# Математические методы сетевого моделирования

## Основные понятия. Примеры применения сетей (графов). Теория графов. Дерево принятия решений

**Граф** (**сеть**) – это множество элементов (**вершин**, **узлов**), соединенных между собой линиями связи (**ребрами**, **дугами**).

Чаще всего термин «граф» используют в математике, а «сеть» – в практических приложениях (менеджменте, экономике, информатике).

Основное назначение графов – отображение взаимосвязей между объектами.

И вершины, и ребра обычно обозначают – буквами, числами, развернутыми подписями. На схемах ребра показывают кружками, прямоугольниками, пиктограммами и т.д., ребра – прямыми, ломаными или изогнутыми линиями.

Примеры графов:













### Примеры практического применения графов

1. *Транспортные* сети: вершинами являются некоторые пункты, ребрами – пути перемещения между ними (автомобильные, железные дороги, линии электропередач, трубопроводы, компьютерные сети и т.п.). Задачи логистики, организации грузоперевозок, снабжения, прокладки маршрутов.
2. *Технологические задачи*: вершины – производственные элементы (заводы, цеха, станки и т.п.), дуги – потоки сырья, материалов, финансов, продукции. Задача – определение оптимальной загрузки для максимизации объемов производства, прибыли.
3. *Обменные схемы*: моделируют бартерные схемы, взаимозачеты и пр. Вершины – участники обмена, дуги – потоки материальных и финансовых ресурсов.
4. *Управление проектами*: календарное планирование работ в ходе выполнения проекта. Дополнительно к срокам выполнения работ может учитываться их стоимость, качество, необходимые ресурсы. Работы могут быть представлены как вершинами, так и дугами. Раздел – *сетевое планирование и управление* (СПУ).
5. *Модели коллективов и групп*: применяется в социологии. Вершины – люди или группы людей, ребра – отношения между ними (знакомство, доверие, симпатии, лидерство и др.). Решаются задачи исследования структуры социальных групп, их сравнения, определения агрегированных показателей, отражающих степень напряженности, согласованности взаимодействия и др.
6. *Модели организационных структур*: вершины – должностные лица, отделы, филиалы и другие организационные элементы, ребра – управленческие, информационные, финансовые, административные связи между ними.
7. *Блок*-*схемы* алгоритмов: изображают последовательность действий для решения какой-либо задачи. Чаще всего применяются в программировании, но с помощью блок-схемы можно представить практически любую последовательность действий. Вершинами будут действия и условия, а дугами – переходы между ними.

### Основные понятия

Графы изучает специальный раздел дискретной математики – **теория графов**.

Вершины, соединенные одним ребром, и ребра, выходящие из одной вершины, называются **смежными**. Если ребро соединено с вершиной, то оно называется **инцидентным** этой вершине. Число ребер, выходящих из вершины, называется ее **степенью**.

3

1

2

4

Вершины A и B – смежные. Ребра 2,3,4 – смежные. Ребра 2,3,4 инцидентны вершине С, ребро 1 инцидентно вершинам A и B. Степени вершин A и B раны 1, степень вершины C равна 3

Различают **связные** и **несвязные** графы. В связанном графе все вершины имеют связи, а несвязный состоит из отдельных, не соединенных между собой подграфов.

а

б

Граф (а) – связный, граф (б) – несвязный. По сути, (б) представляет собой два отдельных графа. изображенных вместе.

Связи в графе могут быть направленными или ненаправленными:

* **ориентированный** граф (**орграф**) – связи называются *дугами* и обозначаются стрелками;
* **неориентированных** граф – связи называются *ребрами* и изображаются линиями без стрелок.

Возможен и смешанный вариант, когда на одном графе есть и дуги, и ребра. Например, карта, на которой часть дорог с односторонним движением.

Для каждого ребра может быть задано число, являющееся ее качественной характеристикой – **длина** (**вес)**. Это может быть расстояние между городами, время пути от одной вершины до другой, стоимость перевозки или проезда, пропускная способность (трубы, канала связи) и т.д. На графе длину ребра подписывают рядом с ним, можно в скобках. Длина ребра не обязательно равна длине самой линии на рисунке. Можно каждому ребру присвоить несколько разных (по сути) длин. Например, одновременно указать расстояние в километрах и время пути в часах.

(19)

(153; 1,7)

В графах могут присутствовать:

* кратные ребра – ребра, соединяющие одни и те же вершины;

дуги кратные

дуги **не** кратные

кратные ребра

* петля – ребро, соединяющее вершину саму с собой; при вычислении степени вершины петля считается за 2;

петля – степень вершины равна 4

* **циклы** – последовательность смежных ребер, начинающаяся и заканчивающаяся в одной вершине.

*В этом графе один цикл*

*В этом графе три цикла*

**Путь** или **маршрут**– это последовательность дуг, такая, что конец одной дуги является началом другой дуги. В неориентированном графе аналогичная последовательность ребер называется **цепью**.

Если маршрут начинается и заканчивается в одной и той же вершине, то он называется **контуром**. Замкнутая цепь – это **цикл**.

**Длина** **маршрута** определяется либо числом входящих в него ребер, либо суммой их длин, если они заданы. Большое число практических задач связано с поиском кратчайшего маршрута, минимальной стоимости или, наоборот, максимального потока.

(8)

(1)

(5)

(2,5)

(4)

(6)

(12)

(5)

(2)

Длина маршрута A-C-E-G = 1 + 6 + 12 = 19. Это **не** самый короткий из возможных путей от A до G.

В некоторых задачах длина пути может вычисляться не как сумма длин ребер, а иначе. Например, длина ребра может представлять собой вероятность успешной доставки.

(0,98)

(0,95)

В данном примере 0,98 и 0,95 – вероятности доставки из A в B и из B в C. Вероятность доставки из A в C рассчитывается по формуле
$1–\left(1–0,98\right)-\left(1–0,95\right)=1-0,02-0,05=0,93$.

### Частные виды графов

**Полный граф** – граф, в котором все вершины смежные. Степень каждой вершины равна числу вершин в графе.

**Двудольный граф** – все вершины такого графа можно разбить на две подгруппы, при этом любая вершина из одной подгруппы будет связана со всеми вершинами другой подгруппы.

**Регулярный** **граф** – все вершины имеют одинаковую степень. Все полные графы являются регулярными. Частный случай – **циклический** **граф** (степень всех вершин равна 2).

циклический, регулярный, полный, степень 2

циклический, регулярный, степень 2

регулярный, степень 3

**Дерево** – связный граф без циклов. Число ребер в дереве всегда на 1 меньше числа вершин.

дерево, вершин 9, ребер 8

Деревья – вид графов, часто использующийся на практике для представления иерархических структур. В виде деревьев изображают классификации, организационные структуры предприятий, родословные и др. *Деревья принятия решений* кратко рассматриваются в конце лекций.

Иногда сетью называют такой граф, который не является деревом, т.е. в нем могут быть циклы.

### Способы представления графов

Один и тот же по сути граф можно изобразить по-разному, расположив его вершины тем или иным способом. Такие графы называют **изоморфными**(графы с одинаковым набором вершин и соединяющих их дуг/ребер).

1

2

3

4

5

2

2

2

1

1

1

3

3

3

4

4

4

5

5

5

2

1

3

4

5

Изоморфные графы. По сути, это один и тот же граф, нарисованный по-разному.

Желательно выбирать наиболее наглядное представление графа:

* минимизировать число пересечений ребер (если пересекающиеся ребра отсутствуют, то такой граф называют *планарным*);
* связанные вершины располагать поблизости друг от друга;
* унификация направлений стрелок орграфа (стрелки направлены вниз или влево);
* симметричное расположение вершин;
* длина ребер на рисунке приблизительно соответствует их весам;
* не загромождать граф надписями;
* если граф достаточно большой, его можно разбить на несколько частей.

Помимо графического, существуют и другие способы представления графов.

**Список вершин** – перечисляются все вершины, и для каждой указывается, с какими вершинами она соединена.

Список вершин для графа с предыдущей картинки:

|  |  |
| --- | --- |
| Вершина | Смежные вершины |
| 1 | 2, 3 |
| 2 | 1, 3, 4 |
| 3 | 1, 2, 5 |
| 4 | 2, 5 |
| 5 | 3, 4 |

**Список ребер** – аналогично для ребер или дуг. Часто ребра называются по вершинам, которые они соединяют, тогда достаточно их просто перечислить.

Список ребер для того же графа:

*a* = (1; 2)

*b* = (1; 3)

*c* = (2; 3)

*d* = (2; 4)

*e* = (3; 5)

*f* = (4; 5)

**Матрица инцидентности** – каждый столбец соответствует вершине, каждая строка – ребру. Если ребро инцидентно вершине, то соответствующий элемент равен 1. Если ребро является петлей – то 2. Для направленных графов –1 означает, что ребро выходит из вершины, +1 – что входит. Тогда сумма всех элементов матрицы должна быть равна 0.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ВершинаРебро | 1 | 2 | 3 | 4 | 5 |
| *a* | 1 | 1 | 0 | 0 | 0 |
| *b* | 1 | 0 | 1 | 0 | 0 |
| *c* | 0 | 1 | 1 | 0 | 0 |
| *d* | 0 | 1 | 0 | 1 | 0 |
| *e* | 0 | 0 | 1 | 0 | 1 |
| *f* | 0 | 0 | 0 | 1 | 1 |

Для человека такой способ не очень удобен, но он часто используется для компьютеров.

**Матрица смежности** – квадратная матрица, и столбцы и строки которой соответствуют вершинам графа, а элементы равны числу ребер, соединяющих эти вершины, или длине соответствующего ребра. Для неориентированных графов матрица будет симметричной, поэтому достаточно заполнить ее половину.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вершина | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 2 |  | 1 | 1 | 1 | 0 |
| 3 |  |  | 1 | 0 | 1 |
| 4 |  |  |  | 1 | 1 |
| 5 |  |  |  |  | 1 |

В этой матрице хорошо видно, сколько ребер соединяет вершины или какова их длина.

Для орграфов считается, что в строках указываются вершины, из которых дуги исходят, а в столбцах – в которые входят (или наоборот).

## Транспортные сети и потоки

Транспортные сети, пожалуй, одно из самых распространенных и высокоавтоматизированных приложений математического сетевого моделирования. Они используются в навигаторах, в логистике, в электрических и компьютерных сетях, в коммунальных службах, при прокладке трубопроводов и маршрутов городского транспорта и др.

### Задачи о соединении

В задаче о соединении требуется обеспечить минимальную стоимость проекта по созданию сети, если известны затраты на соединение каждой пары пунктов:

* строительство сети автомобильных или железных дорог между городами;
* строительство сети нефтепроводов и газопроводов;
* прокладка кабельной сети между некоторыми географическими точками;
* планирование авиарейсов;
* и др.

Все возможные соединения пунктов можно представить в виде графа, каждому ребру которого назначен вес (стоимость). Из всех этих вариантов соединения, нужно выбрать самый дешевый, который соединит все пункты.

**Пример 1. Задача о соединении городов.**

Известна стоимость постройки дорог между каждой парой населенных пунктов. Необходимо построить дорожную сеть наименьшей стоимости, соединяющую все города.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поселок | A | B | C | D |
| A |  | 11 | 13 | 10 |
| B |  |  | 6 | 9 |
| C |  |  |  | 10 |
| D |  |  |  |  |

Этот граф можно изобразить следующим образом (если бы у нас была карта, можно было бы расположить поселки по ней):

A

B

C

D

(11)

(13)

(10)

(6)

(9)

(10)

Чтобы решить задачу о соединении необходимо найти *остов наименьшего веса*.

**Остов** графа – это такая его часть (подграф), которая включает все его вершины и является связной (деревом).

**Алгоритм Краскала (Крускала)** для построения остова наименьшего веса**.** Всегда содержит (n – 1) шагов, n – число вершин. На каждом шаге выбирается ребро, которое следует включить в остов.

Шаг 1. Выбрать самое короткое ребро в графе.

Шаг 2...(n – 1). Из оставшихся ребер выбрать самое короткое ребро, которое не образует цикла с уже выбранными ребрами.

Для удобства можно на каждом шаге вычеркивать ребра, которые образуют цикл с уже имеющимися.

**Пояснение к решению** для **примера 1.**

В графе 4 вершины, значит, алгоритм займет 3 шага.

Шаг 1. Самое короткое ребро B-C.

A

B

C

D

11

13

10

(6)

9

10

Шаг 2. Самое короткое ребро B-D, оно не образует цикла с B-C. Теперь мы можем исключить из рассмотрения ребро C-D, т.к. оно образует цикл B-C-D.

A

B

C

D

(11)

(13)

(10)

(6)

(9)

Шаг 3 (последний). Самое короткое из оставшихся ребер A-D, оно не образует цикла.

A

B

C

D

(11)

(13)

(10)

(6)

(9)

Таким образом, для соединения этих поселков необходимо проложить дороги от A-D, B-D, B-C. Общая стоимость строительства S = 9+6+10=25.

**Решение:**

1. Добавляем B-C.
2. Добавляем B-D.
3. Добавляем A-D.

S= 9+6+10=25.

**Ответ:**

S = 25.

A

B

C

D

(10)

(6)

(9)

Задачу можно решать и «от обратного», удаляя из сети наиболее «дорогие» ребра.

**Пример 2.**

Построить остов минимального веса для графа, показанного на рисунке, удалив «лишние» ребра. Обратите внимание, граф неполный – не все вершины соединены друг с другом напрямую. Т.е. некоторые «дороги» невозможно построить в принципе. В матрице смежности такие ребра обозначаются прочерком или знаком бесконечности ∞.

(6)

(5)

(3)

(1)

(4)

(8)

(5)

(3)

(2)

(6)

**Алгоритм**

Найти ребро наибольшего веса. Если его удаление не нарушает связности графа, удалить это ребро.

Продолжать до тех пор, пока граф не станет деревом (не останется n–1 ребро).

**Пояснения к решению:**

Сначала изобразим сеть более наглядно и обозначим вершины:

(6)

(5)

(3)

(1)

(4)

(8)

(5)

(3)

(2)

(6)

A

C

D

E

G

F

B

Всего в графе 7 вершин и 10 ребер. Значит, необходимо удалить 4 ребра. Алгоритм Краскала занял бы 6 шагов.

Шаг 1. Удалим из графа самое длинное ребро D-G (8).

(6)

(5)

(3)

(1)

(4)

(5)

(3)

(2)

(6)

A

C

D

E

G

F

B

Шаг 2. Самые длинные ребра (6) – это A-B и B-E. Однако удаление A-B нарушит целостность графа, вершина A останется без соединения. Удаление B-E целостность графа не нарушает.

(6)

(5)

(3)

(1)

(4)

(5)

(3)

(2)

A

C

D

E

G

F

B

Шаги 3 и 4. Следующие кандидаты на удаление – B-C и E-G (5). Удаление любого из этих ребер не нарушает целостность графа, поэтому шаги 3 и можно объединить, удалив два ребра сразу.

(6)

(3)

(1)

(4)

(3)

(2)

A

C

D

E

G

F

B

В графе осталось 6 ребер, алгоритм завершен, и мы получили остов наименьшего веса. Суммарная стоимость S = 6 + 1+ 3 + 4 + 3 + 2 = 19.

**Решение:**

1. D-G (8).
2. B-E (6).
3. B-C (5).
4. E-G (5).

S = 6 + 1+ 3 + 4 + 3 + 2 = 19.

**Ответ:** S = 19.

(6)

(3)

(1)

(4)

(3)

(2)

A

C

D

E

G

F

B

### Кратчайший маршрут

Найти кратчайший путь от одной вершины и до всех остальных вершин (или до другой конкретной вершины).

Именно эта задача решается при прокладке маршрута на навигаторе, установлении телефонного или интернет-соединения. Она может возникать и при принятии экономических и управленческих решений. Граф может описывать, например, возможные пути модернизации производства.

**Волновой метод (алгоритм Дейкстра).** Все ребра в сети должны иметь неотрицательный вес (≥0).

В алгоритме поочередно отмечаются вершины. Отметить вершину – значит найти самый короткий путь до нее, *отметка* – длина маршрута до вершины и номер предыдущей вершины (W;j).

Шаг 1. Отмечаем начальную вершину.

Шаг 2. Среди неотмеченных вершин, смежных с уже отмеченными, отметить ту, к которой ведет наименьший путь.

Шаг 3. Повторять шаг 2 до тех пор, пока не будут отмечены все вершины (или пока не будет достигнута целевая вершина).

Аналогично можно найти путь максимальной длины, только выбирать наибольший путь на шаге 2.

Необходимо учитывать, что не во всех задачах длины ребер нужно складывать, чтобы получить длину маршрута.

**Пример 3**.

Найти кратчайшее расстояние от вершины a до вершины e.

154

289

105

130

73

85

167

211

1270

**Пояснения к решению:**

Шаг 1. Отмечаем начальную вершину a(0;-).

154

289

105

130

73

85

167

211

1270

Шаг 2. Из a можно попасть в b и d.

W(a-b) = 154

W(a-d) = 289

Кратчайший путь ведет в b(154;a).

154

289

105

130

73

85

167

211

1270

Шаг 3. Из b можно попасть в c и d. Добавляем пути:

W(a-b-c) = 154 + 105 = 259

W(a-b-d) = 154 + 130 = 284

Отмечаем вершину c(259;b).

154

289

105

130

73

85

167

211

1270

Шаг 4. Из c можно попасть в d, e, f.

W(a-b-c-d) = 259 + 73 = 332

W(a-b-c-e) = 259 + 211 = 470

W(a-b-c-f) = 259 + 85 = 334

Кратчайший из всех путей – a-b-d. Отмечаем вершину d(284;b). Обратите внимание, что хотя мы уже можем попасть в вершину e, мы еще не можем быть уверены, что путь a-b-c-e – кратчайший.

154

289

105

130

73

85

167

211

1270

Шаг 5. Из d можно попасть в e.

W(a-b-d-e) = 284 + 167 = 451

Теперь самый короткий путь: W(a-b-c-f) = 334

154

289

105

130

73

85

167

211

1270

Шаг 6. Из f тоже можно попасть в e.

W(a-b-c-f-e) = 334 + 127 = 461

Кратчайший из всех путей – W(a-b-d-e) = 451. Мы отметили вершину e, задача решена.

**Решение:**

|  |  |  |
| --- | --- | --- |
| Шаг | Новые пути | Отмеченные вершины |
|  |  | a(0;-) |
|  | W(a-b) = 154W(a-d) = 289 | b(154;a) |
|  | W(a-b-c) = 154 + 105 = 259W(a-b-d) = 154 + 130 = 284 | c(259;b) |
|  | W(a-b-c-d) = 259 + 73 = 332W(a-b-c-e) = 259 + 211 = 470W(a-b-c-f) = 259 + 85 = 334 | d(284;b) |
|  | W(a-b-d-e) = 284 + 167 = 451 | f(334;c) |
|  | W(a-b-c-f-e) = 334 + 127 = 461 | e(451;d) |

**Ответ:**

W(a-b-d-e) = 332.

154

289

105

130

73

85

167

211

1270