11. Графы

Достаточно широкий спектр задач можно решить при помощи теории графов. Например, нахождение максимального потока в сети, кратчайшего расстояния, максимального паросочетания, проверка планарности графа и др. Как особый класс можно выделить задачи оптимизации на графах.

Для работы с графами в Maple V предназначена библиотека networks. Команда подключения этой библиотеки – стандартная, т.е. достаточно воспользоваться оператором with.

Граф в Maple представляется особой процедурой типа GRAPH. Для работы с графами можно воспользоваться любой из 75-ти функций, содержащихся в библиотеке networks. Основные функции, которые позволяют создавать и изменять графы, а также процедуры решения классичес-

Основные функции библиотеки networks

ФУНКЦИЯ	ОПИСАНИЕ
addedge	добавление ребер в граф
addvertex	добавление вершин в граф
adjacency	составление матрицы смежности
complete	создание полного графа
delete	удаление из графа ребер и вершин
draw	построение чертежа графа
duplicate	создание копии графа
ends	возвращает имена вершин
eweight	нахождение весов ребер
flow	нахождение максимального потока в сети
head	нахождение начальных вершин
incidence	составление матрицы инцидентности
isplanar	проверка графа на планарность
new	создание пустого графа
petersen	создание графа Петерсена
random	создание случайного графа
shortpathtree	нахождение дерева кратчайших расстояний
show	вывод полной информации о графе
tail	нахождение конечных вершин
void	создание графа без ребер
vweight	нахождение весов вершин

ких задач из теории графов приведены в таблице.

Рассмотрим пример решения задачи нахождения максимального потока в сети.

Вначале с помощью команды new создаем пустой граф G:

```
> with(networks):
> new(G):
```

Теперь добавим в граф G шесть вершин:

```
> addvertex({a1,a2,a3,a4,a5,a6},G);
```

a1, *a2*, *a3*, *a5*, *a4*, *a6*

Затем соединим вершины ребрами:

```
> addedge([{a1,a2}, {a2,a3}, {a2,a5}, {a2,a4},
{a3,a4}, {a3,a6}, {a5,a6}, {a1,a4}],
weights=[8,3,2,4,5,8,6,5], G);
```

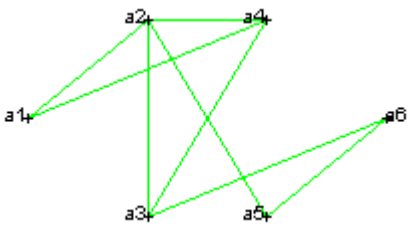
e1, *e2*, *e3*, *e4*, *e5*, *e6*, *e7*, *e8*

Следует заметить, что в Maple ненаправленное ребро, соединяющее вершины e1 и e2 обозначаются как {e1,e2} или {e2,e1}. Направленное ребро обозначается [e1,e2] - ребро направлено от e1 к e2.

В предыдущей команде выражение weights определило веса ребер. Для задачи отыскания максимального потока в сети вес ребра трактуется как пропускная способность.

Теперь просмотрим созданный граф:

```
> draw(Linear([a1],[a3,a2],[a5,a4],[a6]),G);
```



Для нахождения максимального потока от источника a1 к стоку a6 надо воспользоваться функцией flow. Рассмотрим подробно формат вызова этой функции:

flow(G,s,t);

flow(G,s,t,'maxflow'=n);

flow(G,s,t,eset,comp);

flow(G,s,t,eset,comp,'maxflow'=n);

Команда находит максимальный поток в сети G от вершины s к вершине t (от источника к стоку).

eset - имя множества, в котором будут содержаться ребра, "работающие" с максимальной нагрузкой.

comp - имя множества, в котором будут содержаться имена вершин через которые проходит поток.

Параметр maxflow определяет, что необходимо найти поток n. Если n больше максимально возможного потока, то возвращается значение максимально возможного потока.

```
> flow(G,a1,a6,'set1','set2');
```

10

Команда flow выдала значение максимального потока в заданной сети. Теперь просмотрим ребра с максимальной загрузкой:

```
> set1;
```

$$\{\{a1, a2\}, \{a2, a3\}, \{a2, a5\}, \{a3, a4\}, \{a3, a6\}\}$$

```
> set2;
```

$$\{a1, a2, a4\}$$

Таким образом, задача решена: найден максимальный поток, а, используя информацию о ребрах с максимальной загрузкой, можно получить данные о загруженности оставшихся ребер.

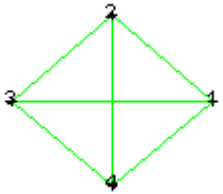
Широко распространены задачи нахождения кратчайшего пути. В таких задачах задается граф (сеть дорог) и начальная вершина (пункт отправления). Каждому ребру можно присвоить вес - длина дороги; кроме этого ребра могут быть как ориентированными, так и не ориентированными. Задача нахождения кратчайшего пути решается с помощью алгоритма Дейкстры. Результат работы алгоритма - дерево с началом в начальной вершине, причем ко всем остальным вершинам идут кратчайшие пути. В Maple V задачу нахождения кратчайшего пути можно решить при помощи команды shortpathtree.

Формат команды:

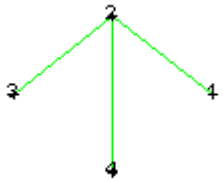
shortpathtree(G,v), где G - граф, v - начальная вершина.

В следующем примере создадим полный граф с четырьмя вершинами, причем вес каждого ребра равен 1.

```
> g:=complete(4): draw(g);
```




```
> g1:=shortpathtree(g,2): draw(g1);
```



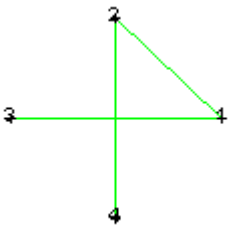
Таким образом, g1 - дерево, полученное в результате работы алгоритма Дейкстры.

Теперь изменим исходный граф g: ребру, которое соединяет вершины 2 и 3 присвоим вес, равный 100.

```
> del_edg:=edges({2,3},g);
```

$$del\_edg := \{e3\}$$

```
> delete(del_edg[1],g):
addedge({2,3},weights=100,g):
> t:=shortpathtree(g,2): draw(t);
```



Таким образом, в измененном графе найдены новые кратчайшие пути ко всем вершинам.

Команда shortpathtree также присваивает длины кратчайших путей весам вершин.

```
> vweight(t);
```

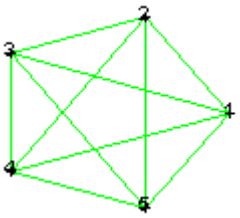
table(*sparse*, [
3 = 2
4 = 1
1 = 1
2 = 0
])

Из данной таблицы видно, что для того, чтобы попасть из вершины 2 в вершину 3, требуется пройти расстояние равное двум.

В теории графов существует понятие планарности графа. Граф называется планарным, если его можно изобразить на плоскости без самопересечений. Задача определения планарности встречается при разводке печатных плат, где ребра графа - печатные проводники, вершины - контактные площадки.

В Maple V проверить планарность графа можно при помощи команды isplanar, которая возвращает true - если граф планарный и false - в противном случае. Вначале isplanar упрощает граф, т.е. удаляет циклы и повторяющиеся ребра, а затем граф проверяется на планарность (упростить граф можно при помощи gsimp).

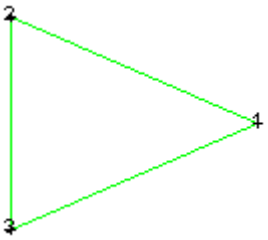
```
> g:=complete(5): draw(g);
```




```
> isplanar(g);
```

false

```
> g1:=complete(3): draw(g1);
```




```
> isplanar(g1);
```

true

В этом разделе приведены лишь краткие сведения о библиотеке networks, но перечисленных команд будет достаточно для решения элементарных задач теории графов. При необходимости найти дополнительные сведения о данной библиотеке можно воспользоваться справочной системой Maple.