




Автоматизация управления жизненным циклом продукции

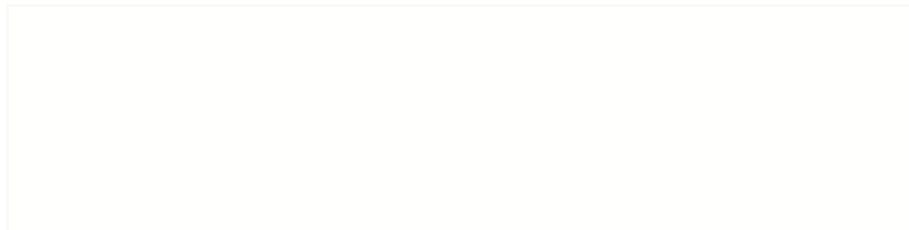
Конспект лекций

для студентов, обучающихся по направлению
15.03.04 — «Автоматизация технологических процессов
и производств» (бакалавры)





Рассмотрены вопросы управления издательской деятельностью на всех этапах производства печатной продукции. Определены основные задачи и функции управления производством с применением автоматизированных систем управления предприятием. Сформулированы основные принципы построения и функционирования систем управления. На примере реально существующих и используемых систем рассмотрены процессы оформления заказов на изготовление полиграфической продукции, прохождения их в производстве, нормирования технологических операций, учета и контроля производства.



Содержание

Введение	4
ТЕМА 1. Методы оптимизации в задачах управления	10
Лекция 1. Задачи линейного программирования.....	10
Лекция 2. Симплекс-метод решения задачи линейного программирования.....	14
Лекция 3. Транспортная задача. Метод потенциалов.....	20
Лекция 4. Динамическое программирование.....	28
Лекция 5. Принцип оптимальности Беллмана.....	31
Лекция 6. Многокритериальные задачи.....	39
Лекция 7. Нелинейное программирование	42
Лекция 8. Численные методы нахождения экстремума	48
Лекция 9. Недетерминированные системы	52
ТЕМА 2. Автоматизированные информационно-управляющие системы	60
Лекция 10. Общая характеристика АИУС	60
Лекция 11. Системный подход к проектированию АИУС	66
Лекция 12. Моделирование рабочего потока.....	76
Лекция 13. Методики моделирования IDEF	80
Лекция 14. Методология функционального моделирования	83
Лекция 15. Многоуровневая система управления производством	87
Лекция 16. Современные системы управления производством	93
Библиографический список.....	99

ВВЕДЕНИЕ

Наличие жесткой конкуренции на мировом рынке предполагает внедрение оптимальных методов создания продукции, использование новых технологий, включение в научно-техническую и экономическую сферы всего интеллектуального потенциала персонала предприятия.

Результатами деятельности любого предприятия являются продукция и услуги, производимые за счет реализации бизнес-процессов.

Термины «продукция» и «бизнес-процесс» необходимо рассматривать в наиболее широком смысле. Результат деятельности в виде оказания услуг технологического или информационного характера также можно включить в понятие «продукция».

Согласно определению международной ассоциации WMC (Коалиция управления потоками работ) под бизнес-процессом в рамках определенной организационной структуры понимается совокупность процедур и отдельных операций, позволяющих реализовать некоторую бизнес-задачу и обеспечить достижение коммерческой прибыли.

К важнейшим бизнес-процессам предприятия следует отнести производственные и, в частности, технологические процессы. Их целевыми функциями являются экономические показатели: минимизация себестоимости, получение прибыли, возврат инвестиций и т.п. К вспомогательным целям можно отнести укрепление позиций в заданных секторах рынка, выход на новые рынки, повышение имиджа фирмы и др.

Ускорение процессов создания изделий, необходимость повышения конкурентоспособности предприятия предполагает отслеживание всего процесса жизненного цикла продукции. Это понятие охватывает все стадии информационного представления продукции: от выявления потребностей и анализа научно-технических возможностей выпуска продукции наивысшего на

данный момент качества до снятия ее с производства, вывода из эксплуатации и утилизации после окончания срока службы.

Жизненный цикл (ЖЦ) продукции — это совокупность процессов, выполняемых в течение реального времени от момента выявления потребностей общества в определенной продукции до момента удовлетворения этих потребностей и утилизации продукции. ЖЦ представляет собой совокупность стадий и этапов, через которые проходит изделие в течение времени своего существования: научные исследования, разработка технического задания, проектирование и конструирование, технологическая подготовка производства (ТПП), изготовление, испытания и приемка, поставка, эксплуатация, утилизация и др.

Реализация любого достаточно сложного изделия основана на необходимости соблюдения единства требований и концепций, выражение которых предполагает использование информации значительной емкости.

В современном понимании обеспечивается информационная поддержка ЖЦ продукции в рамках создания единого информационного пространства (ЕИП), что создает условия для свободного информационного обмена сведениями о продукции. ЕИП должно создавать условия для выработки, распределения, передачи, обработки, хранения и использования информации по всем необходимым направлениям ее функционального назначения.

Внедрение современных информационных и компьютерных технологий для поддержки ЖЦ продукции позволяет за счет компьютерного моделирования процессов создать конечную продукцию за минимальное количество стадий. Для этого в современных издательствах на разных этапах используются высокоэффективные компьютерные системы: графические станции, растровые процессоры, цветовые корректоры и т.д. Широко распространены локальные и глобальные компьютерные сети.

В последнее время развитие бизнеса выравнивало технические возможности типографий и их технологические преимущества. Создание конкурентного преимущества за счет технического оснащения становится менее возможным. В структуре затрат типографии увеличивается доля заработной платы и материалов. По-

вышается стоимость услуг квалифицированных сотрудников. Внимание руководителей стало переключаться на другие направления сохранения конкурентного преимущества типографий, такие как перестройка и оптимизация бизнес-процессов на основе внедрения автоматизированных систем управления административно-хозяйственной, финансовой и производственной деятельностью предприятия.

Уровень автоматизации управления предприятия и использование для этого различных компьютерных систем зависит от общего уровня развития предприятия, включая его техническое, информационное и кадровое обеспечение, а также от существующей в конкретном регионе конкуренции.

Можно выделить 5 типов систем, каждый из которых соответствует определенному уровню организации автоматизированного управления.

Первый тип систем управления характерен тем, что все управление и учет ведутся на так называемых бумажных носителях. Расчет выполняется в рабочей тетради менеджера продаж. Для оформления технологической карты используются самокопирующиеся бланки. Планирование ведется на специализированной доске загрузки оборудования, где расположены ярлыки технологических операций. Бухгалтерские и отгрузочные документы оформляются на стандартных бланках вручную. Регистрация факта выполнения операций ведется в бланках рапортчек. Получение отчетной и аналитической информации выполняется специальной службой в учетных таблицах. Такой метод управления встречается сейчас крайне редко и характерен либо для предприятий старой формации, либо для мини-предприятий с невысокой рентабельностью.

Второй тип систем определяется сочетанием «бумажных» и компьютерных технологий. Это наиболее распространенный тип систем управления. Более 45% предприятий используют именно такие системы управления производством. Планирование ведется на планировочной доске или в среде электронных таблиц. Для обмена документами применяют системы электронной почты или документооборота, такие как Lotus Notes или MS Exchange. Для подготовки отчетной и аналитической информации используются

персональные системы управления базами данных типа MS Access.

Третий тип систем управления аналогичен предыдущему, но полностью исключаются «бумажные технологии». Не более 17% предприятий применяют полностью электронные системы управления. Стоит заметить, что подобные системы достаточно часто используют самостоятельно разработанные компоненты. Для таких решений характерны те же проблемы, что и для систем второго типа.

Четвертый тип — системы, разработанные по специальному заказу типографии или печатного салона либо фирмой, специализирующейся на разработке подобных систем, либо командой программистов, входящей в штат предприятия. Таким системам свойственна низкая гибкость и привязка к технологическим и экономическим методам работы конкретного предприятия. Системы подобного типа становятся «заложниками» команды разработчиков. Количество систем невелико, чуть более 7%. Для них характерны проблемы с надежностью, целостностью и достоверностью данных, так как эти системы являются штучным продуктом и не всегда могут быть обеспечены достаточным уровнем отладки и сопровождения.

Пятый тип — специализированные системы, созданные непосредственно для управления полиграфическим производством. Доля подобных систем среди всех используемых составляет более 20%, и эта доля стремительно увеличивается. Для европейских и североамериканских типографий использование специально разработанных для целей управления систем намного более характерно, чем для российских.

Изучение дисциплины «Автоматизация управления жизненным циклом продукции» позволит обучающимся обладать профессиональными **компетенциями**:

- способностью собирать и анализировать исходные информационные данные для проектирования технологических процессов изготовления продукции, средств и систем автоматизации, контроля, технологического оснащения, диагностики, испыта-

ний, управления процессами, жизненным циклом продукции и ее качеством (ПК-1);

- способностью использовать прикладные программные средства при решении практических задач профессиональной деятельности, методы стандартных испытаний по определению физико-механических свойств и технологических показателей материалов и готовых изделий, стандартные методы их проектирования, прогрессивные методы эксплуатации изделий (ПК-4);
- готовностью применять способы рационального использования сырьевых, энергетических и других видов ресурсов, современные методы разработки малоотходных, энергосберегающих и экологически чистых технологий (ПК-5);
- способностью выполнять работы по расчету и проектированию средств и систем автоматизации, контроля, диагностики, испытаний, управления процессами, жизненным циклом продукции и ее качеством в соответствии с техническими заданиями и использованием стандартных средств автоматизации расчетов и проектирования (ПК-18);
- способностью к практическому освоению и совершенствованию систем автоматизации производственных и технологических процессов, контроля, диагностики, испытаний, управления процессами, жизненным циклом продукции и ее качеством (ПК-20);
- способностью выполнять работы по автоматизации технологических процессов и производств их обеспечению средствами автоматизации и управления, использовать современные методы и средства автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством (ПК-21);
- способностью определять номенклатуру параметров продукции и технологических процессов ее изготовления, подлежащих контролю и измерению, устанавливать оптимальные нормы точности продукции, измерений и достоверности контроля, выбирать технические средства автоматизации, контроля, ди-

агностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством (ПК-22).

В результате изучения дисциплины студент должен

знать:

- типовые алгоритмы решения вычислительных задач;
- правила составления графических блок-схем алгоритмов;
- синтаксис языка высокого уровня Visual Basic;

уметь:

- разработать алгоритм решения математической задачи;
- написать и отладить программу на языке высокого уровня;

владеть:

- специальной терминологией в области программирования;
- принципами и методами отладки программ.

(причем, как со знаком « \leq », так и со знаком « \geq »), а другая часть — в виде уравнений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1, \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m, \\ a_{m+1,1}x_1 + a_{m+1,2}x_2 + \dots + a_{m+1,n}x_n = b_{m+1}, \\ \dots \\ a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n = b_k, \\ x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0. \end{cases}$$

При этом может отыскиваться как максимум, так и минимум целевой функции:

$$c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \max \text{ (или } \min \text{)}.$$

Очевидно, что стандартная задача линейного программирования является частным случаем общей.

Путем добавления новых переменных, по одной в каждое из неравенств, общую задачу линейного программирования можно привести к каноническому виду, когда все ограничения имеют вид равенств. При этом необходимо соблюдать следующее правило: если в неравенстве стоит знак « \leq », то добавляется новая переменная со знаком «+», в противном случае — со знаком «-». В целевой функции также добавляются эти новые переменные, но с коэффициентом 0. Можно доказать, что решения канонической задачи являются решениями соответствующей ей общей задачи, и наоборот.

В качестве примера рассмотрим задачу

$$\begin{cases} -2x_1 + 3x_2 \leq 14, \\ x_1 + x_2 \leq 8, \\ x_1 \geq 0, x_2 \geq 0, \\ f(x_1, x_2) = 2x_1 + 7x_2 \rightarrow \max. \end{cases}$$

Для того чтобы привести эту задачу к канонической форме, в первые два неравенства добавим по одной новой переменной и заменим знаки неравенства на знаки равенства:

$$\begin{cases} -2x_1 + 3x_2 + x_3 = 14, \\ x_1 + x_2 + x_4 = 8, \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, \\ f(x_1, x_2, x_3, x_4) = 2x_1 + 7x_2 + 0x_3 + 0x_4 \rightarrow \max. \end{cases}$$

К канонической форме переходят потому, что именно для этого типа задач линейного программирования разработаны основные вычислительные методы (симплекс-метод и его модификации).

Любой набор значений переменных величин x_1, x_2, \dots, x_n , удовлетворяющих системе ограничений задачи линейного программирования, называется допустимым решением или допустимым планом. Тот набор, для которого целевая функция принимает максимальное (или минимальное, в зависимости от поставленной задачи) значение, называется оптимальным решением или оптимальным планом.

Задача с двумя неизвестными допускает наглядную геометрическую интерпретацию. Ее решение можно найти с помощью простых геометрических построений. Рассмотрим, например, задачу, условие которой приведено в начале этой лекции. Точки, координаты которых удовлетворяют заданным неравенствам, образуют на плоскости выпуклый многоугольник (область допустимых решений).

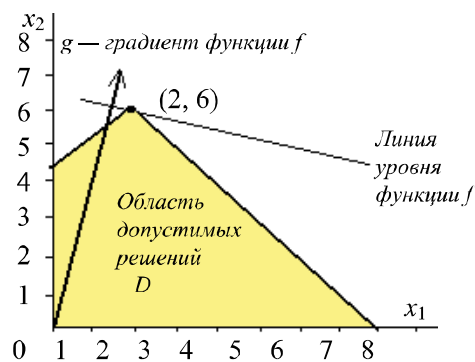


Рис. 1.1. Геометрический метод решения задачи

На рис. 1.1 показана область допустимых решений D , а также вектор g , который показывает направление наибольшего возрастания целевой функции $f(x_1, x_2)$ из начала координат. Этот вектор называется градиентом, его координатами являются частные производные от f по переменным x_1 и x_2 соответственно: $g = \text{grad } f(x_1, x_2) = (\partial f / \partial x_1, \partial f / \partial x_2)$. В данном случае этот вектор имеет координаты 2 и 7.

Линия, перпендикулярная градиенту, является линией уровня целевой функции, т.е. линией, в каждой точке которой функция принимает одно и то же значение. Если перемещать эту линию в направлении вектора g , то по мере удаления от начала координат значения f , соответствующие точкам линии уровня, возрастают. Линия уровня, которая касается самой удаленной от начала координат точки области D , и определяет искомое решение — оптимальный план: $x_1 = 2$, $x_2 = 6$. Целевая функция достигает при этом максимального значения 46.

Из рис. 1.1 видно, что оптимальное решение достигается в угловой точке области D . Можно доказать, что и в общем случае при большем числе переменных, когда решение задачи существует, оно достигается в одной из угловых точек области допустимых решений. В этом и состоит первая геометрическая интерпретация задачи линейного программирования.

Для того чтобы доказать приведенное выше утверждение, необходимо вначале вспомнить некоторые свойства выпуклых множеств.

Определение. Множество точек называется выпуклым, если отрезок, соединяющий две любые его точки, целиком принадлежит данному множеству.

Пусть x^1, x^2, \dots, x^m — угловые точки выпуклого множества D . Следует обратить внимание на то, что здесь и далее верхний индекс означает номер вектора. Нижний индекс используется для скалярных величин, в частности для обозначения соответствующей координаты вектора. Например, x^1 — это вектор, направленный из начала координат в первую угловую точку. Если точки множества D являются точками трехмерного пространства, то координатами вектора x^1 являются числа x^1_1, x^1_2 и x^1_3 .

Для выпуклого множества справедливо утверждение: любая его внутренняя точка может быть представлена как линейная комбинация всех его угловых точек, причем сумма коэффициентов этой линейной комбинации равна единице:

$$x = \sum_{i=1}^m \alpha_i x^i, \quad \sum_{i=1}^m \alpha_i = 1$$

Для одномерного пространства это утверждение очевидно: выпуклым множеством в этом случае является любой отрезок числовой оси. Если a и b — границы отрезка (угловые точки множества), то любая его точка x представима в виде $x = (1 - \lambda)a + \lambda b$, где λ — числовой параметр.

Также нетрудно доказать это утверждение и для случая трех угловых точек, когда выпуклое множество представляет собой треугольник на плоскости. Обобщение на случай любого числа угловых точек проводится методом математической индукции.

Используя это свойство угловых точек, можно доказать, что если решение задачи линейного программирования существует, то оно достигается в угловой точке области допустимых значений, поскольку эта область всегда представляет собой выпуклое множество.

Таким образом, первая геометрическая интерпретация задачи линейного программирования позволяет утверждать, что оптимальное решение, если оно существует, надо искать в одной из угловых точек области допустимых решений. Строгое доказательство этого утверждения можно найти в работе [1].

Лекция 2. Симплекс-метод решения задачи линейного программирования

Для направленного просмотра угловых точек множества допустимых решений применяют симплекс-метод, предложенный американским ученым Дж. Данцигом в 1949 г. и затем усовершенствованный многими математиками.

Этот метод позволяет переходить от одного допустимого базисного решения к другому так, чтобы целевая функция возрастала (или по крайней мере не убывала). При этом на каждом таком шаге имеется возможность проверить критерий оптимальности полученного решения, т.е. установить, можно ли улучшить полученный результат.

Одна из модификаций симплекс-метода состоит в следующем.

Пусть первые m столбцов a^1, a^2, \dots, a^m матрицы A выбраны в качестве базисных, тогда им будут соответствовать m базисных переменных x_1, x_2, \dots, x_m . Остальные $n - m$ переменные $x_{m+1}, x_{m+2}, \dots, x_n$ являются небазисными.

Введем еще одну вспомогательную переменную x_0 и положим

$$x_0 = c_1x_1 + c_2x_2 + \dots + c_nx_n,$$

где c_i ($i = 1, 2, \dots, n$) — коэффициенты целевой функции. Очевидно, что для каждого допустимого плана x_0 принимает значение, равное значению целевой функции. Добавим это уравнение в систему ограничений задачи):

$$\begin{cases} -x_0 + c_1x_1 + c_2x_2 + \dots + c_mx_m + c_{m+1}x_{m+1} + \dots + c_nx_n = 0, \\ a_{11}x_1 + a_{12}x_2 + \dots + a_{1m}x_m + a_{1,m+1}x_{m+1} + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2m}x_m + a_{2,m+1}x_{m+1} + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{k1}x_1 + a_{k2}x_2 + \dots + a_{km}x_m + a_{k,m+1}x_{m+1} + \dots + a_{kn}x_n = b_k, \\ x_0 \geq 0, x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0. \end{cases}$$

Если ввести расширенную матрицу \tilde{A} :

$$\tilde{A} = \begin{pmatrix} -1 & c_1 & c_2 & \dots & c_m & c_{m+1} & \dots & c_n \\ 0 & a_{11} & a_{12} & \dots & a_{1m} & a_{1,m+1} & \dots & a_{1n} \\ 0 & a_{21} & a_{22} & \dots & a_{2m} & a_{2,m+1} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & a_{m1} & a_{m2} & \dots & a_{mm} & a_{m,m+1} & \dots & a_{mn} \end{pmatrix}$$

и расширенный вектор правых частей $B = (b_0 \ b_1 \ b_2 \ \dots \ b_n)^T$ с коэффициентом $b_0 = 0$, то система уравнений относительно $n + 1$ неизвестных x_0, x_1, \dots, x_n будет иметь вид

$$\tilde{A}x = B$$

Легко видеть, что если первые m столбцов исходной матрицы A были линейно независимы, то в расширенной матрице \tilde{A} $m + 1$ -й столбец также будет линейно независимым. С помощью элементарных преобразований (умножение строки на число, сложение двух строк, перестановка столбцов) расширенную матрицу можно привести к виду, в котором каждый базисный столбец будет состоять из нулей и одной единицы, которая стоит на главной диагонали. Тогда система уравнений в развернутом виде будет иметь вид:

$$\left\{ \begin{array}{l} -x_0 + \qquad \qquad \qquad c'_{m+1}x_{m+1} + \dots + c'_n x_n = b'_0, \\ \qquad x_1 + \qquad \qquad \qquad a'_{1,m+1}x_{m+1} + \dots + a'_{1,n}x_n = b'_1, \\ \qquad \qquad x_2 + \qquad \qquad \qquad a'_{2,m+1}x_{m+1} + \dots + a'_{2,n}x_n = b'_2, \\ \qquad \qquad \dots \qquad \qquad \qquad \dots \qquad \qquad \qquad \dots \qquad \qquad \dots \\ \qquad \qquad \qquad \qquad \qquad \qquad x_m + a'_{m,m+1}x_{m+1} + \dots + a'_{m,n}x_n = b'_m, \end{array} \right.$$

где штрихи означают, что в результате преобразований коэффициенты системы получают новые значения.

Таким образом, все базисные переменные входят в уравнения с коэффициентом единица. Как было сказано выше, если мы хотим получить базисное решение, соответствующее выбранным базисным столбцам, необходимо все небазисные переменные положить равными нулю. В этом случае правые части b'_1, b'_2, \dots, b'_m являются значениями базисных переменных x_1, x_2, \dots, x_m . Что касается величины b'_0 , то после указанных преобразований она будет равна значению целевой функции с обратным знаком.

Представление системы ограничений в указанном виде позволяет не только получить некоторое допустимое базисное решение, но и определить, является это решение оптимальным или его можно улучшить.

Если среди коэффициентов $c'_{m+1}, c'_{m+1}, \dots, c'_n$ в первом уравнении нет положительных, то полученное решение является оптимальным. Если хотя бы один из этих коэффициентов положителен, то решение можно улучшить. Для этого нужно заменить один из

базисных столбцов или, что тоже самое, заменить одну из базисных переменных на небазисную.

Существует следующее правило перехода к новому базису. Среди положительных коэффициентов $c'_{m+1}, c'_{m+1}, \dots, c'_n$ выбирается самый большой, его индекс определяет номер той небазисной переменной, которую следует ввести в число базисных. Пусть это будет номер k . Для определения переменной, которую следует вывести из числа базисных, необходимо вычислить отношения b'_i/a'_{ik} ($i = 1, 2, \dots, m$) и среди них выбрать наименьшее (необходимо, чтобы при этом b'_i и a'_{ik} были одного знака, если таких нет, решения не существует). Та строка, в которой отношение минимально, и определит выводимую из набора базисную переменную.

Для задачи об оптимальном распределении ресурсов указанному выше правилу перехода к новому базису можно дать следующую экономическую интерпретацию. Пусть среди коэффициентов $c'_{m+1}, c'_{m+1}, \dots, c'_n$ положительным является коэффициент $c'_k, m < k \leq n$.

Тогда, как это следует из первого уравнения системы, если небазисной переменной x_k дать некоторое положительное значение и перенести в правую часть, то величина x_0 (значение целевой функции, т.е. прибыль) увеличится. Это означает, что текущий набор базисных переменных не является оптимальным.

Если среди коэффициентов $c'_{m+1}, c'_{m+1}, \dots, c'_n$ имеется несколько положительных, следует выбирать для включения в число базисных ту переменную, номер которой совпадает с номером самого большого положительного коэффициента, тогда и увеличение прибыли будет большим.

Что касается переменной, которую следует вывести из числа базисных, то ее индекс q определяется из условия минимума отношения $b'_q/a'_{qk}, 1 \leq q \leq m$. Величина b'_q — это планируемое количество продукции q -го вида, a'_{qk} — затраты k -го вида ресурса на этот вид продукции. Таким образом, из плана исключается продукция, которая требует больше всего ресурсов на единицу продукции.

Существует несколько способов табличной реализации симплекс-метода решения задачи линейного программирования.

Рассмотрим один из них на примере решения задачи, рассмотренной в предыдущей лекции. В канонической форме она имеет вид:

$$\begin{cases} -2x_1 + 3x_2 + x_3 = 14, \\ x_1 + x_2 + x_4 = 8, \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, \\ f(x) = 2x_1 + 7x_2 + 0x_3 + 0x_4 \rightarrow \max. \end{cases}$$

Запишем числовые данные задачи в виде таблицы (табл. 1.1), заменив первый столбец столбцом правых частей $(0, b_1, b_2)$.

Таблица 1.1

Исходная симплекс-таблица

Базисные переменные	x_1	x_2	x_3	x_4	Правые части
x_3	-2	3	1	0	14
x_4	1	1	0	1	8
$f(x)$	2	7	0	0	0

В эту таблицу записаны коэффициенты расширенной матрицы \tilde{A} с небольшими изменениями. Нижнюю строку таблицы иногда называют строкой состояния, поскольку стоящие в ней числа показывают, достигнуто или нет оптимальное решение задачи. Также для удобства в левой части таблицы указываются имена базовых переменных. В данном случае очевидно, что столбцы 3 и 4 матрицы являются линейно независимыми, поэтому их можно выбрать в качестве базисных, соответственно базисными переменными тогда будут x_3 и x_4 . Значение целевой функции находится в правом нижнем углу таблицы, оно равно нулю.

В соответствии с результатами предыдущего раздела из этой таблицы можно сделать вывод, что полученное базовое решение $x = (0, 0, 14, 8)$ не является оптимальным, поскольку в строке состояния есть положительные числа. Наибольшее из них 7, поэтому кандидатом на новый базисный столбец является второй столбец матрицы (он выделен рамкой). Элементом, для которого отноше-

ние b'_i/a'_{ik} минимально, является первый в этом столбце, а первая строка (она выделена рамкой) соответствует базисной переменной x_3 . Таким образом, из числа базисных необходимо вывести переменную x_3 и ввести вместо нее x_2 .

Элемент, стоящий на пересечении выделенного столбца и выделенной строки называется ведущим. Для того чтобы выделенный столбец стал базисным, необходимо сначала все элементы выделенной строки разделить на величину ведущего элемента, а затем вычитать из других строк данную строку, умножая ее на такое число, чтобы остальные элементы в новом базисном столбце оказались равными нулю. Результаты преобразований представлены в табл. 1.2.

Таблица 1.2

Первый шаг преобразований симплекс-таблицы

Базисные переменные	x_1	x_2	x_3	x_4	Правые части
x_2	-2/3	1	1/3	0	14/3
x_4	5/3	0	-1/3	1	10/3
$f(x)$	20/3	0	-7/3	0	-98/3

В строке состояния по-прежнему есть положительное число. Это значит, что поиск оптимального решения надо продолжить. Повторяя описанную выше процедуру, приходим к выводу, что в число базисных нужно ввести переменную x_1 взамен x_4 . Результат второго преобразования показан в табл. 1.3.

Таблица 1.3

Второй шаг преобразований симплекс-таблицы

Базисные переменные	x_1	x_2	x_3	x_4	Правые части
x_2	0	1	2/3	2/5	6
x_1	1	0	-1/5	3/5	2
$f(x)$	0	0	-1	-4	-46

В строке состояния нет положительных чисел, следовательно, достигнуто оптимальное решение $x^* = (2, 6, 0, 0)$, при котором целевая функция $f(x^*) = 46$. Полученное решение, как и следовало ожидать, совпадает с ранее найденным решением этой задачи, полученным геометрическим методом (см. лекцию 1).

В рассмотренном примере базисные решения были получены в такой последовательности: $x = (0, 0, 14, 8)$, $(0, 14/3, 0, 10/3)$, $(2, 6, 0, 0)$. Если обратиться к рис. 1.1, можно заметить, что первые две координаты этих векторов являются координатами угловых точек области допустимых решений D , и переход от одной угловой точки к другой происходит так, что целевая функция все время возрастает, достигая своего максимума.

Лекция 3. Транспортная задача. Метод потенциалов

Имеется n поставщиков и m потребителей однородного продукта, например, типографской краски. Известны стоимости C_{ij} перевозки единицы продукта от i -го поставщика к j -му потребителю. Пусть A_i — количество продукта, имеющегося у i -го поставщика, а B_j — количество продукта, необходимое j -му потребителю. Задача формулируется следующим образом: определить, какое количество X_{ij} продукта нужно перевезти от i -го поставщика к j -му потребителю ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$), чтобы общая стоимость перевозки L была минимальной

$$L = \sum_{i=1}^n \sum_{j=1}^m C_{ij} X_{ij} \rightarrow \min$$

и при этом были выполнены условия

$$\sum_{j=1}^m X_{ij} \leq A_i \quad (i = 1, 2, \dots, n), \quad \sum_{i=1}^n X_{ij} \geq B_j \quad (j = 1, 2, \dots, m),$$

а также условие неотрицательности неизвестных величин:

$$X_{ij} \geq 0 \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, m).$$

Такая задача называется однородной транспортной задачей. Однородной она именуется потому, что в ней речь идет о перевозке лишь одного вида продукта.

Первые n неравенств в данной системе выражают условие, чтобы количество продукта, взятого от i -го поставщика, не превышало то количество, которое у него имеется. Следующие m неравенств выражают условие, чтобы количество продукта, привезенное от различных поставщиков к j -му потребителю, оказалось не меньше того количества, которое этому потребителю необходимо.

Таким образом, имеется mn неизвестных X_{ij} и $m + n$ линейных неравенств, определяющих область допустимых значений для этих неизвестных. Кроме того, задана линейная функция $L = f(X_{ij})$ (целевая функция), которую необходимо минимизировать.

Это стандартная задача линейного программирования, и для ее решения в принципе можно использовать рассмотренный ранее симплекс-метод. Однако система ограничений в этой задаче весьма специфична (матрица коэффициентов при неизвестных состоит только из нулей и единиц), поэтому для ее решения был разработан специальный метод, который будет рассмотрен ниже.

Если суммарное количество продукта, имеющееся у поставщиков, равно тому количеству, которое требуется для всех потребителей, т.е. при выполнении условия

$$\sum_{i=1}^n A_i = \sum_{j=1}^m B_j,$$

задача называется закрытой или сбалансированной и всегда имеет решение. В этом случае все неравенства в системе ограничений заменяются на равенства. В получившейся при этом системе линейных уравнений одно из них будет следствием других, поэтому для определения допустимого решения (допустимого плана перевозок) имеется $r = m + n - 1$ уравнений относительно mn неизвестных (доказательство того, что ранг системы ограничений равен именно r , см., например, в [2]).

Для нахождения начального плана перевозок (любого допустимого решения) и последующего улучшения этого плана разработан табличный метод, который состоит в следующем: строится таблица, состоящая из $m + 1$ строки и $n + 1$ столбца (по числу m потребителей и n поставщиков). В правый крайний столбец заносятся числа B_1, B_2, \dots, B_m , а в нижнюю строку — числа A_1, A_2, \dots, A_n .

После этого начинают заполнять таблицу числами так, чтобы их сумма в каждой строке оказалась равна соответствующему значению B , а сумма по каждому столбцу — соответствующему значению A . Каждая клетка таблицы соответствует одной переменной X_{ij} : первый индекс которой соответствует номеру строки, а второй — номеру столбца.

Некоторые из клеток таблицы могут оставаться пустыми — они соответствуют нулевым значениям переменных X_{ij} . Будем называть такие клетки и соответствующие им переменные небазисными, а заполненные клетки и соответствующие им переменные — базисными.

Существует простой способ построения начального плана перевозок. Он получил название метода северо-западного угла, поскольку заполнение таблицы начинается ее с левого верхнего угла. В эту клетку заносится меньшее из чисел A_1, B_1 . Если $B_1 < A_1$, это означает, что потребности первого потребителя можно полностью удовлетворить за счет запасов первого поставщика. В этом случае в первую строку ничего вносить больше не надо, переходим ко второй строке. В противном случае в первую клетку заносится число A_1 , т.е. первому потребителю передаются все запасы первого поставщика, но, поскольку их потребность в продукте у потребителя остается, заполняется вторая клетка в этой строке — используются запасы второго поставщика, затем третья и так до тех пор, пока потребности первого потребителя не будут удовлетворены полностью. После этого начинают заполнять вторую строку, также двигаясь по ней слева направо.

В результате выполнения указанной процедуры в таблице оказываются заполненными $r = m + n - 1$ клетка, и соответствующие базисные переменные получают некоторые значения. При этом ав-

томатически будут удовлетворены условия задачи: весь продукт, имеющийся у поставщиков, будет распределен между потребителями, при этом каждый из них получит ровно столько, сколько ему нужно.

Метод северо-западного угла позволяет получить начальный план перевозок, но он никак не учитывает стоимости перевозок, поэтому вряд ли полученное с его помощью решение будет оптимальным.

Чтобы улучшить план, нужно перейти к новому базисному решению, аналогично тому, как это делается в симплекс-методе. Для этого нужно изменить таблицу так, чтобы одна из базисных клеток оказалась пустой (т.е. с нулевым значением), а одна из пустых (небазисных) клеток получила неотрицательное значение (стала базисной). При этом, разумеется, суммы по строкам и столбцам таблицы должны остаться прежними, поскольку это условие допустимости любого решения.

Перейти к новому базисному решению можно с помощью так называемого цикла пересчета. Циклом пересчета называется цепочка клеток таблицы перевозок, удовлетворяющих следующим условиям:

- одна из клеток в этой цепочке свободная, остальные — заполненные числами;
- каждые две соседние клетки в цепочке находятся либо в одной строке, либо в одном столбце;
- никакие три соседние клетки, входящие в цепочку, не могут находиться в одной строке или в столбце;
- если каждые две соседние клетки цепочки соединить отрезком прямой так, чтобы получился замкнутый контур из ломаных линий, то одна вершина этого контура будет находиться в свободной клетке таблицы, остальные — в заполненных, т.е. в базисных.

Таких цепочек можно построить столько, сколько имеется свободных (небазисных) клеток в таблице. Построив такую цепочку, можно осуществить переход к другому набору базисных клеток с помощью следующей процедуры:

- последовательность клеток в цепочке помечается чередующимися знаками «+» и «-», при этом свободная клетка помечается знаком «+»;
- среди клеток, помеченных знаком «-», отыскивается та, в которой записана наименьшая величина, эта величина затем прибавляется к числам в клетках, помеченных знаком «+», и вычитается из чисел в клетках, помеченных знаком «-».

В результате такого циклического перемещения будет получен другой набор базисных клеток, при этом сумма чисел в каждой строке и в каждом столбце останутся прежними.

В качестве иллюстрации рассмотрим случай, когда цикл пересчета включает четыре смежные клетки таблицы, при этом небазисная клетка находится на пересечении p -ой строки и q -го столбца: $X_{pq} = 0$.

		Номера столбцов	
		q	$q+1$
Номера строк	p	+	-
	$p+1$	-	+

Рис. 1.2. Цикл пересчета из четырех клеток.
Небазисная (пустая) клетка соответствует переменной X_{pq} .

Пусть значения остальных переменных:

$$X_{p, q+1} = a, \quad X_{p+1, q} = b, \quad X_{p+1, q+1} = c.$$

Предположим для определенности, что наименьшее значение среди базисных переменных, соответствующих клеткам, помеченных знаком «-», имеет переменная, стоящая в одной строке с небазисной, т.е. $a < b$. После циклического перемещения величины a (прибавления к числам, находящимся в клетках со знаком «+» и вычитание из чисел в клетках со знаком «-»), соответствующие переменные окажутся равными:

$$X_{p,q} = a, X_{p,q+1} = 0, X_{p+1,q} = b - a, X_{p+1,q+1} = c + a.$$

Легко видеть, что, во-первых, все значения указанных переменных останутся неотрицательными, а во-вторых, суммы значений переменных, стоящих в одной строке и в одном столбце, остались прежними. Поскольку при такой перестановке другие числа в таблице перевозок не изменяются, неизменными останутся и суммы по любой строке и по любому столбцу.

Рассмотрим теперь вопрос, как повлияет переход к новым базисным переменным на величину целевой функции L . Ясно, что изменения коснутся только тех слагаемых в правой части, которые соответствуют переменным, участвующим в цикле пересчета. Для представленного на рис. 1.2 случая те слагаемые в L , которые соответствуют переменным цикла до перестановки, в сумме давали величину

$$L' = C_{p,q}0 + C_{p,q+1}a + C_{p+1,q}b + C_{p+1,q+1}c,$$

а после перестановки дают величину

$$L'' = C_{p,q}a + C_{p,q+1}0 + C_{p+1,q}(b - a) + C_{p+1,q+1}(c + a).$$

Таким образом, изменение значения целевой функции равно

$$\Delta L = L'' - L' = a(C_{p,q} - C_{p,q+1} - C_{p+1,q} + C_{p+1,q+1}).$$

Отсюда видно, что увеличение или уменьшение значения целевой функции при переходе к новому базису зависит от того, каковы значения элементов матрицы C стоимостей перевозок, соответствующих тем переменным, которые вошли в цикл пересчета, при этом значения стоимостей нужно брать со знаком «+» или «-» в зависимости от того, какой знак был приписан соответствующей клетке матрицы перевозок при построении цикла пересчета.

Указанное правило может быть обобщено на любые циклы пересчета, с любым количеством входящих в них клеток матрицы перевозок.

Обеспечить целенаправленный перебор возможных базисных решений, приводящий к уменьшению целевой функции при переходе к новому базису, позволяет метод потенциалов, который состоит в следующем.

Введем n вспомогательных переменных $\alpha_i, i = 1, 2, \dots, n$ и m переменных $\beta_j, j = 1, 2, \dots, m$ и рассмотрим систему уравнений

$$\alpha_i + \beta_j = C_{ij},$$

где C_{ij} — те элементы матрицы стоимостей, значения индексов i и j которых соответствуют номерам строк и столбцов матрицы перевозок, на пересечении которых находятся базисные клетки. Поскольку на каждой строке и в каждом столбце матрицы перевозок имеется хотя бы одна базисная клетка, в систему войдут все введенные выше вспомогательные переменные.

Как отмечалось выше, число базисных клеток равно $n + m - 1$. Таким образом, в этой системе число уравнений на единицу меньше числа неизвестных. Если одной из неизвестных произвольно задать некоторое значение, например положить $\alpha_1 = 0$, то остальные определяются однозначно.

Найденные значения $\alpha_1, \alpha_2, \dots, \alpha_n$ и $\beta_1, \beta_2, \dots, \beta_m$ называются потенциалами. Им можно придать следующий смысл: если C_{pq} — стоимость перевозки единицы продукта от p -го поставщика q -му потребителю, то α_p и β_q — вклад в эту стоимость со стороны поставщика и потребителя соответственно.

Введем теперь матрицу \hat{C} такую, что $\hat{C}_{ij} = \alpha_i + \beta_j$ ($i = 1, 2, \dots, n$ и $j = 1, 2, \dots, m$). Ее обычно называют матрицей фиктивных стоимостей. Очевидно, что для тех значений индексов i и j , которые соответствуют базисным переменным X_{ij} , элементы фиктивной и исходной матриц стоимостей совпадают, для других значений индексов они могут различаться.

Матрица фиктивных стоимостей, построенная для некоторого набора базисных клеток таблицы перевозок, позволяет определить, является ли текущий набор базисных переменных оптимальным, и, если нет, то какую из небазисных переменных следует включить в

число базисных, чтобы получить лучший план с меньшим значением целевой функции. Для этого составляется матрица разностей между истинной и фиктивной матрицами стоимостей:

$$\Delta C_{ij} = C_{ij} - \hat{C}_{ij} = C_{ij} - (\alpha_i + \beta_j).$$

Если все $\Delta C_{ij} \geq 0$, текущий план перевозок оптимален. Если среди ΔC_{ij} есть отрицательные, план можно улучшить. Пусть ΔC_{pq} — самый большой по абсолютной величине отрицательный элемент, тогда, выбрав переменную X_{pq} в качестве кандидата на включение в число базисных и выполнив для нее цикл пересчета, можно получить новый план перевозок с меньшим значением целевой функции.

Пусть, например, для переменной X_{pq} цикл пересчета включает те клетки таблицы перевозок, которые показаны на рис. 1.2. Тогда при переходе к новому базису целевая функция изменится на величину

$$\Delta L = L'' - L' = a(C_{p,q} - C_{p,q+1} - C_{p+1,q} + C_{p+1,q+1}),$$

где a — наименьшее из значений базисных переменных, вошедших в цикл и соответствующих клеткам, помеченных знаком « \leftarrow ». Поскольку в правой части все слагаемые кроме первого относятся к базисным переменным, можно записать:

$$\begin{aligned} \Delta L &= a(C_{p,q} - \alpha_p - \beta_{q+1} - \alpha_{p+1} - \beta_q + \alpha_{p+1} + \beta_{q+1}) = \\ &= a(C_{p,q} - \alpha_p - \beta_q) = a\Delta C_{pq}. \end{aligned}$$

Но по предположению $\Delta C_{pq} < 0$, причем ΔC_{pq} — самое большое по модулю отрицательное число. Поэтому при переходе к базису с новой базисной переменной X_{pq} целевая функция L уменьшится.

Итак, алгоритм решения однородной транспортной задачи можно описать следующим образом:

- 1) строится начальный план перевозок;
- 2) вычисляется матрица фиктивных стоимостей \hat{C} ;

- 3) если среди элементов матрицы разностей $\Delta C_{ij} = C_{ij} - \hat{C}_{ij}$ нет отрицательных, получен оптимальный план и задача решена;
- 4) в противном случае выбирается небазисная переменная X_{pq} для включения в число базисных из условия, что ΔC_{pq} — самое большое по модулю отрицательное число в матрице разностей;
- 5) после цикла пересчета с переменной X_{pq} переход к п. 2.

Лекция 4. Динамическое программирование

Динамическое программирование — это метод оптимизации процессов, которые могут быть разбиты на этапы или шаги. Каждый такой этап может быть исследован независимо от других и выявлены условия, при которых возможно оптимальное в некотором смысле его выполнение. Зная, как нужно выполнять каждый из этапов и в каком состоянии должна оказаться управляемая система, можно, двигаясь от конца процесса к его началу, построить оптимальную цепочку шагов, которая приведет систему от начального состояния к заданному конечному.

Такая задача возникает, например, при календарном планировании текущего и капитального ремонта полиграфического оборудования и его замены, при распределении дефицитных капитальных вложений между новыми производствами и т.д.

Будем рассматривать дискретный случай задачи динамического программирования, постановка которой состоит в следующем. Рассматривается процесс управления системой S , переходящей из некоторого начального состояния в конечное. Предположим, что управление можно разбить на n шагов и обозначим через X_i управление на i -ом шаге, $i = 1, 2, \dots, n$. На первом шаге система переходит из начального состояния s_0 в состояние s_1 , на втором — в состояние s_2 и т.д.

Здесь и далее будут рассматриваться системы, состояние которых после k -го шага s_k определяется только предыдущим состоянием s_{k-1} и управляющим воздействием X_k , т.е. s_k не зависит от того, каким образом система оказалась в состоянии s_{k-1} и какие были воздействия на предыдущих шагах. Такие системы называются

системами с отсутствием последствия. В таком случае переход системы от одного состояния к другому описывается выражением

$$s_i = \varphi_i (s_{i-1}, X_i), i = 1, 2, \dots, n,$$

которое называется уравнением состояний. На возможные значения величин накладываются ограничения вида

$$a_i \leq s_i \leq b_i, A_i \leq X_i \leq B_i,$$

где a_i, b_i, A_i, B_i — некоторые числовые константы ($i = 1, 2, \dots, n$).

Для оценки качества управления системой вводится интегральный показатель эффективности (критерий оптимальности). Это некоторая функция от вектора состояний системы $S = (s_0, s_1, s_2, \dots, s_n)$, значение которой нужно либо сделать максимальным (если она характеризует, например, прибыль) либо минимальным (если она характеризует убытки):

$$F(S) \rightarrow \max \text{ (либо } \min \text{)}.$$

Если это условие выполнено, то последовательность состояний системы $s_0, s_1, s_2, \dots, s_n$ называется оптимальной траекторией.

Задача динамического программирования труднее задачи линейного программирования, поскольку здесь ищется не один вектор, оптимизирующий целевую функцию, а два — вектор оптимальной траектории S и вектор управляющих воздействий X .

Иногда найти оптимальное решение можно достаточно просто. В некоторых случаях удастся вывести условия, при которых переход из некоторого состояния системы в момент k в другое состояние в следующий момент $k + 1$ происходит оптимальным образом. Эти условия называются разрешающим правилом, которое позволяет строить оптимальную траекторию от начального состояния системы к конечному. Такой способ решения задачи динамического программирования связан с применением так называемого разрешающего правила.

В качестве примера рассмотрим задачу об оптимальной последовательности обработки заготовок деталей.

Имеется n заготовок, пронумерованных от 1 до n . Известно, что время обработки i -й заготовки равно t_i часов, а стоимость хранения этой заготовки в течение часа равна r_i у.е. Требуется определить, в какой последовательности нужно обрабатывать заготовки, чтобы суммарные затраты на хранение всех ожидающих своей очереди заготовок были минимальны.

Итак, процесс обработки разбивается на n этапов, длительность каждого из которых равна продолжительности обработки заготовки, выбранной на этом этапе.

Предположим, что первые k заготовок уже расположены в том порядке, который обеспечивает минимум затрат на хранение, и эти затраты равны $F(k)$. Нужно определить, какую из двух заготовок с номерами $k + 1$ и $k + 2$ следует поставить на обработку в первую очередь. Если первой поставить $k + 1$ -ю, а потом $k + 2$ -ю, то затраты на $k + 2$ -м этапе составят

$$F^1(k + 2) = F(k) + r_{k+1}(t_1 + \dots + t_{k-1} + t_k) + r_{k+2}(t_1 + \dots + t_k + t_{k+1}),$$

а если наоборот, сначала $k + 2$ -ю, а потом $k + 1$ -ю, то

$$F^2(k + 2) = F(k) + r_{k+2}(t_1 + \dots + t_{k-1} + t_k) + r_{k+1}(t_1 + \dots + t_k + t_{k+2}).$$

Вычитая из последнего равенства предыдущее, получим

$$F^2(k + 2) - F^1(k + 2) = r_{k+1}t_{k+2} - r_{k+2}t_{k+1}.$$

Затраты $F^1(k + 2)$ будут меньше, если $r_{k+1}t_{k+2} - r_{k+2}t_{k+1} > 0$, т.е. когда

$$\frac{r_{k+1}}{t_{k+1}} > \frac{r_{k+2}}{t_{k+2}}.$$

Отсюда вытекает разрешающее правило для данной задачи: при выборе последовательности обработки из двух заготовок надо

отдать предпочтение той, у которой отношение стоимости хранения к времени обработки больше.

Пример. Время обработки и стоимости хранения десяти заготовок приведены в табл. 1.4. Необходимо определить оптимальный порядок их обработки.

Таблица 1.4

Данные по десяти заготовкам

i	1	2	3	4	5	6	7	8	9	10
r_i	4,2	2,2	5,3	1,9	2,8	3,9	4,4	6,1	3,1	2,8
t_i	0,5	1,3	2,0	1,8	1,1	0,8	1,9	2,5	0,9	1,4

Решение. Найдем отношения r_i/t_i и отсортируем таблицу в порядке их убывания. Результат приведен в табл. 1.5.

Таблица 1.5

Оптимальная последовательность обработки заготовок

i	1	6	9	3	5	8	7	10	2	4
r_i	4,2	3,9	3,1	5,3	2,8	6,1	4,4	2,8	2,2	1,9
t_i	0,5	0,8	0,9	2,0	1,1	2,5	1,9	1,4	1,3	1,8
r_i/t_i	8,4	4,4	3,4	2,65	2,55	2,44	2,3	2,0	1,5	1,0

Сумма получающихся при этом затрат равна 171,2 у.е. При любом другом порядке обработки сумма затрат будет больше.

Лекция 5. Принцип оптимальности Беллмана

Если для некоторой системы без обратных связей оптимальная траектория от начального состояния s_0 до конечного s_n такова, что любой ее участок на этапах от k до m ($0 \leq k < m \leq n$) также является оптимальным для задачи, в которой s_k является начальным, а s_m — конечным состояниями, то говорят, что данная система удовлетворяет принципу оптимальности Беллмана.

Для таких систем разработан метод динамического программирования, получивший название метода условной оптимизации,

который включает два этапа: на первом строится последовательность управляющих воздействий, оптимизирующих этапы от последнего до первого (обратный ход), а на втором этапе по найденной последовательности управляющих воздействий определяются состояния системы от первого до последнего (прямой ход).

Можно доказать, что динамическая система удовлетворяет принципу Беллмана, если интегральный показатель эффективности управления является аддитивной функцией своих аргументов:

$$F(s_1, s_2, \dots, s_n) = \sum_{i=1}^n f(s_i),$$

где $f(s_i)$ — показатель эффективности на i -м шаге.

Доказательство. Пусть s_0, s_1, \dots, s_n — оптимальная траектория, по которой показатель эффективности $F(s_0, s_1, \dots, s_n)$ достигает оптимального (максимального) значения F_{\max} . Допустим, что s_k — некоторое промежуточное состояние ($k < n$), и существует некоторая последовательность управляющих воздействий, которая переводит систему из состояния s_0 в состояние s_k по оптимальной траектории $s_0, s'_1, \dots, s'_{k-1}, s_k$, отличной от найденной ранее, при этом показатель эффективности будет больше предыдущего:

$$f(s_0) + f(s'_1) + \dots + f(s'_{k-1}) + f(s_k) > f(s_0) + f(s_1) + \dots + f(s_{k-1}) + f(s_k).$$

Тогда, продолжая эту новую траекторию до s_n по ранее найденным состояниям s_{k+1}, \dots, s_n , получим новое решение исходной задачи с более высоким показателем эффективности. Это означает, что прежнее решение не оптимально, что противоречит условию. Отсюда следует, что в этом случае система удовлетворяет принципу оптимальности Беллмана.

Введем вспомогательную функцию

$$W(i) = \sum_{k=i}^n f(s_k), \quad i = 0, 1, \dots, n.$$

Из этой формулы следует, что если показатель эффективности имеет смысл прибыли, полученной при функционировании систе-

мы, то $W(n)$ — прибыль, полученная на последнем этапе, а $W(0)$ — прибыль, полученная на всех этапах функционирования системы.

Введенная таким образом функция используется для нахождения условных оптимальных управлений, начиная от последнего участка траектории. Для этого отыскивается такое значение X_n , которое обеспечивает максимальное значение целевой функции в s_n :

$$W_m(n) = \max_{X_n \in V} f(s_n(X_n)),$$

где V — область допустимых управляющих воздействий, а зависимость $s_n(X_n)$ определяется уравнением состояний $s_n = \varphi_n(s_{n-1}, X_n)$.

Найденное оптимальное управление зависит от s_{n-1} и называется условным оптимальным управлением на n -ом шаге и обозначается $X_n^*(s_{n-1})$. Соответственно, от s_{n-1} зависит и максимум целевой функции $W_m(n)$. Значения X_n^* и $W_m(n)$ для различных значений s_{n-1} можно сохранить в виде таблиц и использовать при дальнейших расчетах.

Найдем теперь максимальное значение целевой функции на двух участках (последнем и предпоследнем):

$$W_m(n-1) = \max_{X_{n-1} \in V} \{f(s_{n-1}(X_{n-1}) + W_m(n))\}.$$

Как и в предыдущем случае, здесь $s_{n-1} = \varphi_{n-1}(s_{n-2}, X_{n-1})$. Таким образом, выражение в фигурных скобках для каждого фиксированного значения s_{n-2} представляет собой функцию от X_{n-1} . Значения X_{n-1}^* , которые дают максимум величине $W_m(n-1)$ для различных s_{n-2} запоминаются для дальнейших расчетов. Отметим, что таблица значений $W_m(n)$, полученная на предыдущем шаге, уже не понадобится, для экономии памяти ЭВМ ее можно удалить и хранить только таблицу, полученную на этом шаге.

Повторяя этот процесс, можно получить последовательность управляющих воздействий X_i^* :

$$W_m(i) = \max_{X_i \in V} \{f(s_i(X_i) + W_m(i+1))\}, \quad i = n-2, n-3, \dots, 1.$$

На i -м шаге отыскиваются X^*_i , которые доставляют максимум функции $W_m(i)$ при различных допустимых значениях s_{i-1} . Табличные значения X^*_i и $W_m(i)$ как функции s_{i-1} запоминаются.

На последнем шаге обратного хода вместо таблицы получается единственное значение $X^*_1(s_0)$, поскольку начальное состояние системы s_0 известно. Полученная при этом величина $W_m(0)$ представляет собой максимально возможное значение целевой функции.

Таким образом, в результате выполнения обратного хода получается оптимальное значение управляющего воздействия $X_1 = X^*_1(s_0)$, а также максимально возможное значение целевой функции F .

После этого начинается прямой ход метода. Вначале вычисляется состояние s_1 , в которое система переходит из s_0 при воздействии X_1 :

$$s_1 = \varphi(s_0, X_1),$$

затем из таблицы $X^*_2(s_1)$, которая была получена на предпоследнем шаге обратного хода, по вычисленному значению определяется оптимальное управление X_2 .

Продолжая таким образом, находятся все остальные значения управляющих воздействий. Прямой ход метода описывается формулой:

$$s_i = \varphi_i(s_{i-1}, X_i), \quad i = 1, 2, \dots, n.$$

Формулы обратного и прямого хода метода решения задачи динамического программирования называются формулами Беллмана.

Замена оборудования на полиграфическом предприятии является важной экономической проблемой. Задача состоит в определении оптимальных сроков замены старого оборудования при его физическом и моральном износе, в результате чего растут расходы на ремонт и обслуживание. Критерием оптимальности в этой задаче является либо прибыль от эксплуатации оборудования (с уче-

том всех затрат на обслуживание и ремонт), либо затраты на обслуживание и ремонт. В первом случае это будет задача максимизации, во втором — минимизации.

При построении динамической модели будем считать, что решение о замене оборудования выносится в начале каждого промежутка времени, на которые разбит весь срок эксплуатации, например в начале календарного года, а состояние оборудования оценивается в конце года. Кроме того, будем предполагать, что оборудование можно эксплуатировать до определенного срока, после чего оно должно быть продано.

Основным параметром, характеризующим оборудование, является возраст, т.е. число лет t , прошедшее с момента его ввода в эксплуатацию. Начальное состояние $s_0 = 0$ означает, что машина новая, после года эксплуатации ее состояние будет $s_1 = 1$, и т.д. до того момента, когда будет принято решение о замене. Если это решение принято в начале k -го года, то $s_{k-1} = k - 1$, а в конце года $s_k = 1$. Таким образом, если принять, что управляющее воздействие может принимать два значения: 0 — сохранить оборудование и 1 — заменить его новым, уравнение состояния на i -м этапе имеет вид:

$$s_i = \begin{cases} s_{i-1} + 1, & \text{если } X_i = 0, \\ 1, & \text{если } X_i = 1, i = 1, 2, \dots, n. \end{cases}$$

Для оценки эффективности управленческих решений на каждом этапе введем функцию затрат $F(s_i, X_i)$. Пусть A — стоимость нового оборудования, а затраты на эксплуатацию и ремонт зависят от возраста t и равны $Z(t) = B(t + 1)$, где B — некоторый числовой коэффициент. После t лет эксплуатации оборудование можно продать за $R(t) = A2^{-t}$ денежных единиц, т.е. цена падает в геометрической прогрессии. Таким образом,

$$F(s_i, X_i) = \begin{cases} B(s_i + 1), & \text{если } X_i = 0, \\ A + B - A2^{-s_i}, & \text{если } X_i = 1. \end{cases}$$

Для того чтобы продемонстрировать применение описанного в предыдущем разделе метода решения данной задачи, рассмотрим конкретный пример, когда $A = 8000$, $B = 1000$. Предположим, что планируется эксплуатация оборудования на период 5 лет, в конце которого оборудование должно быть продано. Стоимость нового оборудования составляет $A = 8000$ у.е., а затраты на обслуживание в течение года $Z(t) = 1000(t + 1)$ у.е., где t — количество лет эксплуатации, предшествующих текущему этапу. Решение о замене оборудования на новое принимается в начале года, при этом выручка от продажи старого оборудования составляет $R(t) = 8000 \cdot 2^{-t}$ у.е.

Разобьем планируемый срок работы оборудования на $n = 5$ этапов.

Найдем оптимальное управляющее решение X_5 , при котором затраты на последнем этапе будут минимальны. Учтем, что возраст оборудования на начало этого этапа может находиться в интервале $1 \leq t \leq 4$. Оптимальные значения X_5^* для каждого значения t приведены в табл. 1.6, там же приведены значения целевой функции — затрат W^* . Затраты подсчитаны с учетом того, что оборудование в конце этого этапа продается.

Таблица 1.6

**Оптимальные управляющие воздействия X_5^*
и соответствующие им затраты W^* на пятом этапе**

s_4	0	1	2	3	4
X_5^*	—	0	0	0	0
$W^*(5)$	—	0	2000	3500	4750

Поясним, как получились значения в правом крайнем столбце таблицы, который соответствует случаю, когда к началу пятого этапа (пятого года эксплуатации) возраст оборудования составлял 4 года, т.е. при $s_5 = 4$.

Тогда при управляющем воздействии $X_5 = 0$ затраты будут равны $B(s_5 + 1) = 1000(4 + 1) = 5000$ у.е., а при $X_5 = 1$ (приобретается новое оборудование, а старое продается) затраты окажутся равными $A - A2^{-4} = 8000 - 8000 \cdot 2^{-4} = 7500$ у.е., т.е. выгоднее последний год эксплуатировать старое оборудование. По условию после пяти

лет оборудование должно быть продано, при этом будет выручено $8000 \cdot 2^{-5} = 250$ у.е. Окончательные затраты при оптимальном управлении $X^*_5 = 0$ будут равны $W^*(5) = 4750$ у.е.

Прочерки в первом столбце отражают тот факт, что, поскольку решение о замене оборудования принимается в начале года, к началу рассматриваемого пятого периода возраст оборудования не может быть меньше единицы.

Рассмотрим теперь четвертый период эксплуатации, к началу которого состояние оборудования s_5 (число лет эксплуатации к началу четвертого периода) может иметь значение 1, 2 или 3.

Найдем оптимальное управление X_4 , если $s_4 = 3$.

При $X_4 = 0$ (оборудование не заменяется) затраты на обслуживание в течение четвертого года составят $B(s_4 + 1) = 1000(3 + 1) = 4000$ у.е., к началу пятого года возраст оборудования станет равным четырем годам, и в соответствии с табл. 1.6 затраты на последнем этапе при оптимальном управлении составят 4750 у.е. Всего затраты на двух последних этапах составят 8750 у.е.

При $X_4 = 1$ (оборудование заменяется новым, а старое продается) затраты на приобретение нового оборудования и его обслуживание в течение четвертого года составят, за вычетом суммы от продажи старого, $A + B - A2^{-3} = 9000 - 8000 \cdot 2^{-3} = 8000$ у.е. К началу пятого года возраст будет $t = 1$, и согласно табл. 1.6 общие затраты на последнем году эксплуатации будут нулевыми. Всего затраты на двух последних этапах составят 8000 у.е.

Итак, если к началу четвертого года эксплуатации возраст оборудования составляет 3 года, то его выгоднее продать и купить новое. Оптимальное управляющее воздействие при этом $X^*_4 = 1$, а максимальные затраты на четвертом и пятом годах эксплуатации составят $W^*(4) = 8000$.

Проведя аналогичные расчеты для $s_4 = 2$ и $s_4 = 1$, получим, что оптимальными управлениями в обоих случаях будут $X^*_4 = 0$, при этом затраты на двух последних этапах составят 6500 и 4000 у.е. соответственно. Результаты расчетов приведены в табл. 1.7.

Таблица 1.7

**Оптимальные управляющие воздействия X^*_4
и соответствующие им затраты $W^*(4)$**

s_3	0	1	2	3	4
X^*_4	–	0	0	1	–
$W^*(4)$	–	4000	6500	8000	–

Рассматривая подобным образом третий и второй этапы, получим оптимальные управления X^*_3 и X^*_2 , а также соответствующие им затраты $W^*(3)$ и $W^*(2)$. Они приведены в табл. 1.8.

Таблица 1.8

**Оптимальные управляющие воздействия X^*_3, X^*_2
и соответствующие им затраты W^***

s_2, s_1	0	1	2	3	4
X^*_3	–	0	0	–	–
$W^*(3)$	–	8500	9500	–	–
X^*_2	–	1	–	–	–
$W^*(2)$	–	9000	–	–	–

На первом этапе эксплуатируется новое оборудование, поэтому затраты за первый год эксплуатации будут включать стоимость его приобретения и обслуживания в течение года, т.е. 9000 у.е. Таким образом, состояние оборудования к началу второго этапа будет $s_1 = 1$ (один год эксплуатации). Расходы за весь период эксплуатации $W^*(1)$ при оптимальном управлении складываются из расходов на первом этапе и значения $W^*(2)$: $9000 + 8000 = 17000$.

С помощью табл. 1.4–1.6 можно определить, как будет изменяться состояние оборудования в процессе эксплуатации при оптимальном управлении.

Поскольку оптимальным на втором этапе является управление $X^*_2 = 1$ и $s_1 = 1$, то $s_2 = 1$. Из табл. 1.6 следует, что следующим оптимальным будет управление $X^*_3 = 0$, следовательно, $s_3 = 2$. Из этой же таблицы находим $X^*_3 = 0$, а $s_4 = 3$. Аналогично из табл. 1.5 видно, что при $s_4 = 3$ оптимальным является управление $X^*_4 = 1$, поэтому $s_4 = 1$ а из табл. 1.4 следует, что $X^*_5 = 0$.

Таким образом, затраты на приобретение и обслуживание оборудования будут минимальными и равными 17000 у.е., если в начале каждого года управляющие воздействия характеризуются вектором $X = (0, 1, 0, 1, 0)$: первый год эксплуатируется новое оборудование, в начале второго оно продается и приобретается новое, которое используется в течение двух лет и затем продается, новое оборудование используется затем до конца планируемого пятилетнего срока. При этом состояние оборудования (его возраст) на конец каждого года эксплуатации характеризуется вектором $S = (1, 1, 2, 1, 2)$.

Лекция 6. Многокритериальные задачи

В задачах линейного программирования, рассмотренных ранее, необходимо было найти такое решение из области допустимых, при котором выполняется критерий оптимальности — целевая функция принимает максимальное или минимальное решение.

На практике часто встречаются задачи, когда критериев оптимальности несколько, при этом, как правило, не существует решения, при котором выполняются одновременно все критерии оптимальности. В таких случаях приходится идти на компромисс, отдавая предпочтение одному из критериев. Обычно при этом руководствуются соображениями, вытекающими из особенностей решаемой задачи, и эти соображения часто носят субъективный характер.

Существует, однако, способ получения компромиссного решения многокритериальной задачи, в определенном смысле оптимального. Этот способ основан на понятии множества Парето.

Рассмотрим замкнутое множество Ω точек на плоскости с декартовой системой координат, по оси абсцисс которой определяются координата u точки, а по оси ординат — координата v . Замкнутость означает, что граница принадлежит области.

Говорят, что точка принадлежит множеству Парето данной области, если перемещение из нее в любую другую точку области возможно только, если при этом либо обе координаты уменьшают-

ся, либо уменьшается одна из них, в то время как другая растет или остается постоянной. На рис. 1.3. точка B является одной из таких точек.

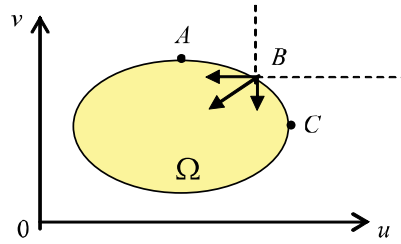


Рис. 1.3. Точка A , удовлетворяющая условию Парето

Другими словами, точка из Ω принадлежит множеству Парето, если любая другая точка, находящаяся по отношению к данной в первом квадранте (на рис. 1.3 показан пунктиром) не принадлежит Ω . Легко проверить, что для представленной на этом рисунке области множество Парето образуют те точки на границе Ω , которые лежат на дуге AB .

Рассмотрим следующую задачу: в области D на плоскости (x, y) найти точку, в которой две функции $f(x, y)$ и $g(x, y)$ принимают наибольшие значения.

Ясно, что в общем случае максимумы функций $f(x, y)$ и $g(x, y)$ достигаются в различных точках области D , поэтому речь может идти о нахождении такой точки в D , в которой значения заданных функций будут близки к максимальным. Очевидно, что таких точек можно указать много, одни из них будут находиться ближе к точке максимума функции $f(x, y)$, другие — к точке максимума функции $g(x, y)$. Одним из способов нахождения компромиссного решения является метод идеальной точки, который состоит в следующем.

Введем две переменные u и v , связанные с переменными x и y соотношениями $u = f(x, y)$, $v = g(x, y)$.

Эти соотношения задают отображение точек области D на точки области Ω (рис. 1.4). Допустим, что уравнения перехода можно разрешить относительно x и y , т.е. выразить их через u и v :

$$x = F(u, v),$$

$$y = G(u, v).$$

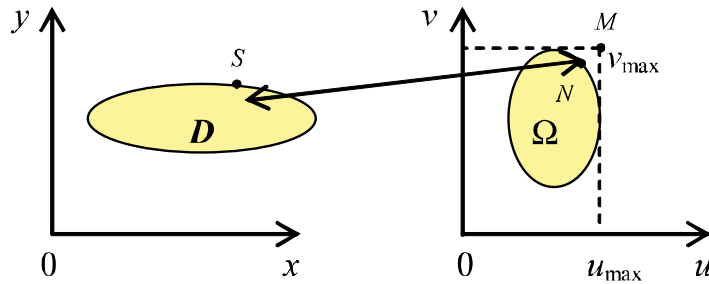


Рис. 1.4. Точка M с координатами (u_{\max}, v_{\max}) называется идеальной точкой

Тем самым между точками областей D и Ω устанавливаются взаимно однозначные отношения. Точка M с координатами (u_{\max}, v_{\max}) называется идеальной точкой (рис. 1.5). Ближайшая к ней точка N , лежащая на границе области Ω , очевидно, принадлежит множеству Парето. По приведенным выше формулам можно определить координаты соответствующей ей точки S области D .

Координаты x_s, y_s найденной указанным выше способом точки S и принимаются в качестве оптимального решения рассматриваемой двухкритериальной задачи. Действительно, любая внутренняя точка области Ω имеет обе координаты, меньшие чем у N . Лишь соседние с N точки, принадлежащие границе области Ω , обладают тем свойством, что при переходе в них из N одна из координат возрастает, а другая уменьшается. Это означает, что, взяв эти точки и определив соответствующие им точки области D , мы будем получать решения, в которых один критерий увеличивается за счет уменьшения другого. И лишь точка $N \in \Omega$ и соответствующая ей точка $S \in D$ определяют решение, оптимизирующее одновременно оба критерия.

Пример. Найти максимально возможные значения функций

$$u = x + y + 2,$$

$$v = x - y + 6$$

при условиях

$$\begin{cases} 0 \leq x \leq 4, \\ 0 \leq y \leq 2, \\ X + 2y \leq 4. \end{cases}$$

Функции, задающие отображение $D \rightarrow \Omega$, являются линейными, поэтому отрезки прямой отображаются в отрезки прямой и, если D — выпуклый многоугольник, то Ω также будет выпуклым многоугольником. Для его построения достаточно определить, в какие точки плоскости (u, v) отобразятся угловые точки A, B, C, D, E . Так, точка E с координатами $(4; 0)$ отобразится в точку E' с координатами $(6; 10)$, а точка D $(4; 1)$ — в точку D' $(7; 9)$.

Формулы перехода от Ω к D имеют вид:

$$x = (u + v) / 2 - 4,$$

$$y = (u - v) / 2 + 2.$$

Внутренние точки отрезка $E'D'$ образуют множество Парето для области Ω . Идеальная точка M имеет координаты $(7; 10)$, а ближайшая к ней точка N' области D лежит на пересечении линии $E'D'$ и перпендикуляра к ней, проведенного из точки M . Проведя соответствующие вычисления, нетрудно показать, что точка N' имеет координаты $(6,5; 9,5)$. Подставив эти числа в правые части уравнений и решив их, получим $x^* = 4, y^* = 0,5$. Это и будет оптимальное решение двухкритериальной задачи.

Лекция 7. Нелинейное программирование

В большинстве случаев ставится задача оптимизации, по форме совпадающая с рассмотренной ранее задачей линейного программирования: найти максимум (или минимум) целевой функции n переменных

$$F(x_1, x_2, \dots, x_n) \rightarrow \max \text{ (или } \min)$$

при условии, что переменные x_1, x_2, \dots, x_n удовлетворяют системе неравенств (ограничений)

$$\varphi_i(x_1, x_2, \dots, x_n) \leq b_i, i = 1, 2, \dots, m, m < n.$$

Иногда система ограничений имеет вид:

$$\varphi_i(x_1, x_2, \dots, x_n) = 0, i = 1, 2, \dots, m, m < n.$$

В этом случае ограничения называются уравнениями связей.

Для общего случая нет методов решения подобных задач. Существующие методы разработаны при некоторых допущениях относительно функций F и φ_i .

Далее будем предполагать, что как целевая функция, так и функции, входящие в условия ограничений, являются дифференцируемыми. Кроме того, ограничимся случаем, когда область допустимых решений D , определяемая системой ограничений, является выпуклой, а целевая функция имеет лишь одну точку экстремума (максимума или минимума).

Для определенности будем рассматривать задачу нахождения минимума целевой функции в области D допустимых значений ее аргументов.

Определение. Функция $F(X) = F(x_1, x_2, \dots, x_n)$ называется выпуклой в области D n -мерного пространства, если для любых точек X^1, X^2 , принадлежащих D , и любого числа $\alpha \in [0, 1]$ выполняется условие:

$$F(\alpha X^1 + (1 - \alpha)X^2) \leq \alpha F(X^1) + (1 - \alpha)F(X^2).$$

Если в этом выражении стоит знак строгого неравенства, функция называется строго выпуклой. Примерами строго выпуклых функций одной переменной являются: $y = x^2$ на всей числовой оси, $y = e^x$ на интервале $(0, \infty)$, $y = \sin(x)$ на отрезке $[\pi, 2\pi]$.

Отметим два важных свойства выпуклых функций:

- любая линейная комбинация нескольких выпуклых функций является выпуклой функцией;
- если функция $F(X)$ является выпуклой, то множество решений неравенства $F(X) < \alpha$ является либо пустым, либо выпуклым.

Если функция $F(x_1, x_2, \dots, x_n)$ дважды дифференцируема, то для того, чтобы проверить, является ли она выпуклой, можно применить критерий Сильвестра: функция n переменных выпукла тогда и только тогда, когда неотрицательны все главные миноры матрицы вторых частных производных $a_{ij} = \partial^2 F(X) / \partial x_i \partial x_j$, т.е. определители $\Delta_1 = a_{11}$, $\Delta_2 = (a_{11}a_{22} - a_{12}a_{21})$ и т.д., до определителя Δ_n всей матрицы. Если все миноры строго положительны, т.е. $\Delta_1 > 0$, $\Delta_2 > 0$, ..., $\Delta_n > 0$, то функция $F(x_1, x_2, \dots, x_n)$ является строго выпуклой.

В том случае, если и целевая функция, и функции, входящие в систему ограничений, являются выпуклыми, такая задача называется задачей выпуклого программирования. Для этих задач разработаны специальные методы решения. В частности, если целевая функция и функции в системе ограничений являются линейными по своим переменным, то получается задача линейного программирования, рассмотренная ранее.

Пример. Задача о консервной банке.

Необходимо изготовить из куска жести консервную банку заданного объема V в виде цилиндра высотой h с радиусом основания r . Требуется отыскать такие значения параметров h и r , при которых а) минимальна площадь поверхности банки S ; б) минимальна длина шва L .

Объем банки, площадь ее поверхности и длина шва рассчитываются по формулам:

$$V = \pi r^2 h, S = 2\pi r^2 + 2\pi r h, L = 4\pi r + h.$$

Сначала рассмотрим задачу минимизации

$$S(r, h) \rightarrow \min$$

при условии (ограничении) $\pi r^2 h - V = 0$.

В данном случае уравнение, определяющее ограничение, легко разрешить относительно h :

$$h = V/(\pi r^2).$$

Поэтому задача сводится к нахождению минимума функции одной переменной r . Необходимым условием минимума является равенство нулю производной:

$$dS/dr = 4\pi r - 2V/r^2 = 0.$$

Решением этого уравнения является $r^* = (V/2\pi)^{1/3}$. Вторая производная для всех $r > 0$ положительна:

$$d^2S/dr^2 = 4\pi + 4V/r^3 > 0,$$

поэтому r^* является точкой минимума. Соответствующее значение параметра h равно $h^* = V/(\pi r^{*2})$.

Вторая задача минимизации

$$L(r, h) \rightarrow \min,$$

$$\pi r^2 h - V = 0.$$

решается аналогично. Минимум достигается при $r^{**} = (V/2\pi^2)^{1/3}$, $h^{**} = 2\pi r^{**}$.

Необходимым условием экстремума дифференцируемой функции $F(X)$ в точке X^* является равенство нулю всех ее частных производных:

$$\partial F(X)/\partial x_i |_{X^*} = 0, i = 1, 2, \dots, n.$$

Точек, в которых производные равны нулю, может быть несколько. Они называются стационарными или точками, подозрительными на экстремум. Для того чтобы из этих точек выбрать ту, в которой достигается максимум или минимум функции, необхо-

дим сравнить между собой значения функции, вычисленные в этих точках.

В качестве примера рассмотрим функцию

$$F(X) = x_1^4 + x_2^4 - x_1^2 - 2x_1x_2 - x_2^2.$$

Приравняв частные производные нулю, получим систему уравнений

$$\begin{cases} 4x_1^3 - 2x_1 - 2x_2 = 0, \\ 4x_2^3 - 2x_1 - 2x_2 = 0. \end{cases}$$

Решениями системы являются: $X^1 = (0; 0)$, $X^2 = (1; 1)$, $X^3 = (-1; -1)$. Таким образом, имеются три стационарные точки, в которых возможен экстремум.

Вычислим вторые производные:

$$\begin{aligned} \partial^2 F / \partial x_1 \partial x_1 &= 12x_1^2 - 2, \\ \partial^2 F / \partial x_1 \partial x_2 &= \partial^2 F / \partial x_2 \partial x_1 = -2, \\ \partial^2 F / \partial x_2 \partial x_2 &= 12x_2^2 - 2. \end{aligned}$$

Составим матрицу из вторых производных, так называемую матрицу Гессе, или гессиан функции F :

$$A = \begin{bmatrix} 12x_1^2 - 2 & -2 \\ -2 & 12x_2^2 - 2 \end{bmatrix}.$$

Ее главные миноры (определители):

$$\Delta_1 = 12x_1^2 - 2, \quad \Delta_2 = (12x_1^2 - 2)(12x_2^2 - 2) - 4.$$

Подставляя координаты стационарных точек, получаем, что для первой значения определителей $\Delta_1 = -2$, $\Delta_2 = 0$, для второй и третьей $\Delta_1 = 10$, $\Delta_2 = 96$ соответственно. Таким образом, по крите-

рию Сильвестра точки X^2 и X^3 являются для функция $F(X)$ точками строгого минимума.

При ограничениях на допустимые значения переменных задача отыскания минимума или максимума функции усложняется. В этом случае говорят о нахождении условного экстремума функции.

Рассмотрим задачу отыскания минимума.

$$F(x_1, x_2, \dots, x_n) \rightarrow \min$$

при наличии связей между переменными

$$\varphi_i(x_1, x_2, \dots, x_n) = 0, i = 1, 2, \dots, m, m < n.$$

Один из способов определения условного минимума функции состоит в том, что уравнения связей разрешаются относительно m переменных, например x_1, x_2, \dots, x_m , т.е. они явно выражаются через остальные переменные. В таком случае F становится функцией $n - m$ переменных, и задача сводится к нахождению безусловного минимума.

Если не удастся выразить одни переменные через другие, применяют другой метод, получивший название метод множителей Лагранжа.

Составляется функция, которая называется функцией Лагранжа:

$$L = F(X) + \lambda_1 \varphi_1(X) + \lambda_2 \varphi_2(X) + \dots + \lambda_m \varphi_m(X),$$

где λ_i — пока неизвестные числовые множители (множители Лагранжа).

Эта функция зависит от $n + m$ переменных $x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_m$. Необходимым условием экстремума функции является равенство нулю частных производных по всем переменным:

$$\begin{cases} \partial L / \partial x_i = 0, i = 1, 2, \dots, n, \\ \partial L / \partial \lambda_j = 0, j = 1, 2, \dots, m. \end{cases}$$

Поскольку $\partial L / \partial \lambda_j = \varphi_j(X)$, в стационарных точках $\varphi_j(X) = 0$. Поэтому, если X^* — точка минимума функции $L(X)$, она же является точкой условного минимума функции $F(X)$.

Лекция 8. Численные методы нахождения экстремума

Одним из самых простых является метод покоординатного спуска. Выбирается начальная точка X^0 , которую называют начальным приближением, и вычисляется значение функции $F(X^0)$. Одной из координат, например первой, дают приращение $+\lambda$ и $-\lambda$ и определяют, в каком случае значение функции уменьшается, после чего координату изменяют на эту величину до тех пор, пока значение функции не перестанет уменьшаться. После этого изменяют другую координату, затем третью и т.д. Для случая функции двух переменных $f(x_1, x_2)$ этот процесс показан на рис. 1.5.

Недостатком данного метода является то, что он никак не учитывает свойства функции $f(x_1, x_2)$.

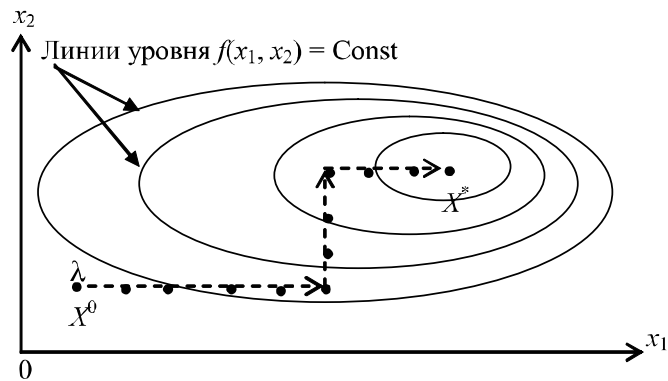


Рис. 1.5. Метод покоординатного спуска

Более эффективным является градиентный метод, который применяется в случае, когда функция нескольких переменных является дифференцируемой. Здесь, как и в предыдущем случае, выбирается начальное приближение X^0 , но шаг в сторону точки ми-

нимума делается не вдоль какой-либо координатной линии, а в сторону наискорейшего убывания функции.

Определение. Градиентом функции $f(x_1, x_2, \dots, x_n)$ в точке X называется вектор, обозначаемый ∇f (читается «набла эф»), компонентами которого являются частные производные от этой функции по всем переменным, вычисленные в точке X :

$$\nabla f = (\partial f / \partial x_1, \partial f / \partial x_2, \dots, \partial f / \partial x_n).$$

Этот вектор показывает направление из точки X , в котором функция возрастает быстрее всего. Противоположно направленный ему вектор $-\nabla f$ называется антиградиентом, он указывает направление скорейшего убывания функции.

Алгоритм нахождения минимума градиентным методом можно описать следующим образом:

- 1) выбирается начальное приближение X^0 ;
- 2) вычисляются частные производные от f в этой точке;
- 3) если все производные близки к нулю (меньше по модулю некоторого порогового значения), то найденное решение принимается за точку минимума, и на этом процесс вычислений заканчивается;
- 4) в противном случае из точки X^0 делается шаг длиной λ в сторону наискорейшего убывания функции по формуле $X^1 = X^0 - \lambda \nabla f / |\nabla f|$. Найденная точка X^1 принимается за новое начальное приближение X^0 , после чего происходит переход к п. 2.

Схематично этот процесс показан на рис. 1.6.

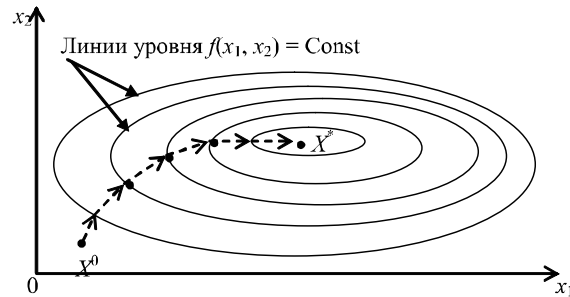


Рис. 1.6. Метод градиентного спуска

Описанный алгоритм имеет множество модификаций, связанных с выбором величины шага λ и его уточнения по мере приближения к точке минимума.

Более эффективным является метод сопряженных градиентов, который позволяет выбрать оптимальное значение λ на каждом шаге. В этом методе используется тот факт, что градиент функции перпендикулярен линии уровня, проходящей через точку, в которой вычисляются частные производные.

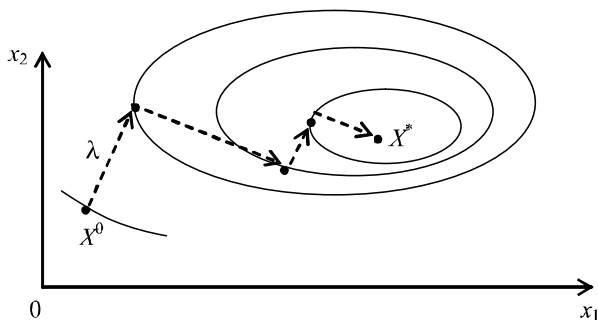


Рис. 1.7. Метод сопряженных градиентов

Пусть X^0 — начальное приближение и $\nabla f(X^0)$ — вектор градиента в этой точке. Обозначим λ — шаг от X^0 до следующей точки X^1 . Линия, по которой происходит перемещение (показана пунктиром на рис. 1.7), является касательной к линии уровня $f(X) = f(X^1)$, поэтому должно выполняться условие перпендикулярности

$$\nabla f(X^0) \nabla f(X^1) = 0.$$

Подставив сюда $X^1 = X^0 - \lambda \nabla f(X^0)$, получим уравнение относительно неизвестной величины λ . Решив его, определим величину шага и координаты точки X^1 .

В качестве примера рассмотрим задачу нахождения минимума функции

$$f(X) = 2x_1^2 + x_2^2 - 4(x_1 + x_2) + 6.$$

Частные производные этой функции равны

$$\partial f / \partial x_1 = 4x_1 - 4, \quad \partial f / \partial x_2 = 2x_2 - 4.$$

Пусть $X^0 = (0, 0)$ — начальное приближение. Градиент в этой точке

$$\nabla f(X^0) = (-4, -4).$$

Обозначим λ коэффициент, определяющий величину шага из начальной точки в точку X^0 в направлении антиградиента:

$$X^1 = X^0 - \lambda \nabla f(X^0) = (4\lambda, 4\lambda).$$

Подставляя координаты этой точки, найдем градиент в точке X^1 :

$$\nabla f(X^1) = (16\lambda - 4; 8\lambda - 4).$$

Условие перпендикулярности

$$\nabla f(X^0) \nabla f(X^1) = (-4; -4)(16\lambda - 4; 8\lambda - 4) = -96\lambda + 32 = 0.$$

Отсюда $\lambda = 1/3$, $X^1 = (4/3; 4/3)$ — следующее приближение и $\nabla f(X^1) = (4/3; -4/3)$ — градиент в этой точке.

Повторяя расчеты по приведенным выше формулам, в которых X^0 и X^1 нужно заменить на X^1 и X соответственно, получим $X^2 = (4/3 - \lambda \cdot 4/3; 4/3 + \lambda \cdot 4/3)$ и $\nabla f(X^2) = (4/3 - \lambda 16/3; -4/3 + \lambda \cdot 8/3)$. Из условия перпендикулярности $\nabla f(X^1)$ и $\nabla f(X^2)$ находим $\lambda = 1/3$ и $X^2 = (4/3 - 4/9; 4/3 + 4/9) = (8/9; 16/9)$ и $\nabla f(X^2) = (-4/9; -4/9)$.

Третий шаг в сторону точки минимума функции $f(X)$, сделанный аналогично первым двум, позволяет найти точку X^3 с координатами $(1,04; 1,96)$, четвертый шаг уже приводит в точку минимума $X^* = (1; 2)$.

Следует подчеркнуть, что применение численных методов подразумевает задание допустимой погрешности результата ε . Эта величина определяет размер такой окрестности с центром в точке истинного экстремума, при попадании в которую при очередной итерации задачу можно считать решенной.

Лекция 9. Недетерминированные системы

Рассмотрим систему, которая в каждый момент времени может находиться в одном из нескольких состояний. Примером может служить печатная машина, которая может находиться в состоянии работы, подготовки к работе, ожидания, ремонта и т.д. В процессе эксплуатации такая машина может переходить из одного состояния в другое, причем эти переходы могут носить случайный характер. Поэтому предсказать, в каком состоянии окажется система в последующие моменты времени, невозможно, можно лишь рассчитать вероятности системы оказаться в том или ином состоянии, если известны условные вероятности переходов из одного состояния в другое.

Обозначим S_1, S_2, \dots, S_n возможные состояния системы, а $P_1(t), P_2(t), \dots, P_n(t)$ — вероятности системы оказаться в каждом из n возможных состояний в некоторый момент времени t . Переходы из одного возможного состояния в другое образуют полную группу событий, поэтому

$$P_1(t) + P_2(t) + \dots + P_n(t) = 1.$$

Время будем считать дискретным: $t = 0, 1, 2, \dots, k$.

Прогнозируемое состояние системы в каждый момент времени будем характеризовать вектором-строкой $P(t) = (P_1(t), P_2(t), \dots, P_n(t))$. Если в начальный момент состояние системы известно, то одна из координат вектора $P(t)$ равна единице, остальные равны нулю.

Будем считать, что переход из состояния S_i в состояние S_j определяется величиной P_{ij} , которая называется переходной вероятностью. Если известны переходные вероятности из любого возможного состояния в каждое из возможных состояний, т.е. задана матрица вероятностей переходов M , то, зная вектор состояния в момент t , можно определить этот вектор в следующий момент $t + 1$ по формуле

$$P(t + 1) = P(t)M.$$

Таким образом, состояние системы в некоторый момент времени определяется ее состоянием в предыдущий момент, при этом не имеет значения, каковы были предыдущие состояния, в том числе и начальное.

Процесс, в результате которого состояние системы в некоторый момент времени зависит лишь от состояния в предыдущий момент времени и определяется переходными вероятностями, называется марковским.

Если начальное состояние системы $P(0)$ известно, то в следующий момент оно будет характеризоваться вектором $P(1) = P(0) \cdot M$. В следующие моменты времени $P(2) = P(1)M = P(0)M^2$, ..., $P(k) = P(0)M^k$.

При больших k система переходит в стационарное состояние. Это означает, что $P(k)$ и $P(k+1)$ почти не различаются, т.е. $P(k+1) = P(k)M = P(k)$.

Таким образом, для того чтобы найти установившийся режим работы системы, можно использовать матричное уравнение

$$P(E - M) = 0.$$

Здесь E — единичная матрица, а в правой части стоит нулевой вектор — строка из нулей. Однородная система уравнений имеет решения, если определитель матрицы коэффициентов равен нулю: $\det(E - M) = 0$, при этом решений бесконечно много. Единственное решение можно получить, если добавить это уравнение, которое справедливо для любых t , в том числе и в установившемся режиме.

Пример. Рассмотрим систему, которая может находиться в двух состояниях: 1 — включенном и 2 — выключенном. В начальный момент система была включена: $P(0) = (1; 0)$. На рис. 1.8 вероятности переходов заданы в виде диаграммы состояний.

Матрица вероятностей переходов в этом случае имеет вид

$$M = \begin{bmatrix} 0,4 & 0,6 \\ 0,3 & 0,7 \end{bmatrix} .$$

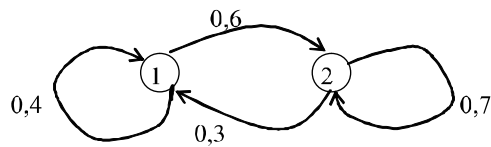


Рис. 1.8. Диаграмма состояний системы из двух элементов

Чтобы определить состояние системы в следующий момент после начального, необходимо найти произведение $P(0)M$:

$$(P_1(1), P_2(1)) = \begin{bmatrix} 0,4 & 0,6 \\ 0,3 & 0,7 \end{bmatrix} (1; 0) = (0,4; 0,6).$$

Для определения установившегося режима $P^* = (P^*_1, P^*_2)$ воспользуемся уравнением:

$$(P^*_1, P^*_2) \begin{bmatrix} 0,6 & -0,6 \\ -0,3 & 0,3 \end{bmatrix} = (0; 0).$$

Отсюда $0,6P^*_1 - 0,3P^*_2 = 0$. Добавив условие $P^*_1 + P^*_2 = 1$, получаем систему из двух уравнений, решив которую, получим $P^*_1 = 1/3$ и $P^*_2 = 2/3$.

Рассмотрим теперь процессы с непрерывным временем.

Пусть по-прежнему система может переходить из некоторого состояния в любое из n возможных, но в случайные моменты времени, которое предполагается непрерывным. В этом случае исходными данными для изучения протекания марковского процесса являются не переходные вероятности, а интенсивности переходов между состояниями A_{ij} . В этом случае условная вероятность перехода из i -го в j -е состояние за время Δt равна $A_{ij} \Delta t$.

Следует отметить, что в матрице интенсивностей переходов A диагональные элементы не задаются, т.е. элементы A_{ij} , $i = 1, 2, \dots, n$ не определены.

Как и в случае с дискретным временем, введем вектор $P(t) = (P_1(t), P_2(t), \dots, P_n(t))$, i -я компонента которого является вероятностью системы оказаться в состоянии S_i в момент t .

Пусть в момент t вероятность системы оказаться в k -м состоянии равна $P_k(t)$. Какова вероятность системы оказаться в этом же состоянии в момент $t + \Delta t$?

Для того чтобы вывести формулы для расчета вероятности $P_k(t + \Delta t)$, рассмотрим два взаимоисключающих случая: в момент t система находилась в k -м состоянии и в момент t система не находилась в k -м состоянии.

Если система находилась в k -м состоянии (событие ω_k), то вероятность за время Δt перейти в i -е состояние ($i \neq k$) равна $A_{ij}\Delta t$. Поэтому условная вероятность остаться в k -м состоянии (событие Ω_k) равна

$$P(\Omega_k / \omega_k) = 1 - \sum_{i=1, i \neq k}^n A_{ki} \Delta t.$$

Эту формулу можно упростить, если ввести обозначение

$$A_{kk} = - \sum_{i=1, i \neq k}^n A_{ki}.$$

Напомним, что в матрице интенсивностей переходов A диагональные элементы не заданы. Доопределим ее, положив диагональные элементы равными сумме остальных элементов в соответствующих строках со знаком «-». Такую модифицированную матрицу интенсивностей будем обозначать по-прежнему буквой A .

В этом случае приведенная выше формула принимает вид

$$P(\Omega_k / \omega_k) = 1 + A_{kk} \Delta t.$$

Рассмотрим теперь случай, когда в момент t система не находилась в состоянии k . Если система находилась в i -м состоянии (событие ω_i), то условная вероятность за время Δt перейти в k -е состояние равна

$$P(\Omega_k / \omega_i) = A_{ik} \Delta t, \quad i = 1, 2, \dots, k-1, k+1, \dots, n.$$

в виде приведенной ниже диаграммы (рис. 1.9). Начальное состояние определяется вектором $P(0) = (0, 1, 0)$. Найти состояния в последующие моменты времени и в установившемся режиме.

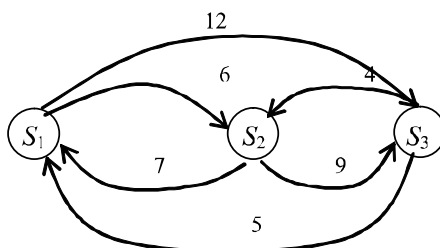


Рис. 1.9. Интенсивности переходов между состояниями системы

Матрица частот переходов для рассматриваемого случая системы с тремя возможными состояниями и непрерывным временем имеет вид

$$A_{\text{пер}} = \begin{pmatrix} * & 6 & 12 \\ 7 & * & 9 \\ 5 & 4 & * \end{pmatrix}.$$

Диагональные элементы в этой матрице не определены, поскольку, как было сказано выше, частоты переходов системы в те же самые состояния не задаются. Модифицируем матрицу частот переходов, доопределив ее диагональные элементы отрицательными значениями, такими, что сумма элементов в каждой строке оказывается равной нулю

$$A = \begin{pmatrix} -18 & 6 & 12 \\ 7 & -16 & 9 \\ 5 & 4 & -9 \end{pmatrix}.$$

Если обозначить вероятности системы оказаться в момент t в состояниях S_1, S_2, S_3 через $P_1(t), P_2(t)$ и $P_3(t)$ соответственно, то определить эти вероятности при известных вероятностях состояний в

начальный момент можно с помощью системы дифференциальных уравнений Колмогорова — Чепмена.

В координатной форме для приведенного выше случая матрицы A эта система имеет вид

$$dP_1(t)/dt = -18P_1(t) + 7P_2(t) + 5P_3(t),$$

$$dP_2(t)/dt = 6P_1(t) - 16P_2(t) + 4P_3(t),$$

$$dP_3(t)/dt = 12P_1(t) + 9P_2(t) - 9P_3(t),$$

с начальными условиями $P_1(0) = 0$, $P_2(0) = 1$, $P_3(0) = 0$.

Учитывая, что в любой момент времени

$$P_1(t) + P_2(t) + P_3(t) = 1,$$

можно понизить порядок системы, исключив одну из неизвестных функций, например $P_3(t)$:

$$dP_1(t)/dt = -23P_1(t) + 2P_2(t) + 5,$$

$$dP_2(t)/dt = 2P_1(t) - 20P_2(t) + 4,$$

$$P_3(t) = 1 - P_1(t) + P_2(t),$$

$$P_1(0) = 0, P_2(0) = 1.$$

Решение можно получить аналитически, используя известный метод для систем линейных дифференциальных уравнений с постоянными коэффициентами. Другой способ — использовать пакет MathCad, который позволяет получить графики искомых функций $P_1(t)$, $P_2(t)$ и $P_3(t)$ (показаны на рис. 1.110).

Из рис. 1.10 видно, что начиная с некоторого значения t , решения начинают приближаться к некоторым постоянным значениям — к установившемуся режиму. Найти вероятности состояний системы оказаться в одном из установившихся режимов можно, не интегрируя систему уравнений Колмогорова. Для достаточно больших t величины $P_i(t)$ становятся практически постоянными, это означает, что производные от этих функций по времени близки к нулю.

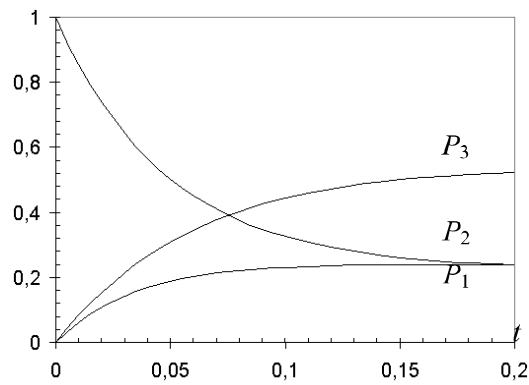


Рис. 1.10. Графики зависимостей вероятностей состояний системы

Заменяя в системе дифференциальных уравнений производные нулями и заменяя $P_1(t)$ и $P_2(t)$ на P_1^* и P_2^* , приходим к системе алгебраических уравнений относительно двух неизвестных, решая которую найдем, что $P_1^* \approx 0,237$ и $P_2^* \approx 0,224$. Вероятность третьего состояния в установившемся режиме найдем, учитывая, что в сумме все три величины P_1^* , P_2^* и P_3^* должны давать единицу, поэтому $P_3^* \approx 0,539$. Как и следовало ожидать, полученные решения совпадают с найденными ранее решениями системы дифференциальных уравнений, если в последних сделать предельный переход при $t \rightarrow \infty$.

ТЕМА 2

АВТОМАТИЗИРОВАННЫЕ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ

Лекция 10. Общая характеристика АИУС

Управление — это функция организованных систем (технических, экономических, социальных), обеспечивающую сохранение их определенной структуры, поддержание режима деятельности, реализацию программы, направленной на достижение цели функционирования системы. Автоматизацией управления называется замена труда человека машиной при выполнении операций управления. Объект управления в совокупности с управляющим комплексом образует систему управления. В том случае, когда часть операций управления выполняет машина, а другую часть — человек, система носит наименование автоматизированной системы управления (АСУ).

Внедрение новых информационных технологий управления позволяет перейти к качественно новому типу автоматизированных систем — интегрированных систем обработки информации и управления или, иначе говоря, автоматизированных информационно-управляющих систем (АИУС).

Классификация АИУС может быть проведена по различным признакам:

- по типу объекта управления;
- по типу закона управления;
- по цели управления;
- по типу управления;
- по принципу управления;
- по типу структурной организации системы управления.

Различают следующие типы управления:

- координирующий — согласование отдельных взаимосвязанных процессов;
- стабилизирующий — обеспечение соответствия управляемых координат объекта управления заданным координатам, изменяющимся заранее неизвестным образом;
- программно-регулируемый — обеспечение соответствия значений управляемых координат, которые изменяются по заранее известному закону;
- оптимальный — достижение оптимального значения показателя качества;
- экстремальный — достижение и поддержание экстремальных значений показателей качества, если их аналитическое выражение неизвестно, и в ходе работы имеет место дрейф экстремума;
- терминальный — перевод объекта из начального в конечное состояние за заданный интервал времени или к заданному моменту времени;
- финитный — перевод объекта управления из заданного начального в заданное конечное состояние за ограниченный интервал времени;
- противоаварийный — предотвращение возникновения аварийных ситуаций.

По принципу управления классификация такова:

- ручное управление — осуществляется при непосредственном участии человека в выработке и реализации управляющих воздействий;
- автоматическое управление — осуществляется без участия человека;
- автоматизированное управление — комбинация первого и второго типов;
- управление по отклонениям — управление с обратной связью, при котором управляющее воздействие вырабатывается по заданным и текущим значениям выходных координат;
- управление без обратной связи — управление, при котором управляющее воздействие рассчитывается без учета результатов предыдущего управляющего воздействия;

- управление по возмущению — управляющее воздействие вырабатывается по значениям измеряемого возмущения (задающее воздействие в этом случае называется корректирующим);
- автономное управление — управление каждой выходной переменной объекта осуществляется независимо от других выходных переменных; для этого необходимо на каждый выход иметь свое управляющее воздействие;
- адаптивное управление — управляющее воздействие вырабатывается при заранее неизвестных или изменяющихся в процессе эксплуатации свойствах объекта управления.

По характеру математических моделей различают линейные и нелинейные, стационарные и нестационарные, детерминированные и стохастические, с запаздыванием и без запаздывания, инерционные и неинерционные, распределенные и сосредоточенные объекты управления.

Сосредоточенные объекты — это объекты, в которых расстояние между элементами несущественно для организации управления. Другими словами, в таких объектах отсутствует зависимость каких-либо величин, описывающих состояние объекта, от пространственной координаты. Большинство объектов рассматривается как сосредоточенные. Если математическая модель сосредоточенного объекта представлена в виде системы дифференциальных уравнений, то они являются обыкновенными, т.е. не содержат частных производных по пространственным переменным.

По характеру протекания процесса функционирования различают дискретные, непрерывные, условно-непрерывные, дискретно-непрерывные объекты управления.

По характеру связи между входными и выходными координатами различают односвязные и многосвязные объекты управления. В односвязных каждая выходная координата зависит только от соответствующей ей входной координаты. В многосвязных одна или несколько выходных координат могут зависеть более чем от одной входной координаты.

По количеству входных и выходных координат различают одномерные и многомерные объекты управления. В многомерных объектах управления входные и выходные координаты — векторные величины.

При всей важности рассмотренных выше признаков классификации типов АИУС значительную роль играет еще один: место человека в системе управления.

Человек в системе управления может участвовать абсолютно во всех стадиях управленческого цикла. Рассмотрим основные моменты участия человека в процессе функционирования АИУС.

До 1980-х гг. были широко распространены АСУТП, работавшие в режиме советчика. В таких системах вычислительная машина предлагала человеку-оператору конкретный набор управляющих воздействий для реализации, который он мог принять или отвергнуть по своему усмотрению. За редким исключением подобные системы порочны в своей основе. Их слабым местом было отсутствие обратных связей в отслеживании оператором управляющих воздействий. Оператор всегда мог отнестись к своим промахам при выборе управляющего решения на счет ошибок АСУТП. Кроме того, большое влияние на качество управления оказывал так называемый человеческий фактор: если оператору приходится выполнять много настроек регуляторов за минуту, то к концу рабочей смены неизбежны ошибки, которые повлекут за собой ухудшение качества работы системы.

Исследования в области инженерной психологии показали, что если вычислительная техника работает непрерывно с определенной цикличностью, то включение человека в цикл управления должно быть асинхронно этому циклу. Если оператор периодически вводит исходные данные в систему, он не должен это делать в заданный промежуток времени или с жестко заданной периодичностью.

Более эффективными стали АСУТП, которые работали в режиме непосредственного цифрового управления. Эти системы, в отличие от упомянутых выше «советующих», использовали управляющие сигналы непосредственно для приведения в действие управляющих органов (исполнительных механизмов) системы. На вычислительные средства АСУ возлагались такие функции, как реализация различных законов регулирования, связей между отдельными контурами многосвязного регулирования, адаптивное управление. Оператор имел возможность изменять установки,

контролировать некоторые наиболее важные параметры, варьировать диапазоны допустимого изменения измеряемых параметров и т.д. Основной недостаток таких систем состоит в том, что при отказе АСУ оператор не всегда мог обеспечить нормальную работу системы в режиме ручного управления.

В настоящее время перед проектировщиками автоматизированных информационно-управляющих систем ставятся следующие основные задачи:

- стремиться к максимальной дружелюбности пользовательского интерфейса;
- вводить в АИУС элементы экспертных систем, дающие возможность не только выбирать из предъявленного списка то или иное управленческое решение, но и в любой момент получить разъяснение, как и на основании каких рассуждений эти решения были получены;
- включать в управляющую систему элементы обучающих игровых моментов с возможностью их функционирования в фоновом режиме (для осуществления тренинга человека-оператора).

Надежность систем, содержащих в качестве одного из звеньев человека-оператора, в значительной мере определяется безошибочностью его действий. По зарубежным данным, в военной авиации 75% летных происшествий происходит по вине личного состава и только 25% обусловлено отказами оборудования.

Повышение надежности действий человека должно основываться в первую очередь на знании закономерностей его ошибочных действий. Они возможны на различных уровнях его деятельности, начиная от простейшей ошибочной реакции на сигнал определенного вида и заканчивая неверным анализом сложной обстановки и принятием ошибочного решения.

Большое внимание при проектировании АИУС должно уделяться организации рабочего места человека-оператора. От того, как оно организовано, зависят психофизические свойства персонала: утомляемость, скорость реакции и, в конечном счете, качество управления процессом. Этими вопросами занимаются эргономика и инженерная психология. Эргономика — наука об улучшении условий труда путем оптимизации рабочего места, орудий и предме-

тов труда. Инженерная психология делает основной акцент на психологическую и социальную совместимость человека и техники. Задача эргономики и инженерной психологии при проектировании АИУС заключается в рациональной организации взаимодействия персонала с комплексом технических средств АИУС, приведении его параметров в соответствие с психическими и физиологическими параметрами человека.

Важной частью в составе рабочего места является мнемосхема, комплексно отражающая структурные, функциональные, временные, а при необходимости и пространственные особенности объекта управления. Мнемосхема помогает человеку в решении производственных задач, поддерживая психологический ритм производственной деятельности. По степени сложности исполнения выделяют следующие виды мнемосхем:

- слепые, фиксирующие структуру подсистемы и элементов управляемого объекта с отражением многообразия их связей между собой;
- с частичной индикацией светом или цветом информации о состоянии основного оборудования или основных материальных и энергетических потоков. Такие мнемосхемы отражают динамику объекта управления и дают возможность оценить состояние оборудования;
- с командно-квитирующими элементами, т.е. когда на мнемосхеме непосредственно присутствуют управляющие органы, позволяющие включать и выключать отдельные части оборудования, переключать направления потоков, принимать сложные управляющие решения. Чаще всего управление через мнемосхему дублирует управление с рабочего стола.

При управлении технологическим процессом важное место в рабочем помещении оператора занимают щиты, пульты, панели. На них располагаются вторичные приборы и органы дистанционного управления.

Практически все технологические процессы обладают инерционностью, поэтому часто не так важно значение сигнала, как динамика его изменения. В большинстве случаев динамику измене-

ния сигнала необходимо отображать на дисплее и на вторичных приборах.

Принципиальным является то, как сгруппированы вторичные приборы и исполнительные органы друг относительно друга. Их размещение должно способствовать повышению оперативности оценки состояния, максимальной эффективности в реализации управляющих решений, а также обеспечивать комфорт. Оператор всегда должен иметь возможность оценить состояние объекта даже после беглого взгляда на информационную панель.

К ключевым принципам компоновки оборудования АИУС относятся следующие:

- функциональный принцип;
- принцип ответственности — на самом видном и доступном месте концентрируется максимальное количество наиболее важных приборов;
- эргономичность размещения — компоновка осуществляется с позиции максимального удобства оператора;
- принцип последовательности использования;
- принцип частоты использования.

Данные принципы преследуют конкретные цели и в комплексе могут вступать в противоречие друг с другом, поэтому при компоновке надо применять то оптимальное решение, которое удовлетворяет всем принципам в максимально возможной степени.

Значительную роль играет разумно организованный диалог человека и ЭВМ. Язык общения определяется в рамках лингвистического обеспечения системы, которое должно соответствовать организационному и информационному обеспечению.

Лекция 11. Системный подход к проектированию АИУС

Как и всякое фундаментальное понятие, понятие системы конкретизируется в процессе рассмотрения основных свойств. Таких свойств у системы можно выделить три:

- система есть прежде всего совокупность элементов. При определенных условиях множество элементов может рассматриваться как система;
- совокупность элементов должна быть определенным образом организована. Куча камней, например, или скопление людей в автобусе — примеры неорганизованных совокупностей, элементы которых систему не образуют;
- между элементами системы должны быть существенные связи, благодаря которым данная совокупность обладает свойствами, изначально отсутствующими у всех ее отдельных элементов, что дает возможность выделить данную совокупность из окружающей среды в виде целостного объекта.

Таким образом, в самом общем случае понятие «система» характеризуется:

- наличием множества элементов;
- наличием устойчивых связей между ними;
- целостным характером рассматриваемого устройства, явления или процесса.

Все существующие в мире системы принято делить на два класса: материальные и абстрактные.

Абстрактные системы — это умозрительные представления образов или моделей материальных систем, которые подразделяются на описательные (иногда их называют семиотическими) и формальные (математические модели).

Материальные системы могут быть естественными и искусственными. Естественные системы подразделяются на физические, химические, планетарные и космические.

Искусственные системы можно классифицировать по многим признакам, главный из которых — роль человека в системе. По этому признаку можно выделить два основных класса искусственных систем: технические и организационно-экономические. В основе функционирования технических систем лежат процессы, совершаемые машинами, в основе функционирования организационно-экономических систем — процессы, совершаемые человеко-машинными комплексами.

По количеству образующих систему элементов принято различать малые, сложные, сверхсложные и суперсистемы. К малым относятся системы с числом элементов не более 10^3 . Примерами малых систем могут служить утюг, пылесос, другие бытовые приборы. Если число элементов более 10^3 , но менее 10^7 , систему относят к сложной (телевизор, компьютер, АТС, транспортная система города и т.д.). Сверхсложные системы имеют число элементов более 10^7 . Это, например, организм человека, социальные группы населения, звездные системы и т.д.

Одним из основных понятий в теории систем является понятие связи. Связь предметов между собой можно определить следующим образом: два или более различных предметов связаны, если по наличию или отсутствию некоторых свойств у одного из них мы можем судить о наличии или отсутствии тех или иных свойств у других (возникновение или исчезновение предметов можно рассматривать как частный случай). Например, температура и давление массы газа связаны так, что с увеличением температуры при постоянном объеме давление газа увеличивается.

Выявление связей в сложных системах позволяет познавать предметы или явления (в общем случае — элементы систем) не непосредственно, а косвенно, через другие предметы или явления, находящиеся с ними в той или иной связи.

Понятие связи относится к первичным, т.е. к философским категориям, поэтому точное, исчерпывающее определение ему дать невозможно. Однако, поскольку в теории систем связь является центральным понятием, необходимо ввести определенную классификацию типов связей, имеющихся в технических и организационно-экономических системах. Существует много вариантов классификации типов связей. Согласно одному из них различают связи следующих типов:

- связи взаимодействия, среди которых можно выделить связи между свойствами и связи между объектами;
- связи порождения (генетические связи) — связи типа «А есть отец В»;
- связи преобразования, приводящие к изменению свойств взаимодействующих объектов. При этом различают косвенное и

прямое взаимодействия. Прямое взаимодействие возникает в том случае, когда один объект непосредственно воздействует на другой (например, передача импульса при соударении одной молекулы с другой). Косвенное взаимодействие имеет место в том случае, когда объект способствует изменению свойств взаимодействующих элементов (такова функция химических катализаторов);

- структурные связи, определяющие пространственное расположение элементов в системе (например, расположение атомов в молекуле вещества);
- связи функционирования, обеспечивающие реальную жизнедеятельность системного объекта или его работу. Многообразие функций в объектах различного рода определяет и многообразие видов связей функционирования. Общим для всех этих видов является то, что объекты, объединяемые связью, совместно осуществляют определенную функцию. В самом общем виде связи функционирования можно подразделить на связи состояний (когда следующее по времени состояние является функцией от предыдущего) и связи энергетические (когда объекты связаны единством реализуемой функции).

Для того чтобы представить изучаемый объект как систему, необходимо так или иначе расчленить объект, выявить его пространственно ограниченные части или найти другие формы расчленения, а затем констатировать существование отношений этих частей в целостной картине объекта. Представляя объект как систему, мы даем предварительную картину составных частей объекта в их взаимных отношениях. Система часто определяется как некоторая совокупность отношений частей или элементов, и такое определение способствует более точному формулированию задачи исследования, позволяющему затем перейти к структурному анализу системы.

Структура — это устойчивая картина взаимных отношений элементов целостного объекта. Переход от системы целостных свойств к структуре может быть осуществлен при условии, что найдены элементы и их устойчивые отношения.

Сложные технические и организационно-экономические системы характеризуются большим количеством образующих их элементов. Описание структуры объекта становится затруднительным. В этом случае вводят частичное упорядочение по каким-либо признакам, что позволяет объединить отдельные совокупности элементов в группы, т.е. ввести в структуру системы определенную иерархию.

Системный подход к изучению сложных объектов подразумевает органическое сочетание двух методов. Первый — расчленение объектов на части и исследование связей и отношений между частями (системный анализ). Второй — рассмотрение целей и задач, общих для всех частей, и нахождение оптимального в каком-либо смысле решения (синтез системы).

Системный анализ — взаимосвязанное логико-математическое и комплексное рассмотрение всех вопросов, относящихся не только к замыслу, разработке, производству, эксплуатации и последующей ликвидации современных технических средств, но и к методам руководства этими этапами с учетом социальных, психологических, правовых, географических и других вопросов.

Большинство сложных объектов можно разделить на три группы:

- хорошо структурированные системы, в которых существенные зависимости известны настолько хорошо, что они могут быть выражены в виде строгой математической модели;
- неструктурированные системы, сведения о которых имеются лишь на качественном, описательном уровне, количественные характеристики элементов и связей неизвестны;
- слабо структурированные или смешанные системы, описание большинства элементов которых представлено на качественном уровне приближенно, однако имеются и количественные характеристики отдельных свойств и признаков.

При создании хорошо структурированных систем используются математические методы, совокупность которых составляет раздел математики, называемый исследованием операций. Сюда входит применение математических моделей и методов (линейного, нелинейного, динамического программирования, теории массово-

го обслуживания, теории игр и т.д.) для отыскания оптимальной стратегии управления целенаправленными действиями. Основная проблема применения методов исследования операций состоит в том, чтобы правильно подобрать типовую или разработать новую математическую модель и путем анализа результатов математических расчетов убедиться, что данная модель правильно отражает существо решаемой задачи.

В неструктурированных системах традиционным является эвристический метод, который состоит в том, что опытные специалисты (эксперты) собирают максимум сведений о решаемой проблеме и на основе имеющегося опыта принимают соответствующие решения.

Слабо структурированные системы связаны с решением важных экономических, технических, политических задач крупного масштаба. При осуществлении системного анализа в этих случаях на первых этапах не обязательна четкая и исчерпывающая постановка задачи. Четкость достигается в процессе самого анализа и рассматривается как одна из его главных целей.

Система управления — это совокупность управляемого объекта или процесса и устройства управления, к которому относится комплекс средств приема, сбора, передачи информации и формирования управляющих сигналов и команд. При этом действие системы управления направлено на улучшение и поддержание работы процесса или объекта. В некоторых случаях без АСУ вообще невозможно решение задачи в силу сложности процесса управления. Управляемый объект — это элемент системы, который для нормального функционирования нуждается в систематическом контроле и регулировании. Управляющий объект — элемент системы, который обеспечивает слежение за деятельностью управляемого объекта, выявляет возможные отклонения от заданной программы и обеспечивает своевременное приведение к его нормальному функционированию.

С точки зрения логики функционирования все системы управления решают три основные задачи:

- сбор информации об управляемом объекте;
- обработка информации;
- выдача управляющих воздействий в той или иной форме.

В зависимости от вида системы управление представляет собой воздействия на физическом или информационном уровне, направленные на поддержание или улучшение функционирования управляемого объекта в соответствии с имеющейся программой или целью управления.

Различают два типа систем управления:

- системы управления технологическими процессами в широком смысле этого слова, предназначенные для непосредственного управления производственными процессами на физическом уровне;
- системы информационного управления, предназначенные для решения задач управления такими объектами, как крупные технические подразделения, военные, строительные и иные объекты разных рангов.

Главное отличие между ними заключается в характере объекта управления. В первом случае это всевозможные установки, приборы, станки и прочее, во втором — прежде всего люди.

Другое отличие между указанными системами заключается в форме передачи информации. Если в системах управления технологическими процессами основной формой являются различного рода сигналы, то в системах организационного управления — документы. Четкую границу между двумя рассматриваемыми типами систем провести невозможно, обычно информация передается как с помощью документов, так и с помощью сигналов.

Автоматизация систем управления осуществляется с помощью вычислительной техники. В зависимости от степени участия человека в управлении системы можно классифицировать следующим образом:

- автоматические;
- полуавтоматические;
- автоматизированные.

Автоматизированная система не исключает, а, наоборот, предполагает участие человека в управлении объектом и принятие человеком ключевых решений, в то время как автоматическая система исключает участие человека в управлении объектом.

Полуавтоматическая система может рассматриваться как вариант автоматической, в которой уровень развития техники еще не позволяет исключить человека из контуров системы управления.

Центральным ядром системы управления, с помощью которого осуществляется ее автоматизация, является вычислительная машина. Возможны два способа взаимодействия между ЭВМ, объектом управления и органом управления.

В первом случае ЭВМ используется, как правило, для решения отдельных периодически повторяющихся трудоемких задач. Сбор информации ведется вручную, так же осуществляется и подготовка документов с управляющими воздействиями. Подобная система может быть названа системой обработки данных. Обращение пользователей к такой системе приводит к обновлению информации; вывод информации может вовсе отсутствовать или представлять собой результат программной обработки хранимых сведений, а не сами сведения. Примером системы обработки данных может служить система сберегательного банка города. Она содержит сведения о вкладах жителей города, большинство обработок банковской информации предполагает обновление сумм вкладов, расчет процентов, подведение итогов за некоторый период работы и т.п. Во втором случае основная информация о состоянии управляемого объекта собирается автоматически машиной (в общем случае — вычислительным центром). ЭВМ перерабатывает поступающую информацию и в том или ином виде готовит выходную документацию, после чего выносятся решение о воздействии на объект. Автоматизированные системы управления классифицируются также в зависимости от вида выдаваемой ими выходной документации. Последняя может быть представлена в виде:

- переработанной, упорядоченной совокупности сведений об управляемом объекте. На их основании человек (или группа людей) принимает решение о характере воздействия на объект. Это свойственно системе обработки данных, но не собственно автоматизированной системе;
- совокупности рекомендаций (вариантов решения) относительно характера воздействия на управляемый объект. Окончательное решение в данном случае принимает человек. Такая реали-

зация наиболее типична для автоматизированных систем управления.

Жизненный цикл любой системы, включая и АИУС, состоит из определенной последовательности этапов:

- предпроектное исследование;
- проектирование системы;
- создание системы;
- ввод системы в эксплуатацию;
- вывод системы на проектные мощности с целью достижения заданных показателей функционирования;
- эксплуатация системы — основной жизненный период;
- окончание работы системы.

Предпроектным исследованиям должен предшествовать этап целевого определения и обоснования необходимости создания системы. При создании любой системы необходимо учитывать все перечисленные выше этапы жизненного цикла, поскольку только комплексное рассмотрение технико-экономических показателей этих этапов предопределяет общую эффективность системы. Целью этого этапа является формализованное представление процесса управления производством.

Проект — это среднее звено между научно-технической идеей и ее реализацией, а проектирование — процесс перехода от первичного описания пока не существующего проектируемого объекта в виде проектного задания к описанию его в виде проектной документации, достаточной для создания этого объекта. В основе проектирования лежат творческая деятельность инженера-проектировщика и эвристические методы анализа и синтеза систем.

Этапы жизненного цикла с первого по четвертый относятся к стадиям создания системы, и от организации работ на них в существенной степени зависит эффективность системы. Рассматривая вопросы создания и эксплуатации АИУС, нужно учитывать следующие ее особенности:

- АИУС создается для управления некоторым объектом, поэтому ее жизненный цикл должен быть равен жизненному циклу объекта управления;

- в связи с ускоренными темпами развития вычислительной техники многие решения могут быстро морально устаревать, что накладывает жесткие ограничения на сроки создания АИУС;
- АИУС является человеко-машинной системой, т.е. человек здесь выступает как часть системы, поэтому для повышения эффективности большое внимание следует уделять проблеме взаимодействия человека с остальными частями системы;
- АИУС всегда надо рассматривать как многоцелевые системы. Это создает дополнительные сложности, так как целевые установки бывают противоречивыми. Кроме того, в отдельные моменты функционирования объекта управления целевые установки могут меняться. Например, в обычных условиях цех работает с максимальной производительностью при ограничении на качество продукции. При поступлении специального заказа может потребоваться максимально возможное качество продукции при ограничении на ее количество;
- АИУС немисливо рассматривать без объекта управления, а следовательно, специфика создания системы заключается в том, что задача повышения эффективности функционирования АИУС является не самоцелью, а лишь средством достижения максимальной производительности объекта управления.

В стадиях создания АИУС можно выделить и более мелкие структурные единицы. Так, предпроектное исследование заканчивается выдачей технического задания на проектирование системы. Иногда ограничиваются техническими требованиями к системе. От подготовки документации на этом этапе зависит не только будущее развитие системы, но и последующие взаимоотношения заказчика и разработчика.

Проектирование разделяют на две стадии: разработку технического проекта и подготовку технической документации. Для ускорения проектирования эти две части объединяют в технико-рабочий проект.

При проектировании важно предусмотреть все возможные проблемы, которые могут возникнуть на стадии ввода системы в эксплуатацию и вывода ее на проектную мощность. Для удачного прохождения промежуточных этапов в АИУС включают специ-

альные подсистемы, которые после окончательного внедрения оказываются ненужными. Например, при вводе системы в эксплуатацию можно использовать интерфейс с избыточным количеством подсказок.

Лекция 12. Моделирование рабочего потока

Современная организация производства базируется на понятии рабочего потока — последовательности действий и операций, в процессе выполнения которых исходные материалы преобразуются в выходной продукт — результат производства. Этот поток может разветвляться, при этом некоторые операции могут выполняться параллельно, независимо друг от друга. На каком-то этапе происходит слияние выполняемых операций в один рабочий поток. Выполняемые операции и действия должны быть согласованы по времени в соответствии со схемой рабочего потока. Эта схема разрабатывается так, чтобы процесс преобразования исходных материалов в конечный продукт был в определенном смысле оптимальным. Критерий оптимальности может быть разным в зависимости от условий производства или требований заказчика. Например, в качестве критерия оптимальности может быть максимально возможная прибыль при выполнении заказа. Для решения этой задачи должна быть разработана соответствующая схема рабочего потока, учитывающая возможности используемого оборудования. Эта схема, по всей вероятности, будет отличаться от схемы рабочего потока, при которой минимизируются сроки выполнения заказа, когда требование экономии средств отходит на второй план.

Следует отметить, что термин «оптимальный» для оценки качества управления рабочим потоком не вполне оправдан, поскольку оптимизация — это достижение наилучшего результата в строго определенных условиях, например при заданных скоростях выполнения отдельных операций. На практике при переходе от одного режима работы печатной машины (связанного с изменениями схемы рабочего потока) многие параметры изменяются, по-

этому предпочитают говорить не об оптимальном решении задачи управления, а о некотором рациональном способе управления, при котором достигается лучший результат из числа рассмотренных.

Существует несколько способов представления моделей [3]. Практически все способы функциональных спецификаций имеют общие черты:

- иерархическая структура представляется в виде диаграмм нескольких уровней;
- частью диаграммы каждого уровня является конструкция «вход — функция — выход»;
- необходимая информация содержится в файлах поясняющего текста.

В большинстве случаев функциональные диаграммы являются диаграммами потоков данных (DFD — Data Flow Diagram). В DFD блоки (прямоугольники) соответствуют функциям, дуги — входным и выходным потокам данных. Поясняющий текст дается в виде словарей данных, в которых указываются компонентный состав потоков данных, число повторений циклов и т.п. Для описания структуры информационных потоков можно использовать нотацию Бэкуса — Наура.

Разработка DFD начинается с построения диаграммы верхнего уровня, отражающей связи программной системы, представленной в виде единого процесса, с внешней средой. Декомпозиция процесса проводится до уровня элементарных процессов, которые могут быть представлены одностраничными описаниями алгоритмов (мини-спецификациями) на терминальном языке программирования.

Для описания информационных моделей наибольшее распространение получили диаграммы «сущность — связь» (ERD — Entity — Relations Diagrams), используемые, например в методике IDEF1X.

Поведенческие модели описывают процессы обработки информации. В системах CASE их представляют в виде граф-схем, диаграмм перехода состояний, таблиц решений, псевдокодов (языков спецификаций), языков программирования, в том числе языков четвертого поколения.

В графических схемах, как и в DFD, блоки используют для задания процессов обработки, но дуги имеют иной смысл — они описывают последовательность передач управления (вместе со специальными блоками управления).

В диаграммах перехода состояний узлы соответствуют состояниям моделируемой системы, дуги — переходам из состояния в состояние, атрибуты дуг — условиям перехода и инициируемым при их выполнении действиям. Как и в других конечно-автоматных моделях, кроме графической формы представления диаграмм перехода состояний можно использовать табличные формы. Так, при изоморфном представлении с помощью таблиц перехода состояний каждому переходу соответствует строка таблицы, в которой указываются исходное состояние, условие перехода, инициируемое при этом действие и новое состояние после перехода.

Близкий по характеру способ описания процессов основан на таблицах (или деревьях) решений. Каждый столбец таблицы соответствует определенному сочетанию условий, при выполнении которых осуществляются действия, указанные в нижерасположенных клетках столбца.

В псевдокодах алгоритмы записываются с помощью средств языка программирования (преимущественно для управляющих операторов) и естественного языка (для выражения содержания вычислительных блоков). Используются конструкции (операторы) следования, условные, цикла.

Языки четвертого поколения направлены на описание программ как совокупностей заранее разработанных программных модулей. Поэтому одна команда языка 4GL может соответствовать значительному фрагменту программы на языке 3GL. Примерами языков 4GL могут служить Informix-4GL, JAM, NewEra.

Мини-спецификации процессов могут быть выражены с помощью псевдокодов (языков спецификаций), визуальных языков проектирования или языков программирования.

Примерами широко известных инструментальных сред RAD являются VB (Visual Basic), Delphi, PowerBuilder фирм Microsoft, Borland, PowerSoft соответственно. Применение инструментальных

сред существенно сокращает объем ручной работы программистов, особенно при разработке интерфейсных частей программы.

В средах быстрой разработки приложений обычно реализуется способ программирования, называемый управлением событиями. При этом достигается автоматическое создание каркасов программ, существенно сокращается объем ручного кодирования, особенно при разработке интерфейсных частей программ. В этих средах пользователь может работать одновременно с несколькими экранами (окнами). Типичными являются окна из следующего списка:

- окно меню с пунктами «file», «edit», «windows» и т.п., реализующими функции, очевидные из названия пунктов;
- окно формы, в котором и создается прототип экрана будущей прикладной программы;
- палитра инструментов — набор изображений объектов пользовательского интерфейса, из которых можно компоновать окно формы;
- окно свойств и событий, с помощью которого ставятся в соответствие друг другу объекты окна формы, события и обработчики событий. Событием в прикладной программе является нажатие клавиши или установка курсора мыши на объект формы. Каждому событию должна соответствовать событийная процедура (обработчик события), которая проверяет код клавиши и вызывает нужную реакцию;
- окно редактора кода, в котором пользователь записывает создаваемую вручную часть кода;
- окно проекта — список модулей и форм в создаваемой программе.

Для написания событийных процедур в Visual Basic используется язык и текстовый редактор языка Basic, в Delphi — язык и редактор языка Object Pascal. В CASE-системе фирмы IBM, включающей части VisualAge (для клиентских приложений) и VisualGen (для серверных приложений), базовым языком выбран SmallTalk. В среде разработки приложений «клиент-сервер» SQLWindows оригинальные фрагменты программ пишутся на спе-

циальном языке SAL. Для реализации вычислительных процедур, в частности для написания мини-спецификаций, используется обычная технология программирования.

После написания программы на базовом языке компилятор системы переводит ее на промежуточный *p*-код. Вместе с интерпретатором кода эта программа рассматривается как исполнимый файл. В некоторых развитых средах компилируется EXE-файл, не требующий интерпретации для своего исполнения.

Помимо упрощения написания пользовательского интерфейса в средах RAD предусматриваются средства для реализации и других функций. Так, в наиболее развитой версии Visual Basic к ним относятся средства выполнения следующих функций:

- поддержка ODBC (Open DataBase Connectivity), что дает возможность работать с различными СУБД;
- разработка баз данных;
- разработка трехзвенных систем распределенных вычислений;
- интерактивная отладка процедур на SQL Server;
- управление версиями при групповой разработке программного обеспечения;
- моделирование и анализ сценариев распределенных вычислений.

Лекция 13. Методики моделирования IDEF

Взаимосвязанная совокупность методик концептуального проектирования IDEF (Integrated Definition) разработана по программе Integrated Computer-Aided Manufacturing в США. В этой совокупности имеются методики функционального, информационного и поведенческого моделирования и проектирования, в ее состав в настоящее время входят IDEF-методики, отмеченные в табл. 2.1.

Таблица 2.1

Методики моделирования IDEF

Название	Назначение
-----------------	-------------------

IDEF0	Функциональное моделирование
IDEF1 и IDEF1X	Информационное моделирование
IDEF2	Поведенческое моделирование
IDEF3	Моделирование процессов
IDEF4	Объектно-ориентированное проектирование
IDEF5	Систематизации объектов приложения
IDEF6	Использование рационального опыта проектирования
IDEF8	Взаимодействие человека и системы
IDEF9	Учет условий и ограничений
IDEF14	Моделирование вычислительных сетей

Поведенческое моделирование используется для определения динамики функционирования сложных систем. В его основе лежат модели и методы имитационного моделирования систем массового обслуживания, сети Петри, возможно применение конечно-автоматных моделей, описывающих поведение системы как последовательности смены состояний.

Поведенческие аспекты приложений отражает методика IDEF3 [6]. Если методика IDEF0 связана с функциональными аспектами и позволяет ответить на вопрос «Что делает система?», то в IDEF3 детализируются и конкретизируются IDEF0-функции, IDEF3-модель отвечает на вопрос «Как система это делает?». В IDEF3 входят два типа описаний:

- процесс-ориентированные в виде последовательности операций;
- объектно-ориентированные, представляемые диаграммами перехода состояний, характерными для конечно-автоматных моделей. В этих диаграммах имеются средства для изображения состояний системы, активности, переходов из состояния в состояние и условий перехода.

Системы информационного моделирования реализуют методики инфологического проектирования баз данных. Широко используются язык и методика IDEF1X создания информационных моделей приложений, развивающая более раннюю методику IDEF1. Кроме того, развитые коммерческие СУБД, как правило, имеют в своем составе совокупность CASE-средств проектирования приложений.

В IDEF1X имеется ясный графический язык для описания объектов и отношений в приложениях. Это язык диаграмм «сущность-связь». Разработка информационной модели по IDEF1X включает несколько стадий.

Стадия 0. Выяснение целей проекта, составление плана сбора информации. Обычно исходные положения для информационной модели следуют из IDEF0-модели.

Стадия 1. Выявление и определение сущностей. Это неформальная процедура.

Стадия 2. Выявление и определение основных отношений. Результат представляется графически в виде ER-диаграмм или в виде матрицы отношений, элемент которой $A_{ij} = 1$, если имеется связь между сущностями i и j , иначе $A_{ij} = 0$.

Стадия 3. Детализация неспецифических отношений, определение ключевых атрибутов, установление внешних ключей. Детализация неспецифических отношений заключается в замене связей «многие ко многим» на связи «многие к одному» и «один ко многим» введением сущности-посредника.

Стадия 4. Определение атрибутов и их принадлежности сущностям.

Методика IDEF4 реализует объектно-ориентированное проектирование больших систем. Она предоставляет пользователю графический язык для изображения классов, диаграмм наследования, таксономии методов.

Методика IDEF5 направлена на представление онтологической информации приложения в удобном для пользователя виде. Для этого используются символические обозначения (дескрипторы) объектов, их ассоциаций, ситуаций и схемный язык описания отношений классификации, «часть — целое», перехода и т.п. В методике имеются правила связывания объектов (термов) в предложения и аксиомы интерпретации термов.

Развитие BPR-методик продолжается в США по программе IISE (Information Integration for Concurrent Engineering). Разработаны следующие методики:

- IDEF6 — направлена на сохранение рационального опыта проектирования информационных систем, что способствует предотвращению повторных ошибок;
- IDEF8 — для проектирования диалога человека с технической системой;
- IDEF9 — для анализа имеющихся условий и ограничений (в том числе физических, юридических, политических) и их влияния на принимаемые решения в процессе реинжиниринга;
- IDEF14 — для представления и анализа данных при проектировании вычислительных сетей на графическом языке с описанием конфигураций, очередей, сетевых компонентов, требований к надежности и т.п.

Технология CALS призвана разрешить проблему согласования содержания и формы представления данных о промышленной продукции в территориально распределенной сети проектных и производственных узлов на основе совокупности международных стандартов. Только в этих условиях станут возможными оптимальная специализация предприятий, распределенное проектирование, минимизация затрат на освоение и эксплуатацию созданных систем.

Лекция 14. Методология функционального моделирования

Методологию SADT разработал Д. Росс. На ее основе создана, в частности, известная методология IDEF0 (Icam DEFinition), которая является основной частью программы ICAM (Интеграция компьютерных и промышленных технологий).

Методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Методология SADT может использоваться для моделирования широкого круга систем и определения требований и функций, а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции. Результатом применения данной методологии является модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга.

Диаграммы — главные компоненты модели, все функции ИС и интерфейсы на них представлены как блоки и дуги. Место соединения дуги с блоком определяет тип интерфейса. Управляющая (регламентирующая) информация входит в блок сверху, в то время как информация, которая подвергается обработке, показана с левой стороны блока, а результаты выхода — с правой стороны. Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу (рис. 2.1).

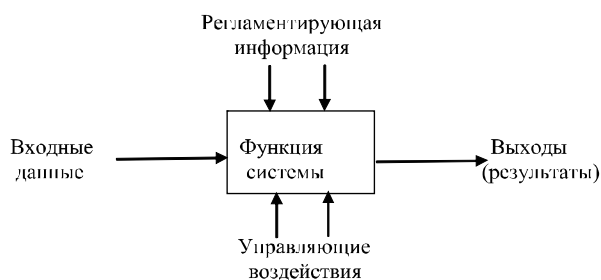


Рис. 2.1. Функциональный блок и направленные дуги

Построение модели начинается с представления всей системы в виде простейшего компонента — одного блока и дуг, изображающих связи с функциями вне системы. Поскольку единственный блок представляет всю систему как единое целое, имя, указанное в блоке, является общим. Это верно и для направленных дуг — они также представляют полный набор внешних связей системы в целом.

Затем блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных направленными дугами. Эти блоки пред-

ставляют основные подфункции исходной функции. Данная декомпозиция выявляет полный набор подфункций, каждая из которых представлена как блок, границы которого определены связями. Каждая из этих подфункций может быть декомпозирована подобным образом для более детального представления.

Во всех случаях каждая подфункция может содержать только те элементы, которые входят в исходную функцию. Кроме того, модель не может опустить какие-либо элементы, т.е. родительский блок и его связи обеспечивают контекст. К нему нельзя ничего добавить, и из него не может быть ничего удалено.

Модель SADT — это серия диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части, которые представлены в виде блоков. Детали каждого из основных блоков показаны в виде блоков на других диаграммах. Каждая детальная диаграмма является декомпозицией блока из более общей диаграммы. На каждом шаге декомпозиции более общая диаграмма называется родительской по отношению к более детальной.

Дуги, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются точно теми же, что и дуги, входящие в диаграмму нижнего уровня и выходящие из нее, потому что блок и диаграмма представляют одну и ту же часть системы.

Некоторые дуги присоединены к блокам диаграммы обоими концами, у других один конец остается не присоединенным. Не присоединенные дуги соответствуют входам, управлениям и выходам родительского блока. Источник или получатель этих пограничных дуг может быть обнаружен только на родительской диаграмме. Не присоединенные концы должны соответствовать дугам на исходной диаграмме. Все граничные дуги должны продолжаться на родительской диаграмме, чтобы она была полной и непротиворечивой.

На SADT-диаграммах явно не указаны ни последовательность, ни время. Обратные связи, итерации, продолжающиеся процессы и перекрывающиеся (по времени) функции могут быть изображены с помощью дуг. Обратные связи выступают в виде комментариев, замечаний, исправлений и т.д. (рис. 2.2).

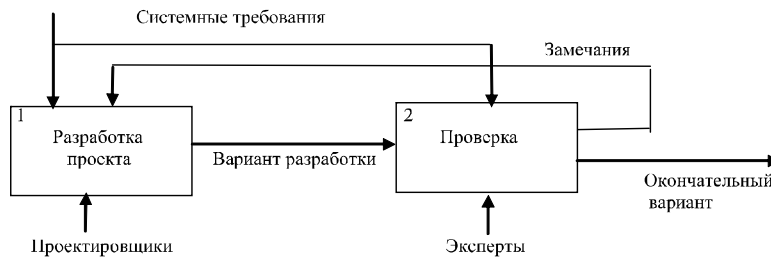


Рис. 2.2. Пример фрагмента с обратной связью

Как было отмечено, механизмы (дуги с нижней стороны) показывают средства, с помощью которых осуществляется выполнение функций. Механизм может быть человеком, компьютером или любым другим устройством, которое помогает выполнять данную функцию (рис. 2.3).

Блоки на диаграмме пронумерованы. Любой блок может быть затем описан диаграммой нижнего уровня, которая, в свою очередь, может быть детализирована с помощью необходимого числа диаграмм. Таким образом формируется иерархия диаграмм.

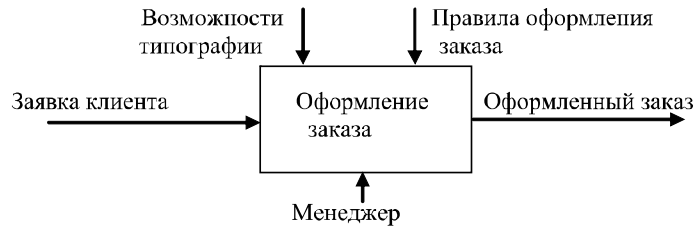


Рис. 2.3. Пример фрагмента модели

Одним из важных моментов при проектировании ИС с помощью методологии SADT является точная согласованность типов связей между функциями. Различают по крайней мере семь типов связности: случайная, логическая, временная, процедурная, коммуникационная, последовательная, функциональная.

Лекция 15. Многоуровневая система управления производством

Как правило, любая автоматизируемая производственная система имеет сложную многоуровневую, иерархическую структуру, которая преобразует исходные полуфабрикаты сырья или материалов в конечный продукт, отвечающий потребительским требованиям рынка. Основой любого производства является технологический процесс — определенное взаимодействие орудий и предметов труда, обслуживающей и транспортной систем, в результате чего выпускается продукция, отвечающая заданному критерию качества. Перемещение предметов труда от одной стадии обработки к другой можно характеризовать как рабочий материальный поток в производственном пространстве. По характеру материального потока технологические процессы делят на два типа: непрерывные и дискретные.

В непрерывных технологических процессах материальный поток, проходящий через технологическое оборудование, является неразрывным. Для дискретных технологических процессов характерна выходная продукция в виде изделий, исчисляемых в штуках. Исходные компоненты преобразуются циклически, и готовая продукция выпускается партиями. Каждый цикл получения выходного продукта или партии включает в себя время, затрачиваемое на приведение технологического оборудования в исходное состояние, которое является невозвратимой утратой. Соответственно используемым технологическим процессам производство принято делить на три основных типа: непрерывное, дискретное и непрерывно-дискретное. К непрерывным производствам можно отнести большинство процессов химической, нефтеперерабатывающей, энергетической промышленности. Продукция таких производств — вещество или энергия, а параметры технологических процессов — непрерывные величины: давление, концентрация и др. К дискретным производствам относятся предприятия машиностроения, приборостроения и т.п. Материальный поток здесь дискретен, а производственный процесс характеризуется дискретными параметрами:

числом изготавливаемых изделий, наличием заготовок на складах и операциях, числом бракованных изделий. Операции обработки также дискретны, так как имеют начало и конец. Информационный поток, отражающий ход дискретного производства, также носит дискретный характер. К непрерывно-дискретным производствам, сочетающим особенности первых двух типов, относятся производства металлургической, пищевой, полиграфической, цементной промышленности и др.

На первом (нижнем) уровне комплексной системы автоматизации функционируют контуры управления отдельными технологическими агрегатами, работу которых контролируют операторы с помощью пультов управления.

На втором (среднем) уровне управления в соответствии с плановым заданием и приоритетом производства продукции распределяется работа между технологическими агрегатами первого уровня в зависимости от их исправности, обеспеченности материалами и полуфабрикатами, производится оперативное планирование и управление работой первого уровня и составляется отчет об исполнении работ. Работа этого уровня оценивается по результатам выполнения планового задания.

Третий уровень (верхний) управления производственной системой охватывает координацию работы отдельных участков различных видов производства и реализацию плановых заданий в целом, а также работы по подготовке производства, включая проектирование самих изделий, технологических процессов, технологической оснастки и управляющих программ для локальных объектов управления. На этом уровне управления действуют экономические критерии качества. Далее, на этом уровне анализируются связи с поставщиками, конъюнктура рынка, прогнозируется выпускаемая в будущем номенклатура изделий.

На всех уровнях управления соответствующие задачи решаются как с помощью технических средств, так и с использованием интеллектуальных и физических возможностей производственного персонала. Одним из важных показателей качества системы является ее производительность. Объем и качество производимой продукции определяется всеми уровнями управления, но производит-

ся она непосредственно на технологическом уровне в результате взаимодействия трех компонентов: рабочей среды, инструмента и человека, который либо непосредственно выполняет работу, либо управляет инструментом, иногда сложным.

Существуют две основные тенденции: стремление к организации рабочей среды или адаптации к среде при управлении оборудованием. Примером организации соответствующей среды и достижения при этом высокой производительности являются автоматические роторные и роторно-конвейерные линии, создание которых относится к одному из основных направлений ускорения технического прогресса. Альтернативой организации среды является адаптация к ней на всех уровнях управления. Однако оперативно решить вопросы адаптации могут лишь специально организованные высокопроизводительные системы, обладающие средствами автоматизированного получения, обработки и передачи информации. Только использование развитой информационной и вычислительной техники, успешно работающей в условиях производства, позволяет обеспечить адаптационные возможности в целом и технологических машин в частности. В таких системах обслуживающий персонал освобождается от переработки формализуемой информации и выступает как субъект, принимающий основные решения, которые часто не поддаются формализации.

Применение специальных сложных методов обработки данных и управления в АСУТП стало одним из основных направлений в автоматизации промышленного производства. Методы повышенной сложности с определенной долей условности принято называть современными. В зарубежной практике создания систем управления в промышленности соответствующее направление автоматизации получило наименование «Advanced control». Это понятие носит неформальный характер, однако в него вкладывается конкретный научно-технический и коммерческий смысл: специальные алгоритмы повышенной сложности по сравнению с традиционными алгоритмами первичной переработки и представления информации. Методы управления повышенной сложности создаются с приложением научных сил и представляют собой наукоемкую продукцию. К ним относят, например, алгоритмы адаптивного

и многосвязного регулирования, контроля и управления с применением моделей объекта, оптимального управления, специальные логико-динамические алгоритмы и т.д., в них находят применение методы искусственного интеллекта. Использование методов повышенной сложности для контроля и управления процессом является неперенным признаком современной АСУТП, обеспечивает высокий уровень автоматизации производства и представляет собой неперенный атрибут понятия «комплексная автоматизация». Успешное промышленное применение современных методов управления определяет собой весомую долю технологических и экономических эффектов автоматизации.

Самостоятельную роль в современных методах управления играют технологии искусственного интеллекта, которые включают в свой состав искусственные нейронные сети, экспертные системы, нечеткую логику, генетические алгоритмы и др. Они обычно встраиваются в схемы алгоритмов управления и применяются при дефиците формализованных моделей объекта или алгоритмов решения стоящих проблем. Подчеркнем, что значительная часть применяемых методов управления известна давно. Новое качество в применении современных методов управления в промышленности появилось в последние два десятилетия. Необходимая база для развития и тиражного применения современных методов в промышленности создана прогрессом в области программно-технических средств автоматизации. Мощное программное обеспечение, высокий уровень сервиса и свободные вычислительные ресурсы, поддерживаемые информационными технологиями, создают благоприятную среду для применения достаточно сложных решений.

Методы косвенного управления используются для контроля неизмеряемых параметров и характеристик автоматизируемого процесса. Они состоят в вычислении соответствующих оценок состояния объекта по уравнениям модели при измеренных значениях опорных переменных. Эти методы получили достаточно широкое распространение в различных сферах, начиная от простых вычислений с использованием технологических зависимостей и заканчивая решением достаточно сложных систем уравнений. Например, в

контурах контроля и управления часто осуществляется коррекция измерений расхода газов в зависимости от их температуры и давления среды, материальных и тепловых балансов, проводится уточнение тепловой нагрузки оборудования по температурным перепадам и расходам теплоносителей и т.д. Особенно важную роль играет косвенный контроль ключевых в управлении технологией характеристик качества продукции, обычно измеряемых трудно и с большим запаздыванием. Системы косвенного контроля позволяют представить информацию операторам в содержательной форме, подготовленной к применению для анализа состояния процесса. Они позволяют «вскрыть изнутри» параметры состава, состояние потоков сырья, реагентов, теплоносителей, подаваемых в технологический аппарат; параметры и характеристики процессов в его рабочем пространстве; результаты процесса — данные о количестве и качестве продуктов. В алгоритме в качестве модели может фигурировать либо отдельная технологическая зависимость, описывающая конкретный эффект, либо система уравнений той или иной природы — алгебраических, дифференциальных, логических и т.д.; природа располагаемой модели зависит от источника ее получения: аналитический вывод, активный эксперимент, статистика и т.д.

Адаптивные методы управления предназначены для сохранения необходимого качества регулирования при изменениях характеристик объекта и условий его функционирования за счет адекватного перестроения алгоритма. Существует много подходов к построению адаптивных регуляторов, каждый из которых предпочтительно применять в определенных условиях. Адаптивный регулятор содержит в своем составе надстройку над обычным регулятором — блок адаптации, — который в зависимости от изменения условий работы и характеристик основного замкнутого контура воздействует на структуру и/или параметры регулятора. В адаптивных регуляторах обратная связь по характеристикам поведения образуется с использованием входных и выходных сигналов объекта. Большой выбор вариантов порождает огромное множество возможных алгоритмических схем. От алгоритмов требуются хорошее быстродействие, робастность (неприхотливость) и малое

потребление вычислительных ресурсов. При реализации и практическом использовании адаптивных регуляторов возникают объективные трудности из-за принципиальных особенностей их структур. Блоки параметрического воздействия в составе регуляторов имеют принципиально нелинейный характер, что приводит к затруднениям в теоретических исследованиях систем и к парадоксальным явлениям в их поведении. При этом часто обостряется вопрос об устойчивости адаптивной системы, которая в конкретных условиях не может быть гарантирована при некоторых сочетаниях параметров основного контура и блока адаптации. В то же время очень велика ответственность в решении доверить автоматом работу замкнутого контура на ответственном объекте управления.

Методы оптимального управления обычно используются для автоматизации непрерывных технологических процессов. При этом оптимизация статического режима обеспечивается вычислением координат оптимального режима и реализацией режима путем воздействия на задания локальных регуляторов и на регулирующие органы. Координаты оптимального режима — значения управляемых (регулируемых) переменных процесса — получаются в результате решения задачи математического программирования на уравнениях модели процесса при технологических, плановых и ресурсных ограничениях. Функция цели имеет характер технико-экономического показателя (прибыль, минимум затрат при заданной производительности и т.д.). Часто применяются линейные модели и постановки линейного или квадратичного программирования. Оптимальное управление требует для своего применения хорошего знания процесса и адекватной модели. В цифровых системах автоматического регулирования точность системы определяют аналого-цифровые и цифро-аналоговые преобразователи. Чем выше число двоичных разрядов, приходящихся на аналоговый сигнал в его диапазоне, тем выше разрешение и тем выше точность.

Анализ перспектив и тенденций развития технических средств полиграфического производства, средств электроники, автоматики и вычислительной техники показывает, что важнейшим направлением совершенствования полиграфического производства является

ся создание интегрированных автоматизированных производств. Интеграция производства в полиграфии идет по нескольким направлениям. Так, вывод информации из лазерных наборных автоматов непосредственно на полиграфическую форму устраняет из технологической цепи большой комплекс формного оборудования, в котором применяются энергоемкие и экологически вредные процессы. Еще больший эффект достигается в том случае, когда текстовая и иллюстрационная информация выводится на формный цилиндр непосредственно в печатную машину. Такие машины получили наименование цифровых печатных машин. При этом существенно сокращается время на подготовку печатной машины к работе.

Лекция 16. Современные системы управления производством

Конкуренция на полиграфическом рынке возрастает с каждым годом, поэтому актуальной проблемой является повышение эффективности работы предприятий за счет внедрения систем управления производством.

Основными задачами создания и внедрения подобных систем являются: оформление и расчет полиграфического заказа, подготовка производственной документации, взаиморасчеты с заказчиками, подготовка расчетных и отгрузочных документов, планирование и диспетчеризация производства, производственный учет, расчет плановой и фактической себестоимости, планирование расхода материалов, резервирование материалов, учет бумаги и материалов, учет готовой продукции, оперативная производственная отчетность, контроль прохождения заказа в режиме on-line.

Рассмотрим основные характеристики систем управления, используемых в настоящее время в полиграфии.

АСУ «Адьютант» позволяет осуществлять прием заказов в полном соответствии с выбранным технологическим процессом, повысить достоверность информации о заказе в связи с формализацией процесса принятия и прохождения заказа, осуществлять полную диспетчеризацию работ на основании выбранного техно-

логического процесса, оптимально вести позаказный учет всей продукции, иметь единый план производства.

Главные функции, выполнение которых обеспечивает «Адьютант»: автоматизация единого документооборота типографии, оптимизация планирования производства, контроль производственных процессов в режиме реального времени, автоматизация складской деятельности, автоматизация учетной деятельности предприятия.

Система поддерживает все виды допечатного, печатного и послепечатного полиграфического оборудования. Основой системы является концепция развивающегося программного комплекса, который может интегрироваться с существующими на предприятиях средствами автоматизации управления. «Адьютант» работает с документами, формы которых приняты у заказчика. Формы разделяются на группы по признакам принадлежности к функциональным модулям системы. Экранные формы позволяют отображать и выполнять работы с входящими и исходящими документами, а также документами внутреннего пользования. Модификация и расширение модулей программного комплекса возможны на любом этапе внедрения системы.

АСУ «АРМЕКС» предназначена для организаций, сферой деятельности которых является производство и продажа полиграфической продукции. Программа разработана с учетом требований полиграфических предприятий и имеет сертификат «1С: Совместимо», выданный фирмой «1С».

В программе реализован учет на всех основных участках, характерных для данного вида производства: от поступления и расчета заказа до отгрузки и оплаты продукции покупателем. В программе возможна настройка своих типов интерфейсов, прав доступа и меню.

АСУ «Аплер» — автоматизированная система оперативного управления и анализа, применяемая на малых и средних предприятиях полиграфии. Поддерживает сквозную автоматизацию работы всех подразделений и служб типографии. Пользователи системы имеют установленные роли, например, «Менеджер отдела продаж», «Технолог», «Мастер цеха» и т.д., что позволяет управ-

лять доступом к информации. В небольших типографиях один пользователь может совмещать несколько ролей, в крупных типографиях несколько пользователей могут выполнять одну роль, работая параллельно.

АСУ «ЛИМ–Корпорация» — это интегрированная информационная система управления предприятием, созданная на базе издательско-полиграфического объединения «Лев Толстой» фирмой ЗАО «ЛИМ». Потенциальные пользователи системы — крупные и средние полиграфические предприятия.

Функции системы достаточно широки. Она позволяет оформлять заказ, проводить расчет предварительной и фактической себестоимости заказов, расхода необходимых материалов (особо следует отметить наличие средства для определения раскладки деталей на листе), а также формировать укрупненную технологическую цепочку для предварительного расчета требуемых ресурсов. Кроме того, система осуществляет контроль прохождения заказа в производстве, планирование и диспетчеризацию, учет бумаги и других материалов, учет готовой продукции, формирование соответствующих бухгалтерских проводок.

АСУ «Печатный цех» представляет собой систему бухгалтерского, оперативного учета и расчета заработной платы для предприятий полиграфического производства, разработанную специалистами ФГУП «Типография № 12» им. М.И. Лоханкова. Продукт является конфигурацией «1С: Предприятие 7.7», поэтому выполняет в основном учетные функции.

Основные функции программного комплекса «Печатный цех»: расчет себестоимости заказа, учет движения товарно-материальных ценностей (ТМЦ), бухгалтерские функции, контроль всех стадий производства продукции, анализ плановых и фактических показателей по производству, формирование различных отчетов.

АСУ «Типография» компании *Profound Solutions* позиционируется как решение для малых и средних предприятий, работающих в области полиграфического производства. Ядро системы разработано с помощью классических инструментальных средств программирования, а дальнейшие настройка и адаптация к требовани-

ям конкретного предприятия происходят уже на самом предприятии. Данный программный комплекс, представляющий собой набор модулей, объединенных общей оболочкой, обеспечивает выполнение типичных задач автоматизации учета и планирования в полиграфическом производстве.

Пока эта система не обладает достаточными функциональными возможностями в области составления бухгалтерской отчетности, однако разработчики однозначно заявляют о производственной направленности своего программного продукта.

«1С: Предприятие 8.0. Полиграфия» — это разработка фирм 1С и «АРМЕКС» на базе «1С:Предприятие 8.0. Управление производственным предприятием».

Система предназначена для автоматизации предприятий, сферой деятельности которых является производство и продажа различной полиграфической продукции (листовок, приглашений, билетов, папок, буклетов, брошюр, книг и др.

Продукт является комплексным решением, охватывающим основные контуры управления и учета, которое позволяет организовать единую информационную систему для управления различными аспектами деятельности полиграфического предприятия.

АСУ АSystem — комплекс программных средств, предназначенный для управления рабочими процессами полиграфического предприятия. Данная система реализована по технологии «клиент-сервер», что обеспечивает стабильно высокую производительность независимо от количества одновременно работающих пользователей.

Сбор первичной информации может осуществляться различными способами. В систему заложена возможность интеграции с оборудованием на базе JDF, но в коммерческом секторе достаточно велик парк оборудования, не поддерживающего аппаратную регистрацию. В этом случае зафиксировать временные и технологические параметры выполнения операций можно вручную — с компьютера непосредственно в цехе или на основании рапортчиков на рабочем месте учетчика или мастера цеха. Если регистрировать события раз в два-три часа, можно получить картину деятельности предприятия в реальном масштабе времени.

В системе возможно компонентное и территориальное масштабирование. Нормирование операций происходит по следующему сценарию: регистрация заявки, расчет стоимости заказа, подготовка документации для передачи в производство. Следующий этап — отработка механизмов управления загрузкой оборудования. Одновременно система поддерживает возможность работы с удаленными участками производства.

АСУ DISO позволяет полностью контролировать процесс прохождения заказов на полиграфическом предприятии — от приема заказа и составления калькуляции вплоть до отгрузки готовой продукции. Наибольший интерес представляет программный модуль, позволяющий рассчитать плановую себестоимость заказа и поэтапно сравнить его с реальной себестоимостью выполненного заказа.

Система предназначена для решения административных задач с учетом специфики полиграфического производства. Имеются в виду такие важнейшие функции, как прием и калькуляция заказа, создание производственного технического задания, планирование производства, расчет плановой себестоимости заказа, контроль его прохождения в производстве и в конечном итоге получение информации о фактических затратах по каждому заказу. Объединение программных продуктов PECOM и DISO осуществлено с помощью протокола JDF.

АСУ HIFLEX предназначена для типографий с распределенной структурой производства и сбором информации посредством формата JDF. Она позволяет управлять работами в пределах крупной типографии.

Система основана на базе данных SQL, написана на языках программирования C+ и C++ и гарантирует платформенную независимость, совместима со всеми версиями ОС Windows и Macintosh.

АСУ Prinect предназначена для управления непрерывным рабочим потоком для полиграфического производства. Она объединяет данные всех стадий производства и интегрирует их с системой управления данными рабочего потока. Система представляет собой программно-аппаратный комплекс, обеспечивающий опти-

мальное взаимодействие между отдельными компонентами производственного процесса. Основной принцип рабочего потока — доступность управляющих данных в любой момент и на каждом этапе производства.

Prinect базируется на модульной структуре и характеризуется открытой архитектурой, что позволяет использовать как уже установленное оборудование и программное обеспечение, так и расширения, объединяя их в цифровую систему. В управлении рабочим потоком система использует стандарт JDF.

Благодаря открытому интерфейсу с другими программными продуктами система позволяет обеспечить интеграцию производственных процессов. Она поставляет данные печатных заданий производственным программам и сама получает информацию из производства, которая необходима для выполнения позаказной калькуляции и статистических расчетов. В результате реализуется возможность анализа и выбора наиболее перспективного заказа.

Prinect обеспечивает не только автоматическую интеграцию всех звеньев в производственной цепочке, но и их связь с рабочими потоками на всех этапах процесса управления. Интеграция процесса сокращает или вообще устраняет необходимость вмешательства оператора, обеспечивая тем самым значительно большую рентабельность и производительность. Prinect объединяет множество отдельных процессов — от калькуляции себестоимости работы, планирования производства, допечатной подготовки, печати и послепечатной обработки до поставки готовой продукции и выставления счетов. Это цифровое связующее звено для компонентов системы обмена данными.

ACU PrintEffect — автоматизированная система управления печатным салоном или типографией оперативной полиграфии, предприятием широкоформатной печати или производством сувенирной продукции, дизайн-бюро.

Предназначена для оперативного учета и управления предприятием. PrintEffect — программный продукт, который может быть установлен и настроен пользователем без специальной подготовки. Система работает в режиме «клиент-сервер» и поддерживает до 10 рабочих мест.

В системе заложены типовые бизнес-процессы, описывающие технологические операции, существующие на небольших полиграфических предприятиях. Благодаря возможностям системы обеспечивается автоматизация всех подразделений: продаж, маркетинга, предпечатной подготовки, печати, финишной обработки, доставки готовой продукции, учета материалов и готовой продукции.

Библиографический список

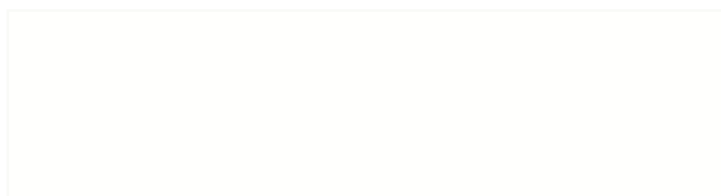
1. *Агеев В.Н.* Информационное обеспечение издательско-полиграфических систем / В.Н. Агеев. — М. : МГУП, 2007. — 316 с.
2. *Косоруков О.А.* Исследование операций : учебник для вузов / О.А. Косоруков, А.В. Мищенко. — М. : Экзамен, 2003. — 448 с.
3. *Агеев В.Н.* Интегрированные системы проектирования и управления : учеб. пособие / В.Н. Агеев, П.К. Иванов, В.В. Ковалева. — М. : МГУП, 2008. — 248 с.
4. *Иванова Т.М.* Компьютерная обработка информации. Допечатная подготовка : учеб. пособие / Т.М. Иванова. — СПб. : Питер, 2004. — 367 с.
5. *Агеев В.Н.* Автоматизированные информационно-управляющие системы : метод. указания по выполнению лабораторных работ / В.Н. Агеев. — М. : МГУП, 2007. — 42 с.
6. *Коваленко А.Н.* Управление рабочими потоками : учеб. пособие. — М. : МГУП, 2004. — 110 с.

Учебное издание

Автоматизация управления жизненным циклом продукции

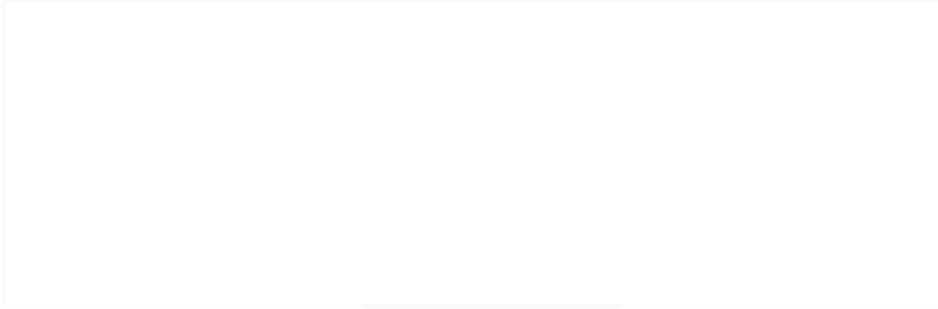
Конспект лекций

для студентов, обучающихся по направлению
15.03.04 — «Автоматизация технологических процессов
и производств» (бакалавры)



Подписано в печать 06.10.15. Формат 60×84/16.
Бумага офсетная. Печать на ризографе. Усл. печ. л. 5,81.
Тираж 100 экз. (1-й завод 50 экз.). Заказ № 134.

Московский государственный университет печати имени Ивана Федорова.
127550, Москва, ул. Прянишникова, д. 2А.
Отпечатано в Издательстве МГУП имени Ивана Федорова.



Автоматизация управления жизненным циклом продукции

Конспект лекций

для студентов, обучающихся по направлению
15.03.04 — «Автоматизация технологических процессов
и производств» (бакалавры)

