1. Предмет ИСО, основные определения

предмет исо был создан для решения проблем создания базы моделей и методов как основы систем поддержки принятия решений (DSS - decision support system). также можно сказать что исо - научный подход к решению задач организационного управления.

такая необходимость возникла из за усложнения нашего мира: решения должны приниматься быстро и быть максимально корректными, в них также тяжело просчитать какие либо случайные отклонения и практически невозможно провести предварительный эксперимент, и любому руководителю уже недостаточно просто опыта, знаний и интуиции. (*короче слишком много факторов которые надо учитывать а человеку это тяжело и нужна помощь компуктера/компьюнтара*) именно для этого возникли системы DSS, а следовательно и задача их создания и управления.

для определения исо необходимо обозначить некоторые ключевые моменты

1. установление объекта исследования - <u>решение в сложных системах</u>
   напр: выбор плана производства, расстановка сил, осуществляемые действия
   это все объединятся тем, что последствия принимаемых решений не могут быть достоверно предсказаны
2. подчеркивание целенаправленности действий
   в силу того что исследуется <u>сложная система</u>, необходимо четко ставить цель и стремиться достичь ее с помощью своих решений (понятия сложная система и цель неотделимы)
3. определение метода исследований
   модели и методы в исо в первую очередь включают в себя добавление цели управления. это помогает не только принимать оптимальные решения, но и в каком то смысле улучшает модель (*типа не все модели учитывают незначительные факторы а их добавление развивает саму науку*)

стоит упомянуть что основой исо также является кибернетика - наука об управлении в сложных динамических системах. <u>сложная система </u>- совокупность материальных объектов, объединенных какими-либо связями и предназначенных для достижения целей. => единство понятий системы: элементы, связи, операции

2. **Основные типы моделей. Приемы и принципы моделирования.**

моделирование - замена исследуемой системы аналогичной системой, которая называется модель, и проведение экспериментов с ней

- **аналоговые модели**
  одни носители свойств заменяются на другие физические носители (выбирается исследователем)
  пр: моделирование транспортных потоков с помощью гидравлических сетей или биологических систем с помощью электрических схем
- **аналитические модели**
  замена материальных носителей основных свойств системы на абстрактные, математические символы и различные соотношения между ними
  пр: аналитической моделью колебания маятника является определенное дифференциальное уравнение; график или математическая функция
  - чем управлять? - математический объект
  - при каких ограничениях управлять? - взаимосвязи этих объектов и ресурсов системы
  - для чего управлять? - критерий сравнения решений и выбор наилучшего
- **имитационные модели**
  алгоритм, реализующий поведение системы в зависимости от решения
  этапы:
  - подготовительный (постановка задачи, выбор критериев, ограничений, подготовка данных)
  - рабочий (моделирование путем генерирования случайных управляющих решений)
  - заключительный (анализ и интеграция результатов)

  важная составляющая часть ИМ - диалоговый режим: обмен информацией, использование опыта и интуиции, контроль
  недостатки ИМ:
  - ограниченная точность моделирования
  - отсутствие общности результатов
  - высокая стоимость и продолжительность разработок моделей
- **ситуационные модели**
  предсказание процессов функционирования системы в виде последовательности отдельных ситуаций в определенные моменты времени (*совокупность ситуаций - ситуационная модель*)
  недостатки:
  - статичность
  - невозможность учета альтернативных вариантов поведения

деловая игра - объединение ИМ и СМ (чаще всего используется для обучения)

принципы моделирования:

- целенаправленность моделей: модель соответствует цели, ведь не для каждой цели подойдет одна и та же модель
- нейтральность модели: процессы в модели не воздействуют на функционирование системы
- согласование точности моделирования: нет смысла описывать одни элементы точно, другие с большими погрешностями

приемы моделирования:

- формализация связи: необходимо попытаться выразить и исследовать явную зависимость между целью и параметрами управления
- упрощение: целесообразно сложную систему заменить на аналогичную простую
- идентификация модели: определить неизвестные параметры с помощью доп экспериментов
- восстановление структуры по наблюдениям: если система неизвестной структуры, то построить гипотетическую (анализ данных, которые описывают ее функционирование)
- имитация с эволюцией: если нельзя применить приемы выше, то строится ИМ и СМ с их последующей эволюцией (усложнением моделей)

3. Типовые этапы операционных исследований.

основной цикл научного метода:

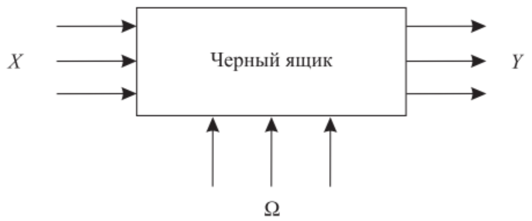
- индукция: переход от наблюдений к теории
- дедукция: переход от теории к логике и математике
- верификация: испытание выводов

постановка задачи и разработка концептуальной модели

- правильная постановка задачи - половина решения
- первоначально задачу ставит заказчик
- проведение исследований - операционная группа (системные аналитиков, спецы, психологи и тд)
- подробный анализ данных -> постановка цели и определение структуры системы
- проведение консультаций для уточнения задачи
- результат: концептуальная модель системы (состав, компоненты, факторы, стратегии управления) (*другими словами содержательная постановка задачи*)

построение модели (формализация задачи)

общая модель построения - черный ящик - устройство, в котором можно влиять на входы и наблюдать выходы



- входы(X, Ω - внешние факторы) - описание управляющих воздействий на систему
- выходы(Y) - результаты этих воздействий
- описание черного ящика - установление соответствиями между входами и выходами в зависимости от внешних факторов

(*принцип черного ящика позволяет определить реакцию системы на входы X по наблюдению выходов Y*)

Пути определения зависимости (модель системы):

1. внутренний путь (проникновение внутрь системы + анализ взаимодействий)
   - *то есть нахождение каких то часто встречающихся приколов в системе -> построить типовую модель*
2. внешний путь (наблюдение за входами и выходами в процессе функционирования)

построение модели операции:

f(Y) - ф-я полезности, отображение выходных параметров на мн-во действ. чисел (*ф-ия f - характеристика качества принимаемого решения*)

сбор необходимой входной информации (*так как все решения принимаются на основе информации*)

- все входные значения должны быть научно обоснованы и актуальны

выбор метода и алгоритма решения (в зависимости от типа модели и неопределенности входной информации)

модели условной оптимизации: (выработка программы действий в определенной практической ситуации). математическим языком: требуется определить числовые значения компонента вектора X, доставляющие максимум (минимум) функции f(X) при условиях X > 0, $g_i(X) > 0$, i = 0, 1, …, m

- линейная
- нелинейная
- целочисленная

комбинаторная модель: конечное, но огромное множество допустимых решений (комбинаторные объекты - перестановки элементов множества N = {1, 2, …, n})

графовая модель: задачи поиска определенных подграфов с минимальной (максимальной) суммой весов

модель массового обслуживания: вычисляются только параметры возникающих очередей на обслуживание при различных законах распределения случайных величин
модель управления запасами: -
модель конфликтной ситуации: решение принимает две (или более стороны); изучает теория игр
принятие решений:

- статические задачи - информация не изменяется
- динамические задачи - информация изменяется
- детерминированная модель - все параметры модели однозначно определены
- вероятностные модели - менее точное описание реальных условий

проверка адекватности и корректировка модели (*так как в сложных системах модель упрощенно описывает реальный объект*)
сравнение выходных характеристик модели Y_M с фактическими характеристиками объекта Y_O при изменении входных данных (X) и внешних факторов (Ω)
мера адекватоности - мера близости между Y_M и Y_O:

- абсолютное отклонение: $\varepsilon = Y_M - Y_O \le \varepsilon_{доп}$
- относительное отклонение: $\delta = |(Y_M - Y_O) / Y_O| * 100\% \le \delta_{доп}$
- вероятностная оценка: $P\{\delta \le \delta_{доп}\} \ge P_0$
  где $\varepsilon_{доп}$ и $\delta_{доп}$ - допустимое отклонение, которое определяет заданную степень адекватности модели; P_0 - заданная величина вероятности выполнения неравенства $\delta \le \delta_{доп}$.
  Если величина ε превышает $\varepsilon_{доп}$, или δ превышает $\delta_{доп}$, или $P < P_0$, то это свидетельствует о том, что упущены некоторые важные факторы и модель требует корректировки.

корректировка (доп исследования; может повторяться многократно)

- расширение набора внешних факторов \ управляющих переменных \ выходных характеристик модели
- переход от линейных зависимостей к нелинейным или повышению степени нелинейности
- расширение набора ограничений или их комбинаций

поиск решения на модели

- аналитическое
- численное
- алгоритмическое

важно: проверка стабильности решения - оценка влияния небольших изменений входных данных
реализация найденного решения на практике
полученную на модели оптимальную стратегию управления необходимо представить в виде соответствующих инструкций и правил (что и как делать), которые были бы понятны административному персоналу данной организации и реализовались в производственных условиях

4. Экспертные оценки в принятии решений. Метод Дельфи.

Необходимо организовать процесс принятия решений так как построение математический моделей неотделимо от средств ее исследования.
Важную роль в этом отводят интуиции, опыту и знаниям исследователей системы - экспертам
цель - получение объективно обоснованной информации
<u>схема работы</u> с **экспертом(**прибор**)**:

- предъявляется гипотеза H, и эксперт участвует в выборе множества признаков E, характеризующих эту гипотезу;
- на основании признаков E вычисляется некоторая оценка P(H | E)
- эксперт определяет интервал, в котором может находиться оценка P(H | E).

<u>требования оценок</u>:

- относительная стабильность во времени при неизменности множества E;
- возможность обобщения серии индивидуальных оценок при работе с группой экспертов;
- независимость оценок, полученных разными экспертами.

метод Дельфи - метод научного и технического прогнозирования для получения и обработки оценок
<u>этапы</u>:

1. формируется руководящая группа, ответственная за сбор и обработку выводов;
2. определяется количество и состав группы экспертов;
3. определяется общий показатель мнения группы и показатель согласованности ;
4. формулируется основной вопрос; он должен позволять экспертам давать ответ в количественной форме; одновременно составляется анкета с условиями проведения эксперимента;
5. проводится первый тур опроса;
6. анализируются ответы на предмет согласованности мнений, выявляются дополнительные факторы, которые необходимо учесть экспертам, чьи мнения не совпадают с большинством;
7. составляется и выдается экспертам дополнительная информация, и в связи с этим ставятся новые вопросы; «оригинальных» экспертов просят обосновать свои заключения;
8. проводится анализ ответов, и определяется необходимое количество повторений этапов 6) – 7);
9. корректируются полученные ответы;
10. обобщаются окончательные экспертные заключения и выдаются рекомендации по исследуемым вопросам.

5. **Методы ранжирования.**

задача ранжирования: (*расположение элементов конечного множества в порядке их значимости*)

При построении ранжированного ряда применяется метод непосредственного ранжирования или метод парных сравнений. Пусть имеется m отобранных экспертов и множество, состоящее из n элементов.

у экспертов есть оценка компетентности: коэффициент, матрица, средний коэф. компетентности

Метод непосредственного ранжирования заключается в присваивании каждым экспертом различных рангов элементам множества: i-й эксперт присваивает j-му элементу ранг $r_{ij} \in \{1, 2, ..., n\}$. В случае если эксперт считает некоторые элементы равнозначными, то каждому из них присваивается ранг, равный среднему значению рангов, которые эксперт присвоил бы элементам при неравнозначности.

Далее вычисляется средний ранг каждого элемента $r_j = \frac{1}{m}\sum_{i=1}^{m} r_{ij},\ j = 1, 2, ..., n.$ Ранжированный ряд строится по невозрастанию средних рангов.

Для определения совпадения мнений экспертов вычисляется сумма оценок отклонений

$$D(r) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(r_j - r_{ij})^2.$$

Чем ближе D(r) к нулю, тем более точно совпадают мнения экспертов

Метод парных сравнений применяется, если часть экспертов затрудняется построить ранжированный ряд, но может оценить элементы по значимости попарно. Обозначим отношение предпочтения на множестве элементов $\{e_1, e_2, ..., e_n\}$ через $\succ$, отношение эквивалентности - через ~. Каждый i-й эксперт строит n×n-матрицу предпочтений $B^i = [b_{kj}^i]$, где

$$b_{kj}^i = \begin{cases} 1, \text{ если } & e_k \succ e_j, \\ 0, \text{ если } & e_k \sim e_j, \\ -1, \text{ если } & e_k \prec e_j. \end{cases}$$

Далее вычисляется усредненная матрица предпочтений $B = \frac{1}{m}\sum_{i=1}^{m} B^i$, элементы которой не обязательно равны 1, 0, –1. Поэтому матрица B преобразуется в матрицу $\overline{B} = [\overline{b}_{kj}]$ явных предпочтений по следующему правилу:

$$\overline{b}_{kj} = \begin{cases} 1, \text{ если } & b_{kj} \ge \delta, \\ 0, \text{ если } & b_{kj} < \delta, \\ -1, \text{ если } & b_{kj} \le -\delta. \end{cases}$$

Здесь δ > 0 – порог, выбираемый из условия минимальности без нарушения правила транзитивности. По матрице B строится ранжированный ряд.

Для оценки единодушия экспертов применяется сумма оценок отклонений $D(b) = \frac{1}{mn^2}\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{j=1}^{n}(\overline{b}_{kj} - b_{kj}^i)^2.$

Чем ближе D(b) к нулю, тем более точно совпадает мнение экспертов.

Если при использовании методов ранжирования единодушие экспертов не достигнуто, то либо повторяют опрос, либо состав экспертов изменяется.

При построении ранжированного ряда важным является также выбор отношения предпочтения каждого элемента относительно других. Именно этот выбор устанавливает частичный порядок элементов в рассматриваемом множестве.

6. Основные типы неопределенности.

Если информация о системе и/или внешней среде частично отсутствует, то имеет место задача принятия решения в условиях неопределённости.

Типы неопределённости:

1. неопределённость целей;
   - многокритериальная (векторная) оптимизация
   - переменные $X = (x_1, x_2, \ldots, x_n)$ при которых $f_1(X) \to max \ldots f_n(X) \to max$
2. неопределённость знаний об окружающей обстановке и действующих в данном явлении факторах (неопределённость природы);
3. неопределённость действий активного/пассивного партнёра/противника.

Необходимо учитывать отношение к случайности.

- стохастическая (вероятностная) неопределённость;
- неопределённость не стохастического вида (не можем делать предположение о стохастических характеристиках);
- неопределённость промежуточного типа, решение принимается на основе гипотез о законах распределения случайных величин.

7. **Способы решения многокритериальных задач**

Многокритериальная задача оптимизации состоит в том, чтобы найти такие значения компонент вектора $X = (x_1, x_2, ..., x_r),$ для которых выполняются условия $g_j(X) \geq 0, \quad j = 1, 2, ..., m.$ При этом целевые функции

$$f_1(X) \to \max(\min), \ f_2(X) \to \max(\min), ..., f_n(X) \to \max(\min).$$

Многокритериальная задача оптимизации – математическая модель принятия оптимального решения одновременно по нескольким критериям.

Существуют два основных подхода к решению:

- сведение к стандартным задачам с одним критерием
- сужение неопределенности

Сведение к стандартной задаче с одним критерием

1. ***Линейная свертка.***

   Если все критерии измеряются по одной шкале, то строят обобщенный критерий вида:

$$F(X) = \sum_{i=1}^{n} c_i f_i(X), \quad \sum_{i=1}^{n} c_i = 1,$$

   где ci – «веса» соответствующих критериев (положительные числа), подбираются на основе оценок экспертов или экспериментально. Они отражают представление оперирующей стороны о важности каждого критерия. Таким образом, проблема неопределенности целей сведена к задаче их ранжирования на основе экспертных оценок.

2. ***Использование контрольных показателей.***

   Пусть экспертами установлены контрольные нормативные показатели $f_i^*, \ i = 1, ..., n,$, относительно которых критерии $f_i(X)$ должны удовлетворять условию:

$$f_i(X) \geq f_i^*, \quad i = 1, 2, ..., n.$$

   Предположим, что среди функций fi выделен основной критерий, например f1(X). Тогда снова приходим к однокритериальной задаче $f_1(X) \to \max$ при дополнительных условиях для i = 2, ..., n.

3. ***Введение метрики в пространство целевых функций.***

   Предположим, решено n однокритериальных задач оптимизации соответственно с критериями $f_i(X) \to \max, \quad i = 1, 2, ..., n,$ и для каждой i-й задачи определен вектор $\bar{X}_i$, доставляющий максимум критерию fi (X). Векторы $\bar{X}_i$ в n-мерном пространстве критериев определяют точку $(\bar{f}_1, \bar{f}_2, ..., \bar{f}_n),$ где $\bar{f}_i = \hat{f}_i(\bar{X}_i),$, которая называется абсолютным максимумом. Можно ввести расстояние от произвольной точки $\left(f_1(X), f_2(X), ..., f_n(X)\right)$ пространства критериев до абсолютного максимума, например, вычисляя стандартное евклидово расстояние

$$h(X) = \sqrt{\sum_{i=1}^{n} [f_i(X) - \bar{f}_i]^2}.$$

8. Сужение неопределенности, компромиссы Парето

Другой подход к решению многокритериальных задач заключается в попытке сократить множество допустимых вариантов - исключить заведомо плохие варианты. Предположим, что найдено допустимое решение-вектор X* и существует некоторый другой вектор X , при котором для всех критериев fi (X) выполняется условие

$$\begin{cases} f_i(\bar{X}) \le f_i(X^*), \quad i = 1, 2, \dots, n, \\ \text{причем хотя бы одно из неравенств строгое.} \end{cases}$$

Множество всех таких векторов X* называют **множеством Парето**. В теории принятия решений существует принцип Парето: в качестве решения следует выбирать вектор, который принадлежит множеству Парето. Таким образом, при использовании принципа Парето не выделяется единственное решение, а сужается множество возможных вариантов.

вектор X* предпочтительнее X. Поэтому имеет смысл все векторы типа X исключить из рассмотрения

Теорема 1.1 (Карлина). Пусть X 0 – точка Парето для исходной многокритериальной задачи при условии, что все $f_i(X),\ i = 1, 2, \dots, n,$ являются вогнутыми, а область Gx - невырожденной и выпуклой. Тогда существует такой вектор $\lambda(X^0) = (\lambda_1, \dots, \lambda_n),$ что в линейной свертке

$$f = \sum_{i=1}^{n} \lambda_i f_i(X)$$

точка X^0 является точкой оптимума функции f, т. е.

$$f(X^0) \ge f(X)$$

для любых $X \in G_x$.

9. **Линейные модели. Геометрическая интерпретация задач ЛП.**

Математическими моделями процессов управления организационно-экономическими системами очень часто являются задачи оптимизации функции многих переменных на множестве их возможных значений, т. е. задачи условной оптимизации. Исследование таких задач и методов их решения составляет содержание раздела прикладной математики, называемого математическим программированием.

Общая задача условной оптимизации может быть сформулирована следующим образом.

$$g_i(x_1, x_2, ..., x_n) \le (\ge) 0, \quad i = 1, 2, ..., m,$$

и для которых функция

$$f(x_1, x_2, ..., x_n)$$

достигает минимального (или максимального) значения.

В содержательной постановке задачи явно или неявно должны присутствовать два условия, приводящие к линейности модели: аддитивность и прямая пропорциональность.

Условие аддитивности. Если значение каждой управляемой переменной определено, то полное количество каждого из потребляемых ресурсов равно сумме одноименных ресурсов, затраченных на реализацию всех применяемых технологических процессов, а полная прибыль (или другая оценка) равна сумме прибылей от реализации всех процессов.

Условие прямой пропорциональности. Все показатели управляемого процесса могут быть увеличены или уменьшены при сохранении их взаимной пропорциональности. Например, себестоимость продукта должна прямо пропорционально зависеть от объема продукции. Естественно, что на практике это условие выполняется для сравнительно небольших плановых периодов. Это демонстрируют графики, представленные на рис. 2.1 и 2.2

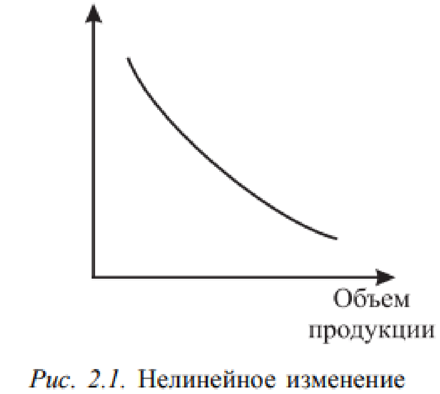


Рис. 2.1. Нелинейное изменение себестоимости на практике

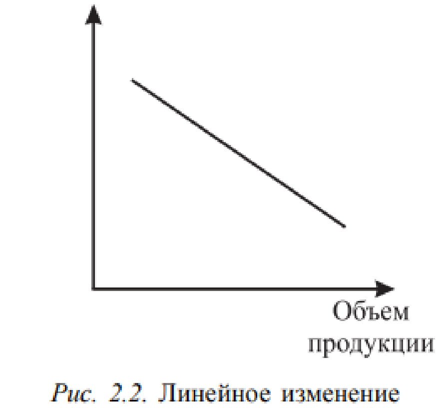


Рис. 2.2. Линейное изменение себестоимости в модели

10. Двойственная задача

Для каждой конкретной задачи ЛП можно построить двойственную ей задачу. Простейшие правила построения двойственных задач без оговорок о знаках двойственных переменных формируются при переходе от одной разновидности стандартной формы ЛП к другой и состоят в следующем:

1) каждому основному ограничению прямой задачи ЛП соответствует своя двойственная неотрицательная переменная;

2) в двойственной задаче число ограничений равно числу переменных в прямой задаче;

3) коэффициенты целевой функции прямой задачи ЛП являются свободными членами соответствующих ограничений в двойственной задаче;

4) свободные члены ограничений прямой задачи становятся коэффициентами целевой функции двойственной задачи;

5) матрица коэффициентов основных ограничений двойственной задачи ЛП - транспонированная матрица ограничений прямой задачи;

6) прямая задача ЛП - максимизация, двойственная - минимизация. Таким образом, задача ЛП в стандартной форме (например, при всех ограничениях ≤ и максимизации целевой функции) имеет двойственную задачу в иной стандартной форме (при всех ≥ и минимизации).

Доказаны следующие утверждения.

Теорема 2.1. Если у некоторой задачи ЛП есть оптимальное решение, то двойственная задача также имеет оптимальное решение, при этом оптимальные значения целевых функций равны.

Теорема 2.2. Задача, двойственная к двойственной задаче ЛП, совпадает с прямой задачей.

Теорема 2.3 (Джона фон Неймана). Если дана пара, состоящая из прямой и двойственных задач ЛП, то возможна в точности одна из трех ситуаций, показанных в таблице:

Прямая	Двойственная		
	Конечный оптимум	Неограниченная	Недопустимая
Конечный оптимум	✓ 1	✗	✗
Неограниченная	✗	✗	✓ 2
Недопустимая	✗	✓ 2	✓ 3

Теорема 2.4 (условие дополняющей нежесткости). Векторы X и Y, допустимые соответственно в прямой и двойственной задаче, оптимальны в том и только том случае, если:

$$y_i(a_i X - b_i) = 0 \text{ для } i = 1, 2, \ldots, m;$$

$$(c_i - Y^T A_j)x_i = 0 \text{ для } j = 1, 2, \ldots, n,$$

где ai - i-я строка матрицы основных ограничений A; Aj - j-й столбец матрицы A.

11. Анализ на устойчивость.

Пусть любой (действующий или воображаемый) технологический способ производства описывается вектором $a^S = (a_1^S, a_2^S, ..., a_m^S, a_{m+1}^S, ..., a_{m+n}^S)$,
где ai^S (i = 1 2, ..., N, N = n+m = 1 2, , ..., r) – объемы производства или затрат соответствующих продуктов при единичной интенсивности использования s-го технологического способа производства. Если ai^S > 0 , то i-й продукт производится, если ai^S < 0 – затрачивается, ai^S = 0 – не производится и не затрачивается. Ресурсы (произведенные или израсходованные), задаются вектором ограничений $\gamma = (b_1, b_2, ..., b_m)$.

План организации производства определяется вектором π = (x^1,x^2,...,x^r) с неотрицательными компонентами и указывает на интенсивность использования соответствующих способов производства. Пусть при фиксированном плане производства π различные продукты затрачиваются и производятся в количествах

$$\alpha_i^\pi = \sum_{S=1}^{r} a_i^S x^S, \qquad i = 1, 2, ..., N.$$

Таким образом, предполагается возможность линейного увеличения объемов производства за счет изменения интенсивностей использования технологических способов производства. По первым m продуктам вводятся ограничения $\alpha_i^\pi \geq b_i,\ i = 1, 2, ..., m,$ что означает: для bi < 0 суммарные затраты не должны превосходить имеющиеся ресурсы, для bi > 0 план производства соответствующих продуктов должен быть выполнен.

целевая функция - число ассортиментных наборов, определяемое величиной

$$\mu(\pi) = \min_{j=1,n} \frac{\alpha_{m+j}^\pi}{K_j}; \qquad \alpha_{m+j}^\pi = \sum_{S=1}^{r} a_{m+j}^S x^S,$$

где $K_j,\ j = 1, 2, ..., n,$ – количество продукта j в одном наборе.

Итак, возникает задача оптимизации: найти величины x^S, для которых

$$x^S \geq 0, \quad S = 1, 2, ..., r,$$

$$\alpha_i^\pi = \sum_{S=1}^{r} a_i^S x^S \geq b_i, \quad i = 1, 2, ..., m,$$

и

$$\min_j \frac{\alpha_{m+j}^\pi}{K_j} \to \max,$$

т. е. максимизируется количество ассортиментных наборов.

Теорема 2.5. Для оптимизации допустимого плана $\pi = (x^1, x^2, ..., x^r)$ необходимо и достаточно, чтобы существовал вектор оценок $y = (y_1, ..., y_N)$, такой, что:

а) $y_i \geq 0$, $i = 1, 2, ..., N$, $\max\limits_j y_{m+j} > 0$ – оценки продуктов неотрицательны и хотя бы один из производимых продуктов имеет положительную оценку;

б) $\sum\limits_{i=1}^{N} a_i^S y_i \leq 0$, $S = 1, 2, ..., r$ – суммарная оценка произведенной продукции не должна превышать оценки затрат;

в) $\sum\limits_{i=1}^{N} a_i^S y_i = 0$, $S = 1, 2, ..., r$, если $x^S > 0$, – в технологических способах, входящих в план, суммарная оценка продуктов равна суммарной оценке затрат;

г) $y_i = 0$, если $a_i^S > b_i$, $i = 1, 2, ..., m$, – ресурсы, не минимизирующие производство, получают нулевую оценку;

д) $y_{m+j} = 0$, если $\alpha_{m+j}^{\pi} > \mu(\pi) K_j$, $j = 1, 2, ..., n$, – продукция, производимая сверх требуемого ассортимента, получает нулевую оценку.

12. Модель Леонтьева, условия разрешимости.

2.7. ЛИНЕЙНАЯ МОДЕЛЬ МАКРОЭКОНОМИКИ (МОДЕЛЬ МЕЖОТРАСЛЕВОГО БАЛАНСА ЛЕОНТЬЕВА)

Ученый-экономист В. В. Леонтьев в 1920-х гг. предложил изучать функционирование национальной экономики в частично дезагрегированном виде. При таком подходе производственно-экономические процессы разукрупняются до уровня n отраслей (секторов) (многосекторная модель производства) и производится анализ потоков продуктов между отраслями.

Основные предпосылки анализа таковы:

а) в экономической системе производятся, покупаются, продаются, потребляются n типов продуктов $(i = 1, 2, ..., n)$;

б) каждая отрасль производит только один тип продукта, т. е. n отраслей и n продуктов находятся во взаимно-однозначном соответствии: i-й продукт выпускает i-я отрасль;

в) под производственным процессом в каждой отрасли подразумевается преобразование некоторых типов продуктов (в определенных количествах) в продукт данной отрасли.

Предполагается, что соотношение затрачиваемых и выпускаемых продуктов постоянно (во времени), причем если для производства единицы j-го продукта в j-й отрасли нужно непосредственно a_{ij} единиц i-го продукта, то для производства λ единиц необходимо λa_{ij} единиц i-го продукта. Величины a_{ij} называются коэффициентами (нормами) прямых затрат.

Обозначим через x_i количество i-го продукта, производимого в течение единицы времени (например, года), т. е. так называемый валовой выпуск.

Часть валового выпуска, а именно величина $\sum_{j=1}^{n} a_{ij} x_j$ потребляется в виде затрат, необходимых для производства. Соответственно, чистый выпуск получается вычитанием из валового выпуска полного количества продукта, потребляемого в виде производственных затрат, т. е. выражается величиной

$$x_i - \sum_{j=1}^{n} a_{ij} x_j .$$

Приравняв чистый выпуск конечному спросу b_i на i-й продукт, получим систему

$$x_i - \sum_{j=1}^{n} a_{ij} x_j = b_i , \quad i = 1, 2, ..., n, \tag{2.12}$$

или в виде

$$x - Ax = b, \tag{2.13}$$

где A – матрица коэффициентов прямых затрат, $b = b_1, b_2, ..., b_n$; b_i – величина потребления продукта i-го типа в непроизводственной сфере.

Сущность метода «затраты-выпуск» состоит в определении уровней (величин) валового выпуска по заданному конечному спросу с учетом коэффициентов прямых затрат, т. е. в определении величин $x_1, x_2, ..., x_n$.

Очевидно, что коэффициенты a_{ij} и значения x_j неотрицательны. Возникает вопрос: каково условие существования неотрицательного решения системы (2.12) при $b_i \geq 0$ и $a_{ij} \geq 0$ или, по-другому, при каких условиях модель Леонтьева является продуктивной?

Перейдем к системе

$$\sum_{j=1}^{n} d_{ij} x_j = b_i, \quad i = 1, 2, ..., n, \tag{2.14}$$

коэффициенты которой неположительны при $i \neq j$.

Теорема 2.6. Условия:

а) система (2.14) разрешима в неотрицательных переменных $x_i \geq 0$, $i = 1, 2, ..., n$, при некотором наборе $b_i \geq 0$;

б) система (2.14) разрешима в неотрицательных переменных $x_i \geq 0$, $i = 1, 2, ..., n$, при любом наборе $b_i \geq 0$;

в) в матрице коэффициентов системы (2.14) все n главных миноров положительны, т. е.

$$\begin{vmatrix} d_{11} & \dots & d_{1k} \\ \dots & \dots & \dots \\ d_{k1} & \dots & d_{kk} \end{vmatrix} > 0, \quad k = 1, 2, ..., n,$$

взаимно эквивалентны.

13. Транспортная задача.

Классическая ТЗ состоит в следующем. Имеется n поставщиков A_1, A_2, ..., A_n определенного однородного товара, которые предлагают соответственно a_1, a_2, ..., a_n единиц этого товара m потребителям B_1, B_2, ..., B_m, создающим спрос на него в количествах b_1, b_2, ..., b_m единиц соответственно. Для каждой пары «i-й поставщик - j-й потребитель» известна стоимость c_{ij} (i = 1, 2, ..., n; j = 1, 2, ..., m) перевозки единицы товара. Требуется определить такой план удовлетворения спроса потребителей, при котором общие транспортные затраты минимальны. Очень часто на практике предполагается, что такие затраты являются суммой затрат на перевозку товара для каждой пары «i-й поставщик – j-й потребитель», которые, в свою очередь, прямо пропорциональны с коэффициентом c_{ij} количеству перевозимого товара. Таким образом, выполняются свойства аддитивности и прямой пропорциональности

нахождение плана - определить *xij* - количество товара, перевозимого от i-го поставщика к j-му потребителю, для которых выполняются условия

$$x_{ij} \ge 0, \quad i = 1, 2, ..., n; \; j = 1, 2, ..., m; \qquad (2.18)$$

$$\sum_{j=1}^{m} x_{ij} \le a_i, \quad i = 1, 2, ..., n; \qquad (2.19)$$

$$\sum_{i=1}^{n} x_{ij} \ge b_j, \quad j = 1, 2, ..., m, \qquad (2.20)$$

и для которых величина суммарных затрат

$$f = \sum_{i=1}^{n} \sum_{j}^{m} c_{ij} x_{ij} \to \min. \qquad (2.21)$$

Построенная модель называется открытой моделью ТЗ. В целях стандартизации и упрощения изложения часто принимают, что выполняются условия баланса суммарного предложения и суммарного спроса, т. е. равенство

$$\sum_{i=1}^{n} a_i = \sum_{j=1}^{m} b_j. \qquad (2.22)$$

Тогда неравенства (2.19) и (2.20) превращаются в равенства

$$\sum_{j=1}^{m} x_{ij} = a_i, \quad i = 1, 2, ..., n; \qquad (2.23)$$

$$\sum_{i=1}^{n} x_{ij} = b_j, \quad j = 1, 2, ..., m. \qquad (2.24)$$

Для решения ТЗ можно было бы применить любой общий метод решения ЗЛП. Однако в силу специфики задачи применяются иные, существенно более эффективные методы.

14. Метод потенциалов.

Для решения ТЗ используют так называемую транспортную таблицу:

Запасы продукта		Потребность в продукте					
		B_1	B_2	....	B_j	....	B_m
		b_1	b_2	....	b_j	....	b_m
A_1	a_1	c_{11} x_{11}	c_{12} x_{12}	....	c_{1j} x_{1j}	....	c_{1m} x_{1m}
A_2	a_2	c_{21} x_{21}	c_{22} x_{22}	....	c_{2j} x_{2j}	....	c_{2m} x_{2m}
....	....	....	....	....	....	....	....
A_i	a_i	c_{i1} x_{i1}	c_{i2} x_{i2}	....	c_{ij} x_{ij}	....	c_{im} x_{im}
....	....	....	....	....	....	....	....
A_n	a_n	c_{n1} x_{n1}	c_{n2} x_{n2}	....	c_{nj} x_{nj}	....	c_{nm} x_{nm}

Удобно, если в такой таблице выполняется условие баланса (2.22).

Если на практике $\sum_{i=1}^{n} a_i > \sum_{j=1}^{m} b_j$, то для получения замкнутой модели к транспортной таблице добавляют фиктивного потребителя B_{m+1}, спрос которого равен $b_{m+1} = \sum_i a_i - \sum_j b_j$ и до которого стоимость перевозки единицы товара $c_{i,m+1} = 0$ $(i = 1, 2, ..., n)$.

Если на практике $\sum_{i=1}^{n} a_i < \sum_{j=1}^{m} b_i$, то для получения замкнутой модели к транспортной таблице добавляют фиктивного поставщика A_{n+1}, предложение которого равно $a_{n+1} = \sum_j b_j - \sum_i a_i$ и от которого стоимость перевозки единицы товара $c_{n+1,j} = 0$ $(j = 1, 2, ..., m)$.

Анализируя структуру матрицы коэффициентов при неизвестных x_{ij} в основных ограничениях (2.23), (2.24), легко доказать, что ее ранг равен $n + m - 1$. Это, в частности, означает, что любому базису вектор-столбцов этой матрицы будет соответствовать допустимое решение ТЗ, содержащее не более чем $n + m - 1$ клеток транспортной таблицы с ненулевыми значениями перевозок. В остальных клетках они будут нулевыми. Такие решения называются допустимыми базисными решениями.

Теорема 2.9. Решение ТЗ находится во множестве допустимых базисных решений.

Схема любого эффективного метода решения ТЗ (например, метода потенциалов или так называемого венгерского метода) состоит в следующем.

Строится исходное допустимое базисное решение, для него оценивается значение целевой функции (2.21) и проверяется так называемый критерий оптимальности, который устанавливает, является ли полученное значение целевой функции минимальным. Если этот критерий не выполняется, то базисное решение преобразуется так, чтобы значение целевой функции по возможности уменьшилось (или, по крайней мере, не возросло). Снова проверяется критерий оптимальности, и такая итерация повторяется до тех пор, пока критерий оптимальности не выполнится. Для ТЗ доказано, что число итераций будет конечно, т. е. что не случится так называемого зацикливания или повторения допустимых базисных решений.

Исходное допустимое базисное решение, содержащее не более чем $n + m - 1$ клеток с ненулевыми перевозками, строится следующим образом.

Общий шаг. В транспортной таблице выбирается любая клетка, например клетка (s, t), и в ней устанавливается значение перевозки

$$x_{st} = \min(a_s, b_t).$$

Для поставщика A_s и потребителя B_t обновляются соответственно значения $a'_s = a_s - x_{st}$ и $b'_t = b_t - x_{st}$. Далее в ряду (строке или столбце) транспортной таблицы, где после такого обновления получили нулевое значение a'_s или b'_t, во всех оставшихся клетках устанавливают нулевые значения перевозок. После этого общий шаг повторяют для оставшейся, еще не заполненной части транспортной таблицы.

Примечание. Если после обновления величин a'_s и b'_t получилось, что обе они равны 0, то нулевые значения перевозок устанавливаем только в одном соответствующем ряду (например, только в строке), другой (неиспользованный) нуль подчеркиваем. Это так называемый активный нуль, который будет использоваться для заполнения клеток на последующих общих шагах и тем самым войдет в допустимое базисное решение.

После построения допустимого базисного решения вычисляем значение целевой функции. Поскольку решение ТЗ состоит в минимизации суммарных транспортных затрат, то уже на этапе построения исходного допустимого базисного решения имеет смысл использовать для заполнения стратегию выбора на каждом общем шаге клетки (s, t), которая соответствует минимальному значению стоимости перевозки единицы товара. Часто (но не всегда) такая стратегия сразу приводит к оптимальному решению.

Как уже отмечалось, построенное допустимое базисное решение оценивается на оптимальность с помощью специального критерия. Например, в так называемом методе потенциалов этими потенциалами фактически будут двойственные переменные u_i и v_j, причем для всех $n + m - 1$ клеток, принадлежащих допустимому базисному решению, выполняется равенство $u_i + v_j = c_{ij}$.

Доказано, что если при этом величины $\Delta_{ij} = u_i + v_j - c_{ij} \leq 0$ для всех остальных (небазисных) клеток таблицы, то имеющееся допустимое базисное решение является оптимальным. Это и есть критерий оптимальности в методе потенциалов для решения ТЗ.

После получения решения ТЗ необходима его практическая интерпретация. Если в исходной постановке уравнение (2.22) выполнено, то полученные ненулевые значения представляют собой количество единиц товара перевозимого от поставщиков соответствующим потребителям.

Если в исходной постановке был вариант $\sum_{i=1}^{n} a_i > \sum_{j=1}^{m} b_j$, то оптимальные значения перевозок фиктивному потребителю будут в оптимальном плане означать величину неиспользованного предложения для соответствующих поставщиков. Если же в исходной постановке $\sum_{i=1}^{n} a_i < \sum_{j=1}^{m} b_j$, то оптимальные значения перевозок от фиктивного поставщика будут означать величину неудовлетворенного спроса для соответствующих потребителей.

Метод построения исходного базисного решения и определения значений перевозок в клетках транспортных таблиц может быть основанием для следующего утверждения.

Теорема 2.10. Если величины $a_1, a_2, ..., a_n$ и $b_1, b_2, ..., b_m$ целочисленные, то и оптимальное решение ТЗ будет целочисленно.

Таким образом, для ТЗ условие делимости исходных неизвестных не существенно.

15. Элементы теории графов, основные определения.

Многочисленные задачи принятия оптимальных решений сводятся к математическим моделям в виде конечных графов или сетей.

Конечным графом G называется пара (X, U), где X – конечное множество элементов любой природы; U – некоторое подмножество множества всех пар элементов из X.

Относительно элементов множества U (при $U \neq \varnothing$) возможны два случая, когда рассматриваются:

1) упорядоченные пары;

2) неупорядоченные пары.

В первом случае граф называется *ориентированным* (*орграфом*), во втором – *неориентированным*. Если граф ориентированный, тогда U – подмножество прямого произведения множества X на себя. Ориентированные графы можно задавать в виде пары (X, Γ), где Γ – отображение множества X на себя.

Всегда, если это осуществимо, граф можно изображать на плоскости в виде произвольно разбросанных помеченных точек множества X, которые связываются линиями в соответствии с определением пар из U. Если какая-то пара $[a, b] \in U$ или $(a, b) \in U$, то точки a и точки b соединяются между собой самонепересекающейся линией. Если пара не упорядочена – линия не имеет ориентации, если упорядочена, то указывается ориентация линии в виде стрелки, направленной от первой координаты пары ко второй.

Например, пусть $X = \{a, b, c, d, f\}$, $U = \{[a, b], [b, b], [a, f], [c, a], [d, f]\}$, тогда имеем неориентированный граф (рис. 3.1).

Если же $U = \{(a, b), (b, b), (a, f), (d, f), (c, a)\}$, то имеем ориентированный граф (рис. 3.2).

В ориентированном графе пара (a, b) отличается от пары (b, a), следовательно, возможна конфигурация, представленная на рис. 3.3.

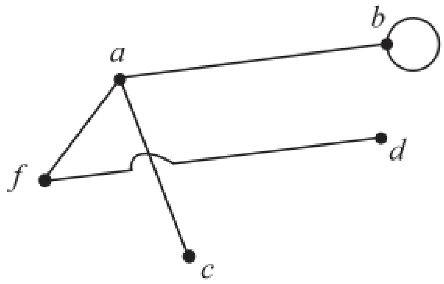


Рис. 3.1

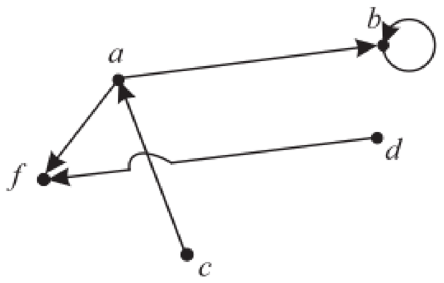


Рис. 3.2

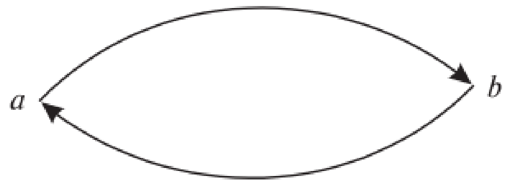


Рис. 3.3

В связи с геометрической интерпретацией в графе (X, U) элементы из X называют *вершинами*, упорядоченные пары из U – *дугами*, неупорядоченные – *ребрами*. Возможны и смешанные графы, в которых часть пар из U упорядочены, а часть нет.

Путем в орграфе называется последовательность вида $\left((x_1, x_2), (x_2, x_3), \ldots, (x_{n-1}, x_n)\right)$, причем конец каждой предыдущей пары в этой последовательности является началом последующей пары (рис. 3.4).

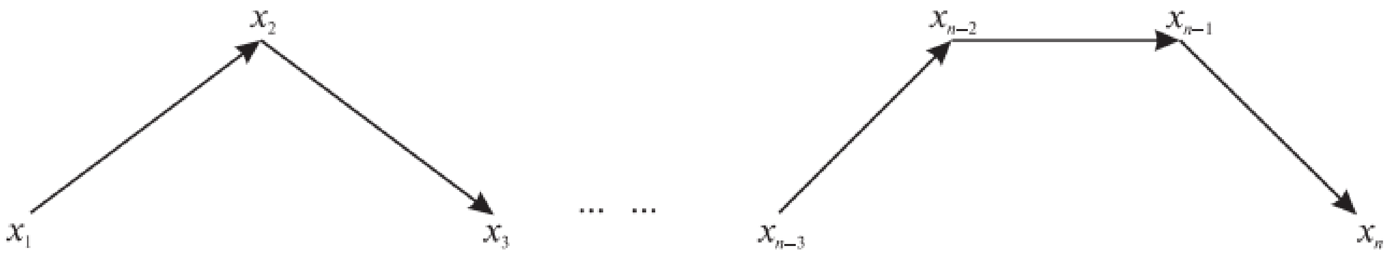


Рис. 3.4

Длиной пути обычно называется количество входящих в него дуг.

Путь называется *контуром*, если его начало совпадает с его концом, т. е. $x_n = x_1$. *Петлей* называется контур длиной 1.

Для неориентированных графов вводят понятие цепи аналогично понятию пути и понятие цикла аналогично понятию контура.

Неориентированный граф называется *связным*, если между любой парой вершин существует хотя бы одна цепь.

В орграфах существует аналогичное понятие сильной связности: орграф называется *сильно связным*, если между любой парой его вершин существует хотя бы один путь.

Если имеем любой неориентированный связный граф G, то из него можно последовательно удалять ребра так, чтобы сохранялась связность, пока не останется ровно $n - 1$ ребер, где n – число вершин графа. Полученный таким образом подграф называется деревом-остовом. Если к дереву-остову добавить одно ребро из вне остова, то в подграфе образуется ровно один цикл. Если из этого цикла удалить любое ребро, то получим новое дерево-остов.

Часто вместо обычных графов изучают более сложные объекты – мультиграфы. Мультиграф определяется в случае, если в множестве U его дуг (или ребер) одна и та же пара встречается несколько раз. Таким образом, в мультиграфах между парами вершин допускается наличие нескольких дуг (или ребер), идущих в одном направлении.

В ориентированных графах различают такие понятия, как полустепень захода и исхода для каждой вершины из X. *Полустепенью захода вершины* $x \in X$ называется количество входящих в эту вершину дуг, *полустепень исхода* – это количество дуг, исходящих из вершины.

Для неориентированных графов существует аналогичное понятие *степени вершины* – количество связанных с данной вершиной ребер.

Пусть задан граф $G = (X, U)$, причем множество X состоит из n элементов. *Матрицей смежности графа* G (или мультиграфа) называется квадратная матрица A порядка n, каждый элемент a_{ij} которой означает количество дуг (или ребер), связывающих вершину x_i с вершиной x_j. Естественно, что для обычных графов элементами матрицы смежности будут нули и единицы. Для мультиграфов в матрице смежности возможны и другие целочисленные элементы матрицы.

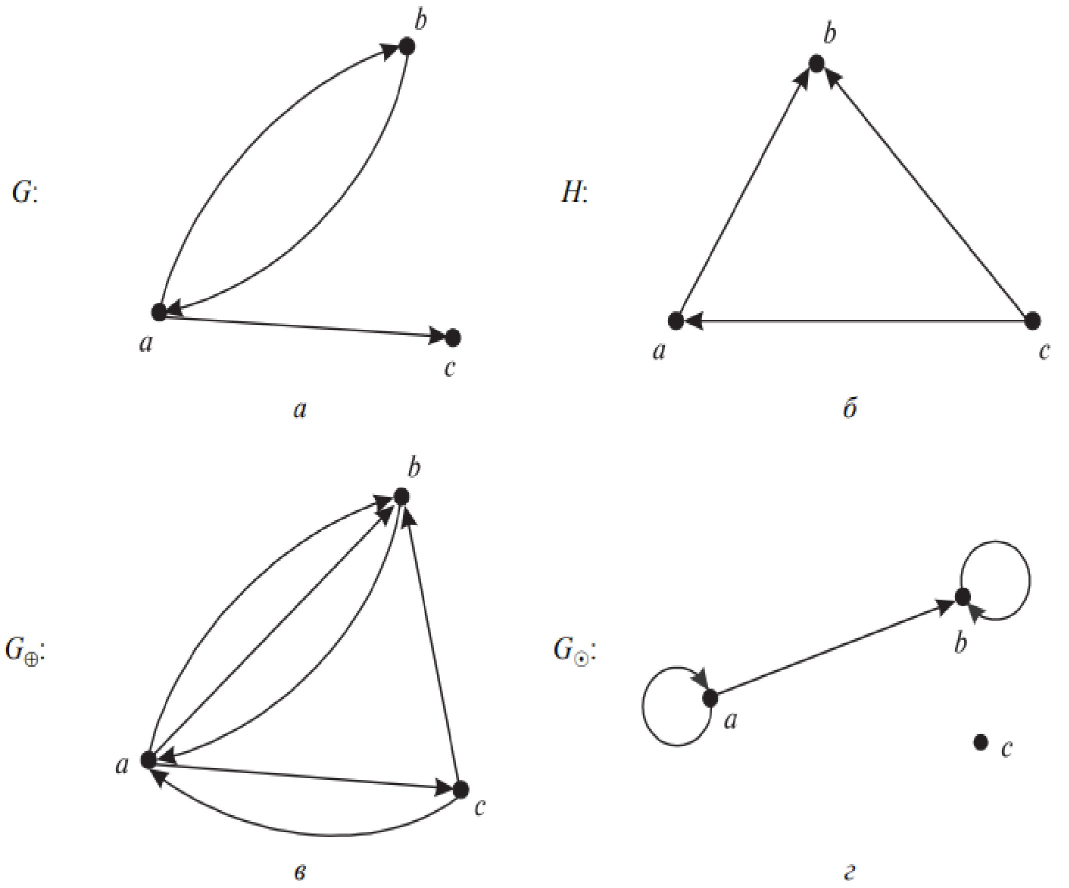
16. Теорема о мультиграфах и следствия.

Теорема 3.1. Пусть $G = (X, U)$ и $H = (X, V)$ – два ориентированных мультиграфа с матрицами смежности $A = (a_{ij})_{n\times n}$ и $B = (b_{ij})_{n\times n}$. Тогда:

1) матрице $A + B$ соответствует мультиграф $G_\oplus$, который получается объединением множеств дуг U и V;

2) матрице $A \cdot B$ соответствует мультиграф $G_\odot$, который построен на множестве вершин X, и каждые две его вершины x и y соединяются таким количеством дуг, сколько существует между ними путей длины 2, причем первая дуга каждого такого пути должна принадлежать графу G, а вторая – графу H.

Пример 3.2. Пусть имеются два графа: G (рис. 3.6, *а*) и H (рис. 3.6, *б*). Результаты выполнения соответствующих операций $G_\oplus$ и $G_\odot$ представлены на рис. 3.6, *в*, *г*.



а *б* *в* *г*

Используя матрицы смежности, эти операции можно представить следующим образом:

$$A + B = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix},$$

$$A \cdot B = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Доказательство. 1. В мультиграфе $G_{\oplus} = (X, U \cup V)$ между любой парой вершин x_i и x_j будет количество дуг, равное $a_{ij} + b_{ij}$, а это общий элемент матрицы $A + B$.

2. Посчитаем число путей вида (x_i, x_k, x_j), в которых $(x_i, x_k) \in U$, $(x_k, x_j) \in V$. Если зафиксировать вершину x_k, то число таких путей, что очевидно, равно $a_{ik} \cdot b_{kj}$. Если рассмотреть всевозможные пути при переменных x_k, то получим $\sum_{k=1}^{n} a_{ik} b_{kj}$, т. е. общий элемент матрицы $A \cdot B$.

Следствие 3.1. Если G – мультиграф и A – его матрица смежности, то каждый элемент p_{ij} матрицы A^{λ}, где λ – произвольное натуральное число, означает количество путей длины λ, связывающих вершину x_i с вершиной x_j графа G.

Доказательство. Для случая $\lambda = 1$ утверждение очевидно по определению матрицы смежности. Пусть утверждение верно для $\lambda - 1$. Учитывая, что $A^{\lambda} = A \cdot A^{\lambda-1}$, можно применить условие теоремы. Тогда длина пути будет равна сумме $1 + \lambda - 1 = \lambda$.

Пример 3.3. Если A – матрица смежности графа G, найдена матрица A^7 и ее элемент $p_{35} = 10$, то это означает, что в графе G между вершинами x_3 и x_5 существует 10 путей длины 7.

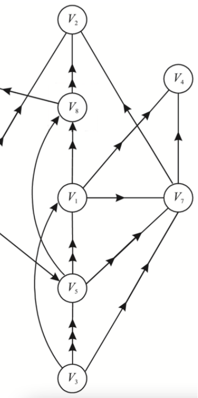
Следствие 3.2. В любом графе G существует путь длины λ (λ – любое натуральное число) тогда и только тогда, когда $A^{\lambda} \neq 0$. Не существует контуров тогда и только тогда, когда $A^{\lambda} = 0$ начиная с некоторого натурального λ.

17. Задача о разузловании.

Пусть производится изделие, состоящее из различных узлов и деталей. Относительно любой пары «узел – изделие» или «деталь – узел» известно количество деталей или узлов, используемых при изготовлении более сложных узлов или изделия. Задача разузлования состоит в том, чтобы определить, сколько всего необходимо всевозможных деталей и узлов для изготовления изделия.

Пусть задача о разузловании описывается мультиграфом (рис. 3.7), на котором V2, V4, V9 - изделия; V3, V6 - детали. Остальные вершины - узлы разной степени сложности. Дуги указывают вхождение, количество деталей или узлов, используемых для более сложных, указано количеством стрелок на дугах.

Требуется определить, сколько всего деталей и узлов нужно иметь, чтобы произвести по одному изделию V2, V4, V9.



$$A=\begin{array}{c}1\\2\\3\\4\\5\\6\\7\\8\\9\end{array}\begin{bmatrix}0&0&0&2&0&0&1&1&0\\0&0&0&0&0&0&0&0&0\\1&0&0&0&3&0&2&0&0\\0&0&0&0&0&0&0&0&0\\2&0&0&0&0&0&2&1&0\\0&2&0&0&1&0&0&0&3\\0&1&0&1&0&0&0&0&0\\0&2&0&0&0&0&0&0&1\\0&0&0&0&0&0&0&0&0\end{bmatrix},\quad T=\begin{array}{c}1\\2\\3\\4\\5\\6\\7\\8\\9\end{array}\begin{bmatrix}1&3&0&3&0&0&1&1&1\\0&1&0&0&0&0&0&0&0\\7&33&1&27&3&0&15&10&10\\0&0&0&1&0&0&0&0&0\\2&10&0&8&1&0&4&3&3\\2&12&0&8&0&1&4&3&6\\0&1&0&1&0&0&1&0&0\\0&2&0&0&0&0&0&1&1\\0&0&0&0&0&0&0&0&1\end{bmatrix}$$

Построим матрицу непосредственных (прямых) затрат как матрицу *A* смежности данного мультиграфа.

Тогда матрицу *T* полных затрат можно считать по формуле $T = E + A + A^2 + \ldots$ (3.1) Формула (3.1) интерпретируется следующим образом: при подсчете всевозможных прямых и косвенных вхождений учитываются непосредственные вхождения (это слагаемое *A*), далее к ним нужно добавить всевозможные косвенные вхождения через одну промежуточную вершину, т. е. необходимо перечислить пути длины два. По следствию 3.1. такие пути задаются матрицей *A*^2, затем следует учитывать косвенные вхождения, проходящие через две промежуточные вершины, т. е. всевозможные пути длины три, которые задаются матрицей *A^3*, и т. д. Матрица *E* отражает необходимость наличия хотя бы единицы каждой детали, узла и изделия.

В графе нет контуров: деталь или узел не может использоваться для производства самого себя. это означает, что существует натуральное λ, начиная с которого *A^λ=0*, т. е. (3.1) примет вид $T = E + A + A^2 + \ldots + A^{\lambda-1} = (E - A)^{-1}$ (3.2).

Значение λ – 1 означает длину максимального пути в исследуемом мультиграфе. При использовании формулы (3.1) это значение может быть заранее не известно, поэтому придется выполнить лишнее умножение матриц, пока не получим *A^λ=0*.

Матрица *T* полных затрат. Если план производства узлов и изделий задан в виде вектора *b*, то, $T \cdot b = (E - A)^{-1} \cdot b$, определяет общие потребности во всех деталях и узлах для выполнения плана.

Каждый элемент t_{ij} матрицы *T* означает количество деталей или узлов *i*-го вида, необходимых для изготовления более сложных узлов (или изделия) *j*-го вида. Так, $t_{32} = 33$ означает, что для изготовления V_2 необходимо 33 детали V_3. Знание элементов матрицы полных затрат *T* является особо важным при планировании производства.

На практике существует другой способ решения задачи о разузловании, когда последовательно, сверху вниз, расписываются уравнения вхождения каждого узла и детали. Например, $A_9 = 3A_6 + A_8$.

Аналогичные приемы нахождения матрицы полных затрат по формуле типа (3.1) применяются и для случаев, когда мультиграф имеет контуры. Для отыскания матрицы полных затрат обычно используют формулу (3.1), но при этом необходимо доказать сходимость матричного ряда. Ряд сходится, если все собственные значения матрицы *A* по модулю меньше 1.

18. Потоки в сетях, основные определения.

Сетью будем называть любой орграф без петель, у которого каждой дуге приписано фиксированное действительное число. Пусть каждой дуге (x, y) сети G=(X, U) поставлено в соответствие неотрицательное число $c(x, y)$ - пропускная способность дуги. Пусть в сети зафиксированы две вершины s, t.

Определение 3.1. Стационарным потоком величины v из вершины s в вершину t в сети $G = (X, U)$ называется функция $f(x, y)$, определенная на всех дугах $(x, y) \in U$, которая удовлетворяет следующим условиям:

$$0 \le f(x, y) \le c(x, y) \text{ для всех } (x, y) \in U; \qquad (3.3)$$

$$\sum_{y \in A(x)} f(x,y) - \sum_{y \in B(x)} f(y,x) = \begin{cases} v, & \text{если } x = s, \\ 0, & \text{если } x \ne s, t, \\ -v, & \text{если } x = t, \end{cases} \qquad (3.4)$$

где $A(x)$ – множество вершин сети, которые являются концами дуг, выходящих из вершины x; $B(x)$ – множество вершин сети, которые являются началами дуг, входящих в вершину x. Вершина s в случае задания потока по формулам (3.3), (3.4) называется *источником*, вершина t – *стоком.* Условие (3.3) означает, что поток по каждой дуге не должен превышать ее пропускную способность. Условия (3.4) показывают, что для источника s алгебраическая сумма величин выходящего и входящего потоков равна v – величине потока в сети. Аналогично для стока t суммарное количество входящего и выходящего потоков также равно v. Для всех остальных промежуточных вершин сети выполняется равенство $\sum_y f(x, y) - \sum_y f(y, x) = 0$ т. е. должен выполняться баланс.

Определение 3.2. Разрезом сети, отделяющим источник s от стока t, называется множество дуг, обозначаемое через (X_1, X_1^-), такое, что каждая дуга начинается в множестве X_1 и оканчивается в множестве X_1, и $\bar{X}_1 = X \setminus X_1$

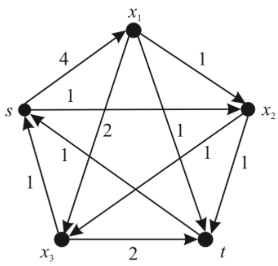


Рис. 3.8

Для построения произвольного разреза достаточно задать множество вершин X_1 как непустое подмножество X, содержащее s и не содержащее t, тогда X_1^- будет дополнением X_1 до множества *X.*

(пример к рис. 3.8):

$$x_1 : 4 = 2 + 1 + 1,$$
$$x_2 : 1 + 1 = 1 + 1,$$
$$x_3 : 1 + 2 = 2 + 1;$$

19. Основная лемма о стационарных потоках.

Сеть - орграф без петель, у которого каждой дуге приписано фиксированное действительное число. Пусть каждой дуге (x, y) сети G = (X, U) поставлено в соответствие неотрицательное число c(x, y). Это число будем называть пропускной способностью дуги. Наглядно можно представить, что значение c(x, y) является максимальным количеством какого-то однородного товара, которое можно провезти по дуге (x, y). Пусть в сети зафиксированы две вершины s (источник), t (сток).

Стационарным потоком величины v из вершины s в вершину t в сети G = (X, U) называется функция f(x, y), определенная на всех дугах (x, y) ∈ U, которая удовлетворяет следующим условиям:

0 ≤ f(x, y) ≤ c(x, y) для всех (x, y) ∈ U;

$$\sum_{y \in A(x)} f(x,y) - \sum_{y \in B(x)} f(y,x) = \begin{cases} v, & \text{если } x = s, \\ 0, & \text{если } x \neq s, t, \\ -v, & \text{если } x = t, \end{cases}$$ s - источник, t - сток

Разрезом сети, отделяющим источник s от стока t, будем называть множество дуг, обозначаемое через (X1 , X1’), такое, что каждая дуга начинается в множестве X1 и оканчивается в множестве X1’, и X1’=X\X1.

Лемма (3.1) Любой путь из s в t на заданной сети содержит, по крайней мере, одну дугу произвольного разреза, отделяющего s от t.

Док-во:

Пусть задан произвольный путь из s в t, проходящий, например,
через вершины s x , 1 2 , , , x x n, . t При движении по этому пути с обеих сторон, от s и от t,
обязательно возникает ситуация, когда две соседние вершины, лежащие на этом пути,
будут принадлежать двум соседним множествам X1 и X’1, т. е. определять дугу разреза.

Следствие (3.3) Если из сети убрать дуги произвольного разреза, то не останется путей, ведущих из s в t.

Лемма (3.2) Пусть f – произвольный поток величины v из s в t на сети G = (X, U), и пусть (X1,X1’), – произвольный разрез, отделяющий источник s от стока t на сети. Тогда имеют место следующие

соотношения: $$v = f(X_1, \overline{X_1}) - f(\overline{X_1}, X_1) \leq c(X_1, \overline{X_1}).$$, где

$$f(X_1, \bar{X}_1) = \sum_{(x,y) \in (X_1, \bar{X}_1)} f(x,y)$$

$$f(\bar{X}_1, X_1) = \sum_{(x,y) \in (\bar{X}_1, X_1)} f(y,x)$$

$$c(X_1, \bar{X}_1) = \sum_{(x,y) \in (X_1, \bar{X}_1)} c(x,y).$$

Величина c (X1,X1’) называется пропускной способностью разреза (X1,X1’).
Вид целевой функции, т. е. f = v, показывают, что задача о максимальном потоке представляет собой задачу линейного программирования.

20. Теорема о максимальном потоке.

Разрезом в сети называется разбиение вершин на две части таким образом, что исток находится в одной части, а сток - в другой. **Пропускной способностью** разреза называется сумма пропускных способностей ребер, которые соединяют вершины из разных частей разреза (то есть исходят из одной и попадают в другую).

Теорема(форд-фалкерсон) о максимальном потоке утверждает, что максимальный поток в сети равен минимальной стоимости разреза. Другими словами, чтобы найти максимальный поток в сети, нужно найти разрез с минимальной стоимостью, а затем поток, который проходит через этот разрез, будет максимальным.

Доказательство от противного предполагает, что максимальный поток не равен минимальной пропускной способности разреза. Тогда существует путь от истока к стоку, на котором все ребра имеют неотрицательную остаточную пропускную способность. Остаточная пропускная способность - это разница между пропускной способностью ребра и текущим потоком по этому ребру.
Далее, мы можем увеличить поток вдоль этого пути на величину, равную наименьшей остаточной пропускной способности на этом пути. Это не нарушит ограничения на максимальный поток, так как этот поток будет меньше или равен остаточной пропускной способности любого разреза между истоком и стоком.
Повторяя этот процесс, мы можем увеличивать поток, пока не достигнем максимальной величины потока, что противоречит начальному предположению о том, что максимальный поток не равен минимальной пропускной способности разреза.
Таким образом, мы доказали, что максимальная величина потока в сети равна минимальной пропускной способности разреза между истоком и стоком.

21. Алгоритм расстановки пометок.

Алгоритм расстановки пометок

Пусть имеется сеть $G = (X, U)$, на каждой дуге (x, y) которой заданы величины $c(x, y)$. Считаем, что $c(x, y)$ – рациональные числа. Не уменьшая общности (из-за возможности подбора масштаба измерений), можно рассматривать задачу при целочисленных $c(x, y)$. В практических задачах предположения о рациональности $c(x, y)$ достаточно.

Алгоритм начинается с произвольного начального потока в сети. В качестве такого начального потока можно выбирать нулевой поток, когда все $f(x, y) = 0$.

Алгоритм состоит из двух этапов:

На I этапе систематизируется поиск пути, увеличивающего поток. Это осуществляется с помощью расстановки пометок. При этом каждая вершина находится в одном из трех состояний:

а) *непомеченная*; б) *помеченная* и *непросмотренная*; в) *просмотренная*.

Этап I начинается в момент, когда все вершины не помечены. После завершения этапа I на этапе II в случае обнаружения пути, увеличивающего поток, происходит изменение (увеличение) этого потока. Шаг повторяется.

22. Обобщение задачи о максимальном потоке.

Обобщения, сводящиеся к исходной задаче [править | править код]

Некоторые обобщения задачи о максимальном потоке легко сводятся к исходной задаче.

Произвольное число источников и/или стоков [править | править код]

Если источников больше одного, добавляем новую вершину S, которую делаем единственным источником. Добавляем рёбра с бесконечной пропускной способностью от S к каждому из старых источников.

Аналогично, если стоков больше одного, добавляем новую вершину T, которую делаем единственным стоком. Добавляем рёбра с бесконечной пропускной способностью от каждого из старых стоков к T.

Неориентированные рёбра [править | править код]

Каждое неориентированное ребро (u, v) заменяем на пару ориентированных рёбер (u, v) и (v, u).

Ограничение пропускной способности вершин [править | править код]

Каждую вершину v с ограниченной пропускной способностью c_v расщепляем на две вершины v_{in} и v_{out}. Все рёбра, до расщепления входящие в v, теперь входят в v_{in}. Все рёбра, до расщепления исходящие из v, теперь исходят из v_{out}. Добавляем ребро (v_{in},v_{out}) с пропускной способностью c_v.

Ограничение пропускной способности рёбер снизу [править | править код]

В данном варианте постановки задачи значение потока каждого ребра дополнительно ограничено снизу функцией $c': E \to \mathbb{N}^+$. Таким образом величина потока для любого ребра не может превысить его пропускную способность, но и не может быть меньше заданного минимума, т.е. $c'(e) \leqslant f(e) \leqslant c(e)$. Для решения задачи необходимо преобразовать исходную транспортную сеть $G = (V, E, s, t, c, c')$ в транспортную сеть $G' = (V', E', s', t', c'')$ следующим образом:

1. Добавь новые источник s' и сток t'.
2. Для каждого ребра (v, w):
    1. Создай две новые вершины x и y.
    2. Установи $c''(v, x) = c(v, w)$ и $c''(y, w) = c(v, w)$.
    3. Установи $c''(x, y) = c(v, w) - c'(v, w)$.
    4. Установи $c''(s', y) = c'(v, w)$ и $c''(x, t') = c'(v, w)$.
3. Установи $c''(t, t') = c''(s, s') = \sum\limits_{e \in E} c(e)$.

В G определён поток, удовлетворяющий условию об ограничении пропускной способности ребёр снизу, тогда и только тогда, когда в G' определен максимальный поток, в котором все рёбра вида (s', y) и (x, t') "насыщены". Благодаря такому построению алгоритм нахождения потока, ограниченного снизу будет следующим:

1. Из G построй G'.
2. Найди поток f' графа G', предпочитая рёбра вида (s', y) и (x, t').
3. Если $f' < f'' + \sum\limits_{e \in E(G)} c'(e)$, где f'' - поток графа G в котором опущена пропускная способность рёбер снизу, то решения не существует.
4. Иначе вычисли поток f из потока f', т.е. $f(a, b) = f(a, x)$.

23. Двудольные графы. Алгоритм нахождения ММНДК.

Двудольный граф - граф G=(X, U), у которого вершины разделены на подмножества S и T: причем $S \cap T = \varnothing$ и для каждой дуги $(x, y) \in U$, $x \in S$ и $y \in T$.

S,T-рассекающим множеством вершин двудольного графа будем называть любое множество вершин, исключение которого из графа G приводит к блокированию дуг, ведущих из S в T. Например, S,T-рассекающим множеством может быть множество S.

Паросочетания - аналог множества дуг двудольного графа, попарно не имеющих общих вершин.

Теорема Кенинга-Эгевари. Максимальное число дуг двудольного графа G, попарно не имеющих общих вершин, равно минимальному числу вершин в некотором S, T-рассекающем множестве.

Сведем исходную задачу к табличному варианту.

Таблица, соответствующая исходному двудольному графу, строится следующим образом:

Если имеем множества $S = \{x_1, ..., x_n\}$ и $T = \{y_1, ..., y_m\}$, то таблица будет состоять из n строк, помеченных метками $x_1, ..., x_n$, и m столбцов с метками $y_1, ..., y_m$. Клетки таблицы - допустимые и недопустимые. Клетка $\{x_i, y_i\}$ допустима $\Leftrightarrow$ существует соответствующая дуга. Строки и столбцы таблицы называются рядами.

Множество рядов покрывает таблицу, если каждая допустимая клетка принадлежит хотя бы одному ряду.

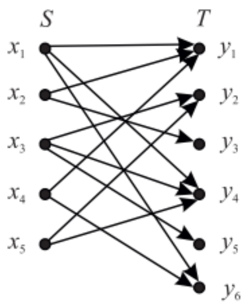
Множество допустимых клеток называется независимым, если любая пара клеток этого множества не покрывается одним рядом.

Легко видеть, что множество независимых допустимых клеток на исходном двустороннем графе соответствует множеству дуг, попарно не имеющих общих вершин.

Покрывающее множество рядов соответствует некоторому S,T-рассекающему множеству вершин.

Теорема Кенинга-Эгевари в табличном виде. Максимальное число независимых допустимых клеток равно минимальному числу рядов, покрывающих все допустимые клетки таблицы.

Пример



	y_1	y_2	y_3	y_4	y_5	y_6
x_1	1	×	×		×	
x_2		×	1	×	×	×
x_3	×	1	×			×
x_4		×	×	×	×	1
x_5	×		×	1	×	×

1- независимые клетки

Алгоритм построения ММНДК (максимального множества независимых допустимых клеток)

- *предварительный этап.*

Необходимо построить произв. независимое множество независимых допустимых клеток(можно помечать единицами): постановка 1 в любую допуст. клетку + вычеркивание соотв. строки и столбца и т. д. На очередных шагах алгоритма требуется расширить это множества или убедиться в его максимальности.

- *этап 1*

Для проверки МНДК на максимальность исп. процедура поочередных пометок строк и столбцов: вначале помечаем (черточкой) строки без 1. Теперь в любом порядке выбирается каждая помеченная строка и в них ищутся допустимые клетки. Столбец, где нашлась допустимая клетка, помечается номером просматриваемой строки. Если столбец был до этого помечен, его пометка не изменяется. Теперь просматриваются все помеченные столбцы. В каждом таком столбце ищут единицу. При ее наличии соотв. строку помечают номером просматриваемого столбца. Если в просматриваемом столбце не оказалось единицы, это означает, что МНДК может быть улучшено.

- *переход к этапу 2*

Если все помеченные столбцы просмотрены и среди них не нашлось столбца без единицы, то переходим к просмотру вновь помеченных строк. В них отыскивают допустимые клетки и помечают и помечают соотв. столбцы и т. д. Если на каком-то шаге возникает ситуация, когда никак пометок поставить больше нельзя и нет столбцы, не сод. единицу, то это говорит, что построенное на предыдущем шаге МНДК является максимальным и алгоритм заканчивает работу.Кроме ММНДК можно сразу найти минимальное число покрывающих рядов. Оно будет состоять из помеченных на последнем шаге столбцов и непомеченных строк.

- *этап 2*

Если в каком-то столбце не нашлось единиц, ставят единицу в этом столбце в клетке, соотв. номеру его пометки.В строке, где только что поставлена единица, необходимо убрать бывшую там единицу, после этого в столбце, где была единица, необходимо поставить единицу в клетке, соотв. номеру пометки этого столбца. Стоящая в этой строке прежняя единица убирается. Процесс продолж. до тех пор, пока не будет достигнута строка, первоначально помеченная черточкой. Данная процедура увеличивает количество единиц на 1. После этого все пометки строк и столбцов стираются и алгоритм повторяет свою работу с новым МНДК.

Может быть несколько ММНДК. Каждый из них можно получить, меняя порядок просмотра помеченных строк и столбцов.

24. Классическая задача о назначении, алгоритм.

Задано n работ и n исполнителей, причем каждую работу может делать любой исполнитель.Стоимость выполнения каждой i-ой работы каждым j-ым исполнителем - c_{ij}. Требуется распределить исполнителей так, чтобы все работы были сделаны, каждый исполнитель осуществлял только одну работу, каждая работа выполнялась только одним исполнителем и суммарная стоимость всех работ была минимальной. (Вместо стоимости могут быть другие критерии, например, временные).

Мат. модель задачи:

Пусть $x_{ij} = \begin{cases} 1 \text{ если } i\text{-й исполнитель назначен на } j\text{-ю работу,} \\ 0 \text{ в противном случае.} \end{cases}$

Тогда

$$\begin{cases} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min, \\ \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, 2, \ldots, n, \\ \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, 2, \ldots, n, \\ x_{ij} = 0 \vee 1. \end{cases}$$

Пусть u_i и v_j – произвольные рациональные числа

Некоторые свойства решений задачи о назначениях:

- **Теорема.** Если план X(решение) является оптимальным решением задачи, то он будет оптимальным и для задачи с целевой функцией

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(c_{ij} - u_i - v_j)x_{ij} \to \min,$$

и наоборот.

Доказательство:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(c_{ij} - u_i - v_j)x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} u_i x_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} v_j x_{ij} =$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} - \sum_{i=1}^{n} u_i \sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} v_i \sum_{i=1}^{n} x_{ij} = \sum_{i,j=1}^{n} c_{ij}x_{ij} - \sum_{i=1}^{n} u_i - \sum_{j=1}^{n} v_j.$$

, где X - оптимальный план задачи.

- **Теорема.** Если все $c_{ij} \geq 0$ и можно отыскать набор неизвестных x_{ij}, такой, что

$$\sum_i \sum_j c_{ij}x_{ij} = 0,$$

то это решение оптимально.

Алгоритм (венгерский). Пусть задана матрица стоимостей $C = [c_{ij}]_{n\times n}$.

- *Предварительный шаг*

Находим в каждом столбце минимальный элемент v_i и вычитаем его из всех элементов столбца. Затем находим минимальный элемент в каждой строке и вычитаем его из этой строки. - приведение матрицы.

- *Общий шаг*

В таблице nxn допустимые клетки соответствуют нулям в приведенной матрице стоимостей. Решаем на этой таблице задачу о ММНДК.

Если полученное ММНДК содержит n допустимых клеток, то получено оптимальное решение задачи о назначениях.

Если полученной ММНДК содержит менее n допустимых клеток, то определяем соответствующее ему минимальное множество покрывающих рядов, состоящее из непомеченных на последнем шаге алгоритма ММНДК непомеченных строк и помеченных столбцов. Покрываем этими рядами приведенную матрицу стоимостей и выбираем наименьший элемент, через который не проведена линия(ряд). Вычитаем это число из всех элементов приведенной матрицы, через которые не проведена ни одна линия и прибавляем ко всем элементам, через которые проведены 2 линии. Эта операция приведет к появлению в приведенной матрице стоимостей, по крайней мере, одного нуля. Далее общий шаг повторяется.

Пример:

$$C = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 4 & 3 & 6 & 3 \\ 2 & 2 & 1 & 2 & 4 \\ 3 & 5 & 5 & 5 & 2 \\ 4 & 3 & 4 & 5 & 1 \end{array}.$$

[после приведения] —>

$$C_1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 & 3 \\ 3 & 2 & 3 & 2 & 0 \\ 4 & 1 & 3 & 3 & 0 \end{array}.$$

решаем задачу о ММНДК на

	1	2	3	4	
1	1		×		
2		1		×	
3	×	×	×		–
4	×	×	×	1	4
				3	

Строим новую приведенную матрицу стоимостей

$$C_2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 2 & 1 \\ 2 & 0 & 0 & 0 & 4 \\ 3 & 1 & 2 & 1 & 0 \\ 4 & 0 & 2 & 2 & 0 \end{array}.$$

Новое решение задачи о ММНДК:

	1	2	3	4
1		1	×	×
2			1	×
3	×	×	×	1
4	1	×	×	

4 клетки с единицами - найдено оптимальное решение задачи о назначении.

1-ый исполнитель - 2-я работа

2-й исполнитель - 3-я работа и т. д.

Возможны задачи, в которых количество исполнителей больше количества работ. В этом случае матрицу дополняют нулями и решают обычным методом.

Если нужно решать задачу на максимум, то матрицу стоимостей умножают на -1 и, чтобы оперировать неотрицательными числами, добавляют ко всем элементам любое достаточно большое положительное число M. На матрице -C + M решается задача минимизации обычным методом.

25. Задача о назначении на "узкие места", алгоритм.

Задача о назначении на “узкие места” или же задача о назначениях с максиминной функцией цели имеет следующую общую формулировку:

Предположим, что имеется n лиц и n работ, причем известно, что каждый i-й исполнитель может выполнять любую j-ю работу с эффективностью aij (i, j = 1,2, ..., n). Требуется найти такое назначение работников на работы, чтобы наименьшая эффективность при этом назначении была максимальной.

В этой задаче, так же как и в классической задаче о назначениях, каждое назначение рабочих на работы задается отображением типа P : i → p(i), которое означает, что каждый i-й работник назначается на определенную работу p(i).

$$P = \begin{pmatrix} 1 & 2 & \dots & n \\ p(1) & p(2) & \dots & p(n) \end{pmatrix}.$$

Цель - найти такое отображение P, при котором $F(P) = min\{a_{1p(1)}, \dots, a_{np(n)}\}$ максимально, т. е. требуется отыскать $max\, F(P),\ \forall P.$

Алгоритм:

Выбирается произвольное назначение в виде подстановки.

$$P_0 = \begin{pmatrix} 1 & 2 & \dots & n \\ p_0(1) & p_0(2) & \dots & p_0(n) \end{pmatrix}.$$

При этом назначении вычисляется значение целевой функции, т. е. величина
$F(P_0) = min\{a_{1p_0(1)}, \dots, a_{np_0(n)}\}.$

Поскольку требуется найти отображение, максимизирующее F(P), клетки таблицы с n строками и n столбцами можно считать допустимыми или недопустимыми путем сравнения конкретной эффективности соответствующего назначения с достигнутой величиной F(P0). Клетка (i, j) будет допустимой в том и только в том случае, когда величина aij > F(P0). На построенной таким образом таблице находим ММНДК. Для того чтобы ММНДК определяло подстановку, число выбранных клеток в этой ММНДК должно быть в точности равно n. Координаты тех клеток зададут какую-то новую подстановку P1, причем по определению допустимых клеток таблицы F(P1) > F(P0). Далее процесс повторяется для новой подстановки. Если число клеток в ММНДК для некоторой очередной подстановки Pk меньше n, то предыдущее назначение оптимально и улучшить его нельзя.

Пример.

Дана матрица эффективностей:

$$
A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{pmatrix}
1 & 3 & 2 & 6 & 0 & 1 \\
4 & 2 & 3 & 8 & 3 & 1 \\
8 & 1 & 1 & 5 & 0 & 9 \\
3 & 4 & 4 & 8 & 8 & 3 \\
2 & 9 & 9 & 5 & 2 & 9 \\
3 & 3 & 3 & 6 & 7 & 1
\end{pmatrix}
\end{array}.
$$

Выбираем произвольное начальное значе Iоние:

$$
P_0 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 4 & 3 & 6 & 5 \end{pmatrix},
$$

тогда $F(P_0) = min(3,\ 4,\ 5,\ 4,\ 9,\ 7) = 3.$

Строим таблицу допустимых и недопустимых клеток и решаем на ней задачу о ММНДК:

	1	2	3	4	5	6
1	×	×	×	1	×	×
2	1	×	×		×	×
3		×	×		×	1
4	×	1				×
5	×		1		×	
6	×	×	×		1	×

Такому решению соответствует назначение

$$
P_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 6 & 2 & 3 & 5 \end{pmatrix},
$$

для которого получаем $F(P1) = min(6,\ 4,\ 9,\ 5,\ 9,\ 7) = 4.$

Если попробуем продолжить строить таблицу, получаем следующий вариант:

	1	2	3	4	5	6
1	×	×	×		×	×
2	×	×	×		×	×
3		×	×		×	
4	×	×	×			×
5	×				×	
6	×	×	×		1	×

В этой таблице меньше 6 допустимых клеток, следовательно решение P_1 оптимально.

26. Задача о кратчайших путях, алгоритм Дейкстра.

Пусть моделью некоторой практической ситуации **является ориентированная сеть** G = (V, U), V = (v1, v2, ..., vn) и каждой дуге вида (vi , vj)∈U поставлено в соответствие определенное рациональное число dij, названное длиной (или «весом») этой дуги.

Путем из вершины vi до вершины vj называется (как и ранее) последовательность дуг ((vi , v1), (v1, v2), ..., (vk, vj)), в которой вершина-конец каждой предыдущей дуги является началом последующей. Длиной пути в этом случае будем называть сумму длин принадлежащих ей дуг.

Задача о кратчайшем пути состоит в определении пути минимальной длины между двумя вершинами. Эта задача может не иметь решения (быть некорректной) только в случае, если в сети существуют контуры с суммарной отрицательной длиной. Попадая на такой контур, можно обходить его сколь угодно раз, тем самым получая путь со сколь угодно малым «весом»

При <u>неотрицательных</u> дугах алгоритм Дейкстры используют для решения следующих вариантов задачи:

а) найти кратчайший путь между двумя фиксированными вершинами vi и vj;

б) найти кратчайший путь от фиксированной вершины vi до всех остальных вершин сети;

Алгоритм Дейкстры:

Сущность метода состоит в приписывании пометок вершинам сети, начиная от вершины v1. От нее двигаемся по направлению дуг, и тем самым помеченными могут быть только вершины, достижимые с помощью путей из вершины v1.

Обозначим через W ⊆ V множество вершин, пометки которых уже не будут изменяться на последующих шагах. Вначале W = {v1}. Пометку ui приписываем вершине vi. Она означает длину кратчайшего пути от v1 до vi, проходящего только через вершины из множества W. Вначале u1 = 0, ui = d1i, для всех i = 2, ..., m, причем если дуги (v1, vi) не существует, то полагаем, что d1i – достаточно большое число, например ∞. На очередном шаге находим вершину vr ∈ V \W с наименьшей пометкой. Эту вершину присоединяем к множеству W и запоминаем индекс r. Далее для остальных вершин vi ∈ V \W проверяем неравенство ui > ur + dri. Если оно выполняется, то вершине vi приписываем новую пометку, равную ur + dri.

Если это неравенство не выполняется, то пометку вершины не изменяем. Такой шаг повторяем до тех пор, пока не получим W = V. Не трудно доказать, что пометки ui, полученные для всех вершин на последнем шаге метода, будут длинами кратчайших путей от вершины v1 до всех остальных вершин сети.

Замечание. Если при определении вершины v_r ∈ V \ W с наименьшей пометкой имеется несколько вариантов, то выбираем любой из них. Выбор другого варианта приведет в конечном итоге к построению иного кратчайшего пути такой же длины.

Число шагов в методе Дейкстры будет не больше, чем число n вершин сети, при этом на шаге перебирается не более чем n пометок. Таким образом, эффективность метода Дейкстры оценивается как n^2 обобщенных операций.

Замечание. Если сеть содержит ненаправленные дуги, т. е. ребра, то ее можно превратить в ориентированную сеть путем замены каждого ребра (v_i, v_j) на две противоположно направленные дуги с одинаковым «весом» dij = dji. Такое преобразование не увеличит число шагов в методе Дейкстры.

27. Задача о кратчайших путях, алгоритм Флойда.

Пусть моделью некоторой практической ситуации **является ориентированная сеть** G = (V, U), V = (v1, v2, ..., vn) и каждой дуге вида (vi , vj)∈U поставлено в соответствие определенное рациональное число dij, названное длиной (или «весом») этой дуги.

Путем из вершины vi до вершины vj называется (как и ранее) последовательность дуг ((vi , v1), (v1, v2), ..., (vk, vj)), в которой вершина-конец каждой предыдущей дуги является началом последующей. Длиной пути в этом случае будем называть сумму длин принадлежащих ей дуг.

Задача о кратчайшем пути состоит в определении пути минимальной длины между двумя вершинами. Эта задача может не иметь решения (быть некорректной) только в случае, если в сети существуют контуры с суммарной отрицательной длиной. Попадая на такой контур, можно обходить его сколь угодно раз, тем самым получая путь со сколь угодно малым «весом»

Замечание. Подобно задаче о кратчайшем пути можно сформулировать задачу о наидлиннейшем пути.

Все возможные варианты задачи о кратчайшем пути:

а) найти кратчайший путь между двумя фиксированными вершинами vi и vj ;

б) найти кратчайший путь от фиксированной вершины vi до всех остальных вершин сети;

в) найти кратчайший путь между всеми парами вершин.

Для решения задач в вариантах **а)** и **б)** часто используют **алгоритм Дейкстры.**

Метод Флойда – Уоршелла.

Этот метод применяется для решения задачи о кратчайших путях в варианте **в)** для графов с n вершинами, причем он может работать с отрицательными длинами дуг и позволяет установить факт наличия цикла отрицательной длины.

Метод преобразует исходную матрицу $D = [d_{ij}]_{n\times n}$, где элемент d_{ij} равен длине дуги между вершинами v_i и v_j , если она существует, или d_{ij} = ∞, если такой дуги нет.

Обозначим через $d_{ij}^{(k)}$ длину кратчайшего пути от вершины vi до вершины vj , у которой все промежуточные вершины принадлежат множеству {1, 2, ..., k}. Для k = 0 принимаем $d_{ij}^{(0)} = d_{ij}$

Матрицы $D^{(k)} = \left[d_{ij}^{(k)}\right]_{n\times n}$ определяем последовательно для k = 1, 2, ..., n, применяя формулу

$$d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}.$$

Элементы матрицы $D^{(n)} = [d_{ij}^{(n)}]_{n\times n}$ являются длинами кратчайших путей между соответствующими парами вершин.

Если при вычислениях для какого-то k окажется, что один из диагональных элементов матрицы $D^{(k)}$, например $d_{hh}^{(k)}$, является отрицательным, то это означает, что в графе есть цикл отрицательной длины, содержащий вершину v_h.

Для восстановления кратчайших путей между каждой парой вершин графа можно параллельно с вычислением матрицы $D^{(k)}$ вычислять так называемые матрицы предшественников $E^{(k)}$ для *k* = 0, 1, 2, ..., *n*. Для *k* = 0 полагаем

$$e_{ij}^{(0)} = \begin{cases} 0, & \text{если} \quad i = j \quad \text{или} \quad d_{ij} = \infty, \\ i, & \text{если} \quad i \neq j \quad \text{и} \quad d_{ij} < \infty. \end{cases}$$

Элементы $e_{ij}^{(k)}$ для *k* = 1, 2, ..., *n* определяются в зависимости от того, какой из аргументов под знаком минимума в формуле $d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}.$ выбирается: если выбирается первый, то $e_{ij}^{(k)} = e_{ij}^{(k-1)}$ если выбирается второй аргумент то $e_{ij}^{(k)} = k$.

При восстановлении кратчайшего пути для конкретной пары вершин *(i, j)* в матрице $E^{(n)}$ находится элемент $e_{ij}^{(n)} = k_0$ который является вершиной – предшественником вершины *j* на кратчайшем пути из *i* до *j*. Далее на $E^{(n)}$ устанавливается вершина – предшественник вершины k_0 на кратчайшем пути из *i* до k_0 и т. д., пока не достигается вершина *i*.

28. Минимальные остовные деревья, алгоритм. Задача Штейнера.

Если **имеется неориентированный связный граф** G = (V, U), то, как уже отмечалось, из него можно последовательно удалять некоторые ребра так, чтобы сохранялась связность, до тех пор пока не получится дерево. Такое дерево называется **остовным**. Понятно, что в зависимости от выбора удаляемых ребер можно получить различные остовные деревья. Если при этом каждому ребру *(vi , vj)* исходного графа поставлено в соответствие рациональное число d_{ij} - длина этого ребра, то каждому остовному дереву будет соответствовать его **длина**, **как сумма длин** принадлежащих ему **ребер**. Для данного графа возникает **задача о нахождении минимального остовного дерева.**

Теорема 3.8.

Остовное дерево *T* будет минимальным тогда и только тогда, когда для любого ребра *(vi , vk)* вне дерева выполняется условие:

$$d_{ik} \geq \max(d_{i1}, d_{12}, ..., d_{k-1,k}), \quad (3.19)$$

где последовательность *[vi , v1, v2, ..., vk–1, vk]* является цепью (единственной) в дереве *T*, соединяющей вершины *vi* и *vk*.

Условие (3.19) говорит о том, что для минимального остовного дерева длины ребер вне этого дерева будут не меньшими, чем длины ребер в этом дереве.

Метод Краскала.

В неориентированном графе G = (V, U) выбираем ребро наименьшей длины; на каждом очередном шаге среди еще не просмотренных ребер выбираем ребро наименьшей длины и, если оно не образует цикла с уже выбранными ребрами, добавляем его к множеству уже выбранных ребер. Когда это множество будет содержать ровно *n – 1* ребер $(n = |V|)$, то тем самым оно будет минимальным остовным деревом.

Замечание. Легко модифицировать теорему – критерий минимальности и метод Краскала для построения максимальных остовных деревьев.

Замечание. Минимальное остовное дерево, построенное для связного неориентированного графа, часто можно использовать для решения задачи о кратчайших путях между всеми парами вершин. Каждая пара вершин в этом остовном дереве соединена единственным кратчайшим путем.

Существуют различные **важные** для практики **модификации** задачи о минимальном остовном дереве, например задача Штейнера и задача о минимальном покрывающем дереве на прямоугольной метрике.

Задача Штейнера.

Имеется *n* точек, случайно разбросанных на плоскости с евклидовой системой координат и, соответственно, с евклидовой метрикой. Требуется соединить эти точки покрывающим деревом минимальной суммарной длины, причем допускается использование вспомогательных точек. Такая задача эффективно **решена только для случаев n** = 3 и **n** = 4. Например, при n = 3 минимальное дерево Штейнера строится следующим образом. Доказано, что если исходные точки образуют остроугольный треугольник, то требуется одна вспомогательная точка и минимальное покрывающее дерево соединяет эту точку с каждой из исходных вершин так, чтобы каждая сторона треугольника была «видна» из вспомогательной точки под углом 120° (рис. 3.22).

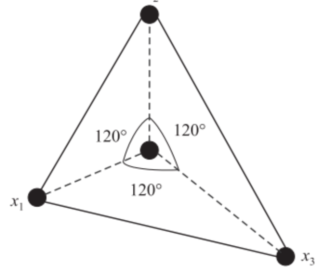


Рис. 3.22

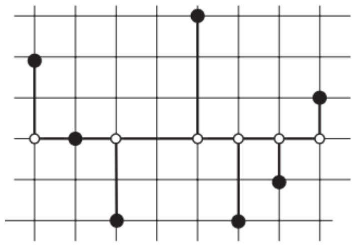
В этом случае минимальное покрывающее дерево Штейнера будет короче длины минимального остовного дерева, найденного алгоритмом Краскала. В случае когда исходные точки образуют треугольник с прямым или тупым углом, вспомогательная точка совпадает с вершиной такого угла и длина дерева Штейнера равна длине дерева Краскала. При n = 4 в общем случае требуются две вспомогательные точки – и задача Штейнера также решена с помощью эффективного алгоритма. Для произвольного n доказано, что вспомогательных точек должно быть не более чем n – 2. Однако эффективного алгоритма для построения дерева Штейнера в общем случае пока не найдено.

Задача о минимальном покрывающем дереве при прямоугольной метрике.

Имеется n точек, случайно разбросанных на плоскости. Эти точки являются узлами прямоугольной решетки. Расстояние между точками, например, *v1* с координатами *(x1, y1)* и *v2* с координатами *(x2, y2)* вычисляются по формуле:

$$d(v_1,\ v_2) = |x_1 - x_2| + |y_1 - y_2|.$$

Требуется соединить эти точки покрывающим деревом так, чтобы суммарная длина всех горизонтальных и вертикальных отрезков этого дерева была минимальна. В этой задаче также допускается использование вспомогательных точек, которые также будут узлами прямоугольной решетки.



(Пример)

Поставленная задача имеет практическое применение, особенно при проектировании интегральных схем. К сожалению, **для общего случая пока не разработаны эффективные алгоритмы ее решения.**