высоту, либо — только скорость.

Тогда траектория самолета на плоскости $\{V,H\}$ (рис. 6), где V – скорость, а H – его высота будет иметь вид некоторой ступенчатой кривой:

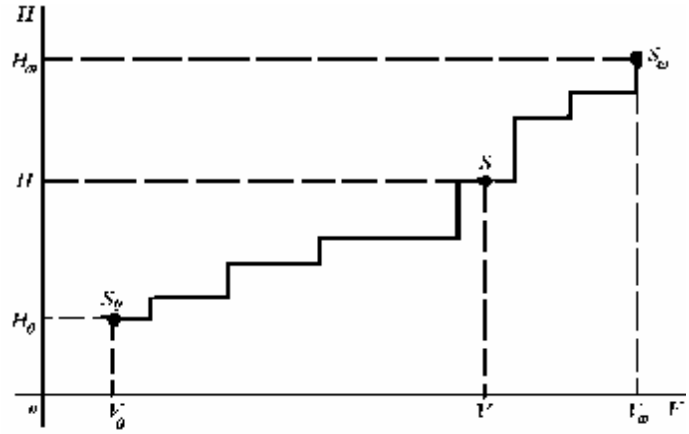

Рис. 6

Ясно, что существует множество таких ступенчатых траекторий, по которым можно перевести самолет из начального состояния S_0 в конечное S_ω .

Из всех этих траекторий нужно выбрать ту, на которой расход горючего будет минимальным.

Для того, чтобы решить нашу задачу методом динамического программирования этого разделим интервал скоростей $V_\omega - V_0$ на n_1 равных частей с шагом по скорости:

$$\Delta V = \frac{V_\omega - V_0}{n_1},$$

а интервал высот $H_\omega - H_0$ – на n_2 равных частей с шагом по высоте:

$$\Delta H = \frac{H_\omega - H_0}{n_2}.$$

Число частей n_1 и n_2 может быть выбрано, например, исходя из требований к точности решения задачи. Так как за каждый шаг мы обязательно меняем высоту или скорость, то общее число m шагов будет:

$$m = n_1 + n_2$$

Например, для случая, изображенного на (рис. 7) $n_1 = 8$, $n_2 = 6$, $m = 14$.

Управление, посредством которого реализуется выбор того или иного режима набора высоты и скорости (и соответствующая

ступенчатая траектория самолета) заключается в том, что на каждом шаге процесса мы указываем один из двух вариантов полета — набор высоты или скорости. Такой способ управления самолетом можно описать формально, если ввести на каждом шаге управляющее воздействие u_k, $k=1,2,...,14$ и считать, что переменные u_k могут принимать лишь два значения — $u_k=0$, если на k-том шаге производится набор скорости без изменения высоты, или $u_k=1$, если на k-том шаге производится набор высоты без изменения скорости.

Таким образом, выбор той или иной ступенчатой траектории осуществляется нами посредством задания определенного набора (вектора) двузначных управляющих переменных $\mathbf{U}=(u_1,u_2,...,u_{14})$.

Чтобы выбрать траекторию, на которой расход горючего минимален, необходимо знать расход на каждом шаге (горизонтальном или вертикальном участке траектории). Предположим, что эти расходы заданы, например в виде рис. 7, где на каждом отрезке записан расход горючего в условных единицах.

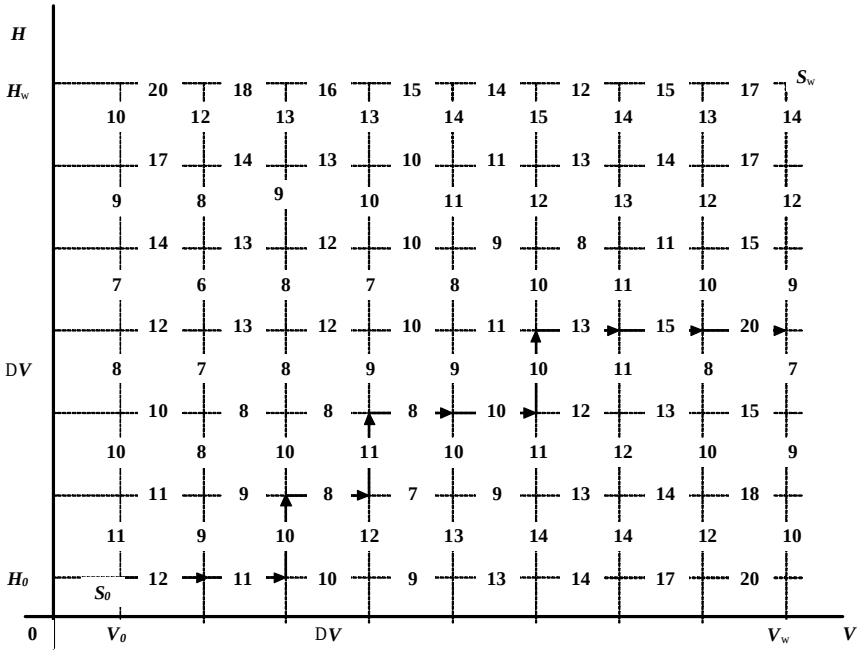

Рис. 7

Любой ступенчатой траектории из S_0 в S_ω, соответствует теперь вполне определенный расход горючего, равный сумме чисел, указанных на отрезках. Например, траектория, помеченная на рисунке стрелками, дает расход горючего:

W= 12 + 11 + 10 + 8 + 11 + 8 + 10 + 10 + 13 + 15 + 20+ 9 + 12 + 14 =163.

Для того, чтобы выбрать траекторию с минимальным расходом горючего, можно было бы, конечно, просто перебрать всевозможные траектории, но их слишком много. Потому задачу будет решать методом динамического программирования.

Начнем с последнего шага. Конечное состояние самолета S_ω задано; последний шаг должен привести именно в эту точку. Посмотрим, из каких состояний можно переместиться в точку S_ω за один шаг, т.е. каковы возможные состояния самолета после предпоследнего, 13-го шага? Рассмотрим отдельно правый верхний угол прямоугольной сетки (рис. 8) с конечной точкой S_ω.

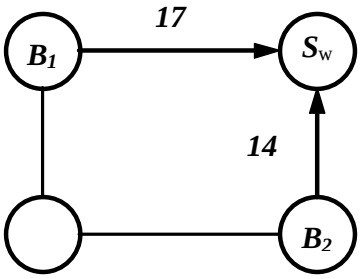

Рис. 8

За один шаг в конечную точку можно переместиться из точек: B_1 и B_2 причем из каждой – только одним способом, так что выбора управления на последнем шаге у нас нет – оно единственно. Если предпоследний шаг привел нас в точку B_1, мы должны набирать скорость с расходом горючего 17 единиц; если же в точку B_2 –набирать высоту с расходом горючего 14 единиц. Пометим этими минимальными (в данном случае просто неизбежными) расходами точки B_1 и B_2 (рис.9). Пометка «17» у B_1 означает: «если в процессе управления самолет попадет в B_1, то минимальный расход горючего, переводящий его в точку S_ω, составит 17 единиц». Аналогичный смысл имеет запись «14» в квадрате у точки B_2. Соответствующее оптимальное управление указано в каждом случае стрелкой, которая показывает направление движения из данной точки

к S_ω, при условии, что под действием предыдущих управлений самолет попадет в данную точку.

Таким образом, условное оптимальное управление на последнем, 14-м шаге, найдено для любого исхода тринадцатого шага. Для каждого из найден найден, кроме того, условный минимальный расход горючего, за счет которого можно переместиться в точку S_ω. Термин «условный» означает, что найденные нами оптимальное управление и оптимальный расход горючего зависят от состояния самолета, достигнутого на предыдущем шаге (B_1 или B_2), которого мы не знаем, т.к. задача еще не решена.

Перейдем к планированию предпоследнего, 13-го шага. Для этого рассмотрим возможные состояния самолета, достигнутые после 12-го шага. К исходу шага 12 самолет может быть в одной из точек C_1, C_2, C_3 (рис.10). Из каждой такой точки мы должны найти оптимальный путь в точку S_ω и соответствующий этому пути минимальный расход горючего.

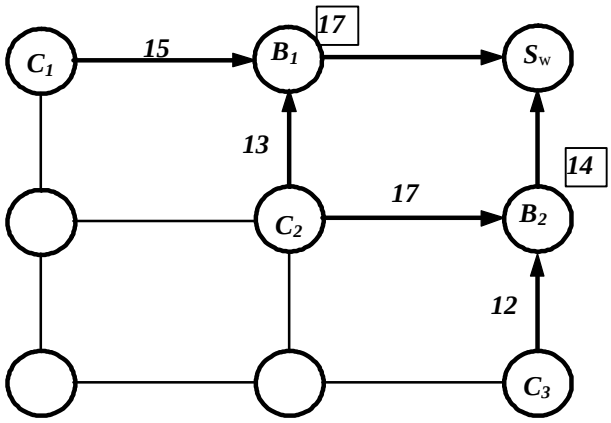

Рис. 9

Если после 12-го шага самолет был в точке C_1, то выбора нет: можно только набирать скорость и на это требуется 15+17 = 32 единицы горючего. Значение этого расхода используем в качестве дополнительной метки C_1, а оптимальное (в данном случае единственное) управление из точки C_1 обозначим стрелкой. Из точки C_3 путь в S_ω также единственный: только с набором высоты и расходом горючего 12 +14 = 26.

В точке C_2 выбор есть: из нее можно достичь S_ω либо через B_1, либо через B_2. В первом случае потребуется 13+17=30 единиц горючего, во втором — 17+14=31 единицу. Следовательно оптимальный путь из C_2 в S_ω начинается набором высоты, а минимальный расход горючего равен 30. Пометим этим числом дополнительно точку C_2.

Продолжая таким образом для шагов номера $k = 12,11,...,2$, можно для каждой узловой точки найти условное оптимальное управление на следующем шаге (заданное стрелкой или значением $u_k = 0$ или 1) и условно оптимальный расход горючего от данной точки до конечной точки S_ω при условии что управляемый процесс достигнет этой узловой точки. Наконец на последнем шаге, при k=1, возникнет ситуация показанная на рис. 10.

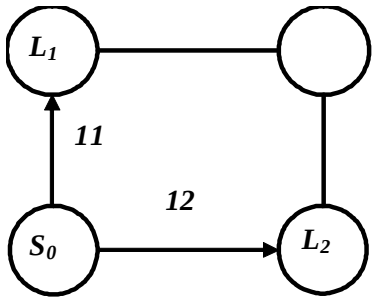

Рис. 10

Как видно из рисунка при k=1 существует два варианта управления самолетом: при наборе высоты ($u_1 = 1$) точка, изображающая состояние самолета, переместится из S_0 в L_1; при наборе скорости ($u_1 = 0$) точка переместится из S_0 в L_2. Но для каждой точки L_1 и L_2 уже известны условное оптимальное управление и условно оптимальный расход горючего. Поэтому для выбора минимального расхода горючего на всю траекторию теперь достаточно сложить значения расхода горючего для этих двух вариантов с соответствующими условно оптимальными расходами и сравнить полученные значения.

Тем самым будут найдены оптимальный расход горючего на всю траекторию и оптимальное управление u_1. Дальнейшие (уже безусловные) оптимальные управления и оптимальную ступенчатую

траекторию нетрудно теперь восстановить перемещаясь по стрелкам из S_0 в S_ω.

Рекомендуется самостоятельно проделать все необходимые вычисления, пользуясь данными рис. 7. Полученный в результате минимальный расход горючего должен быть:

$$W_{min}=139.$$

Рекомендуется также проверить, что меньший расход нельзя получить ни на какой другой траектории кроме оптимальной.

2.3.3. Задача об оптимальном распределении ресурсов

Рассмотрим проблему оптимального вложения некоторых ресурсов $u_1,u_2,...,u_n$ в различные активы (инвестиционные проекты, предприятия и т.п.). Предполагается, что все виды ресурсов с помощью заданных числовых коэффициентов a_j приведены к одной размерности (например, денег), активы характеризуются функциями прибыли $f_j(u_j)$, а суммарный объем всех видов ресурсов ограничен. Требуется найти такое распределение ограниченного объема ресурса b, которое максимизирует суммарную прибыль:

$$f(u_1,u_2,...,u_n)=\sum_{1}^{n} f_j(u_j)\to max, \qquad (54)$$

при ограничениях:

$$\sum_{j=1}^{n} a_j u_j \le b, \quad a_j>0, \quad u_j\ge 0. \qquad (55)$$

Представим, что задача решается последовательно для каждого актива. Пусть на первом шаге принято решение о вложении в n-й актив u_n единиц ресурса. Тогда на остальных шагах мы сможем распределить $b-a_n u_n$ единиц ресурса. Отвлекаясь (временно) от соображений, на основе которых принималось решение на первом шаге, вполне естественно поступить так, чтобы на оставшихся шагах распределение текущего объема ресурса произошло оптимально, что равнозначно решению задачи:

$$\sum_{j=1}^{n-1} f_j(u_j) \to max, \tag{56}$$

при ограничениях:

$$\sum_{j=1}^{n-1} a_j u_j \le b - a_n u_n, \quad a_j > 0, \quad u_j \ge 0. \tag{57}$$

Очевидно, что максимальное значение в (56) зависит от размера распределяемого остатка. Поэтому, если оставшееся количество ресурса $b - a_n u_n$ обозначить через ξ, то величину (56) можно представить как функцию от ξ:

$$\Phi_{n-1}(\xi) = \max_{u_1, u_2, \ldots u_{n-1}: \sum_1^{n-1} a_j u_j \le \xi} \sum_{j=1}^{n-1} f_j(u_j), \tag{58}$$

где индекс $n-1$ указывает на оставшееся количество шагов. Тогда суммарный доход, получаемый как следствие решения, принятого на первом шаге, и оптимальных решений, принятых на остальных шагах, будет:

$$W_n(u_n) = f_n(u_n) + \Phi_{n-1}(b - a_n u_n). \tag{59}$$

Предположим, что для оставшихся шагов задача решена, т.е. функция (58) построена. Вернемся теперь к выбору решения на первом шаге. Если имеется возможность влиять на u_n, то для получения максимальной прибыли следует максимизировать W_n по переменной u_n, т.е. фактически решить задачу:

$$\max_{0 \le u_n \le \frac{b}{a_n}} W_n(u_n) = \max_{0 \le u_n \le \frac{b}{a_n}} \left[f_n(u_n) + \Phi_{n-1}(b - a_n u_n) \right]. \tag{60}$$

В результате мы получаем выражение для значения целевой функции задачи при оптимальном поэтапном процессе принятия решений о распределении ресурса. Оно в силу построения данного процесса равно глобальному оптимуму целевой функции:

$$\max_{u_1, u_2, \ldots, u_n} \sum_{j=1}^{n} f_j(u_j) = \Phi_n(b), \tag{61}$$

т.е. значению целевой функции при одномоментном распределении ресурса.

При этом выражение (60) можно рассматривать (заменяя b на ξ, и n на к) как рекуррентную формулу, позволяющую последовательно вычислять оптимальные значения целевой функции при распределении объема оставшегося ресурса ξ, за k шагов:

$$\Phi_k(\xi) = \max_{0 \le u_k \le \frac{\xi}{a_k}} \left[f_k(u_k) + \Phi_{k-1}(\xi - a_k u_k) \right]. \qquad (62)$$

Зависящее от ξ значение переменной u_k, при котором достигается рассматриваемый максимум, обозначим $\psi_k(\xi)$. При k = 1 формула (61) принимает вид:

$$\Phi_1(\xi) = \max_{0 \le u_1 \le \xi / a_1} f_1(u_1), \qquad (63)$$

что допускает непосредственное вычисление функций $\Phi_1(\xi)$ и $u_1^*(\xi)$.

Если из (63) определены функции $\Phi_1(\xi)$ и $u_1^*(\xi)$, то далее можно с помощью (62) последовательно вычислить $\Phi_k(\xi)$ и $\psi_k(\xi)$ для k=2,…,n.

Положив на последнем шаге в силу (61), найдем глобальный максимум функции (58) $\Phi_n(b)$ и компоненту оптимального плана $u_n^* = \psi_n(b)$. Так как ранее по ходу решения функции $\Phi_{n-1}(\xi)$ и $\psi_{n-1}(\xi)$ уже вычислены, то полученное значение u_n^* позволяет найти нераспределенный остаток на предпоследнем шаге. Именно, при оптимальном планировании $\xi_{n-1} = b - a_n u_n^*$, и, соответственно, $u_{n-1}^* = \psi_{n-1}(\xi_{n-1})$. В результате последовательно будут найдены все компоненты оптимального плана.

Таким образом, динамическое программирование представляет собой целенаправленный перебор вариантов, который приводит к нахождению глобального максимума. Уравнение (62), выражающее оптимальное решение на k-том шаге через решения, принятые на предыдущих шагах, называется основным рекуррентным соотношением динамического программирования.

Следует заметить, что рассмотренная схема решения в столь общей постановке имеет чисто теоретическое значение, так как сводит исходную задачу (56),(57) к другой сложной проблеме построения

функций $\Phi_k(\xi)$ (k=1,2,…,n). Однако в тех случаях, когда удается решить эту проблему, метод динамического программирования может оказаться весьма полезным.

2.3.4. Принцип оптимальности Беллмана

Еще раз подчеркнем, что смысл подхода, реализуемого в динамическом программировании, заключен в замене решения исходной многомерной задачи последовательностью задач меньшей размерности.

Перечислим основные требования к задачам, выполнение которых позволяет применить данный подход:

- объектом оптимизации является управляемая система с заданными допустимыми состояниями и допустимыми управлениями;
- задача допускает интерпретацию как многошаговый процесс, каждый шаг которого состоит из принятия решения о выборе одного из допустимых управлений, приводящих к изменению состояния системы;
- последующее состояние, в котором оказывается система после выбора решения на k-м шаге, зависит только от исходного состояния к началу k-го шага и управления. Данное свойство является основным с точки зрения возможности использовать метод динамического программирования и называется **отсутствием последействия**.

Рассмотрим вопросы применения модели динамического программирования в обобщенном виде. Пусть стоит задача управления некоторым абстрактным объектом, который может пребывать в различных **состояниях**. Текущее состояние объекта отождествляется с некоторым набором параметров, который обозначается в дальнейшем через ξ, и именуется **вектором состояния**. Предполагается, что задано множество $\aleph$ всех возможных состояний. Для объекта определено также множество допустимых управлений (управляющих воздействий) $\Re$, которое, не умаляя общности, можно считать числовым множеством. Управляющие воздействия могут осуществляться в дискретные моменты времени k=1,2,…,n, причем управленческое решение заключается в выборе одного из управлений $u_k \in \Re$. **Планом задачи** или **стратегией управления** называется вектор $\mathbf{U} = (u_1, u_2, ..., u_{n-1})$, компонентами которого служат управления, выбранные на каждом шаге процесса. Ввиду предполагаемого **отсутствия последействия**, существует определенная функциональная зависимость между каждыми двумя

последовательными состояниями объекта ξ_k и ξ_{k+1}, включающая также выбранное управление: $\xi_{k+1} = \varphi_k(u_k, \xi_k)$, $k = 1,2,...,n-1$. Тем самым задание начального состояния объекта $\xi_1 \in \aleph$ и выбор плана $\mathbf{U}$ однозначно определяют дальнейшее поведение объекта.

Эффективность управления на каждом шаге зависит от текущего состояния ξ_k, выбранного управления u_k и количественно оценивается с помощью функций $f_k(u_k, \xi_k)$, являющихся слагаемыми **аддитивной целевой функции**, которая характеризует общую эффективность управления объектом. Отметим, что в определение функции обычно включается область допустимых значений u_k, и эта область, как правило, зависит от текущего состояния ξ_k. **Оптимальное управление**, при заданном начальном состоянии ξ_1, сводится к выбору такого оптимального плана $\mathbf{U}^*$, при котором достигается **максимум суммы** значений слагаемых аддитивной целевой функции на всей траектории.

Основной принцип динамического программирования заключается в том, что на каждом k-том шаге оптимальное управление u_k должно выбираться исходя из оптимальности всех последующих шагов, а не исходя из условий оптимальности только k-той функции $f_k(u_k, \xi_k)$ изолированно. Формально указанный принцип реализуется путем отыскания на каждом шаге **условных оптимальных управлений** $u_k(\xi), \xi \in \aleph$, обеспечивающих наибольшую суммарную эффективность начиная с данного шага, в предположении, что текущим является состояние ξ.

Обозначим $\Phi_k(\xi)$ максимальное значение суммы функций f_k начиная от текущего шага k и до конца процесса (шага n), при условии, что объект в начале текущего шага находится в состоянии ξ. Тогда функции $\Phi_k(\xi)$ должны удовлетворять рекуррентному соотношению:

$$\Phi_k(\xi) = \max_{u_k}[f_k(u_k, \xi) + \Phi_{k+1}(\xi_{k+1})], \qquad (64)$$

где $\xi_{k+1} = \varphi_k(u_k, \xi)$.

Соотношение (64) называют **функциональным уравнением Беллмана**. Оно является основным рекуррентным соотношением динамического программирования и реализует его базовый принцип, известный также как принцип оптимальности Беллмана.

Оптимальное управление обладает тем свойством, что каково бы ни были предыдущие управления и достигнутое в результате их применения текущее состояние системы, последующие управления на оставшихся шагах должны быть оптимальными относительно этого достигнутого состояния.

Основное соотношение (64) позволяет найти функции $\Phi_k(\xi)$ только в сочетании с **начальным условием**, каковым в нашем случае является равенство:

$$\Phi_n(\xi) = \max_{u_n}[f_n(u_n, \xi)].$$

Сравнение рекуррентной формулы (64) с аналогичными соотношениями в рассмотренных выше примерах указывает на их внешнее различие. Это различие обусловлено тем, что в задаче распределения ресурсов фиксированным является конечное состояние управляемого процесса. Поэтому принцип Беллмана применяется не к последующим, а к начальным этапам управления, и начальное соотношение имеет вид:

$$\Phi_1(\xi) = \max_{u_1}[f_1(u_1, \xi)].$$

Важно еще раз подчеркнуть, что сформулированный выше принцип оптимальности применим только для управления объектами, у которых выбор оптимального управления не зависит от предыстории управляемого процесса, т.е. от того, каким путем система пришла в текущее состояние. Именно это обстоятельство делает возможным решение задачи.

Говоря о динамическом программировании как о методе решения оптимизационных задач, необходимо отметить и его слабые стороны. Так, в предложенной схеме решения задачи (54)-(55) существенно используется тот факт, что система ограничений содержит только одно неравенство. Как следствие, состояние можно задать одним числом — нераспределенным ресурсом ξ. При наличии нескольких ограничений состояние управляемого объекта на каждом шаге характеризуется уже набором параметров $\xi_1, \xi_2, ..., \xi_m$, и табулировать значения функций $\Phi_k(\xi_1, \xi_2, ..., \xi_m)$ необходимо для многократно большего количества точек. Последнее обстоятельство может сделать применение метода динамического программирования нерациональным или просто невозможным. Данную проблему метода его основоположник Р. Беллман эффектно назвал «проклятием многомерности». В настоящее

время разработаны определенные пути преодоления указанной проблемы.

Рассмотрим вопросы применения методов динамического программирования в конкретных моделях исследования операций.

2.3.5. Задача о найме работников

Одним из важнейших классов задач, для которых применение динамического программирования оказывается плодотворным, являются задачи последовательного принятия решений. В качестве примера опишем так называемую задачу о найме работников (задачу об использовании рабочей силы).

В данной задаче рассматривается некоторый экономический объект (фирма, магазин, завод и т.п.), функционирующий в течение конечного числа периодов, обозначаемых номерами $k = 1,2,...,n$. Каждый период k характеризуется нормативной потребностью в определенном количестве работников m_k.

Тот же объем работ может быть выполнен другим количеством сотрудников ξ_k, что, однако, влечет дополнительные затраты либо за счет нерационального использования рабочей силы, либо ввиду повышения оплаты за интенсивный труд. Размеры этих дополнительных издержек описываются функциями $g_k(\xi_k - m_k)$, где $(\xi_k - m_k)$ — отклонение фактической численности работающих ξ_k от планово необходимой m_k, причем $g_k(0) = 0$. Управленческое решение на шаге k заключается в выборе величины изменения численности сотрудников u_k (т.е. числа увольняемых или дополнительно набираемых сотрудников), чем также однозначно определяет количество работающих в течение следующего периода: $\xi_{k+1} = \xi_k + u_k$. Затраты по изменению количества работников (найму или увольнению) при переходе от периода k к периоду k+1 задаются функцией $h_k(u_k)$, где также $h_k(0) = 0$.

Тогда суммарные издержки, вызванные принятым на шаге k решением, определяются значением функции:

$$f_k(u_k, \xi_k) = g_k(\xi_k - m_k) + h_k(u_k), \quad u_k \geq -\xi_k .$$

План задачи (стратегия управления) $\mathbf{U} = (u_1, u_1, ..., u_{n-1}, 0)$ заключается в выборе поэтапных изменений количества работников, а его суммарная эффективность описывается аддитивной функцией:

$$F(\mathbf{U}) = \sum_{k=1}^{n} f_k(u_k, \xi_k), \tag{65}$$

где $\xi_k = \xi_{k-1} + u_{k-1}, k > 1$.

На основе сформулированной модели ставится задача минимизации целевой функции (издержек) (65). Добавим, что постановка задачи не будет корректной, если не задать начальное условие на количество работников. Существуют две модификации данной задачи, определяемые типом начального условия: в первом случае задается исходное значение на первом этапе m_1, а во втором — требуемое количество в n-м периоде m_n.

Рассмотрим первый случай. Поскольку фиксированным является начальное количество работников и, напротив, ничего не известно о том, каким это количество должно быть на последнем этапе, то рассмотрение процесса принятия решений удобнее начать с конца. Оптимальное управление на последнем этапе n по условию равно $u_n^* = \mathfrak{u}_n(\xi) = 0$, поэтому минимальные издержки здесь полностью определяются количеством работников в последнем периоде:

$$\Phi_n(\xi) = f_n(0, \xi) = g_n(\xi - m_n), \quad \xi \geq 0. \tag{66}$$

Для остальных предшествующих шагов основное рекуррентное соотношение имеет вид:

$$\Phi_k(\xi) = \min_{u_k \geq -\xi} [f_k(u_k, \xi) + \Phi_{k+1}(\xi + u_k)], \quad k = 1,2,...,n-1, \tag{67}$$

где $\Phi_k(\xi)$ — минимальные затраты с k-го по n-й периоды, в предположении, что количество работников в k-й период равно ξ. Точки $\mathfrak{u}_n(\xi)$, в которых достигаются минимумы (67), определяют условное оптимальное управление на каждом шаге.

Последовательно определяя $\mathfrak{u}_n(\xi)$ и дойдя до этапа 1, мы сможем найти безусловное оптимальное управление u_1^* из того условия, что на начало первого периода численность работников должна быть $\xi_1^* = m_1$, а именно:

$$u_1^* = \psi_1(\xi_1^*) = \psi_1(m_1).$$

Остальные компоненты оптимального плана u_k^* и состояния ξ_k^*, образующие оптимальную траекторию, последовательно находятся по рекуррентным формулам:

$$\xi_{k+1}^* = \xi_k^* + u_k^*, \quad u_{k+1}^* = \psi_{k+1}(\xi_{k+1}^*), \quad k = 1,2,...,n-1,$$

после чего не составляет труда вычислить оптимальное значение целевой функции (65).

Остановимся теперь на втором случае, когда задано желаемое количество работников на последнем периоде $\xi_n^* = m_n$. Очевидно, что в данной ситуации следует рассмотреть процесс принятия решений от начала к концу. Условное оптимальное управление на первом шаге $\psi_1(\xi)$ будет найдено в процессе вычисления функции:

$$\Phi_1(\xi) = \min_{u_1 \geq -\xi} f_1(u_1, \xi),$$

где состояние $\xi \geq 0$ является возможным количеством работников на начальном шаге. Соответственно, основное рекуррентное соотношение выразит минимальные издержки вплоть до k-того периода через таковые для предыдущих периодов (с первого по (k–1)-й) при условии, что численность работников в k-й период будет равна ξ:

$$\Phi_k(\xi) = \min_{u_k \geq -\xi} [f_k(u_k, \xi) + \Phi_{k-1}(\xi - u_k)], \quad k = 2,...,n.$$

Попутно будут найдены функции $\psi_k(\xi)$, $k = 2,...,n$, определяющие условные оптимальные управления. На последнем периоде, в силу начального условия, $\xi_n^* = m_n$. Отсюда путем последовательного решения рекуррентных уравнений могут быть найдены оптимальные численности работников ξ_k^* и безусловные оптимальные управления:

$$\xi_k^* + \psi_k(\xi_k^*) = \xi_{k+1}^*, \quad u_k^* = \psi_k(\xi_k^*), \quad k = 1,2,...,n-1.$$

В заключение, как и в первом случае, подсчитывается минимальная величина издержек.

Обобщая изложенные схемы решения, можно прийти к следующему выводу. Если задано начальное состояние управляемой системы, то задача решается в обратном направлении, а если конечное, то — в прямом. Наконец, если заданы как начальное, так и конечное состояния, то задача существенно усложняется. В качестве компромисса в этом

случае можно отказаться от оптимизации на первом или последнем этапе.

2.3.6. Динамические задачи управления запасами

Одной из наиболее известных сфер приложения методов динамического программирования является принятие решений при управлении запасами. При математическом описании процессов управления запасами часто приходится использовать скачкообразные, недифференцируемые и кусочно-непрерывные функции. Как правило, это определяется необходимостью учета эффектов концентрации, фиксированных затрат и платы за заказ, В связи с этим получаемые задачи с трудом поддаются аналитическому решению классическими методами, однако могут быть успешно решены с помощью аппарата динамического программирования. Рассмотрим достаточно типичную задачу, возникающую в процессе планирования деятельности системы снабжения, — так называемую динамическую задачу управления запасами.

Пусть имеется некоторая система снабжения (склад, оптовая база и т.п.), планирующая свою работу на n периодов. Ее деятельность сводится к обеспечению спроса конечных потребителей на некоторый продукт, для чего она осуществляет заказы производителю данного продукта. Спрос клиентов (конечных потребителей) в данной модели рассматривается как некоторая интегрированная величина, принимающая заданные значения для каждого из периодов, при этом спрос должен всегда удовлетворяться, т.е. не допускаются задолженности и отказы. Также предполагается, что заказ, посылаемый производителю, удовлетворяется им полностью, и временем между заказом и его выполнением можно пренебречь, т.е. рассматривается система с мгновенным выполнением заказа. Введем обозначения:

- y_k — остаток запаса после (k-1) -го периода;
- d_k — заранее известный суммарный спрос в k-м периоде;
- u_k — объем заказа (поставки от производителя) в k-м периоде;
- $c_k(u_k)$ — затраты на выполнение заказа объема u_k в k-м периоде;
- $s_k(\xi_k)$ — затраты на хранение запаса объема ξ_k в k-м периоде.

После получения поставки и удовлетворения спроса объем товара, подлежащего хранению в период k, составит $\xi_k = y_k + u_k - d_k$. Учитывая смысл параметра y_k, можно записать соотношение:

$$\xi_k = \xi_{k-1} + u_k - d_k, \quad k = 2,...,n. \tag{68}$$

Расходы на получение и хранение товара в период k описываются функцией:

$$f_k(u_k, \xi_k) = c_k(u_k) + s_k(\xi_k), \quad k = 1,2,...,n.$$

Планом задачи можно считать вектор $\mathbf{U} = (u_1, u_2, ..., u_n)$, компонентами которого являются последовательные заказы в течение рассматриваемого промежутка времени. Соотношение между запасами (68) в сочетании с некоторым начальным условием связывает состояния системы с выбранным планом и позволяет выразить суммарные расходы за все n периодов функционирования управляемой системы снабжения в форме аддитивной целевой функции:

$$F(\mathbf{U}) = \sum_{k=1}^{n} f_k(u_k, \xi_k). \tag{69}$$

Естественной в рамках сформулированной модели представляется задача нахождения последовательности оптимальных управлений (заказов) u_k^* и связанных с ними оптимальных состояний (запасов) ξ_k^*, которые обращают в минимум (69). В качестве начального условия используем требование о сохранении после завершения управления заданного количества товара y_{n+1}, а именно:

$$\xi_n^* = y_{n+1}. \tag{70}$$

При решении поставленной задачи методом динамического программирования в качестве функции состояния управляемой системы $\Phi_k(\xi)$ логично взять минимальный объем затрат, возникающих за первые k периодов при условии, что в k-й период имеется запас ξ. Тогда можно записать основное рекуррентное соотношение:

$$\Phi_k(\xi) = \min_{0 \le u_k \le \xi + d_k} [f_k(u_k, \xi) + \Phi_{k-1}(\xi - u_k + d_k)], \quad k = 2,...,n, \tag{71}$$

поскольку $y_k = \xi - u_k + d_k \ge 0$, и :

$$\Phi_1(\xi) = \min_{0 \le u_1 \le \xi + d_1} [c_1(u_1) + s_1(\xi)]. \tag{72}$$

Система рекуррентных соотношений (71)-(72) позволяет найти последовательность функций состояния $\Phi_1(\xi), \Phi_2(\xi), ..., \Phi_n(\xi)$ и условных оптимальных управлений $u_1(\xi), u_2(\xi), ..., u_n(\xi)$. На n-м шаге с помощью

начального условия (70) можно определить $u_n^* = u_n(y_{n+1})$. Остальные значения оптимальных управлений u_k^* определяются по формуле:

$$u_k^* = u_k\left(y_{n+1} + \sum_{j=k+1}^{n}\left(d_j - u_j^*\right)\right). \qquad (73)$$

Особый интерес представляет частный случай задачи (68)–(69), при котором предполагается, что функции затрат на пополнение запаса $c_k(u_k)$ являются вогнутыми по u_k, а функции затрат на хранение $s_k(\xi_k)$ являются линейными относительно объема хранимого запаса, т.е. $s_k(\xi_k) = \sigma_k \xi_k$, где σ_k – некоторая постоянная. Параллельно заметим, что обе предпосылки являются достаточно реалистичными.

Обозначим функцию затрат в течение k-го периода через:

$$f_k(u_k, \xi_k) = c_k(u_k) + \sigma_k \xi_k, \qquad (74)$$

или, что то же самое:

$$f_k(u_k, y_k) = c_k(u_k) + \sigma_k y_{k+1}. \qquad (75)$$

В силу сделанных допущений относительно функций $c_k(u_k)$ и $s_k(\xi_k)$ все функции затрат $f_k(u_k, \xi_k)$ являются вогнутыми[1] (как суммы вогнутой и линейной функций). Данное свойство значительно упрощает процесс решения, так как минимум вогнутых функций достигается в крайних точках множества, на котором отыскивается минимум. Следовательно достаточно рассмотреть только эти две точки. С учетом сделанных обозначений задачу (68)-(69) можно записать в виде:

$$F(\mathbf{U}) = \sum_{k=1}^{n} f_k(u_k, y_{k+1}) \to min, \qquad (76)$$

при условиях:

$$u_k + y_k - y_{k+1} = d_k, \quad k = 1,2,...,n. \qquad (77)$$

Последовательность решения (76)–(77) может быть следующей. Так как ищется минимум суммы вогнутых функций $f_k(u_k, y_k)$, то решение будет достигаться на одной из крайних точек множества, определяемого

---

[1] Функция $f(\cdot)$ называется вогнутой, если для любых двух значений ее аргумента $a \neq b$ выполняется неравенство $\omega \cdot f(a) + (1-\omega) \cdot f(b) \le f(\omega \cdot a + (1-\omega) \cdot b)$ при всех $0 \le \omega \le 1$

условиями (77). Общее число переменных u_k и y_k в системе уравнений (77) равно 2n. Однако, поскольку в ней только n уравнений, в оптимальном плане будет не более n ненулевых компонент, причем для каждого периода k значения x_k и y_k не могут равняться нулю одновременно ввиду необходимости удовлетворения спроса либо за счет заказа, либо за счет запаса. Формально это утверждение можно представить в виде условий (дополняющей нежесткости):

$$y_k^* u_k^* = 0, \quad k = 1,2,...,n, \tag{78}$$

где

$$\begin{cases} y_k^* = 0 \Rightarrow u_k^* > 0, \\ u_k^* = 0 \Rightarrow y_k^* > 0. \end{cases} \tag{79}$$

Условия (78)–(79) означают, что при оптимальном управлении заказ поставщику на новую партию не должен поступать, если в начале периода имеется ненулевой запас, или размер заказа должен равняться величине спроса за целое число периодов. Отсюда следует, что запас на конец последнего периода должен равняться нулю: $y_{n+1}^* = 0$. Последнее позволяет решать задачу в прямом направлении, применяя рекуррентное соотношение:

$$\Phi_k(\xi) = \min_{u_k} \left[f_k(u_k, \xi) + \Phi_{k-1}(\xi - u_k + d_k) \right], \tag{80}$$

где $\xi = y_{k+1} = u_k + y_k - d_k$.

Учитывая (78)–(79) и вогнутость $f_k(u_k, \xi)$, заключаем, что минимум (80) достигается в одной из крайних точек $u_k = 0$ или $u_k = \xi + d_k$, поэтому:

$$\Phi_k(\xi) = \min \begin{Bmatrix} f_k(\xi + d_k, \xi) + \Phi_{k-1}(0) \\ f_k(0, \xi) + \Phi_{k-1}(\xi + d_k) \end{Bmatrix},$$

тогда для предыдущего периода функция состояния может быть выражена как:

$$\Phi_{k-1}(\xi + d_k) = \min \begin{Bmatrix} f_k(\xi + d_k + d_{k-1}, \xi + d_k) + \Phi_{k-2}(0) \\ f_k(0, \xi + d_k) + \Phi_{k-2}(\xi + d_k + d_{k-1}) \end{Bmatrix},$$

на основании чего, в общем виде, получаем модифицированную формулу рекуррентного соотношения:

$$\Phi_k(\xi)=\min_{1\le i\le k}\left[f_i\left(\xi+\sum_{j=i}^{k}d_j,\xi+\sum_{j=i+1}^{k}d_j\right)+\sum_{q=i+1}^{k}f_q\left(0,\xi+\sum_{j=i+1}^{k}d_j\right)+\Phi_{i-1}(0)\right].$$

Дальнейшие конкретизирующие предположения о виде функций $f_k(u_k, y_{k+1})$ позволяют получить еще более компактные формы рекуррентных соотношений, однако эти вопросы носят уже достаточно частный характер и, поэтому, рассматриваться здесь не будут. Отметим лишь, что приведенные здесь преобразования неплохо иллюстрируют методы, применяемые в динамическом программировании, а также те свойства задач, которые открывают возможности для их эффективного использования.

2.4. Марковские модели

2.4.1. Случайные процессы и марковское свойство

Пусть имеется некоторая физическая система, состояние которой меняется с течением времени. Если изменение состояния системы во времени зависит от вмешательства случая, то говорят, что в системе протекает **случайный процесс**. Случайными являются процессы:

– функционирования ЭВМ;

– наведения на цель управляемой ракеты или космического летательного аппарата;

– обслуживание клиентов парикмахерской или ремонтной мастерской;

– выполнение плана снабжения группы предприятий и т. д.

Протекание каждого из этих процессов зависит от случайных, заранее непредсказуемых факторов, таких как:

– темп и характер команд и запросов поступающих в ЭВМ от оператора; выход ЭВМ из строя;

– возмущения и помехи в системе управления ракетой;

– поток заявок (требований на обслуживание), со стороны клиентов парикмахерской или ремонтной мастерской;

– перебои в выполнении плана снабжения и т. д.

Случайный процесс, протекающий в системе, называется **марковским процессом** или «процессом без последействия», если он обладает следующим (марковским) свойством.

Для каждого момента времени t_0 вероятность любого состояния системы в будущем (при $t > t_0$) зависит только от ее состояния в текущий

момент (при $t = t_0$) и не зависит от того, каким образом система пришла в это состояние, т.е. от того, как развивался процесс в прошлом.

Другими словами, будущее развитие марковского случайного процессеа зависит только от его текущего состояния и не зависит от его «предыстории», т.е. от того, каким образом процесс попал в это состояние и как долго в нем пребывает.

Пример I. Пусть система представляет собой техническое устройство, которое уже эксплуатировалось некоторое время и пришло в состояние, характеризующееся определенной степенью износа. Нас интересует, как будет работать система в будущем. По крайней мере в первом приближении ясно, что характеристики системы в будущем (частота отказов, потребность в ремонте) зависят в основном от степени износа в настоящий момент и не зависят от того, каким образом устройство достигло своего настоящего состояния.

Часто встречаются случайные процессы, которые, с той или другой степенью приближения, можно считать марковскими.

Случайный процесс называется **процессом с дискретными состояниями**, если множество возможных состояний системы $\varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots$ конечно или счетно, т.е. все состояния системы можно перечислить (перенумеровать). С точки зрения стороннего наблюдателя процесс с дискретными состояниями заключается в том, что время от времени система скачком (мгновенно) переходит (перескакивает) из одного состояния в другое.

Рассмотрим пример. Техническое устройство состоит из двух узлов I и II, каждый из которых может в процессе эксплуатации выйти из строя. Возможны следующие состояния устройства: S_1 - оба узла работают; S_2 - узел I вышел из строя, узел II работает; S_3 - узел II вышел из строя, узел I работает; S_4 - оба узла вышли из строя. Всего имеется четыре возможных состояния устройства, которые мы перенумеровали. Случайный процесс функционирования устройства, состоит в том, что оно случайным образом, в заранее непредсказуемые моменты времени переходит из состояния в состояние. Перед нами — процесс с дискретными состояниями.

Кроме процессов с дискретными состояниями существуют **случайные процессы с непрерывными состояниями**, множество возможных состояний которых можно представить как отрезок числовой оси или область в вещественном пространстве большего числа измерений. Например, процесс изменения напряжения в осветительной сети представляет собой случайный процесс с непрерывными состояниями.

При анализе случайных процессов с дискретными состояниями удобно пользоваться геометрической схемой — так называемым графом состояний. Вершины графа состояний отображают возможные состояния системы, а ориентированные ребра (стрелки) — возможные переходы между ними.

Пусть имеется система S с пятью дискретными состояниями: S_1,S_2,...,S_5. Ее граф состояний может быть таким, как показано на рис.11.

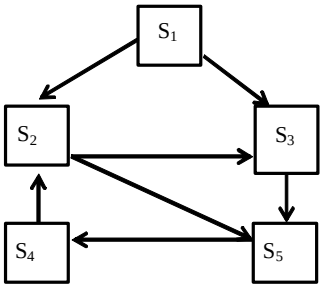

Рис. 11

Ребрами (стрелками) на графе состояний отмечаются только те переходы из состояния в состояние, которые могут произойти за один шаг процесса. Если, например, система может перейти за два шага из состояния S_1 в состояние S_5, через состояние S_2, то стрелками отмечаются только переходы $S_1 \rightarrow S_2$, и $S_2 \rightarrow S_5$ но не $S_1 \rightarrow S_5$.

Пример 2. Система S - автомашина, которая может находиться в одном из пяти возможных состояний: S_1 - исправна, работает; S_2 - неисправна, ожидает осмотра; S_3 - осматривается; S_4 - ремонтируется; S_5 - списана. Рис. 12 показывает граф состояний системы.

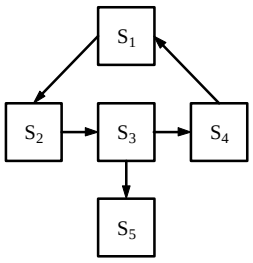

Рис. 12

Метод математического описания марковского случайного процесса в системе с дискретными состояниями зависит от того, в какие

моменты времени — заранее известные или случайные — могут происходить переходы из состояния в состояние.

Марковский случайный процесс с дискретными состояниями называется **процессом с дискретным временем**, или **цепью Маркова**, если переход из состояния в состояние возможен лишь в определенные, заранее фиксированные моменты времени: t_1, t_2, ... , а в промежутках между этими моментами система сохраняет свое состояние.

Марковский случайный процесс с дискретными состояниями называется **процессом с непрерывным временем**, или просто **марковским случайным процессом**, если переход из состояния в состояние возможен в любой (случайный) момент времени.

2.4.2. Цепи Маркова

Условно будем говорить о некоторой физической системе, шаг за шагом меняющей свое состояние. Будем считать, что имеется лишь конечное или счетной число различных состояний системы $\varepsilon_1, \varepsilon_2, \ldots$ Обозначим $\xi(n)$ состояние системы через n шагов. Предполагается, что цепочка последовательных переходов:

$$\xi(0) \to \xi(1) \to \ldots ,$$

зависит от вмешательства случая, причем соблюдается следующая закономерность: если на каком-либо шаге n система находится в состоянии ε_i , то, независимо от предшествующих обстоятельств, она на следующем шаге с вероятностью p_{ij} переходит в состояние ε_j :

$$p_{ij} = \mathbf{P}\left(\xi(n+1) = \varepsilon_j \,|\, \xi(n) = \varepsilon_i\right), \quad i, j = 1,2,\ldots$$

т.е. p_{ij} — условная вероятность перехода системы на следующем шаге в состояние ε_j, $j = 1,2,\ldots$ при условии, что текущее состояние системы есть ε_i .

Описанная выше модель марковского процесса называется однородной цепью Маркова, а вероятности p_{ij} называются переходными вероятностями этой цепи. Кроме них, еще задается начальное распределение вероятностей

$$p_i^0 = \mathbf{P}(\xi(0) = \varepsilon_i), \quad i = 1,2,\ldots .$$

Какова вероятность того, что через n шагов система будет находиться в состоянии ε_j . Обозначим эту вероятность $p_j(n)$:

$$p_j(n) = \mathbf{P}\left(\xi(n) = \varepsilon_j\right) .$$

Через n–1 шагов система обязательно будет находиться в одном из состояний ε_k, $k=1,2,\ldots$, причем в состоянии ε_k она будет с вероятностью, которую мы обозначили $p_k(n-1)$. При условии, что через n–1 шагов система будет находиться в состоянии ε_k, вероятность оказаться через n шагов в состоянии ε_j равна вероятности перехода из ε_k в ε_j т.е. p_{ij}. Используя формулу полной вероятности, получаем, что:

$$\mathbf{P}\left(\xi(n)=\varepsilon_j\right)=\sum_k \mathbf{P}\left(\xi(n)=\varepsilon_j \mid \xi(n-1)=\varepsilon_k\right)\cdot \mathbf{P}\left(\xi(n-1)=\varepsilon_k\right).$$

Это дает следующие рекуррентные соотношения для вероятностей $p_j(n)$, $j=1,2,\ldots$:

$$p_j(0)=p_j^0,\quad p_j(n)=\sum_k p_k(n-1)p_{kj},\quad n=1,2,\ldots$$

Если в начальный момент времени система находится в определенном состоянии ε_i, то начальное распределение вероятностей таково, что $p_i^0=1$, $p_k^0=0$, для $k\neq i$, а вероятность $p_j(n)$ совпадает с вероятностью $p_{ij}(n)$ того, что система за n шагов перейдет из состояния ε_i в состояние ε_j. При начальном распределении вероятностей вида $p_i^0=1$, $p_k^0=0$, для $k\neq i$ последняя формула дает следующее соотношение между вероятностями перехода $p_{ij}(n)$, $i,j=1,2,\ldots$:

$$p_{ij}(n)=\sum_k p_{ik}(n-1)p_{kj},\quad n=1,2,\ldots \qquad (81)$$

где $p_{ii}(0)=1$ и $p_{ij}(0)=0$ при $j\neq i$. Для процессов с конечным числом m состояний удобно ввести $m\times m$ матрицу $P(n)=\|p_{ij}(n)\|$. В соответствии с формулой (81), условием, наложенным на начальное распределение вероятностей и правилом умножения матриц

$$P(0)=I,\quad P(1)=P,\quad P(2)=P(1)\cdot P=P^2,\quad \mathbf{L}$$

где I — единичная матрица, и

$$P = \left\| p_{ij} \right\| = \begin{Vmatrix} p_{11} & p_{12} & \mathbf{K} & p_{1j} & \mathbf{K} & p_{1m} \\ p_{21} & p_{22} & \mathbf{K} & p_{2j} & \mathbf{K} & p_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{i1} & p_{i2} & \mathbf{K} & p_{ij} & \mathbf{K} & p_{im} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{m1} & p_{m2} & \mathbf{K} & p_{mj} & \mathbf{K} & p_{mm} \end{Vmatrix} \qquad (82)$$

матрица переходных вероятностей, которая была определена ранее. Видно, что имеет место следующее равенство:

$$P(n) = P^n, \quad n = 1,2,...$$

Рассмотрим однородную цепь Маркова с конечным числом состояний более подробно. Из предыдущих выкладок очевидно, что вероятностное описание ее поведения определяется матрицей переходных вероятностей P.

Некоторые из ее элементов p_{ij} могут быть равны нулю: это означает, что за один шаг переход системы из состояния ε_i в состояние ε_j невозможен. По главной диагонали матрицы P стоят вероятности p_{ii} того, что за один шаг система не выйдет из состояния ε_i, а останется в нем. Для наглядного представления структуры матрицы переходных вероятностей часто бывает удобно пользоваться графом состояний (см. выше), на котором у стрелок дополнительно проставлены соответствующие переходные вероятности. Такой граф называется **размеченным графом состояний**.

Предположим теперь дополнительно, что структура матрицы переходных вероятностей (или, что то же, графа состояний) такова, что каждое из состояний марковской цепи достижимо из любого другого состояния, в том смысле, что существует такое число N, что за N шагов система с положительной вероятностью может перейти из каждого состояния ε_i в любое другое состояние ε_j, т.е.:

$$\min_{i,j} p_{ij}(N) = \delta > 0. \qquad (83)$$

При этом дополнительном условии справедливы следующие две теоремы, описывающие основные свойства марковских цепей.

ТЕОРЕМА 1. Вероятности $p_j(n)$, с которыми через n шагов система будет находиться в соответствующих состояниях ε_j, $j = 1,2,...,m$ сходятся при $n \to \infty$ к предельным значениям:

$$p_j^* = \lim_{n\to\infty} p_j(n),$$

причем финальные вероятности p_j^*, $j=1,2,...,m$ не зависят от начального распределения и, более того, скорость сходимости к финальным вероятностям оценивается как:

$$\max_i \left| p_{ij}(n) - p_j^* \right| \le C\,exp(-Dn),$$

где $C,D>0$ — некоторые постоянные .

Финальные вероятности, если они существуют, должны удовлетворять системе линейных уравнений:

$$p_j^* = \sum_{i=1}^{m} p_i^* p_{ij}, \quad j=1,2,...,m. \tag{84}$$

Эти уравнения получаются, если в формуле (81), согласно которой:

$$p_j(n) = \sum_{i=1}^{m} p_i(n-1) p_{ij},$$

перейти к пределу при $n\to\infty$.

ТЕОРЕМА 2. При условии (83) финальные вероятности являются единственным решением системы линейных уравнений (84), которое удовлетворяет дополнительному требованию $p_j^* \ge 0, \sum_j p_j^* = 1$ и образует стационарное распределение вероятностей.

Таким образом, цепи Маркова обладают замечательным свойством (весьма быстрой) сходимости к стационарному распределению вероятностей состояний, которое не зависит от причин вызвавших его искажение, а определяется только свойствами самой цепи, заданными матрицей переходных вероятностей.

Наряду с однородными иногда рассматриваются также неоднородные цепи Маркова. У неоднородной цепи вероятности перехода зависят от номера шага. Основная формула (81) в этом случае будет иметь:

$$p_i(k) = \sum_{j=1}^{n} p_j(k-1) p_{ji}^{(k)} (i=1,\mathbf{K},n),$$

где $p_{ji}^{(k)}$ — переходные вероятности на k-том шаге.

2.4.3. Марковские процессы с дискретными состояниями и непрерывным временем. Дифференциальные уравнения Колмогорова

Весьма часто встречаются ситуации, когда переход системы из состояния в состояние происходят в заранее нефиксированные, случайные моменты времени. Это означает, что переход может произойти в любой момент. Например, выход из строя (отказ) любого элемента аппаратуры может произойти в любой момент времени; момент окончания ремонта (восстановления) этого элемента также может случиться раньше или позже и т.д. Для описания процессов такого рода часто и с успехом используется модель марковского случайного процесса с дискретными состояниями и непрерывным временем, который иногда также называют **непрерывной цепью Маркова**.

Так же как и при рассмотрении марковских цепей, будем условно говорить о некоторой физической системе, с течением времени t меняющей свое состояние. По-прежнему будем считать, что множество различных состояний системы конечно или счетно.

Обозначим $\xi(t)$ состояние системы в момент t. Будем предполагать, что течение процесса зависит от вмешательства случая, причем соблюдается следующая закономерность: если в какой-либо момент времени s система находится в состоянии ε_i, то, независимо от своего поведения до момента s, она через время t с вероятностью $p_{ij}(t)$ переходит в состояние ε_j:

$$p_{ij}(t) = \mathbf{P}\left(\xi(s+t) = \varepsilon_j \,|\, \xi(s) = \varepsilon_i\right), \quad i, j = 1,2,...$$

Описанная модель называется однородным марковским случайным процессом, а вероятности p_{ij} называются переходными вероятностями этого процесса. Кроме них, дополнительно должно быть задано начальное распределение вероятностей:

$$p_i^0 = \mathbf{P}(\xi(0) = \varepsilon_i), \quad i = 1,2,...$$

Обозначим $p_j(t)$ вероятность того, что через время t система будет находиться в состоянии ε_j:

$$p_j(t) = \mathbf{P}\left(\xi(t) = \varepsilon_j\right), \quad j = 1,2,...$$

Легко установить (аналогично тому, как это сделано в предыдущем разделе), что:

$$p_j(t) = \sum_i p_i^0 p_{ij}(t),$$

$$p_{ij}(s+t) = \sum_k p_{ik}(s) p_{kj}(t), \quad i, j = 1,2,...$$

при любых s и t, где:

$$p_{ij}(0) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

Предположим, что в некоторый момент времени $t=t_0$ марковский процесс $\xi(t)$ находится в определенном состоянии ε. Спустя некоторое случайное время τ процесс перейдет в некоторое другое состояние. Какова вероятность того, что изменение состояния произойдет не раньше, чем через время t?

Эта вероятность $\mathbf{P}(\tau > t)$ зависит, конечно, от t. Рассматривая ее как функцию от t, положим:

$$\varphi(t) = \mathbf{P}(\tau > t), \quad t \geq 0.$$

Пусть в момент t_0 процесс находится в некотором состоянии ε. При условии, что $\tau > s$, в момент $t_0 + s$ процесс будет находиться в том же состоянии ε, и дальнейшее его поведение подчиняется тем же закономерностям, что и при $s = 0$. В частности, при условии, что $\tau > s$, вероятность события $\tau > s + t$ равна $\varphi(t)$:

$$\mathbf{P}(\tau > s + t | \tau > s) = \varphi(t).$$

Используя формулу полной вероятности, получим, что:

$$\mathbf{P}(\tau > s + t) = \mathbf{P}(\tau > s + t | \tau > s) \cdot \mathbf{P}(\tau > s) = \varphi(t) \cdot \varphi(s).$$

Это дает следующее соотношение для функции φ:

$$\varphi(t + s) = \varphi(s)\varphi(t),$$

при любых s и t, откуда

$$ln\,\varphi(t + s) = ln\,\varphi(s) + ln\,\varphi(t).$$

Видно, что функция $ln\,\varphi(t)$ является линейной:

$$ln\,\varphi(t) = -\lambda t, \quad t \geq 0,$$

где λ — некоторая неотрицательная постоянная. Таким образом, искомая вероятность есть:

$$\mathbf{P}(\tau > t) = e^{-\lambda t}, \quad t \geq 0 \cdot$$

Параметр λ называется **плотностью перехода** из соответствующего состояния ε. При $\lambda = 0$ процесс навсегда остается в состоянии ε. При $\lambda > 0$ вероятность того, что состояние процесса изменится за малый промежуток времени Δt, есть:

$$1 - \varphi(\Delta t) = \lambda \Delta t + o(\Delta t),$$

где $o(\Delta t)$ означает величину более высокого порядка малости, чем Δt, т.е. $\lim\limits_{\Delta t \to 0} \dfrac{o(\Delta t)}{\Delta t} = 0$.

Распределение вероятностей случайной величины τ — времени до момента перехода в новое состояние — таково, что:

$$\mathbf{P}(t_1 \le \tau \le t_2) = \varphi(t_1) - \varphi(t_2) = \int_{t_1}^{t_2} \lambda e^{-\lambda t} dt,$$

при любых $t_1, t_2 \ge 0$. Оно называется **показательным** или **экспоненциальным** распределением. Видно, что плотность такого распределения вероятностей имеет вид:

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

При этом среднее значение (математическое ожидание) $\mathbf{E}_\tau$ случайной величины τ — среднее время ожидания перемены состояния — определяется формулой:

$$\mathbf{E}_\tau = \int_0^\infty t p_\tau(t) dt = \frac{1}{\lambda}.$$

Рассмотрим теперь марковский процесс с переходными вероятностями $p_{ij}(t)$, $i, j = 1,2,...$ Будем считать, что переходные вероятности $p_{ij}(t)$ удовлетворяют следующим условиям:

$$1 - p_{ii}(\Delta t) = \lambda_i(\Delta t) + o(\Delta t),$$

$$p_{ij}(\Delta t) = \lambda_{ij}(\Delta t) + o(\Delta t), \quad j \ne i; i, j = 1,2,...$$

иначе говоря,

$$\lambda_{ij} = \lim_{\Delta t \to 0} \frac{p_{ij}(\Delta t)}{\Delta t}.$$

Здесь λ_i — плотность перехода из состояния i (см. выше), а λ_{ij} — так называемые плотности перехода из состояния ε_i в состояние ε_j. Положим:

$$\lambda_{ii} = -\lambda_i, \quad i = 1,2,...$$

Справедлива следующая основная теорема, которую мы приведем без доказательства.

ТЕОРЕМА 1. Для марковского процесса с конечным числом состояний переходные вероятности $p_{ij}(t)$ удовлетворяют дифференциальным уравнениям Колмогорова:

$$\frac{dp_{ij}}{dt} = \sum_k \lambda_{ik} p_{kj}(t), \quad i, j = 1,2,...,m,$$

с начальными условиями:

$$p_{ij}(0) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

Следует отметить, что **дифференциальные уравнения Колмогорова** для переходных вероятностей имеют место не только в случае конечного числа состояний, но, при некоторых дополнительных ограничениях, и в случае счетного числа состояний $\varepsilon_1, \varepsilon_2, ...$

Если предварительно построить размеченный граф состояний системы, то уравнения Колмогорова можно будет выписать формально. Напомним, что в размеченном графе состояний вершины представляют состояния, а ребра — возможные переходы между состояниями за один шаг процесса. Дополнительно, рядом с каждой стрелкой, на графе должны быть указаны плотности вероятностей перехода λ_{ij} для всех пар состояний $\varepsilon_i, \varepsilon_j$.

В левой части уравнений Колмогорова стоят производные по времени (т.е. скорость изменения) вероятности состояния, а правая часть каждого уравнения содержит столько членов, сколько стрелок связано с соответствующим состоянием. Если стрелка направлена из состояния, соответствующий член имеет знак «минус», если к состоянию — знак «плюс». Каждый член равен произведению плотности вероятности перехода, соответствующей данной стрелке, умноженной на вероятность того состояния, из которого исходит стрелка.

Напомним, что решение уравнений Колмогорова с начальными условиями:

$$p_{ij}(0)=\begin{cases}1, & i=j\\ 0, & i\neq j,\end{cases}$$

имеет смысл вероятности пребывания системы в состоянии ε_j спустя время t после начала процесса, при условии что в начальный момент система находилась в состоянии ε_i с вероятностью 1.

Рассмотрим марковский процесс с конечным числом состояний $\varepsilon_1,\varepsilon_2,...,\varepsilon_m$, каждое из которых достижимо из любого другого состояния. Основное свойство таких марковских процессов выражается следующей теоремой 2, которая показывает, что в определенном смысле начальное состояние не имеет значения.

ТЕОРЕМА 2. Вероятности $p_j(t)$, с которыми через время t система может находиться в соответствующих состояниях ε_j, $j=1,2,...,m$, при $t\to\infty$ имеют предельные значения:

$$p_j^*=\lim_{t\to\infty} p_j(t),\quad j=1,2,...,m.$$

Указанные финальные вероятности не зависят от начального распределения и, более того,

$$\max_j\left|p_{ij}(t)-p_j^*\right|\le Ce^{-Dt},$$

где C,D — некоторые положительные постоянные.

Замечание 1. Для случая непрерывного времени предполагавшееся ранее для цепей Маркова дополнительное условие (83) выполняется автоматически; именно:

$$\min_{i,j} p_{ij}(t)=\delta(t)>0,$$

при любом $t>0$.

Замечание 2. Из уравнений Колмогорова вытекает следующая система уравнений относительно финальных вероятностей:

$$\sum_k p_k^*\lambda_{kj}=0,\quad j=1,2,... \tag{85}$$

Наряду с рассмотренной здесь моделью однородного марковского процесса, в которой плотности вероятности перехода постоянны, иногда рассматривается случай, когда эти плотности зависят от времени, т.е.

$\lambda_{ij} = \lambda_{ij}(t)$. В последнем случае марковский процесс называется **неоднородным**.

В заключение рассмотрим пример.

Рассмотрим систему, которая должна обслуживать поступающие в нее требования. Будем считать, что в системе имеется m обслуживающих приборов и очередное требование поступает на обслуживание, если хотя бы один из них свободен. В противном случае, если все приборы заняты, требование получает отказ и уходит из системы. Предположим, что промежутки времени между моментами поступления требований независимы и распределены по показательному закону с параметром λ, а время обслуживания требований (любым из приборов) распределено по показательному закону с параметром μ.

Рассмотрим состояния $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_m$, где ε_k — означает, что занято ровно k приборов. В частности, ε_0 означает, что система свободна, а ε_m — что система полностью занята. Переход системы из состояния в состояние с течением времени t представляет собой марковский процесс , для которого плотности перехода имеют вид:

$$\lambda_{0j} = \begin{cases} \lambda & j = 1, \\ 0 & j \neq 1; \end{cases} \quad \lambda_{kj} = \begin{cases} k\mu & \mathrm{j} = \mathrm{k}-1, \\ \lambda & j = k+1, \\ 0 & j \neq \mathrm{k}-1, \mathrm{k}+1. \end{cases}$$

Действительно, переход из ε_k в ε_j происходит при поступлении очередного требования, что происходит за время Δt с вероятностью $\lambda\Delta t + o(\Delta t)$. Вероятность того, что ни один из k занятых приборов не освободится за время Δt есть $(1 - \mu\Delta t - o(\Delta t))^{\mathrm{k}}$, поскольку приборы обслуживают требования независимо друг от друга, и вероятность освобождения одной из линий, т.е. перехода из состояния ε_k в состояние ε_j, $\mathrm{j} = \mathrm{k}-1$, есть $1 - [1 - \mu\Delta t - o(\Delta t)]^{\mathrm{k}} = \mathrm{k}\mu\Delta t + o(\Delta t)$. Вероятность других изменений в системе за промежуток времени Δt есть $o(\Delta t)$.

Как следует из теоремы 2, вероятности состояний экспоненциально быстро стремятся к финальным вероятностям, которые могут быть найдены из соотношений (85). Последние дают следующую систему уравнений для определения неизвестных финальных вероятностей:

$$\begin{gathered} -\lambda \mathrm{p}_0^* + \mu \mathrm{p}_1^* = 0, \\ \lambda \mathrm{p}_{\mathrm{k}-1}^* - (\lambda + k\mu)\,\mathrm{p}_{\mathrm{k}}^* + (\mathrm{k}+1)\,\mu \mathrm{p}_{\mathrm{k}+1}^* = 0, \quad 1 \leq \mathrm{k} \leq \mathrm{m}, \\ \lambda \mathrm{p}_{\mathrm{m}-1}^* - \mathrm{m}\mu \mathrm{p}_{\mathrm{m}}^* = 0. \end{gathered}$$

Решая эти уравнения, получаем, что:

$$p_k^* = \frac{\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k}{\sum_{q=0}^{m}\frac{1}{q!}\left(\frac{\lambda}{\mu}\right)^q}, \quad k = 0,1,\ldots,m. \qquad (86)$$

Найденные выражения для финальных вероятностей называются формулами Эрланга.

2.5. Модели теории игр

2.5.1. Предмет теории игр. Основные понятия

При решении ряда практических задач исследования операций (в области экономики, военного дела и т.д.) приходится анализировать ситуации, в которых сталкиваются две (или более) враждующие стороны, преследующие различные цели, причем результат любого мероприятия для каждой из сторон зависит от того, какой образ действий выберет противник. Такие ситуации мы будем называть **конфликтными ситуациями**.

Примеры конфликтных ситуаций. Любая ситуация, складывающаяся в ходе военных действий, принадлежит к конфликтным — каждое решение в этой области должно приниматься с учетом сознательного противодействия разумного противника. К той же категории принадлежат и ситуации, возникающие при выборе системы вооружения, способов его боевого применения и вообще при планировании боевых операций. Ряд ситуаций в экономике (особенно при наличии конкуренции) также можно отнести к конфликтным; в роли борющихся сторон здесь выступают торговые фирмы, промышленные предприятия, тресты, монополии и т.д. Конфликтные ситуации возникают также в судопроизводстве, спорте и в других областях человеческой деятельности.

Необходимость анализировать такие ситуации вызвала к жизни **теорию игр** — математическую теорию конфликтных ситуаций.

Почти каждая непосредственно взятая из практики конфликтная ситуация достаточно сложна, и ее анализ затруднен наличием многих относящихся к ней факторов. Чтобы сделать возможным анализ ситуации, в частности — с целью отделения существенных факторов от

несущественных, необходимо построить ее абстрактную математическую модель. Такую модель мы будем называть **игрой.**

Человечество издавна пользуется формализованными моделями конфликтов – «играми» в буквальном смысле слова (шашки, шахматы, карточные игры и т. д.). Все эти игры носят характер соревнования, происходящего по известным правилам и заканчивающегося «победой» (выигрышем) того или другого игрока.

Поэтому реальные игры представляют собой наиболее удобный материал для иллюстрации и усвоения основных понятий теории игр. Это отражается и на терминологии, используемой в теории игр: стороны, участвующие в конфликте, условно именуются «игроками», исход конфликта — «выигрышем» и т.д.

В игре могут сталкиваться интересы двух или более противников; в первом случае игра называется «парной», во втором – «множественной». Участники множественной игры могут образовывать коалиции (постоянные или временные). Множественная игра с двумя постоянными коалициями обращается в парную. Мы ограничимся рассмотрением только парных игр.

Пусть имеется парная игра, в которой участвуют два игрока А и В с противоположными интересами. Под «игрой» будем понимать мероприятие, состоящее из ряда действий или «ходов» игроков А и В. Чтобы игра могла быть подвергнута математическому анализу, должны быть четко сформулированы ее правила, т.е. система условий, регламентирующая:

- возможные варианты действий игроков;
- объем доступной игрокам информации о поведении друг друга;
- результат (исход) игры, к которому приводит каждая та или иная последовательность ходов.

Результат игры (выигрыш или проигрыш), вообще говоря, не всегда имеет непосредственное количественное выражение. Однако обычно можно, хотя бы условно, выразить его числом, например, в шахматной игре выигрышу можно приписать значение 1, проигрышу – минус 1, ничьей – 0.

Игра называется игрой с нулевой суммой, если один игрок выигрывает ровно столько, сколько проигрывает другой, т.е. сумма выигрышей сторон равна нулю. В игре с нулевой суммой интересы противников прямо противоположны. Здесь мы будем рассматривать только такие игры, которые называют также антагонистическими.

Обозначим через a выигрыш игрока **А**, а b – выигрыш игрока **В** в игре с нулевой суммой. Так как a = – b, то при анализе такой игры нет

необходимости рассматривать оба эти числа – достаточно рассматривать выигрыш одного из игроков.

Развитие игры во времени мы будем представлять состоящим из ряда последовательных этапов или **ходов**. Ходом в теории игр называется выбор и осуществление (игроком) одного из вариантов действий, предусмотренных правилами игры. Ходы могут быть **личными** и **случайными**. Личным ходом называется сознательный выбор игроком одного из возможных вариантов действий и его осуществление (например, любой ход в шахматной игре). Случайным ходом называется выбор из ряда возможностей, осуществляемый не решением игрока, а каким-либо механизмом случайного выбора (бросание монеты, выбор карты из перетасованной колоды и т. п.). Для каждого случайного хода должно быть определено, т.е. случайный ход задается, распределением вероятностей возможных исходов.

Некоторые игры состоят только из случайных ходов (так называемые «чисто азартные») или только из личных ходов (шахматы, шашки). Большинство карточных игр содержит как личные, так и случайные ходы.

Поскольку выигрыш в «чисто азартных» играх от поведения игроков не зависит, постольку теория такими играми не занимается. Ее цель — оптимизация поведения игроков в игре, т. е. рекомендовать им определенные, в определенном смысле оптимальные «стратегии».

Стратегией игрока называется правило (или — правила, если их несколько) выбора варианта действий при каждом ходе игрока в зависимости от ситуации, сложившейся в процессе игры.

Понятие стратегии – одно из основных в теории игр; остановимся на нем несколько подробнее. Обычно, принимая участие в игре, игрок не следует каким-то жестким, фиксированным правилам: выбор (решение) при каждом ходе принимается им в ходе игры, в зависимости от сложившейся конкретной ситуации. Однако теоретически дело не изменится, если мы представим себе, что все эти решения приняты игроком заранее («если сложится такая-то ситуация, я поступлю так-то»). Если такая система решений будет принята, это будет означать, что игрок выбрал определенную стратегию. Теперь он может и не участвовать в игре лично, а заменить свое участие списком правил, которые за него будет применять незаинтересованное лицо (судья). Стратегия может быть также задана машине-автомату в виде программы; именно так играют в шахматы электронные вычислительные машины.

В зависимости от числа возможных стратегий игры делятся на «конечные» и «бесконечные». Игра называется **конечной**, если у каждого игрока имеется только конечное число стратегий, и

бесконечной, если хотя бы у одного из игроков имеется бесконечное число стратегий.

Как уже было сказано, одной из целей теории игр является выработка рекомендаций для разумного поведения игроков в конфликтной ситуации, т.е. определение «оптимальной стратегии» для каждого из них.

Оптимальной стратегией игрока называется такая стратегия, которая при многократном повторении игры обеспечивает данному игроку максимально возможный средний выигрыш (или, что то же, минимально возможный средний проигрыш). При этом в основе выбора оптимальной стратегии лежит допущение о том, что игроки одинаково разумны, и оба способны сделать все для того, чтобы помешать друг другу добиться своей цели.

В теории игр та или иная модель игры вырабатывается именно исходя из этого допущения. Поэтому просчеты и ошибки игроков, а также — элементы азарта и риска, неизбежные в каждой реальной конфликтной ситуации, в ней не учитываются.

Реальные конфликтные ситуации приводят к различным игровым математическим моделям. Наряду с парными антагонистическими играми, которые будут подробнее рассмотрены далее, в теории игр существуют модели:

- антагонистических позиционных игр с полной и неполной информацией (крестики-нолики, шашки, шахматы, домино и др.),
- неантагонистических биматричных игр (борьба за рынки, классическая дилемма узника, семейные споры и пр.),
- неантагонистические игры с оценкой оптимальности стратегий по Парето (т.е. по нескольким критериям), и др. Тем не менее, любая игровая модель, как и всякая математическая модель сложного явления, имеет свои ограничения. Сознавая эти ограничения и, поэтому, не придерживаясь слепо рекомендаций, полученных игровыми методами, можно разумно использовать математический аппарат теории игр для выработки искомых стратегий.

2.5.2. Платежная матрица

Рассмотрим игру, в которой участвуют два игрока, причем каждый из них имеет конечное число стратегий.

Обозначим для удобства одного из игроков через **A**, в другого — через **B**. Предположим, что игрок **A** имеет m стратегий $A_1,A_2,...,A_m$. а игрок **B** — n стратегий $B_1,B_2,...,B_n$.

Пусть игрок **A** выбрал стратегию A_i, а игрок **B** — стратегию B_k. Будем считать, что выбор игроками стратегий A_i и B_k однозначно определяет исход игры — выигрыш a_{ik} игрока **A** и выигрыш b_{ik} игрока **B**, причем эти выигрыши связаны равенством:

$$b_{ik} = -a_{ik}.$$

Последнее условие показывает, что в рассматриваемых обстоятельствах выигрыш одного из игроков равен выигрышу другого, взятому с противоположным знаком. Поэтому при анализе такой игры можно рассматривать выигрыши только одного из игроков. Пусть это будут, например, выигрыши игрока **A**.

Если нам известны значения a_{ik} выигрыша при каждой паре стратегий (в каждой ситуации) $\{A_i,B_k\}$, i=1,2,...,m, к=1,2,...,n, то их удобно записывать или в виде прямоугольной таблицы (матрицы), строки которой соответствуют стратегиям игрока **A**, а столбцы — стратегиям игрока **B**,

$$\mathbf{A} = \begin{Vmatrix} a_{11} & a_{12} & \mathbf{L} & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \mathbb{M} & & \mathbf{O} & \mathbb{M} \\ a_{m1} & a_{m2} & \mathbf{L} & a_{mn} \end{Vmatrix}$$

Полученная матрица имеет размер m×n и называется **матрицей игры** или **платежной матрицей**, отсюда еще одно название игры — матричная.

Рассматриваемую игру также называют игрой m×n или m×n – игрой. Матричные игры относятся к разряду антагонистических игр, т.е. игр, в которых интересы игроков прямо противоположны.

Заметим, что построение платежной матрицы, особенно для игр с большим числом стратегий, может само по себе представлять весьма непростую задачу. Например, для шахматной игры, которую, как известно, в принципе можно представить как матричную игру[1], число возможных стратегий столь велико, что построение платежной матрицы является пока неосуществимым, даже с использованием самых современных вычислительных машин.

---

[1] Шахматы относятся к т.н. позиционным играм, которые сводятся к матричным с помощью стандартного процесса нормализации.

Рассмотрим несколько примеров матричных игр и построим для них платежные матрицы.

Пример 1. Игра «прятки». Имеется два игрока **А** и **В**. Игрок **А** прячется, а **В** его ищет. В распоряжении **А** имеется два убежища (I и II), которые он может выбрать по своему усмотрению. Условия игры: **А** прячется, после чего его ищет **В**. Если **В** найдет **А**, то **А** платит ему штраф 10 руб.; иначе (когда **А** спрятался в другом убежище) **В** сам должен заплатить **А** ту же сумму. Требуется построить платежную матрицу.

Решение. Игра состоит всего из двух ходов, оба — личные. У игрока **А** две стратегии: A_1 – прятаться в убежище I, A_2 – прятаться в убежище II. У игрока **В** тоже две стратегии: B_1 – искать в убежище I, B_2 – искать в убежище II. Перед нами — игра 2x2. Ее матрица имеет вид:

$$\begin{Vmatrix} -10 & 10 \\ 10 & -10 \end{Vmatrix}$$

2.5.3. Равновесная ситуация

Пример 2. Два игрока **А** и **В**, не глядя друг на друга, кладут на стол по картонному кружку красного (1), зеленого (2) или синего (3) цвета, сравнивают цвета кружков и расплачиваются друг с другом так, как показано в матрице игры:

$$\begin{Vmatrix} -2 & 2 & -1 \\ 2 & 1 & 1 \\ 3 & -3 & 1 \end{Vmatrix}$$

(напомним, что у этой 3 x 3-матрицы строки соответствуют стратегиям игрока **А**, а столбцы — стратегиям игрока **В**).

Считая, что эта 3×3 — игра повторяется многократно, попробуем определить оптимальные стратегии каждого из игроков.

Начнем с последовательного анализа стратегий игрока **А**, не забывая о том, что, выбирая стратегию игрока **А**, необходимо принимать в расчет, что его противник **В** может ответить на нее той из своих стратегий, при которой выигрыш игрока **А** будет минимальным.

Так, на стратегию A_1 он ответит стратегией B_1 (минимальный выигрыш равен —2, что на самом деле означает проигрыш игрока **А**, равный 2), на стратегию A_2 — стратегией B_2 или B_3 (минимальный выигрыш игрока **А** равен 1), а на стратегию A_3 — стратегией B_2 (минимальный выигрыш игрока **А** равен —3).

Запишем эти минимальные выигрыши в правом дополнительном столбце платежной матрицы:

A_1	-2	2	-1	-2
A_2	2	1	1	1
A_3	3	-3	1	-3

Максимин (maxmin). Неудивительно, что игрок **А** останавливает свой выбор на стратегии A_2, при которой его минимальный выигрыш максимален (из трех чисел —2, 1 и —3 максимальным является 1), maxmin = 1.

Если игрок **А** будет придерживаться этой стратегии, то ему гарантирован выигрыш, не меньший 1, при любом поведении противника.

Аналогичные рассуждения можно провести и за игрока **В**. Так как игрок **В** заинтересован в том, чтобы обратить выигрыш игрока **А** в минимум, то ему нужно проанализировать каждую свою стратегию с точки зрения максимального выигрыша игрока **А**.

Выбирая свою стратегию, игрок **В** должен учитывать, что при этом стратегией его противника **А** может оказаться та, при которой выигрыш игрока **А** будет максимальным.

Так, на стратегию B_1 он ответит стратегией A_3 (максимальный выигрыш игрока **А** равен 3), на стратегию B_2 — стратегией A_1 (максимальный выигрыш игрока **А** равен 2), а на стратегию B_3 — стратегией A_2 или А3 (максимальный выигрыш игрока **А** равен 1).

Эти максимальные выигрыши записаны в нижней дополнительной строке платежной матрицы:

	B_1	B_2	B_3	
A_1	-2	2	-1	-2
A_2	2	1	1	1
A_3	3	-3	1	-3
	3	2	1	

Минимакс (minmax). Неудивительно, если игрок **В** остановит свой выбор на стратегии B_3, при которой максимальный выигрыш игрока **А** минимален (из трех чисел 3, 2 и 1 минимальным является 1), minmax = 1.

Если игрок **B** будет придерживаться этой стратегии, то при любом поведении противника он проиграет не больше 1.

Итак, в рассматриваемой игре числа maxmin и minmax совпали: maxmin = minmax = 1 (соответствующие элементы в платежной матрице выделены фоном):

	B_1	B_2	B_3
A_1	-2	2	-1
A_2	2	1	**1**
A_3	3	-3	1

Выделенные стратегии A_2 и B_3 являются оптимальными для игроков **A** и **B**,

$$A_2 = A_{opt} \quad B_3 = B_{opt},$$

в следующем смысле: при многократном повторении игры отказ от выбранной стратегии любого из игроков уменьшает его шансы на выигрыш (увеличивает шансы на проигрыш).

В самом деле, если игрок **A** не будет придерживаться стратегии A_{opt}, а выберет иную стратегию, например A_1, то вряд ли стоит рассчитывать на то, что игрок **B** этого не заметит. Конечно, заметит и не преминет воспользоваться своим наблюдением. Ясно, что в этом случае он отдаст предпочтение стратегии B_1. А на выбор A_3 игрок **B** ответит, например, так: B_2. В результате отказа от стратегии A_2 выигрыш игрока **A** уменьшится.

Если же от оптимальной стратегии отказывается игрок **B**, выбирая, например, стратегию B_1, то игрок **A** может ответить на это стратегией A_3 и тем самым увеличить свой выигрыш. В случае стратегии B_2 ответ игрока **A** — A_1.

Тем самым ситуация $\{A_2,B_3\}$ оказывается равновесной: при многократном повторении игры ни одному из игроков не выгодно отклоняться от этих стратегий.

Еще раз подчеркнем, что элементами матрицы игры являются числа, описывающие выигрыш игрока **A**. Более точно, выигрыш соответствует положительному элементу платежной матрицы, а отрицательный указывает на проигрыш игрока **A**. Матрица выплат

игроку **B** получается из матрицы игры заменой каждого ее элемента на противоположный.

Рассмотрим теперь произвольную матричную игру:

$$\mathbf{A} = \begin{Vmatrix} a_{11} & a_{12} & \mathbf{L} & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \mathbb{M} & & \mathbf{O} & \mathbb{M} \\ a_{m1} & a_{m2} & \mathbf{L} & a_{mn} \end{Vmatrix}$$

(строки заданной $m \times n$ матрицы соответствуют стратегиям игрока **A**, а столбцы — стратегиям игрока **B**) и опишем общий алгоритм, посредством которого можно определить, есть ли в этой игре ситуация равновесия или ее нет.

В теории игр предполагается, что оба игрока действуют разумно, т.е. стремятся к получению максимального выигрыша, считая, что соперник действует наилучшим (для него) образом.

Действия игрока **A**.

1-й шаг. В каждой строке матрицы **A** находится минимальный элемент:

$$\alpha_i = \min_k a_{ik}, \quad i = 1,2,...,m.$$

Полученные числа $\alpha_1,\alpha_2,...,\alpha_m$ приписываются к заданной таблице в виде правого добавочного столбца:

$$\begin{matrix} a_{11} & a_{12} & \mathbf{L} & a_{1n} & \alpha_1 \\ a_{21} & a_{22} & & a_{2n} & \alpha_2 \\ \mathbb{M} & & \mathbf{O} & & \mathbb{M} \\ a_{m1} & a_{m2} & \mathbf{L} & a_{mn} & \alpha_m . \end{matrix}$$

Выбирая стратегию A_i, игрок **A** должен рассчитывать на то, что при разумных действиях противника (игрока **B**) он выиграет не меньше чем α_i.

2-й шаг. Среди чисел $\alpha_1,\alpha_2,...,\alpha_m$ выбирается максимальное число:

$$\alpha = \max_i \alpha_i,$$

или подробнее:

$$\alpha = \max_i \min_k a_{ik}.$$

Специально отметим, что выбранное число является одним и: элементов заданной матрицы **А**.

Действуя наиболее осторожно и рассчитывая на наиболее разумное поведение противника, игрок **А** должен остановиться на той стратегии A_i, для которой число α_i является максимальным. Если игрок **А** будет придерживаться стратегии, которая выбрана описанным выше способом, то при любом поведении игрока **В** игроку **А** гарантирован выигрыш, не меньший α.

Число α называется **нижней ценой игры**.

Принцип построения стратегии игрока **А**, основанный на максимизации минимальных выигрышей, называется **принципом максимина**, а выбираемая в соответствии с этим принципом стратегия A_{i^*} — **максиминной** стратегией игрока **А**.

Действия игрока В.

1-й шаг. В каждом столбце матрицы **А** ищется максимальный элемент:

$$\beta_k = \min_i a_{ik}, \quad k = 1,2,...,n.$$

Полученные числа $\beta_1,\beta_2,...,\beta_n$ приписываются к заданной таблице в нижней добавочной строке:

$$\begin{array}{ccccc} a_{11} & a_{12} & \mathbf{L} & a_{1n} & \alpha_1 \\ a_{21} & a_{22} & & a_{2n} & \alpha_2 \\ \mathbf{M} & & \mathbf{O} & \mathbf{M} & \mathbf{M} \\ a_{m1} & a_{m2} & \mathbf{L} & a_{mn} & \alpha_m \\ \beta_1 & \beta_2 & \mathbf{L} & \beta_n & \end{array}.$$

Выбирая стратегию $B_к$, игрок **В** должен рассчитывать на то, что в результате разумных действий противника (игрока **А**) он проиграет не больше, чем β_k

2-й шаг. Среди чисел $\beta_1,\beta_2,...,\beta_n$ выбирается минимальное число:

$$\beta = \min_k \beta_k,$$

или подробнее:

$$\beta = \min_k \max_i a_{ik}.$$

Выбранное число β также является одним из элементов заданной матрицы **А**.

Действуя наиболее осторожно и рассчитывая на наиболее разумное поведение противника, игрок **B** должен остановиться на той стратегии B_k, для которой число β_k является минимальным. Если игрок **B** будет придерживаться выбранной таким образом стратегии, то при любом поведении игрока **A** игроку **B** гарантирован проигрыш, не больший β.

Число β называется **верхней ценой игры**.

Построение стратегии игрока **B**, основанной на минимизации максимальных потерь, называется **принципом минимакса**, а выбираемая в соответствии с этим принципом стратегия B_{k^*} — **минимаксной** стратегией игрока **B**.

Нижняя цена игры α и верхняя цена игры β всегда связаны неравенством:

$$\alpha \le \beta .$$

Реализация описанного алгоритма требует 2mn—1 сравнений элементов матрицы **A**:

$$(n - 1)\, m + m - 1 = mn - 1,$$

сравнений для определения α,

$$(m - 1)n + n - 1 = mn - 1,$$

сравнений для определения β и одно сравнение полученных чисел α и β.

Если

$$\alpha = \beta ,$$

или подробнее

$$\max_i \min_k a_{ik} = a_{i^o k^o} = \min_k \max_i a_{ik} ,$$

то ситуация $\{ A_{i^o}, B_{k^o} \}$ оказывается равновесной и ни один из игроков не заинтересован в том, чтобы ее нарушить (в этом нетрудно убедиться путем рассуждений, подобных проведенным при анализе игры в примере 2).

В том случае, когда нижняя цена игры равна верхней, их общее значение называется просто ценой игры и обозначается через ν.

Цена игры совпадает с элементом $a_{i^o k^o}$ матрицы игры **A**, расположенным на пересечении i^o—той строки (стратегия A_{i^o} игрока **A**) и $к^o$—того столбца (стратегия B_{k^o} игрока **B**), — минимальным в своей строке и максимальным в своем столбце.

Этот элемент называют **седловой точкой** матрицы **A** или точкой равновесия, а про игру говорят, что она имеет седловую точку.

Стратегии A_{i^o} и B_{k^o}, соответствующие седловой точке, называются оптимальными, а совокупность оптимальных ситуаций и цена игры — решением матричной игры с седловой точкой.

Замечание. Седловых точек в матричной игре может быть несколько, но всем им соответствует одно и то же значение цены игры.

Матричные игры с седловой точкой важны и интересны, однако более типичным является случай, когда применение описанного алгоритма приводит к неравенству:

$$\alpha < \beta .$$

Как показывает следующий пример, в этом случае предложенный выбор стратегий уже, вообще говоря, к равновесной ситуации не приводит и при многократном ее повторении у игроков вполне могут возникнуть мотивы к нарушению рекомендаций, основанных на описанной последовательности действий игроков **A** и **B**.

Пример 3. Рассмотрим 3 x 3-игру, заданную матрицей:

$$\begin{Vmatrix} 4 & -1 & -3 \\ -2 & 1 & 3 \\ 0 & 2 & -3 \end{Vmatrix}$$

Применим предложенный алгоритм. Находим нижнюю и верхнюю цену игры $\alpha = -2$ $\beta = 2$ и соответствующие им стратегии A_2 и B_2.

Нетрудно убедиться в том, что, пока игроки придерживаются этих стратегий, средний выигрыш при многократном повторении игры будет равен 1. Он больше нижней цены игры, но меньше верхней.

Однако если игроку **B** станет известно, что игрок **A** придерживается стратегии A_2, он немедленно ответит стратегией B_1 и сведет выигрыш **A** к проигрышу —2. В свою очередь, на стратегию B_1 у игрока **A** имеется ответная стратегия A_1, дающая ему выигрыш 4. Тем самым ситуация $\{ A_2 , B_2 \}$ равновесной не является.

Вернемся к примеру 1 с игрой «прятки». Предположим сначала, что играется только одна, единственная «партия». Тогда, очевидно, нет смысла говорить о преимуществах тех или других стратегий – каждый из, игроков может с равным основанием принять любую из них. Однако при многократном повторении игры положение меняется.

Действительно, допустим, что игрок **A** выбрал стратегию A_1 и придерживается ее. Тогда, уже по результатам первых нескольких партий, противник догадается о выборе **A**, начнет всегда искать **A** в убежище I и выигрывать. То же самое будет, если игрок **A** выберет стратегию A_2. Игроку **A** явно невыгодно придерживаться какой-то одной

стратегии; чтобы не оказаться в проигрыше, он должен чередовать их. Однако, если **А** будет чередовать убежища I и II в определенной последовательности (например, через одну партию), то **В** тоже догадается об этом и ответит наихудшим для **А** образом. Очевидно, надежным способом, гарантирующим **А** от верного проигрыша, будет такой его выбор в каждой партии, когда он сам не знает его наперед. Например, **А** может бросить монету, и, если выпадет герб, выбрать убежище I, а если решетка – убежище II.

Положение, в котором оказался игрок **А** (чтобы не проигрывать, выбирать убежище случайным образом), очевидно, присуще не только ему, но и его противнику **В**, к которому также относятся все вышеприведенные рассуждения. Оптимальной стратегией каждого оказывается «смешанная» стратегия, в которой две возможные стратегии игроков чередуются случайным образом.

Таким образом, путем интуитивных рассуждений можно прийти к одному из основных понятий теории игр — к понятию **смешанной стратегии** — стратегии, в которой отдельные «чистые» стратегии чередуются случайным образом с некоторыми вероятностями. В данном примере из соображений симметрии ясно, что стратегии A_1 и A_2 должны применяться с одинаковыми вероятностями; в более сложных примерах решение может быть не столь тривиальным.

2.5.4. Смешанные стратегии

В случае когда нижняя цена игры α и верхняя цена игры β не совпадают,

$$\alpha < \beta$$

игрок **А** может обеспечить себе выигрыш, не меньший α, а игрок **В** имеет возможность не дать ему выиграть больше, чем β.

Возникает вопрос: а как разделить между игроками разность β-α?

Предыдущие построения на этот вопрос ответа не дают — тесны рамки возможных действий игроков. Ясно, что механизм, обеспечивающий получение каждым из игроков как можно большей доли этой разности, следует искать в определенном расширении стратегических возможностей игроков.

Оказывается, что компромиссного распределения разности β-α между игроками и уверенного получения каждым игроком своей доли при многократном повторении игры можно достичь путем случайного применения ими своих первоначальных, чистых стратегий.

Такие действия:

– во-первых, обеспечивают наибольшую скрытность выбора стратегии (результат выбора не может стать известным противнику, поскольку он неизвестен самому игроку);

– во-вторых, при разумном построении механизма случайного выбора стратегий последние оказываются оптимальными.

Случайная величина, значениями которой являются стратегии игрока, называется его **смешанной стратегией**. Задание смешанной стратегии игрока, очевидно, состоит в указании тех вероятностей, с которыми выбираются его первоначальные чистые стратегии.

Рассмотрим произвольную $m \times n$-игру, заданную $m \times n$-матрицей

$$\mathbf{A} = \|a_{ik}\|$$

Так как игрок **A** имеет m чистых стратегий, то его смешанная стратегия может быть описана набором m неотрицательных чисел:

$$p_1 \geq 0, p_2 \geq 0, \ldots, p_m \geq 0,$$

сумма которых равна 1,

$$\sum_{i=1}^{m} p_i = 1.$$

Смешанная стратегия второго игрока **B**, имеющего n чистых стратегий, описывается набором n неотрицательных чисел:

$$q_1 \geq 0, q_2 \geq 0, \ldots, q_n \geq 0,$$

сумма которых равна 1,

$$\sum_{k=1}^{n} q_k = 1.$$

Заметим, что каждая чистая стратегия является частным случаем смешанной стратегии: например, чистая стратегия A_i является смешанной стратегией, описываемой набором чисел p_1, p_2, …, p_m, в котором:

$$p_i = 1, \quad p_j = 0, j \neq i.$$

Подчеркнем, что для соблюдения секретности каждый из игроков применяет свои стратегии независимо от другого игрока.

Таким образом, задав два набора:

$$P = (p_1, p_2, \ldots, p_m), \quad Q = (q_1, q_2, \ldots, q_n),$$

мы оказываемся в ситуации в смешанных стратегиях.

В этом случае каждая обычная ситуация (в чистых стратегиях) $\{A_i, B_k\}$ по определению является случайным событием и ввиду независимости

P и Q реализуется с вероятностью $p_i q_k$. Поскольку в этой ситуации игрок **A** получает выигрыш a_{ik}, то математическое ожидание выигрыша в условиях ситуации в смешанных стратегиях {P,Q} равно:

$$\mathbf{E}(\mathbf{A},P,Q)=\sum_{i=1}^{m}\sum_{k=1}^{n}a_{ik}p_i q_k,$$

где $\mathbf{E}(\cdot)$ — оператор математического ожидания. Это число и принимается за средний выигрыш игрока **A** в смешанных стратегиях {P,Q}.

Стратегии:

$$P^o=\left(p_1^o,p_2^o,\ldots,p_m^o\right),\quad Q^o=\left(q_1^o,q_2^o,\ldots,q_n^o\right),$$

называются **оптимальными смешанными стратегиями** игроков **A** и **B** соответственно, если выполнено следующее соотношение:

$$\mathbf{E}\left(\mathbf{A},P,Q^o\right)\le\mathbf{E}\left(\mathbf{A},P^o,Q^o\right)\le\mathbf{E}\left(\mathbf{A},P^o,Q\right).$$

Последнее равносильно тому, что:

$$\max_P\min_Q\mathbf{E}(\mathbf{A},P,Q)=\mathbf{E}\left(\mathbf{A},P^o,Q^o\right)=\min_Q\max_P\mathbf{E}(\mathbf{A},P,Q).$$

Величина:

$$v=\mathbf{E}\left(\mathbf{A},P^o,Q^o\right),$$

определяемая последней формулой, называется **ценой игры**.

Набор $\left(P^o,Q^o,v\right)$, состоящий из оптимальных смешанных стратегий игроков **A** и **B** и цены игры, называется **решением матричной игры**. Естественно, возникают два ключевых вопроса*:*

1) какие матричные игры имеют решение в смешанных стратегиях?

2) как находить решение матричной игры, если оно существует?

Ответы на эти вопросы дают следующие две теоремы.

ТЕОРЕМА 1 (Основная теорема теории матричных игр, Дж. фон Нейман).

Для матричной игры с любой матрицей A величины:

$$\max_P\min_Q\mathbf{E}(\mathbf{A},P,Q),\quad\min_Q\max_P\mathbf{E}(\mathbf{A},P,Q),$$

существуют и равны между собой.

Более того, существует хотя бы одна смешанная стратегия {P°,Q°}, для которой выполняется соотношение:

$$\max_{P}\min_{Q}\mathbf{E}(\mathbf{A},P,Q)=\min_{Q}\max_{P}\mathbf{E}(\mathbf{A},P,Q).$$

Теорема 1, таким образом, утверждает, что любая матричная игра имеет решение в смешанных стратегиях.

ТЕОРЕМА 2 (Основные свойства оптимальных смешанных стратегий).

Пусть $P^o=\left(p_1^o,p_2^o,...,p_m^o\right)$, $Q^o=\left(q_1^o,q_2^o,...,q_n^o\right)$ — оптимальные смешанные стратегии и ν — цена игры.

Оптимальная смешанная стратегия P° игрока **А** смешивается только из тех чистых стратегий A_r, r=1,2,…,m (т.е. только те вероятности p_r, r=1,2,…,m, могут быть отличны от нуля), для которых:

$$\sum_{k=1}^{n}a_{ik}q_k^o=\nu.$$

Аналогично, только те вероятности q_k, к=1,2,...,n, могут быть отличны от нуля, для которых:

$$\sum_{i=1}^{m}a_{ik}p_i^o=\nu.$$

Имеют место соотношения:

$$v=\min_{1\le k\le n}\sum_{i=1}^{m}a_{ik}p_i^o=\max_{P}\min_{1\le k\le n}\sum_{i=1}^{m}a_{ik}p_i=\min_{Q}\max_{1\le i\le m}\sum_{k=1}^{n}a_{ik}q_k=\max_{1\le i\le m}\sum_{k=1}^{n}a_{ik}q_k^o=v.$$

В этой последовательности равенств, по существу, и лежат истоки, питающие методы построения решений матричных игр.

2.5.5. Упрощение игр

В принципе, как будет показано ниже, решение любой матричной игры сводится к решению стандартной задачи линейного программирования. При этом требуемый объем вычислений напрямую зависит от числа чистых стратегий игроков — растет с его увеличением и, значит, с увеличением размеров матрицы игры. Поэтому любые приемы предварительного анализа игры, позволяющие уменьшать размеры ее платежной матрицы или еще как-то упростить эту матрицу, не нанося ущерба решению, играют на практике весьма важную роль.

Правило доминирования. В целом ряде случаев анализ платежной матрицы обнаруживает, что некоторые чистые стратегии не

могут внести никакого вклада в искомые оптимальные смешанные стратегии. Отбрасывание подобных стратегий позволяет заменить первоначальную платежную матрицу на матрицу меньших размеров. Опишем одну из таких возможностей более подробно.

Сравнение строк и столбцов матрицы.

Будем говорить, что i-я строка матрицы **A**:

$$a_{i1} \quad a_{i2} \quad \mathbf{L} \quad a_{in},$$

не больше j-й строки этой матрицы:

$$a_{j1} \quad a_{j2} \quad \mathbf{L} \quad a_{jn},$$

если одновременно выполнены следующие n неравенств:

$$a_{i1} \le a_{j1}, \quad a_{i2} \le a_{j2}, \quad \mathbf{L}, \quad a_{in} \le a_{jn}.$$

При этом говорят также, что j-я строка доминирует i-ю строку или что стратегия A_j игрока **A** доминирует стратегию A_i.

Заметим, что игрок **A** поступит разумно, если будет избегать стратегий, которым в матрице игры отвечают доминируемые строки.

Если в матрице **A** одна из строк (j-я) доминирует другую строку (i-ю), то число строк в матрице **A** можно уменьшить путем отбрасывания доминируемой i–той строки.

Далее, будем говорить, что k-й столбец матрицы **A**:

$$\begin{matrix} a_{1k} \\ a_{2k} \\ \mathbf{M} \\ a_{mk} \end{matrix},$$

не меньше q–того столбца этой матрицы

$$\begin{matrix} a_{1q} \\ a_{2q} \\ \mathbf{M} \\ a_{mq} \end{matrix},$$

если одновременно выполнены следующие m неравенств:

$$a_{1k} \ge a_{1q}, \quad a_{2k} \le a_{2q}, \quad \mathbf{L}, \quad a_{mk} \ge a_{mq}.$$

При этом говорят также, что q–тый столбец доминирует k–тый столбец или что стратегия B_q игрока **B** доминирует стратегию B_k.

Заметим, что игрок **B** поступит разумно, если будет избегать стратегий, которым в матрице игры отвечают доминируемые столбцы.

Если в матрице **A** один из столбцов (q-й) доминирует другой столбец (k–тый), то число столбцов в матрице **A** можно уменьшить путем отбрасывания доминируемого столбца k.

Важное замечание. Оптимальные смешанные стратегии в игре с матрицей, полученной усечением исходной за счет доминируемых строк и столбцов, дадут оптимальное решение в исходной игре: доминируемые чистые стратегии игроков в смешении не участвуют — соответствующие им вероятности следует взять равными нулю.

При отбрасывании доминируемых строк и столбцов некоторые из оптимальных стратегий могут быть потеряны. Однако цена игры не изменится, и по усеченной матрице может быть найдена хотя бы одна пара оптимальных смешанных стратегий.

Аффинное правило. При поиске решения матричных игр часто оказывается полезным следующее свойство.

Оптимальные стратегии у матричных игр с платежными матрицами **A** и **C**, элементы которых связаны равенствами:

$$c_{ik} = \lambda a_{ik} + \mu \quad i = 1,2,\ldots,m; \quad k = 1,2,\ldots,n,$$

где $\lambda > 0$, а μ произвольно, имеют одинаковые равновесные ситуации (либо в чистых, либо в смешанных стратегиях), а их цены удовлетворяют следующему условию:

$$v_C = \lambda v_A + \mu .$$

Основные этапы поиска решения матричной игры, таким образом будут следующие:

1. Проверка наличия (или отсутствия) равновесия в чистых стратегиях; при наличии равновесной ситуации указываются соответствующие оптимальные стратегии игроков и цена игры.

2. Поиск доминирующих стратегий; в случае успеха — отбрасывание доминируемых строк и столбцов в исходной матрице игры.

3. Замена игры на ее смешанное расширение и отыскание смешанных стратегий и цены игры.

2.5.6. Решение игр $m \times n$

В общем случае, при больших m и n, решение игры $m \times n$ представляет собой довольно трудоемкую задачу, но принципиальных трудностей оно не содержит. Покажем, что решение любой конечной игры $m \times n$ сводится к решению уже известной нам задачи линейного программирования.

Рассмотрим игру $m \times n$. Заданы m стратегий $A_1, A_2, \ldots, A_m$ игрока **A**, и n стратегий $B_1, B_2 \ldots, B_n$ игрока **B** и платежная матрица игры $\|a_{ij}\|$, $i = 1,2,\ldots,m; j = 1,2,\ldots,n$.

Требуется найти решение игры — две оптимальные смешанные стратегии игроков **A** и **B** — $S_A^* = (p_1, p_2, \ldots p_m)$ и $S_B^* = (q_1, q_2, \ldots q_n)$, где:

$$p_1, p_2, \ldots, p_m \geq 0; \quad \sum_{i=1}^{m} p_i = 1,$$

$$q_1, q_2, \ldots, q_n \geq 0; \quad \sum_{j=1}^{n} q_j = 1.$$

– вероятности использования чистых стратегий; причем некоторые из них, соответствующие неактивным стратегиям, могут быть равными нулю.

Найдем, на первом этапе, оптимальную стратегию игрока **A** (стратегию S_A^*), обеспечивающую выигрыш, не меньший цены игры v, при любом поведении противника **B**, и выигрыш, равный v, при оптимальном поведении **B** (при стратегии S_B^*).

Цена игры v нам пока неизвестна. Не нарушая общности, можно предположить ее равной некоторому положительному числу $v > 0$. Действительно, для того чтобы выполнялось условие $v > 0$, достаточно, чтобы все элементы матрицы $\|a_{ij}\|$ были неотрицательными. Этого всегда можно добиться, прибавляя ко всем элементам матрицы $\|a_{ij}\|$ одну и ту же достаточно большую положительную величину M, при этом цена игры увеличится на M, а решение не изменится. Итак, будем считать $v > 0$.

Предположим, что **A** применяем свою оптимальную стратегию S_A^*, а его противник **B** — одну из своих чистых стратегий B_j, $j = 1,2,\ldots,n$. Тогда средний выигрыш **A** будет равен:

$$r_j = \sum_{i=1}^{m} p_i a_{ij}, \quad j = 1,2,\ldots,n.$$

Оптимальная стратегия S_A^* обладает тем свойством, что при любом поведении противника обеспечивает **A** выигрыш, не меньший, чем цена игры v. Следовательно, любой из средних выигрышей r_j не может быть меньше v. Получаем отсюда ряд условий:

$$\sum_{i=1}^{m} p_i a_{ij} \geq v, \quad j = 1,2,...,n \,. \tag{87}$$

Введем новые переменные $x_i = p_i / v, j = 1,2,...,n$ (делить на v можно, поскольку $v > 0$). В новых переменных условия—неравенства (87) запишутся в виде:

$$\sum_{i=1}^{m} x_i a_{ij} \geq 1, \quad j = 1,2,...,n \,, \tag{88}$$

где x_1, x_2, ..., x_m – новые неотрицательные переменные. По определению x_i и ввиду того, что $p_1 + p_2 + ... + p_m = 1$, новые переменные должны также удовлетворять условию:

$$L(x_1, x_2, ..., x_m) = \sum_{i=1}^{m} x_i = \frac{1}{v} \,. \tag{89}$$

Для того, чтобы сделать свой гарантированный выигрыш максимально возможным, игрок **A** должен стремиться к максимизации цены игры v, когда правая (а значит — и левая) часть (89) минимальна. Таким образом, задача решения игры свелась к следующей математической задаче.

Определить неотрицательные значения переменных x_1, x_2, ..., x_m так, чтобы они удовлетворяли линейным ограничениям (88) и при этом их линейная функция (89) обращалась в минимум. Перед нами – типичная задача линейного программирования.

Таким образом, решая задачу линейного программирования, мы можем найти оптимальную стратегию S_A^* игрока **A**.

Найдем теперь оптимальную стратегию S_B^* игрока **B**. Все будет аналогично решению игры для игрока **A**, с той разницей, что игрок **B** стремится минимизировать свой проигрыш, а значит, не минимизировать, а максимизировать величину $1/v$.

Таким образом, мы свели решение конечной игры $m \times n$ к решению пары задач линейного программирования. Попутно возможно более детально проанализировать свойства ЗЛП, возникающих при решении матричных игр[1].

---

[1] Заметим, что вышеприведенные выкладки существенно опираются на теорему существования решений матричных игр в смешанных стратегиях Дж фон Неймана. Поэтому никаких дополнительных соображений по поводу существования решения игр m n из них получить нельзя.

Итак, задача о нахождении оптимальной стратегии игрока **А** сведена к задаче линейного программирования на минимум целевой функции (89) с условиями—неравенствами (88).

Всегда ли существует ее решение? Как нам уже известно, возможны два случая, когда задача линейного программирования не имеет решений:

1) ограничения (равенства или неравенства) не имеют неотрицательных решений и допустимые решения не существуют;

2) допустимые решения существуют, но среди них нет оптимального, так как подлежащая минимизации функция не ограничена снизу.

Нетрудно убедиться, что допустимое решение ЗЛП в нашем случае всегда существует. Действительно, сделаем элементы платежной матрицы строго положительными, прибавив к каждому из них достаточно большое число М и обозначим наименьший элемент новой матрицы $\|a_{ij}\|$ через μ:

$$\mu = \min_{ij} a_{ij}.$$

Положим теперь $x_1 = 1/\mu$, $x_i = 0$, $i = 2,3,...,m$. Можно видеть, что в этом случае переменные $x_1, x_2, ..., x_m$ представляет собой допустимое решение ЗЛП, т.к. все они неотрицательны и в совокупности удовлетворяют условиям (88).

Теперь убедимся, что линейная функция (89) ограничена снизу. Действительно, все $x_1, x_2, ..., x_m$ неотрицательны, а коэффициенты при них в выражении (89) положительны. Следовательно, функция L в формуле (89) тоже неотрицательна, а значит — она ограничена снизу (нулем) и решение задачи линейного программирования (и игры $m \times n$) существует.

2.6. Модели массового обслуживания

2.6.1. Задачи теории массового обслуживания

Часто приходится сталкиваться с анализом работы своеобразных систем, называемых **системами массового обслуживания** (СМО). Примерами таких систем могут служить: телефонные станции, ремонтные мастерские, билетные кассы, справочные бюро, магазины, парикмахерские и т.п.

Каждая СМО состоит из какого-то числа обслуживающих единиц, которые мы будем называть каналами обслуживания. Каналами обслуживания являются линии связи, рабочие точки, приборы, железнодорожные пути, лифты, автомашины и т. д.

Системы массового обслуживания могут быть одноканальными или многоканальными.

Каждая СМО предназначена для обслуживания (выполнения) заявок (требований на обслуживание), которые поступают в СМО, вообще говоря, в случайные моменты времени. Обслуживание поступившей заявки продолжается в течение некоторого времени, после чего канал освобождается и готов к принятию следующей заявки. Случайный характер потока заявок приводит к тому, что в какие-то промежутки времени на входе СМО скапливается излишне большое число заявок (они либо образуют очередь, либо покидают СМО необслуженными); в другие же периоды СМО будет работать с недогрузкой или вообще простаивать.

Для описания процесса поступления в систему требований на обслуживание и процесса обслуживания требований вводится понятие **потока событий** или просто потока. Потоком называют последовательность событий. Соответственно, временная последовательность событий поступления требований называется потоком требований, временная последовательность событий окончания обслуживания — потоком обслуживания и т.д.

Математическое описание потока характеризует закономерности, которым подчиняются моменты поступления требований на обслуживание.

Поток называется **регулярным**, если интервал времени между событиями в потоке один и тот же. Поток называется случайным, если его события происходят в случайные моменты времени. Случайный поток может быть описан как случайный вектор, компоненты которого суть интервалы времени между событиями потока. Тогда случайный поток задается совместным распределением этих величин.

Важное значение имеют потоки, удовлетворяющие дополнительным условиям стационарности, ординарности и отсутствия последействия.

Случайный поток называется **стационарным**, если вероятность появления любого числа n событий на интервале времени (t,t+T) не зависит от его начала t на временной оси.

Поток событий называется **ординарным**, если вероятность появления двух или более событий в течении элементарного интервала времени Δt есть величина бесконечно малая по сравнению с вероятностью появления одного события на этом интервале, т.е.

$$\lim_{\Delta t \to 0} \mathbf{P}(\mathrm{n}, \Delta t) = 0,$$

при n=2,3,…, где $\mathbf{P}(n,\Delta t)$ — вероятность того, что n событий потока происходят в течение интервала времени Δt .

Поток событий называется **потоком без последствия**, если для любых непересекающихся интервалов времени число событий, попадающих на один из них, не зависит от числа событий попадающих на другой.

Поток, обладающий свойствами стационарности, ординарности и отсутствия последействия называется **пуассоновским потоком**. Доказано, что случайная величина интервала времени T между двумя последовательными событиями пуассоновского потока распределена по показательному закону:

$$f(T)=\lambda e^{-\lambda T},$$

с параметром λ, называемым плотностью потока. Математическое ожидание и дисперсия такой случайной величины, как легко проверить, равны:

$$\mathbf{E}(T)=\frac{1}{\lambda},\quad \mathbf{D}(T)=\frac{1}{\lambda^2}.$$

Каждая система массового обслуживания, в зависимости от числа каналов и их производительности, а также от характера потока заявок обладает определенной пропускной способностью, позволяющей ей более или менее успешно справляться с потоком заявок. Предмет теории массового обслуживания – установление зависимости между характером потока заявок, числом каналов, их производительностью, правилами работы СМО и успешностью (эффективностью) обслуживания.

В качестве характеристик эффективности обслуживания, в зависимости от условий задачи и целей исследования, могут применяться различные величины и функции, например:

– среднее число заявок, которое может обслужить СМО в единицу времени;

– средний процент заявок, получающих отказ и покидающих СМО необслуженными;

– вероятность того, что поступившая заявка немедленно будет принята на обслуживанию;

– среднее время ожидания в очереди;

– закон распределения времени ожидания;

– среднее количество заявок, находящихся в очереди;

– закон распределения числа заявок в очереди;

– средний доход, приносимый СМО в единицу времени и т. д.

Случайный характер потока заявок, а в общем случае — и длительности обслуживания приводит к тому, что в системе массового обслуживания будет происходить какой-то случайный процесс. Чтобы дать рекомендации по рациональной организации этого процесса и предъявить разумные требования к СМО, необходимо математически описать случайный процесс, протекающий в системе. Этим и занимается теория массового обслуживания. Близкие задачи возникают также при анализе надежности технических устройств.

Математический анализ работы СМО существенно упрощается, если протекающий в системе случайный процесс является марковским. Тогда удается построить достаточно простое математическое описание СМО и в явном виде выразить основные характеристики эффективности обслуживания в зависимости от параметров СМО и потока заявок.

Для того, чтобы процесс, протекающий в системе, был марковским, нужно, чтобы все потоки событий, переводящие систему из состояния в состояние, были пуассоновскими потоками без последействия. Для СМО потоки событий – это потоки заявок и потоки их «обслуживаня». Если эти потоки не являются пуассоновскими, математическое описание процессов, происходящих в СМО, становится намного более сложным, и требует более громоздкого математического аппарата, который позволяет получить явные аналитические формулы лишь в редких, простейших случаях.

Тем не менее, модели **марковских систем массового обслуживания** достаточно часто используется для приближенного описания СМО даже и в тех случаях, когда протекающий в СМО процесс отличается от марковского. С его помощью характеристики эффективности СМО оцениваются приближенно. Следует также отметить, что чем сложнее СМО, чем больше в ней каналов обслуживания, тем точнее оказываются приближенные формулы, полученные с помощью марковской теории. Кроме того, для принятия обоснованных решений по управлению работой СМО во многих случаях вовсе и не требуется точного знания всех ее характеристик – достаточно их приближенных оценок.

Работу СМО можно рассматривать как случайный процесс с дискретными состояниями и непрерывным временем: состояние СМО меняется скачком в моменты свершения событий прихода новых заявок, окончания обслуживания или ухода необслуженных заявок из-за превышения лимита времени пребывания в системе (см. ниже). При этом, подобно тому, как это происходит в марковских процессах с непрерывным временем, в СМО с течением времени может устанавливаться предельный стационарный режим, который описывается

вероятностными характеристиками уже независящими от времени. Именно эти предельные характеристики стационарного режима СМО и представляют основной интерес при их моделировании.

Несмотря на то, что получить аналитические модели переходного режима немарковских СМО в практически интересных случаях не удается, для стационарного режима такого рода аналитические результаты существуют, причем весьма общего характера. В качестве примера приведем здесь (без доказательства) замечательные формулы Литтла, играющие большую роль в теории массового обслуживания. Пусть λ – интенсивность потока заявок, поступающих в систему (величина, обратная среднему времени между моментами прихода заявок) и пусть в системе установился предельный стационарный режим, который характеризуется:

– средним числом $L_{ОЧ}$ и средним временем пребывания $t_{ОЧ}$ заявок в очереди;

– средним числом $L_{СИСТ}$ и средним временем пребывания $t_{СИСТ}$ заявок в системе в целом.

Тогда при любом распределении промежутков времени между моментами прихода заявок, любом распределении времени обслуживания и любой дисциплине обслуживания:

$$t_{ОЧ} = \frac{L_{ОЧ}}{\lambda}, \quad t_{СИСТ} = \frac{L_{СИСТ}}{\lambda}. \tag{90}$$

Далее в настоящем разделе излагаются элементы теории массового обслуживания, главным образом в той форме, которую они имеют в рамках марковской теории.

2.6.2. Классификация и основные характеристики систем массового обслуживания

Системы массового обслуживания могут быть двух типов.

1. **Системы с отказами**. В таких системах заявка, поступившая в момент, когда все каналы заняты, получает «отказ», покидает СМО и в дальнейшем процессе обслуживания не участвует.

2. **Системы с ожиданием** (с очередью). В таких системах заявка, поступившая в момент, когда все каналы заняты, становится в очередь и ожидает, пока не освободится один из каналов. Как только освободится канал, принимается к обслуживанию одна из заявок, стоящих в очереди.

Обслуживание в системе с ожиданием может быть «упорядоченным» (заявки обслуживаются в порядке поступления) и «неупорядоченным» (заявки обслуживаются в случайном порядке). Кроме того, в некоторых СМО применяется так называемое «обслуживание с приоритетом», когда некоторые заявки обслуживаются в первую очередь, предпочтительно перед другими.

Системы с очередью делятся на системы с неограниченным ожиданием и системы с ограниченным ожиданием.

В системах с неограниченным ожиданием каждая заявка, поступившая в момент, когда нет свободных каналов, становится в очередь и, не покидая системы, ждет освобождения канала, который примет ее к обслуживанию. Любая заявка, поступившая в СМО, рано или поздно обслуживается.

В системах с ограниченным ожиданием на пребывание заявки в очереди накладываются те или другие 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 систему), общего времени пребывания заявки в СМО и т.д.

В зависимости от типа СМО, применяться те или иные показатели ее эффективности.

Одной из важнейших характеристик эффективности СМО с отказами является так называемая **абсолютная пропускная способность** – среднее число заявок, которое может обслужить система за единицу времени. Наряду с этим, в качестве характеристики эффективности обслуживания часто рассматривается **относительная пропускная способность** СМО – отношение среднего числа заявок, обслуживаемых системой в единицу времени, к среднему числу поступающих за это время заявок.

Помимо абсолютной и относительной пропускной способностей, при анализе СМО с отказами нас могут, в зависимости от задачи исследования, интересовать и другие ее характеристики, например:

– среднее число занятых каналов;

– среднее относительное время простоя системы в целом или отдельного канала;

– и другие характеристики.

Перейдем к рассмотрению характеристик СМО с ожиданием.

Как абсолютная, так и относительная пропускная способность в качестве характеристик эффективности СМО с неограниченным ожиданием в значительной степени теряют смысл, так как каждая

поступившая заявка рано или поздно будет обслужена. Зато для подобных СМО весьма важными характеристиками будут:

– среднее число и среднее время пребывания заявок в очереди;

– среднее число и среднее время пребывания заявок в системе (в очереди и под обслуживанием);

– и др. характеристики ожидания.

Для СМО с ограниченным ожиданием интерес представляют обе группы характеристик: как абсолютная и относительная пропускная способности, так и характеристики ожидания.

Для анализа процесса, протекающего в СМО, необходимо знать основные параметры системы: число каналов n, интенсивность потока заявок λ, производительность каждого канала (среднее число заявок μ обслуживаемое каналом в единицу времени), а также — правила формирования и ограничения действующие на очереди. Эффективность работы СМО будет рассматриваться далее в зависимости именно от этих параметров.

Будем считать, что все потоки событий, переводящие СМО из состояния в состояние, являются пуассоновскими. Случаи, когда это не так, в дальнейшем каждый раз будут оговариваться отдельно.

2.6.3. Одноканальная СМО с отказами

Рассмотрим простейшую из всех систем одноканальную СМО с отказами.

Пусть система массового обслуживания состоит только из одного канала (n = 1) в нее поступает пуассоновский поток заявок с интенсивностью λ, которую в общем случае можно считать зависимой от времени.

Будем считать, что заявка, заставшая канал занятым, получает отказ и покидает систему. Если же в момент поступления заявки канал свободен, то она поступает на обслуживание и обслуживается в течение случайного промежутка времени $T_{об}$, распределенного по показательному закону с параметром μ:

$$f(T_{об}) = \mu e^{-\mu T_{об}},$$

т.е. поток «обслуживания» – пуассоновский поток с интенсивностью μ. Такой поток событий можно было бы наблюдать на выходе канала обслуживания, если бы он был непрерывно занят. При этом условии канал будет выдавать обслуженные заявки потоком с интенсивностью μ.

Требуется найти абсолютную (A) и относительную пропускную способность (q) СМО.

Рассмотрим единственный канал обслуживания как систему, которая может находиться в одном из двух состояний: S_0 – свободен, S_1 – занят.

Из состояния S_0 в S_1 систему, очевидно, переводит поток заявок с интенсивностью λ; из S_0 и S_1 – «поток обслуживания» с интенсивностью μ.

Обозначим вероятности состояний S_0 и S_1 как $p_0(t)$ и $p_1(t)$ соответственно. Очевидно, для любого момента t:

$$p_0(t)+p_1(t)=1. \tag{91}$$

Составим дифференциальные уравнения Колмогорова для вероятностей состояний:

$$\begin{aligned} \frac{dp_0}{dt} &= -\lambda p_0 + \mu p_1, \\ \frac{dp_1}{dt} &= -\mu p_1 + \lambda p_0. \end{aligned} \tag{92}$$

Из двух уравнений (92) одно является лишним, так как p_0 и p_1 связаны соотношением (91). Учитывая это, отбросим второе уравнение, а в первое подставим вместо p_1 его выражение $(1 - p_0)$:

$$\frac{dp_0}{dt} = -(\mu+\lambda)p_0 + \mu. \tag{93}$$

Это уравнение естественно решать при начальных условиях:

$$p_0(0) = 1, p_1(0) = 0,$$

т.е. считая, что в начальный момент канал свободен.

Линейное дифференциальное уравнение (93) с одной неизвестной функцией p_0 легко решается не только для простейшего потока заявок с постоянной интенсивностью λ = const, но и для случая, когда интенсивность этого потока меняется со временем $\lambda = \lambda(t)$. Не останавливаясь на последнем случае, приведем решение уравнения (93) для λ = const:

$$p_0 = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)t}. \tag{94}$$

Из последнего соотношения видно, что с увеличением t вероятность p_0 уменьшается от $p_0(0)=1$, при t = 0 до (в пределе при $t\to\infty$) $\frac{\mu}{\lambda+\mu}$. Для

одноканальной СМО с отказами вероятность p_o есть не что иное, как относительная пропускная способность q.

Предельное (при $t \to \infty$) значение относительной пропускной способности для СМО в установившемся состоянии будет равно:

$$q = \frac{\mu}{\lambda + \mu}. \tag{95}$$

Относительная и абсолютная пропускная способность связаны очевидным соотношением $A = \lambda q$. Поэтому в пределе, при $t \to \infty$, абсолютная пропускная способность также достигнет установившегося значения:

$$A = \frac{\lambda\mu}{\lambda + \mu}. \tag{96}$$

Зная относительную пропускную способность системы q (вероятность того, что пришедшая в момент t заявка будет обслужена), легко найти вероятность отказа $P_{отк}$ (среднюю долю не обслуженных заявок среди поданных). В пределе, при $t \to \infty$:

$$P_{отк} = 1 - q = 1 - \frac{\mu}{\lambda + \mu} = \frac{\lambda}{\lambda + \mu}. \tag{97}$$

Пример. Одноканальная СМО с отказами представляет собой одну телефонную линию. Заявка – вызов, пришедший в момент, когда линия занята, получает отказ. Интенсивность потока вызовов λ = 0,8 (вызовов в минуту). Средняя продолжительность разговора $\bar{t}_{об}$ = 1,5 мин. Все потоки событий – простейшие пуссоновские.

Определить предельные (при $t \to \infty$) (а) относительную пропускную способность q, (б) абсолютную пропускную способность A, (в) вероятность отказа $P_{отк}$.

Сравнить фактическую пропускную способность СМО с номинальной, которая была бы, если бы каждый разговор длился в точности 1.5 мин, и разговоры следовали бы один за другим без перерыва.

Решение. Определяем параметр μ потока обслуживания:

$$\mu = 1 / t_{ОБ} = 1 / 1.5 = 0.667.$$

По формуле (95) получаем относительную пропускную способность СМО:

$$q = \frac{0.667}{0.8 + 0.667} \approx 0.455.$$

Таким образом, в установившемся режиме система будет обслуживать около 45% поступающих вызовов.

По формуле (96) находим абсолютную пропускную способность:

$$A = \lambda q = 0.8 \cdot 0.455 \approx 0.364,$$

т. е. линия способна осуществить в среднем 0,364 разговора в минуту.

Вероятность отказа:

$$P_{ОТК} = 1 - q = 0.545,$$

значит около 55% поступивших вызовов будет получать отказ.

Номинальная пропускная способность канала:

$$A_{НОМ} = \frac{1}{t_{ОБ}} = 0.667,$$

разговора в минуту, что почти вдвое больше, чем фактическая пропускная способность, при случайном характере потока заявок и случайном времени обслуживания.

2.6.4. Многоканальная СМО с отказами

Рассмотрим n-канальную СМО с отказами. Пронумеруем состояния системы по числу занятых каналов или, что то же, по числу заявок, находящихся в системе. Состояния будут:

S_0 – все каналы свободны,

S_1 –занят ровно один канал, остальные свободны,

.....

S_k –заняты ровно k каналов, остальные свободны,

.....

S_n – заняты все n каналов.

Если система находится в состоянии S_k (занято k каналов) и пришла новая заявка, система переходит (перескакивает) в состояние S_{k+1} .

Пусть система находится в состоянии S_1 (занят один канал). Тогда, как только закончится обслуживание заявки, занимающей этот канал, система перейдет в состояние S_0. Поток событий, переводящий систему из состояния S_1 в состояние S_0, имеет интенсивность μ. Если занят не один, а два канала, то, т.к. каналы независимы, поток обслуживания, переводящий систему из состояния S_2 в состояние S_1, будет вдвое интенсивнее (2μ). Аналогично, если занято k каналов, – в k раз интенсивнее ($k\mu$).

Пользуясь общими правилами, можно составить уравнения Колмогорова для вероятностей состояний:

$$\frac{dp_0}{dt} = -\lambda p_0 + \mu p_1,$$
$$\frac{dp_k}{dt} = -(\lambda + k\mu)p_k + \lambda p_{k-1} + (k+1)\mu p_{k+1}, \quad k = 1,2,...,n. \tag{98}$$

Естественными начальными условиями для их решения являются:

$$p_0(0) = 1, \quad p_i(0) = 0, \quad i = 1,2,...,n,$$

т.е. в начальный момент система свободна.

На практике, если интересуются переходным режимом СМО, система дифференциальных уравнений (98) численно решается на ЭВМ. Ее решение дает нам все вероятности состояний:

$$p_0(t), p_1(t), ..., p_n(t),$$

как функции времени.

Однако основной интерес представляют предельные вероятности состояний, характеризующие установившийся при $t \to \infty$ режим работы СМО. Для нахождения предельных вероятностей нет необходимости численно решать систему уравнений (98). Предельные вероятности уже были получены в виде формул Эрланга (86), см. пример в разделе 2.4:

$$p_k = \frac{(\lambda/\mu)^k}{k!} p_0, \qquad k = 1,2,...,n$$
$$p_0 = \frac{1}{1 + \frac{\lambda/\mu}{1!} + \frac{(\lambda/\mu)^2}{2!} + ... + \frac{(\lambda/\mu)^n}{n!}}. \tag{99}$$

В эти формулы интенсивность потока заявок λ и интенсивность потока обслуживаний (для одного канала) μ входят только своим отношением λ/μ, но не по отдельности. Обозначим

$$\rho = \lambda/\mu.$$

Величина ρ называется «приведенной интенсивностью» потока заявок и имеет смысл среднего числа заявок, приходящих в СМО за среднее время обслуживания одной заявки. С учетом сделанного обозначения, формулы (99) примут вид:

$$p_k = \frac{\rho^k}{k!} p_0, \qquad k = 1,2,...,n$$
$$p_0 = \left(1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + ... + \frac{\rho^n}{n!}\right)^{-1}. \tag{100}$$

Формулы (100) также называют формулами Эрланга. Они выражают предельные вероятности всех состояний системы в зависимости от интенсивности потока заявок λ, интенсивность обслуживания μ и числа каналов СМО n.

Зная вероятности состояний можно найти характеристики эффективности СМО: относительную пропускную способность q, абсолютную пропускную способность А и вероятность отказа $P_{отк}$.

Действительно, заявка получает отказ, если приходит в момент, когда все n каналов заняты. Вероятность этого равна:

$$P_{ОТК} = p_n = \frac{\rho^n}{n!} P_0 .$$

Вероятность того, что заявка будет принята к обслуживанию (она же относительная пропускная способность q) дополняет $P_{отк}$ до единицы:

$$q = 1 - p_n . \qquad (101)$$

Абсолютная пропускная способность:

$$A = \lambda q = \lambda(1 - p_n). \qquad (102)$$

Одной из важных характеристик СМО с отказами является среднее число занятых каналов (в данном случае оно совпадает со средним числом заявок, находящихся в системе). Обозначим эту величину через $\overline{L}$. Ее можно вычислить по формуле:

$$\overline{L} = 0 \cdot p_0 + 1 \cdot p_1 + ... + n \cdot p_n , \qquad (103)$$

как математическое ожидание дискретной случайной величины, принимающей значения 0,1,...,n с вероятностями p_0, p_1, ..., p_n. Однако значительно проще выразить среднее число занятых каналов через абсолютную пропускную способность А, которую мы уже знаем. Действительно, А есть не что иное, как среднее число заявок, обслуживаемых в единицу времени, один занятый канал обслуживает в среднем за единицу времени μ заявок; среднее число занятых каналов получится делением А на μ:

$$\overline{L} = \frac{A}{\mu} = \frac{\lambda(1 - p_n)}{\mu},$$

или, обозначая $\lambda/\mu = \rho$,

$$\overline{L} = \rho(1 - p_n). \qquad (104)$$

Пример. В условиях предыдущего примера (λ=0.8, μ=0.667), вместо одноканальной СМО (n = 1) рассмотрим трехканальную (n = 3), т.е. число

линий связи увеличено до трех. Найти вероятности состояний, абсолютную и относительную пропускную способности, вероятность отказа и среднее число занятых каналов.

Решение. Приведенная интенсивность потока заявок:

$$\rho = \lambda / \mu = 0.8 / 0.667 = 1.2.$$

По формулам Эрланга (100) получаем:

$$p_1 = \frac{\rho}{1!} p_0 = 1.2 p_0;$$

$$p_2 = \frac{\rho^2}{2!} p_0 = 0.72 p_0;$$

$$p_3 = \frac{\rho^3}{3!} p_0 = 0.288 p_0;$$

$$p_0 = \frac{1}{1 + 1.2 + 0.72 + 0.288} \approx 0.312.$$

$p_1 \approx 1.2 \cdot 0.312 \approx 0.374$; $p_2 \approx 0.72 \cdot 0.312 \approx 0.224$; $p_3 \approx 0.288 \cdot 0.312 \approx 0.090$.

Вероятность отказа:

$$P_{ОТК} = p_3 = 0.090.$$

Относительная и абсолютная пропускные способности равны:

$$q = 1 - p_3 = 0.910; \; A = \lambda q = 0.8 \cdot 0.910 \approx 0.728.$$

Среднее число занятых каналов:

$$\bar{k} = \rho(1 - p_3) = 1.2 \cdot 0.91 \approx 1.09,$$

т.е. в установившемся режиме работы СМО в среднем будет занят один (1.09) канал из трех — остальные два будут простаивать. Этой ценой добывается сравнительно высокий уровень эффективности обслуживания — около 91% всех поступивших вызовов будет обслужено.

2.6.5. Одноканальная СМО с ожиданием

Рассмотрим простейшую из всех возможных СМО с ожиданием – одноканальную систему (n=1), на которую поступает поток заявок с интенсивностью λ; интенсивность обслуживания μ. Заявка, поступившая в момент, когда канал занят, становится в очередь и ожидает обслуживания.

Предположим, сначала, что количество мест в очереди ограничено числом *m*, т. е. если заявка пришла в момент, когда в очереди уже стоят *m* заявок, она покидает систему необслуженной.

Будем нумеровать состояния СМО по числу заявок, находящихся в системе (как обслуживаемых, так и ожидающих обслуживания):

– S_0 – канал свободен,

– S_1 – канал занят, очереди нет,

– S_2 – канал занят, одна заявка стоит в очереди,

.....

– S_k – канал занят, k – 1 заявок стоит в очереди,

.....

– S_{m+1} – канал занят, m заявок стоят в очереди.

Напишем выражения предельных вероятностей состояний:

$$p_0 = \left[1 + \sum_{k=1}^{m+1} \rho^k\right]^{-1},$$

$$p_k = \rho^k p_0, \quad k = 1, \ldots, m+1 \qquad (105)$$

$$\rho = \lambda / \mu.$$

Заметим, что в знаменателе последней формулы (105) стоит геометрическая прогрессия с первым членом 1 и знаменателем p; суммируя эту прогрессию, находим:

$$p_0 = \frac{1}{(1 + \rho^{m+2})/(1 + \rho)} = \frac{1+\rho}{1 + \rho^{m+2}}.$$

Таким образом, формулы (105) окончательно примут вид:

$$p_0 = \frac{1-\rho}{1-\rho^{m+2}},$$

$$p_k = \rho^k p_0, \quad k = 1, \ldots, m+1. \qquad (106)$$

Обратим внимание на то, что формула (106) справедлива только при $\rho \neq 1$ (при $\rho = 1$ она дает неопределенность вида 0/0). Но сумму геометрической прогрессии со знаменателем $\rho = 1$ найти еще проще, чем по формуле (106): она равна m + 2, и в этом случае $p_o = 1/(m + 2)$. Заметим, что тот же результат мы могли бы получить более сложным способом, раскрывая неопределенность (106) по правилу Лопиталя.

Определим характеристики СМО: вероятность отказа $P_{отк}$, относительную пропускную способность q, абсолютную пропускную

способность А, среднюю длину очереди $L_{OЧ}$, среднее число заявок, находящихся в системе $L_{СИСТ}$.

Очевидно, заявка получает отказ только в случае, когда канал занят и все m мест в очереди – тоже:

$$P_{ОТК} = p_{m+1} = \frac{\rho^{m+1}(1-\rho)}{1-\rho^{m+2}}. \qquad (107)$$

Находим относительную пропускную способность:

$$q = 1 - P_{ОТК} = 1 - \frac{\rho^{m+1}(1-\rho)}{1-\rho^{m+2}}. \qquad (108)$$

Абсолютная пропускная способность:

$$A = \lambda q.$$

Найдем среднее число заявок находящихся в очереди $L_{OЧ}$. Определим эту величину как математическое ожидание случайного числа заявок, находящихся в очереди L:

$$L_{\text{ОЧ}} = \mathbf{E}(L).$$

С вероятностью p_2 в очереди стоит одна заявка, с вероятностью p_3 две заявки, вообще с вероятностью p_k в очереди стоят k–1 заявок, и, наконец, с вероятностью p_{m+1} в очереди стоят m заявок. Среднее число заявок в очереди получим, умножая число заявок в очереди на соответствующую вероятность и складывая результаты:

$$L_{ОЧ} = 1 \cdot p_2 + 2 \cdot p_3 + \ldots + (k-1) \cdot p_k + \ldots + m \cdot p_{m+1} = \rho^2 p_0 + 2 \cdot \rho^3 p_0 + \ldots + (k-1) \cdot \rho^k p_0 + \ldots + m \cdot \rho^{m+1} p_0,$$

или

$$L_{ОЧ} = \frac{\rho^2 (1 - \rho^m (m+1-m\rho))}{(1-\rho)(1-\rho^{m+2})}. \qquad (109)$$

Математическое ожидание числа заявок, находящихся на обслуживании:

$$n_{об} = 0 \cdot p_0 + 1 \cdot (1 - p_0) = \frac{\rho - \rho^{m+2}}{1 - \rho^{m+2}}.$$

Таким образом, среднее число заявок, связанных с СМО, будет

$$L_{СИСТ} = L_{ОЧ} + \frac{\rho - \rho^{m+2}}{1-\rho^{m+2}}. \qquad (110)$$

Существенными характеристиками СМО с ожиданием являются среднее время ожидания заявки в очереди $t_{ож}$ и среднего времени пребывания заявки в системе $t_{сист}$. По формулам Литтла (90):

$$t_{ОЖ} = \frac{l_{ОЧ}}{\lambda}, \quad t_{СИСТ} = \frac{l_{СИСТ}}{\lambda} = \frac{l_{ОЧ}}{\lambda} + q/\mu, \qquad (111)$$

т.е. среднее время ожидания и среднее время пребывания равны, соответственно, среднему числу заявок в очереди или системе, деленному на интенсивность потока заявок.

Пример. Автозаправочная станция (АЗС) представляет собой СМО с одним каналом обслуживания (одной колонкой). Площадка при станции допускает пребывание в очереди на заправку не более трех машин одновременно (m = 3). Если в очереди уже находится три машины, очередная машина, прибывшая к станции, в очередь не становится, а проезжает мимо. Поток машин, прибывающих для заправки, имеет интенсивность λ = 1 (машина в минуту). Процесс заправки продолжается в среднем 1.25 мин. Определить:

– вероятность отказа;

– относительную и абсолютную пропускную способности СМО;

– среднее число машин, ожидающих заправки;

– среднее число машин, находящихся на АЗС (включая и обслуживаемую);

– среднее время ожидания машины в очереди;

– среднее время пребывания машины на АЗС (включая обслуживание).

Решение. Находим приведенную интенсивность потока заявок:

$$\mu = 1/1.25 = 0.8; \rho = \lambda/\mu = 1/0.8 = 1.25.$$

По формулам (106):

$$p_0 = \frac{1-1.25}{1-3.05} \approx 0.122, \; p_1 = 1.25 \cdot 0.122 \approx 0.152,$$

$$p_2 = 1.56 \cdot 0.122 \approx 0.191, \; p_3 = 1.95 \cdot 0.122 \approx 0.238,$$

$$p_4 = 2.44 \cdot 0.122 \approx 0.297.$$

Вероятность отказа $P_{отк} \approx 0.297$.

Относительная пропускная способность СМО $q = 1 - P_{отк} = 0.703$.

Абсолютная пропускная способность СМО $A = \lambda q = 0.703$ (машины в мин.).

Среднее число машин в очереди находим по формуле (109):

$$\bar{r} = \frac{1.25^2\,[1-1.25^3(3+1-3.75)]}{(1-1.25^5)(1-1.25\}} \approx 1.56,$$

т. е. среднее число машин, ожидающих в очереди на заправку, равно 1.56. Прибавляя к этой величине среднее число машин, находящихся под обслуживанием:

$$\overline{\omega} = \frac{1{,}25-1{,}25^5}{1-1{,}25^5} \approx 0.88,$$

получаем среднее число машин, находящихся на АЗС:

$$L_{СИСТ} = L_{Оч} + \overline{\omega} \approx 2{,}44.$$

Среднее время ожидания машины в очереди и среднее время, которое машина проводит на АЗС, по формулам (111) равны:

$$t_{ОЖ} = \frac{L_{ОЧ}}{\lambda} = 1.56 \text{ (мин)},$$

$$t_{СИСТ} = 1.56 + 0{,}88 = 2{,}44 \text{ (мин)}.$$

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Составьте логическую схему базы знаний по теме юниты.

МОДЕЛИРОВАНИЕ СИСТЕМ

ЮНИТА 1

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СИСТЕМ

Ответственный за выпуск Е.Д. Кожевникова
Оператор компьютерной верстки В.О. Шепелев

НОУ “Современная Гуманитарная Академия”
Подписано в печать Тираж
Заказ

Современная Гуманитарная Академия