

22.18
М.171

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ БССР
ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

И. В. Максименя

МОДЕЛИ ЗАДАЧ ИССЛЕДОВАНИЯ ОПЕРАЦИЙ
(учебное пособие)



Гомель 1984

РЕПОЗИТОРИЙ ГГУ

СКОРИНЫ

Рецензенты: С.А. Русяков, кандидат физико-математических наук Гомельского отделения ИМ АН БССР;
М.И. Кравчук, кандидат физико-математических наук Гомельского политехнического института

В данном учебном пособии излагается постановка задач исследования операций и их формализация в виде моделей. Рассмотрено семь наиболее употребительных типов моделей исследования операций. Пособие предназначено для студентов IУ-У курсов специальностей "Математика" (2013) и "Прикладная математика" (0647).

М 20204 - 021 I - 64 I702070000
М 339 - 64

© Гомельский государственный университет (ГГУ), 1984

I. ВВЕДЕНИЕ

Исследование операций (ИСО) является молодой наукой, которая появилась во время второй мировой войны и использовалась для подготовки решений командующих. Были созданы специальные группы, которые рассматривали вопросы применения оружия и распределения сил и средств по военным объектам. Затем ИСО расширило область применения на промышленность, сельское хозяйство, бытовое обслуживание, транспорт, охрану природы.

I.1. Отличительные особенности ИСО

Отличительными особенностями ИСО можно считать: системный подход. Для оценки любого решения в рамках системы необходимо определить все существенные взаимосвязи и установить его влияние с учетом этих взаимосвязей на поведение всей организации как единого целого, т.е. исследование сознательно усложняет формулировки задачи, пока все существенные взаимосвязи не будут введены в рассмотрение.

Использование комплексных коллективов. При исследовании сложных систем невозможно заранее предсказать, какой из возможных научных подходов окажется наиболее эффективным. Нужно искать такую комбинацию подходов, которая обеспечит решение данной проблемы. Это по плечу коллективу ученых разных специальностей.

Операционный метод. Обычно ИСО имеет дело с системами, не допускающими лабораторного эксперимента. Поэтому операционные группы строят модели. Все операционные модели имеют вид уравнения, в котором общий критерий функционирования всей системы в целом (W) приравнивается некоторому соотношению (f), связывающему между собой множество управляемых (X_i) и неуправляемых (d_j) переменных, определяющих поведение системы:

$$W = f(X_i, d_j)$$

Помощь и участие операционистов в практической реализации результатов. В ИСО можно лишь частично "испытать" результаты. Истинная оценка является в ходе применения в реальных условиях. Поэтому операционисты вынуждены преобразовывать свои решения в набор инструкций и правил для руководителей и исполнителей. Т.е., операционисты составляют некоторые программы для

людей как программы для ЭВМ.

1.2. Типы задач ИСО

Существует две категории задач ИСО: прямые и обратные.

Прямые отвечают на вопрос: что будет, если в заданных условиях мы примем такое-то решение? Обратные задачи отвечают на вопрос: как следует выбрать элементы решения для того, чтобы показатель эффективности был максимальным? Для решения обратной задачи нужно решать прямую. Этой цели служит математическая модель (ММ) операции, позволяющая вычислить показатель эффективности N для любых заданных условий при любом решении.

Если число вариантов велико, то применяются специальные методы целенаправленного их выбора. Составляется зависимость вида $N = f(L_i, X_j)$. При этом считается, что вид (f) нам известен и с помощью ММ можно вычислить для любых заданных L_i и X_j значение N (т.е. прямая задача решена). Обратную задачу можно сформулировать так: при заданных условиях L_i найти такие элементы решения X_j , которые обратят показатель N в максимум (вариационная задача).

Реальнее задачи более сложны: помимо факторов (L_i - условия, X_j - элементов решения) присутствуют неизвестные факторы ξ_l , значение которых нельзя заранее предсказать. Показатель эффективности имеет вид $N = f(L_i, X_j, \xi_l)$, и задача оптимизации решения может быть сформулирована так: при заданных условиях L_i с учетом наличия неизвестных факторов ξ_l найти такие элементы решения X_j , которые по возможности обеспечивают $\max N$ (т.е. имеем задачу о выборе решения в условиях неопределенности).

Если ξ_l являются случайными величинами, статистические характеристики которых известны, то решаем стохастическую задачу исследования операций.

Если операция обладает повторяемостью, то максимизируется среднее значение N , поскольку проигрыш в одном случае будет компенсироваться выигрышем в другом.

Если же операция уникальна, то обычно гарантируют себя от лишнего риска, добиваясь чтобы вероятность события $\{N < a_0\} \leq \delta$ была меньше заданной величины, где δ настолько малое число, что оно означает практически невозможное событие.

Пораздо хуже, когда ξ_l нельзя описать статистическими ме-

тодами. В этом случае либо используется метод экспертных оценок, либо применяются адаптирующиеся алгоритмы. Если же имеет место "враждебная" неопределенность и конфликтные ситуации, которые характерны для боевых действий, спортивной и конкурентной борьбы, то решаются задачи теории игр.

1.3. Классы операционных задач

Можно выделить следующие восемь классов операционных задач:

- 1) управление запасами;
- 2) распределение;
- 3) массовое обслуживание;
- 4) упорядочение;
- 5) выбор маршрута;
- 6) замены оборудования;
- 7) составительские задачи;
- 8) поиски.

Некоторые математические методы можно применить для решения задач разных классов и в то же время для решения задач одного и того же класса.

Управление запасами применяется в тех случаях, когда ресурсы ограничены и существуют два вида издержек, связанных с хранением ресурсов и перевозкой ресурсов. Руководителя в таких случаях интересует выбор такого соотношения хранения и перевозки ресурсов, который обеспечит ему минимум затрат.

В задачах распределения также рассматривается вопрос, связанные с ресурсами, необходимыми для выполнения некоторого заданного вида работ. Однако в отличие от задач управления запасами руководителя прежде всего интересует возможность перераспределения ресурсов, а именно: где необходимо добавить и от каких ресурсов и в каком именно месте целесообразно отказаться от дефицитных ресурсов или же эти ресурсы можно заменить на менее дефицитные. Большинство методов для решения задач распределения носит название методов математического программирования. Различные виды программирования - линейные, нелинейные, стохастические, параметрические и динамические, отличающиеся видом используемой информации и принятыми допущениями.

Задачи массового обслуживания. Часто где выполняется операция, называет средством обслуживания. Вот, что становится

в очередь и подвергается операции независимо от своей природы, называется "заявкой". Если несколько операций может выполняться одновременно, то говорят о различных "каналах" обслуживания, а сам канал может состоять из пунктов обслуживания. Ожидающие "заявки" называют очередью, а порядок, в соответствии с которым клиенты покидают очередь, называется "дисциплиной обслуживания".

Задача сводится к минимизации издержек на обслуживание, связанных с потерями от простоя оборудования и потерями заявок.

Математическая теория массового обслуживания использует широко аппарат теории вероятности и теории дифференциальных и интегральных уравнений.

Задачи упорядочения. Выбор дисциплины обслуживания с таким расчетом, чтобы критерий функционирования системы был минимальным (время обслуживания группы заявок). Либо простые задачи упорядочения решаются аналитически. Для решения почти любой практической задачи упорядочения нужно моделирование. Задачи выбора маршрута. Выбрать маршрут, проходящий через каждый из остальных городов только один раз, который характеризуется минимальной длиной или затратами.

Задачи замены. Существуют два типа элементов, из которых состоит техническая система:

- а) элементы с постоянно ухудшающимися характеристиками;
- б) элементы внезапно выходят из строя.

Задача сводится к определению порядка и сроков замены элементов, при которых сводятся к минимуму общие эксплуатационные затраты и капиталовложения.

Наиболее общим методом решения таких задач является динамическое программирование.

Задача сводится к определению целесообразности замены, а также частоты этой операции. Часто используются методы аналитического моделирования; но для успеха нужно знание распределений отказов.

Состязательные задачи. Задачи теории игр, в которых на решение, принимаемое одним руководителем, влияют решения, принимаемые другим. Используется при военных играх для обучения офицеров в дипломатии для принятия решений при особых ситуациях.

Задачи поиска. Имеется фиксированное количество ресурсов, нужно принять решение: какой нужен выбор (объем и структура выборки). Чем больше ресурсов привлекается, тем выше стоимость поиска, но тем меньше и ожидаемые потери от ошибки. Используются при ревизии или инспекциях, в геологии. В других задачах поиска метод поиска не поддается контролю, а отыскиваемый объект является управлением. Например, изменение спроса на товары от изменения расположения товаров в торговых залах универсамов.

1.4. Содержание задач ИСО

Независимо от метода решения задачи ИСО можно классифицировать по областям применения:

а) Закупки. Сколько, когда и у кого закупить; учитывается емкость складов; ухудшение качества материала со временем, цена т.д.

Для разработки стратегий разведки запасов природных ресурсов и подготовки планов их разработки.

б) Производство. Проектирование предприятий; выбор пунктов размещения ассортимента продукции. Распределение производственных заказов, давшие минимум затрат.

в) Сбыт. (распределение, продажа и обслуживание). Размещение оптовых складов, их емкости и ассортимента запасов. Определяются круги потребителей этих складов. Для определения издержек сбыта продукции, распределение между непосредственными издержками сбыта, рекламой и премиями людям. Определение спроса потребителей на товары.

г) НИР и ОКР. Определение размеров ассигнований на научно-исследовательские работы (НИР). Выбор областей, где целесообразно проводить НИР и опытно-конструкторские разработки (ОКР). Используется также для календарного планирования крупных программ разработки НИР.

д) Упорядочение и выявление узких мест. Определение критической последовательности операций дающих максимум времени реализации комплекса работ. Широко используется метод оценки и пересмотра планов (PERT), с помощью которого производится оценка критического пути (затраты и реализации).

е) Кадры. Определение оптимального соотношения между возрастными группами рабочих, определении квалификации. Разработка методов найма рабочей силы, улучшения методов стимулирования а следовательно, повышения производительности.

з) функционирование и бухгалтерский учет. Методы ревизий, минимизирующие затраты и ошибки. Используются для получения для руководителей экспресс-информации о бухгалтерской стороне дела; изучения кредитных стратегий, формируют и разрабатывают процедуры оценки риска при предоставлении кредита обработки информации по кредитной задолженности.

Вывод. ИСО занимается не объектом изучения, а лишь методами исследований различных объектов.

1 2. МОДЕЛИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Первый из классов моделей ИСО характеризуется следующими чертами:

- показатель эффективности Z представляет собой линейную функцию от элементов решения X_1, \dots, X_n ;
- элементы решения представляют собой ряд неотрицательных переменных $X_1 \geq 0; \dots, X_n \geq 0$;
- ограничительные условия, налагаемые на возможные решения, имеют вид равенств или неравенств;
- имеется такое решение, которое обращало бы в *min* или *max* функцию Z .

Такие задачи принято называть задачами линейного программирования. Разработан специальный аппарат для решения таких задач. Но может возникнуть вопрос: нужен ли такой специальный аппарат? Нельзя ли просто продифференцировать Z по аргументам X_1, \dots, X_n , приравнять производные нулю и решить полученную систему уравнений?

Оказывается, нельзя, так как Z - линейна, то производные от неё по всем аргументам постоянны и нигде в нуль не обращаются.

max или *min* функции Z если он существует, достигается всегда где-то на границе области возможных значений X_1, \dots, X_n , т.е. там, где начинают действовать ограничения.

Математический аппарат линейного программирования позволяет последовательно и в кратчайшие сроки обследовать границы области возможных решений и найти на этих границах то решение, которое является оптимальным, т.е. такую совокупность значений X_1, \dots, X_n , при которой Z обращается в *max* или *min*.

Для того чтобы можно было обеспечить линейность модели

производства, необходимы следующие допущения:

1. Делимость. Для каждого производственно-технологического процесса суммарное количество каждого из потребляемых ресурсов и соответствующая прибыль строго пропорциональны объему выпускаемой продукции, т.е. все показатели технологического процесса могут быть увеличены или уменьшены при сохранении их взаимной пропорциональности.

2. Аддитивность. Если значение каждой из управляемых переменных X_j определено, то полное количество каждого из потребляемых ресурсов, затраченных при реализации всех применявшихся технологических процессов, суммируется, а полная прибыль равна сумме прибылей, полученных в результате реализации этих технологических процессов.

Одновременно с желанием определить оптимальные значения для каждого из решений X_j у руководителя может возникнуть намерение выяснить:

- а) каким образом отразится на получаемой прибыли увеличение каждого из потребляемых ресурсов;
- б) совершенствование того или иного технологического процесса;
- в) изменение стоимости сырья за счет замены ресурса менее дефицитным.

Такого рода вопросы сводятся к вопросу о значении оптимальных значений каждой из переменных X_j представляют собой анализ на чувствительность линейной модели.

Применение линейных оптимизационных моделей при решении практических задач производственного характера сопряжено со следующими трудностями.

При формулировке ограничений оптимизации нужно быть очень осторожным. С одной стороны, можно просмотреть какое-либо существенное ограничение, а с другой - можно наложить такое количество ограничений, что оптимизация окажется невозможной.

На практике приходится учитывать от 100 до 200 ограничений. Число неизвестных в 2-3 раза превышает число ограничений. Получение данных для моделирования представляет собой весьма трудоемкую задачу.

Необходимо обязательно использовать ЭМ, поскольку число перебора вариантов очень велико.

Поскольку математическая модель - абстракция, то нет ни-

какой гарантии, что первая же пробная модель приведет к практически полезному результату.

Полученное решение нужно тщательно изучить на здравый смысл и конечный результат должен быть проверен на чувствительность, а также выяснена зависимость результата от точности представления исходных данных. Если зависимость сильная, то необходимо обеспечить требуемую точность исходных данных.

2.1. Общая формулировка линейных задач ИСО

Линейные задачи формулируются следующим образом. Требуется найти $\max f(x_1, \dots, x_n)$ при ограничениях вида

$$g_1(x_1, \dots, x_n) \leq b_1$$

$$g_2(x_1, \dots, x_n) \leq b_2$$

$$g_m(x_1, \dots, x_n) \leq b_m$$

где $f(x_1, \dots, x_n)$ - целевая функция или эффективность систем, x_1, \dots, x_n - варьируемые параметры, $g_1(x), \dots, g_m(x)$ - ограничения на ресурсы. Существует несколько канонических форм представления линейных моделей:

$$\max \sum_{j=1}^n c_j x_j \quad \text{при наличии ограничений}$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad (i=1, \dots, m); \quad x_j \geq 0 \quad (j=1, \dots, n);$$

$$\min \sum_{j=1}^n c_j x_j \quad \text{при условии } \sum_{j=1}^n a_{ij} x_j = b_i$$

$$b_i \geq 0; \quad x_j \geq 0; \quad i=1, \dots, m; \quad j=1, \dots, n; \quad n > m.$$

2.2. Геометрическая интерпретация линейных моделей

1. Единственное оптимальное решение. Необходимо решить следующую задачу. Найти максимум целевой функции:

$$\max 12x_1 + 15x_2$$

при ограничениях

$$4x_1 + 5x_2 \leq 12; \quad x_1 \geq 0;$$

$$2x_1 + 5x_2 \leq 10; \quad x_2 \geq 0.$$

Построим в системе координат $x_1 O x_2$ многоугольник

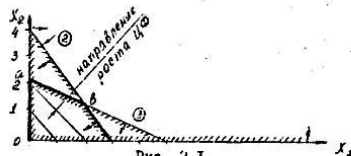


Рис. 2.1.

O, b, a, c - вершины многоугольника допустимых решений.
 a, b, c - экстремальные точки.

$obbc$ - множество решений (рис.2.1), которое является выпуклым множеством (любой допустимый отрезок лежит внутри или проходит вдоль границы). Параллельные прямые представляют собой графическое изображение различных значений целевой функции в направлении градиента их роста. b - оптимальное решение, определяемое экспериментальной точкой, для которой $x_1 = \frac{4b}{7}$ и $x_2 = \frac{6}{7}$. Максимальное значение Z равно $12x_1 + 15x_2 = \frac{120}{7}$. Как видим, имеем единственное оптимальное решение.

Если изменить вид ЦФ на $4x_1 + 10x_2$, тогда графическое изображение целевых функций будет параллельно ab (рис.2.1). В этом случае все точки отрезка ab являются оптимальными, т.е. имеем множество альтернативных оптимальных решений.

2. Неограниченные оптимальные решения. Решить задачу:

$$Z = -2x_1 + 6x_2 \quad \max$$

$$-1x_1 - 1x_2 \leq -2$$

$$-1x_1 + 1x_2 \leq 1$$

$$x_1 \geq 0; \quad x_2 \geq 0.$$

Аналогичным образом представим графически эти уравнения

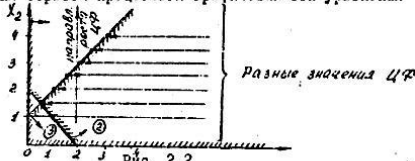


Рис. 2.2

На прямой $-1x_1 + 1x_2 = 1$ при разных значениях Z можно найти точку, удовлетворяющую обоим ограничениям, которая имеет еще большее значение целевой функции Z , чем предыдущая.

3. Не имеющая решения. Решим задачу:

$$\max 1x_1 + 1x_2$$

II

при $-1x_1 + 1x_2 \leq -1 \quad x_1 \geq 0;$
 $1x_1 - 1x_2 \leq -1 \quad x_2 \geq 0.$

Аналогичным образом представим графически эти уравнения

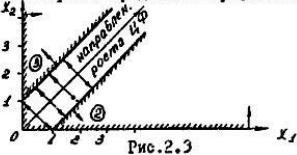


Рис. 2.3

Данная задача не имеет решения, поскольку допустимая область распадается на две несовместимые части.

Для случая $n > 2$ необходимо рассматривать n -мерное евклидово пространство. Каждая точка задается с помощью n координат (x_1, x_2, \dots, x_n) .

Каждому уравнению $\sum_{j=1}^n a_{ij} x_j = b_i \quad (i=1, 2, \dots, m)$ соответствует гиперплоскость, которая делит n -мерное пространство на два подпространства.

Пересечение n гиперплоскостей, соответствующих числу уравнений (ограничений), определяет множество допустимых решений. Целевую функцию также можно представить с помощью гиперплоскости в пространстве решений. Гиперплоскости, соответствующие различным целевым функциям, параллельны друг другу.

2.3. Решение линейных моделей

Планом задачи линейного программирования называется вектор $x = (x_1, \dots, x_n)$, удовлетворяющий системе ограничений

$$a_{11}x_1 + \dots + a_{1n}x_n = b_1 \quad x_j \geq 0;$$

$$a_{m1}x_1 + \dots + a_{mn}x_n = b_m \quad j=1, \dots, n; \quad m > n.$$

План x называется опорным, если векторы P_j , входящие в разложение $P_b = \sum_{j=1}^n P_j x_j$ с положительными коэффициентами x_j , являются линейно независимыми.

Оптимальным планом или решением задачи линейного программирования называется план, доставляющий целевой функции

$$C_1x_1 + C_2x_2 + \dots + C_nx_n$$

экстремальное (\max или \min) значение.

Существует такая крайняя точка выпуклого множества планов K , в которой линейная форма задачи достигает своего экстре-

мального значения.

План, соответствующий крайней точке, имеет не более m положительных компонент, которым соответствуют линейно независимые векторы.

Таким образом, план, соответствующий крайней точке, является опорным.

Для отыскания оптимального плана линейного программирования достаточно исследовать лишь крайние точки множества K . Поскольку в данной системе из n векторов P_j содержится не более чем C_n^m систем, каждая из которых состоит из m линейно независимых векторов, величина C_n^m является верхней границей числа опорных планов задачи.

Необходимо иметь вычислительную схему, позволяющую осуществить упорядоченный переход от данного опорного плана к другому.

Для решения задач линейного программирования используется симплекс-метод. Эта процедура позволяет отыскать начальный опорный план; проверить полученный опорный план на оптимальность; если рассматриваемый опорный план не является оптимальным, метод позволяет перейти к другому плану, на котором значение оптимизируемой линейной формы не больше, чем на предыдущем шаге; через конечное число шагов (от m до $2m$) приходит к оптимальному плану.

2.4. Примеры линейных моделей

2.4.1. Транспортная задача линейного программирования. Эта задача является типовой для промышленных фирм, имеющих несколько предприятий, складов, рынков сбыта и оптовых баз.

Модель применяется при решении логистических задач. В этом случае стратегические решения сводятся к выбору транспортных маршрутов, по которым продукция различных предприятий доставляется на несколько складов или в конечные пункты назначения (рис. 2.4)

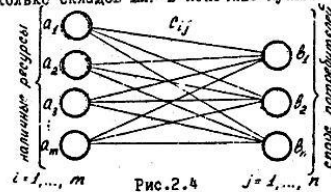


Рис. 2.4

РЕПОЗИТОРИЙ ГИ

C_{ij} - затраты на перевозку единицы груза из пункта i в пункт j ;

A_i и B_j - постоянные величины (соответственно ресурсы и спрос) на рассматриваемом плановом периоде.

Целевой функцией является выбор плана для заданного интервала времени (планового периода), минимизирующий общие транспортные затраты.

С математической точки зрения транспортная задача линейного программирования формулируется следующим образом:

- отыскать набор переменных $\{X_{ij}\}$, $i=1, \dots, m$; $j=1, \dots, n$, минимизирующий линейную целевую функцию

$$L(X) = \sum_{i=1}^m \sum_{j=1}^n C_{ij} X_{ij}$$

и удовлетворяющий системе линейных ограничений:

$$\sum_{j=1}^n X_{ij} = A_i, \quad i=1, \dots, m, \quad X_{ij} \geq 0,$$

$$\sum_{i=1}^m X_{ij} = B_j, \quad j=1, \dots, n, \quad C_{ij} \geq 0;$$

При этом предполагается, что для совместности последних уравнений выполняется условие

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

т.е. общие ресурсы (общая мощность) поставщиков равна общему спросу потребителей.

Транспортную задачу можно решить с помощью симплекс-метода.

2.4.2. Задача распределения ресурсов. Пусть предприятие выпускает неоднородную продукцию; руководитель определяет уровни производства в течение некоторого и перед заданного периода.

Пусть X_i - объем продукции;

C_i - доход с единицы продукции в рублях;

i - на предприятии выпускается два типа продукции A и B , каждый из которых может быть получен с помощью одного из двух технологических процессов (j).

Расходы, связанные с каждым из технологических процессов, определяются:

- трудозатратами - A_j ;

- количеством (в кг) потребляемого материала Y - B_j ;

- количеством (един.) материала Z - d_j .

Всего в наличии имеется:

- трудозатрат - A ;

- материала Y - B ;

- материала Z - D .

Представим задачу в табличной форме.

Таблица 2.1

	На единицу продукции А		На единицу продукции В		Имеется в наличии
	Технич. проц. 1	Технич. проц. 2	Технич. проц. 3	Технич. проц. 4	
Трудозатраты	a_1	a_2	a_3	a_4	A
Материала Y	b_1	b_2	b_3	b_4	B
Материала Z	d_1	d_2	d_3	d_4	D
Доход с единицы продукции в руб.	c_1	c_2	c_3	c_4	max
Объем выпускаемой продукции X_i	X_1	X_2	X_3	X_4	

Когда математическая постановка задачи имеет вид

$$max (C_1 X_1 + C_2 X_2 + C_3 X_3 + C_4 X_4),$$

при ограничениях

$$a_1 X_1 + a_2 X_2 + a_3 X_3 + a_4 X_4 \leq A$$

$$b_1 X_1 + b_2 X_2 + b_3 X_3 + b_4 X_4 \leq B$$

$$d_1 X_1 + d_2 X_2 + d_3 X_3 + d_4 X_4 \leq D$$

$$X_1 \geq 0; X_2 \geq 0; X_3 \geq 0; X_4 \geq 0.$$

2.4.3. Задача выбора оптимального маршрута. На рис. 2.5 представлены возможные маршруты движения из города 1 в город 5. Назовем города узлами.

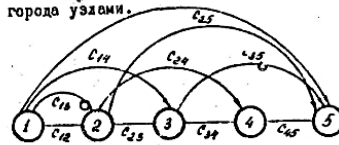


Рис. 2.5

Существует путь из каждого узла i в любой другой узел $j > i$. Путь может проходить либо по одной дуге, либо через промежуточные узлы.

Перемещение вдоль каждой дуги требует затрат C_{ij} . Нужно найти наилучший маршрут из узла 1 в узел 5. Нужно определить, чему равны затраты для самого дешевого маршрута; какой маршрут сопряжен с минимальными затратами.

а) Задачи, связанные с использованием наиболее дешевого маршрута.

Пусть Y_i - затраты, связанные с использованием наиболее дешевого маршрута от узла i в узел 5. ($i=1, \dots, 4$)

$$Y_i = \min [C_{ij} + Y_j] \quad i=1, \dots, 4; \quad j=1, \dots, 5 \quad Y_5 = 0$$

Задача сводится к сравнению затрат по различным маршрутам,

т.е. нужно для $i=1$ $\min Y_1$

при $Y_1 \leq C_{12} + Y_2$

$$Y_1 \leq C_{13} + Y_3$$

$$Y_1 \leq C_{14} + Y_4$$

$$Y_1 \leq C_{15}$$

для $i=2$; $\min Y_2$

при $Y_2 \leq C_{23} + Y_3$

$$Y_2 \leq C_{24} + Y_4$$

$$Y_2 \leq C_{25}$$

для $i=3$; $\min Y_3$

при $Y_3 \leq C_{34} + Y_4$

$$Y_3 \leq C_{35}$$

для $i=4$; $\min Y_4$; $Y_4 \leq C_{45}$

Таким образом, общая задача ставится следующим образом:

$$\min \sum_{i=1}^4 Y_i \quad \text{при всех ограничениях.}$$

б) Выбор маршрута, связанного с наименьшими затратами.

Пусть $X_{ij} \geq 0$ - интенсивность транспортировки потока от узла i до узла j . Целевая функция имеет вид

$$\min \sum_{i=1}^4 \sum_{j=i+1}^5 C_{ij} \cdot X_{ij}$$

при удовлетворении ограничений

$$1 \cdot X_{12} + 1 \cdot X_{13} + 1 \cdot X_{14} + 1 \cdot X_{15} = 1$$

$$-1 \cdot C_{12} + 1 \cdot X_{23} + 1 \cdot X_{24} + 1 \cdot X_{25} = 0$$

$$-1 \cdot X_{13} + 1 \cdot X_{23} + 1 \cdot X_{34} + 1 \cdot X_{35} = 0$$

$$-1 \cdot X_{14} - 1 \cdot X_{24} - 1 \cdot X_{34} + 1 \cdot X_{45} = 0$$

2.4.4. Задача рационального составления смеси. Пусть фирма "Алхимик" торгует химпродуктами, являющимися смесями жидкостей (3-х видов) [1].

Пусть X_{ij} - количество литров i -й компоненты в j -м продукте. В табл. 2.2 представлены основные условия выпуска продукции.

Таблица 2.2

	I литр						Ограничения
	Химический компонент 1 в продукте			Химический компонент 2 в продукте			
	1	2	3	1	2	3	
Наличие хим. компоненты 1	I	I	I				$\leq S_1$
Наличие хим. компоненты 2				I	I	I	$\leq S_2$
Спрос 1 прод.	I			I			$\geq D_1$
Спрос 2 прод.		I			I		$\geq D_2$
Спрос 3 прод.			I			I	$\geq D_3$
Технология 1 прод.	$a_1 \cdot \tau_1$			$a_2 \cdot \tau_2$			$\geq D_1$
Технология 2 прод.		$b_2 \cdot \tau_2$			$b_2 \cdot \tau_2$		≤ 0
Технология 3 прод.			1			$b_3 \cdot \tau_3$	≥ 0
Доход	P_{11}	P_{12}	P_{13}	P_{21}	P_{22}	P_{23}	max
Уровень производства	X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	

Объемы потребляемых химических компонент удовлетворяют ограничениям

$$X_{11} + X_{12} + X_{13} \leq S_1$$

$$X_{21} + X_{22} + X_{23} \leq S_2$$

наличие компонент j фирмы

Запланированный выпуск продукции должен удовлетворять минимальный спрос

$$X_{11} + X_{21} \geq D_1$$

$$X_{12} + X_{22} \geq D_2$$

$$X_{13} + X_{23} \geq D_3$$

D_j - минимальные потребности j -м продукте

Существуют еще и технологические ограничения при приготовлении смеси

$$\frac{a_1 x_{11} + a_2 x_{21}}{x_{11} + x_{21}} \geq \gamma_1 \quad \text{на 1-й продукт}$$

$$\frac{b_1 x_{12} + b_2 x_{22}}{x_{12} + x_{22}} \leq \gamma_2 \quad \text{на 2-й продукт}$$

$$\frac{x_{13}}{x_{23}} \geq \gamma_3 \quad \text{на 3-й продукт}$$

Доход с единицы продукта X_{ij} равен P_{ij} , тогда задача представляется в виде

$$\max \sum P_{ij} \cdot X_{ij}$$

при выполнении всех перечисленных выше ограничений. Задача легко решается симплекс-методом.

2.4.5. Модель назначений. Задача о назначениях формулируется следующим образом: задано n работ, каждую из которых может выполнить любой из n исполнителей. C_{ij} - стоимость выполнения работы i исполнителем j . Нужно распределить исполнителей по работам минимизировать общие затраты.

Определим переменные X_{ij} следующим образом:

$$0 \leq X_{ij} = \begin{cases} 1, & \text{если заказ } i \text{ выполняется} \\ & \text{исполнителем} \\ 0 & \text{в остальных случаях.} \end{cases}$$

$i=1, \dots, n; j=1, \dots, n$

нужно $\min \sum_{i=1}^n \sum_{j=1}^n C_{ij} \cdot X_{ij}$

при ограничениях

$$\sum_{j=1}^n X_{ij} = 1; i=1, \dots, n; \quad \sum_{i=1}^n X_{ij} = 1; j=1, \dots, n$$

Модель является частным случаем транспортной задачи, когда

$$m=n; S_i = D_j = 1.$$

2.4.6. Модель выбора кратчайшего пути. На рис. 2.6 представлена схема связи 8 узлов (городов) друг с другом. Необходимо выбрать кратчайший (самый дешевый) путь передвижения от узла 8 (исток) к узлу 1 (сток).

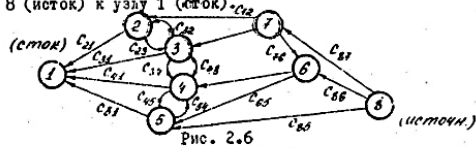


Рис. 2.6

* Дуги в графе связей ориентированы. Установлена стоимость проезда из i узла к j - C_{ij} (стоимость проезда, длина пути, время проезда). Если узлы не связаны, то $C_{ij} = \infty$. Сеть, представленная на рис. 2.6, может содержать циклы. С математической точки зрения данная задача эквивалентна рассмотренной задаче и назначениям. Нужно

$$\min \sum_{(i,j) \in \text{сети}} C_{ij} \cdot X_{ij}$$

$$\sum_{(i,j) \in \text{сети}} X_{ij} = \sum_{(i,k) \in \text{сети}} X_{ik}$$

$$X_{ij} \geq 0 \quad \text{для всех } (i,j) \in \text{сети}$$

при ограничениях

$$1: K = S \text{ источник}$$

$$0: - \text{остальн } K$$

$$-1: K = T \text{ (сток)}$$

На рис. 2.7 представлена матрица стоимости проезда $\|C_{ij}\|$ от i -го узла к j -му узлу (городу)

	1	2	3	4	5	6	7	8
1 (источник)	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}
2		C_{22}	C_{23}	C_{24}	C_{25}	C_{26}	C_{27}	C_{28}
3			C_{33}	C_{34}	C_{35}	C_{36}	C_{37}	C_{38}
4				C_{44}	C_{45}	C_{46}	C_{47}	C_{48}
5					C_{55}	C_{56}	C_{57}	C_{58}
6						C_{66}	C_{67}	C_{68}
7							C_{77}	C_{78}
8 (сток)								C_{88}

Рис. 2.7

Согласно этой матрице можно составить по I уравнению для каждого узла и использовать по I переменной для каждой дуги:

$$X_{15} + X_{16} + X_{17} = 1$$

$$X_{12} + X_{13} + X_{14} - X_{21} - X_{22} = 0$$

$$X_{24} + X_{25} - X_{32} - X_{33} = 0$$

$$X_{34} + X_{35} - X_{43} - X_{44} - X_{45} = 0$$

$$X_{45} + X_{46} + X_{47} - X_{54} - X_{55} - X_{56} = 0$$

и т.д.

$$X_{56} + X_{57} + X_{58} - X_{65} - X_{66} - X_{67} = 0$$

$$Z = \min \sum_{(i,j) \in \text{сети}} C_{ij} \cdot X_{ij}$$

Таким образом, решение указанной системы уравнений, обеспечивающее минимизацию целевой функции, и будет решением задачи выбора кратчайшего пути

3. МОДЕЛИ НА ГРАФАХ

3.1. Задача о кратчайшем пути между вершинами связного неориентированного графа

В задачах ИСО имеет большое значение задача о нахождении кратчайшего пути между двумя вершинами связного неориентированного графа.

3.1.1. Ребра графа единичной длины. Общее правило состоит в том, чтобы каждой вершине X_i присписать индекс L_i , равный длине кратчайшего пути из данной вершины в конечную. Присписывание индексов вершинам в случае графа с ребрами единичной длины производится в следующем порядке:

- 1) X_0 (конечная вершина) присписывается индекс 0.
 - 2) всем вершинам, из которых идет ребро в конечную вершину, присписывается индекс 1.
 - 3) всем вершинам, еще не имеющим индексов, из которых идет ребро в вершину с L_i , присписывается индекс $(L_i + 1)$.
- Этот процесс продолжается до тех пор, пока не будет помечена начальная вершина.

По окончании разметки индексы у начальной вершины будут равны длине кратчайшего пути.

Сам кратчайший путь находят, если двигаться из начальной вершины в направлении убывания индексов.

Пример:

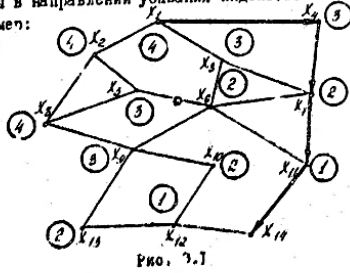


Рис. 3.1

20

3.1.2. Ребра графа произвольной длины. Задача усложняется, так как часто путь, проходящий через *m* или *n* вершин, имеет большую длину, чем некоторые обходные пути.

Алгоритм следующий:

1. Каждая вершина X_i помечается индексом L_i . Первоначально конечной вершине X_0 присписывается индекс $L_0 = 0$. Для остальных вершин полагаем $L_i = \infty (i \neq 0)$.
2. Ищем такую дугу (X_i, X_j) , для которой $L_j - L_i > \ell(X_i, X_j)$, и заменяем индекс L_j индексом $L_j = L_i + \ell(X_i, X_j) < L_j$.
3. Продолжаем этот процесс замены индексов до тех пор, пока остается хотя бы одна дуга, для которой можно уменьшить L_j .

Рассмотрим пример:

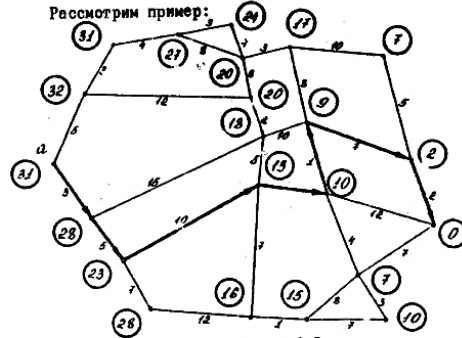


Рис. 3.2

Отметим важное свойство, которым будут обладать присписанные вершинам индексы. Пусть X_p - произвольная вершина. Индекс L_p монотонно уменьшается. Пусть X_q - последняя вершина, послужившая для его уменьшения. Тогда $L_p = L_q + \ell(X_q, X_p)$. Значит, для произвольной вершины X_p с индексом L_p найдется вершина X_q , соединенная ребром с X_p такой, что $L_p - L_q = \ell(X_q, X_p)$.

Это свойство позволяет сформулировать правило нахождения кратчайшего пути.

Пусть $X_n = a$ - начальная вершина с индексом $L_n (L_n = 31)$. Идем вершину X_{p1} , чтобы $L_n - L_{p1} = \ell(X_{p1}, X_n)$. Далее идем вершину X_{p2} так, чтобы $L_{p1} - L_{p2} = \ell(X_{p2}, X_{p1})$ и т.д. до

тех пор, пока не дойдем до конечной вершины $x_{n+1} = x_0 = v$. Путь $M_0(x_1, x_2, \dots, x_n, x_0)$, длина которого будет L_n , является кратчайшим.

3.2. Построение графа наименьшей длины

Имеется несколько графов a, b, c, \dots , которые нужно соединить между собой сетью дорог. Для каждой пары городов (x_i, x_j) известна стоимость строительства соединяющей их дороги $l(x_i, x_j)$. Задача состоит в том, чтобы построить самую дешевую из возможных сетей дорог (сети линий передач, сети нефтепроводов). Называя в графе b , изображающем сеть дорог, величину $l(x_i, x_j)$ длиной ребра, приходим к задаче о построении графа наименьшей длины.

Граф наименьшей длины всегда является деревом, так как если бы он содержал цикл, можно было бы удалить одно из ребер этого цикла и все вершины всё еще остались бы соединенными.

Таким образом, соединения n вершин графа нужно построить $n-1$ ребер. Граф наименьшей длины можно построить, пользуясь правилом:

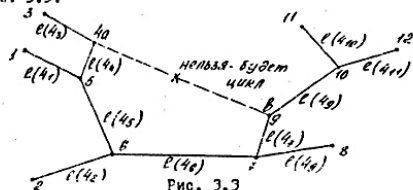
соединим 2 вершины с наиболее коротким соединяющим ребром U_i ;

на каждом из следующих шагов добавляем самое короткое из ребер U_i , при присоединении которого к уже имеющимся ребрам не образуется никакого цикла.

Если имеется несколько ребер одинаковой длины, выбираем любое из них.

Каждое дерево a , построенное таким образом, называют экономическим деревом. Его длина равна сумме длин отдельных ребер $l(a) = l(U_1) + \dots + l(U_{n-1})$;

Пример построения экономического дерева представлен на рис. 3.3.



3.3. Задача разгнания производства в сетевой постановке

Транспортной сетью называется ориентированный связный мультиграф без петель, в котором: а) существует одна и только одна вершина x_0 , называемая входом сети; б) существует одна и только одна вершина x_n , называемая выходом сети; в) каждой дуге q графа отнесено целое число $c(q) > 0$, называемое пропускной способностью дуги.

С понятием транспортной сети тесно связано понятие потока. Пусть x - произвольная вершина. Обозначим U_i^x - множество дуг, заходящих в x ; U_x^i - множество дуг, выходящих из x . Поток $\psi(q)$ транспортной сети называется целочисленная функция $\psi(q)$, определенная на множестве дуг графа и удовлетворяющая свойствам

- 1) $\psi(q) \geq 0 \quad q \in \{x, c(x)\}$ множество дуг графа;
- 2) $\sum_{q \in U_i^x} \psi(q) - \sum_{q \in U_x^i} \psi(q) = 0 \quad \begin{cases} x \neq x_0 \\ x \neq x_n \end{cases}$

Условие сохранения показывает, что в любой промежуточной вершине потока не создается и не исчезает вещество.

- 3) $\psi(q) \leq c(q)$ где $c(q)$ - максимальное количество вещества, протекающего по дуге q в единицу времени.

$\psi(q)$ - реальное количество вещества по дуге в единицу времени. Притоком в вершине x_n (величиной потока в транспортной сети) называется величина

$$\psi_{x_n} = \sum_{q \in U_{x_n}^i} \psi(q) - \sum_{q \in U_{x_n}^o} \psi(q)$$

Для исследования распределения потока по транспортной сети введем понятие разреза сети: множество дуг U_A , заходящих в множество A .

Пропускной способностью разреза называли число

$$c(U_A) = \sum_{q \in U_A} c(q), \text{ т.е. сумма пропускных способностей дуг, входящих в множество } A.$$

Любой элемент вещества, движущегося от x_0 к x_n , пройдет хотя бы однажды по какой-либо дуге разреза U_A . Значит, каковы бы ни были поток ψ и разрез U_A , всегда имеет место неравенство $\psi_{x_n} \leq c(U_A)$ и можно сформулировать лемму.

Если для какого-либо разреза U_A имеем $\psi_{x_n} = c(U_A)$, то:

поток имеет наибольшую величину;
разрез U_i^* обладает наименьшей пропускной способностью.

3.3.1. Задача о наибольшем потоке. Дан граф $G(X)$, являющийся сетью с множеством ребер U и n вершин. X_0 и X_{n+1} соответственно вход и выход сети.

$$c_j^i = \begin{cases} 0 & (x_i, x_j) \notin U \\ c(x_i, x_j) & (x_i, x_j) \in U. \end{cases}$$

Задача о наибольшем потоке сводится к определению такой системы чисел $\varphi(x_i, x_j)$ - потоков в дугах графа, для которой:

$$0 \leq \varphi(x_i, x_j) \leq c_j^i,$$

т.е. поток по ребру (x_i, x_j) не может быть больше пропускной способности дуги и меньше нуля.

Поток, входящий в X_i , равен потоку, выходящему из нее:

$$\sum_{x_k} \varphi(x_k, x_i) = \sum_{x_j} \varphi(x_i, x_j), \text{ т.е. } \varphi(Ux_i) = \varphi(Ux_i);$$

Условие максимальной пропускной способности потока на выходе сети имеет следующий вид: $\varphi_{\max} = \max \sum_{x_i} \varphi(x_i, X_{n+1})$.

Это классическая задача линейного программирования. Форд и Фалкерсон предложили алгоритм, который состоит в постепенном увеличении потока φ_k до тех пор, пока он не станет максимальным. При этом предполагается что пропускные способности дуг $c(x_i, x_j)$ представляют собой целые числа, так что потоки по дугам будут выражаться целыми числами.

3.3.2. Теорема Форда-Фалкерсона. Прямая теорема:

В любой транспортной сети максимальный поток равен минимальной пропускной способности разреза.

Доказывается эта теорема применением алгоритма Форда-Фалкерсона.

Дуга q называется насыщенной, если $\varphi(q) = c(q)$. Поток называется полным если любой путь, ведущий из X_0 и X_{n+1} содержит, по крайней мере, одну насыщенную дугу.

Если данный начальный поток является неполным, то можно определить путь S , который соединяет вершины сети X_0 и X_{n+1} и не содержит насыщенной дуги.

Начиная с данного неполного потока, можно получить полный поток, если прибавлять к различным значениям потоков $c(q)$ пути S наименьшее из чисел $[c(q) - \varphi(q)]$, вычисленное для дуг рассматриваемого пути.

После этого рассматриваемый путь будет содержать, по крайней мере, 1 насыщенную дугу.

Если таким же образом поступить с другими путями такого типа, то получим полный поток φ .

Иначе, алгоритм определения максимального потока состоит из шагов:

1. Строим начальный поток X (состоящий из 0 компонент).

2. Пробегаем, попадая ли X_{n+1} в множество вершин S достижимых по ненасыщенным ребрам из X_0 . Если не попала, то построенный поток максимален и алгоритм останавливается.

3. Если X_{n+1} попала в множество S , то выделяет путь, состоящий из ненасыщенных ребер и ведущий из X_0 в X_{n+1} ;

увеличивает поток через каждое ребро этого пути на величину $\min(c_{ij} - \varphi_{ij})$;

строит новый поток и переходит к пункту 2.

Обратная теорема: Если поток максимален, то найдется такое сечение, пропускная способность которого равна мощности потока. Обычно постулат следующим образом:

а) заданную сеть описывает матрицей $\Sigma = \|c_{ij}\|$ - пропускных способностей всех ребер сети;

б) $X^k = \|\varphi_{ij}^k\|$ - матрица значений потоков на дугах;

в) строит матрицу $\Sigma - X^k = \|c_{ij} - \varphi_{ij}^k\|$;

в этой матрице насыщенным ребром при потоке X^k будут соответствовать нулевые элементы, а ненасыщенным ребрам - ненулевые элементы.

Постому вычисления матрицы $(\Sigma - X^k)$ достаточно как для построения множества вершин, достижимых из X_0 по ненасыщенным ребрам, так и для построения последовательных ребер, ведущих от источника X_0 в сток X_{n+1} .

При выполнении пункта 2 сначала составляется список вершин, в каждую из которых ведет ненасыщенное ребро из вершины i . Далее, для каждой i -й вершины из этого списка составляется свой список вершин, в каждую из которых из данной вершины i ведет ненасыщенное ребро, за исключением тех вершин, которые уже входили в один из ранее составленных списков и т.д.

Этот процесс выписывания списков заканчивается: либо тем, что появилась вершина X_{n+1} , либо приходим к случаю, когда для всех вершин выписаны все списки.

Этот итеративный алгоритм содержит следующую последователь-

ность операции:

- 1) вычисление матрицы $\Sigma - X^k$;
- 2) построение списков вершин;
- 3) определение последовательности ненасыщенных ребер, соединяющих источник X_0 со стоком X_{n+1} ;
- 4) определение величины дополнительного потока $\text{min}\{C_{ij} - \varphi_{ij}\}$;
- 5) построение нового потока X^{k+1} ;
- 6) если в списке отсутствует X_{n+1} , поток X^k представляет собой решение задачи.

3.3.3. Пример задачи о наибольшем потоке. Пусть дана сеть вида, представленного на рис. 3.4, там же указаны и пропускные способности дуг.

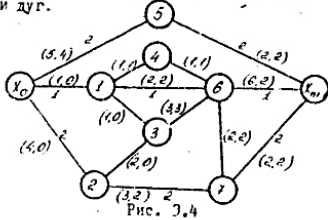


Рис. 3.4

Матрица пропускных способностей имеет вид

Σ	X_0	1	2	3	4	5	6	7	X_{n+1}
X_0	0	1	4	0	0	5	0	0	0
1	0	0	0	1	1	0	2	0	0
2	0	0	0	2	0	0	0	3	0
3	0	0	0	0	0	0	3	0	0
4	0	1	0	0	0	3	1	0	0
5	4	0	0	0	3	0	0	0	2
6	0	2	0	3	1	0	0	2	6
7	0	0	2	0	0	0	2	0	2
X_{n+1}	0	0	0	0	0	2	2	2	0

Исходный поток X_0 представляется матрицей

X^0	X_0	1	2	3	4	5	6	7	X_{n+1}
X_0	0	1	2	0	0	2	0	0	0
1	-1	0	0	0	0	0	1	0	0
2	-2	0	0	0	0	0	0	2	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	-2	0	0	0	0	0	0	0	2
6	0	-1	0	0	0	0	0	0	1
7	0	0	-2	0	0	0	0	0	2
X_{n+1}	0	0	0	0	0	-2	-1	-2	0

Первый шаг алгоритма. Вычисляем матрицу $\Sigma - X^0$:

$\Sigma - X^0$	X_0	1	2	3	4	5	6	7	X_{n+1}
X_0	0	0	2	0	0	3	0	0	0
1	1	0	0	1	1	0	1	0	0
2	2	0	0	2	0	0	0	1	0
3	0	0	0	0	0	0	3	0	0
4	0	1	0	0	0	3	1	0	0
5	6	0	0	0	3	0	0	0	0
6	0	3	0	3	1	0	0	2	5
7	0	0	4	0	0	0	2	0	0
X_{n+1}	0	0	0	0	0	4	3	4	0

Списки вершин

- $(X_0 - 2, 5)$; $(2 - 3, 7)$; $(5 - 4)$; $(3 - 6)$;
 $(7 - \dots)$; $(4 - 1)$; $(6 - X_{n+1})$.

Последовательность ненасыщенных ребер:

- $(X_0, 2)$; $(2, 3)$; $(3, 6)$; $(6, X_0)$. Определяем приращение потока:
 $\text{min}\{2, 2, 3, 5\} = 2$ (величина дополнительного потока = 2)

Тогда новый поток можно представить матрицей X^1 .

X^1	X_0	1	2	3	4	5	6	7	X_{n+1}
X_0	0	0	4	0	0	2	0	0	0
1	-1	0	0	0	0	0	1	0	0
2	-4	0	0	2	0	0	0	2	0
3	0	0	-2	0	0	0	2	0	0
4	0	0	0	0	0	0	0	0	0
5	-2	0	0	0	0	0	0	0	2
6	0	-1	0	-2	0	0	0	0	3
7	0	0	-2	0	0	0	0	0	2
X_{n+1}	0	0	0	0	0	-2	-3	-2	0

Второй шаг. Вычисляем матрицу $\Sigma - X^1$.

$\Sigma - X^1$	X_0	1	2	3	4	5	6	7	X_{n+1}
X_0	0	0	0	0	0	3	0	0	0
1	1	0	0	1	1	0	1	0	0
2	4	0	0	0	0	0	1	0	0
3	0	0	2	0	0	0	0	1	0
4	0	1	0	0	0	3	1	0	0
5	6	0	0	0	3	0	0	0	0
6	0	3	0	5	1	0	0	2	3
7	0	0	4	0	0	0	2	0	0
X_{n+1}	0	0	0	0	0	4	5	4	0

Списки вершин: $(X_0 - 5)$; $(5 - 4)$; $(4 - 2, 6)$; $(2 - 3)$; $(6 - 7, X_{n+1})$.
 Последовательность ненасыщенных ребер: $(X_0, 5)$; $(5, 4)$; $(4, 6)$.
 (6, X_{n+1}). Определяем приращение потока: $\min\{2, 3, 1, 3\} = 1$.
 Величина дополнительного потока = 1.
 Новый поток представляем матрицей X^2 .

X^2	X_0	1	2	3	4	5	6	7	X_{n+1}
X_0	0	1	4	0	0	3	0	0	8
1	-1	0	0	0	0	0	1	0	0
2	-4	0	0	2	0	0	0	2	0
3	0	0	-2	0	0	0	2	0	0
4	0	0	0	0	0	-1	1	0	0
5	-3	0	0	0	1	0	0	0	2
6	0	-1	0	-2	-1	0	0	0	4
7	0	0	-2	0	0	0	0	0	2
X_{n+1}	0	0	0	0	0	-2	-4	-2	0

Третий шаг. Вычисляем матрицу $\Sigma - X^2$.

$\Sigma - X^2$	X_0	1	2	3	4	5	6	7	X_{n+1}
X_0	0	0	0	0	0	2	0	0	0
1	1	0	0	1	1	0	1	0	0
2	4	0	0	0	0	0	0	1	0
3	0	0	2	0	0	0	1	0	0
4	0	1	0	0	0	4	0	0	0
5	7	0	0	0	2	0	0	0	0
6	0	3	0	5	2	0	0	2	2
7	0	0	4	0	0	0	2	0	0
X_{n+1}	0	0	0	0	0	4	6	4	0

Списки вершин: $(X_0 - 5)$; $(5 - 4)$; $(4 - 1)$; $(1 - 3, 6)$; $(3 - 2)$; $(6 - 7, X_{n+1})$.
 Последовательность ненасыщенных ребер: $(X_0, 5)$; $(5, 4)$; $(4, 1)$; $(1, 6)$.
 (6, X_{n+1}). Определяем приращение потока: $\min\{2, 2, 1, 1, 2\} = 1$.
 (величина дополнительного потока = 1).
 Новый поток представляем матрицей X^3 .

X^3	X_0	1	2	3	4	5	6	7	X_{n+1}
X_0	0	1	4	0	0	4	0	0	9
1	-1	0	0	0	-1	0	2	0	0
2	-4	0	0	2	0	0	0	2	0
3	0	0	-2	0	0	0	2	0	0
4	0	1	0	0	0	-2	1	0	0
5	-4	0	0	0	2	0	0	0	2
6	0	-2	0	-2	-1	0	0	0	5
7	0	0	-2	0	0	0	0	0	2
X_{n+1}	0	0	0	0	0	-2	-5	-2	0

Четвертый шаг. Вычисляем матрицу $\Sigma - X^3$.

$\Sigma - X^3$	X_0	1	2	3	4	5	6	7	X_{n+1}
X_0	0	0	0	0	0	1	0	0	0
1	1	0	0	1	2	0	0	0	0
2	4	0	0	0	0	0	0	1	0
3	0	0	2	0	0	0	1	0	0
4	0	0	0	0	0	5	0	0	0
5	8	0	0	0	1	0	0	0	0
6	0	4	0	5	2	0	0	2	1
7	0	0	4	0	0	0	2	0	0
X_{n+1}	0	0	0	0	0	4	7	4	0

РЕПОЗИТОРИЙ ГГУ

Списки вершин: $(X_0, 5)$; $(5, 4)$; $(4, 0)$.

Большее списком предложить нельзя: алгоритм остановлен. Поток X^* является максимальным и его мощность равна 9. Нарисуем его в виде направленного графа (рис. 3.5):

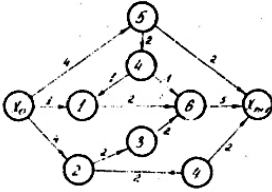


Рис. 3.5

3.4. Задача об оптимальном назначении

Выполнение комплексной работы связано с выполнением совокупности более мелких работ P_1, \dots, P_m , которые могут выполняться независимо. Имеется ряд организаций-исполнителей J_1, \dots, J_n , каждая из которых способна выполнять только некоторые из работ. Каждый из исполнителей может выполнять 1 работу. Нужно распределить работы между исполнителями таким образом, чтобы одновременно выполнялось наибольшее количество работ.

Введем матрицу выполнения работ:

$$\begin{cases} d_{ij} = 1 & \text{— выполняется } i \text{ работ исполнителем } j; \\ d_{ij} = 0 & \text{— не выполняется работа.} \end{cases}$$

Введем переменные ξ_{ij} $\begin{cases} = 1 & \text{— Если } i \text{ -я работа поручена исполнителю } j; \\ = 0 & \text{— не выполняется} \end{cases}$

Переменные ξ_{ij} должны удовлетворять n - ограничениям:

$$\begin{cases} \sum_{j=1}^n \xi_{ij} \leq 1, \quad (i=1, \dots, m), \\ \xi_{ij} \leq d_{ij} \end{cases} \text{ работу } P_i \text{ можно поручить только исполнителю } J_j; \\ \sum_{i=1}^m \xi_{ij} \leq 1, \quad (j=1, \dots, n) \text{ работа поручается только 1 исполнителю, либо никому не поручается.}$$

Нужно максимизировать $\sum_{i=1}^m \sum_{j=1}^n \xi_{ij}$.

Итак, мы имеем задачу об оптимальном назначении. Построим сеть, содержащую в себе $(m+n+2)$ вершин, в которой m вершин соответствует работам, а n вершин-исполнителям плюс источник S и S' сток S' . В дальнейшем обозначим вершины через P_i и J_j .

Ребра в этой сети проведем следующим образом: соединим источник S с каждой вершиной P_i ребром, имея пропускную способность, равную 1 и 0 в обратном направлении; соединим вершину P_i с вершиной J_j только в случае $d_{ij} = 1$ с пропускной способностью, равной 1 и равной 0 в обратном направлении;

соединим каждую вершину J_j со сток S' ребром с единичной пропускной способностью и нулевой в обратном направлении; поток от P_i к J_j обозначим как ξ_{ij} , аналогично обозначим и ξ_{S_i} и $\xi_{j_s'}$.

Каждому распределению работ можно поставить в соответствие поток в только что построенной сети. Любому потоку, если только его компоненты принимают целочисленные значения, соответствует некоторое распределение работ в задаче об оптимальном назначении. Число распределенных работ в точности равно суммарному потоку из вершины S , т.е. мощности потока. Поэтому, для того чтобы решить задачу об оптимальном назначении, достаточно найти в соответствующей сети максимальный поток с целочисленными компонентами.

4. МОДЕЛИ СЕТЕВОГО ПЛАНИРОВАНИЯ А

При ИСО приходится встречаться с задачей рационального планирования сложных комплексных работ. Такого рода работы состоят из ряда элементарных работ, которые взаимно обуславливают друг друга, так что выполнение некоторых работ не может быть начато раньше, чем будут завершены некоторые другие. Если число звеньев большое и они сложным образом обуславливают друг друга, то возникает необходимость в специальных расчетах по методу сетевого планирования. Этот метод позволяет решать как прямые, так и обратные задачи ИСО. Прямые задачи отвечают на вопрос: что будет, если мы примем данную схему организации операции, а обратные отвечают: как нужно организовать операцию, чтобы она обладала максимальной эффективностью.

В основе метода сетевого планирования лежит сетевая модель процесса, с помощью которой можно отобразить объем предстоящих работ, последовательность их выполнения, а также логическую взаимосвязь работ. Сетевая модель изображается в виде единого сетевого графика (сети), который состоит из стрелок, обозначающих те или иные работы (операции), и кружков, которые характе-

РЕПОЗИТОРИЙ ГГУ

ривают свершение отдельных, вполне конкретных событий, отражающих результаты выполнения работ.

Исходят из трех основных понятий: работы, события и пути.

Действительная работа - процесс, требующий затрат труда, ресурсов и времени.

Ожидание - требует только затрат времени (затвердение бетона, осушение).

Фиктивная работа или зависимость - логическая связь между работами, не требующая затрат времени и ресурсов, но показывающая, что начало одной работы обусловлено окончанием другой. Фиктивную работу вводят для отражения правильной взаимосвязи между работами при построении сетевого графика и изображают пунктирной стрелкой.

Событие - результат выполнения одной или нескольких работ. Не имеет продолжительности. Свершение события означает, что можно приступить к последующим работам. Факт свершения события - обязательное условие для начала выполнения выходящих из него работ.

Исходным называется событие, которому не предшествует никакие работы.

Завершающее событие не имеет последующих работ.

Каждая работа сетевого графика объединяет 2 события (начальное и конечное событие), т.е. работа $i-j$ является действием, которое необходимо выполнить, чтобы перейти от предшествующего данной работе события i к событию j . События присваиваются номер. Любая работа кодируется номерами начального и конечного событий $i-j$.

Путьем называем любую последовательность работ сети, в которой конечное событие каждой работы совпадает с начальным событием последующей работы. Путь, начало которого совпадает с исходным событием, а конец - с завершающим событием сети, называют полным. На рис. 4.1 представлен пример сетевого графика.

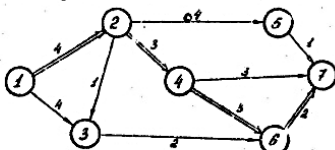


Рис. 4.1
32

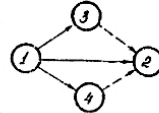
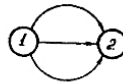
Полные пути: 1-2, 2-5, 5-7; 1-3, 3-6, 6-7; и т.д. Продолжительность каждого из путей L (длина) равна Σ продолжительности составляющих работ:

$$L(1-2-5-7) = 4+4+1 = 9.$$

Полный путь, имеющий максимальную длину (L), называется критическим путем, а работы, лежащие на нем, будут критическими работами. Задержка выполнения любой из работ этого пути приводит к более позднему наступлению завершения события. Полные пути, длины которых меньше длины критического пути, называются не критическими. У них имеются резервы времени, в пределах которых продолжительность отдельных работ может быть увеличена.

Основные правила:

1. В сетевом графике не должно быть работ, имеющих одинаковые обозначения, т.е. работ с общими начальными событиями. Чтобы различать такие работы, нужно ввести дополнительные события и фиктивные работы (рис. 4.2):



а) неверное обозначение

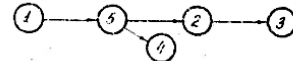
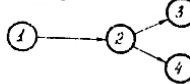
б) верное обозначение

Рис. 4.2

2. Все события, кроме завершающего, должны иметь последующую работу. "Тупики" на графике свидетельствуют либо об ошибке, либо указывают на то, что эта работа является лишней. Не должно быть событий, в которые не входит ни одна работа, так как условия наступления таких событий не обеспечены и не могут совершаться.

3. В сетевом графике не должно быть замкнутых контуров, т.е. событий, соединенных между собой.

4. Если некоторые работы могут быть начаты после частичного выполнения предшествующих работ, то предшествующую работу надо разбить на части и ввести дополнительное событие.



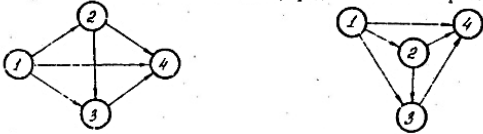
а) неверное обозначение

б) верное обозначение

Рис. 4.3

РЕПОЗИТОРИЙ ГГУ

5. Избегать обозначение вида, представленных на рис. 4.4а.



а) неверное обозначение

б) верное обозначение

Рис. 4.4а

6. Если для свершения событий, кроме работ, указанных на сетевом графике, нужны результаты от других исполнителей, то это отображается способами, указанными на рис. 4.5.

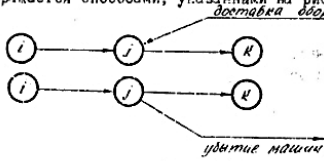


Рис. 4.5

Процесс разработки сетевого графика можно разделить на два этапа: этап подготовки исходных данных; этап построения сетевого графика.

1-й этап: определение взаимозависимости и продолжительности работ. Работы А и Б называют взаимозависимыми, если одна из них может быть начата только по завершению другой. Они могут выполняться только последовательно. В противном случае работы независимы и могут выполняться параллельно. Взаимосвязь между работами можно представить в виде матрицы. В клетке на пересечении строки j -й работы и столбца i -й работы ставим "1", если j -я работа может быть начата только после выполнения i -й работы, и "0" в случае независимости (матрица треугольная внизу).

Для оценки продолжительности работ сети существует два подхода к определению этой величины: детерминистический и вероятностный. В первом случае продолжительность работы оценивают нормативным временем t_{ij} . Во втором случае t_{ij} представляет собой случайную величину.

2-й этап: Построение сетевого графика. Сводится к нанесению на бумагу работ в соответствии с составленной матрицей в двух формах: полигональной и органограммой:

1) работы изображаются в виде безмасштабных стрелок, длина которых не зависит от t_{ij} ;

2) стрелки, изображающие работы, строятся в масштабе времени, вводится шкала рабочего времени в нижней части изображения сети. Работы на масштабном сетевом графике изображаются параллельными отрезками от времени, которые заключены между двумя кружками - событиями.

4.1. Расчет и анализ параметров сетевого графика при фиксированном t_{ij} .

Выполняется для вычисления критического пути и резервов времени для работ, расположенных на других путях и включает операции:

расчет наиболее ранних сроков завершения событий t_i^* ;

расчет наиболее поздних сроков t_i^* ;

расчет резервов времени свершившихся событий R_i ;

Определение критического пути в сетевом графике необходимо для того, чтобы выявить работы, последовательное выполнение которых определяет общий срок окончания работы, и сосредоточить внимание на их своевременном выполнении, поскольку увеличение времени выполнения любой из них приводит к увеличению продолжительности работ всего комплекса.

Для определения R_i для каждой работы ij определяются величины:

- t_{ij}^{pn} - раннее начало;
- t_{ij}^{pn} - позднее начало;
- t_{ij}^{po} - раннее окончание;
- t_{ij}^{po} - позднее окончание.

Каждый кружок разделим на четыре сектора и обозначим следующим образом:



Рассмотрим следующий пример сетевого графика (рис.4.6)

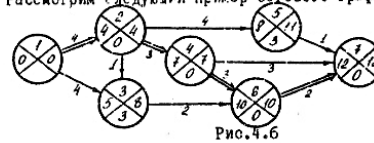


Рис.4.6

Расчет наиболее ранних сроков свершения событий производится в направлении от исходного к завершающему событию. Для любого события сети ранний срок свершения равен длине максимального из путей к данному событию от исходного. Любое событие считается свершившимся лишь в том случае, когда выполнены все работы, входящие в него. Если к событию j сети подходит несколько работ с известными t_i^a , то

$$t_j^a = \max [t_i^a + t_{ij}].$$

Практически при расчете сроков свершения событий последовательно переходят от исходного события к событиям все более от него удаленным. Вычисленные значения t_j^a записываются в соответствующие сектора сетевого графика.

Поздние сроки свершения (t_j^b) рассчитывают последовательно переходя от завершающегося события к событиям, приближаясь к исходному (в обратном порядке). Поздний срок завершающегося события равен его раннему сроку. Любые другие события определяют по формуле:

$$t_i^b = \min [t_j^b - t_{ij}];$$

т.е. от позднего срока всех j -х событий вычитается продолжительность работ, связывающих i -е событие с j -ми, и выбирается минимальная величина.

Расчет резерва времени R_i сводится к определению разности:

$$R_i = t_i^b - t_i^a.$$

Он показывает, на какой предельно допустимый период времени можно задержать свершение события, не изменяя времени свершения завершающегося события сети.

Для каждой из работ сети можно определить остальные параметры:

$$\begin{aligned} t_{ij}^{PH} &= t_i^a && \text{(раннее начало);} \\ t_{ij}^{PN} &= t_{ij}^a - t_{ij} && \text{(позднее начало);} \\ t_{ij}^{PO} &= t_i^b + t_{ij} && \text{(раннее окончание);} \\ t_{ij}^{NO} &= t_j^b && \text{(позднее окончание).} \end{aligned}$$

Когда все эти характеристики для каждого события определены, переходят к анализу сетевого графика:

1. Строят критический путь (выбирают все события с нулевым резервом ($R_i = 0$)).

2. Если через два события с $R_i = 0$ проходят несколько пу-

тей, то критическим является наибольший по продолжительности.

3. Определяют полный и частный резервы.

Полным резервом R_{ij} называется время, в течение которого можно увеличивать продолжительность t_{ij} , не изменяя критический путь.

$$R_{ij} = t_j^a - t_i^a - t_{ij}.$$

Например: $R_{13} = 6 - 0 - 4 = 2$, т.е. работа (1-3) может начаться спустя 4 единицы времени, не вызывая задержки выполнения всей работы. Если полный резерв времени расходуется на какой либо работе пути, то все последовательные работы уже имеют нулевой резерв времени и становятся критическими.

Частный резерв r_{ij} - время, в течение которого можно продолжить t_{ij} не меняя срок раннего начала последовательных работ:

$$r_{ij} = t_j^b - t_i^b - t_{ij}.$$

Например: $r_{13} = 5 - 0 - 4 = 1$. Частный резерв образуется только у работ, непосредственно предшествующих событию j , в котором пересекаются пути разной длины. Он показывает, какая часть полного резерва работы ($i-j$) может быть использована на продолжительности этой и предшествующих ей работ. При этом изъятие этого резерва не вызывает сокращения полного резерва ни у одной из работ, следующих за событием j , т.е. даже не меняет величины полных резервов у работ, следующих за событием j .

4. Для характеристики напряженности сроков выполнения работ используется один из следующих коэффициентов.

$$\text{Коэффициент напряженности работы} \\ R_{ij}^H = \frac{T(L_{max}) - T_{KP}(L_{max})}{T_{KP} - T_{KP}(L_{max})}$$

где $T(L_{max})$ - продолжительность полного пути максимальной длины, переходящего через работу ($i-j$);

T_{KP} - продолжительность критического пути;

$T_{KP}(L_{max})$ - продолжительность участков критического пути, совпадающих с полным путем максимальной длины, проходящего через работу ($i-j$);

L_{max} - максимальный по длине путь, проходящий через работу ($i-j$).

Значения коэффициента напряженности показывает, насколько свободно можно располагать резервами работ и характеризует срочность их выполнения, если это не определено технологичес-

кими связями. Обычно $K_{ij} < 1$ (для критической работы $K_{ij} = 1$).
Коэффициент полного резерва.

$$K_{ij}^p = \frac{R_{ij}}{T(L_{max}) - T_{кр}(L_{max})}$$

Характеризует продолжительность максимального пути, проходящего через работу $(i-j)$ и не совпадающего с критическим путем.

4.2. Расчет сетевого графика при случайном времени выполнения работ

В ряде случаев нельзя пренебречь случайным характером времени выполнения работ t_{ij} , однако исследователю известна $f(t_{ij})$. Предполагая независимость длительности выполнения работ на S -м полном пути, можно найти $f_s(t)$ как композицию законов распределения $f(t_{ij})$. Затем вычисляют вероятность выполнения работ на каждом пути за заданное время t_j по формуле

$$P_s(t_j) = \int_0^{t_j} f_s(t) dt$$

Критическому пути соответствует минимальное значение этой вероятности. Анализ этого пути, а также путей, имеющих вероятности, близкие к вероятности выполнения работ критического пути за t_j , лежит в основе улучшения сетевого графика.

Обычно используют упрощенный метод, основанный на определении средних значений \bar{t}_{ij} и σ_{ij}^2 . Перед началом исследований определяют:

- средние значения и дисперсии длительностей работ t_{ij} ;
- наиболее ранний средний срок свершения каждого события;
- наиболее поздний средний срок свершения каждого события;
- средние резервы времени свершения событий;
- критический путь сети;

$P(t_j)$ - вероятность наступления завершающего события к t_j .

Если же $f(t_{ij})$ неизвестно, то авторы метода PERT (*Program Evaluation and Review Technique* - техника оценки и контроля производственных программ) предположили использовать аппроксимацию Бета-распределением (рис. 4.7) по известным значениям t_{min}, t_{max} и $t_{н.с.}$ (наиболее вероятным значениям):

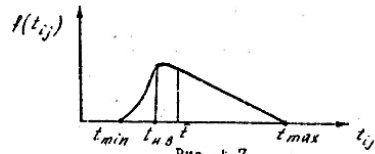


Рис. 4.7

$$\bar{t}_{ij} = \frac{t_{min} + 4t_{н.с.} + t_{max}}{6}$$

$$\sigma_{ij}^2 = \left(\frac{t_{max} - t_{min}}{6} \right)^2$$

При определении вероятности свершения завершающего события за t_j исходят из того, что время распределено по нормальному распределению $N(T_{ок}, \sigma^2)$ как сумма большого числа независимых слагаемых. Здесь

$$T_{ок} = \sum \bar{t}_{ij}$$

$$\sigma^2 = \sum \sigma_{ij}^2$$

определяются на всем критическом пути. Вероятность наступления завершающего события определяется из выражений;

$$Z = \left(\frac{t_j - T_{ок}}{\sigma} \right) - \text{нормированная нормальная величина}$$

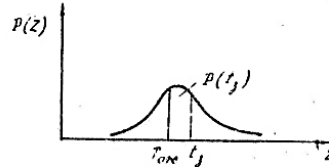


Рис. 4.8

$$P(t_j) = \Phi_0 \left(\frac{t_j - T_{ок}}{\sigma} \right) - \phi \left(-\frac{T_{ок}}{\sigma} \right)$$

т.е. $P(t_j) = 0,5 + \Phi_0 \left(\frac{t_j - T_{ок}}{\sigma} \right)$.

РЕПОЗИТОРИЙ ГГУ

4.3. Оптимизация плана комплекса работ

Оптимизация плана комплекса работ может проводиться с разными целями:

а) Может оказаться, что общее время выполнения работ нас не устраивает, поэтому ставится вопрос - как нужно форсировать работы, чтобы $T \leq T_0$.

Имеет смысл форсировать критические работы. Но критический путь может измениться и наиболее слабыми местами окажутся другие работы. Форсирование работ требует вложения каких-то средств и поэтому возникает задача ИСО: какие дополнительные средства и в какие работы следует вложить, чтобы общий срок выполнения комплекса работ был не больше заданной величины T_0 , в расход дополнительных средств был минимальным.

б) Перераспределение имеющихся средств между отдельными работами. Иногда оказывается возможным, перебросив силы и средства с не критических участков плана на критические, добиться уменьшения Σt выполнения операций. Снова возникает типичная задача исследования операций: какие силы и средства надо перебросить с одних работ на другие для достижения минимального времени выполнения комплекса работ.

в) Еще одна постановка задачи. Стало известно, что $T < T_0$, т.е. у нас есть резерв времени. Можно растянуть работы и сэкономить средства. Ставится вопрос: до каких пределов можно увеличить t_{ij} и у каких работ (i, j), чтобы экономия была максимальная, т.е. ставится задача оптимизации только отдельных не критических путей, на которых выявлены временные резервы.

Предположим, что критический путь один. Тогда возможны следующие три задачи ИСО.

Задача 1. Комплекс НИР состоит из работ A_1, \dots, A_n с временем выполнения t_1, \dots, t_n критический путь известен и $T = \Sigma t_i > T_0$.

Известно, что вложение определенной суммы X дополнительных средств в работу A_i сокращает время выполнения с t_i до $t_i^* = f_i(X) < t_i$.

Какие дополнительные средства X_1, X_2, \dots, X_n следует вложить в каждую из работ, чтобы:

- срок выполнения комплекса НИР был не выше заданного T_0 , т.е. $T' = \Sigma f_i(X_i) \leq T_0$;

сумма вложенных средств достигала минимума:

$$X = \sum_{i=1}^n X_i (\min)$$

В общем случае поставленная задача относится к классу задач нелинейного программирования. Если $f_i(X_i)$ - линейные или изменяемые планы небольшие и их можно считать линейными, то приходишь к задаче линейного программирования, которую можно решить, например, симплекс-методом. Если критический путь изменяется, то нужно добавить большое число ограничений в задачу.

Задача 2. Имеется совокупность работ A_1, \dots, A_n с временами выполнения t_1, \dots, t_n . Время выполнения комплекса работ выражается $T = \Sigma t_i$ (на критическом пути). На некритических работах имеются резервы времени.

Имеется некоторый неизменный запас подвижных средств, который распределяется между работами соответственно

$$b_1, \dots, b_n; \sum_{i=1}^n b_i = B$$

Известно, что количество средств $X > 0$, снятое с работы A_i , время её выполнения

$$t_i = f_i(X_i) < t_i,$$

а количество средств X_i , вложенное в работу A_i , уменьшает время её выполнения до

$$t_i^* = \varphi_i(X_i) < t_i.$$

Величины X_i должны удовлетворять ограничениям $X_i > -b_i$. Причем ($X_1 + \dots + X_n = 0$) - переброска работ. Для тех работ, на которые перебрасываются средства $t_i = \varphi_i(X_i)$, а для работ, с которых средства снимают $t_j = f_j(|X_j|)$. Общий срок выполнения комплекса работ будет

$$T' = \sum_{(i \in \Pi)} \varphi_i(X_i) + \sum_{(j \in \Pi')} f_j(|X_j|),$$

где i - работы, на которые переносятся средства, если они входят в критический путь,

j - работы, с которых забираются средства, если они входят в критический путь.

Стоит задача; найти такие значения переменных X_i ($i=1, \dots, n$), чтобы выполнялись ограничения $X_i > -b_i$ и T' было минимальным. Это задача нелинейного программирования, даже в случае когда функции f_i и φ_j линейны.

РЕПОЗИТОРИЙ ГГУ

Существенно нелинейным является то, что значения i, j - номеров работ, на которые распространяется сумма (т.е. критических работ), сами зависят от T_i .

Задача 3. Имеется комплекс работ A_1, \dots, A_n с временами выполнения t_1, \dots, t_n . Для него установлен критический путь $T > T_0$ (заданный срок исполнения). Предполагается снизить темпы выполнения некоторых работ с тем, чтобы срок выполнения комплекса довести до заданного значения T_0 , за счет чего предполагается получить экономия средств.

Донесение $(t_i + T)$ высвобождает некоторые средства. Известно, что $T_i = f_i(T)$. Найти, насколько следует задержать выполнение каждой работы A_i , чтобы $T = T_0$, а экономия была максимальной, т.е.

$$\sum_{i \in K} (t_i + T_i) < T_0$$

$$\max = \sum f_i(T_i)$$

Снова имеем задачу нелинейного программирования, и успех её решения зависит от вида функции $f_i(T_i)$.

5. МОДЕЛИ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Модели динамического программирования применяются в случае, когда операция O распадается на ряд "шагов" или "этапов". Показатель эффективности W складывается из слагаемых, полученных на отдельных этапах.

Примеры:

1⁰. Планируется деятельность группы промышленных предприятий P_1, \dots, P_k на период в m лет. В начале каждого года суммарные средства заново перераспределяются. Ставится задача: определить количество средств в начале каждого года, которое нужно выделить каждому предприятию, чтобы максимизировать суммарный доход за m лет.

2⁰. Космическая ракета состоит из $(m+1)$ ступеней. Процесс её выхода на орбиту состоит из m этапов. На каждом этапе очередная ступень сбрасывается:

$$Q_{\text{рак}} = Q_{\text{рак}} + \sum_{i=1}^m G_i$$

где G_i - вес горючего i -го этапа.

Необходимо определить потребление веса G_i на каждом этапе, чтобы минимизировать расход горючего.

3⁰. Владелец машины эксплуатирует её в течение m лет. В начале каждого года он может принять решение продать, отремонтировать, продолжить эксплуатировать машину. Определить: какие нужно принять решения по годам для того, чтобы суммарные расходы на эксплуатацию, ремонт и приобретение новой машины были минимальными.

Любую многошаговую задачу можно решать по-разному:

- или искать элементы решений на всех шагах;
- или же строить оптимальное управление постоянно, шаг за шагом.

Процесс динамического программирования разворачивается от конца к началу:

- раньше всего планируется последний $(m-1)$ шаг;
- затем для каждого из предположений находят условные оптимальные управления на каждом шаге. Например, определяется управление $(m-1)$ шаг плюс условный оптимальный выигрыш на оставшихся двух шагах;
- далее оптимизируется управление на $(m-2)$ шаге и условный выигрыш на оставшихся 3-х шагах и т.д., пока не дойдем до 1 шага.

Процесс оптимизации управления проводится дважды. Сдин раз - от конца к началу. С определением условных оптимальных управлений и условными оптимальными выигрышами до конца всех шагов. И второй раз - от начала к концу в результате находят шаговые управления, обеспечивающие оптимум в целом.

5.1. Модель управления запасами

Пусть X_t - выпуск продукции в течение отрезка t ;

i_t - уровень запасов на конец отрезка;

D_t - спрос на продукцию для t -го отрезка ≥ 0 .

На значения X_t и i_t наложены следующие ограничения:

1) целочисленность объемов выпуска, т.е.

$$X_t = 0, 1, 2, 3, \dots \quad (t=1, \dots, N);$$

2) для администрации предприятия, чтобы $i_N = 0$ (все запасы к концу N -го отрезка были распроданы);

3) возможность полного и своевременного удовлетворения спроса в пределах каждого отрезка:

$$i_t = i_{t-1} + X_t - D_t;$$

$$\begin{pmatrix} \text{Уровень запасов к концу отрезка } t \end{pmatrix} = \begin{pmatrix} \text{Уровень запасов на предыдущем отрезке } (t-1) \end{pmatrix} + \begin{pmatrix} \text{Выпуск продукции на отрезке } t \end{pmatrix} - \begin{pmatrix} \text{Спрос на отрезке } t \end{pmatrix}$$

Кроме того, i_{t-1} и X_t должны быть достаточно велики для того, чтобы уровень запасов на конец отрезка был $b_i \geq 0$.

Существенным для данной задачи является тот факт, что функция затрат $C_t(x_t, i_t)$ является нелинейной. Например, если рассмотреть задачу из 4 периодов, то матрица ограничений будет иметь вид (табл. 5.1).

Таблица 5.1

Периоды	X_1	i_1	X_2	i_2	X_3	i_3	X_4	Спрос
1	1	-1						$D_1 - i_0$
2		1	1	-1				D_2
3				1	1	-1		D_3
4						1	1	D_4

Здесь $d_t = D_N$; $d_N = D_1$; $C_t(X_{ij}) = C_N(X_{ij})$. Обозначим $f_n(i)$ - стоимость, отвечающая стратегии минимизации затрат на n оставшихся отрезках при начальном уровне запасов равном i . $X_n(i)$ - выпуск продукции, обеспечивающий достижение затрат $f_n(i)$. На конце планового периода A имеем:

$$f_0(0) = 0; \quad (n=0) \quad \text{запасов нет;}$$

$$n=1; \quad f_1 = c_1(x_1, i+x-d_1) + 0;$$

$n=2$; $C_2(x_1, i+x-d_1) + f_1(i+x-d_2)$ - общие затраты для двух шагов;

$$f_2(i) = \min_x [C_2(x, i-x-d_2) + f_1(i+x-d_2)],$$

где $i=0, 1, \dots, d_1+d_2$;

$$d_2 - i \leq x \leq d_1 + d_2 - i \quad \text{- любые целые } \geq 0 \text{ значения } X.$$

Общее рекуррентное соотношение записывается следующим образом:

$$f_n(i) = \min_x [C_n(x, i+x-d_n) + f_{n-1}(i-x-d_n)],$$

где $i=0, 1, \dots, N$; $d_{n-1} \leq x \leq N-i$.

§ 5.2. Задача о наборе высоты и скорости самолета

Самолет находится на высоте H_0 и летит со скоростью V_0 . Нужно подняться на $H_{кон}$ и лететь при этом скоростью $V_{кон}$.

Можно поочередно менять либо скорость, либо высоту. При этом известен расход горючего, требуемый для подъема H_2 при постоянной скорости V . Известен расход горючего, необходимый для набора скорости $V_2 > V_1$ при постоянной высоте H . Дана матрица расхода горючего (рис. 5.1) $\|C_{ij}\|$.

H_k	V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
12	8	10	12	14	16	18	20	22	24	26
10	14	12	10	8	6	4	2	0	-2	-4
12	10	8	6	4	2	0	-2	-4	-6	-8
10	12	10	8	6	4	2	0	-2	-4	-6
10	10	12	14	16	18	20	22	24	26	28
10	10	12	14	16	18	20	22	24	26	28
10	10	12	14	16	18	20	22	24	26	28
10	10	12	14	16	18	20	22	24	26	28
10	10	12	14	16	18	20	22	24	26	28
10	10	12	14	16	18	20	22	24	26	28

Рис. 5.1

Нужно минимизировать расход горючего при достижении $(H_{кон}, V_{кон})$.

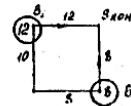
Каждый шаг увеличивает либо только высоту, либо только скорость на величину

$$\Delta H = \frac{H_{кон} - H_0}{n_1}; \quad \Delta V = \frac{V_{кон} - V_0}{n_2},$$

где $n = n_1 + n_2$ - общее количество шагов процесса набора высоты.

Обозначим расход горючего через W . Согласно матрице $\|C_{ij}\|$. На каждом отрезке изменения ΔH и ΔV имеется свой расход горючего C_{ij} (в условных единицах).

Шаг I.



РЕПОЗИТОРИЙ ГИ

лами игры вариантов. Личным ходом называется сознательный выбор одним из игроков одного из возможных в данной ситуации ходов и его осуществление. Случайным ходом – выбор с помощью механизма случайного выбора. Правила игры должны для каждого хода указывать распределение вероятностей возможных исходов. Некоторые игры могут состоять только из случайных ходов (чисто в-чартные) или только из личных ходов (шахматы). Карточные игры – смешанного типа.

Игры могут классифицироваться не только по характеру ходов, но по характеру и объему информации, доступной каждому игроку относительно действий другого. Игра с полной информацией называется, когда каждый игрок при каждом личном ходе знает результаты всех предыдущих ходов (шахматы). Большинство игр, имеющих практическое значение, не принадлежит к этому классу, так как известны будущие действия противника.

Стратегией игрока называется совокупность правил, определяющих однозначно выбор при каждом личном ходе данного игрока в зависимости от ситуации, сложившейся в процессе игры. В зависимости от числа возможных стратегий игры делятся на конечные и бесконечные.

Конечная игра, в которой игрок А имеет m стратегий, а игрок В – n стратегий, называется игрой $m \times n$.

Обозначим эти стратегии через A_1, A_2, \dots, A_m ; B_1, B_2, \dots, B_n . Пусть каждая сторона выбрала стратегии A_i и B_j . Если игра состоит из личных ходов, то выбор стратегий A_i, B_j однозначно определяет наш выигрыш Q_{ij} . Если игра содержит, кроме личных, случайные ходы, то выигрыш при паре стратегий A_i, B_j есть величина случайная, зависящая от всех случайных ходов. Одной выигрыша будет его математическое ожидание, также обозначаемой через Q_{ij} .

Пусть нам известны значения Q_{ij} выигрыша при каждой паре стратегий. Значения Q_{ij} можно записать в виде матрицы, строки которой соответствуют каждой стратегии (A_i) , а столбцы – стратегиям (B_j) . Такая матрица носит название платежной. Платежная матрица или матрица игры имеет вид (рис. 6.1)

$A \setminus B$	B_1	B_2	...	B_n
A_1	Q_{11}	Q_{12}	...	Q_{1n}
A_2	Q_{21}	Q_{22}	...	Q_{2n}
...
A_m	Q_{m1}	Q_{m2}	...	Q_{mn}

Рис. 6.1

Будем обозначать платежную матрицу, представленную на рис. 6.1. Рассмотрим пример. В нашем распоряжении имеется 3 вида вооружения. У противника A_1, A_2, A_3 также имеется 3 вида самолетов: B_1, B_2, B_3 . Наша задача – поразить самолет. Задача противника – сохранить самолет. Матрица игры имеет вид (рис. 6.1).

$A \setminus B$	B_1	B_2	B_3
A_1	0,9	0,4	0,2
A_2	0,3	0,6	0,8
A_3	0,5	0,7	0,2

Рис. 6.2

Итак, имеем игру 3×3 с двумя личными ходами и одним случайным:

- наш личный ход – выбор вооружения A_i ;
- личный ход противника – выбор типа самолета B_j ;
- случайный ход – признание вооружения (поражение самолета) с заданной вероятностью.

Наш выигрыш равен 1, если самолет поражен и он будет равен 0 в противном случае. Среднее значение выигрыша соответствует вероятности поражения данного самолета j данным оружием i .

Целью теории игр является выработка рекомендаций для разумного поведения игроков в конфликтных ситуациях, т.е. определение "оптимальной стратегии" каждого из них. Оптимальной стратегией игрока называется такая стратегия, которая при многократном повторении игры обеспечивает данному игроку максимально возможный средний выигрыш. Основой стратегии является предположение, что противник по меньшей мере так же разумен, как и мы, и сделает все для того, что бы помешать нам добиться своей цели, т.е. не учитываются элементы риска, просчеты и ошибки игроков. Важнейшее ограничение игры состоит в том, что выигрыш искусственно сводится к одному единственному числу.

На практике приходится принимать несколько критериев успеха. Стратегия оптимальная по одному критерию, не обязательно будет оптимальной по другим.

6.1. Принцип минимакса

Выбирая стратегии A_i мы всегда должны рассчитывать, что противник ответит той из стратегий B_j , для которой наш выигрыш a_{ij} минимальный, т.е. значение выигрыша $d_i = \min a_{ij}$; т.е. в результате разумных действий противника мы не выиграем больше d_i . Мы должны оставаться на той из стратегий A_i , для которой d_i максимально, т.е. $d = \max d_i$. Таким образом, $d = \max \min a_{ij}$.

Величина d называется нижней ценой игры (максиминный выигрыш или максимин). Число d лежит в определенной строке матрицы $\|a_{ij}\|$. Стратегия игрока А, которая соответствует этой строке, называется максиминной стратегией, т.е. при любом поведении противника нам гарантирован выигрыш не меньше d (нижняя цена игры). Это чисто перестраховочная стратегия и дает гарантированный минимум.

Аналогичное рассуждение можно провести и за противника В. Противник заинтересован, чтобы обзавестись наш выигрыш в минимум, т.е.

$$\beta_j = \max a_{ij}; \quad \beta = \min \beta_j;$$

т.е. $\beta = \min \max a_{ij}$ - верхняя цена игры (минимакс).

Соответствующая минимаксному выигрышу стратегия противника называется его минимаксной стратегией. Противник при этом гарантирует себе следующее: что бы мы ни предприняли, против него, он во всяком случае проиграет сумму не большую, чем β .

Принцип осторожности, диктующий игрокам выбор соответствующих стратегий, называет "принципом минимакса".

Если $d = \beta$, т.е. чистая цена игры равна ψ , то для игры минимаксные стратегии становятся устойчивыми. Элемент в платежной матрице, который имеет одновременно минимум по строке и максимум по столбцу, называет седловой точкой матрицы, а про игру говорят, что она имеет седловую точку. Седловой точке соответствует пара минимаксных стратегий (A_i, B_j) . Эти стратегии называются оптимальными, а (A_i, B_j) будет решение игры.

Если один из игроков (А) придерживается своей оптимальной стратегии, а другой игрок (В) будет отклоняться от своей оптимальной стратегии, то это никогда ему не выгодно.

Итак, для каждой игры с седловой точкой существует реше-

ние, определяющее пару оптимальных стратегий обеих сторон, отличающиеся своими свойствами:

1) средний выигрыш равен чистой цене игры ψ , являющейся ее верхней и нижней ценой.

2) если одна из сторон придерживается своей оптимальной стратегии, а другая отклоняется от своей, то от этого отклоняющаяся сторона может только потерять, а не увеличить свой выигрыш.

Ставится вопрос - нельзя ли гарантировать себе выигрыш больше d , если применять не одну стратегию, а чередовать случайным образом несколько стратегий?

Каждая чистая стратегия есть частный случай смешанной, в которой все стратегии, кроме одной, применяются с нулевыми частотами, а данная с частотой 1.

Теорема Неймана. Каждая конечная игра имеет, по крайней мере, одно решение.

Выигрыш, полученный в результате решения, называется ценой игры: $d \leq \psi \leq \beta$.

Для смешанных стратегий A_1, A_2, A_3 с частотами p_1, p_2, p_3 смешанная стратегия обозначается в виде:

$$S_A = \begin{pmatrix} A_1 & A_2 & A_3 \\ p_1 & p_2 & p_3 \end{pmatrix}$$

Аналогично смешанную стратегию противника будем обозначать в виде:

$$S_B = \begin{pmatrix} B_1 & B_2 & B_3 \\ q_1 & q_2 & q_3 \end{pmatrix},$$

где $q_1 + q_2 + q_3 = 1$.

Пусть нами найдено решение игры, состоящее из двух смешанных стратегий (S_A^*, S_B^*) . Будем называть стратегии, входящие в оптимальную стратегию игрока, его "полезными" стратегиями. Если один из игроков придерживается своей оптимальной смешанной стратегии $S_A^* (S_B^*)$, то выигрыш остается неизменным и равным цене игры ψ , независимо от того, что делает другой игрок (если мы только не выходим за пределы своих "полезных" стратегий).

6.2. Геометрическая интерпретация игровых моделей

Иногда задачу удается упростить путем вычеркивания некоторых линий стратегий. Для примера рассмотрим игру 2×2 с матрицей игры равной

	B_1	B_2
A_1	a_{11}	a_{12}
A_2	a_{21}	a_{22}

Пусть у данной матрицы седловой точки нет ($\alpha \neq \beta$).
 Нужно найти $S_A^* = (A_1, A_2)$ и оптимальную смешанную стратегию, т.е. определить вероятности P_1 и P_2 .
 Составим уравнения для цены игры

$$\begin{cases} a_{11}P_1 + a_{12}P_2 = Y \\ a_{21}P_1 + a_{22}P_2 = Y \end{cases}; \quad P_1 + P_2 = 1;$$

$$a_{11}P_1 + a_{21}(1-P_2) = a_{12}P_1 + a_{22}(1-P_2), \quad \text{откуда}$$

$$P_1 = \frac{a_{22} - a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}; \quad P_2 = 1 - P_1.$$

Цену игры Y найдем, подставляя значения P_1, P_2 в любое из указанных уравнений.

Если цена игры известна, то для определения оптимальной стратегии противника $S_B^* = (q_1, q_2)$ достаточно 1 уравнения:

$$a_{11}q_1 + a_{12}q_2 = Y,$$

поскольку $q_1 + q_2 = 1$, то для вычисления q_1 и q_2 имеем уравнения

$$q_1 = \frac{Y - a_{12}}{a_{11} - a_{12}}; \quad q_2 = 1 - q_1.$$

Решениям игры 2×2 можно дать простую геометрическую интерпретацию (рис. 6.4)

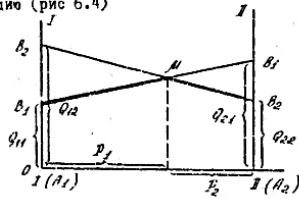


Рис. 6.4

Здесь линии I-I и II-II соответствуют нашим стратегиям A_1 и A_2 . Тогда линии B_1, B_2 будут стратегиями B_1 и B_2 стратегиями B_2 противника B .
 Нужно найти оптимальную стратегию S_A^* .

Строим нижнюю границу выигрыша при стратегиях B_1 и B_2 , т.е. ломаную B_1MB_2 . Эта нижняя граница будет выражать минимальный выигрыш игрока A при любых его смешанных стратегиях. Точка M , в которой этот минимальный выигрыш достигает максимума, и определяет решение и цену игры Y .

Итак, решение игры определяется точкой пересечения стратегий. Но это не всегда так. На рис. 6.5 показан другой случай, когда матрица игры имеет седловую точку, а стратегия A_1 заведомо невыгодна, так как при любой чистой стратегии дает меньший выигрыш, чем A_2 .

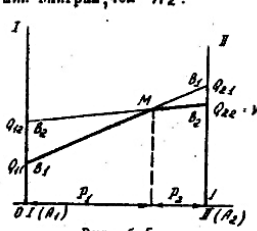


Рис. 6.5

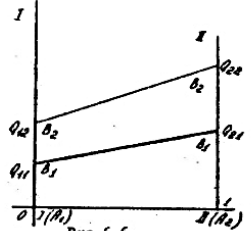


Рис. 6.6

На рис. 6.6 показан случай, когда нижняя граница выигрыша совпадает со стратегией B_1 , а стратегия B_2 также для противника является заведомо невыгодной. Пользуясь геометрической интерпретацией, можно дать простой способ решения любой игры $2 \times n$ (рис. 6.7).

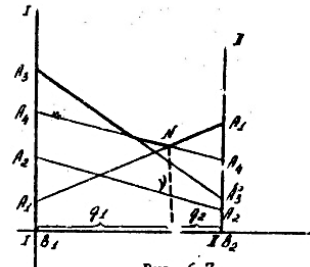


Рис. 6.7

Непосредственно по чертежу (рис 6.7) находим пару "полезных" стратегий противника B_j и B_k , пересекающихся в точке M (если в точке M пересекаются более двух стратегий, выбирает любые две из них).

6.3. Общие методы решения конечных игр

Дана игра $m \times n$

A_1, A_2, \dots, A_m
 B_1, B_2, \dots, B_n

$\|a_{ij}\|$ - платежная матрица.

Найти

$$S_A^* = (A_1, A_2, \dots, A_m);$$

$$S_B^* = (B_1, B_2, \dots, B_n);$$

$$P_1 + \dots + P_m = 1; \quad Q_1 + \dots + Q_n = 1;$$

γ - неизвестно. Ясно, что $\gamma > 0$.

Для оптимальной стратегии средний S_A^* выигрыш при стратегии B_j будет равен

$$a_j = P_1 a_{1j} + P_2 a_{2j} + \dots + P_m a_{mj}.$$

Наша оптимальная стратегия обладает тем свойством, что любое $a_j \geq \gamma$, т.е. получают условия

$$P_1 a_{11} + P_2 a_{21} + \dots + P_m a_{m1} \geq \gamma$$

$$P_1 a_{12} + P_2 a_{22} + \dots + P_m a_{m2} \geq \gamma$$

$$P_1 a_{1n} + P_2 a_{2n} + \dots + P_m a_{mn} \geq \gamma$$

Поделим неравенство на γ и введем обозначения

$$P_j = \xi_j; \quad \dots \quad P_m = \xi_m;$$

$$\xi_j \geq 0;$$

$$\xi_1 + \xi_2 + \dots + \xi_m = 1.$$

Тогда неравенства запишутся в виде

$$a_{11} \xi_1 + a_{21} \xi_2 + \dots + a_{m1} \xi_m \geq 1$$

$$a_{12} \xi_1 + a_{22} \xi_2 + \dots + a_{m2} \xi_m \geq 1$$

$$a_{1n} \xi_1 + a_{2n} \xi_2 + \dots + a_{mn} \xi_m \geq 1$$

Или хотим гарантировать выигрыш максимально возможным, т.е. задаче нахождения решения игры сводится к следующей математической задаче: определить неотрицательные величины $\xi_1, \xi_2,$

\dots, ξ_m , удовлетворяющие неравенствам таким образом, чтобы

$$\Phi = \xi_1 + \xi_2 + \dots + \xi_m,$$

т.е. имеем задачу линейного программирования

6.4. Задача "Полицейские и воры"

Имеется универсальный магазин с двумя отделениями А и В. В А толпа всегда густая и определяется ассортиментом товаров в отделении. В В толпа редкая, товары более дорогие. Оба отделения А и В снабжены телевизионной установкой Т. Магазин обслуживает двое полицейских. Предполагается в "игре", что они ловят одного вора. На основе многолетних наблюдений

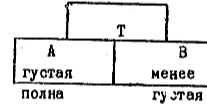


Рис. 6.8

Полицейские установили вероятности поимки воров, задаваемые матрицей игры (рис.6.9):

	A	B
ТТ	0,54	0,75
АВ	0,64	0,36
ВВ	0,19	0,91
ТА	0,50	0,60
ТВ	0,37	0,85
АА	0,46	0,76

Рис. 6.9

Полицейские имеют множество стратегий (X_1, \dots, X_6) , вор имеет две стратегии (Y_1, Y_2) , а именно: брать товары в А или В. Частоты стратегий определяются из соотношений

$$X_1 + X_2 + X_3 + X_4 + X_5 = 1;$$

$$Y_1 + Y_2 = 1;$$

$$X_i \geq 0; \quad Y_i \geq 0;$$

Полицейские и вори в равной степени хитры и осторожны: полицейские имеют стратегии (X_1, \dots, X_6) такую, чтобы обеспечить задержание вора с вероятностью Y не зависящую от поведения воров. Аналогично вори имеют стратегии (Y_1, Y_2) такую, чтобы вероятность их задержания не превышала бы Y , как бы ни действовали полицейские.

Соотношения для стратегий полицейских имеют вид

$$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 = 1;$$

$$0,51 X_1 + 0,64 X_2 + 0,49 X_3 + 0,58 X_4 + 0,37 X_5 + 0,46 X_6 \geq Y$$

(полицейские гарантируют себе хотя бы Y), если вори приходит только в А)

$$0,75 X_1 + 0,36 X_2 + 0,91 X_3 + 0,6 X_4 + 0,85 X_5 + 0,76 X_6 \geq Y$$

(полицейские гарантируют себе также Y), если вори приходит только в В). Соотношения для воров имеют вид

$$Y_1 + Y_2 = 1;$$

$$0,51 Y_1 + 0,75 Y_2 \leq Y;$$

$$0,64 Y_1 + 0,36 Y_2 \leq Y;$$

$$0,49 Y_1 + 0,91 Y_2 \leq Y;$$

$$0,58 Y_1 + 0,6 Y_2 \leq Y;$$

$$0,37 Y_1 + 0,85 Y_2 \leq Y;$$

$$0,46 Y_1 + 0,76 Y_2 \leq Y$$

Подставив $Y_2 = 1 - Y_1$ во все неравенства стратегия воров и рассматривая их как уравнения, получим

$$Y = 0,75 - 0,24 Y_1 \quad (1)$$

$$Y = 0,36 + 0,28 Y_1 \quad (2)$$

$$Y = 0,91 - 0,72 Y_1 \quad (3)$$

$$Y = 0,6 - 0,02 Y_1 \quad (4)$$

$$Y = 0,85 - 0,48 Y_1 \quad (5)$$

$$Y = 0,76 - 0,30 Y_1 \quad (6)$$

$$0 \leq Y \leq 1$$

Начертим эти шесть прямых в системе координат Y, Y_1 , указанной на рис. 6.10.

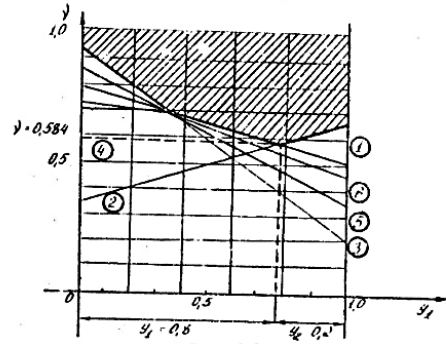


Рис. 6.10

Каждая точка заштрихованной области определяет возможное решение, ибо она удовлетворяет теперь всем рассмотренным неравенствам для воров. Из рис. 6.10 ясно, что наименьшая величина Y находится на пересечении прямых (2) и (4), ибо прямая (4) имеет отрицательный наклон. Итак, имеем

$$0,36 + 0,28 Y_1 = 0,60 - 0,02 Y_1;$$

откуда $Y_1 = 0,8$ и $Y_2 = 0,2$. Таким образом, выбирая стратегию $Y_1 = 0,8$ и $Y_2 = 0,2$, вор имеет минимальную вероятность задержания, равную $Y = 0,584$. Положим, $X_1 = X_3 = X_5 = X_6 = 0$, подставив значение

$Y = 0,584$ из выражения для полицейских находим $X_2 = \frac{1}{15}$ и $X_4 = \frac{7}{15}$. В итоге, если оба полицейские проводят $\frac{1}{15}$ своего времени вместе в А, а $\frac{14}{15}$ времени один из них проводит в Т, а другой в А, то вероятность поймать вора будет максимальной и равной $0,584$, каков бы стратегии ни выбрали вори.

7. МОДЕЛИ МАССОВОГО ОБСЛУЖИВАНИЯ

Под системой массового обслуживания (СМО) понимается время-логическая система, представляющая собой единство:

- входящего потока заявок или требований на обслуживание;
- параметров обслуживающих приборов или узлов;
- дисциплины обслуживания заявок, находящихся в системе.

Основные элементы СМО показаны на рис. 7.1.

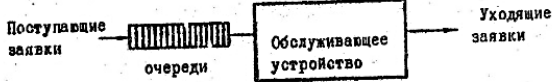


Рис. 7.1

В простейшем случае СМО может состоять из одного обслуживающего прибора, на вход которого поступает единственный поток заявок извне. Заявка, поступающая в прибор, проходит в системе единственную фазу обслуживания, после чего покидает систему. Простейшими дисциплинами обслуживания являются две предельные дисциплины. В первой из них заявка, поступившая на вход системы, терпит, если она застала обслуживающий прибор занятым. Вторая дисциплина предполагает, что на входе обслуживающего прибора имеется накопитель бесконечной емкости, в котором образуется очередь заявок, заставших обслуживающий прибор занятым. Для окончательного определения дисциплины обслуживания во втором случае необходимо указать порядок, по которому обслуживающий прибор выбирает из очереди следующую заявку для обслуживания.

7.1. Поток заявок

В основу описания потока заявок можно положить время τ между двумя соседними моментами времени поступления заявок в систему. Если при этом заявки поступают поодиночке, то поток заявок считается ординарным. Обычно предполагают, что τ является стационарной случайной величиной, заданной одним из своих законов распределения. Момент поступления заявок называют моментами восстановления, а τ - интервалом восстановления. Стационарный ординарный поток заявок характеризуется величиной, обратной среднему интервалу восстанов-

ления, называемому интенсивностью потока.

$$m = \frac{1}{\lambda} = \int_0^{\infty} t f(t) dt$$

Если в стационарном потоке отсутствует последствие, мы приходим к простейшему или пуассоновскому потоку, целиком определяемому своей интенсивностью

$$f(t) = \lambda \cdot e^{-\lambda t}$$

$$P(\tau < t) = F(t) = 1 - e^{-\lambda t}$$

Отсутствие последствия означает, что распределение времени до ближайшего восстановления не зависит от времени наблюдения. Свое название пуассоновские потоки получили в силу того, что распределение случайной величины N количества восстановлений в произвольном интервале времени подчиняется закону Пуассона:

$$P(n, t) = \frac{(\lambda t)^n e^{-\lambda t}}{n!}$$

Для простейшего потока среднее квадратичное отклонение длины интервалов восстановления от их среднего значения с их средним значением: $\sigma = 1/\lambda = m$. Это условие может быть критерием возможной замены реального распределения интервалов восстановления пуассоновским.

7.2. Обслуживающие приборы. Фазы обслуживания

Основной характеристикой обслуживающего прибора в СМО, если он одноканален, является время, затрачиваемое им на одно обслуживание заявки (одну фазу обслуживания). В общем случае время обслуживания заявки прибором определяется видом заявок, интенсивностью их поступления и порядком следования. Вид распределения интервалов восстановления определяет название потока заявок. Если такому же распределению подчиняется время обслуживания заявки, название, относящееся к потоку, дается и обслуживанию. Для определения характеристик прибора дополнительно к временным характеристикам каждого канала необходимо задать количество каналов и дисциплину включения каналов на обслуживание. Заявка за время пребывания в системе может пройти обслуживание в нескольких приборах или на нескольких фазах в каждом приборе. Систему, в которой все или некоторые заявки требуют нескольких фаз обслуживания, называют многофазной системой массового обслуживания. Для каждой заявки последовательность

фаза обслуживания считается определенной и называется путем обслуживания заявки. Фаза, после которой заявка покидает систему (обслуживается до конца), называется конечной фазой для данной заявки. Все остальные фазы называются промежуточными. Если для обслуживания заявки на разных фазах в системе имеется несколько узлов, систему называют многоузловой. Обслуживающий прибор, через который проходят заявки на различных фазах обслуживания, называют многоходовым. Многоходовый прибор в качестве одной из характеристик должен включать в свое описание дисциплину приоритета входов - какому входу отдать предпочтение, если заявки имеются на двух или более входах одновременно.

Выбор оптимального количества каналов многоканальной узла и установление наиболее рациональной дисциплины приоритета является задачей, которые приходится решать путем анализа соответствующих моделей СМО. В многофазной СМО в процессе обслуживания перед каждой фазой могут образовываться очереди заявок.

7.3. Показатели эффективности СМО

Прежде всего исследователя интересуют характеристики производительности системы. К ним относится среднее время, затрачиваемое системой на обслуживание одной заявки.

Для потребителя не менее важное значение имеет случайная величина времени пребывания в системе и время, затрачиваемое системой на обслуживание одной заявки.

Под эффективностью обслуживающей системы понимают характеристику уровня выполнения этой системой функций, для которой она предназначена.

1. Наиболее часто используются следующие показатели.

1° Вероятность потери требования в СМО. При оценке эффективности ПВО она характеризует вероятность прерыва воздушных целей к объекту, т.е. вероятность того, что все n приборов заняты обслуживанием P_n ($P_{ок}$).

2° Вероятность того, что обслуживания заняты именно k приборов P_k ; P_0 - вероятность того, что все приборы свободны.

3° Среднее число занятых приборов $N_3 = \sum_{k=1}^n k \cdot P_k$ - характеризует степень загрузки системы.

4° Среднее число приборов, свободных от обслуживания.

$$N_0 = \sum_{k=0}^n (n-k) P_k$$

5° Производным от этого показателя является коэффициент простоя приборов $K_n = \frac{N_0}{n}$.

6° Коэффициент загрузки приборов $K_3 = \frac{N_3}{n}$.

Для систем с ожиданием используют дополнительные характеристики:

а) Среднее время ожидания требований

$$T \cdot M(t_{ож}) = \int_0^{\infty} t dP_1(t_{ож} > t);$$

где

$$P_1(t_{ож} > t) = \sum_{k=0}^n P_k P_k(t_{ож} > t)$$

$P_k(t_{ож} > t)$ - условная вероятность того, что время ожидания при условии, что в момент поступления требования в систему в ней уже обслуживалось k требований.

б) Вероятность того, что время пребывания требования в очереди не превысит больше определенной величины t :

$$P_2[t_{ож} < t] = \sum_{k=0}^n P_k \cdot P_k(t_{ож} < t);$$

в) Средняя длина очереди

$$M_{оч} = \sum_{k=1}^n (k-n) P_k; \quad n \geq 1;$$

г) Среднее число заявок в сфере обслуживания

$$M = \sum_{k=1}^n k \cdot P_k = M_{оч} + N_3;$$

д) Вероятность того, что число требований в очереди, ожидающих начала обслуживания, больше некоторого числа

$$P_{>n} = \sum_{k=n+1}^{\infty} P_k$$

Иногда используются и ряд стоимостных показателей:

$q_{обсл}$ - стоимость обслуживания каждого требования в СМО;

$q_{ож}$ - стоимость потерь, связанных ожиданием в единицу времени;

$q_у$ - стоимость убытков, связанных с уходом неослуженных требований из СМО;

q_k - стоимость эксплуатации каждого прибора в СМО в единицу времени;

$q_{пр}$ - стоимость единицы времени простоя.

При выборе оптимальных параметров системы по экономическим

показателям иногда используют следующие функции стоимости потерь:

а) для систем с ожиданием

$$C_n = (q_{ож} M_{ож} + q_{пк} N_0 + q_{к} N_3) T,$$

где T - интервал времени исследования;

б) для систем с отказами

$$C_n = (q_{к} N_3 + q_{р} P_n R) T;$$

в) для смешанных систем

$$C_n = (q_{пк} N_0 + q_{ож} M_{ож} + q_{р} P_n R + q_{к} N_3) T;$$

Между приведенными характеристиками СМО существуют следующие взаимосвязи:

$$N_3 + N_0 = M; \quad M = M_{ож} + M_{со};$$

где M - число требований в системе; N_3 , $M_{со}$;

$$M = \sum_{k=1}^{\infty} k \cdot P_k;$$

$$t_{ож} + t_{со} = t_{проб};$$

$$\lambda t_{ож} = M_{ож};$$

$$\lambda t_{проб} = M;$$

- суммарная интенсивность выходящего потока для систем с отказами $\lambda = \lambda_{вых}$.

7.1. Двухфазная модель вычислительной системы

Одной из простейших моделей мультипрограммной ВС с постоянным уровнем мультипрограммирования является замкнутая двухфазная СМО, представленная на рис. 7.2.

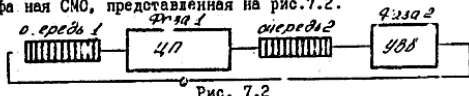


Рис. 7.2

Первое обслуживающее устройство интерпретируется как центральный процессор (ЦП), второе - как устройство ввода-вывода (УВВ), объединяющее канал и внешнюю память (ВП), а заявками являются m программ, одновременно находящихся в оперативной памяти ЭВМ. Предполагается, что в процессе выполнения каждой программы возникают запросы к ЦП, что соответствует перемещению заявки в рассматриваемой СМО из очереди I в очередь 2.

редь 2. После удовлетворения запроса к ВП программа готова к выполнению ЦП и этому соответствует перемещение заявки из очереди 2 в очередь I. Предполагается, что длительности обслуживания заявок независимы между собой, на каждой фазе в отдельности и на различных фазах и это обслуживание заявок происходит в порядке их поступления. Считаем, что входной поток является пуассоновским с параметром λ а длительность обслуживания заявок имеет экспоненциальное распределение с параметром μ . Постоянной мерой загруженности каждой фазы является нагрузка

$$\rho = \frac{\text{средняя длительность обслуживания}}{\text{среднее время между моментами поступления}} = \frac{1}{\mu} : \frac{1}{\lambda} = \frac{\lambda}{\mu}.$$

Вероятность того, что в системе находится n_i требований равна

$$P_{n_i} = \rho P_{n_i-1} = \rho^{n_i} \cdot P_0.$$

Если $\rho < 1$, то вероятность того, что в системе нет ни одной заявки будет равна $P_0 = 1 - \rho$.

Тогда, стационарная вероятность того, что в фазе находится равно n_i заявок, будет равна $P_{n_i} = (1 - \rho) \rho^{n_i}$; $n_i \geq 0$; мы получили геометрическое распределение и его среднее легко вычисляется.

$$\bar{N}_i = \sum_{n_i=0}^{\infty} n_i P_{n_i} = \rho / (1 - \rho).$$

Аналогичным образом можно рассчитать характеристики для второй фазы.

Поскольку распределение длительности обслуживания на каждой фазе является экспоненциальным, то в силу свойства отсутствия последствии замкнутой двухфазной СМО можно свести к типовой модели, у которой

$$P_n = \frac{(1 - \rho)^2 \rho^n}{1 - \rho^{n+1}}; \quad n = 0, 1, \dots, m.$$

Если n_1, n_2, \dots, m , то совместное стационарное распределение числа заявок в фазах I и 2 будет равно $P(n_1, n_2) = (1 - \rho)^2 \rho^{n_1 + n_2} / (1 - \rho^{n_1 + n_2 + 1})$.

$$\gamma = \frac{\text{средняя длительность обслуживания в фазе I}}{\text{средняя длительность обслуживания в фазе 2}}$$

7.5. Модель многодоступной вычислительной системы

Пусть в качестве ВС выступает обслуживающий прибор СМО, вход которого имеет k источников заявок. Каждый из этих источников соответствует терминалу пользователя ВС.

Модель многодоступной ВС представлена на рис. 7.3.

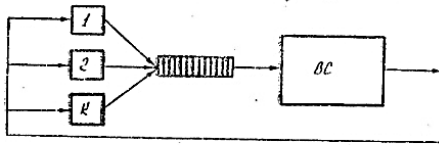


Рис. 7.3

Пусть λ - продуктивная способность, т.е. число заявок, содержащих обслуживание в единицу времени в рассматриваемой системе. Определим случайную величину U как время обслуживания пользователя, равное интервалу времени с момента завершения выполнения его предыдущего запроса к ВС до момента выдачи им следующего запроса к ВС, и время ответа T как интервал времени с момента выдачи запроса пользователем до момента завершения его выполнения системой. Время ответа и время обслуживания составляют цикл одного взаимодействия пользователя, среднее значение которого в установившемся режиме равно $\bar{T} \cdot \bar{U} = K / \lambda$.

Пусть P_0 - вероятность простоя обслуживаемого устройства в установившемся режиме и \bar{S} - средняя длительность обслуживания одного запроса. Тогда в установившемся режиме прибор ВС может обработать $\lambda \cdot (1 - P_0) / \bar{S}$ запросов в единицу времени. Таким образом, среднее время ответа системы будет равно:

$$\bar{T} = K \cdot \bar{S} / (\lambda \cdot (1 - P_0) \cdot \bar{U}),$$

Определим интенсивность обслуживания $\mu = 1 / \bar{S}$ и интенсивность поступления заявок $\nu = 1 / \bar{U}$. Тогда их соотношение

$$\chi = \mu \bar{U} / \nu = \frac{\text{среднее время обслуживания}}{\text{средняя длительность обслуживания}}$$

определяет новую характеристику.

Тогда нормированное среднее время ответа \bar{T} / \bar{S} равно

$$\mu \bar{T} = K / (\lambda \cdot (1 - P_0) - \chi).$$

Средняя длина очереди в установившемся режиме будет равна

$$\bar{N} = K - (1 - P_0) \chi.$$

Качество обслуживания отдельного терминального пользователя характеризуется средним временем ответа системы. Тогда нормированное среднее время ответа будет равно $U(N) = \frac{K}{1 - P_0(N)}$.

64

Для вычисления $U(N)$ удобно воспользоваться следующим рекуррентным соотношением:

$$U(N) = K - \frac{K(N-1)}{U(N-1) \cdot \chi};$$

$$U(1) = 1.$$

8. ВЕРоятностные модели сложных систем

8.1. Идея метода Монте-Карло

Когда аналитические методы неприменимы, например, при описании стохастических процессов, используются универсальный метод Монте-Карло.

Идея метода Монте-Карло состоит в следующем. Вместо того чтобы описывать случайный процесс с помощью дифференциальных уравнений, производится "розыгрыш" случайного явления с помощью какой-то процедуры, дающей случайный результат.

Так же как и в реальности, конкретная реализация случайного процесса каждый раз складывается по-разному. В результате получает множество реализаций случайного процесса, которое статистически обрабатывается и по нему определяют вероятностные характеристики результата: вероятности каких-то событий, математическое ожидание, дисперсии случайных величин и т.д.

Иногда такой прием оказывается проще, чем попытки построить аналогичную модель и на ней исследовать зависимость между условиями операции, принятым решением и результатом. Для этой цели имеется механизм формирования случайных чисел ξ с плотностью вероятности $\psi(x)$.

В задачах исследования операций метод Монте-Карло применяется при:

- 1) моделировании сложных операций, в которых присутствует большое число сложно взаимодействующих случайных факторов;
- 2) построение процедур применимости более простых аналитических методов и выявления условий, в которых ими можно пользоваться;
- 3) в целях выработки системы "эмпирических" формул для характеристик процесса.

Основным элементом, из совокупности которых складывается

статистическая модель, является одна случайная реализация моделируемого явления. Реализация представляется как бы один "экземпляр" случайного явления со всеми присущими ему свойствами. Разыгрывается эта реализация с помощью специально разработанной имитационной процедуры или алгоритма, в котором важную роль играет "бросание жребия".

Каждый раз, когда в ход моделируемого явления вмешивается случайность, её влияние учитывается не расчетом, а жребием.

Условием называть "единичным жребием" любой опыт со случайным исходом, в котором решается:

- 1) произошло или не произошло событие A;
- 2) какое из возможных событий A, A ... , A произошло;
- 3) какое значение приняла случайная величина X;
- 4) какую совокупность значений приняла система случайных величин X_1, X_2, \dots, X_n .

Каждая реализация случайного явления методом Монте-Карло строится из цепочки единичных жребиев, перемежающихся обычными расчетами. Этими расчетами учитывается влияние хода жребия на дальнейшее течение процесса. Учитываются также и условия розыгрыша следующего розыгрыша жребия. Существует стандартный механизм, с помощью которого можно осуществить любую из перечисленных разновидностей жребия: получение случайного числа, значения которого равновероятны на интервале [0, 1], обозначаемого в дальнейшем через ξ .

1° Произойдет ли событие A?

Вероятность события A равна P.

а) Розыгрыш ξ ;

б) Если $\xi < P$, то A = истина A = "ложь".

2° Какое из событий A_1, \dots, A_n произошло?

Пусть вероятности несовместимых событий A_1, \dots, A_n соответственно P_1, \dots, P_n , где

$$\sum_{i=1}^n P_i = 1$$

а) делим интервал (0, 1) на K участков соответственно длиной P_1, P_2, \dots, P_n .

б) Розыгрыш ξ ;

в) Если ξ попадет на i-й участок, то считаем, что произошла i-я событие (A_i = истина).

3° Какое значение приняла случайная величина.

Если X дискретна, т.е. X_1, X_2, \dots, X_n имеет вероятность P_1, P_2, \dots, P_n , то её можно "разыграть" аналогично предыдущей задаче ($X=X_1; X=X_2; \dots; X=X_n$).

Если X-непрерывна и имеет функцию плотности $f(x)$, то переводим к $F(x) = \int_{-\infty}^x f(x) dx$. Находят для F обратную ей функцию ψ .

Разыгрывается величина ξ и берут обратную функцию $X = \psi(\xi)$.

Графически эту процедуру изображают следующим образом

(рис. 8.1):

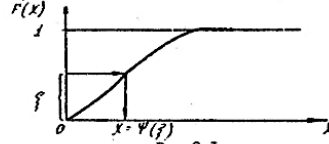


Рис. 8.1

Таким образом, $P(X < x) = P(\xi < F(x)) = F(x)$. Кривую $F(x)$ задают аналитически. Имеется и процедура обратная F (тоже в аналитическом виде).

Если аналитически $F(x)$ и её обратная $\psi(x)$ трудно предоставить, то её заменяют приближенной ломаной линией (аппроксимация ступенчатая или кусочно-линейная) и далее поступают аналогичным образом.

4° Какую совокупность приняла система случайных величин X_1, X_2, \dots, X_n ?

Если X_1, \dots, X_n независимы, то розыгрыш совокупности их значений сводится к K раз повторенному розыгрышу в Π 3°.

Если же они зависимы, то, разыграв одну из них, например, X_1 , надо разыгрывать вторую (X_2), согласно её условному закону распределения, с учетом того, что первая приняла именно то значение, которое было разыграно. Третью величину (X_3) нужно разыгрывать уже при двух условиях и т.д.

8.2. Моделирование псевдослучайных величин

Традиционный способ программной имитации случайных функций любой сложности сводится к генерированию базовых последовательностей и их последующему функциональному преобразованию. Чаще всего в качестве базовой используется последовательность чисел, представляющая собой реализацию равномерно

распределенных на интервале $(0,1)$ случайных величин $\xi_0, \xi_1, \dots, \xi_i$, т.е. к получению повторной выборки из равномерно распределенной генеральной совокупности значений величины ξ . Практически в силу дискретности представления числа в ЭВМ можно оперировать с $M=2^k$ различными возможными значениями чисел $X_i = \frac{x_i^*}{M}$, где x_i^* — реализация случайной величины ξ , имеющей равномерное распределение с вероятностями $P(\xi_i = X_i) = \frac{1}{M}$; $0 \leq i < M-1$; k — разрядность кода ЭВМ. В генераторах псевдослучайных чисел на ЭВМ фактически реализуются формулы вида

$$X_i^* = \varphi(X_{i-1}^*, \dots, X_{i-2}^*); \text{ mod } M$$

$$X_i = X_i^* / M.$$

Реализации X_i получаются внутри алгоритма путем подобранного преобразования φ из ранее имитированных случайных чисел. Здесь X_i^* представляет собой числа больше единицы. Поэтому необходима нормировка их величиной M , определяемой разрядной сеткой ($M=2^k$, k — разрядность слова ЭВМ), для которой реализован алгоритм генератора.

Наиболее распространен конгруэнтный метод генерации псевдослучайных чисел:

$$X_i^* = A X_{i-1}^* + C; \text{ mod } M$$

$$X_i = X_i^* / M.$$

Согласно [6] работу такого алгоритма можно представить на итерационной плоскости (X_{i-1}, X_i) с помощью диаграммы для случая $A=2$ и $C=0$. Как видно из рис. 8.2, эволюция различных начальных значений X_0 можно получить различные режимы работы алгоритма.

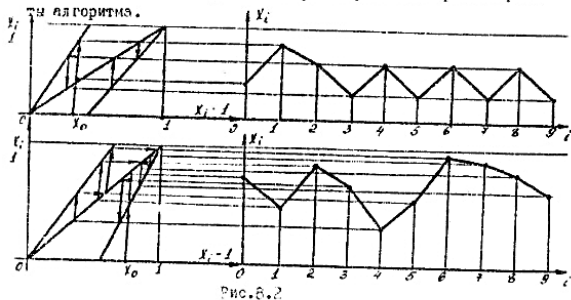


Рис. 8.2

переходный режим и позволяет получать псевдослучайные последовательности. Можно ожидать, что при соответствующем подборе параметров A, C и X_0 этот режим окажется достаточно длительным, а генерируемая последовательность $\{X_i\}$ будет иметь стохастические свойства. Кроме рассматриваемого метода, существует большое число алгоритмов имитации псевдослучайных чисел с равномерным распределением на интервале $(0,1)$.

Наиболее ранним является метод серединных квадратов, суть которого заключается в следующем: берут произвольное m разрядное число X_0 ; возводят число в квадрат; из результата выбирают среднее m цифр и получают новое число X_1 , далее процедура повторяется при начальном значении X_1 и т.д.

Смешанный конгруэнтный метод предполагает двойное обращение к генератору, формирующему псевдослучайные числа конгруэнтным способом, и последующим преобразованием по формуле

$$X_{i+1} = (X_i + X_{i-1}) \text{ mod } M.$$

Для увеличения периода аperiodичности псевдослучайных чисел используется метод возмущения основного алгоритма.

Суть метода заключается в следующем. Основная программа имитирует числа по алгоритму $X_{i+1} = \varphi(X_i)$, где в качестве начальных значений задаются числа, формируемые по второму алгоритму $X_{i+1} = \psi(X_i)$. Более подробно об алгоритмах формирования псевдослучайных чисел, равномерно распределенных на интервале $(0,1)$, можно ознакомиться в работах [4, 6].

Реализация остальных видов случайных воздействий получается путем функциональных преобразований элементов ξ . Так, имитация нормально распределенных величин N может быть проведена множеством способов. Мы рассмотрим два наиболее употребительных алгоритма имитации нормально распределенных величин.

Первый из них основан на центральной предельной теореме теории вероятностей. Согласно этой теореме, при сложении достаточно большого числа независимых случайных величин, суммированных по своим дисперсиям, получается случайная величина, распределенная приблизительно по нормальному закону; причем этот закон тем ближе к нормальному, чем больше случайных величин складывается. Сказывается, что при сложении не менее пяти случайных величин с равномерным распределением

в интервале $(0,1)$ получается случайная величина, которая с точностью, достаточной для большинства практических задач, может считаться нормальной. Отсюда и вытекает способ (сумма), согласно которому необходимо выполнить преобразования двенадцати значений ξ_i .

$$q = \frac{\sqrt{12}}{2} \left[\sum_{i=1}^{12} \xi_i - 6 \right];$$

В ряде случаев ограничиваются n обращениями к базовому генератору псевдослучайных чисел. Тогда для обеспечения требуемой близости к нормальному распределению необходимо осуществлять поправку согласно следующей формуле:

$$q^* = q - \frac{1}{20n} [3q - q^3];$$

где q определяется по предыдущей формуле.

Второй алгоритм [4] является более экономичным, поскольку позволяет по двум обращениям к базовому генератору, формирующим значения ξ_1 и ξ_2 , вычислить две случайные величины q_1 и q_2 , имеющих нормированное нормальное распределение по следующим формулам:

$$q_1 = \sqrt{2 \ln \xi_2} \cos 2\pi \xi_1;$$

$$q_2 = \sqrt{2 \ln \xi_2} \sin 2\pi \xi_1.$$

Более подробно об алгоритмах имитации псевдослучайных величин q , распределенных по нормальному закону распределения, можно ознакомиться в работах [4, 6].

Экспоненциальное распределение весьма широко используется при моделировании больших систем. Наиболее распространены следующие два способа имитации псевдослучайных величин экспоненциально распределенных с функцией плотности $f(x) = \lambda e^{-\lambda x}$ ($x > 0$).

Первый из них основан на методе обратной функции $F(x) = 1 - e^{-\lambda x}$. Обратной по отношению к F будет функция $e^{-\lambda x} = 1 - F$. Тогда $x = -\frac{1}{\lambda} \ln(1 - \xi) = -\frac{1}{\lambda} \ln \xi$.

Второй способ основан на методе композиции. Задавшиеся величинами τ и θ , моделирует с помощью базового генератора θ штук равномерно распределенных величин. Затем осуществляют перерасчет по формуле $x = \tau - \max(\xi_1, \xi_2, \xi_3, \dots, \xi_\theta)$.

Обычно задают $\tau = 4$ и $\theta = 4$ и получают последовательность чи-

сел x , распределенных по экспоненте с точностью, достаточной для практики моделирования больших систем.

В ряде случаев целочисленная случайная величина характеризует заданный процесс случайных испытаний и может имитироваться поэтому путем прямого воспроизведения и анализа этого процесса. Многие дискретные распределения связаны со схемой независимых испытаний и легко моделируются путем имитации последовательности событий $A_i = (\xi_i < p)$; $i = 1, n$.

Выборка из биномиального распределения получается прямым подсчетом числа единиц в рядах случайного кода $q^* = 0, q_1^*, q_2^*, \dots, q_n^*$.

Для имитации случайных величин y , подчиняющихся геометрическому распределению.

$$p = (y - 1) \cdot p(1 - p)^{y-2}; \quad y = 1, 2, \dots$$

Определяют номер испытания, принесшего первый успех (истинности события A_i).

При моделировании больших систем весьма распространено использование дискретных величин, распределенных по закону Пуассона. Обычно используют следующий алгоритм "розыгрыша" псевдослучайных величин, распределенных по закону Пуассона с параметром λ : $P(k) = \frac{\lambda^k}{k!} e^{-\lambda}$. Последовательно моделирует значения ξ_i до тех пор, пока не будет выполняться следующее соотношение: $\prod_{i=1}^n \xi_i > e^{-\lambda} \prod_{i=1}^n \xi_i$. Тогда величина $\sum_{i=1}^n \xi_i$

будет иметь распределение Пуассона с параметром λ .

8.3. Модель поражения ракетми площадной цели

Рассмотрим задачу о поражении ракетми "площадной" сложной цели. Зона разрушений одной ракеты представляет собой круг радиуса r . В результате n выстрелов будет поражена какая-то часть общей площади цели S_U , составляющая какой-то доле полной площади цели:

$$U = \frac{S_n}{S_U};$$

Чтобы избежать ненужных перекрытий зон поражения прицеливание производится n ракетами по n различным точкам прицеливания O_1, O_2, \dots, O_n .

Заданы характеристики рассеивания ракет: средние квадратические отклонения σ_x и σ_y .

Систематические сдвиги отсутствуют, координаты X и Y

каждой точки попадания независим друг от друга и от координат других точек попадания.

Требуется: при заданном расположении точек прицеливания O_1, O_2, \dots, O_n вычислить следующие характеристики эффективности операции:

- средние доли пораженной площади цели: $MU = M[U]$;
- дисперсия доли $Du = D[U]$;
- вероятность того, что будет поражено не менее заданной доли U площади цели $P(U > U)$;
- математическое ожидание Z_0 числа ракет, которые не причинили цели никакого ущерба (попали мимо).

Решение. Если не делать никаких упрощающих предположений о форме цели и зоны поражения, аналитическое решение поставленной задачи чрезвычайно сложно и практически невозможно. Проще будет решить её методом Монте-Карло.

Каждая реализация будет представлять собой "обстрел" цели ракетами, в котором точки попадания ракет будут разиграны по вероятности. Моделирование каждой реализации будет состоять из единичных взрывов, плюс расчет пораженной цели S_n . В каждом единичном взрыве разигрывается точка попадания одной j -й ракеты, т.е. две случайные величины $(X_j; Y_j)$, распределенные по нормальному закону с характеристиками:

$$Mx_j, My_j, \sigma_x, \sigma_y.$$

Здесь Mx_j и My_j - координаты точки O_j . Предполагаем, что коэффициент корреляции $Z_{xy} = 0$, т.е. по условию X_j и Y_j независимы.

Моделирование будем проводить по способу сумм для "блуждающих" пары нормальных величин $(X_j; Y_j)$. Координаты j -й точки попадания разигрываются по формулам

$$X_j = \sigma_x \sqrt{2} \left(\sum_{k=1}^N \xi_k - 3 \right) + Mx_j;$$

$$Y_j = \sigma_y \sqrt{2} \left(\sum_{k=1}^N \xi_k - 3 \right) + My_j.$$

где ξ_k - 12 независимых экземпляров случайной величины ξ , равномерно распределенных на интервале $(0, 1)$.

Предположим, что этот этап единичной имитации выполнен и мы получили N точек попадания в данной реализации (рис. 8.3)

Теперь нужно подсчитать пораженную площадь S_n для данной (i -й) реализации. Для этого вокруг каждой точки опишем круг радиуса r и подсчитаем площадь той части цели, которая покрыта хотя бы одним из кругов. Эта часть разбивается на ряд

элементарных площадок dS .

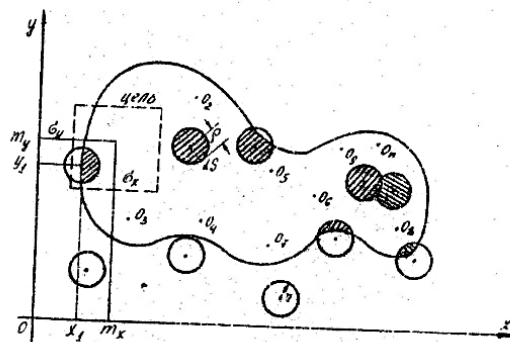


Рис. 8.3

Для каждой площадки определяется расстояние r_j от точки попадания j -й ракеты ($j=1, n$). Если хотя бы для одного из точек попадания это расстояние оказалось меньше r , то площадка dS считается пораженной, после чего производится суммирование пораженных площадок dS_n по всей цели; т.е. $S_n = \sum dS_n$.

Для i -й реализации определяет долю пораженной площади в данной реализации

$$U^{(i)} = S_n^{(i)} / S_c.$$

Одновременно с величиной $U^{(i)}$ для каждой i -й реализации числяем $Z^{(i)}$ - количество ракет, расстояние от точек попадания которых до цели превышает r (число п. омахов). Аналогичным образом проводат N реализаций, вычисляем требуемые характеристики:

$$Mu = \frac{1}{N} \sum_{i=1}^N U^{(i)};$$

$$Du = \frac{1}{N} \sum_{i=1}^N (U^{(i)})^2 - Mu^2.$$

Вероятность того, что доля пораженной площади будет не меньше U , определяется следующим образом:

в каждой реализации связывается число

$$Z_i = \begin{cases} 1 & \text{при } U^i \geq \bar{U} \\ 0 & \text{при } U^i < \bar{U} \end{cases}$$

тогда $P(U \geq \bar{U}) = \frac{1}{N} \sum_{i=1}^N Z_i$.

Математическое ожидание Z_0 числа ракет, не причинявших ущерба цели, определяется по формуле

$$\bar{Z}_0 = \frac{1}{N} \sum_{i=1}^N Z_0^{(i)}$$

При моделировании методом Монте-Карло львиную долю времени занимает само моделирование реализаций и ничтожную долю — обработка результатов. Поэтому стараются зафиксировать максимальное число сведений о каждой реализации, не ограничиваясь подсчетом одного единственного пока затея эффективности. Особую роль при этом приобретает качество генераторов моделирования псевдослучайных величин.

9. ЛИТЕРАТУРА

1. Вагнер Г. Основы исследования операций.-М.: Сов.радио, 1972-1973, т.1-3 329с., 480с., 561с.
2. Вентцель Е.С. Исследование операций.-М.: Сов.радио, 1979, 575с.
3. Раскин А.Г. Анализ сложных систем и элементы теории оптимального управления.-М.: Сов.радио, 1976, 343с.
4. Ермаков С.М., Михайлов Г.А. Курс статистического моделирования.-М.: Наука, 1976, 219с.
5. Зайченко Ю.П. Исследование операций.-Киев: Высшая школа, 1975, 390с.
6. Поляк Ю.Г. Вероятностное моделирование на ЭВМ.-М.: Сов.радио, 1971, 399с.
7. Конвейн В.В. Теория расписаний.-М.: Наука, 1975, 215с.
8. Форд Л. Потoki в сетях.-М.: Мир, 1966, 320с.

СОДЕРЖАНИЕ

I. ВВЕДЕНИЕ	3
I.1. Отличительные особенности ИСО	3
I.2. Типы задач ИСО	4
I.3. Классы операционных задач	5
I.4. Содержание задач ИСО	7
2. МОДЕЛИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ	8
2.1. Общая формулировка линейных задач ИСО	10
2.2. Геометрическая интерпретация линейных моделей ..	10
2.3. Решение линейных моделей	12
2.4. Примеры линейных моделей	13
2.4.1. Транспортная задача линейного программирования	13
2.4.2. Задача распределения ресурсов	14
2.4.3. Задача оптимального выбора маршрута	15
2.4.4. Задача рационального составления смеси	17
2.4.5. Модель назначения	18
2.4.6. Модель выбора кратчайшего пути	18
3. МОДЕЛИ НА ГРАФАХ	20
3.1. Задача о кратчайшем пути между вершинами связанного неориентированного графа	20
3.1.1. Ребра графа единичной длины	20
3.1.2. Ребра графа произвольной длины	21
3.2. Построение графа наименьшей длины	22
3.3. Задача размещения производства в сетевой постановке	23
3.3.1. Задача о наибольшем потоке	24
3.3.2. Теорема Форда-Фалкерсона	24
3.3.3. Пример задачи о наибольшем потоке	26
3.4. Задача об оптимальном назначении	30
4. МОДЕЛИ СЕТЕВОГО ПЛАНИРОВАНИЯ	31
4.1. Расчет и анализ параметров сетевого графика при фиксированном t_{ij}	35
4.2. Расчет сетевого графика при случайном времени выполнения работ	38
4.3. Оптимизация плана комплекса работ	40
5. МОДЕЛИ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ	42
5.1. Модель управления запасами	43

РЕПОЗИТОРИЙ ГГУ

5.2. Задача о наборе высоты и скорости самолетом..	44
6. ИГРОВЫЕ МОДЕЛИ	47
6.1. Принцип минимакса	50
6.2. Геометрическая интерпретация игровых моделей.	51
6.3. Общие методы решения конечных игр	54
6.4. Задачи "Полицейские и воры"	55
7. МОДЕЛИ МАССОВОГО ОБСЛУЖИВАНИЯ	58
7.1. Потоки заявок	58
7.2. Обслуживающие приборы	59
Фазы обслуживания	59
7.3. Показатели эффективности СМО	60
7.4. Двухфазная модель вычислительной системы ...	62
7.5. Модель многодоступной вычислительной системы	63
8. ВЕРОЯТНОСТНЫЕ МОДЕЛИ СЛОЖНЫХ СИСТЕМ	65
8.1. Идея метода Монте-Карло	65
8.2. Моделирование псевдослучайных величин	67
8.3. Модель поражения ракетами "площадной" цели ...	71
9. ЛИТЕРАТУРА	74

Иван Васильевич Максимей
 Модели задач исследования операций
 (учебное пособие)

Редактор Е.Ф. Заяцева

Подписано к печати 10.10. 1984 г. АЗ № 41649. Форм. 60x84 1/16.

Бумага писчая № 1. Печать офсетная. Усл. п. л. 1,2.

Уч.- изд. л. 3,7. Тираж 200. Заказ 320. Цена 12 к.

Отпечатано на ротационной ГТУ г.Гомель, ул.Советская,104.

