# Математические методы сетевого моделирования

## Транспортные сети и потоки (продолжение)

**Максимальный поток**

Пример – город, находящийся на двух сторонах реки, в котором есть три моста. Насколько хорошей бы ни была транспортная сеть на каждой стороне реки, максимальный транспортный поток с одной стороны на другую ограничивается пропускной способностью мостов.

Мосты в являются *разрезом* данной сети, скорее всего, *критическим*. Хотя возможна ситуация, когда очень хороший мост выходит на узкую улицу. В этом случае *критический* *разрез* будет другим.

**Транспортная сеть** – ориентированный граф, в котором имеется единственный **источник** и единственный **сток**, а каждой дуге назначена **пропускная способность**.

**Источник** *S* – вершина, из которой дуги только исходят.

**Сток** *T* – вершина, в которую дуги только входят.

**Пропускная способность** *c*(*i*,*j*) – максимальный **поток** *f*(*i*,*j*) (вещества, транспорта, груза), который можно пустить по дуге (*i*,*j*).

Реальный поток и пропускная способность записываются на графе через дробь:

*f* / *c*

Если реальный поток по дуге равен ее пропускной способности, то она называется **насыщенной**.

3/3

2/2

0/2

3/8

2/9

Какие дуги являются насыщенными?

**Правило сохранения потока** – сумма потоков, входящих в вершину, равна сумме исходящих (кроме S и T):



**Общий поток сети F –** суммарный поток, который передается по сети от источника до стока. Общий поток, выходящий из источника, равен потоку, входящему в сток:



**Разрез (**не обязательно критический**)** транспортной сети можно получить, если разбить все ее вершины на две группы: в одну должен входить источник, а в другую сток. Разрезом будут являться дуги, которые исходят из вершин первой группы, а входят в вершины второй.

Примеры разрезов:

**Разрез 1:**

1 группа вершин: S, 1

2 группа вершин: T, 2, 3, 4

S-3

1-2

1-3

**Разрез 2:**

1 группа вершин: S, 1, 3

2 группа вершин: T, 2, 4

1-2

3-2

3-4

**Разрез 3:**

1 группа вершин: S, 1, 3, 2, 4

2 группа вершин: T

2-T

4-T

Разрез можно понимать так: это такой набор дуг, в обход которого нельзя добраться от источника до стока.

**Критическим** будет разрез с минимальной суммарной пропускной способностью дуг. Критический разрез определяет общий **максимальный поток** по сети.

Существует много (более 20) методов нахождения максимального потока. Однако в любом случае задачу приходится решать итеративно, т.е. берется какой-то начальный поток и он постепенно увеличивается, пока не станет максимальным.

**Алгоритм Форда-Фалкерсона**:

1. Исходный поток – нулевой (0 по всем дугам).
2. Находим *увеличивающий путь* из источника в сток (лучше всего кратчайший). *Допустимые дуги*:
	1. направление дуги совпадает с направлением потока и поток по этой дуге меньше её пропускной способности;
	2. направление дуги противоположно направлению потока и поток по этой дуге больше нуля.
3. Пускаем через найденный путь максимально возможный поток:
	1. На найденном пути в остаточной сети ищем ребро с минимальной пропускной способностью cmin.
	2. Для каждого ребра на найденном пути с направлением увеличиваем поток на cmin, а в противоположном ему – уменьшаем на cmin.
	3. Модифицируем *остаточную сеть*. Для всех рёбер на найденном пути, вычисляем новую пропускную способность. Если она стала ненулевой, добавляем ребро к остаточной сети, а если обнулилась, стираем его.
4. Если увеличивающих путей больше нет, задача решена. Иначе повторить с п.2.

**Пример**. Найти максимальный поток и критический разрез. Как можно увеличить максимальный поток на 1?

9

6

10

3

4

7

4

8

**Решение:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дуги | *с* | I | II | III |
| S-A | 9 | +3 |  | +6 |
| S-C | 8 |  | +4 |  |
| A-B | 6 |  |  | +6 |
| A-D | 3 | +3 |  |  |
| B-T | 10 |  |  | +6 |
| C-D | 4 |  | +4 |  |
| D-B | 4 |  |  |  |
| D-T | 7 | +3 | +4 |  |

1. S-A-D-T (3)
2. S-C-D-T (4)
3. S-A-B-T (6)

9/9

6/6

6/10

3/3

0/4

7/7

4/4

4/8

**Ответ:**

Критический разрез: S-A, C-D

F = 9 + 4 = 6 + 7 = 13

Чтобы увеличить максимальный поток на 1, необходимо увеличить пропускную способность дуги C-D на 1.

Алгоритм поиска **кратчайшего увеличивающего пути**:

1. Отметить источник S (поставить в очередь).
2. Вычеркнуть первую вершину из очереди Vi и отметить новые вершины:
	* 1. в которые идут ненасыщенные дуги из Vi или
		2. из которых в Vi идут дуги с ненулевым потоком.
3. Повторить, пока не будет достигнут сток или пока не останется вершин, которые можно отметить.

После завершения работы алгоритма **критический разрез** можно найти по «недостроенному» увеличивающему пути: это будут насыщенные дуги, идущие из отмеченных вершин.

**Пример**. Найти максимальный поток и критический разрез.

5

5

5

8

3

4

1

1

2

4

1

2

**Решение:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дуги | *с* | I | II | III | IV |
| S-V1 | 5 | +5 |  |  |  |
| V1-V4 | 5 | +5 |  |  | -3 |
| V4-T | 5 | +5 |  |  |  |
| S-V2 | 4 |  | +1 |  | +3 |
| V2-V5 | 1 |  | +1 |  |  |
| V5-T | 8 |  | +1 | +1 | +3 |
| S-V3 | 2 |  |  | +1 |  |
| V3-V5 | 1 |  |  | +1 |  |
| V2-V4 | 4 |  |  |  | +3 |
| V1-V5 | 3 |  |  |  | +3 |

~~S~~

~~V1~~ (5; S)

~~V2~~ (4; S)

~~V3~~ (2; S)

V4 (5; V1)

V5 (3; V1)

T (5; V4)

T – V4 – V1 – S (5)

1.

~~S~~

~~V2~~(4;S)

~~V3~~(2;S)

V5(1;V2)

V4(4;V2)

T(8;V5)

T-V5-V2-S (1)

 III

~~S~~

~~V3~~(2:S)

~~V2~~(3:S)

V5(1:V3)

V4(4:V2)

T(7:V5)

T-V5-V3-S (1)

IV

~~S~~

~~V2~~ (3,S)

~~V3~~ (1,S)

~~V4~~ (4,V2)

~~V1~~ (-5,V4)

V5 (3,V1)

T (6,V5)

T-V5-V1-V4-V2-S (3)

V

~~S~~

~~V3~~ (1,S)

~~V2~~ (2,V3)

~~V4~~ (1,V2)

~~V1~~ (-2,V4)

1) S,V1,V2,V3,V4

2)V5,T

V3-V5,V1-V5,V2-V5,V4-T

5/5

2/5

5/5

5/8

3/3

3/4

1/1

1/1

1/2

4/4

0/1

0/2

**Критический разрез**:

V3-V5,V1-V5,V2-V5,V4-T

F=5+4+1=10

В графе могут быть заданы пропускные способности не только у дуг, но и у вершин. В этом случае формируют новый граф: каждую вершину с ограниченной пропускной способностью заменяют на дугу.

**Пример**. Сформировать транспортную сеть. Найти максимальный поток и критический разрез.

9

6

10

9

3

4

8

8

2

10

**Решение:**

Модифицированная сеть:

9

6

10

9

3

4

8

8

2

10

|  |  |  |
| --- | --- | --- |
| Дуги | *с* | I |
| S-A1 | 9 | +6 |
| A1-A2 | 10 | +6 |
| A2-C | 6 | +6 |
| C-T | 10 | +6 |
|  |  |  |
|  |  |  |
|  |  |  |

~~S~~

~~A1~~ (9; S)

~~B~~ (8; S)

~~A2~~ (10; A1)

~~D1~~ (4; B)

C (6; A2)

D2 (8; D1)

T (10; C)

T-C-A2-A1-S (6)

II.

~~S~~

~~A1~~ (3;S)

~~B~~(8;S)

~~A2~~(4;A1)

~~D1~~(4;B)

C(-6;A2)

D2(8;D1)

T(+4;C)

T-C-A2-A1-S (3)

9/9

3/6

9/10

9

3

4

8

8

2

9/10

Более сложная задача – поиск максимального потока **минимальной стоимости**. Решается методами линейного программирования (транспортная задача) или методами динамического программирования.