Министерство образования и науки Российской Федерации

Амурский государственный университет

Н.В. Назаренко, Л.А. Соловцова

ПРАКТИКУМ ПО ТЕОРИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ

Благовещенск

Издательство АмГУ

2011

ББК 32.973-018 я73
Н 19

Рекомендовано
учебно-методическим советом университета

Рецензенты:
А.Н. Гетман, заведующий кафедрой прикладной информатики и математики БФ МосАП, канд.техн. наук, доцент;
Кафедра информатики и методики преподавания информатики БГПУ.

Назаренко, Н.В., Соловцова, Л.А.
Н 19 Практикум по теории информационных процессов и систем / Н.В. Назаренко, Л.А. Соловцова. – Благовещенск: Изд-во АмГУ , 2011. – 116 с.

Изложены методические рекомендации по выполнению курсовой работы. Приведены основные теоретические положения, задания к выполнению и пример проектирования базы данных.

Пособие предназначено для студентов специальностей 230102 «Автоматизированные системы обработки информации и управления», 230201 «Информационные системы и технологии», 220100 «Информатика и вычислительная техника», 010501 «Прикладная математика и информатика», а также для направлений подготовки бакалавров 230100.62 «Информатика и вычислительная техника», 230400.62 «Информационные системы и технологии», 010500.62 «Прикладная математика и информатика».

ББК 32.973-018 я73

Содержание

ВВЕДЕНИЕ

В пособии рассматриваются основные понятия теории информационных процессов и систем, идеология построения информационных систем, математический аппарат их формализации, возможности и пути использования информационных технологий при анализе, синтезе и проектировании таких систем.

Одно из практических занятий посвящено марковским случайным процессам. В работе рассматривается математический аппарат, который позволяет определить вероятностные характеристики системы, представленной с помощью марковского процесса. Для специалиста в области информационных систем важно знать, что практически любой процесс в окружающем нас мире можно представить как марковский.

Математические методы, используемые для анализа марковских процессов, позволяют найти и характеристики систем массового обслуживания, которые широко используются во всех сферах общественной жизни. Некоторые аспекты теории систем массового обслуживания отображены в данной работе.

В пособии рассматриваются вопросы, посвященные методам и моделям описания систем. Особое внимание уделяется качественным методам описания систем.

Также в работе рассматриваются вопросы системного подхода и системного анализ, который незаменим при комплексном исследовании сложных систем. В разделе рассматриваются некоторые задачи исследования операций, используемые анализа поведения подсистем, входящих в состав больших систем.

Практическое занятие № 1.

Методы анализа и исследования информационных систем

Объектом исследований теории систем является модель, заданная разными способами: кибернетическим подходом; динамическим описанием; каноническим представлением; агрегатным описанием.

Для теории информационных систем наиболее приемлемым является задание модели на основе кибернетического подхода.

Кибернетика рассматривает поведение систем во взаимодействии с другими системами и окружающей средой. Коренными понятиями кибернетики являются система и информация.

Кибернетические системы рассматриваются как системы управления, а процессы управления как процессы переработки информации.

Кибернетическая система — система, для которой принято допущение об относительной изолированности в информационном значении и абсолютной проницаемости в материально-аналитическом значении. Это предполагает, что количество информации в системе конечно, что всякое поступление информации в систему (информационный вход) и поступление информации из системы в среду (информационный выход), контролируемы и наблюдаемы. Материальные и энергетические потоки рассматриваются в качестве носителей информации.

Управление — функция системы, ориентированная либо на сохранение ее основного качества (совокупности свойств, утеря которых влечет к разрушению системы) в условиях изменения среды, либо на выполнение некоторой программы, которая должна обеспечивать устойчивость функционирования, гомеостаз, достижение определенной цели.

Другими словами, управление — целенаправленный перевод системы из одного состояния в другое — желаемое.

Система управления — система, в которой реализуются функции управления.

Простейшая структура приведена на рис.1. Структура системы управления может быть многоуровневой (структурой подчинения).

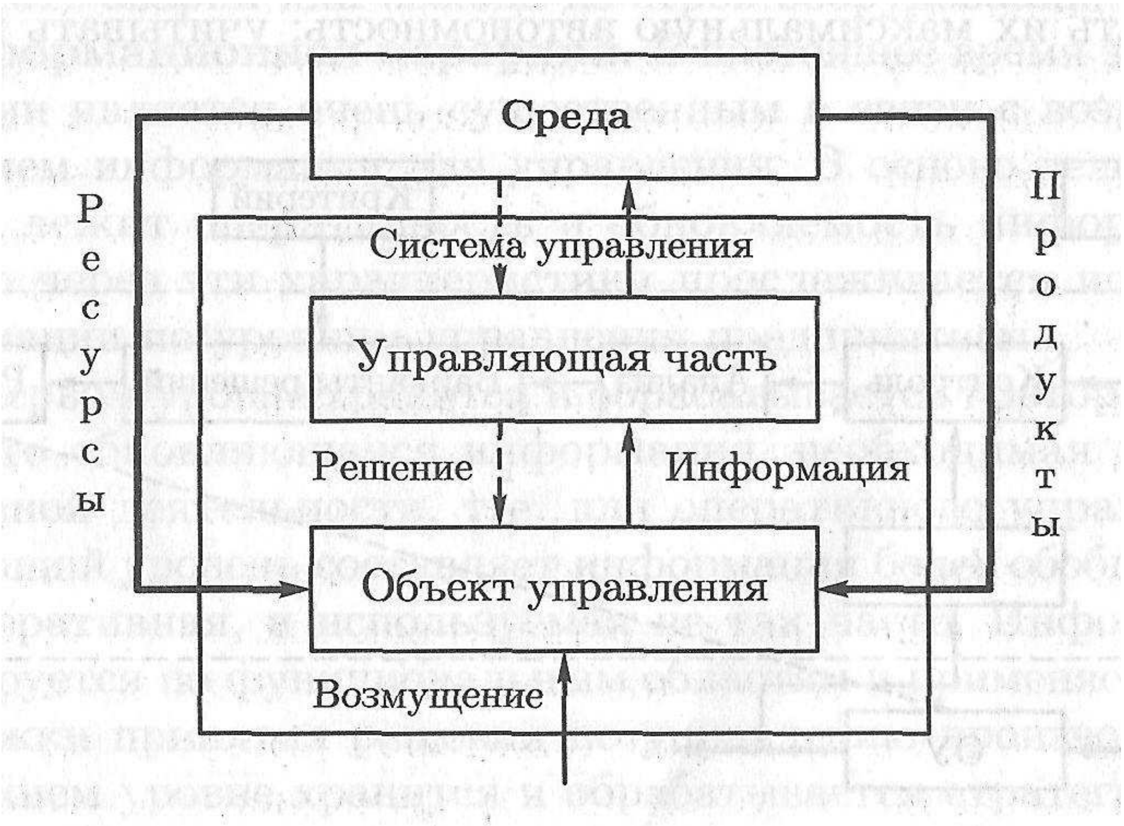


Рис.1. Простейшая структура системы.

Объект управления (ОУ) — динамическая система любой природы, преобразующая ресурсы в продукты и находящаяся под действием управляющих и возмущающих воздействий.

Возмущение — воздействие, выводящее систему в нежелательное состояние.

Решение (управляющее воздействие, управление) — воздействие, выбранное из множества возможных на основе поставленной цели и принятого критерия.

Критерий — оценка вариантов решений.

Цель — состояние, к которому стремится система.

Управляющая часть (УЧ) — часть системы, вырабатывающая решения и передающая их на объект управления.

Среда — метасистема, в которую рассматриваемая система управления входит составной частью.

Функционирование системы — работа системы в рамках заданной структуры.

Развитие системы — работа системы в условиях острых противоречий, которые могут вызвать изменение структуры.

Рассмотрим подробнее объект управления как систему преобразования ресурсов в продукты. Объект управления (ОУ) использует общие ресурсы, представленные на рис. 2**.**

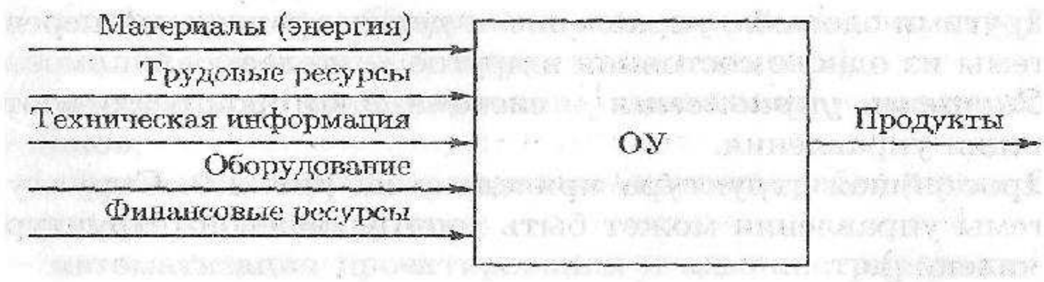


Рис.2. Упрощенная схема объекта.

При более подробном рассмотрении процессов в управляющей части можно получить совокупность этапов, показанных на рис.3**.** Эта схема пригодна для ручного, автоматического и автоматизированного управления.

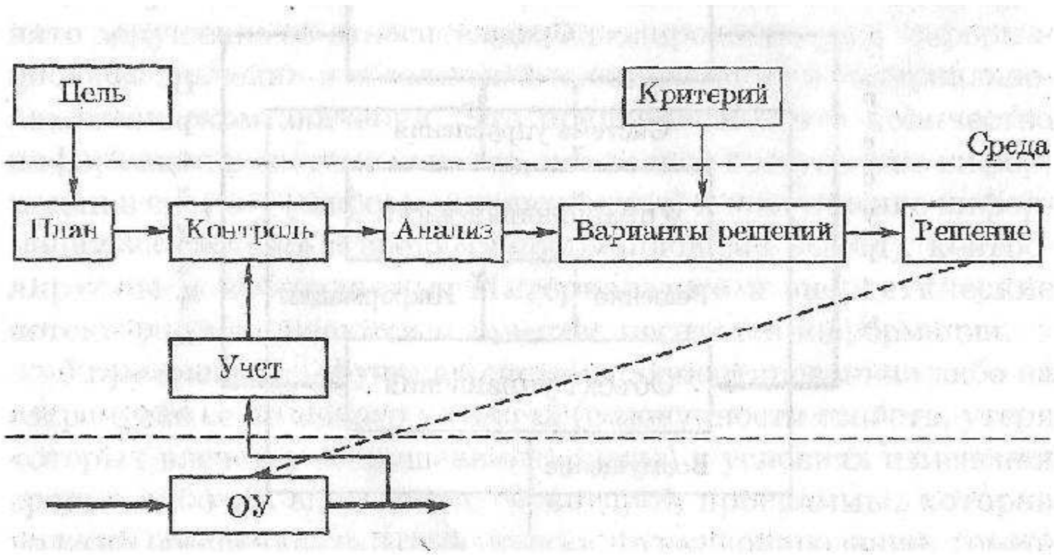


Рис.3. Цикл управления.

При анализе процесса управления из-за сложности объекта производят разделение процесса на части по различным признакам. Одним из главных признаков является вид иерархии: временная, пространственная, функциональная, ситуационная и информационная. Следует отметить, что деление системы на части не может быть однозначным, так как границы между частями являются всегда в какой-то мере субъективными. Выбор того или иного принципа выделения составных частей должен удовлетворять следующим основным условиям: обеспечивать их максимальную автономность; учитывать необходимость координации их действий для достижения общей цели функционирования, а также совмещать отдельные части.

Временная иерархия. Признаком деления здесь является интервал времени от момента поступления информации о состоянии объекта управления до выдачи управляющего воздействия. Чем больше интервал, тем выше уровень (ранг) элемента. Управление может осуществляться в реальном времени, с интервалом в сутки, декаду, месяц, квартал и т.д. Причем управляющий интервал выбирается не произвольно, а исходя из критериев, определяющих устойчивость и эффективность функционирования всей системы.

Пространственная иерархия. Признаком деления здесь является площадь, занимаемая объектом управления. Чем больше площадь объекта, тем выше его ранг. Данный признак является субъективным, так как не всегда площадь, занимаемая объектом, соответствует его значимости, и его можно использовать в случае аналогичности параметров элементов одного уровня.

Функциональная иерархия. В основе лежит функциональная зависимость (подчиненность) элементов системы. Такое разделение также является субъективным, так как в этом случае трудно выделить границы между элементами системы.

Ситуационная иерархия. Деление на уровни в данном случае производится в зависимости от эффекта, вызываемого той или иной ситуацией, например, от ущерба, возникающего в результате аварии или выхода из строя оборудования.

Информационная иерархия. В настоящее время этот вид иерархии является очень существенным в связи с возросшим значением информации для управления. В основе деления на уровни лежит оперативность и обновляемость информации. Именно через эти характеристики прослеживается иерархия информации по уровням управления предприятием.

На первом уровне хранится и обрабатывается повторяющаяся, часто обновляющаяся информация, необходимая для повседневной деятельности, т.е. для оперативного управления. Следующий уровень составляет информация более обобщенная, чем оперативная, и используемая не так часто. Информация группируется по функциональным областям и применяется для поддержки принятия решения по управлению производством. На верхнем уровне хранится и обрабатывается стратегическая информация для долгосрочного планирования. Для нее характерны высокая степень обобщенности, неповторяемость, непредсказуемость и редкое использование.

Функциональная модель процесса управления. В общем виде функциональная модель процесса управления представлена на рис.4**.** Учет информации об объекте управления состоит в регистрации, классификации и идентификации. На основе разнообразных математических моделей, описывающих реальное и требуемое состояние объекта, и критериев оптимальности анализируют информацию о состоянии объекта управления. Окончательная модель прогнозируемого состояния объекта управления формируется в виде плана. Возникающие за счет внешних воздействий отклонения от плана корректируют путем сравнения учетной и плановой информации, нового анализа и формирования управляющих воздействий (регулирования).

В большинстве случаев при информационном анализе процесса управления обычно рассматривают пассивную форму проявления информации, отражающую свойства внешней среды, объекта управления и самой управляющей системы. Однако не менее важное значение имеет и активная форма информации, являющаяся причиной изменения состояния управляемого объекта.

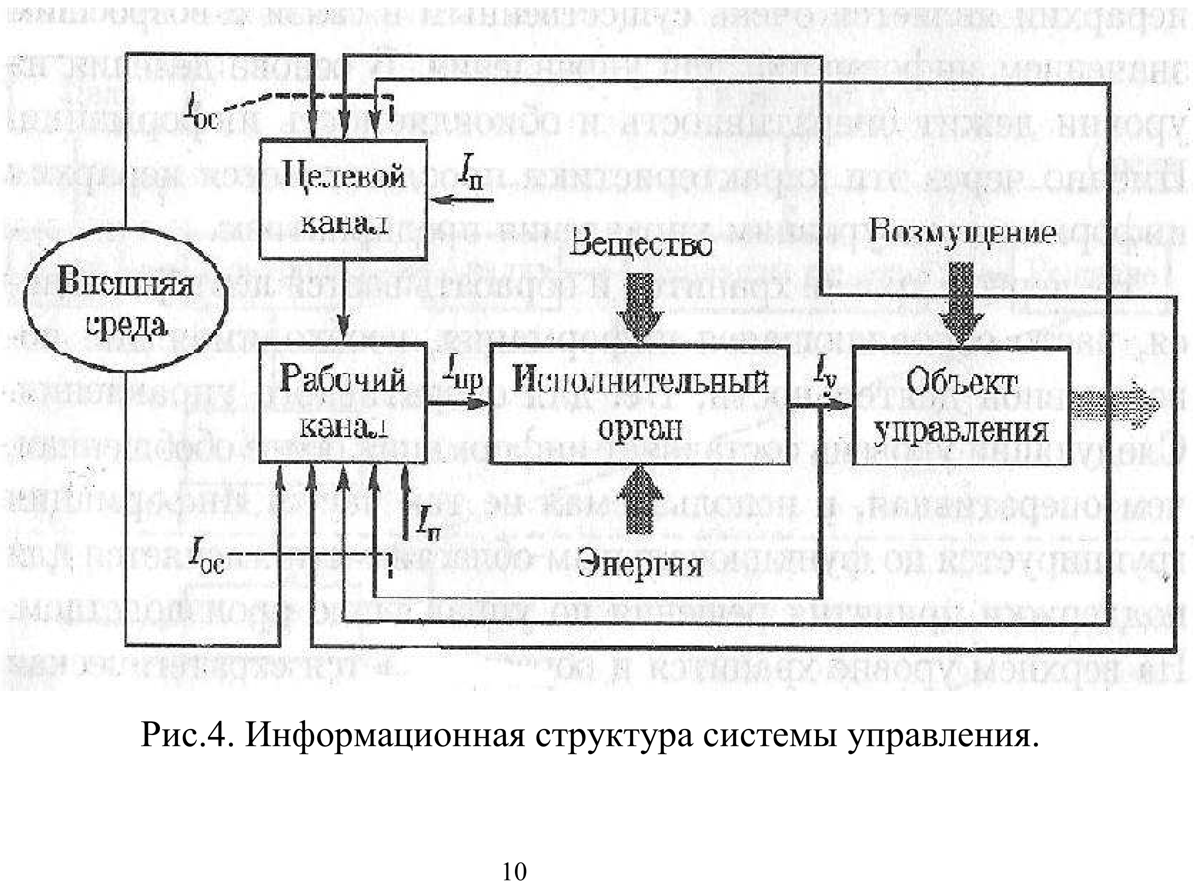


Рис.4. Информационная структура системы управления.

Принято выделять следующие качественно различимые формы проявления информации: осведомляющую — $I_{ос}$, преобразующую — $I_{п}$, принятия решения — $I_{пр}$ и управляющую — $I_{у}$.

К осведомляющей информации относят информацию о состоянии внешней среды, объекта управления и управляющей системы. Преобразующая информация включает информацию, содержащуюся в алгоритмах управления. Информация принятия решения является отражением образов и целей на конечное множество принимаемых решений. Управляющей информацией является информация, вызывающая целенаправленное изменение состояния объекта управления.

В любой системе управления можно выделить два информационных канала: целевой и рабочий. В целевом канале на основе информационных процессов происходит выбор цели и принятие решения по выбору управляющего воздействия. В рабочем канале формируется информация, реализуемая исполнительным органом, осуществляющим целенаправленное изменение состояния объекта управления через вещественно-энергетические характеристики. Целевой канал может находиться как на одном уровне иерархии с рабочим, так и на более высоком. На рис.4. представлена информационная структура системы управления, где выделены целевой и рабочий каналы, а также приведены основные формы проявления информации.

Большая система — система, превосходящая в каком-либо аспекте, важном для достижения цели наблюдателя, его собственные возможности. При этом один и тот же материальный объект в зависимости от целей наблюдателя и средств, имеющихся в его распоряжении, может приводить или не приводить к большой системе. Физические размеры объекта не являются существенными. Важным качеством больших систем является эмерджентность — наличие порождаемых свойств, специфичных именно для системы и не выводимых из известных свойств ее элементов и способов их соединения.

Эмержентность большой системы не дает возможности ограничиться изучением ее элементов и связей между ними, а предполагает целостный анализ большой сложной системы. Для больших систем характерны очень высокие значения такой характеристики как разнообразие, следовательно, в соответствии с законом необходимого разнообразия, для управления большой системой управляющая система должна иметь значительное собственное разнообразие. Это может быть достигнуто структуризацией управляющей системы, что применительно к экономике означает формирование взаимодействующих подсистем управления, каждая из которых решает частную задачу в условиях относительной самостоятельности. Следовательно, управляющей системе необходим центральный орган управления, функцией которого является координация действий подсистем, т.е. стимулирование и регламентация деятельности, обеспечивающие согласование собственных интересов подсистем с целью и интересами всей системы.

Общая теория систем — научная дисциплина, разрабатывающая методологические принципы исследования систем. Для познания она использует научный инструмент - системный подход, главными категориями которого являются: система, структура, иерархия, сложность.

Основные особенности общей теории систем:

- построение на основе общенаучного понятия системы;
- объединение всех специализированных теорий системы;
- использование логико-математического аппарата для исследования формальных систем;
- объединение теории различных аспектов поведения систем;
- широкое использование методов аналогии и моделирования.

Важная задача теории систем — создание общей методологии упрощения абстрактных систем и тем самым общей теории моделирования.

При кибернетическом подходе к исследованию системы рассматривается динамика ее функционирования во времени.

Представление системы на основе кибернетического подхода показано на рис.5. Признаки кибернетической системы — наличие объекта и органа управления $S_у$, связанных обратными и прямыми информационными каналами, образующими замкнутый контур; целей, критериев эффективности $К_{эф}$ и ограничений; стратегии, плана, алгоритма (инструкции) и программ управления.

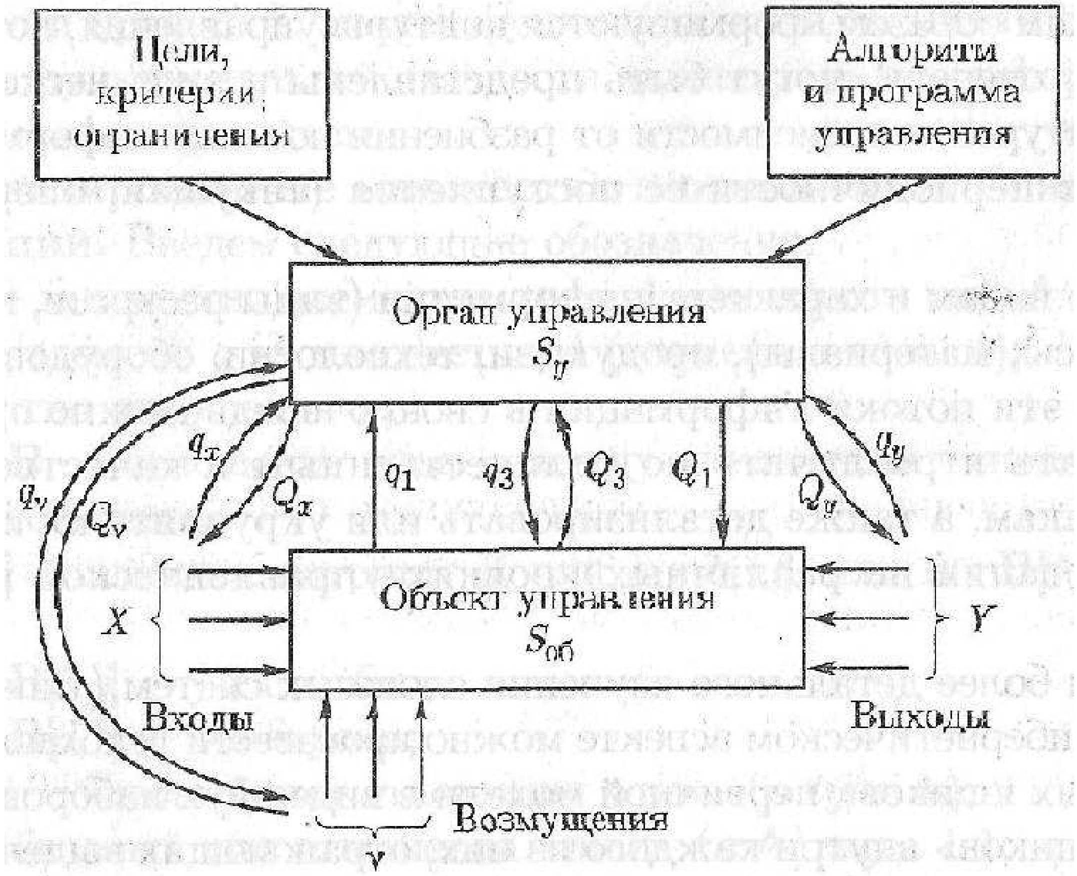


Рис.5. Схема кибернетической системы.

При отклонении объекта управления от заданной программы информация по каналам обратной связи поступает от объекта в орган управления. Поступившая информация разрабатывается и сопоставляется с информацией, характеризующей программу (план) достижения целей, определяется рассогласование соответствующих параметров. В управляющем органе вырабатывается и принимается управленческое решение по устранению рассогласований, которое в виде управляющих воздействий подается на объект управления (через специальные исполнительные устройства). Наличие всех необходимых признаков кибернетической системы обеспечивает устойчиводть ее функционирования.

В общем случае управление объектом в кибернетической системе объектом осуществляется по входам, выходам, по структуре и целям, параметрам внешней среды, если эти источники снабжены специальными средствами сбора, передачи и преобразования информации и каналами обратной и прямой связи с объектом управления.

На рис.5. показаны информационные потоки в зависимости от вида источника и соответствующих каналов обратной связи. Входы и выходы связаны с объектом и представляют собой материальные потоки, перерабатываемые объектом. Каждый компонент материального потока характеризуется совокупностью параметров и переменных, образующих множества информационных признаков, составляющих информационные потоки.

Информационные потоки формируются из документов, содержащих значения параметров, полученных по результатам их измерений в процессе контроля за состоянием входов, выходов и объекта в некоторые моменты времени. Эти потоки являются выходными для объекта и входными для органа управления, поступающими по каналам обратной связи. В результате переработки этой информации в подразделениях управляющего органа принимается решение, которое в виде директивных документов, образующих потоки, передается по каналам прямой связи на объект и реализуется в виде управляющих воздействий.

Таким образом, формируются контуры управления, которые, в свою очередь, могут быть представлены в виде нескольких подконтуров, в зависимости от разбиения потоков информации:

- по периодичности ее поступления (текущая, оперативная);

- по видам и характеру информации (виды ресурсов, кадры, финансы, материалы), продукции, технологии, оборудования.

Все эти потоки информации в свою очередь можно организовывать и различать по количественным и качественным признакам, а также детализировать или укрупнять их при использовании на различных уровнях управленческой иерархии.

Для более детального изучения сложных систем (типа завода) в кибернетическом аспекте можно произвести декомпозицию «черных ящиков» первичной модели в виде двух наборов «черных ящиков» внутри каждого из них, отражающих выделенные основные структурные элементы объекта и органы управления. К таким элементам можно отнести элементы оборудования производственных линий, механизмов, станков, роботов-манипуляторов; устройства приема и переработки и информации; средства и каналы связи и т.п.; персонал.

Взаимодействующие элементы системы соединяются функционально-информационными параметризованными (и при необходимости структурными) связями, образующими локальные специализированные потоки и контуры управления. При этом исходные черные ящики «светлеют» и становятся серыми, так как в них образуются определенные структуры из элементов в соответствии с логическими отношениями отображаемых ими элементов кибернетической системы.

Этапы построения кибернетической системы:

I этап — выявление функций системы, реализующих поставленную задачу;

II этап — выбор варианта выполнения функций, т.е. выбор структурных элементов системы;

III этап — синтез системы: выбор сочетаний компонентов системы.

Развитием кибернетического подхода является логический анализ структур ИС. В рамках построения информационной системы последовательно рассмотрим его составляющие: типизированные множества и отношения; основные операции; прочность и сцепление компонентов, ИС; анализ информационной связности процессов; анализ функциональной связности систем; анализ функциональной связности данных; анализ информационной связности систем.

Типизированные множества и отношения. Основные операции. Введем следующие обозначения:

А — множество элементов типа **A** или просто тип **A ;**

α — элемент **α** из множества **A** (элемент **α** типа **A) ;**

г — тип отношения г;

А г В — отношение типа г между элементами типов **A и B** соответственно;

s[i,j] — элемент матрицы **S,** или **a г b[i,j],** или A **rB[i,j],** или $\mathbf{a_i}$ **г** $\mathbf{b_j}$**,**

A r B[* **,j]** — j-й столбец;

А г В [i,*] — i-я строка;

type — базовый тип отношения «тип» **(a type** A);

: — базовый тип отношения **«тип» (a**:A) вида 1-1;

→ — базовый тип отношения «тип» **(a**→A) произвольного вида (1-1, 1-01, 1-M, 1-OM)

A_i — г-е подмножество множества **A ;**

from — базовый тип отношения «из» **(AJromA);**

class — базовый тип отношения «обобщения/конкретизации»;

det — базовый тип отношения «агрегирования/детализации»;

in — базовый тип отношения «вход»; **out** — базовый тип отношения выход»;

and — логическая операция «И»;

or — логическая операция «ИЛИ»;

not — логическая операция «НЕ»;

⊗ — операция логического перемножения матриц (A **rl** B ⊗ B *r2* C);

× — операция арифметического перемножения матриц **(A r1B** × Вг2С);

(A *r2* **A)$^{\mathbf{t}}$** — транспонирование;

fB — строка весовых характеристик / элементов из B;

A r B * **f B** — операция поэлементного арифметического умножения каждой строки A г B на элементы строки **fB;**

W(B г B): mi — вес элементов матрицы отношения типа г, заданного на множестве элементов типа B;

F(ErA): **int —** функция целого типа, заданная на модели данных.

Рассмотрим некоторые частные случаи использования операции логического перемножения матриц.

1. Результирующее отношение задает пути между элементами из A и C через элементы из B:

$A\ г1\ B \otimes B\ г2C = A\ rl2\ \mathbf{C}$

$ar12c = \exists b \in B$

$a\ rl\ b\ \ and\ \ b\ r2\ c$

2. Результирующее отношение задает пути между элементами из A через элементы из B, содержащие цепочку связей *rl* и *r2***:**

$A\ r1\ B \otimes (A\ r2\ B)^t = A\ r12\ B$

$a_i r12 b_j = \exists b \in B$

$a_i\ rl\ b\ \ and\ \ b\ r2\ c_j$

3. Результирующее отношение задает пути между элементами из A через элементы из A, содержащие одну из связей *rl* или *r2***:**

$A\ r1\ A \otimes (A\ r2\ A)^t = A\ r12\ A$

$a_i r12 a_j = \exists a \in A$

$a_i\ rl\ a\ \ and\ \ a_j\ r2\ a$

4. Результирующее отношение задает пути между элементами из A через элементы из A, содержащие цепочку связей *rl* и *r*2:

$A\ r1\ A \otimes A\ r2\ A = A\ r12\ A$

$a_i r12 a_j = \exists a \in A$

$a_i\ r1\ a$ *and* $a\ r2\ a_j$

5. Результирующее отношение задает пути между элементами из A через элементы из B, содержащие одну или более связей *r1*:

$A\ r1\ B \otimes (A\ r1\ B)^t = A\ r12\ A$

$a_i r12 a_j = \exists b \in B$

$a_i\ r1\ b$ *and* $a_j\ r2\ b$

6. Результирующее отношение задает пути между элементами из A через элементы из A. содержащие цепочку связей *r1*:

$A\ r1\ A \otimes A\ r1\ A = A\ r11\ A = (A\ r1\ A)**2$

$a_i r11 a_j = \exists a \in A$

$a_i\ r1\ a$ *and* $a\ r1\ aj$

7. Результирующее отношение задает пути между элементами из A через элементы из A, содержащие одну или более связей *r1* (общие пост-элементы)

$A\ r1\ A \otimes (A\ r1\ A)^t = A\ r11\ A$

$a_i r11 a_j = \exists a \in A$

$a_i\ r1\ a$ *and* $a_j\ r2\ a$

8. Результирующее отношение задает пути между элементами из A через элементы из A, содержащие одну или более связей *r1*

$(A\ r1\ A)^t \otimes A\ r1\ A = A\ r11\ A$

$a_i r11 a_j = \exists a \in A$

$a\ r1\ a_i$ *and* $a\ r2\ a_j$

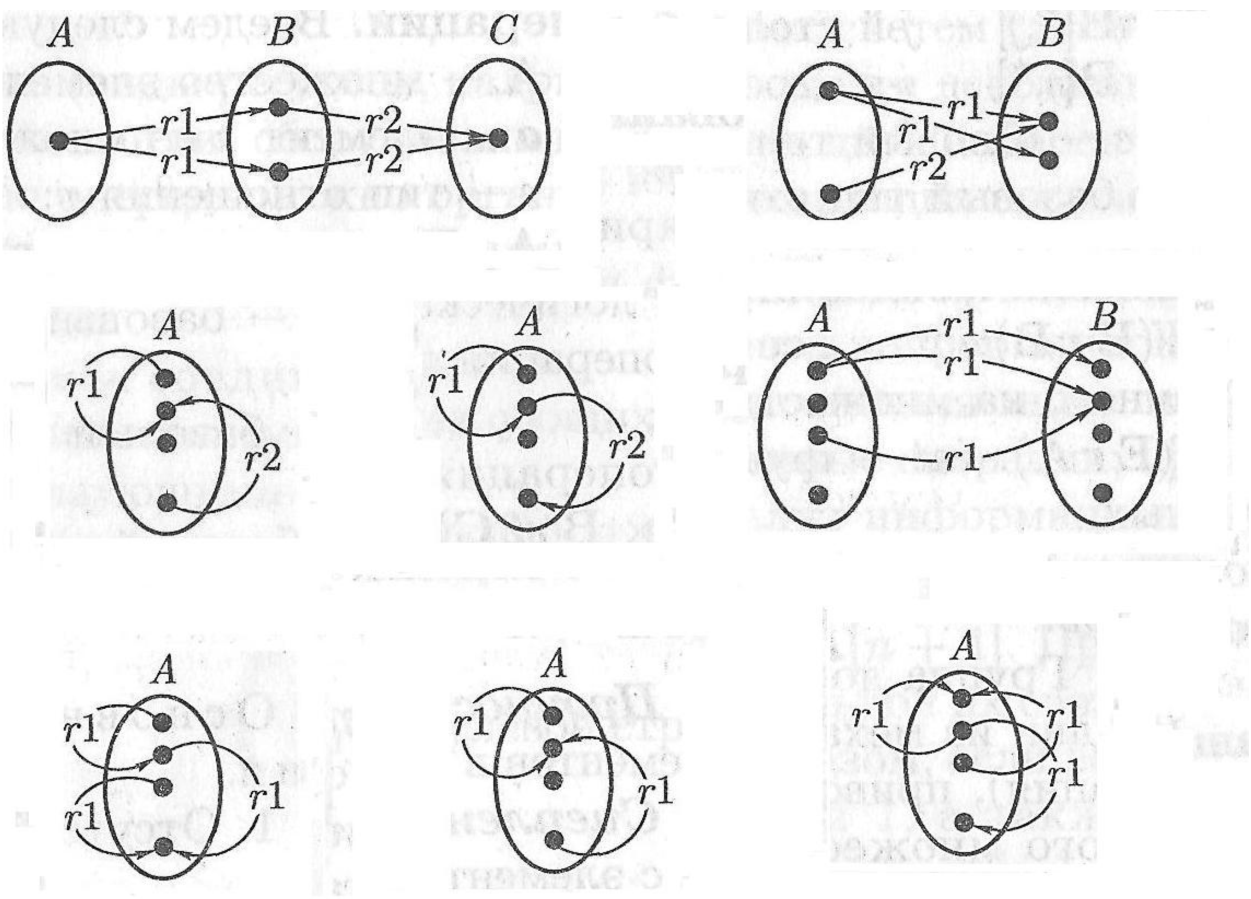


Рис.6. Операции логического перемножения матриц.

Прочность и сцепление компонентов ИС. Приведем основные определения.

Вес взаимосвязи — приведенный к диапазону 0.0...1.0 вес связи между двумя элементами одного вида заданного отношения.

Матрица взаимосвязи — матрица весов взаимосвязи заданного отношения между элементами одного вида.

Абсолютный коэффициент связности — нормированное среднее значение соответствующих взаимосвязей с учетом внутренней связности элементов *к = Sum(A r A)/N** 2*, где *A r A*. матрица взаимосвязи элементов А по виду отношения *r*; *Sum(ArA)* - сумма элементов *A r A*; ***N*** — размерность *A r A*.

Основные свойства отношения детализации.

1. Отсутствие циклов:

$AcircleA = AndetA\ and\ (AndetA)^t = I.$

2. Если задано некоторое отношение *ArB,* и *AdetA*, *BdetB*, то

$$ArB \subseteq A\det A \otimes ArB \quad and \quad ArB \subseteq ArB \otimes (B\det B)^t$$

Основные свойства отношения конкретизации.

1. Отсутствие циклов:

A circle A = A nclass A and (A nclass A)^t *= I.*

2. Если задано некоторое отношение *AгB*, и *AclassA*, то

$$ArB \subseteq AnclassA \otimes ArB$$

Прочность элемента — степень взаимосвязи его внутренних элементов в соответствии с заданной матрицей взаимосвязи.

Сцепление элемента — степень взаимосвязи этого элемента с элементами окружения.

Интегральная оценка связности. Концепция независимости и прочности направлена на выделение и локализацию в проектируемой системе сильно связанных групп элементов.

Группа должна быть прочна и слабо связана с окружением. Один из механизмов группирования — агрегирование (детализация), приводящее к иерархической структуре рассматриваемого множества элементов. Желательно иметь численную оценку, характеризующую качество иерархической структуры. Оценка должна ухудшаться, если сильно связанные элементы оказываются в разных агрегированных группах. Естественно стремиться, чтобы структура каждого агрегированного элемента могла быть оценена аналогичным образом. Будем оценивать качество некоторой структуры на основе матрицы коэффициентов связности *A fun A*, рассчитываемых для различных типов элементов и различной природы связности. Частными случаями являются:

- полносвязная система *S= II*, где *I* — единичная матрица, все элементы которой равны *I*;
- несвязанная система *S= I,* где *I* — единичная диагональная матрица.

Диагональные элементы *S[i.i]* являются интегральной оценкой связности внутренней структуры и характеризуют прочность элементов *A*. Прочие элементы характеризуют сцепление элементов A между собой. Представим A как множество элементов модели первого уровня детализации. Нулевой уровень представим как композиционную модель в виде одного агрегированного элемента путем объединения всех элементов из A. Добавим новый элемент в A:

$$A \det A = \begin{vmatrix} 0 & & & \\ 0 & & I & \\ 0 & & & \\ 1 & 1 & 1 & 1 \end{vmatrix} \quad (1)$$

Где I - единичная диагональная подматрица; $A[n + 1]$ агрегированный элемент. Расширим матрицу S путем добавления первой нулевой строки и нулевого столбца:

$$Q = \begin{vmatrix} & & 0 \\ & S & 0 \\ 0 & 0 & 0 \end{vmatrix} \quad (2)$$

Оценим прочность агрегированного элемента $A[n+ 1]$. При наличии N объединяемых элементов при максимальной их связности ($S=1$) число связей равно $N**2$. Таким образом, если все элементы характеризуются прочностью, равной 1, а также каждый из них максимально связан с остальными ($S[i,j] = S[j,i]= 1$), то сумма всех оценок связности будет равна $N**2$. Определим прочность агрегированного элемента:

$$K_s = S[n+1, n+1] = Sum(S) / N**2 \quad (3)$$

Деление на $N**2$ обеспечивает условие нормировки оценки. Для полносвязной системы

$$S[n+1, n+1] = Sum(S) / N**2 = 1 \quad (4)$$

Для несвязной системы прочность агрегированного элемента определяется только прочностью его несвязных составляющих элементов:

$$S[n+1, n+1] = Sum(S) / N**2 = 1 / N \quad (5)$$

Таким образом, получена интегральная оценка, которая характеризует качество агрегирования группы взаимосвязанных элементов. Следует отметить, что оценка учитывает не только парные коэффициенты взаимосвязи, но и прочность объединяемых элементов. Показатель прочности агрегированного элемента уменьшится при снижении прочности какого-либо элемента. Значит, полученный показатель оказывается чувствительным к изменениям структуры на всю глубину детализации. Коэффициент связности можно представить в виде суммы двух составляющих:

$$K_s = K_{in} + K_{out} \quad (6)$$

Где K_{in} — интегральная оценка внутренней связности объединяемых элементов; K_{out} – интегральная оценка связности объединяемых элементов между собой.

Коэффициент внутренней связности равен:

$$K_{in} = Sum(S[i,i], i = 1,...N) / N**2 \quad (7)$$

Коэффициент внешней связности определяется следующим образом:

$$K_{out} = Sum(S[i, j], i = 1,...N, j = 1,...N, i \neq j) / N**2 \quad (8)$$

Относительный коэффициент связности характеризует долю связности объединяемых элементов между собой по отношению к общей связности:

$$K_{SO} = K_{in} / K_S \quad (9)$$

Ранее рассматривалась оценка связности на одном уровне декомпозиции структуры. При этом не учитывались внутренние взаимосвязи элементов и взаимосвязь объемлющего элемента с окружением. Применим описанный выше подход для оценки связности иерархических структур.

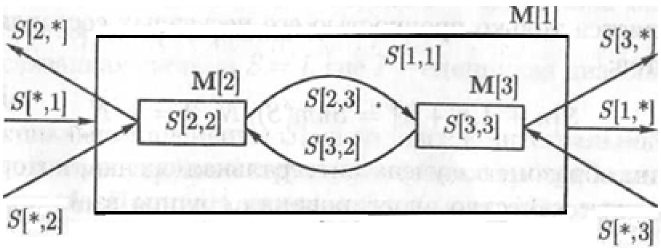


Рис.7. Пример иерархической структуры.

Пусть внутренняя связность элемента *M fun M[i,j]* равна *S[i,j]*. На рис.7. приведен пример иерархической структуры и связности ее элементов.

Известны внутренние связности 2-го и 3-го элементов *S [2,2], S[3,3]*, а также их взаимосвязи между собой *S[2,3], S[3,2]* и окружением *S[2,*]*, *S[3,*]*, *S[* ,2], S[* ,3].*

Требуется найти для агрегированного элемента *M[1]* его прочность *S[1,1]* и его связность окружением *S[1, *]* и *S[* ,1].*

Прочность *M[1]* оцениваем как интегральный коэффициент связности элементов его детализации *M[2]* и *M[3]*:

S[1,1] = (S[2,2] + S[3,3])/2**2 + (S[2,3] + S[3,2])/2**2

Для определения связности агрегированного элемента *M[1,1]* с окружением рассмотрим ряд частных случаев.

Вариант 1. Полная связность внутренних элементов *M[2]* и *M[3]* с окружением (элементом *M[4]*). В этом случае

S[2,4] = S[4,2] = S[3,4] = S[4,3] = 1.

Естественно предположить, что в этом случае *S[1,4] = S[4,1] = 1*. Поэтому по условиям нормирования связность агрегированного элемента с окружением определяется следующим образом:

S[l,4] = (S[2,4] + S[3,4])/2

S[4,1] = (S[4,2] + S[4,3])/2

Это выражение остается справедливым и в случае, когда связность какого-либо из внутренних элементов с окружением равна 0.

В общем случае связь агрегированного элемента с окружением будем оценивать как среднее по множеству связей с окружением его составляющих элементов на первом уровне детализации:

S[m,j]=Sum(S* MdetM[m,*])/count(MdetM[m,*]) (10)

i — входит в состав агрегированного элемента *M[m,*]*.

Для перерасчета связности при выполнении операции агрегирования введем специальную операцию *Agr*.

$\mathrm{S_{agr}} = Agr(S, M\,\mathrm{det}\,M)$ – операция расчета связности и прочности агрегированных элементов *M* на основе исходной матрицы связности и отношения детализации *MdetM*.

Операция *Agr* на основе матрицы связности *S N* элементов и матрицы группирования этих элементов *GdetS* рассчитывает матрицу связности S_{agr}, в которой каждой группе соответствуют новые строки и столбцы.

Общий случай операции агрегирования имеет вид:

$$\mathrm{QffunQ}= \begin{matrix} \mathrm{MffunM} & \mathrm{MffunF} & 0 \\ \mathrm{FffunM} & \mathrm{FffunF} & \mathrm{FffunG} \\ \mathrm{GffunM} & \mathrm{GffunF} & \mathrm{GffunG} \end{matrix} \qquad (11)$$

где *M* – множество агрегируемых элементов для данной операции;

F –множество элементов, которые не агрегируются в данной операции;

G —множество групп (новых агрегированных элементов, создаваемых в данной операции);

E = M U F — исходное множество рассматриваемых элементов;

$\mathrm{S}= \begin{matrix} \mathrm{MffunM} & \mathrm{MffunF} \\ \mathrm{FffunM} & \mathrm{FffunF} \end{matrix}$ – исходная матрица связности элементов;

GdetM — матрица группирования элементов из М в соответствующие группы;

GffunG — матрица связности агрегированных элементов между собой (*GffunG[i,i]* — прочность группы);

GffunF, FffunG — подматрицы связности агрегированных элементов из *G* с элементами, которые не включены ни в одну группу;

$\mathrm{S_{пр}}= \begin{matrix} \mathrm{FffunF} & \mathrm{FffunG} \\ \mathrm{GffunF} & \mathrm{GffunG} \end{matrix}$ – матрица связности после выполнения операции агрегирования.

Анализ информационной связности процессов. Информационная связность процессов характеризует наличие общих данных, используемых соответствующей парой процессов (как по входу, так и по выходу). Абсолютный коэффициент информационной связности двух процессов равен 1, если эти процессы используют одно множество данных, и равен 0, если общих данных у процессов нет.

В качестве исходных данных должны быть заданы система, связность процессов которой оценивается; глубина детализации процессов по схеме требований действий; веса данных (потоков).

Обозначим множество данных через *E*, множество процессов — через *P*.

Связи процессов и данных по входу и выходу зададим в виде соответствующих отношений:

EinP – входные данные процессов;

EoutP – выходные данные процессов.

Пусть известна частота активизации процессов, заданная вектором *fP*.

Связь процессов с данными по входу или выходу определится следующим образом:

$$EinoutP=EinP \text{ or } EoutP. \quad (12)$$

Совместное использование данных хотя бы одним процессом определится следующим образом:

$$\mathrm{L{=}EcomE} = \mathrm{EinoutP} \otimes (\mathrm{EinoutP})^{t}\ , \quad (13)$$

где *L[i,j]=1*, если существует хотя бы один процесс, использующий совместно данные *E[i]* и *E[j]*.

Для того, чтобы учесть количество процессов, совместно использующих пару данных, необходимо операцию логического умножения заменить на операцию арифметического умножения:

$$\mathrm{C{=}EcountE} = \mathrm{EinoutP} \times (\mathrm{EinoutP})^{t}\,, \quad (14)$$

где $C[i,j] > 0$ — количество процессов, совместно использующих данные $E[i]$ и $E[j]$.

Для анализа связности данных часто необходимо учитывать частоту активизации процессов В этом случае связность данных по использующим их процессам определится следующим образом:

$$\text{fC=EfcountE} = (\text{EinoutP*fP}) \times (\text{EinoutP})^{t}, \tag{15}$$

где $fC[i,j] > 0$, частота совместного использования данных $E[i]$ и $E[j]$ с учетом частоты активизации процессов; × — операция арифметического умножения матриц.

Каждый диагональный элемент $fC[i,i] >= fC[i,j]$ и $fC[i,i] >= fC[j,i]$. Проведем нормирование соответствующих элементов матрицы весов взаимосвязей путем деления каждого из них на диагональный элемент:

$$S[i,j] = fC[i,j]/fC[i,i],\ S[j,i] = fC[j,i]/fC[i,i]. \tag{16}$$

Матрица S определяет относительную частоту совместного использования данных, а каждый ее элемент можно назвать коэффициентом совместного использования. При $S[i,j] = 0$ соответствующие данные совместно не используются. При $S[i,j] = 1$ соответствующие данные на рассматриваемом множестве процессов всегда используются совместно. Введем понятие коэффициента связности.

Коэффициент связности $S[i,j]$ — количественная мера связи заданного вида элемента i с элементом j, каждый из которых относится к одному типу.

Матрица S содержит полезную информация для анализа модели предметной области и проектных решений по разработке ИС. Число элементов такой модели может составлять несколько тысяч, количество взаимосвязей, по крайней мере, на два порядка больше. Расчеты вручную в этом случае практически невозможны. Автоматизация последовательного построения модели ИС с оперативной оценкой коэффициентов взаимосвязей различной природы для элементов заданных типов, несомненно, повысит качество проектных решений и ускорит процесс проектирования за счет более глубокого и полного системного анализа.

Множество данных можно рассматривать как набор сущностей модели данных, разрабатываемой в рамках конкретного проекта. Сущности в различной степени функционально зависимы между собой, что проявляется в наличии связей «*FOREIGN KEY*». Однако величина зависимости сущностей из проекта модели данных получена быть не может. Необходима процедурная модель обработки данных для количественной оценки связности сущностей. Описанный выше подход позволяет количественно оценить функциональные зависимости сущностей.

Модель данных представляет собой двухуровневое описание взаимосвязей двух множеств классов: атрибутов (A) и сущностей (E).

Структура сущностей описывается через состав атрибутов с указанием элементов первичного ключа $(primary\_ key)$ и внешних связей сущностей $(foreign\_кey)$:

$EstrA$ — структура сущностей;

$Eprimery\_ key\ A$ — первичные ключи сущностей;

$Aforeign\_ кeyE$ — внешние ключи сущностей.

Зависимости сущностей по FOREIGN KEY можно описать следующим образом:

$$\mathrm{Eforeign\_keyE} = \mathrm{EstrA} \otimes \mathrm{Aforeign\_keyE}, \qquad (17)$$

где $Eforeign\_keyE[i,j]$ равно 1, если существует хотя бы один атрибут из состава $E[i]$, который является ссылкой на $E[j]$.

Более сильная зависимость, когда связь является идентифицирующей:

$$\mathrm{Eprimary\_ keyE} = \mathrm{Eprimary\_ keyA} \otimes \mathrm{Aforeign\_ keyE}, \qquad (18)$$

где $Eprimary\_\_keyE[i,j] =$ 1, если существует хотя бы один ключевой атрибут из состава $E[i]$, который является ссылкой на $E[j]$.

Оценка функциональной зависимости данных основывается на анализе внешних связей сущностей *(foreign_key)*. Наличие таких зависимостей определяется выражением:

$$EfynE = EstrA \otimes Aforeign\_keyE, \tag{19}$$

где $EfunE[i,j] = 1$, если существует хотя бы один атрибут из состава $E[i]$, который является ссылкой на сущность $E[j]$.

Пусть известна частота активизации связи, заданной атрибутом $A[j]$ из состава этой сущности $E[i]$. Вектор строка fA задает эту меру. Сущность $E[i]$ активизируется, когда активизируется или атрибут из ее состава или атрибут ссылка на данную сущность. Поэтому для оценки взаимосвязи атрибутов предварительно получим матрицу зависимости сущностей и атрибутов:

$$EfunA = (EstrA \text{ or } (Aforeign\_keyEf)^t), \tag{20}$$

где $EfunA[i,j] = 1$, если атрибут $A[j]$ входит в состав $E[i]$ или является ссылкой на $E[i]$.

Тогда количественная оценка взаимосвязей сущностей по внешним связям с учетом их частоты активизации определится следующим образом:

$$fC=EffunA = (EfunA*fA) \times (EfunA)^t, \tag{21}$$

где $fC[i,j] > 0$ — частота совместного использования данных $E[i]|$ и $E[j]$ с учетом частоты активизации ссылочных связей;

* — операция поэлементного арифметического умножения;

× – операция арифметического умножения матриц.

Далее необходимо нормировать $EffunE$ по элементам главной диагонали:

$$S = Norm(EffunE). \tag{22}$$

Интегральная оценка связности элементов модели данных определяется с помощью выражения:

$$K_s = Sum(S)/N**2. \tag{23}$$

Каждая независимая сущность потенциально является ядром соответствующей группы. Каждая зависимая сущность помещается в ту группу, с которой имеет наибольшую зависимость. Пусть G — множество групп. Представим структуру групп в виде матрицы *GstrE.* Тогда функциональные зависимости групп по внешним связям определятся: *GffunG = Agr(S, GstrE)* – матрица функциональной зависимости групп сущностей, где *S= Norm(EffunE)* — нормированная матрица функциональной взаимосвязи сущностей; *GstrE* — матрица группирования сущностей.

Интегральная оценка конкретного варианта группирования по коэффициенту связности может быть получена в соответствии со следующим алгоритмом:

1. Формирование *fC = EffunE;*

2. Нормирование *S — Norm(EffunE)* по элементам главной диагонали;

3. Формирование *GstrE;*

4. Расчет *GffunG = Agr(S, GstrE*) в соответствии с составом групп сущностей;

5. Расчет K_s= *Sum(GffunG)/N**2.*

Пример 1. Требуется оценить связность сущностей для фрагмента модели данных, представленной на рис.8.

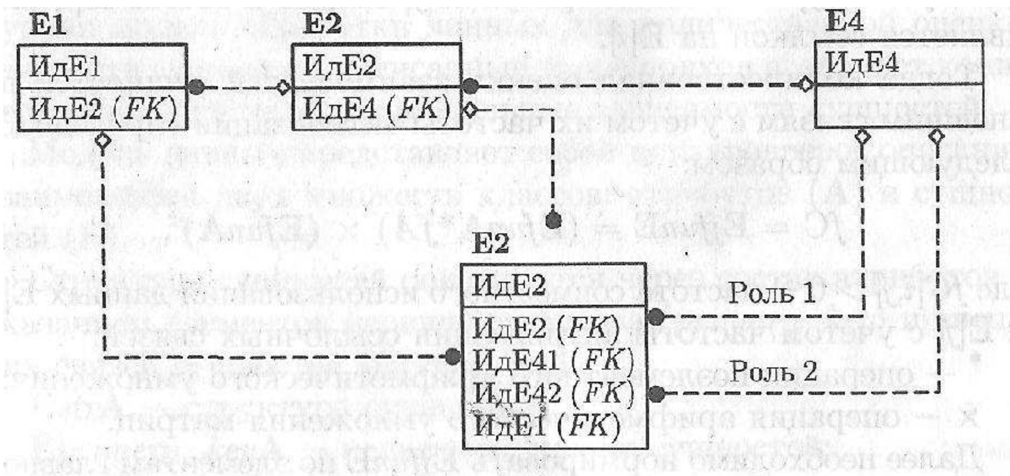


Рис.8. Фрагмент модели данных.

Последовательно строится матричная модель для фрагмента в соответствии с выражениями, приведенными выше.

$$\text{EstrA}=\begin{vmatrix}1&1&0&0&0&0&0&0&0&0\\0&0&1&1&0&0&0&0&0&0\\0&0&0&0&1&1&1&1&1&0\\0&0&0&0&0&0&0&0&0&1\end{vmatrix}$$ – состав атрибутов сущностей

$$\text{Aforeign\_keyE}=\begin{vmatrix}0&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&0&1\\1&0&0&0\\0&0&0&0\end{vmatrix}$$ – внешние ключи сущностей

$\text{fA}=\begin{vmatrix}0{,}0&1{,}0&0{,}0&1{,}0&0{,}0&0{,}5&0{,}5&0{,}5&0{,}5&0{,}0\end{vmatrix}$ – частота активизации атрибутов.

$$\text{EfunA}=\begin{vmatrix}1&1&0&0&0&0&0&0&1&0\\0&1&1&1&0&1&0&0&0&0\\0&0&0&0&1&1&1&1&1&0\\0&0&0&1&0&0&1&1&0&1\end{vmatrix}$$ – функциональные зависимости сущностей и атрибутов;

$$\text{EfunA*fA}=\begin{vmatrix}0{,}0&1{,}0&0{,}0&0{,}0&0{,}0&0{,}0&0{,}0&0{,}0&0{,}0&0{,}0\\0{,}0&1{,}0&0{,}0&1{,}0&0{,}0&0{,}5&0{,}0&0{,}0&0{,}0&0{,}0\\0{,}0&0{,}0&0{,}0&0{,}0&0{,}0&0{,}5&0{,}5&0{,}5&0{,}5&0{,}0\\0{,}0&0{,}0&0{,}0&1{,}0&0{,}0&0{,}0&0{,}5&0{,}5&0{,}5&0{,}0\end{vmatrix}$$

$$\text{EffunE}=\begin{vmatrix}1{,}0&1{,}0&0{,}5&0{,}5\\1{,}0&2{,}5&0{,}5&1{,}0\\0{,}0&0{,}5&2{,}0&1{,}5\\0{,}0&1{,}0&1{,}0&2{,}0\end{vmatrix}$$

$$S=\mathrm{Norm(EffunE)}=\begin{vmatrix} 1,0 & 0,4 & 0,25 & 0,25 \\ 1,0 & 1,0 & 0,25 & 0,5 \\ 0,0 & 0,2 & 1,0 & 0,75 \\ 0,0 & 0,4 & 0,5 & 1,0 \end{vmatrix}$$

$$\mathrm{K}_S = Sum(S)/N**2 = 8.3/16 = 0.53$$

Формируем *GstrE* путем введения двух групп, каждая из которых является агрегированием сущностей *E[1],E[2]* и *E[3],E[4],* соответственно:

$$\mathrm{GstrE}=\begin{vmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{vmatrix}$$

Используя матрицу функциональной зависимости сущностей, приведенную выше, и матрицу группирования сущностей, выполним операцию *Agr*. В результате получим матрицу функциональной зависимости групп *GffunG:*

$$\mathrm{GffunG}=\begin{vmatrix} 1.0 & 0.4 & 0.25 & 0.25 \\ 1.0 & 1.0 & 0.25 & 0.5 \\ 0.0 & 0.2 & 1.0 & 0.75 \\ 0.0 & 0.4 & 0.5 & 1.0 \end{vmatrix} Agr \begin{vmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{vmatrix}$$

$$\mathrm{GffunG}=Agr(S, GstrE)=\begin{vmatrix} 0.85 & 0.31 \\ 0.15 & 0.81 \end{vmatrix}$$

На основе *GffunG* рассчитаем интегральные показатели полученной структуры:

K_s= *Sum(GffunG)/N**2* = 2,12/4 = 0,53; $K_{гп}$ = *Sum(S[i,i\, г = l..N)/N**2* = 0,415; K_{oul} = *Sum(S\i,j\, i = 1..N, j = 1..N, i * j)/N**2* = 0,115; KJK_S= 0,783; K_0JK_S= 0,217.

За основу формирования супергрупп принимается матрица функциональной зависимости групп данных *GffunG.*Необходимо принять решение о количестве супергрупп. Можно последовательно проверять гипотезы о количестве супергрупп, равном 1, 2, 3 и т.д. Сравнение вариантов предлагается проводить на основе интегрированной оценки связности иерархической структуры.

Рассмотрим вариант формирования структуры супергрупп при известном их количестве, например, равном 2. Необходимо так разделить элементы из G, чтобы интегрированный показатель качества разделения был максимален.

Приблизительное решение: -

1. Выбирают две самые прочные группы, которые имеют слабые связи между собой;

2. Для каждой из оставшихся групп решение о принадлежности к одной из супергрупп принимаем на основе интегральной оценки ее связности с каждой из супергрупп.

В результате получим две матрицы: $SGstrG$ — структура супергрупп; $SGffunSG = Agr(GffunG, SGstrG)$ — матрица функциональной зависимости групп сущностей.

Функциональное представление модели данных рассматривает атрибут как функцию доступа. Введем следующие обозначения:

$Q \subset E$ - множество независимых сущностей;

$Qprimery\_key\ A = I$ — независимая сущность, имеющая простой первичный ключ;

$Z \supset E$ - множество зависимых сущностей;

$Zprimery\_keyA \otimes Aforeign\_keyE$ — в каждой строке количество ненулевых элементов > 1.

Строгая типизация атрибутов задается отношением $A \rightarrow D$. Строго говоря, $A{:}B$ — на физическом уровне только базовые типы. Однако на логическом уровне $Aforeign\_\ keyE$ определяет способ реализации функциональной зависимости сущностями путем включения в состав одной сущности атрибута ссылки на элементы другой сущности.

Рассмотрим зависимости между элементами Q. Двойственность атрибутов следует из отношения $E[j]A[i]E$.

Атрибут $A[i]\rightarrow D$ задает отображение $V{:}D$ на множество параметров. Параметры $VB{:}B$ — параметры простых (базовых) типов; $VQ{:}Q$ — параметры, ссылки на независимые сущности.

В частном случае $V{:}E$ заменяется на $Vforeign\_\ keyE$.

С одной стороны, атрибут выступает как свойство конкретной сущности. Этот аспект отражается в отношении $EstrA$. С другой стороны,

$$\mathrm{D[j]A[i]D}=(\mathrm{A[i]}\rightarrow \mathrm{D})^{t}\otimes A[i]inV\otimes V:D \text{ и } \mathrm{f(V{:}E)}\rightarrow \mathrm{E[j]} \qquad (24)$$

Объектно-ориентированный подход требует выделения сильно связанных компонентов при формировании структуры классов. Потенциально каждая сущность является кандидатом для формирования соответствующих классов обработки и/или представления. Однако возможность обоснованно группировать сущности в иерархические структуры, несомненно, поможет конструкторам ИС более обоснованно формировать структуру классов приложения.

Для формирования групп сущностей используют методику оценки связности, на основе которой строится матрица E/согт£E.

Анализ функциональной связности систем. Функциональная связность систем характеризует наличие общих требуемых процессов для процессов, входящих в разные системы.

Коэффициент связности двух систем равен 0, если все процессы каждой из систем требуют для своего выполнения разных подмножеств процессов, и равен 1, если множества требуемых процессов одинаковы.

В качестве исходных данных должны быть заданы:

- система, связность компонентов которой оценивается;
- глубина детализации систем по отношению состава;
- веса процессов.

Анализ функциональной связности данных. Функциональная связность данных характеризует наличие общих процессов, которые используют по входу-выходу оцениваемые данные.

Коэффициент функциональной связности двух данных равен 1, если множество процессов, использующих эти данные по входу-выходу одинаково, и равен 0, если общих процессов у данных нет.

Анализ информационной связности систем. Анализ характеризует наличие общих данных у этих систем.

Задание

1. Требуется оценить связность сущностей для фрагмента модели данных, представленной на рис.9, 10.

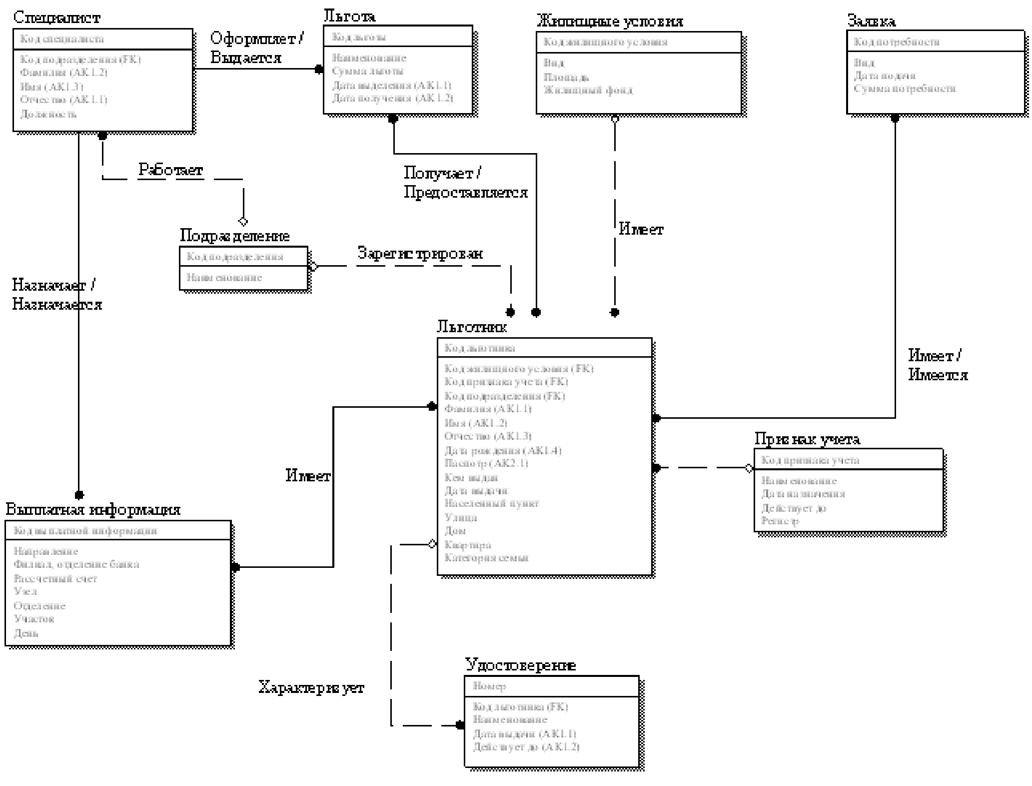


Рис.9. Схема модели данных ИС «Социальная защита населения».

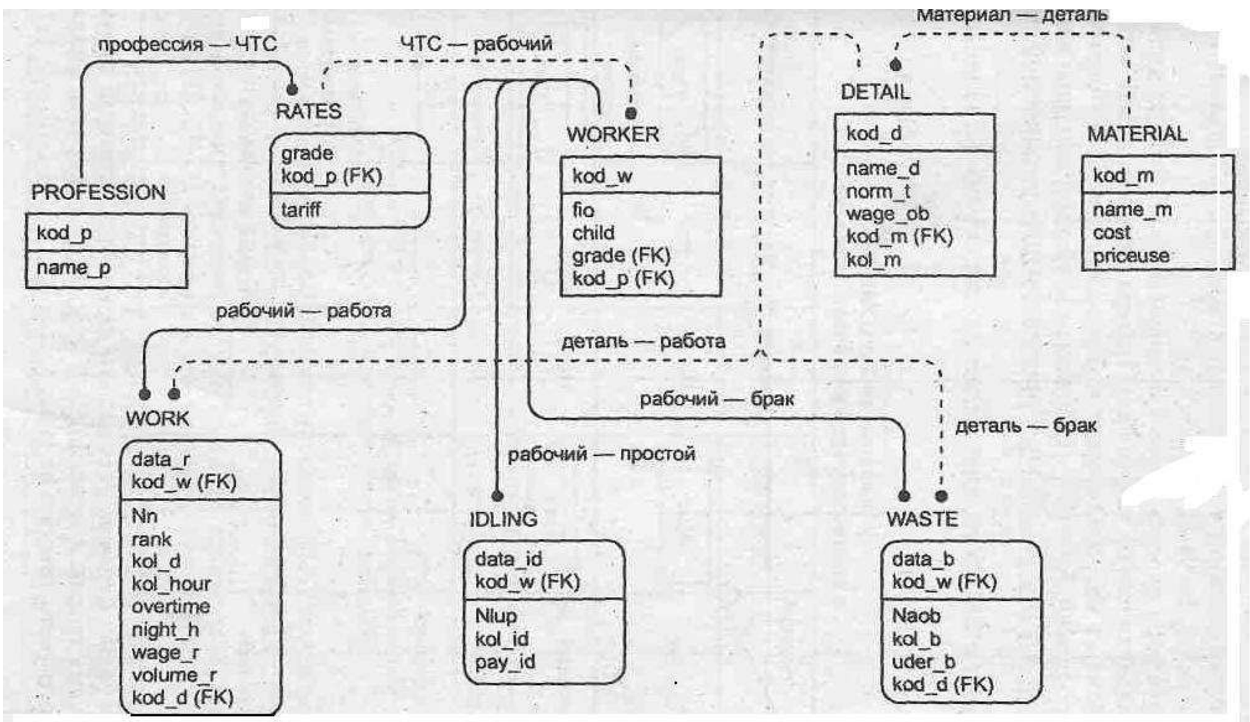


Рис.10. Схема модели данных для задачи учет труда и ЗП.

Контрольные вопросы.

1.Дайте определение системы и укажите ее основные свойства.

2.Охарактеризуйте уровни описания структуры системы.

3.Перечислите основные системные аспекты использования информационных технологий.

4.В чем суть Intranet? 8

5.Выделите основные фазы (поколения) эволюции информационных систем.

6.Укажите основные достоинства систем «клиент-сервер».

7.Перечислите основные уровни рассмотрения информационных технологий.

8.Раскройте содержание прикладного уровня информационных технологий.

9.Приведите классификацию базовых информационных процессов.

10.Охарактеризуйте понятия «системы и управления» с позиций системного подхода.

11.Какие виды иерархии характерны для информационных систем?

12.Выделите основные формы проявления информации.

13.Укажите отличительные признаки большой системы.

14.Что такое эмерджентность?

15.Укажите основные категории системного подхода.

16.Раскройте содержание основных этапов построения кибернетической системы.

17.Укажите основные этапы логического анализа информационных систем.

18.Какие типизированные множества и отношения используются в логическом анализе информационных систем?

19.Перечислите основные операции, применяемые в в логическом анализе информационных систем.

20.Как оцениваются прочность и сцепление компонентов информационных систем?

21.Раскройте содержание этапов анализа информационной связности действий и систем.

Практическое занятие № 2.
Марковские случайные процессы.

Для стохастических задач исследования редко удается построить математическую модель, позволяющую в явном виде найти интересующие величины в зависимости от условий операции α и элементов решения x. Но если процесс представляет собой, так называемый, марковский случайный процесс, то такая модель строится очень просто.

Вначале определим случайный процесс. Пусть имеется некоторая физическая система S, которая с течением времени меняет свое состояние, причем заранее неизвестным случайным образом. Тогда говорят, что в системе S протекает случайный процесс. Пусть, например, система S – космический корабль, выводимый на заданную орбиту. Процесс вывода неизбежно сопровождается случайными ошибками, отклонениями от заданного режима, на который приходится вводить коррекцию, в которой нет необходимости при отсутствии ошибок. Значит процесс вывода на орбиту – случайный процесс. Практически все процесс в окружающем нас мире случайны. Но до тех пор, пока случайные возмущения системы несущественны, мало влияют на интересующие параметры, ими можно пренебрегать и рассматривать процесс как детерминированный и неслучайный. Необходимость учета случайностей возникает, когда они непосредственно касаются исследуемых параметров.

Случайный процесс, протекающий в системе, называется марковским, если для любого момента времени t_0 вероятностные характеристики процесса в будущем зависят только от его состояния в данный момент t_0 и не зависят от того когда и как система пришла в это состояние. Пусть в настоящий момент t_0 система находится в определенном состоянии S_0 (рис. 11). В данный момент времени t_0 известно состояние системы и вся предыстория процесса. Все, что было при $t < t_0$. Если процесс случайный и непредсказуемый невозможно предсказать что произойдет с системой в будущем ($t > t_0$). Но некоторые характеристики процесса можно определить, например, вероятность того, что через некоторое время τ останется система в состоянии S_0 или перейдет в новое состояние S_1.

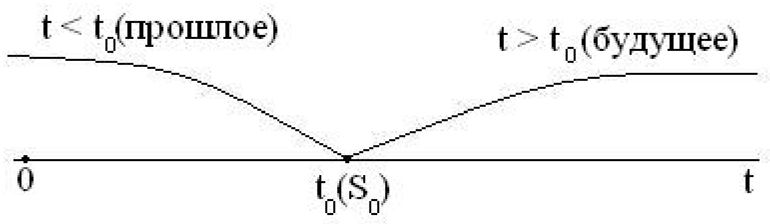


Рис. 11. Схема марковского процесса.

Если процесс марковский, то предсказывать можно, только учитывая настоящее состояние системы S_0 и забыв о его предыстории. Само состояние $S_{0,}$ но как только оно достигнуто, о прошлом можно забыть, в марковском процессе «будущее зависит от прошлого только через настоящее».

Пример марковского процесса: система S – счетчик Гейгера, на который время от времени попадают космические частицы; состояние системы в момент времени t характеризуется показанием счетчика – число частиц, пришедшие до данного момента. Пусть в момент t_0 счетчик показывает $S_{0.}$ Вероятность того, что в момент $t > t_0$ счетчик покажет то или иное число частиц S_1 зависит от t_0, но не зависит от того, в какие именно моменты приходили частицы до момента t_0.

Любой процесс можно рассматривать как марковский, если все параметры из прошлого от которого зависит будущее включить в настоящее. Например, имеется техническое устройство; в какой-то момент t_0 оно еще исправно, необходимо определить вероятность того, что оно проработает еще время τ. Если за настоящее состояние системы считать просто «система исправна», то процесс не марковский, потому что вероятность того, что она не откажет за время τ, зависит от того, сколько времени она уже проработала и когда был последний ремонт. Если оба эти параметры (общее время работы и время после последнего ремонта) включить в настоящее состояние системы, то процесс можно будет считать марковским.

На практике марковские процессы в чистом виде практически не встречаются, но часто приходится иметь дело с процессами, для которых влиянием предыстории можно пренебречь. При изучении таких процессов можно с успехом применять марковские модели.

Классификация марковских случайных процессов производится в зависимости от непрерывности или дискретности множества значений функции X(t) и параметра t. Различают следующие основные виды марковских случайных процессов:

– с дискретным состоянием и дискретным временем (цепь Маркова);

– с непрерывным состоянием и дискретным временем (марковские последовательности);

– с дискретным состоянием и непрерывным временем (непрерывная цепь Маркова);

– с непрерывным состоянием и непрерывным временем.

Марковские процессы с дискретными состояниями удобно представлять с помощью графа состояний (рис. 12), где состояния изображаются прямоугольниками (кругами и даже точками), а стрелками – возможные переходы из состояния в состояние. Возможные задержки в прежнем состоянии изображаются петлей. Число состояний системы может быть конечным и бесконечным (но счетным).

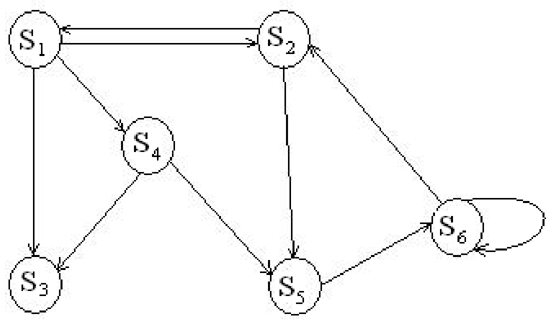


Рис. 12. Граф состояний системы.

Марковские процессы являются частным видом случайных процессов. Особое место марковских процессов среди других классов случайных процессов обусловлено следующими обстоятельствами: для марковских процессов хорошо разработан математический аппарат, позволяющий решать многие практические задачи; с помощью марковских процессов можно описать поведение достаточно сложных систем.

Рассмотрим математический аппарат для марковских цепей. Для такого процесса моменты t1, t2, …, когда система S может менять свое состояние, рассматривают как последовательные шаги процесса, а в качестве аргумента, от которого зависит процесс, выступает не время , а номер шага 1, 2, 3, … k…. Случайный процесс в этом случае характеризуется посдедовательными состояниями S(0), S(1), S(3), … S(k)… , где S(0) – начальное состояние системы (перед первым шагом), S(1) – состояние системы после первого шага, S(k) – состояние системы после k - ого шага.

Последовательность состояний системы S(0), S(1), S(3), … S(k) (S_k) можно рассматривать как последовательность случайных событий. Такая случайная последовательность случайных событий называется марковской цепью, если для каждого шага вероятность перехода из любого состояния S_i в любое состояние S_j не зависит от того, когда и как система пришла в состояние S_i. Начальное состояние задается заранее или определяется случайно.

Вероятностями состояния цепи Маркова называются вероятности $P_i(k)$ того, что после k-го шага система S будет находиться в состоянии S_i (i=1, 2, 3,… n). Очевидно для любого k

$$\sum_{i=1}^{n} P_i(k) = 1 \tag{25}$$

Начальным распределением вероятностей марковской цепи называется распределение вероятностей состояний в начале процесса:

$$P_1(0) \quad P_2(0) \quad ... \quad P_i(0) \quad ... \quad P_n(0)$$

В частном случае, если начальное состояние системы S в точности известно $S_i(0)=S_i$, то начальная вероятность $P_i(0)=1$, а остальные вероятности равны нулю.

Вероятность перехода на k-ом шаге из состояния S_i в состояние S_j называется условная вероятность того, что система S после k-ого шага окажется в состоянии S_j при условии, что непосредственно перед этим она находилась в состоянии S_i.

Поскольку система может пребывать в одном из n состояний, то для каждого момента времени t необходимо задать n^2 вероятностей перехода P_{ij}, которые удобно представит в виде следующей матрицы:

$$\left\| P_{ij} \right\| = \begin{bmatrix} P_{11} & P_{12} & \ldots & P_{1n} \\ P_{21} & P_{22} & \ldots & P_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ P_{i1} & P_{i2} & \ldots & P_{in} \\ \ldots & \ldots & \ldots & \ldots \\ P_{n1} & P_{n2} & \ldots & P_{nn} \end{bmatrix} \qquad (26)$$

Матрица (2) называется матрицей переходных вероятностей или переходной матрицей. Если переходные вероятности не зависят от номера шага (от времени), и зависят только от того, из какого состояния и в какое совершается переход, то соответствующую цепь Маркова называют однородной. Переходные вероятности однородной марковской цепи образуют квадратную матрицу размером n × n. Отметим некоторые особенности этой матрицы:

1. Каждая строка характеризует выбранное состояние, и ее элементы представляют собой вероятности всех возможных переходов за один шаг из выбранного состояния, в том числе и переход в самое себя.

2. Элементы столбцов показывают вероятности всех возможных переходов системы за один шаг в заданное состояние (строка характеризует вероятность перехода из состояния, столбец – в состояние).

3. Сумма вероятностей каждой строки равна единице, так как переходы образуют полную группу несовместных событий.

$$\sum_{j=1}^{n} P_{ij} = 1 \quad i = \overline{1,n} \tag{27}$$

4. По главной диагонали матрицы переходных вероятностей стоят вероятности того, что система не выйдет из состояния S_i, а останется в нем.

Если для однородной марковской цепи заданы начальное распределение вероятностей и матрица переходных вероятностей, то вероятность состояния системы $P_i(k)$ $(i=1,n; j=1,n)$ определяются по формуле:

$$P_i(k) = \sum_{j=1}^{n} P_j(k-1) P_{ji} \quad (i = \overline{1,n} \quad j = \overline{1,n}) \tag{28}$$

Пример. Рассмотрим процесс функционирования системы – автомобиль. Пусть автомобиль (система) в течение одной смены (суток)может находиться в одном из двух состояний: исправном (S_1) и неисправном (S_2). Граф состояний представлен на рис. 13.

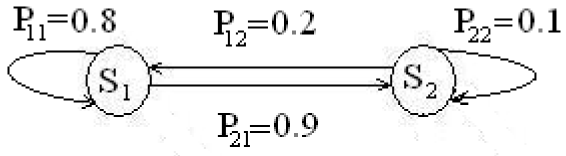


Рис. 13. Граф состояний автомобиля.

В результате проведения массовых наблюдений за работой автомобиля составлена следующая матрица вероятностей перехода:

$$\|P_{ij}\| = \begin{bmatrix} 0.8 & 0.2 \\ 0.9 & 0.1 \end{bmatrix}$$

где $P_{11}=0.8$ – вероятность того, что автомобиль останется в исправном состоянии;

$P_{12}=0.2$ – вероятность перехода автомобиля из состояния «исправен» в состояние «неисправен»;

$P_{21}=0.9$ – вероятность перехода автомобиля из состояния «неисправен» в состояние «исправен»;

$P_{22}=0.1$ – вероятность того, что автомобиль останется в неисправном состоянии.

Вектор начальных вероятностей состояний автомобиля задан P1(0)=0, P2(0)=1, $P(0)=\begin{bmatrix}0\\1\end{bmatrix}$.

Требуется определить вероятности состояний автомобиля через трое суток.

Используя матрицу переходных вероятностей, определим вероятности состояний $P_i(k)$ после первого шага (первых суток).

$$P_1(1)=P_1(0)\cdot P_{11}+P_2(0)\cdot P_{21} \quad P_1(1)=0\cdot 0.8+1\cdot 0.9=0.9$$
$$P_2(1)=P_1(0)\cdot P_{12}+P_2(0)\cdot P_{22} \quad P_2(1)=0\cdot 0.2+1\cdot 0.1=0.1$$

Вероятности состояний после второго шага (после вторых суток)

$$P_1(3)=P_1(2)\cdot P_{11}+P_2(2)\cdot P_{21} \quad P_1(3)=0.81\cdot 0.8+0.19\cdot 0.9=0.819$$
$$P_2(3)=P_1(2)\cdot P_{12}+P_2(2)\cdot P_{22} \quad P_2(3)=0.81\cdot 0.2+0.19\cdot 0.1=0.181$$

рассчитываются следующим образом:

Вероятности состояний после третьего шага (после третьих суток) равны:

$$P_1(2)=P_1(1)\cdot P_{11}+P_2(1)\cdot P_{21} \quad P_1(2)=0.9\cdot 0.8+0.1\cdot 0.9=0.81$$
$$P_2(2)=P_1(1)\cdot P_{12}+P_2(1)\cdot P_{22} \quad P_2(2)=0.9\cdot 0.2+0.1\cdot 0.1=0.19$$

Рассматривая марковские процессы с дискретными состояниями и непрерывным временем, нам удобно будет представлять себе, что все переходы системы S из состояния в состояние происходят под действием каких-то потоков событий (поток вызовов, поток отказов, нoток восстановлений и т. д.). Если все потоки событий, переводящие систему S из состояния в состояние, — простейшие, то процесс, протекающий в системе, будет марковским. Это и естественно, так как простейший поток не обладает последействием: в нем «будущее» не зависит от «прошлого».

Если система S находится в каком-то состоянии S_i, из которого есть непосредственный переход в другое состояние S_j (стрелка, ведущая из S_i в S_j на графе состояний), то мы себе это будем представлять так, как будто на систему, пока она находится в состоянии S_i, действует простейший поток событий, переводящий ее по стрелке $S_i \rightarrow S_j$. Как только появится первое событие этого потока, происходит «перескок» системы из S_i в S_j.

Для наглядности очень удобно на графе состояний у каждой стрелки проставлять интенсивность того потока событий, который переводит систему по данной стрелке. Обозначим λ_{ij} интенсивность потока событий, переводящего систему из состояния S_i в S_j. На рис. 14,а дан граф состояний с проставленными у стрелок интенсивностями (мы будем называть такой граф размеченным).

Построим размеченный граф состояний для системы (техническое устройство из двух узлов). Напомним состояния системы:

S_0 — оба узла исправны,

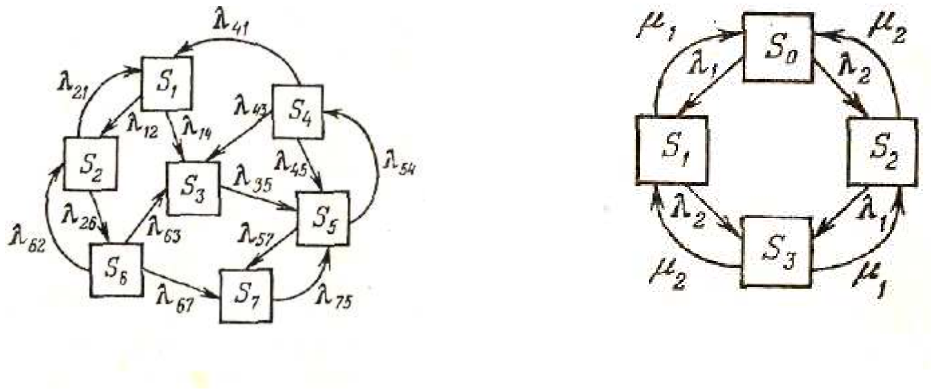
S_1 — первый узел ремонтируется, второй исправен,

S_2 — второй узел ремонтируется, первый исправен,

Простейший характер потоков — достаточное, но не необходимое условие для того, чтобы процесс был марковским.

S_3 — оба узла ремонтируются.

Интенсивности потоков событий, переводящих систему из состояния в состояние, будем вычислять, предполагая, что среднее время ремонта узла не зависит от того, ремонтируется ли один узел или оба сразу.



а) б)

Рис. 14.Граф состояний системы.

Это будет именно так, если ремонтом каждого узла занят отдельный специалист. Найдем все интенсивности потоков событий, переводящих систему из состояния в состояние. Пусть система находится в состоянии S_0. Какой поток событий переводит ее в состояние S_1? Очевидно, поток отказов первого узла. Его интенсивность λ_1 равна единице, деленной на среднее время безотказной работы первого узла. Какой поток событий переводит систему обратно из S_1 в S_0? Очевидно, поток «окончаний ремонтов» первого узла. Его интенсивность μ_1 равна единице, деленной на среднее время ремонта первого узла. Аналогично вычисляются интенсивности потоков событий, переводящих систему по всем стрелкам графа рис. 14, б.

Имея в своем распоряжении размеченный граф состояний системы, легко построить математическую модель данного процесса.

В самом деле, пусть рассматривается система S, имеющая *n* возможных состояний S_1, S_2, ..., S_n. Назовем вероятностью *i*-го состояния вероятность $p_i(t)$ того, что в момент *t* система будет находиться в состоянии S_i. Очевидно, что для любого момента сумма всех вероятностей состояний равна единице:

$$\sum_{i=1}^{n} p_i(t) = 1 \qquad (29)$$

Имея в своем распоряжений размеченный граф состояний, можно найти все вероятности состояний $p_i(t)$ как функции времени. Для этого составляются и решаются так называемые уравнения Колмогорова — особого вида дифференциальные уравнения, в которых неизвестными-функциями являются вероятности состояний.

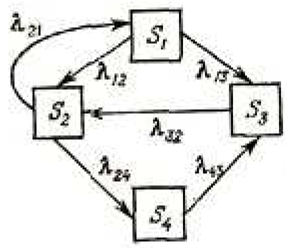


Рис. 15. Граф состояний системы.

Покажем на конкретном примере, как эти уравнения составляются. Пусть система S имеет четыре состояния: S_1, S_2, S_3, S_4, размеченный граф которых показан на рис. 15. Рассмотрим одну из вероятностей состояний, например $p_1(t)$. Это — вероятность того, что в момент t система будет в состоянии S_1. Придадим t малое приращение Δt и найдем $p_1(t+\Delta t)$ — вероятность того, что в момент $t+\Delta t$ система будет в состоянии S_1. Как это может произойти? Очевидно, двумя способами: либо 1) в момент t система уже была в состоянии S_1, а за время Δt не вышла из него; либо 2) в момент t система была в состоянии S_2, а за время Δt перешла из него в S_1. Найдем вероятность первого варианта. Вероятность того, что в момент t система была в состоянии S_1, равна $p_1(t)$. Эту вероятность нужно умножить на вероятность того, что, находившись в момент t в состоянии S_1, система за время Δt не перейдет из него ни в S_2, ни в S_3. Суммарный поток событий, выводящий систему из состояния S_1, тоже будет простейшим, с интенсивностью $\lambda_{12} + \lambda_{13}$ (при наложении суперпозиции двух простейших потоков получается опять простейший поток, так как свойства стационарности, ординарности и отсутствия последействия сохраняются). Значит, вероятность того, что за время Δt система выйдет из состояния S_1, равна $(\lambda_{12} + \lambda_{13})\,\Delta t$; вероятность того, что не выйдет: $1-(\lambda_{12}+\lambda_{13})\,\Delta t$. Отсюда вероятность первого варианта равна $p_1(t)\,[1-(\lambda_{12}+\lambda_{13})\,\Delta t]$.

Найдем вероятность второго варианта. Она равна вероятности того, что в момент t система будет в состоянии S_2, а за время Δt перейдет из него в состояние S_1, т, е. она равна $p_2(t)\lambda_{21}\Delta t$.

Складывая вероятности обоих вариантов (по правилу сложения вероятностей), получим:

$$p_1(t+\Delta t)=p_1(t)\left[1-(\lambda_{12}+\lambda_{13})\Delta t\right]+p_2(t)\lambda_{21}\Delta t \tag{30}$$

Раскроем квадратные скобки, перенесем $p_1(t)$ в левую часть и разделим обе части на Δt:

$$\frac{p_1(t+\Delta t)-p_1(t)}{\Delta t}=\lambda_{21}p_2(t)-(\lambda_{12}+\lambda_{13})p_1(t) \tag{31}$$

Устремим, как и полагается в подобных случаях, Δt к нулю; слева получим в пределе производную функции $p_1(t)$. Таким образом, запишем дифференциальное уравнение для $p_1(t)$:

$$\frac{dp_1(t)}{dt}=\lambda_{21}p_2(t)-(\lambda_{12}+\lambda_{13})p_1(t) \tag{32}$$

или, короче, отбрасывая аргумент t у функций p_1, p_2 (теперь он нам больше уже не нужен):

$$\frac{dp_1}{dt}=\lambda_{21}p_2-(\lambda_{12}+\lambda_{13})p_1 \tag{33}$$

Рассуждая аналогично для всех остальных состояний, напишем еще три дифференциальных уравнения. Присоединяя к ним уравнение (33), получим систему дифференциальных уравнений для вероятностей состояний:

$$
\begin{cases}
\dfrac{dp_1}{dt} = \lambda_{21} p_2 - (\lambda_{12} + \lambda_{13}) p_1 \\
\dfrac{dp_2}{dt} = \lambda_{12} p_1 + \lambda_{32} p_3 - (\lambda_{24} + \lambda_{21}) p_2 \\
\dfrac{dp_3}{dt} = \lambda_{31} p_1 + \lambda_{43} p_4 - \lambda_{32} p_3 \\
\dfrac{dp_4}{dt} = \lambda_{24} p_2 - \lambda_{43} p_4
\end{cases} \qquad (34)
$$

Это — система четырех линейных дифференциальных уравнений с четырьмя неизвестными функциями p_1, p_2, p_3, p_4. Заметим, что одно из них (любое) можно отбросить, пользуясь тем, что $p_1+p_2+p_3+p_4=1$: выразить любую из вероятностей p_i через другие, это выражение подставить в (34), а соответствующее уравнение с производной $\frac{dp_i}{dt}$ отбросить.

Сформулируем теперь общее правило составления уравнений Колмогорова. В левой части каждого из них стоит производная вероятности какого-то (i-го) состояния. В правой части — сумма произведений вероятностей всех состояний, *из которых идут стрелки в данное состояние,* на интенсивности соответствующих потоков событий, минус суммарная интенсивность всех потоков, *выводящих систему из данного состояния,* умноженная на вероятность данного (1-го) состояния.

Пользуясь этим правилом, запишем уравнения Колмогорова для системы S, размеченный граф состояний которой дан на рис. 15:

$$
\begin{cases}
\dfrac{dp_0}{dt} = \mu_1 p_1 + \mu_2 p_2 - (\lambda_1 + \lambda_2) p_0 \\
\dfrac{dp_1}{dt} = \lambda_1 p_0 + \mu_2 p_3 - (\lambda_2 + \mu_1) p_1 \\
\dfrac{dp_3}{dt} = \lambda_2 p_0 + \mu_4 p_3 - (\lambda_1 + \mu_2) p_2 \\
\dfrac{dp_4}{dt} = \lambda_2 p_1 + \lambda_2 p_2 - (\mu_2 + \mu_1) p_3
\end{cases} \qquad (35)
$$

Чтобы решить уравнения Колмогорова и найти вероятности состояний, прежде всего надо задать начальные условия. Если мы точно знаем начальное состояние системы S_i, то в начальный момент (при t = 0) $p_i(0) = 1$, а все остальные начальные вероятности равны нулю. Так, например, уравнения (35) естественно решать при начальных условиях $p_0(0) = 1$, $p_1(0) = p_2(0) = p_3(0) =1$ (в начальный момент оба узла исправны).

Как решать подобные уравнения? Вообще говоря, линейные дифференциальные уравнения с постоянными коэффициентами можно решать аналитически, но это удобно только, когда число уравнений не превосходит двух (иногда — трех). Если уравнений больше, обычно их решают численно — вручную или на ЭВМ. Таким образом, уравнения Колмогорова дают возможность найти все вероятности состояний как функции времени.

Поставим теперь вопрос: что будет происходить с вероятностями состояний при $t \to \infty$? Будут ли $p_1(t)$, $p_2(t)$,… стремиться к каким-то пределам? Если эти пределы существуют и не зависят от начального состояния системы, то они называются финальными вероятностями состояний. В теории случайных процессов доказывается, что если число n состояний системы конечно и из каждого из них можно (за конечное число шагов) перейти в любое другое, то финальные вероятности существуют).

Предположим, что это условие выполнено и финальные вероятности существуют:

$$\lim_{t \to \infty} p_i(t) = p_i \quad (i = 1, 2..., n) \tag{36}$$

Финальные вероятности мы будем обозначать теми же буквами p_1, p_2,..., что и сами вероятности состояний, но разумея под ними уже не переменные величины (функции времени), а постоянные числа. Очевидно, они тоже образуют в сумме единицу:

$$\sum_{i=1}^{n} p_i(t) = 1 \qquad (37)$$

Как понимать эти финальные вероятности? При $t \to \infty$ в системе S устанавливается предельный стационарный режим, в ходе которого система случайным образом меняет свои состояния, но их вероятности уже не зависят от времени. Финальную вероятность состояния S_i, можно истолковать, как среднее относительное время пребывания системы в этом состоянии. Например, если система S имеет три состояния S_1, S_2, S_3 и их финальные вероятности равны 0,2, 0,3 и 0,5, это значит, что в предельном, стационарном режиме система в среднем две десятых времени проводит в состоянии S_1, три десятых — в состоянии S_2, и половину времени — в состоянии S_3.

Как же вычислить финальные вероятности? Очень просто. Если вероятности p_1, p_2,... постоянны, то их производные равны пулю. Значит, чтобы найти финальные вероятности, нужно все левые части в уравнениях Колмогорова положить равными нулю и решить полученную систему уже не дифференциальных, а линейных алгебраических уравнений. Можно и не писать уравнений Колмогорова, а прямо по графу состояний написать систему линейных алгебраических уравнений. Если перенести отрицательный член каждого уравнения из правой части в левую, то получим сразу систему уравнений, где слева стоит финальная вероятность данного состояния p_i,,умноженная на суммарную интенсивность всех потоков, ведущих из данного состояния, а справа — сумма произведений интенсивностей всех потоков, входящих в i-е состояние, на вероятности тех состояний, из которых эти потоки исходят.

Пользуясь этим правилом, напишем линейные алгебраические уравнения для финальных вероятностей состояний системы, граф состояний которой дан на рис. 5.

$$
\begin{cases}
(\lambda_1+\lambda_2)p_0 = \mu_1 p_1 + \mu_2 p_2 \\
(\lambda_2+\mu_1)p_1 = \lambda_1 p_0 + \mu_2 p_3 \\
(\lambda_1+\mu_2)p_2 = \lambda_2 p_0 + \mu_4 p_3 \\
(\mu_2+\mu_1)p_3 = \lambda_2 p_1 + \lambda_2 p_2
\end{cases}
\qquad (38)
$$

Эту систему четырех уравнений с четырьмя неизвестными $p_0, p_1, p_2, p_3,$ казалось бы, вполне можно решить. Но уравнения (38) однородны (не имеют свободного члена) и, значит, определяют неизвестные только с точностью до произвольного множителя. К счастью, мы можем воспользоваться так называемым нормировочным условием:

$$p_0 + p_1 + p_2 + p_3 = 1 \qquad (39)$$

и с его помощью решить систему. При этом одно (любое) из уравнений можно отбросить (оно вытекает как следствие из остальных).

Давайте зададимся численными значениями интенсивностей $\lambda_1=1$, $\lambda_2=2$, $\mu_1=2$, $\mu_3=3$ и решим систему (14). Пожертвуем четвертым уравнением, добавив вместо него нормировочное условно (15). Уравнения примут вид:

$$
\left.
\begin{array}{l}
3p_0 = 2p_1 + 3p_2 \\
4p_1 = p_0 + 3p_3 \\
4p_2 = 2p_0 + 2p_2 \\
p_0 + p_1 + p_2 + p_3 = 1
\end{array}
\right\}
$$

Решая их, получим:

$p_0 = {}^{6}\!/_{15} = 0,40;\quad p_1 = {}^{3}\!/_{15} = 0,20;\quad p_2 = {}^{4}\!/_{15} \approx 0,27;\quad p_3 = {}^{2}\!/_{15} \approx 0,13;$

т. е. в предельном, стационарном режиме система S в среднем 40% времени будет проводить в состоянии S_0 (оба узла исправны), 20% —в состоянии S_1 (первый узел ремонтируется, второй работает), 27% —в состоянии S_2 (второй узел ремонтируется, первый работает) и 13%—в состоянии S_3 полной негодности (оба узла ремонтируются).

$$W = 0.40\times 8 + 0.20\times 3 + 0,27\times 5 = 5.15$$

Теперь оценим загрузку ремонтных органов (рабочих), занятых ремонтом узлов 1 и 2. Узел 1 ремонтируется долю времени, равную

$p_1+p_3=0.20+0.13=0.33$. Узел 2 ремонтируется долю времени $p_2+p_3=0.40$.

Здесь уже может возникнуть вопрос об оптимизации решения. Допустим, что мы можем уменьшить среднее время ремонта того или другого узла (может быть, и того, и другого), но это нам обойдется в какую-то сумму. Спрашивается, выгодно ли это? Т. е. окупит ли увеличение дохода, связанное с ускорением ремонта, повышенные расходы на ремонт?

Задание

1. В моменты времени t1, t2, t3 производится осмотр ЭВМ. Возможны следующие состояния ЭВМ: полностью исправна, незначительные неисправности, существенные неисправности, ЭВМ полностью вышла из строя. Матрица переходных вероятностей имеет вид:

$$P = \begin{vmatrix} 0,5 & 0,3 & 0,3 & 0 \\ 0 & 0,4 & 0,4 & 0,2 \\ 0 & 0 & 0,3 & 0,7 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Построить граф состояний. Найти вероятности состояний ЭВМ после одного, двух и трех просмотров, если в начале (при t=0) ЭВМ была полностью исправна.

2. В городе издаются три журнала С1, С2, С3. Читатели выписывают один из них. Пусть в среднем читатели стремятся поменять журнал, т.е. подписаться на другой не более одного раза в год. Вероятности таких изменений постоянны. Результаты маркетинговых исследований спроса читателей даны следующем процентном соотношении:

80% читателей С1 подписываются на С2;

15% читателей С2 подписываются на С3;

8% читателей С3 подписываются на С1.

Требуется:

1. Записать матрицу переходных вероятностей среднегодовых изменений в подписке;

2. Предположить, что общее число подписчиков в городе будет постоянным и определить, какая доля из их числа будет подписываться на указанные журналы через два года, если по состоянию на 1 января текущего года каждый журнал имел одинаковое число подписчиков.

3. Найти вероятности состояний системы в установившемся режиме и определить журнал, который будет пользоваться наибольшим спросом среди населения.

3. Организация по прокату автомобилей выдает автомобили на прокат в пунктах А, В, С. Клиенты могут возвращать автомобиль в любой из трех пунктов. Анализ процесса возвращения автомобиля в течении года показал, что клиенты возвращают автомобили в пункты проката со следующими вероятностями;

Пункты выдачи	Пункты приема		
	А	В	С
А	0,8	0,2	0
В	0,2	0	0,8
С	0,2	0,2	0,6

Требуется:

1. Найти процентное распределение клиентов к концу года, если в начале года оно было равномерным.
2. Найти вероятности состояний в установившемся режиме.

3. Определить пункт проката, у которого целесообразно строить станцию по ремонту автомобилей.

4. Магазин продает две марки автомобилей A и B. опыт эксплуатации этих марок автомобилей свидетельствует, что для них имеет место различные матрицы переходных вероятностей на годовой период эксплуатации автомобиля, соответствующие состояниям 1- работает хорошо, 2 – требует ремонта.

Автомобиль марки A $P_A = \begin{vmatrix} 0.9 & 0.1 \\ 0.6 & 0.4 \end{vmatrix}$ Автомобиль марки B $P_B = \begin{vmatrix} 0.8 & 0.2 \\ 0.7 & 0.3 \end{vmatrix}$

Требуется:

1. Найти вероятности состояний для каждой марки после двухлетней эксплуатации, если в начале эксплуатации автомобили работают хорошо.
2. Определить марку автомобиля более предпочтительную для приобретения в личное пользование.

5. Размеченный граф состояний системы S имеет вид, показанный на рис. 16. Записать систему дифференциальных уравнений Колмогорова и начальные условия для решения системы, если известно, что в начальный момент система находится в состоянии S_1.

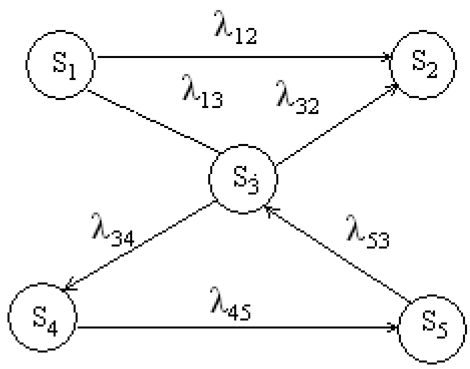


Рис. 16.

6. Граф состояний системы имеет вид, приведенный на рис. 17.

Написать алгебраические уравнения для вероятностей состояний в стационарном режиме и найти выражения для этих вероятностей.

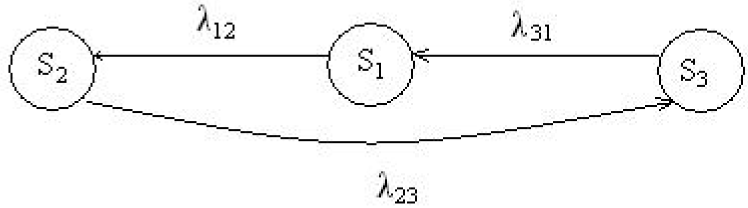


Рис. 17.

7. Найти вероятности состояний в установившемся режиме для процесса гибели и размножения, граф которого представлен на рис. 18.

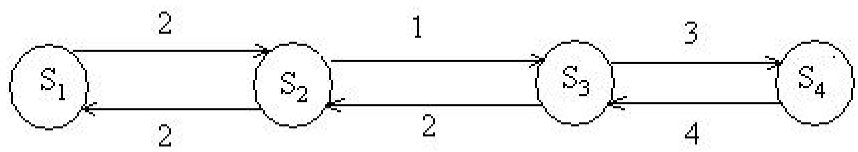


Рис. 18.

8. Граф состояний системы имеет вид, приведенный на рис. 19. Написать алгебраические уравнения для вероятностей состояний в стационарном режиме и найти выражения значения этих вероятностей.

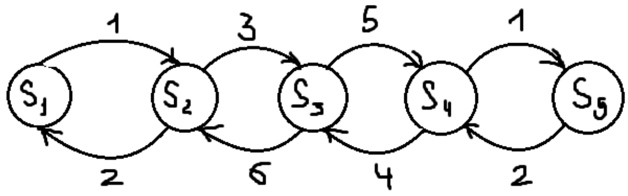


Рис. 19.

9. Граф состояний системы имеет вид, приведенный на рис. 20. Написать алгебраические уравнения для вероятностей состояний в стационарном режиме и найти выражения значения этих вероятностей.

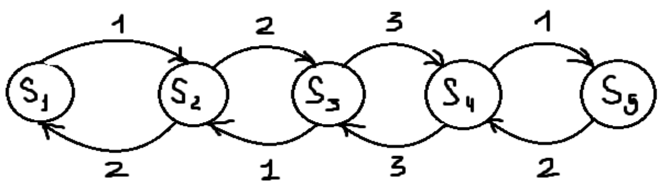


Рис. 20.

10. Граф состояний системы имеет вид, приведенный на рис. 21. Написать алгебраические уравнения для вероятностей состояний в стационарном режиме и найти выражения значения этих вероятностей.

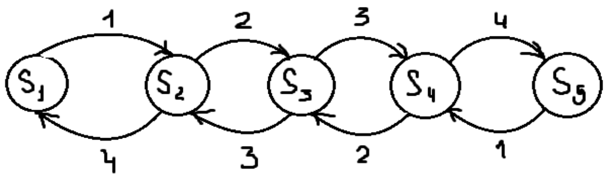


Рис. 21.

11.Граф состояний системы имеет вид, приведенный на рис. 22. Написать алгебраические уравнения для вероятностей состояний в стационарном режиме и найти выражения значения этих вероятностей.

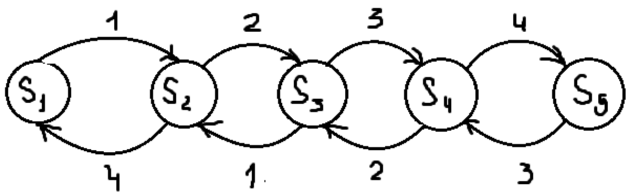


Рис. 22.

12. Граф состояний системы имеет вид, приведенный на рис. 23. Написать алгебраические уравнения для вероятностей состояний в стационарном режиме и найти выражения значения этих вероятностей.

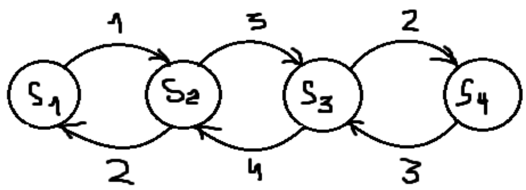


Рис. 23.

13. Граф состояний системы имеет вид, приведенный на рис. 24. Написать алгебраические уравнения для вероятностей состояний в стационарном режиме и найти выражения значения этих вероятностей.

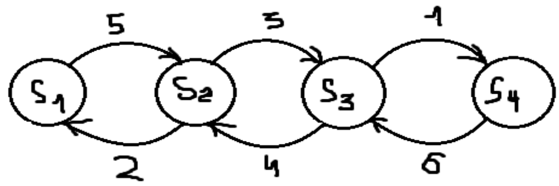


Рис. 24.

14. Граф состояний системы имеет вид, приведенный на рис. 25. Написать алгебраические уравнения для вероятностей состояний в стационарном режиме и найти выражения значения этих вероятностей.

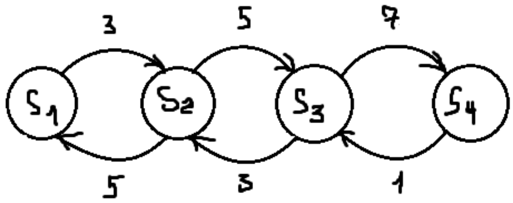


Рис. 25.

Контрольные вопросы

1. Что называется Марковским случайным процессом?
2. Как любой процесс можно превратить в Марковский?
3. Как составляются уравнения Колмогорова?
4. Что такое финальная вероятность состояния?
5. Какие характеристики находятся при решении уравнений Колмогорова?

Практическая работа № 3.
Моделирование систем массового обслуживания.

При исследовании операций часто приходится сталкиваться с работой 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 (или «приборов»), которые мы будем называть каналами обслуживания. Каналами могут быть: линии связи, рабочие точки, кассиры, продавцы, лифты, автомашины и др. СМО могут быть одноканальными и многоканальными.

Всякая СМО предназначена для обслуживания какого-то потока заявок (или «требований»), поступающих в какие-то случайные моменты времени. Обслуживание заявки продолжается какое-то, вообще говоря, случайное время Г„6, после чего канал освобождается и готов к приему следующей заявки. Случайный характер потока заявок и времен обслуживания приводит к тому, что в какие-то периоды времени на входе СМО скапливается излишне большое число заявок (они либо становятся в очередь, либо покидают СМО необслуженными); в другие же периоды СМО будет работать с недогрузкой или вообще простаивать.

Процесс работы СМО представляет собой случайный процесс с дискретными состояниями и непрерывным временем; состояние СМО меняется скачком в моменты появления каких-то событий (или прихода новой заявки, или окончания обслуживания, или момента, когда заявка, которой надоело ждать, покидает очередь).

Предмет теории массового обслуживания — построение математических моделей, связывающих заданные условия работы СМО (число каналов, их производительность, правила работы, характер потока заявок). С интересующими нас характеристиками — показателями эффективности СМО, описывающими, с той или другой точки зрения, ее способность справляться с потоком заявок. В качестве таких показателей (в зависимости от обстановки и целей исследования) могут применяться разные величины, например: среднее число заявок, обслуживаемых СМО в единицу времени; среднее число занятых каналов; среднее число заявок в очереди и среднее время ожидания обслуживания; вероятность того, что число заявок в очереди превысит какое-то значение, и т. д. Среди заданных условий работы СМО мы намеренно не выделяем элементов решения: ими могут быть, например, число каналов, их производительность, режим работы СМО и т. д. Важно уметь решать прямые задачи исследования операций, а обратные ставятся и решаются в зависимости от того, какие именно параметры нам нужно выбирать или изменять. Что касается задач оптимизации, то мы ими здесь почти не будем заниматься, разве только в простейших случаях.

Математический анализ работы СМО очень облегчается, если процесс этой работы — марковский. Мы уже знаем, что для этого достаточно, чтобы все потоки событий, переводящие систему из состояния в состояние (потоки заявок, потоки «обслуживании»), были простейшими. Если это свойство нарушается, то математическое описание процесса становится гораздо сложнее и довести его до явных, аналитических формул удается, лишь в редких случаях. Однако все же аппарат простейшей, марковской теории массового обслуживания может пригодиться для приближенного описания работы СМО даже в тех случаях, когда потоки событий — не простейшие. Во многих случаях для принятия разумного решения по организации работы СМО вовсе и не требуется точного знания всех ее характеристик — зачастую достаточно и приближенного, ориентировочного.

Системы массового обслуживания делятся на типы (или классы) по ряду признаков. Первое деление; СМО с отказами и СМО с очередью. В СМО с отказами заявка, поступившая в момент, когда все каналы заняты, получает отказ, покидает СМО и в дальнейшем процессе обслуживания не участвует. Примеры СМО с отказами встречаются в телефонии: заявка на разговор, пришедшая в момент, когда все каналы связи заняты, получает отказ и покидает СМО необслуженной. В СМО с очередью заявка, пришедшая в момент, когда все каналы -заняты, не уходит, а становится в очередь и ожидает возможности быть обслуженной. На практике чаще встречаются (и имеют большее значение) СМО с очередью; недаром теория массового обслуживания имеет второе название: «теория очередей».

СМО с очередью подразделяются на разные виды, в зависимости от того, как организована очередь — ограничена она или не ограничена. Ограничения могут касаться как длины очереди, так и времени ожидания (так называемые «СМО с нетерпеливыми заявками»). При анализе СМО должна учитываться также и «дисциплина обслуживания» — заявки могут обслуживаться либо в порядке поступления (раньше пришла, раньше обслуживается), либо в случайном порядке. Нередко встречается так называемое обслуживание с приоритетом — некоторые заявки обслуживаются вне очереди. Приоритет может быть как абсолютным — когда заявка с более высоким приоритетом «вытесняет» из-под обслуживания заявку с низшим (например, пришедший в парикмахерскую клиент высокого ранга прогоняет с кресла обыкновенного клиента), так и относительным — когда начатое обслуживание доводится до конца, а заявка с более высоким приоритетом имеет лишь право на лучшее место в очереди.

Существуют СМО с так называемым многофазовым обслуживанием, состоящим из нескольких последовательных этапов или «фаз» (например, покупатель, пришедший в магазин, должен сначала выбрать товар, затем оплатить его в кассе, затем получить на контроле).

Кроме этих признаков, СМО делятся на два класса: «открытые» и «замкнутые». В открытой СМО характеристики потока заявок не зависят от того, в каком состоянии сама СМО (сколько каналов занято), В замкнутой СМО — зависят. Например, если один рабочий обслуживает группу станков, время от времени требующих наладки, то интенсивность потока «требований» со стороны станков зависит от того, сколько их уже неисправно и ждет наладки. Это — пример замкнутой СМО. Классификация ОМО далеко не ограничивается приведенными их разновидностями, но мы ограничимся ими.

Граф состояний СМО соответствует схеме гибели и размножения (рис. 26). Разметим этот граф — проставим у стрелок интенсивности потоков событий. Из S_0 в S_1 систему переводит поток заявок с интенсивностью λ (как только приходит заявка, система перескакивает из S_0 в S_1). Тот же поток заявок переводит

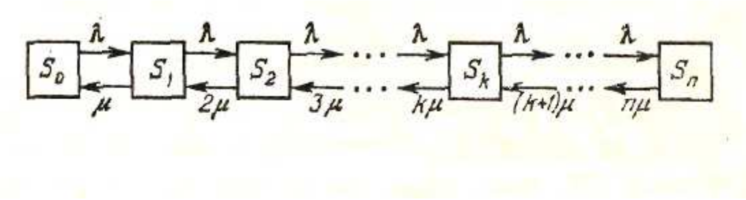


Рис.26. Схема гибели и размножения.

систему из любого левого состояния в соседнее правое (см. верхние стрелки на рис. 26).

Проставим интенсивности у нижних стрелок. Пусть система находится в состоянии S_1 (работает один канал). Он производит μ обслуживаний в единицу времени. Проставляем у стрелки $S_1 \rightarrow S_0$ интенсивность μ. Теперь представим себе, что система находится в состоянии S_2 (работают два канала). Чтобы ей перейти в S_1, нужно, чтобы либо закончил обслуживание первый канал, либо второй; суммарная интенсивность их потоков обслуживании равна 2μ.; проставляем ее у соответствующей стрелки. Суммарный поток обслуживания, даваемый тремя каналами, имеет интенсивность 3μ, k каналами — kμ. Проставляем эти интенсивности у нижних стрелок на рис. 4, а).

А теперь, зная все интенсивности, воспользуемся уже готовыми формулами для финальных вероятностей в схеме гибели и размножения. По формуле (38) получим:

$$p_0 = \left(1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2\mu^2} + \frac{\lambda^3}{2 \cdot 3\mu^3} + \ldots + \frac{\lambda^k}{k!\,\mu^k} + \ldots + \frac{\lambda^n}{n!\,\mu^n}\right)^{-1} \quad (40)$$

Члены разложения $\lambda/\mu,\ \lambda^2/2\mu^2,\ \dots\ \lambda^n/n!\mu^n$ будут представлять собой коэффициенты при p_0 в выражениях для $p_1, p_2, \dots, p_n$:

$$p_1 = \frac{\lambda}{\mu} p_0 \quad p_2 = \frac{\lambda^2}{2\mu^2} p_0 \quad \dots \quad p_k = \frac{\lambda^k}{k!\mu^k} p_0 \quad \dots \quad p_n = \frac{\lambda^n}{n!\mu^n} p_0 \qquad (41)$$

Заметим, что в формулы (40), (41) интенсивности λ и μ входят не по отдельности, а только в виде отношения λ / μ. Обозначим

$$\frac{\lambda}{\mu} = \rho \qquad (42)$$

и будем называть величину р «приведенной интенсивностью потока заявок». Ее смысл — среднее число заявок, приходящее за среднее время обслуживания одной заявки. Пользуясь этим обозначением, перепишем формулы (16), (17) в виде:

$$p_0 = \left(1 + \rho + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} + \dots + \frac{\rho^k}{k!} + \dots + \frac{\rho^n}{n!}\right)^{-1} \qquad (43)$$

$$p_1 = \rho p_0 \quad p_2 = \frac{\rho^2}{2!} p_0 \quad \dots \quad p_k = \frac{\rho^k}{k!} p_0 \quad \dots \quad p_n = \frac{\rho^n}{n!} p_0 \qquad (44)$$

Формулы (43), (44) для финальных вероятностей состояний называются формулами Эрланга — в честь основателя теории массового обслуживания. Большинство других формул этой теории не носит никаких специальных имен.

Таким образом, финальные вероятности найдены. По ним мы вычислим характеристики эффективности СМО. Сначала найдем $P_{отк}$ – вероятность того, что пришедшая заявка получит отказ (не будет обслужена). Для этого нужно, чтобы все n каналов были заняты, значит,

$$P_{отк} = p_n = \frac{\rho^n}{n!} p_0 \qquad (45)$$

Отсюда находим относительную пропускную способность — вероятность того, что заявка будет обслужена:

$$Q = 1 - P_{отк} = 1 - \frac{\rho^n}{n!} p_0 \quad (46)$$

Абсолютную пропускную способность получим, умножая интенсивность потока заявок λ на Q:

$$A = \lambda Q = \lambda \left(1 - \frac{\rho^n}{n!} p_0\right) \quad (47)$$

Осталось только найти среднее число занятых каналов $\bar{k}$. Эту величину можно было бы найти «впрямую», как математическое ожидание дискретной случайной величины с возможными значениями *0, 1, ..., n* и вероятностями $p_0, p_0, \ldots, p_n$,

$$\bar{k} = 0 \cdot p_0 + 1 \cdot p_1 + 2 \cdot p_2 + \ldots + n \cdot p_n$$

Подставляя сюда выражения (20) для p_k $\{k = 0, 1, \ldots, n\}$ и выполняя соответствующие преобразования, мы, в конце концов, получили бы верную формулу для $\bar{k}$. Но мы выведем ее гораздо проще (вот она, одна из «маленьких хитростей»!) В самом деле, нам известна абсолютная пропускная способность *A*. Это — не что иное, как интенсивность потока обслуженных системой заявок. Каждый занятый I канал в единицу времени обслуживает в среднем μ заявок. Значит, среднее число занятых каналов равно

$$\bar{k} = A / \mu$$

или, учитывая (23),

$$\bar{k} = \rho \left(1 - \frac{\rho^n}{n!} p_0\right)$$

Пример. Имеется станция связи с тремя каналами (n = 3), интенсивность потока заявок λ = 1,5 (заявки в минуту); среднее время обслуживания одной заявки $\bar{t} = 2$ (мин.), все потоки событий (как и во всем этом параграфе) — простейшие. Найти финальные вероятности состояний и характеристики эффективности СМО: A, Q, $P_{\text{отк}}$, $\bar{k}$. Результаты:

$$p_0 = 1/13 \quad p_1 = 3/13 \quad p_2 = 9/26 \quad p_3 = 9/26 \approx 0,346$$
$$A \approx 0,346 \quad Q \approx 0,654 \quad P_{отк} \approx 0,346 \quad \bar{k} \approx 1,96$$

Из ответов видно, что наша СМО в значительной мере перегружена: из трех каналов занято в среднем около двух, а из поступающих заявок около 35% остаются необслуженными. Предлагаем читателю, если он любопытен и неленив, выяснить: сколько потребуется каналов для того, чтобы удовлетворить не менее 80% поступающих заявок? И какая доля каналов при этой будет простаивать?

Тут уже проглядывает некоторый намек на *оптимизацию*. В самом деле, содержание каждого канала в единицу времени обходится в какую-то сумму. Вместе с тем, каждая обслуженная заявка приносит какой-то доход. Умножая этот доход на среднее число заявок А, обслуживаемых в единицу времени, мы получим средний доход от СМО в единицу времени. Естественно, при увеличении числа каналов этот доход растет, но растут и расходы, связанные с содержанием каналов. Что перевесит — увеличение доходов или расходов? Это зависит от условий операции, от «платы за обслуживание заявки» и от стоимости содержания канала. Зная эти величины, можно найти оптимальное число каналов, наиболее эффективное экономически. Мы такой задачи решать не будем, предоставляя все тому же «неленивому и любопытному читателю» придумать пример и решить. Вообще, придумывание задач больше развивает, чем решение уже поставленных кем-то.

Одноканальная СМО с неограниченной очередью. На практике довольно часто встречаются одноканальные СМО с очередью (врач, обслуживающий пациентов; телефон-автомат с одной будкой; ЭВМ, выполняющая заказы пользователей). В теории массового обслуживания одноканальные СМО с очередью также занимают особое место (именно к таким СМО относится большинство полученных до сих пор аналитических формул для немарковских систем). Поэтому мы уделим одноканальной СМО с очередью особое внимание.

Пусть имеется одноканальная СМО с очередью, на которую не наложено никаких ограничений (ни по длине очереди, ни по времени ожидания). На эту СМО поступает поток заявок с интенсивностью λ; поток обслуживании имеет интенсивность μ, обратную среднему времени обслуживания заявки $t_{об}$. Требуется найти финальные вероятности состояний СМО, а также характеристики ее эффективности:

$L_{сист}$ — среднее число заявок в системе,

$W_{сист}$ — среднее время пребывания заявки в системе,

$L_{оч}$ — среднее число заявок в очереди,

$W_{оч}$ — среднее время пребывания заявки в очереди,

$P_{зан}$ — вероятность того, что канал занят (степень загрузки канала).

Что касается абсолютной пропускной способности *A* и относительной Q, то вычислять их нет надобности: в силу того, что очередь неограниченна, каждая заявка рано или поздно будет обслужена, поэтому A=λ, по той же причине Q=1.

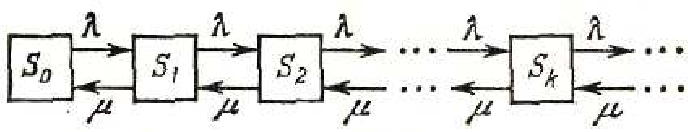


Рис. 27. Граф состояний одноканальной СМО с неограниченной очередью.

Р е ш е н и е. Состояния системы, как и раньше, будем нумеровать по числу заявок, находящихся в СМО:

S_0 — канал свободен,

S_1— канал занят (обслуживает заявку), очереди нет,

S_2 — капал занят, одна заявка стоит в очереди,

. . . .

S_k — канал занят, k-1 заявок стоят в очереди,

. . . .

Теоретически число состояний ничем не ограничено (бесконечно). Граф состояний имеет вид, показанный на рис. 27, Это — схема гибели и размножения, но с бесконечным числом состояний. По всем стрелкам поток заявок с интенсивностью λ переводит систему слева направо, а справа налево — поток обслуживания с интенсивностью μ.

Прежде всего, спросим себя, а существуют ли в этом случае финальные вероятности? Ведь число состояний системы бесконечно, и, в принципе, при $t \to \infty$ очередь может неограниченно возрастать! Да, так оно и есть: финальные вероятности для такой СМО существуют не всегда, а только когда система не перегружена. Можно доказать, что если р строго меньше единицы ($p<1$), то финальные вероятности существуют, а при $p \geq 1$ очередь при $t \to \infty$ растет неограниченно. Особенно «непонятным» кажется этот факт при $p=1$. Казалось бы, к системе не предъявляется невыполнимых требований: за время обслуживания одной заявки приходит в среднем одна заявка, и все должно быть в порядке, а вот на деле — не так. При $p = 1$ СМО справляется с потоком заявок, только если поток этот — регулярен, но время обслуживания — тоже не случайное, равное интервалу между заявками. В атом «идеальном» случае очереди в СМО вообще не будет, канал будет непрерывно занят и будет регулярно выпускать обслуженные заявки. Но стоит только потоку заявок или потоку обслуживании стать хотя бы чуточку случайными — и очередь уже будет расти до бесконечности. На практике этого не происходит только потому, что «бесконечное число заявок в очереди» — абстракция. Вот к каким грубым ошибкам может привести замена случайных величин их математическими ожиданиями!

Но вернемся к нашей одноканальной СМО с неограниченной очередью. Строго говоря, формулы для финальных вероятностей в схеме гибели и размножения выводились нами только для случая конечного числа состояний, но позволим себе вольность — воспользуемся ими и для бесконечного числа состояний.

Подсчитаем финальные вероятности состояний по формулам ($p_0 = (1 + \frac{\lambda_{01}}{\lambda_{10}} + \frac{\lambda_{12}\lambda_{01}}{\lambda_{21}\lambda_{10}} + \cdots + \frac{\lambda_{n-1,n}\cdots\lambda_{12}\lambda_{01}}{\lambda_{n,n-1}\cdots\lambda_{21}\lambda_{10}})^{-1}$) и ($p_k = \frac{\lambda_{k-1,k}\cdots\lambda_{12}\lambda_{01}}{\lambda_{k,k-1}\cdots\lambda_{21}\lambda_{10}} p_0$). В нашем случае число слагаемых в формуле ($p_0 = (1 + \frac{\lambda_{01}}{\lambda_{10}} + \frac{\lambda_{12}\lambda_{01}}{\lambda_{21}\lambda_{10}} + \cdots + \frac{\lambda_{n-1,n}\cdots\lambda_{12}\lambda_{01}}{\lambda_{n,n-1}\cdots\lambda_{21}\lambda_{10}})^{-1}$) будет бесконечным. Получим выражение для p_0

$$p_0 = \left(1 + \lambda/\mu + (\lambda/\mu)^2 + \ldots + (\lambda/\mu)^k + \ldots\right) = \left(1 + \rho + \rho^2 + \ldots + \rho^k + \ldots\right) \quad (48)$$

Ряд в формуле (48) представляет собой геометрическую прогрессию. Мы знаем, что при $\rho<1$ ряд сходится — это бесконечно убывающая геометрическая прогрессия со знаменателем ρ. При $\rho\geq1$ ряд расходится (что является косвенным, хотя и не строгим доказательством того, что финальные вероятности состояний p_0, p_1, ..., p_k, ... существуют только при $\rho<1$. Теперь предположим, что это условие выполнено, и $\rho<1$. Суммируя прогрессию в (48), имеем

$$1 + \rho + \rho^2 + \ldots + \rho^k + \ldots = \frac{1}{1-\rho}$$

откуда

$$p_0 = 1 - \rho \quad (49)$$

Вероятности $p_0, p_1, \ldots, p_k, \ldots$ найдутся по формулам:

$$p_1 = \rho p_0 \quad p_2 = \rho^2 p_0 \quad \ldots \quad p_k = \rho^k p_0 \quad \ldots$$

откуда, с учетом (49), найдем окончательно:

$$p_1 = \rho(1-\rho) \quad p_2 = \rho^2(1-\rho) \quad \ldots \quad p_k = \rho^k(1-\rho) \quad \ldots \quad (50)$$

Как видно, вероятности p_0, p_1, ..., p_k, ...образуют геометрическую прогрессию со знаменателем p. Как это ни странно, максимальная из них p_0—вероятность того, что канал будет вообще свободен. Как бы ни была нагружена система с очередью, если только она вообще справляется с потоком заявок ($p < 1$), самое вероятное число заявок в системе будет 0.

Найдем среднее число заявок в СМО $L_{сист}$. Тут придется немного повозиться. Случайная величина Z — число заявок в системе -— имеет возможные значения $0, 1, 2, ..., k, ...$ с вероятностями p_0, p_1, ..., p_k, ... Ее математическое ожидание равно

$$L_{сист} = 0 \cdot p_0 + 1 \cdot p_1 + 2 \cdot p_2 + ... + k \cdot p_k + ... = \sum_{k=1}^{\infty} k p_k \qquad (51)$$

(сумма берется не от 0 до ∞, а от 1 до ∞, так как нулевой член равен нулю).

Подставим в формулу (27) выражение для p_k (26):

$$L_{сист} = \sum_{k=1}^{\infty} k\rho^k (1-\rho)$$

Теперь вынесем за знак суммы $\rho(1\text{-}\rho)$:

$$L_{сист} = \rho(1-\rho)\sum_{k=1}^{\infty} k\rho^{k-1}$$

Тут мы опять применим «маленькую хитрость»: $k\rho^{k-1}$ есть не что иное, как производная по ρ от выражения ρ^k; значит,

$$L_{сист} = \rho(1-\rho)\sum_{k=1}^{\infty} \frac{d}{d\rho}\rho^k$$

Меняя местами операции дифференцирования и суммирования, получим:

$$L_{сист} = \rho(1-\rho)\frac{d}{d\rho}\sum_{k=1}^{\infty}\rho^k \qquad (52)$$

Но сумма в формуле (52) есть не что иное, как сумма бесконечно убывающей геометрической прогрессии с первым членом ρ и знаменателем ρ; эта сумма равна $\frac{\rho}{1-\rho}$, а ее производная $\frac{1}{(1-\rho)^2}$. Подставляя это выражение в (28), получим:

$$L_{сист} = \frac{\rho}{(1-\rho)} \tag{53}$$

Теперь применим формулу Литтла ($W_{сист} = \frac{1}{\lambda} L_{сист}$). Найдем среднее время пребывания заявки в системе:

$$W_{сист} = \frac{\rho}{\lambda(1-\rho)} \tag{54}$$

Найдем среднее число заявок в очереди $L_{оч}$. Будем рассуждать так: число заявок в очереди равно числу заявок в системе минус число заявок, находящихся под обслуживанием. Значит (по правилу сложения математических ожиданий), среднее число заявок в очереди $L_{оч}$ равно среднему числу заявок в системе $L_{сист}$ минус среднее число заявок под обслуживанием. Число заявок под обслуживанием может быть либо нулем (если канал свободен), либо единицей (если он занят). Математическое ожидание такой случайной величины равно вероятности того, что канал занят (мы ее обозначили $P_{зан}$), Очевидно, $P_{зан}$ равно единице минус вероятность p_0 того, что канал свободен:

$$P_{зан} = 1 - p_0 \tag{55}$$

Следовательно, среднее число заявок под обслуживанием равно

$$L_{об} = \rho \tag{56}$$

отсюда

$$L_{оч} = L_{сист} - \rho = \frac{\rho}{1-\rho} - \rho$$

и окончательно

$$L_{оч} = \frac{\rho^2}{1-\rho}$$

По формуле Литтла найдем среднее время пребывания заявки в очереди

$$W_{оч} = \frac{\rho^2}{\lambda(1-\rho)} \qquad (57)$$

Задание

1. В бухгалтерии предприятия имеются два кассира, каждый из которых может обслужить в среднем 30 сотрудников в час. Поток сотрудников простейший с интенсивностью 40 сотрудников в час. Очередь не ограничена. Определить целесообразность приема на работу третьего кассира.

2. В аудиторскую фирму поступает поток заявок с интенсивностью λ=1,5 заявки в день. Среднее время обслуживания 3 дня. Аудиторская фирма располагает пятью независимыми бухгалтерами, выполняющих обслуживание клиентов. Очередь заявок не ограничена. Найти вероятностные характеристики аудиторской фирмы как СМО, работающей в стационарном режиме.

3. В магазине работает один продавец, который обслуживает в среднем 30 покупателей в час. Интенсивность потока покупателей 60 покупателей в час. Покупатели уходят, если в очереди более 5 человек. Определить Вероятность обслуживания покупателя; среднюю длину очереди; среднее время ожидания в очереди; среднее время всего обслуживания; вероятность простоя продавца.

4. Имеется двухканальная СМО с отказами. На ее вход поступает поток заявок с интенсивностью λ=3 заявки в час. Среднее время обслуживания одной заявки 0,5 час. Каждая обслуженная заявка приносит

доход 5д.е.. Содержание канала обходится 3д.е/час.. Решите выгодно ли увеличить число каналов до трех.

5. На пункт техосмотра поступает простейший поток заявок автомобилей с интенсивностью λ=4 машины в час. Время осмотра в среднем 17 мин. Очередь не ограничена. Определить вероятностные характеристики пункта осмотра в установившемся режиме.

6. Пост диагностики представляет собой трехканальную СМО с отказами. Интенсивность потока заявок на диагностику λ=0,5 автомобиля в час. Средняя продолжительность диагностики 1,2 часа. Определить вероятностные характеристики пункта диагностики в установившемся режиме.

7. На железнодорожную сортировочную горку прибывают составы с интенсивностью λ=2 состава в час. Среднее время обслуживания состава 0,4 часа. Составы, прибывающие, когда горка занята, становятся в очередь, не имеющую ограничений. При установившемся режиме найти : среднее число составов, ожидающих в очереди; среднее время пребывания состава в очереди; среднее время пребывания состава в системе.

8. Автозаправочная станция представляет собой СМО с двумя каналами. Площадка при АЗС допускает пребывание в очереди не более 3 автомобилей. Поток автомобилей прибывающих на заправку имеет интенсивность λ=0,7 автомобилей в мин. Процесс заправки в среднем продолжается 1,25 мин. Определить вероятностные характеристики пункта осмотра в установившемся режиме.

9. Одноканальная СМО – ЭВМ, на которую поступает поток заданий со средним интервалом между заданиями 10 мин. Среднее время обслуживания заявки 8 мин. Определить среднее число заявок в СМО; среднее число заявок в очереди; среднее время пребывания заявки в системе и в очереди.

10. В бухгалтерии предприятия работает два кассира, каждый из которых может обслужить 30 сотрудников в час. Поток сотрудников 40 чел в час. Очередь в кассу не ограничена. Определить целесообразность приема третьего кассира.

11. На пункт техосмотра поступает простейший поток заявок автомобилей с интенсивностью λ=4 машины в час. Время осмотра в среднем 17 мин. В очереди может находиться не более 5 автомобилей. Определить вероятностные характеристики пункта осмотра в установившемся режиме.

12. Одноканальная СМО с отказами представляет собой одну телефонную линию. Заявка, пришедшая в момент, когда линия занята, получает отказ. Все потоки событий простейшие. Интенсивность потока – 0,95 вызовов в мин. Средняя продолжительность разговора - 1 мин. Определить вероятностные характеристики СМО в установившемся режиме работы.

13. В одноканальную СМО с отказами поступает простейший поток заявок с интенсивностью 0,5 заявки в мин. Время обслуживания имеет показательное распределение со средним временем 1,5мин. Определить вероятностные характеристики СМО в установившемся режиме работы.

14. Пусть в филиале банка X города N постоянно работает три оператора. Если клиент заходит в банк, когда все операторы заняты, то он сразу уходит, не ожидая обслуживания. Среднее число клиентов, обращающихся за один час, составляет 24 человека. Среднее время, затрачиваемое на обслуживание одного клиента 5 минут. Определить основные характеристики эффективности функционирования филиала банка в предельном режиме: вероятность того, что клиент получит отказ, среднее число клиентов, обслуживаемых в течении часа.

Контрольные вопросы

1. Что является главной составной частью СМО?

2. Для чего предназначена СМО?
3. На какие классы делятся СМО?
4. Каков порядок обслуживания в СМО с отказами?
5. Что такое замкнутая СМО?
6. Какие характеристики рассчитываются при анализе СМО?

Практическое занятие № 3.

Методы и модели описания систем

Методы описания систем классифицируются в порядке возрастания формализованности - от качественных методов, с которыми в основном и связан был первоначально системный анализ, до количественного системного моделирования с применением ЭВМ. Разделение методов на качественные и количественные носит, конечно, условный характер.

1. В качественных методах основное внимание уделяется организации постановки задачи, новому этапу ее формализации, формированию вариантов, выбору подхода к оценке вариантов, использованию опыта человека, его предпочтений, которые не всегда могут быть выражены в количественных оценках.

2. Количественные методы связаны с анализом вариантов, с их количественными характеристиками корректности, точности и т. п. Для постановки задачи эти методы не имеют средств, почти полностью оставляя осуществление этого этапа за человеком.

Между этими крайними классами методов системного анализа имеются методы, которые стремятся охватить оба этапа — этап постановки задачи, разработки вариантов и этап оценки и количественного анализа вариантов,— но делают это с привлечением разных исходных концепций и терминологии, с разной степенью формализованности. Среди них: кибернетический подход к разработке адаптивных систем управления, проектирования и принятия решений (который исходит из развития основных идей классической теории автоматического регулирования и управления и теории адаптивных систем при- мнительно к организационным системам); информационно-гносеологический подход к моделированию систем (основанный на общности процессов отражения, познания в системах различной физической природы); системно-структурный подход; метод ситуационного моделирования; метод имитационного динамического моделирования.

Качественные методы описания систем

Качественные методы системного анализа применяются, когда отсутствуют описания закономерностей систем в виде аналитических зависимостей.

Методы типа мозговой атаки.

Концепция «мозговой атаки» получила широкое распространение с начала 50-х годов как метод систематической тренировки творческого мышления, нацеленный на открытие новых идей и достижение согласия группы людей на основе интуитивного мышления. Методы этого типа известны также под названиями «мозговой штурм», «конференция идей», а в последнее время наибольшее распространение получил термин «коллективная генерация идей» (КГИ).

Обычно при проведении мозговой атаки или сессий КГИ стараются выполнять определенные правила, суть которых:

1) обеспечить как можно большую свободу мышления участников КГИ и высказывания ими новых идей;

2) приветствуются любые идеи, если вначале они кажутся сомнительными или абсурдными (обсуждение и оценка идей производится позднее);

3) не допускается критика, не объявляется ложной и не прекращается обсуждение ни одной идеи;

4) желательно высказывать как можно больше идей, особенно нетривиальных.

Подобием сессий КГИ можно считать разного рода совещания — конструктораты, заседания научных советов по проблемам, заседания специально создаваемых временных комиссий и другие собрания компетентных специалистов.

Методы типа сценариев.

Методы подготовки и согласования представлений о проблеме или анализируемом объекте, изложенные в письменном виде, получили название сценария. Первоначально этот метод предполагал подготовку текста, содержащего логическую последовательность событий или возможные варианты решения проблемы, развернутые во времени. Однако позднее обязательное требование явно выраженных временных координат было снято, и сценарием стали называть любой документ, содержащий анализ рассматриваемой проблемы или предложения по ее решению, по развитию системы независимо от того, в какой форме он представлен. Как правило, предложения для подготовки подобных документов пишутся вначале индивидуально, а затем формируется согласованный текст.

На практике по типу сценариев разрабатывались прогнозы в некоторых отраслях промышленности. В настоящее время разновидностью сценариев можно считать предложения к комплексным программам развития отраслей народного хозяйства, подготавливаемые организациями или специальными комиссиями.

Сценарий является предварительной информацией, на основе которой проводится дальнейшая работа по прогнозированию развития отрасли или по разработке вариантов проекта. Он может быть подвергнут анализу, чтобы исключить из дальнейшего рассмотрения то, что в учитываемом периоде находится на достаточном уровне развития, если речь идет о прогнозе, или, напротив, то, что не может быть обеспечено в планируемом периоде, если речь идет о проекте. Таким образом, сценарий помогает составить представление о проблеме, а затем приступить к более формализованному представлению системы в виде графиков, таблиц для проведения экспертного опроса и других методов системного анализа.

Методы экспертных оценок.

Термин «эксперт» происходит от латинского слова означающего «опытный». При использовании экспертных оценок обычно предполагается, что мнение группы экспертов надежнее, чем мнение отдельного эксперта. В некоторых теоретических исследованиях отмечается, что это предположение не является очевидным.

Все множество проблем, решаемых методами экспертных оценок, делится на два класса. К первому относятся такие, в отношении которых имеется достаточное обеспечение информацией. При этом методы опроса и обработки основываются на использовании принципа «хорошего измерителя», т. е. эксперт — качественный источник информации; групповое мнение экспертов близко к истинному решению. Ко второму классу относятся проблемы, в отношении которых знаний для уверенности в справедливости указанных гипотез недостаточно. В этом случае экспертов уже нельзя рассматривать как «хороших измерителей» и необходимо осторожно подходить к обработке результатов экспертизы во избежание больших ошибок. В литературе в основном рассматриваются вопросы экспертного оценивания для решения задач первого класса.

При обработке материалов коллективной экспертной оценки используются методы теории ранговой корреляции. Для количественной оценки степени согласованности мнений экспертов применяется коэффициент конкордации

$$W = \frac{12d}{m^2(n^3 - n)}, \tag{58}$$

где

$$d = \sum_{i=1}^{n} d_i^2 = \sum_{i=1}^{n}[\sum_{i=2}^{m} r_{ij} - 0.5m(n+1)]^2$$

и m — количество экспертов, j=$\overline{1,m}$;n — количество рассматриваемых свойств, $i = \overline{1,n}; r_{ij}$ — место, которое заняло i -е свойство в ранжировке j-м экспертом; d_i — отклонение суммы рангов по i -му свойству от среднего арифметического сумм рангов по n свойствам.

Коэффициент конкордации W позволяет оценить, насколько согласованы между собой ряды предпочтительности, построенные каждым экспертом. Его значение находится в пределах0≤W≤1; W=0 означает полную противоположность, а W= 1 —полное совпадение ранжировок. Практически достоверность считается хорошей, если W= 0,7...0,8.

Небольшое значение коэффициента конкордации, свидетельствующее о слабой согласованности мнений экспертов, является следствием следующих причин: в рассматриваемой совокупности экспертов действительно отсутствует общность мнений; внутри рассматриваемой совокупности экспертов существуют группы с высокой согласованностью мнений, однако обобщенные мнения таких групп противоположны.

Для наглядности представления о степени согласованности мнений двух любых экспертов А и В служит коэффициент парной ранговой корреляции

$$\rho_{AB} = 1 - \frac{\sum_{i=1}^{n} \psi_1^2}{\frac{1}{6}(n^3 - n) - \frac{1}{n}(T_A + T_B)}, \tag{59}$$

где ψ_i — разность (по модулю) величин рангов оценок i -го свойства, назначенных экспертами А и В: $\psi_i = \left|R_{A_i} - R_{B_i}\right|$; T_A и T_B—показатели связанных рангов оценок экспертов А и В.

Коэффициент парной ранговой корреляции принимает значения —1< ρ <+1. Значение ρ = +1 соответствует полному совпадению оценок в рангах двух экспертов (полная согласованность мнений двух экспертов), а $\rho = -1$ — двум взаимно противоположным ранжировкам важности свойств (мнение одного эксперта противоположно мнению другого).

Методы типа «Дельфи».

Характерный для середины XX в. бурный рост науки и техники вызвал большие перемены в отношении к оценкам будущего развития систем. Одним из результатов этого периода в развитии методов анализа сложных систем явилась разработка методов экспертной оценки, известных в литературе как «методы Дельфи». Название этих методов связано с древнегреческим городом Дельфи, где при храме Аполлона с IX в. до н.э. до IV в. н.э. по преданиям существовал Дельфийский оракул.

Суть метода Дельфи заключается в следующем. В отличие от традиционного подхода к достижению согласованности мнений экспертов путем открытой дискуссии метод Дельфи предполагает полный отказ от коллективных обсуждений. Это делается для того, чтобы уменьшить влияние таких психологических факторов, как присоединение к мнению наиболее авторитетного специалиста, нежелание отказаться от публично выраженного мнения, следование за мнением большинства. В методе Дельфи прямые дебаты заменены тщательно разработанной программой последовательных индивидуальных опросов, проводимых обычно в форме анкетирования. Ответы экспертов обобщаются и вместе с новой дополнительной информацией поступают в распоряжение экспертов, после чего они уточняют свои первоначальные ответы. Такая процедура повторяется несколько раз до достижения приемлемой сходимости совокупности высказанных мнений. Результаты эксперимента показали приемлемую сходимость оценок экспертов после пяти туров опроса.

Метод Дельфи первоначально был предложен О. Хелмером как итеративная процедура при проведении мозговой атаки, которая должна помочь снизить влияние психологических факторов при проведении повторных заседаний и повысить объективность результатов. Однако почти одновременно Дельфи-процедуры стали основным средством повышения объективности экспертных опросов с использованием количественных оценок при оценке деревьев цели и при разработке сценариев.

Процедура Дельфи-метода:

1) в упрощенном виде организуется последовательность циклов мозговой атаки;

2) в более сложном виде разрабатывается программа последовательных индивидуальных опросов обычно с помощью вопросников, исключая контакты между экспертами, но предусматривающая ознакомление их с мнениями друг друга между турами; вопросники от тура к туру могут уточняться;

3) в наиболее развитых методиках экспертам присваиваются весовые коэффициенты значимости их мнений, вычисляемые на основе предшествующих опросов, уточняемые от тура к туру и учитываемые при получении обобщенных результатов оценок.

Первое практическое применение метода Дельфи к решению некоторых задач Министерства обороны США во второй половине 40-х годов, показало его эффективность и целесообразность распространения на широкий класс задач, связанных с оценкой будущих событий.

Исследуемые проблемы: научные открытия, рост народонаселения, автоматизация производства, освоение космоса, предотвращение войны, военная техника. Результаты статистической обработки мнений экспертов позволили нарисовать вероятную картину будущего мира в указанных шести аспектах. Была оценена также степень согласованности мнений экспертов, которая оказалась приемлемой после проведения четырех туров опроса.

Недостатки метода Дельфи:

1) значительный расход времени на проведение экспертизы, связанный с большим количеством последовательных повторений оценок;

2) необходимость неоднократного пересмотра экспертом своих ответов вызывает у него отрицательную реакцию, что сказывается на результатах экспертизы.

Дальнейшим развитием метода Дельфи являются методы QUWST, SEER, PATTERN.

Методы типа дерева целей.

Идея метода дерева целей впервые была предложена Черчменом в связи с проблемами принята решений в промышленности. Термин «дерево целей» подразумевает использование иерархической структуры, полученной путей разделения общей цели на подцели, а их, в свою очередь, на боле) детальные составляющие — новые подцели, функции и т. д. Как правило, этот термин используется для структур, имеющих от ношение строгого древесного порядка, но метод дерева целей используется иногда и применительно к «слабым» иерархиям в которых одна и та же вершина нижележащего уровня может быть одновременно подчинена двум или нескольким вершина» вышележащего уровня.

Древовидные иерархические структуры используются и при исследовании и совершенствовании организационных структур Не всегда разрабатываемое даже для анализа целей дерево может быть представлено в терминах целей. Иногда, например, при анализе целей научных исследований удобнее говорить о дереве направлений прогнозирования. В. М. Глушковым, например, бы. предложен и в настоящее время широко используется термин) «прогнозный граф». При использовании этого понятия появляется возможность более точно определить понятие дерева как связного ориентированного графа, не содержащего петель, каждая пара вершин которого соединяется единственной цепью.

Морфологические методы. Основная идея морфологических методов — систематически находить все «мыслимые» варианты решения проблемы или реализации системы путем комбинирования выделенных элементов или их признаков. Идеи морфологического образа мышления восходят к Аристотелю, Платону, к известной средневековой модели механизации мышления Р. Луллия. В систематизированном виде морфологический подход был разработан и применен впервые швейцарским астрономом Ф. Цвикки и долгое время был известен как метод Цвикки. Цвикки предложил три метода морфологического исследования.

Первый метод — метод систематического покрытия поля (МСПП), основанный на выделении так называемых опорных пунктов знания в любой исследуемой области и использовании для заполнения поля некоторых сформулированных принципов мышления. Второй — метод отрицания и конструирования (МОК), базирующийся на идее

Цвикки, заключающейся в том, что на пути конструктивного прогресса стоят догмы и компромиссные ограничения, которые есть смысл отрицать, и, следовательно, сформулировав некоторые предложения, полезно заменить их затем на противоположные и использовать при проведении анализа.

Третий — метод морфологического ящика (ММЯ), нашедший наиболее широкое распространение. Идея ММЯ состоит в определении всех «мыслимых» параметров, от которых может зависеть решение проблемы, и представлении их в виде матриц-строк, а затем в определении в этом морфологическом матрице-ящике всех возможных сочетаний параметров по одному из каждой строки. Полученные таким образом варианты могут затем подвергаться оценке и анализу с целью выбора наилучшего. Морфологический ящик может быть не только двумерным. Например, А. Холл использовал для исследования структуры систем трехмерный ящик.

Морфологические ящики Цвикки нашли широкое применение для анализа и разработки прогноза в технике. Для организационных же систем, систем управления такой ящик, который, повидимому, был бы многомерным, практически невозможно построить. Поэтому, используя идею морфологического подхода для моделирования организационных систем, разрабатывают языки моделирования или языки проектирования, которые применяют для порождения возможных ситуаций в системе, возможных вариантов решения и часто — как вспомогательное средство формирования нижних уровней иерархической структуры как при моделировании структуры целей, так и при моделировании организационных структур. Примерами таких языков служат: системно-структурные языки (язык функций и видов структуры, номинально-структурный язык), язык ситуационного управления, языки структурно-лингвистического моделирования.

Методика системного анализа. Методики, реализующие принципы системного анализа в конкретных условиях, направлены на то, чтобы формализовать процесс исследования системы, процесс поставки и решения проблемы. Методика системного анализа разрабатывается и применяется в тех случаях, когда у исследователя нет достаточных сведений о системе, которые позволили бы выбрать адекватный метод формализованного представления системы.

Общим для всех методик системного анализа является формирование вариантов представления системы (процесса решения задачи) и выбор наилучшего варианта. Положив в основу методики системного анализа эти два этапа, их затем можно разделить на подэтапы. Например, первый этап можно разделить следующим образом:

1. Отделение (или ограничение) системы от среды.

2. Выбор подхода к представлению системы.

3. Формирование вариантов (или одного варианта — что часто делают, если система отображена в виде иерархической структуры) представления системы.

Второй этап можно представить следующими под этапами:

1. Выбор подхода к оценке вариантов.

2. Выбор критериев оценки и ограничений.

3. Проведение оценки.

4. Обработка результатов оценки.

5. Анализ полученных результатов и выбор наилучшего варианта (или корректировка варианта, если он был один).

В настоящее время трудно привести примеры методик, в которых все этапы были бы проработаны равноценно.

Задание.

Используя методы мозговой атаки, типа сценариев, экспертных оценок, типа «Дельфи», типа дерева целей выбрать метод решения для следующих задач:

1. Водитель такси обнаружил, что если он находится в городе А, то в среднем в 8 случаях из 10 он везет следующего пассажира в город Б, в остальных случаях будет поездка по городу А. Если же он находится в городе Б, то в среднем в 4 случаях из 10 он везет следующего пассажира в город А, в остальных же случаях будет поездка по городу Б.

   Требуется:

   1. Перечислить возможные состояния процесса и построить граф состояний.
   2. Записать матрицу переходных вероятностей.
   3. Найти вероятности состояний после двух шагов процесса, если:

      А) в начальном состоянии водитель находится в городе А;

      Б) в начальном состоянии водитель находится в городе Б;
   4. Найти вероятности состояний в установившемся режиме.
2. На станцию технического обслуживания(СТО) автомобилей каждые два часа подъезжает в среднем одна машина. Станция имеет 6 постов обслуживания. Очередь автомобилей, ожидающих обслуживания, не ограниченна. Среднее время обслуживания одной машины – 2 часа.

Все потоки в системе простейшие. Определите вероятностные характеристики станции технического обслуживания автомобилей.

3. Звероферма выращивает черно-бурых лисиц и песцов. На звероферме имеется 10 000 клеток. В одной клетке могут быть либо две лисы, либо 1 песец. По плану на ферме должно быть не менее 300 лис и 6000 песцов. В одни сутки необходимо выдавать каждой лисе корма- 4 ед., а каждому песцу – 5 ед. Ферма ежедневно может иметь не более 200 000 единиц корма. От реализации одной шкурки лисы ферма получает прибыль 10 д.е., а от реализации одной шкурки песца – 5 д.е. Какое количество лисиц и песцов нужно держать на ферме, чтобы получить наибольшую прибыль?
4. На складах А,В,С находится сортовое зерно 100, 150, 250 т, которое нужно доставить в четыре пункта. Пункту 1 необходимо поставить 50 т, пункту 2 – 100, пункту 3 – 200, пункту 4 – 150 т сортового зерна. Стоимость доставки 1 т зерна со склада А в указанные пункты соответственно равна (д.е.) 80, 30, 50, 20; со склада В – 40, 10, 60, 70; со склада С – 10, 90, 40, 30. Составьте оптимальный план перевозки зерна из условия минимума стоимости перевозки.
5. Завод имеет три цеха – А, В, С и четыре склада – 1; 2; 3; 4. Цех А производит 30 тыс. шт. изделий, цех В – 40; цех С – 20 тыс. шт. изделий. Пропускная способность складов за то же время характеризуется следующими показателями: склад 1 – 20 тыс. шт. изделий; склад 2 – 30; склад 3 – 30 и склад 4 – 10 тыс. шт. изделий. Стоимость перевозки 1 тыс. шт. изделий из цеха А на склады 1, 2, 3, 4 – соответственно (д.е.): 20, 30, 40, 40, из цеха В – соответственно 40, 30, 20, 60. Составьте такой план перевозки изделий, при котором расходы на перевозку 90 тыс. шт. изделий были бы наименьшими.
6. Из двух сортов бензина образуются две смеси – А и В. Смесь А содержит бензина 60% 1-го сорта и 40% 2-го сорта; смесь В – 80% 1-го

сорта и 20% 2-го сорта. Цена 1 кг смеси A – 10 д.е., а смеси B – 12 д.е. Составьте план образования смесей, при котором будет получен максимальный доход, если в наличии имеется бензина 50 т 1-го сорта и 30 т 2-го сорта.

7. Система массового обслуживания – билетная касса с тремя окошками (с тремя кассирами) и неограниченной очередью. Пассажиров, желающих купить билет, приходит в среднем 5 человек за 20 мин. Поток пассажиров можно считать простейшим. Кассир в среднем обслуживает трех пассажиров за 10 мин. Время обслуживания подчинено показательному закону распределения. Определите вероятностные характеристики СМО в стационарном режиме.
8. Система S состоит из двух узлов – I и II, каждый из которых может в ходе работы системы отказать (выйти из строя). Перечислите возможные состояния системы и постройте граф состояний для двух случаев:

   А) ремонт узлов в процессе работы системы не производится (чистый процесс «гибели» системы);

   Б) отказавший узел немедленно начинает восстанавливаться.

Практическое занятие № 5.
Системный подход и системный анализ

Применения системных представлений для анализа сложных объектов и процессов рассматривают системные направления, включающие в себя: системный подход, системные исследования, системный анализ (системологию, системотехнику и т. п.). За исключением системотехники, область которой ограничена техническими системами, все другие термины часто употребляются как синонимы. Однако в последнее время системные направления начали применять в более точном смысле.

Системный подход. Этот термин начал применяться в первых работах, в которых элементы общей теории систем использовались для практических приложений. Используя этот термин, подчеркивали необходимость исследования объекта с разных сторон, комплексно, в отличие от ранее принятого разделения исследований на физические, химические и др. Оказалось, что с помощью многоаспектных исследований можно получить более правильное представление о реальных объектах, выявить их новые свойства, лучше определить взаимоотношения объекта с внешней средой, другими объектами. Заимствованные при этом понятия теории систем вводились не строго, не исследовался вопрос, каким классом систем лучше отобразить объект, какие свойства и закономерности этого класса следует учитывать при конкретных исследованиях и т. п. Иными словами, термин «системный подход» практически использовался вместо терминов «комплексный подход», «комплексные исследования».

Системные исследования. В работах под этим названием понятия теории систем используются более конструктивно: определяется класс систем, вводится понятие структуры, а иногда и правила ее формирования и т. п. Это был следующий шаг в системных направлениях. В поисках конструктивных рекомендаций появились системные направления с разными названиями: системотехника, системология и др. Для их обобщения стал применяться термин «системные исследования». Часто в работах использовался аппарат исследования операций, который к тому времени был больше развит, чем методы конкретных системных исследований.

Системный анализ. В настоящее время системный анализ является наиболее конструктивным направлением. Этот термин применяется неоднозначно. В одних источниках он определяется как «приложение системных концепций к функциям управления, связанным с планированием» [5]. В других — как синоним термина «анализ систем» (Э. Квейд) или термина «системные исследования» (С. Янг). Однако независимо от того, применяется он только к определению структуры целей системы, к планированию или к исследованию системы в целом, включая и функциональную и обеспечивающую части, работы по системному анализу существенно отличаются от рассмотренных выше тем, что в них всегда предлагается методология проведения исследовании делается попытка выделить этапы исследования и предложить методику выполнения этих этапов в конкретных условиях. В этих работах всегда уделяется особое внимание определению целей системы, вопросам формализации представления целей. Некоторые авторы даже подчеркивают это в определении: системный анализ — это методология исследования целенаправленных систем (Д. Киланд. В. Кинг).

Термин «системный анализ» впервые появился в связи с задачами военного управления в исследованиях RAND Corporation (1948), а в отечественной литературе получил широкое распространение после выхода в 1969 г. книги С. Оптнера «Системный анализ для решения деловых и промышленных проблем».

В начале работы по системному анализу в большинстве случаев базировались на идеях теории оптимизации и исследования операций. При этом особое внимание уделялось стремлению в той или иной форме получить выражение, связывающее цель со средствами, аналогичное критерию функционирования или показателю эффективности, т, е. отобразить объект в виде хорошо организованной системы.

Так, например, в ранних руководящих материалах по разработке автоматизированных систем управления (АСУ) рекомендовалось цели представлять в виде набора задач и составлять матрицы, связывающие задачи с методами и средствами достижения. Правда, при практическом применении этого подхода довольно быстро выяснялась его недостаточность, и исследователи стали прежде всего обращать внимание на необходимость построения моделей, не просто фиксирующих цели, компоненты и связи между ними, а позволяющих накапливать информацию, вводить новые компоненты, выявлять новые связи и т. д,, т. е. отображать объект в виде развивающейся системы, не всегда предлагая, как это делать.

Позднее системный анализ начинают определять как «процесс последовательного разбиения изучаемого процесса на подпроцессы» (С. Янг) и основное внимание уделяют поиску приемов, позволяющих организовать решение сложной проблемы путем расчленения ее на подпроблемы и этапы, для которых становится возможным подобрать методы исследования и исполнителей. В большинстве работ стремились представить многоступенчатое расчленение в виде иерархических структур типа «дерева», но в ряде случаев разрабатывались методики получения вариантов структур, определяемых временными последовательностями функций.

В настоящее время системный анализ развивается применительно к проблемам планирования и управления, и в связи с усилением внимания к программно-целевым принципам в планировании этот термин стал практически неотделим от терминов «целеобразование» и «программно-целевое планирование и управление». В работах этого периода системы анализируются как целое, рассматривается роль процессов целеобразования в развитии целого, роль человека. При этом оказалось, что в системном анализе не хватает средств: развиты в основном средства расчленения на части, но почти нет рекомендаций, как при расчленении не утратить целое. Поэтому наблюдается усиление внимания к роли неформализованных методов при проведении системного анализа. Вопросы сочетания и взаимодействия формальных и неформальных методов при проведении системного анализа не решены. Но развитие этого научного направления идет по пути их решения.

Теория БС с точки зрения системного анализа проблемы включает три основных научных направления:

1. Кибернетику как науку об управлении, включающую анализ информационных процессов в системах с управлением;

2. Исследование операций как науку, дающую количественное обоснование степени соответствия управления целевому назначению системы;

3. Экономические исследования (технико-экономические, военно-экономические исследования), дающие возможность анализировать процесс функционирования основных средств системы.

4. Следовательно, предметом теории систем применительно к большим организационным системам является круг проблем, связанных с анализом целенаправленной деятельности коллективов людей, функционирования техники, которой управляют люди, и техники с силами природы.

Рассмотрим некоторые задачи исследования операций, которые позволяют выполнить количественное обоснование степени соответствия управления целевому назначению системы.

Основная задача линейного программирования. Любую задачу линейного программирования можно свести к стандартной форме, так называемой «основной задаче линейного программирования» (ОЗЛП), которая формулируется так: найти неотрицательные значения переменных x_1, x_2,, … , x_n, которые удовлетворяли бы условиям-равенствам

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m, \end{aligned}\right\} \qquad (60)$$

и обращали бы в максимум линейную функцию этих переменных:

$$L = c_1x_1 + c_2x_2 + \ldots + c_nx_n \to max \tag{61}$$

Убедимся в этом. Во-первых, случай, когда L надо обратить не в максимум, а и минимум, легко сводится к предыдущему, если попросту изменить знак L на обратный (максимизировать не L, а V = -L). Кроме того, от любых условий-неравенств можно перейти к условиям-равенствам ценой введения некоторых новых «дополнительных» переменных. Покажем, как это делается, на конкретном примере.

Пусть требуется найти неотрицательные значения переменных x_1, x_2,, x_3,, удовлетворяющие ограничениям-неравенствам

$$\left.\begin{aligned} 3x_1 + 2x_2 - x_3 \geq 4, \\ x_1 - 2x_2 + 3x_3 \leq 10. \end{aligned}\right\} \tag{62}$$

и обращающие в максимум линейную функцию от этих переменных:

$$L = 4x_1 - x_2 + 2x_3 \to max \tag{63}$$

Начнем с того, что приведем условия (8.3) к стандартной форме, так, чтобы знак неравенства был ≥, а справа стоял нуль. Получим:

$$\left.\begin{aligned} 3x_1 + 2x_2 - x_3 - 4 \geq 0, \\ -x_1 + 2x_2 - 3x_3 + 10 \geq 0. \end{aligned}\right\} \tag{64}$$

А теперь обозначим левые части неравенств (9.5) соответственно через y_1 и y_2:

$$\left.\begin{aligned} y_1 = 3x_1 + 2x_2 - x_3 - 4, \\ y_2 = -x_1 + 2x_2 - 3x_3 + 10. \end{aligned}\right\} \tag{65}$$

Из условий (64) и (65) видно, что новые переменные y_1, y_2 также должны быть неотрицательными.

Задача приобретает следующий вид. Найти неотрицательные значения переменных x_1, x_2,, x_3,, y_1, y_2 такие, чтобы они удовлетворяли условиям-равенствам (65) и обращали в максимум линейную функцию этих переменных (то, что в L не входят дополнительные переменные y_1, y_2,, неважно: можно считать, что они входят, но с нулевыми коэффициентами). Перед нами — основная задача линейного программирования (ОЗЛП). Переход к ней от первоначальной задачи с ограничениями-неравенствами (9.3) «куплен» ценой увеличения числа переменных на два (число неравенств).

Возможен и обратный переход: от ОЗЛП к задаче с ограничениями-неравенствами. Пусть перед нами основная задача линейного программирования с ограничениями-равенствами (60). Предположим, что среди этих m равенств линейно независимыми являются $r \leq m$. В линейной алгебре доказывается, что максимальное число линейно независимых равенств, связывающих n переменных $x_1, x_2,, \ldots , x_n$, равно n, так что вообще $r \leq n$. В линейной алгебре также доказывается, что систему из r независимых равенств с n переменными $x_1, x_2,, \ldots , x_n$ всегда можно разрешить относительно переменных (называемых «базисными») и выразить их через остальные k = n - r переменных (называемых «свободными»). Свободным переменным можно придавать какие угодно значения, не нарушая условий (60). Так вот, для того чтобы перейти от условий-равенств (60) к условиям-неравенствам, достаточно разрешить уравнения (60) относительно каких-то r базисных переменных, выразить их через свободные, а затем вспомнить, что все переменные должны быть неотрицательными, и записать условия их неотрицательности в виде ограничений-неравенств. А потом «забыть» о базисных переменных и манипулировать только свободными, число которых будет $k = n - r$. При этом надо будет освободить от базисных переменных также и функцию L, подставив в нее их выражения через свободные. Таким образом, при переходе от ОЗЛП к задаче с ограничениями-неравенствами число переменных не увеличивается, а уменьшается на число r независимых условий-равенств в ОЗЛП. Примеров такого перехода мы приводить не будем, предоставляя пытливому читателю самому убедиться в его возможности.

Итак, всякая задача линейного программирования может быть сведена к стандартной форме ОЗЛП. Мы не будем подробно останавливаться на способах решения этой задачи. Им посвящены специальные руководства, они описаны во многих книгах по исследованию операций. В следующем параграфе мы изложим только некоторые соображения общего характера относительно существования решения ОЗЛП и способов его нахождения. Никакими расчетными алгоритмами мы заниматься не будем, а отошлем интересующегося читателя к вышеупомянутым руководствам.

Транспортная задача линейного программирования. В предыдущем параграфе мы описали некоторые общие подходы к решению задач линейного программирования. Однако существуют частные типы задач линейного программирования, которые, в силу особой своей структуры, допускают решение более простыми методами. Из них мы остановимся только на одной — так называемой «транспортной задаче» (ТЗ).Она ставится следующим образом: имеются m пунктов от правления (ПО) $A_1, A_2, ..., A_m$, в которых сосредоточены запасы каких-то однородных грузов в количестве соответственно $a_1, a_2, \dots, a_m$ единиц. Имеются n пунктов назначения (ПН) $B_1, B_2, ..., B_n$,подавших заявки соответственно на b_1 , b_2 ,..., b_n единиц груза. Сумма всех заявок равна сумме всех запасов:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \qquad (66)$$

Известны стоимости c_{ij} перевозки единицы груза от каждого пункта отправления A_i до каждого пункта назначения B_i $(i = 1, 2,..., m; j = 1, 2,..., n)$. Все числа c_{ij}, образующие прямоугольную таблицу (матрицу), заданы:

$$\begin{pmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{pmatrix} \qquad (67)$$

Коротко матрицу (10.2) будем обозначать (c_{ij}).

Считается, что стоимость перевозки нескольких единиц груза пропорциональна их числу. Требуется составить такой план перевозок (откуда, куда и сколько единиц везти), чтобы все заявки были выполнены, а общая стоимость всех перевозок минимальна.

Поставим эту задачу как задачу линейного программирования. Обозначим x_{ij} — количество единиц груза, отправляемого из i-го ПО A_i в j-*й* ПН B_i. Неотрицательные переменные x_{ij} тоже можно записать в виде матрицы, которую мы будем коротко обозначать (x_{ij}). Совокупность чисел (x_{ij}) (10.3) мы будем называть «планом перевозок», а сами величины x_{ij} — «перевозками».

$$\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix} \qquad (68)$$

Эти неотрицательные переменные должны удовлетворять следующим условиям.

1. Суммарное количество груза, направляемого из каждого ПО во все ПН, должно быть равно запасу груза в данном пункте. Это даст нам m условий-равенств:

$$
\begin{cases}
x_{11} + x_{12} + \ldots + x_{1n} = a_1, \\
x_{21} + x_{22} + \ldots + x_{2n} = a_2, \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
x_{m1} + x_{m2} + \ldots + x_{mn} = a_m.
\end{cases} \qquad (69)
$$

2. Суммарное количество груза, доставляемого в каждый ПН из всех ПО, должно быть равно заявке, поданной данным пунктом. Это даст нам n условий-равенств.

3. Суммарная стоимость всех перевозок, т. е. сумма величин x_{ij}, ,умноженных на соответствующие стоимости c_{ij},

$$
\begin{cases}
x_{11} + x_{21} + \ldots + x_{m1} = b_1, \\
x_{12} + x_{22} + \ldots + x_{m2} = b_2, \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
x_{1n} + x_{2n} + \ldots + x_{mn} = b_m.
\end{cases} \qquad (70)
$$

должна быть минимальной:

$$
L = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \Rightarrow \min \qquad (71)
$$

где знак двойной суммы означает, что суммирование производится по всем комбинациям индексов i и j, т. е. по всем парам ПО — ПН.

Мы видим, что перед нами – задача линейного программирования с условия–ми-равенствами (69), (70) и минимизируемой линейной функцией (71).

Особенностью этой задачи является то, что все коэффициенты в условиях(69), (70) равны единице — это позволяет решать задачу очень простыми способами. О них и пойдет речь.

Прежде всего, замечаем, что условия-равенства (69), (70) не являются линейно независимыми, так как их правые части связаны условием (66). Число линейно независимых среди уравнений (69), (70) равно не $m + n$

(числу уравнений), а $m + n - 1$. Общее число переменных x_{ij} в нашей задаче равно $m * n$; как бы ни разрешать уравнения (69), (70), число базисных переменных будет равно $m + n-1$, а число свободных переменных

$$k = mn - (m + n - 1) = (m - 1) (n - 1).$$

Мы знаем, что в задаче линейного программирования оптимальное решение достигается в одной из вершин ОДР, в опорной точке, где по крайней мере k переменных равны нулю. Значит, в пашем случае для оптимального плана по крайней мере $(m-1)(n-1)$ перевозок должны быть равны нулю (из соответствующих ПО в соответствующие ПН ничего не перевозится).

Будем называть любой план перевозок допустимым, если он удовлетворяет условиям (10.4), (10.5) (все заявки удовлетворены, все запасы исчерпаны). Допустимый план будем называть опорным, если в нем отличны от нуля не более $m + n - 1$ базисных перевозок, а остальные перевозки равны пулю. План (x_{ij}), будем называть оптимальным, если он, среди всех допустимых планов, приводит к минимальной суммарной стоимости перевозок (L = min).

В силу особой структуры ТЗ при ее решении не приходится долго и нудно разрешать и переразрешать систему уравнений. Все операции по нахождению оптимального плана сводятся к манипуляциям непосредственно с таблицей, где в определенном порядке записаны условия транспортной задачи: перечень ПО и ПН, заявки и запасы, а также стоимости перевозок c_{ij}, По мере заполнения этой таблицы в ее клетках проставляются сами перевозки x_{ij}. Транспортная таблица состоит из m строк и n столбцов. В правом верхнем углу каждой клетки мы будем ставить стоимость c перевозки единицы груза из A_i в B_i , а центр клетки оставим свободным, чтобы помещать в нее саму перевозку x_{ij}. Клетку таблицы, соответствующую пунктам A_i, B_i, будем кратко обозначать (i, j).

Пример транспортной таблицы, где приведены условия задачи и стоимости перевозок, но нет еще самих перевозок, дан в таблице 10.1, где m= 4, n = 5. Прежде всего, займемся составлением опорного плана. Это в транспортной задаче очень просто: можно, например, применить так называемый «метод северо-западного угла». Продемонстрируем его непосредственно на конкретных данных таблицы 1.

Таблица 1.

ПО / ПН	B_1	B_2	B_3	B_4	B_5	Запасы a_i
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 b_j	18	27	42	15	26	128

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 2). Проверим, является ли этот план допустимым: да, потому что в нем сумма перевозок по строке равна запасу соответствующего пункта отправления, а сумма перевозок по столбцу — заявке соответствующего пункта назначения; значит, все в порядке — все заявки удовлетворены, все запасы израсходованы (сумма запасов равна сумме заявок и выражается числом 128, стоящим в правом нижнем углу таблицы).

Здесь и в дальнейшем мы проставляем в таблице только отличные от нуля перевозки, а клетки, соответствующие нулевым перевозкам, оставляем «свободными». Проверим, является ли план перевозок, данный в таблице 2, опорным (не слишком ли много там отличных от нуля, «базисных» перевозок?). Число свободных клеток с нулевыми перевозками в таблице 2 равно как раз $(m\text{-}1)(n\text{-}1)=3*4=12$, так что план — опорный.

Таблица 2.

ПО / ПН	B_1	B_2	B_3	B_4	B_5	Запасы a_i
A_1	13 18	7 12	14	7	5	30
A_2	11	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 b_j	18	27	42	15	26	128

Теперь пора подумать о том, является ли этот план о п т и м а л ь н ы м — т. е. минимальна ли для него общая стоимость перевозок? Скорее всего, нет (ведь составляя опорный план, мы совсем не думали о стоимостях). Так и есть — план не оптимальный. Например, сразу видно, что можно его улучшить, если произвести в нем «циклическую перестановку» перевозок между клетками таблицы, уменьшив перевозки в «дорогой» клетке (2,3) со стоимостью 12, но зато увеличив перевозки в «дешевой» клетке (2.4) со стоимостью 6. Чтобы план оставался опорным, 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). Очевидно, в результате циклического переноса допустимый план остается допустимым — баланс запасов и заявок не нарушается. Ну что же, произведем этот перенос и запишем новый, улучшенный план перевозок в таблице 3.

Теперь посмотрим, чего мы добились, сколько сэкономили? Общая стоимость плана, показанного в таблице 10.2, равна L_1 = 18*13 + 12*7 + 15*8 + 33*12 + 9 *10 + 11*8 + 4*10 + 26*15 = 1442;

Таблица 3.

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

общая стоимость плана, показанного в таблице 10.3, равна L = = 18*13 +12*7 +15*8 + 22*12 + +11*6 +20*10 + 4*10 + 26*15 = 1398.

Таким образом, нам 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 их перенесли целых 11; значит, стоимость должна была уменьшиться на 11*4 = 44 единицы, что и произошло.

Значит, весь секрет оптимизации плана перевозок в том, чтобы переносить («перебрасывать») перевозки по циклам, имеющим отрицательную цену.

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 высматривать — нет ли в таблице соблазнительных, «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. Очевидно, это равно сильно «переразрешению» системы уравнении относительно других базисных переменных, по делается это гораздо легче.

Попробуем еще раз улучшить план, приведенный в таблице 3. Например, нас «соблазняет» дешевая клетка (1.5) со стоимостью 5. Нельзя ли улучшить план, увеличив перевозки в этой клетке, зато уменьшив в других (при этом, конечно, придется кое-где перевозки тоже увеличить). Давайте разглядим внимательно таблицу 3 и найдем в ней цикл, первая вершина которого лежит в свободной клетке (1.5), а остальные — все в базисных клетках. Немного подумав, мы этот цикл обнаружим: 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 4.

Подсчитаем цену этого цикла. Она равна 5 – 15 + 10 – 6 + 8 – 7 = –5. Так как цепа цикла отрицательна, то переброска перевозок по этому циклу выгодна.

Посмотрим, сколько же единиц мы можем по нему перебросить? Это определяется наименьшей перевозкой, стоящей в отрицательной вершине

Таблица 4.

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

цикла. В данном случае это опять 11 (чистое совпадение!). Умножая 11 на цену цикла -5 получим, что за счет переброски 11 единиц груза по данному циклу мы еще уменьшим стоимость перевозок на 55.

Таким образом, разыскивая в транспортной таблице свободные клетки с отрицательной ценой цикла и перебрасывая по этому циклу наибольшее возможное количество груза, мы будем все уменьшать и уменьшать стоимость перевозок. Бесконечно уменьшаться она не может (она никак не может стать меньше нуля!), значит, рано или поздно мы придем к оптимальному плану.

Для такого плана уже не остается ни одной свободной клетки с отрицательной ценой цикла. Это — признак того, что оптимальное решение найдено.

В теории линейного программирования существуют способы, позволяющие автоматически, без размышлений, выделять свободные клетки с отрицательной ценой цикла (так называемый «метод потенциалов»).

Примеры решения задач динамического программирования. В этом параграфе мы рассмотрим (и даже решим до конца) несколько простых (до крайности упрощенных) примеров задач динамического программирования.

Задача прокладки наивыгоднейшего пути между двумя пунктами. Нужно соорудить путь, соединяющий два пункта *A* и *B,* из которых второй лежит к северо-востоку от первого. Для простоты допустим, что про кладка пути состоит из ряда шагов, и на каждом шаге мы можем двигаться либо строго на восток, либо строго на север; любой путь из *A* в *B* представляет собой ступенчатую ломаную линию, отрезки которой параллельны одной из координатных осей (рис. 28). Затраты на сооружение каждого из таких отрезков известны. Требуется проложить такой путь из *A в B,* при котором суммарные затраты минимальны.

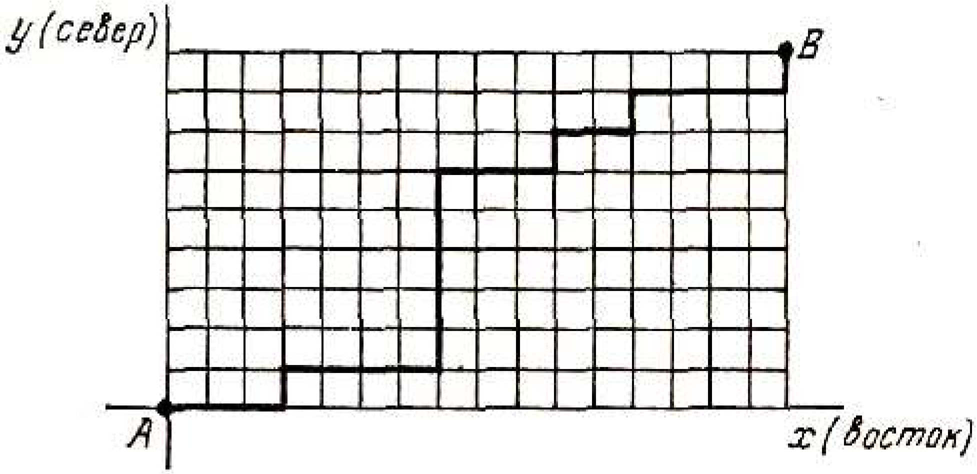


Рис. 28.

Можно поступить одним из двух способов: либо перебрать все возможные варианты пути, и выбрать тот, на котором затраты минимальны (а при большом числе отрезков это очень и очень трудно!); либо разделить процесс перехода из *A* в *B* на отдельные шаги (один шаг – один отрезок) и оптимизировать управление по шагам. Оказывается, второй способ несравненно удобнее! Тут, как и везде в исследовании операций, сказываются преимущества целенаправленного, организованного поиска решения перед наивным «слепым» перебором.

Продемонстрируем, как это делается, на конкретном примере. Разделим расстояние от A до B в во сточном направлении, скажем, на 7 частей, а в север ном — на 5 частей (в принципе дробление может быть сколь угодно мелким). Тогда любой путь из A в B состоит из $m = 7 + 5 = 12$ отрезков, направленных на восток или на север (рис. 29). Проставим на каждом из отрезков число, выражающее (в каких-то условных единицах) стоимость прокладки пути по этому отрезку. Требуется выбрать такой путь из A в B, для которого сумма чисел, стоящих на отрезках, минимальна. Будем рассматривать сооружаемый путь как управляемую систему S, перемещающуюся под влиянием управления из начального состояния A в конечное B. Состояние этой системы перед началом каждого шага будет характеризоваться двумя координатами: восточной *(x)* и северной *(y)*, обе — целочисленные ($0 \leq x \leq 7$, $0 \leq y \leq 5$).

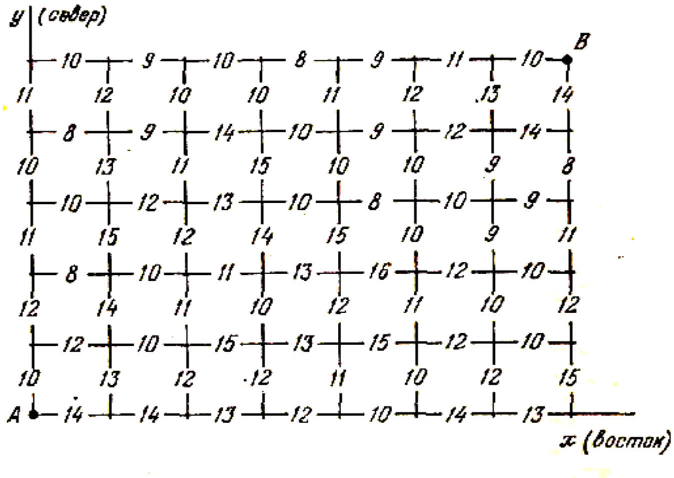
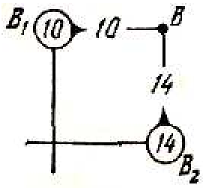


Рис. 29

Для каждого из состояний системы (узловой точки прямоугольной сетки на рис. 29) мы должны найти условное оптимальное управление: идти нам из этой точки на север (управление «с») или на восток (управление «в»). Выбирается это управление так, чтобы стоимость всех оставшихся до конца шагов (включая данный) была минимальна. Эту стоимость мы по-прежнему будем называть «условным оптимальным выигрышем» (хотя в данном случае это не «выигрыш», а «проигрыш») для данного состояния системы 5 перед началом очередного шага.

Процедуру условной оптимизации будем разворачивать в обратном направлении — от конца к началу. Прежде всего, произведем условную оптимизацию последнего, 12-го шага. Рассмотрим отдельно правый верхний угол нашей прямоугольной сетки (рис. 30). Где мы можем находиться после

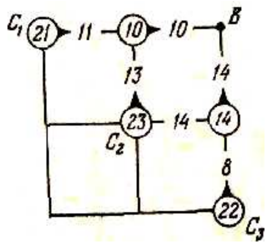


11-го шага? Только там, откуда за один (последний) шаг можно попасть в B, т. е. в одной из точек B_1 или B_2. Если мы находим ся в точке B_1, у нас нет выбора (управление вынужденное): надо идти на восток, и это обойдется нам в 10 единиц.

Рис.30.

Рис. 31.

Запишем это число 10 в кружке у точки B_1, а оптимальное управление покажем короткой стрелкой, исходящей из B_1 и направленной на восток. Для точ ки B_2 управление тоже вынужденное (север), расход до конца равен 14,



мы его запишем в кружке у точки B_2. Таким образом, условная оптимизация последнего шага сделана, и условный оптимальный выигрыш для каждой из точек B_1, B_2 найден и записан в соответствующем кружке.

Теперь давайте оптимизировать предпоследний (11-й) шаг. После предпредпоследнего (10-го) шага мы могли оказаться в одной из точек C_1, C_2 , C_3 (рис. 31). Найдем для каждой из них условное оптимальное управление и условный оптимальный выигрыш. Для точки C_1 управление вынужденное: идти на восток; обойдется это нам до конца в 21 единицу (11 на данном шаге, плюс 10, записанных в кружке при B_1). Число 21 записываем в кружке при точке C_1. Для точки C_2 управление уже не вынужденное: мы можем идти как на восток, так и на север. В первом случае мы затратим на данном шаге 14 единиц и от B_2 до конца — еще 14, всего 28 единиц. Если пойдем на север, затратим 13 + 10, всего 23 единицы. Значит, условное оптимальное управление в точке C_2 — идти на север (отмечаем это стрелкой, а число 23 записываем в кружке у C_2), Для точки C_3 управление снова вынужденное («с»), обойдется это до конца в 22 единицы (ставим стрелку на север, число 22 записываем в кружке при C_3).

Аналогично, «пятясь» от предпоследнего шага назад, найдем для каждой точки с целочисленными координатами условное оптимальное управление («с» или «в»), которое обозначим стрелкой, и условный оптимальный выигрыш (расход до конца пути), который запишем в кружке.

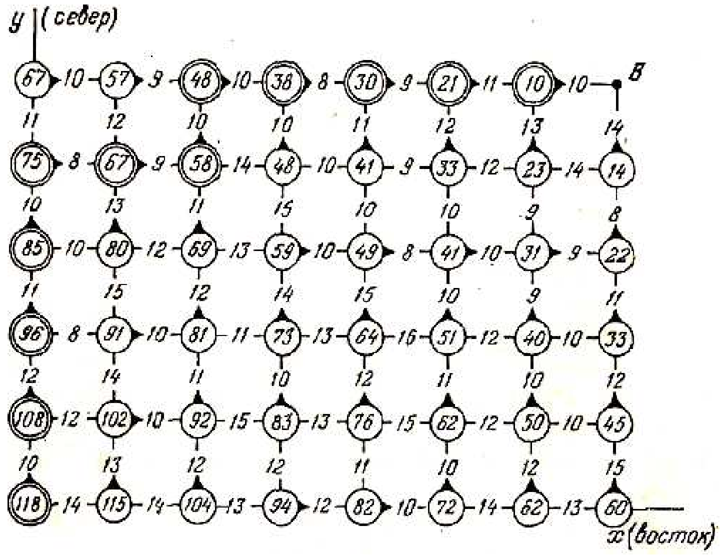


Рис. 32.

Вычисляется он так: расход на данном шаге складывается с уже оптимизированным расходом, записанным в кружке, куда ведет стрелка. Таким образом, на каждом шаге мы оптимизируем только этот шаг, а следующие за ним — уже оптимизированы. Конечный результат процедуры оптимизации показан на рис. 32.

Таким образом, условная оптимизация уже выполнена: в какой бы из узловых точек мы ни находились, мы уже знаем, куда идти (стрелка) и во что нам обойдется путь до конца (число в кружке). В кружке при точке *A* записан оптимальный выигрыш на все сооружение пути из *A* в *B: W** – 118.

Теперь остается построить безусловное оптимальное управление — траекторию, ведущую из *A* и и самым дешевым способом. Для этого нужно только «слушаться стрелок», т. е. прочитать, что они предписывают делать на каждом шаге. Такая оптимальная траектория отмечена на рис. 32 дважды обведенными кружками. Соответствующее безусловное оптимальное управление будет;

$$x^* = (с,с,с,с,в,в,с,в,в,в,в,в),$$

т. е. первые четыре шага мы должны делать на север, следующие два – на восток, затем опять один на север и остальные пять — на восток. Задача решена.

Заметим, что в ходе условной оптимизации мы можем столкнуться со случаем, когда оба управления для какой-то точки на плоскости являются оптимальными, т. е. приводят к одинаковому расходу средств от этой точки до конца, например, в точке с координатами (5; 1) оба управления «с» и «в» являются оптимальными и дают расход до конца равным 62. Из них мы произвольно выбираем любое (в нашем случае мы выбрали «с»; с тем же успехом мы могли бы выбрать «в»). Такие случаи неоднозначного выбора оптимального управления постоянно встречаются в динамическом программировании; в дальнейшем мы специально отмечать их не будем, а попросту выберем произвольно любой из равноценных вариантов. От этого произвола, разумеется, может зависеть оптимальное управление всем процессом, но не оптимальный выигрыш. Вообще, в задачах динамического программирования (как и в задачах линейного) решение далеко не всегда единственное.

А теперь вернемся к началу и попробуем решить задачу «наивным» способом, выбирая на каждом шаге, начиная с первого, самое выгодное (для этого шага) направление (если таких два, выбираем любое). Таким способом мы получим управление

$$x = (с,с,в,в,в,в,с,в,в,в,с,с),$$

Подсчитаем расходы для этой траектории. Они будут равны W=10 +12 + + 8 + 10 + 11 + 13 +15 + 8 + 10 + 9 + 8 +14 = 128, что безусловно больше, чем $W^* = 118$. В данном случае разница не очень велика, но в других она может быть существенной.

В решенной выше задаче условия были намеренно до крайности упрощены. Разумеется, никто не будет вести железнодорожный путь «по ступенькам», перемещаясь только строго на север или строго на восток. Такое упрощение мы сделали для того, чтобы в каждой точке выбирать только из двух управлений: «с» или «в». Можно было бы вместо двух возможных на правлений ввести их несколько и, кроме того, взять шаги помельче; принципиального значения это не имеет, но, разумеется, усложняет и удлиняет расчеты.

Заметим, что задачи, сходные с рассмотренной выше, очень часто встречаются на практике: например, при выборе наискорейшего пути между двумя точками или наиболее экономного (в смысле расхода горючего) набора скорости и высоты летательным аппаратом.

Сделаем одно попутное замечание. , что в нашей задаче точки A и B (начало и конец) в принципе ничем друг от друга не отличаются: можно было бы строить условные оптимальные управления не с конца к началу, а с начала к концу, а безусловные — в обратном направлении. Действительно, это так: в любой задаче динамического программирования «начало» и «конец» можно поменять местами. Это совершенно равносильно описанной ранее методике в расчетном отношении, но несколько менее удобно при словесном объяснении идеи метода: легче аргументировать, ссылаясь на «уже сложившиеся» условия к началу данного шага, чем на те, которые еще «предстоят» после этого шага. По существу же оба подхода совершенно равносильны.

Задача о загрузке машины. Пользуясь методом динамического программирования, можно с успехом решать ряд задач оптимизации, описанных в главе 3, и частности, некоторые задачи целочисленного программирования. Заметим, что целочисленность решений, так затрудняющая задачи линейного программирования, в данном случае не усложняет, а наоборот, упрощает процедуру (как нам ее упростила целочисленность вложений в предыдущей задаче).

В качестве примера рассмотрим задачу о загрузке машины (мы уже упоминали о ней в предыдущей главе): имеется определенный набор предметов $П_1$, $П_2$, ..., $П_n$ (каждый в единственном экземпляре); известны их веса q_1, q_2, ... , q_n и стоимости c_1, c_2, ..., c_n. Грузоподъемность машины равна Q. Спрашивается, какие из предметов нужно взять в машину, чтобы их суммарная стоимость (при суммарном весе $\leq Q$) была максимальна?

Процесс загрузки машины можно представлять себе как состоящий из n шагов; на каждом шаге мы отвечаем на вопрос: брать данный предмет в машину или не брать? Управление на 1-м шаге равно единице, если мы данный (i-й) предмет берем, и нулю — если не берем. Значит, на каждом шаге у нас всего два управления, а это очень приятно.

А чем мы будем характеризовать состояние системы S перед очередным шагом? Очевидно, весом S, который еще остался в нашем распоряжении до конца (полной загрузки машины) после того, как предыдущие шаги выполнены (какие-то предметы погружены в машину). Для каждого из значений S мы должны найти $W_i(S)$ — суммарную максимальную стоимость предметов, которыми можно «догрузить» машину при данном значении S, и положить $x_i(S) = 1$, если мы данный (i-й) предмет берем в машину, и $x_i(S) = 0$, если не берем. Затем эти условные рекомендации должны быть прочтены, и дело с концом!

Решим до конца конкретный числовой пример: имеется шесть предметов, веса и стоимости которых указаны в таблице 8.

Таблица 8.

Предмет $П_i$	$П_1$	$П_2$	$П_3$	$П_4$	$П_5$	$П_6$
Вес q_i	4	7	11	12	16	20
Стоимость c_i	7	10	15	20	27	34

Суммарная грузоподъемность машины Q=35 единиц веса. Требуется указать номера предметов, которые нужно включить в груз, чтобы их суммарная стоимость была максимальна).

Как и ранее, будем придавать S только целые значения. Условная оптимизация решения показана в таблице 9, где в каждой строке для соответствующего номера шага (номера предмета) приведены: условное оптимальное управление x_i (0 или 1) и условный оптимальный выигрыш W_1 (стоимость всех оставшихся до конца предметов при оптимальном управлении на всех шагах). Как эта таблица составляется, мы уже объяснять не будем — тут полная аналогия с предыдущей задачей, с той разницей, что возможные управления только 0 или 1. В таблице 9 выделены:m оптимальный выигрыш W*=57 и оптимальные шаговые управления, при которых этот выигрыш достигается: x1=0, x2=2, x3=0, x4=1, x5=1, x6=0, , т.е. загрузить машину надо предметами 2, 4 и 5, суммарный вес которых равен в точности 35 (вообще это необязательно; при оптимальном выборе грузов может быть и некоторый общий «недогруз»).

Заметим, что в нашем элементарном примере, возможно, было бы проще искать решение «простым перебором»), пробуя все возможные комбинации предметов, проверяя на каждой из них, «влезают» ли они в заданный вес, и выбирая ту, для которой стоимость максимальна. Но при большом числе предметов это было бы затруднительно: число комбинаций неумеренно растет при увеличении числа предметов. Для метода же динамического программирования увеличение числа шагов не страшно, оно только приводит к пропорциональному возрастанию объема расчетов.

Таблица 9.

S	i=6		i=5		i=4		i=3		i=2		i=1	
	X_i	W_i	X_i	W_i	X_i	W_i	X_i	W_i	X_i	W_i	X_i	W_i
0	0	0	0	0	0	0	0	0	0	0		
1	0	0	0	0	0	0	0	0	0	0		
2	0	0	0	0	0	0	0	0	0	0		
3	0	0	0	0	0	0	0	0	0	0		
4	0	0	0	0	0	0	0	0	0	0		
5	0	0	0	0	0	0	0	0	0	0		
6	0	0	0	0	0	0	0	0	0	0		
7	0	0	0	0	0	0	0	0	1	10		
8	0	0	0	0	0	0	0	0	1	10		
9	0	0	0	0	0	0	0	0	1	10		
10	0	0	0	0	0	0	0	0	1	10		
11	0	0	0	0	0	0	1	15	0	15		
12	0	0	0	0	1	20	0	20	0	20		
13	0	0	0	0	1	20	0	20	0	20		
14	0	0	0	0	1	20	0	20	0	20		
15	0	0	0	0	1	20	0	20	0	20		
16	0	0	1	27	0	27	0	27	0	27		
17	0	0	1	27	0	27	0	27	0	27		
18	0	0	1	27	0	27	0	27	0	27		
19	0	0	1	27	0	27	0	27	1	30		
20	1	34	0	34	0	34	0	34	0	34		
21	1	34	0	34	0	34	0	34	0	34		
22	1	34	0	34	0	34	0	34	0	34		
23	1	34	0	34	0	34	1	35	1	37		
24	1	34	0	34	0	34	1	35	1	37		
25	1	34	0	34	0	34	1	35	1	37		
26	1	34	0	34	0	34	1	35	1	37		
27	1	34	0	34	0	34	1	42	1	44		
28	1	34	0	34	1	47	0	47	0	47		
29	1	34	0	34	1	47	0	47	0	47		
30	1	34	0	34	1	47	0	47	0	47		
31	1	34	0	34	1	47	1	49	0	49		
32	1	34	0	34	1	54	0	54	0	54		
33	1	34	0	34	1	54	0	54	0	54		
34	1	34	0	34	1	54	0	54	0	54		
35	1	34	0	34	1	54	0	54	1	57	0	57

Рассмотренные выше простейшие задачи динамического программирования дают понятие об общей идее метода: пошаговая оптимизация, проходимая в одном направлении «условно», в другом — «безусловно». Метод динамического программирования является очень мощным и плодотворным методом оптимизации управления; ему не страшны ни целочисленность решения, ни нелинейность целевой функции, ни вид ограничений, накладываемых на решение. Но в отличие от линейного программирования динамическое программирование не сводится к какой-либо стандартной вычислительной процедуре; оно может быть передано на машину только после того, как записаны соответствующие формулы, а это часто бывает не так-то легко.

Для постановки задачи динамического программирования необходимо решить следующие вопросы. Первый вопрос, на который нужно ответить ставящему задачу: какими параметрами характеризуется состояние управляемой системы S перед каждым шагом? От удачного выбора набора этих параметров часто зависит возможность успешно решить задачу оптимизации. В примерах, которые мы решали, состояние системы характеризовалось очень небольшим числом параметров: двумя координатами — в первом примере, одним числом — во втором. Но такие ультрапростые задачи не так уже часто встречаются на практике. Если, как это обычно и бывает, состояние системы описывается многими параметрами (так называемыми «фазовыми координатами»), то становится трудно перед каждым шагом перебрать все их варианты и для каждого найти оптимальное условное управление. Последнее еще больше затрудняется в случае, когда число возможных вариантов управления велико. В этих случаях над нами повисает, по меткому выражению Р. Беллмана, «проклятие многомерности» — бич не только метода динамического программирования, но и всех других методов оптимизации. Обычно задачи динамического программирования решаются не вручную, а на ЭВМ, однако многие такие задачи не под силу даже современным машинам. Поэтому очень важно уметь правильно и «скромно» поставить задачу, не переобременяя ее лишними подробностями, упрощая елико возможно описание управляемой системы и вариантов управления. Так что в методе динамического программирования очень многое зависит от искусства и опыта исследователя.

Вторая задача после описания системы и перечня управлений — это членение на шаги (этапы). Иногда оно бывает задано в самой постановке задачи(например, хозяйственные годы в экономических задачах), но часто членение на шаги приходится вводить искусственно. Если бы мы в этой задаче не ограничились самым примитивным случаем всего двух управлении («с» и «в»), то было бы удобнее членение на шага произвести иначе, например, считая за «шаг» переход с одной прямой, параллельной оси ординат, на другую. Можно было бы вместо прямых рассмотреть окружности с центром в точке *A* или же другие кривые. Все такие способы выбора наивыгоднейшего пути неизбежно ограничивают выбор возможных направлений. Если за «шаги» считать переходы с одной прямой, параллельной оси ординат, на другую, то здесь множество возможных управлений не предусматривает «пути назад», т. е. с более восточной прямой на более западную. В большинстве задач практики такие ограничения естественны (например, трудно себе представить, чтобы наивыгоднейшая траектория космической ракеты, пущенной с Земли, на каких-то участках включала движение «назад», ближе к Земле). Но бывает и другая обстановка. Например, путь по сильно пересеченной местности («серпантинная» дорога в горах) часто «петляет» и возвращается ближе к исходному пункту. При постановке задачи динамического программирования, в частности при выборе системы координат и способа членения на шаги, должны быть учтены все разумные ограничения, накладываемые на управление.

Второй вопрос – выбор числа шагов m. С первого взгляда может показаться, что чем больше m, тем лучше. Это не совсем так. При увеличении m возрастает объем расчетов, а это не всегда оправдано. Число шагов нужно выбирать с учетом двух обстоятельств: 1) шаг должен быть достаточно мелким для того, чтобы процедура оптимизации шагового управления была достаточно проста, и 2) шаг должен быть не слишком мелким, чтобы не производить ненужных расчетов, только усложняющих процедуру поиска оптимального решения, но не приводящих к существенному изменению оптимума целевой функции. В любом случае практики нас интересует не строго оптимальное, а «приемлемое» решение, не слишком отличающееся от оптимального по значению выигрыша W^*

Сформулируем общий принцип, лежащий в основе решения всех задач динамического программирования (его часто называют «принципом оптимальности»). Каково бы ни было состояние системы S перед очередным шагом, надо выбирать управление на этом шаге так, чтобы выигрыш на данном шаге плюс оптимальный выигрыш на всех последующих шагах был максимальным.

По-видимому, полное понимание этого принципа делается возможным (для лиц с обычным математическим развитием) только после рассмотрения ряда примеров, поэтому мы и приводим этот основной принцип не в начале главы (как это было бы естественно для математика), а лишь после решения ряда примеров.

А теперь сформулируем несколько практических рекомендаций, полезных начинающему при постановке задач динамического программирования. Эту постановку удобно проводить в следующем порядке.

1. Выбрать параметры (фазовые координаты), характеризующие состояние S управляемой системы перед каждым шагом.
2. Расчленить операцию на этапы (шаги).
3. Выяснить набор шаговых управлений x_i, для каждого шага и налагаемые на них ограничения.

4. Определить, какой выигрыш приносит на i –м шаге управление x_i, если перед этим система была в состоянии S, т.е. записать «функции выигрыша»:

$$\omega_i = f_i(S, x_i) \tag{72}$$

5. Определить, как изменяется состояние S системы S под влиянием управления x_i на i-м шаге: оно переходит в новое состояние

$$S' = \varphi_i(S, x_i) \tag{73}$$

«Функции изменения состояния» (73) тоже должны быть записаны).

6.Записать основное рекуррентное уравнение динамического программирования, выражающее условный оптимальный выигрыш $W_i(S)$ (начиная с i-го шага и до конца) через уже известную функцию $W_{i+1}(S)$

$$W_i(S) = max\{f_i(S, x_i) + W_{i+1}(S)(\varphi_i(S, x_i))\} \tag{74}$$

Этому выигрышу соответствует условное оптимальное управление на *i-м* шаге $x_i(S)$ (подчеркнем, что в уже известную функцию $W_{i+1}(S)$ надо вместо S подставить измененное состояние $S' = \varphi_i(S, x_i))$.

7. Произвести условную оптимизацию последнего (m-го) шага, задаваясь гаммой состояний S, из которых можно за один шаг дойти до конечного состояния, вычисляя для каждого из них условный оптимальный выигрыш по формуле

$$W_m(S) = max\{f_m(S, x_m)\} \tag{75}$$

и находя условное оптимальное управление $x_m(8)$, для которого этот максимум достигается.

8. Произвести условную оптимизацию $(m$ — 1)-го, $(m$ — 2)-го и т.д. шагов по формуле (75), полагая в ней $i=(m$— 1), $(m$ — 2), ..., и для каждого из шагов указать условное оптимальное управление $x_i(S)$, при котором максимум достигается. Заметим, что если состояние системы в начальный момент известно (а это обычно бывает так), то на первом шаге варьировать состояние системы не нужно — прямо находим оптимальный выигрыш для

данного начального состояния S_0. Это и есть оптимальный выигрыш за всю операцию

$$W^{*}=W_{1}(S_{0}).$$

9.Произвести безусловную оптимизацию управления, «читая» соответствующие рекомендации на каждом шаге. Взять найденное оптимальное управление на первом шаге $x_1^{*}=x_1(S_0)$; изменить состояние системы по формуле (73); для вновь найденного состояния найти оптимальное управление на втором шаге $x_г$ и т. д, до конца.

Сделаем несколько дополнительных замечаний общего характера. До сих пор мы рассматривали только аддитивные задачи динамического программирования, в которых выигрыш за всю операцию равен сумме выигрышей на отдельных шагах. Но метод динамического программирования применим также и к задачам с так называемым «мультипликативным» критерием, имеющим вид произведения:

$$W=\prod_{i=1}^{m} w_i \tag{76}$$

(если только выигрыши ω_i положительны). Эти задачи решаются точно так же, как задачи с аддитивным критерием, с той единственной разницей, что в основном уравнении (75) вместо знака «плюс» ставится знак умножения:

$$W_i(S) = \max \{f_i(S, x_i)*W_{i+1}(\varphi_i(S, x_i))\}. \tag{77}$$

Задание

1. Построить математическую модель ЗЛП. Из трех видов продуктов – 1,2, 3 составляется смесь. В состав смеси должно входить не менее 6ед. химического вещества А, 8 ед. вещества В и не менее 12 ед. вещества С. Структура химических веществ приведена в таблице:

продукт	Содержание хим. веществ в 1 ед.продукции			Стоимость 1ед.продукции
	А	В	С	
1	2	1	3	2
2	1	2	4	3
3	3	1,5	2	2,5

Составить наиболее дешевую смесь.

2. Построить математическую модель ЗЛП. Имеются две почвенно-климатические зоны, площади которых соответственно равны 0,8 и 0,6 млн.га. Данные об урожайности приведены в таблице:

Зерн. культуры	Урожайность (ц/га)		Стоимость 1-ц Д.е.
	1-ая зона	2-ая зона	
Озимые	20	25	8
яровые	25	20	7

Определить размеры посевных площадей озимых и яровых культур, необходимые для достижения максимального выхода продукции в стоимостном выражении.

4. Решить ЗЛП графическим методом.

$$W=2x1+3x2 \longrightarrow max$$

$$\begin{cases} x1+x2<=8 \\ \end{cases}$$

$x1 >= 4$

$x2 >= 3$

5. Решить ЗЛП графическим методом.

$W=2x1-2x2 \rightarrow min$

$$\begin{cases} x1+x2>=1 \\ -x1+x2<=1 \end{cases}$$

6. Решить ЗЛП графическим методом.

$W=x1+4x2 \rightarrow min$

$$\begin{cases} 3x1+5x2>=8 \\ -3x1+10x2<=16 \end{cases}$$

7. Решить ЗЛП графическим методом.

$W=2x1-5x2 \rightarrow min$

$$\begin{cases} 3x1+4x2<=6 \\ 2x1+3x2<=4 \end{cases}$$

8. Имеются два хранилища с однородными продуктом, в котором сосредоточены 200 и 120 т продукта соответственно. Продукты необходимо перевезти трем потребителям соответственно в количестве 80, 100 и 120 т. Расстояние от хранилищ до потребителей следующие

Хранилища	Потребители		
	1	2	3
1	20	30	50
2	60	20	40

Затраты на перевозку одной тонны продукта на 1 км постоянны и равны 5 д.е. Определить план перевозок продукта от хранилищ до потребителя из условия минимизации транспортных расходов.

9. Промышленный концерн имеет два завода и пять складов в различных регионах страны. Каждый месяц первый завод производит 40, а второй 70 ед. продукции. Вся продукция производимая заводами, должна быть направлена на склады. Вместимость первого склада – 20 ед. продукции, второго – 30, третьего – 15, четвертого– 27, пятого – 28 ед. издержки транспортировки продукции от завода до склада следующие.

Заводы	склады				
	1	2	3	4	5
1	520	480	650	500	720
2	450	525	630	560	750

Распределить план перевозок из условия минимизации ежемесячных расходов на транспортировку.

10. Завод имеет три цеха – А, В, С и четыре склада 1, 2, 3, 4. Цех А производит 30 тыс. шт. изделий, В – 40 тыс. шт.изделий, С– 20 тыс. шт.изделий. Пропускная способность за то же время характеризуется следующими показателями : склад 1- 20 тыс. изделий, склад 2 – 30, склад 3 – 30, склад 4 – 10. Стоимость перевозки 1 тыс изделий из цеха А на склады 1, 2, 3, 4 соответственно равны 20, 30, 40, 40 (д.е.), из цеха В 30, 20, 50, 10, из цеха С 40, 30,20, 60. Составьте такой план перевозки изделий, при котором расходы на перевозку 90 тыс. изделий минимальны.

11. На складах А, В, С находится сортовое зерно 100, 150 и 250 тонн, которое нужно доставить в четыре пункта. Пункту 1 необходимо поставить 50 т зерна, пункту 2 – 100, пункту 3 –200, пункту 4 – 150т сортового зерна.

Стоимость доставки 1 т зерна со склада А в указанные пункты соответственно равна (д.е.) 80, 30, 50, 20, со склада В – 40, 10, 60, 70; со склада С – 10, 90, 40, 30. Составить оптимальный план перевозки зерна из условия мимнимума стоимости перевозки.

Контрольные вопросы

1. Какие задачи относятся к задачам математического программирования?
2. Как характеризуются задачи линейного программирования?
3. Какие трудности возникают при решении задач математического программирования?
4. Какой вид имеют функция цели и ограничения для задач линейного программирования?
5. Что такое допустимое решение основной задачи линейного программирования?

Литература

1. Бусленко Н.П. Моделирование сложных систем. - М.: Наука, 1978.- 399 с.

2. Бусленко Н.П. Лекции по теории сложных систем. - М.: Сов. радио, 1973.- 439 с.

3. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. - М.: Финансы и статистика, 1998.- 176 с.

4. Дегтярев Ю.И. Системный анализ и исследование операций. - М.: Высшая школа, 1996.- 336 с.

5. Иванов П.М. Алгебраическое моделирование сложных систем. - М.: Наука, 1996.- 272 с.

6. Калашников В.В. Качественный анализ поведения сложных систем методом пробных функций. - М.: Наука, 1978.- 247 с.
7. Клейнрок Л. Вычислительные системы с очередями. - М.: Мир, 1979.- 600 с.
8. Максименко А.В., Селезнев М.Л. Основы проектирования информационно-вычислительных систем и сетей. - М.: Радио и связь, 1991.- 320 с.
9. Моисеев Н.Н. Математические задачи системного анализа. - М.: Наука, 1981.-487 с.
10. Николаев В.И., Брук В.М. Системотехника: методы и приложения. - Л.: Машиностроение, 1985.- 199 с.
11. Теория информационных процессов и систем. Учебник/ под ред. Б.Я. Советова – М. Издательский центр «Академия», 2010. – 432с.

Наталья Викторовна Назаренко,

старший преподаватель кафедры ИУС АмГУ

Любовь Александровна Соловцова,

старший преподаватель кафедры ИУС АмГУ

Практикум по теории информационных процессов и систем.

Формат 60х84/16. Усл. печ. л. 6,97. Заказ 269.