第六讲: 图论模型和算法

数学模型和算法的应用与 MATLAB 实现

周吕文

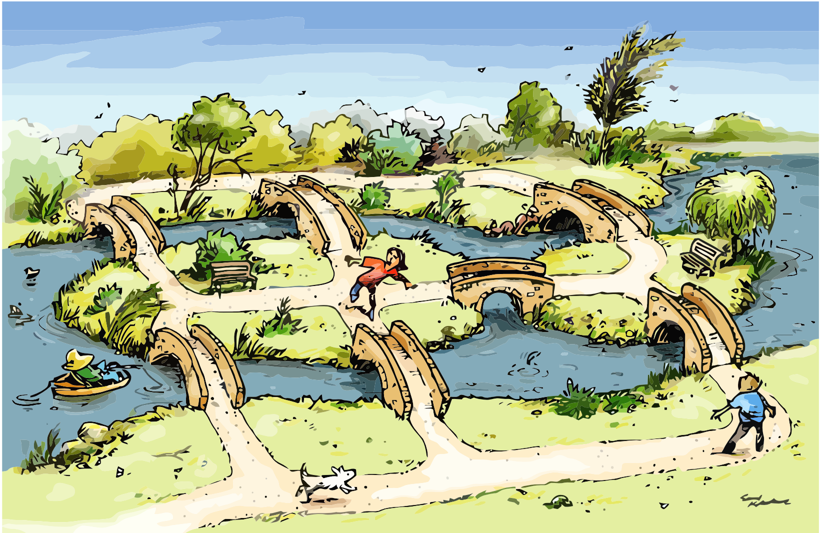
中国科学院力学研究所

2017 年 7 月 3 日

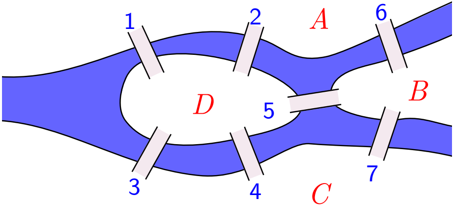
微信公众号: 超级数学建模

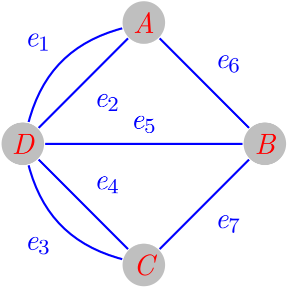
提要

图论的起源：哥尼斯堡七桥问题

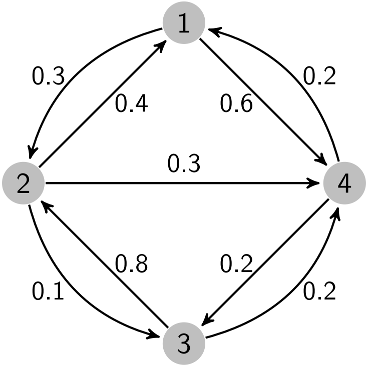


图论的起源：哥尼斯堡七桥问题





图论的定义



- 图论 (Graph theory) 以图为研究对象，研究顶点和边组成的图形的数学理论和方法.
- 图论中的图是由若干给定的顶点及连接两顶点的边所构成的图形.
- 图论中的图通常用来描述某些事物之间的某种特定关系，用顶点代表事物，用边表示相应两个事物间的关系.

数学建模竞赛中的应用

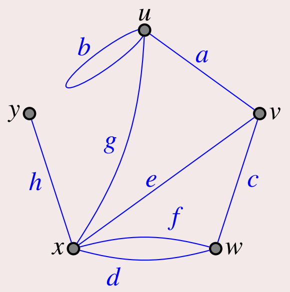
表: 近几年 MCM 中用到图论和网络的特等奖论文统计

年份题号	题目	特等奖论文数
2011 MCM-B	中继器协调问题	4
2012 MCM-B	犯罪克星	7
2013 ICM-C	地球健康的网络模型	5
2014 MCM-B	大学传奇教练	1
2014 ICM-C	使用网络来评估影响和冲击	6
2015 ICM-C	组织人力资本管理	6

提要

1. 图论算法简介
   - 起源
   - 定义
   - 应用
2. 概念、算法和实例
   - 基本概念
   - 常用算法
   - 数模案例
3. 总结
   - 要求

图（无向图）的构成



$$V(G) = \{u, v, w, x, y\}$$

$$E(G) = \{a, b, c, d, e, f, g, h\}$$

$$\varphi_G(e) = vx = xv$$

图的构成

顶点集　边集　关联函数

顶点集 $V(G)$

- 图 G 中所有顶点的集合。

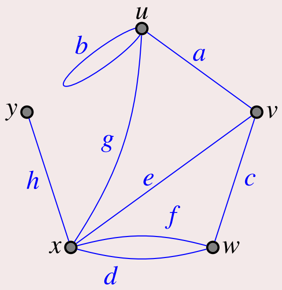
边集 $E(G)$

- 图 G 中所有边的集合。

关联函数 φ_G

- $\varphi_G : E(G) \longrightarrow V(G)$

环 / 连杆 / 重边



b 为环；a 为连杆；d，f 为重边

环

- 端点重合为一点的边。

连杆

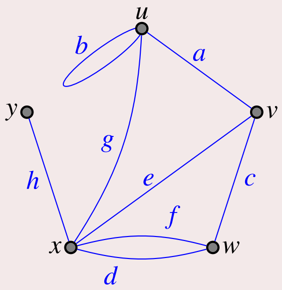
- 端点不重合的边。

重边

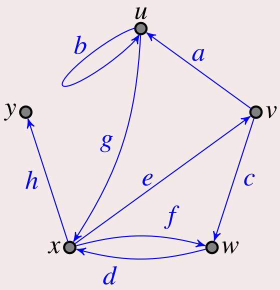
- 具有相同的两个端点的边。

图（无向图）和有向图

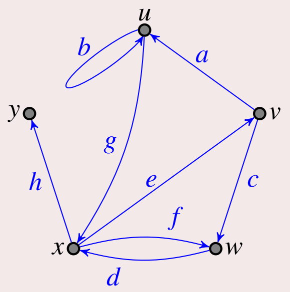
无向图



有向图



有向图的构成



$$V(G) = \{u, v, w, x, y\}$$

$$E(G) = \{a, b, c, d, e, f, g, h\}$$

$$\varphi_G(a) = (u, v) = uv$$

有向图的构成

顶点集　弧集　关联函数

顶点集 $V(G)$

- 图 G 中所有顶点的集合。

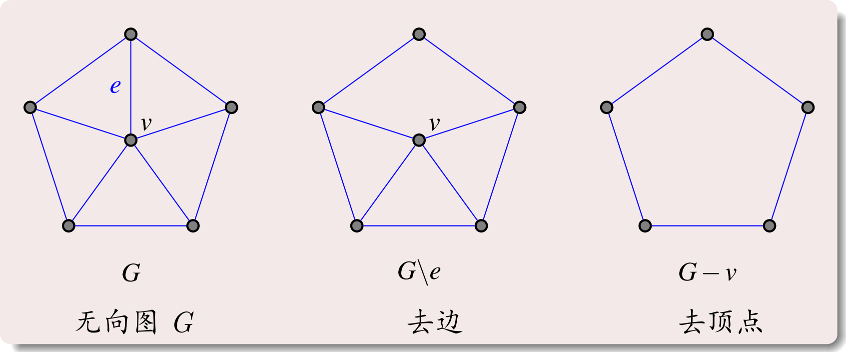
弧集 $A(G)$

- 图 G 中所有弧的集合。

关联函数 φ_G

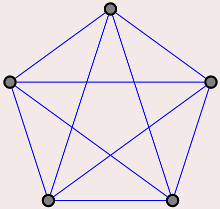
- $\varphi_G : A(a) \longrightarrow V(G)$

子图

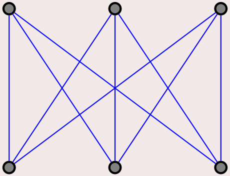


- 若 $V(H) \subset V(G)$ 且 $E(H) \subset E(G)$，则称 H 是 G 的子图。

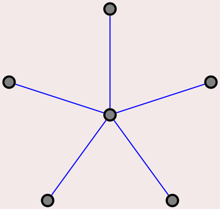
一些特殊的图



完全图

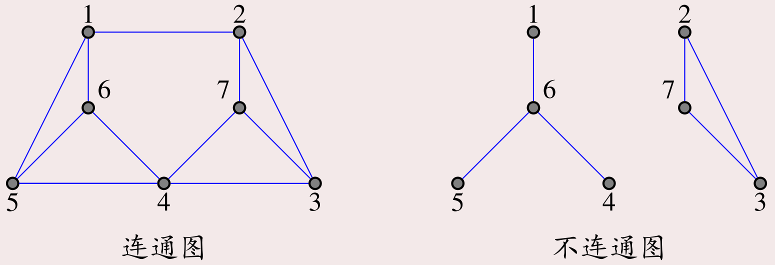


完全二分图

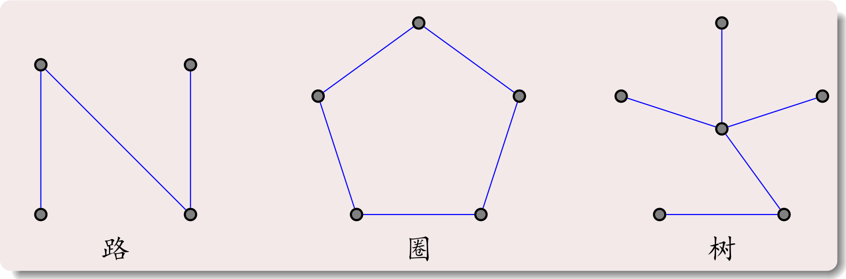


星图

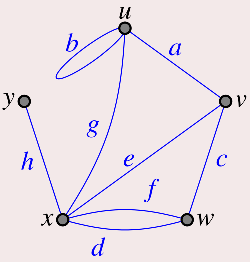
一些特殊的图



一些特殊的图



图与网络的数据结构：无向图关联 / 邻接矩阵



G

	a	b	c	d	e	f	g	h
u	1	2	0	0	0	0	1	0
v	1	0	1	0	1	0	0	0
w	0	0	1	1	0	1	0	0
x	0	0	0	1	1	1	1	1
y	0	0	0	0	0	0	0	1

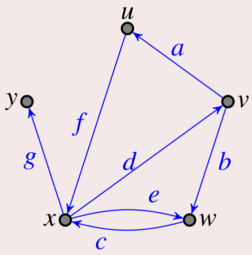
M

	u	v	w	x	y
u	2	1	0	1	0
v	1	0	1	1	0
w	0	1	0	2	0
x	1	1	2	0	1
y	0	0	0	1	0

A

- 关联矩阵 $M=(m_{ve})$，$m_{ve}\in\{0,1,2\}$ 表示边 e 与顶点 v 关联的次数。
- 邻接矩阵 $A=(a_{uv})$，a_{uv} 表示是否存在从顶点 u 到 v 的弧。

图与网络的数据结构：有向图关联 / 邻接矩阵



D

	a	b	c	d	e	f	g
u	1	0	0	0	0	−	0
v	−	−	0	1	0	0	0
w	0	1	−	0	1	0	0
x	0	0	1	−	−	1	−
y	0	0	0	0	0	0	1

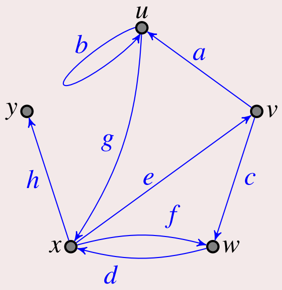
M

	u	v	w	x	y
u	0	0	0	1	0
v	1	0	1	0	0
w	0	0	0	1	0
x	0	1	1	0	1
y	0	0	0	0	0

A

- 关联矩阵 $M=(m_{va})$，$m_{va}\in\{1,-1,0\}$ 分表示弧 a 与顶点 v 关联的关系（尾、头、其它）。
- 邻接矩阵 $A=(a_{uv})$，a_{uv} 表示是否存在从顶点 u 到 v 的弧。

顶点的度和中心度



$$d^-(x) = 3$$

$$d^+(x) = 2$$

度 $d_G(v)$

- G 中与 v 关联的边数，$d_G(v) = d^-(v) + d^+(v)$。

出度 $d^-(v)$

- 以 v 为弧尾，起始于该点的弧数。

入度 $d^+(v)$

- 以 v 为弧头，终止于该点的弧数。

顶点的度和中心度

点度中心度

$$C_D(v) = d^+(v)$$

接近中心度

$$C_C(v) = \frac{1}{\sum_{u \in V} d(u, v)}$$

中间中心度

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

特征向量中心度

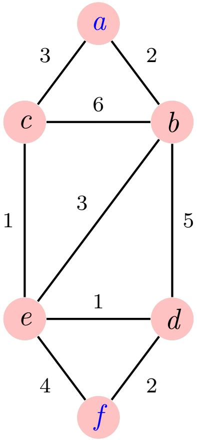
$$C_E(v) = x_v = \frac{1}{\lambda} \sum_{u \in M(v)} x_u = \frac{1}{\lambda} \sum_{u \in V} a_{vu} x_u$$

图论工具箱：函数

图论工具箱的相关命令

函数名	功能
graphallshortestpaths	求图中所有顶点对之间的最短距离
graphconnredcomp	找无（有）向图的（强/弱）连通分支
graphisreddag	测试有向图是否含有圈
graphisomorphism	确定一个图是否有生成树
graphmaxflow	计算有向图的最大流
graphminspantree	在图中找最小生成树
graphpred2path	把前驱顶点序列变成路径的顶点序列
graphshortestpath	求指定一对顶点间的最短距离和路径
graphtopoorder	执行有向无圈图的拓扑排序
graphtraverse	求从一顶点出发，所能遍历图中的顶点

图论工具箱：数据结构



满矩阵和稀疏矩阵 (full⇌sparse)

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 \\
3 & 6 & 0 & 0 & 0 & 0 \\
0 & 5 & 0 & 0 & 0 & 0 \\
0 & 3 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 2 & 4 & 0
\end{bmatrix}
\rightleftharpoons
\begin{array}{ll}
(2,1) & 2 \\
(3,1) & 3 \\
(3,2) & 6 \\
(4,2) & 5 \\
(5,2) & 3 \\
(5,3) & 1 \\
(5,4) & 1 \\
(6,4) & 2 \\
(6,5) & 4
\end{array}
$$

图论工具箱：用法举例

graphshortestpath 函数用法

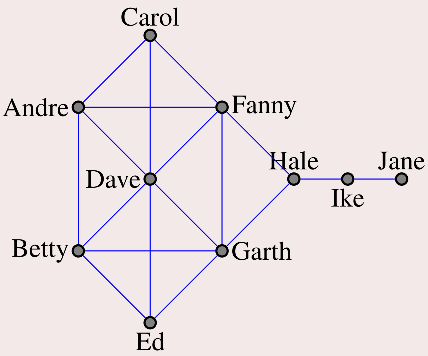
```
[a,b,c,d,e,f] = deal(1,2,3,4,5,6);
%     a  b  c  d  e  f
w = [ 0  2  3  0  0  0   % a
      2  0  6  5  3  0   % b
      3  6  0  0  1  0   % c
      0  5  0  0  1  2   % d
      0  3  1  1  0  4   % e
      0  0  0  2  4  0]; % f

W = sparse(w);
[dist, path, pred] = graphshortestpath(W, a, f)
```

网络分析工具箱：函数

网络分析工具箱的相关命令

函数名	功能
degrees	求图中所有顶点的度，入度和出度
ave_neighbor_deg	求图中所有顶点的相邻顶点平均度
closeness	求图中所有顶点的接近中心度
node_betweenness_faster	求图中所有顶点的中间中心度
edge_betweenness	求图中所有边的中间中心度
eigencentrality	求图中所有顶点的特征向量中心度
clust_coeff	求图中所有顶点的集聚系数

网络分析工具箱：用法举例



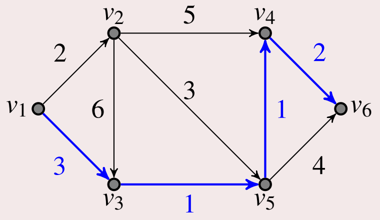
	C_D	C_C
Andre	0.444	0.529
Betty	0.444	0.529
Carol	0.333	0.500
Dave	0.667	0.600
Ed	0.333	0.500
Fanny	0.556	0.643
Garth	0.556	0.643
Hale	0.333	0.600
Ike	0.222	0.429
Jane	0.111	0.310

网络分析工具箱：用法举例

点度中心度的接近中心度的求解

```
n = 10; % 顶点数
% 给Andre, Betty, ..., Jane标号为1, 2, ..., 10.
Andre = 1; Betty = 2; Carol = 3; Dave  = 4; Ed    = 5;
Fanny = 6; Garth = 7; Hale  = 8; Ike   = 9; Jane  =10;
% 根据图构造邻接矩阵.
A = zeros(n);
A(Andre, [Betty, Carol,  Dave, Fanny]) = 1;
A(Betty, [Andre,  Dave,     Ed, Garth]) = 1;
A(Carol, [Andre,  Dave, Fanny]) = 1;
A( Dave, [Andre, Betty, Carol,     Ed, Fanny, Garth]) = 1;
A(   Ed, [Betty,  Dave, Garth]) = 1;
A(Fanny, [Andre, Carol,  Dave, Garth,  Hale]) = 1;
A(Garth, [Betty,  Dave,     Ed, Fanny,  Hale]) = 1;
A( Hale, [Fanny, Garth,   Ike]) = 1;
A(  Ike, [ Hale,  Jane]) = 1;
A( Jane, [  Ike]) = 1;
Cd = degrees(A)' /(n-1)   % 计算点度中心度并标准化.
Cc = closeness(A)*(n-1)   % 计算接近中心度并标准化.
```

最短路径

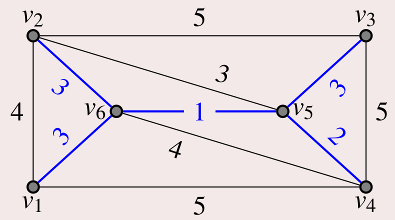


- $G(V, W)$ 边权为 $w(v_i, v_j)$。
- 两个顶点 v_s 和 v_t 间存在一条总权最小的路

$$w(\mu) = \min \sum_{(v_i,v_j)\in\mu} w(v_i, v_j)$$

```
w = [ 0   2   3   0   0   0    % a
      2   0   6   5   3   0    % b
      3   6   0   0   1   0    % c
      0   5   0   0   1   2    % d
      0   3   1   1   0   4    % e
      0   0   0   2   4   0];  % f
W = sparse(w);
[dist, path, pred] = graphshortestpath(W, 1, 6)
```

最小生成树

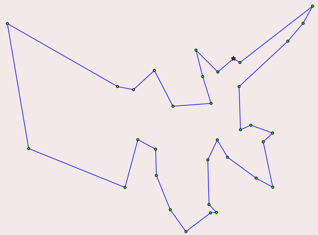


- $G(V,E)$ 边权为 $w(v_i,v_j)$。
- 若存在 $T\subseteq E$ 且为无循环图，使权 T 的总权最小

$$w(T)=\min\sum_{(v_i,v_j)\in T} w(v_i,v_j)$$

```
w = [ 0    4   inf   5   inf   3
      4    0    5   inf   3    3
     inf   5    0    5    3   inf
      5   inf   5    0    2    4
     inf   3    3    2    0    1
      3    3   inf   4    1    0];
W = sparse(w);
[ST, pred] = graphminspantree(W);
```

最短（Hamilton）回路

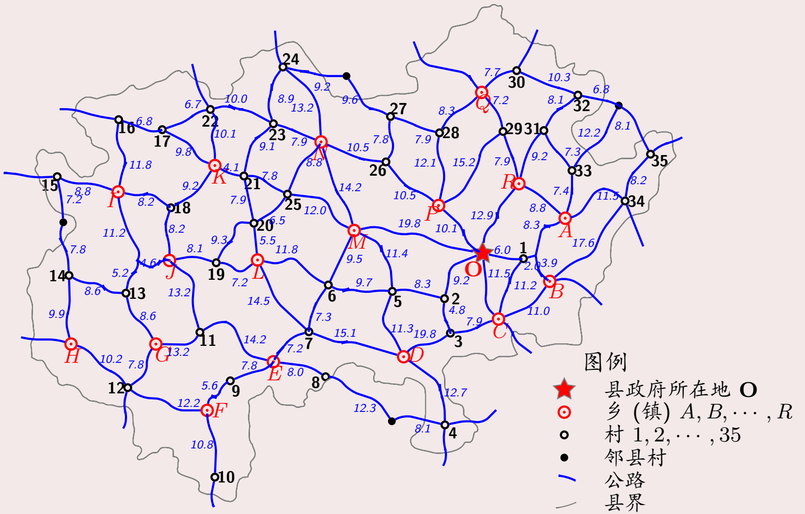


- $G(V, W)$ 边权为 $w(v_i, v_j)$。
- 寻找 G 中的回路 C, 使得 C 的总权最小

$$w(C) = \min \sum_{(v_i, v_j) \in C} w(v_i, v_j)$$

```
R = 6378.137;
dist = zeros(n);
for i = 1:n
    for j = i+1:n
        dist(i,j) = distance(lat(i),lon(i), lat(j),lon(j), R);
    end
end
[order,totdist] = minhamiltonpath(dist)
```

灾情巡视路径：问题



- 分三组（路）巡视，设计总路程最短且各组均衡的巡视路线

灾情巡视路径：思路

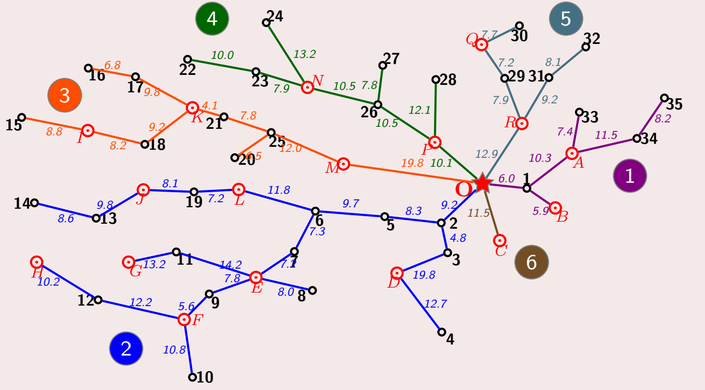
数据预处理

- 构造完全图：由图论工具箱 graphallshortestpaths 函数求得任意两点最短路。

明确目标：将 G 分成三个子图 $G(V_1)$, $G(V_2)$ 和 $G(V_3)$

- 子顶点集中都包含顶点 O: $O \in V_i$, $i = 1, 2, 3$;
- 子顶点集中包含了 V 中所有顶点: $\cup V_i = V$;
- 最小 Hamilton 回路长度总和最小化: $\min C_\Sigma = \min \sum C_i$
- 最小 Hamilton 回路长度均衡化: $\min\{C_{\max} - C_{\min}\}$

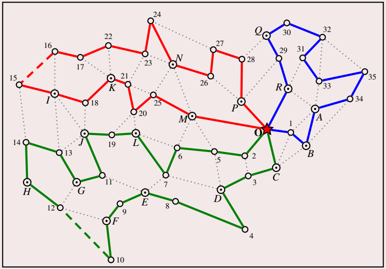
灾情巡视路径：分组



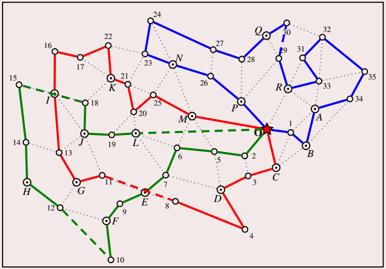
分组方案

1. (❶ ❺), (❷ ❻), (❸ ❹)
2. (❶ ❹ ❺), (❷; 15, 18), (❸ ❻; 22, 3, 4, 8, 11, 13, D, G)

灾情巡视路径：结果



1: $C_{\Sigma} = 554.1$; $C_{\max} = 237.5$



2: $C_{\Sigma} = 607.6$; $C_{\max} = 203.5$

提要

1. 图论算法简介
   - 起源
   - 定义
   - 应用
2. 概念、算法和实例
   - 基本概念
   - 常用算法
   - 数模案例
3. 总结
   - 要求

要求

- 掌握图论常见问题（最短路径、最小生成树等）的数学描述和实际意义。
- 掌握节点中心度的数学描述和实际意义。
- 会使用工具箱函数求解图论常见问题。
- 会使用工具枭函数求解网络常见问题。

作业

- 自行学习所给图论教程，了解最大流、最小费用流问题。
- 结合模拟退火算法，针对灾情巡视路径问题开发一个自动分组程序。
- 使用网络方法，解决“2014 ICM-C 使用网络来评估影响和冲击”问题。

Thank You!!!