第六章 图论

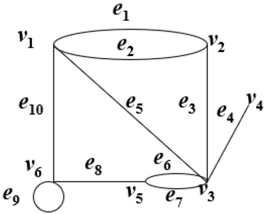
修贤超

https://xianchaoxiu.github.io

图与网络的基本知识

图与网络的基本概念

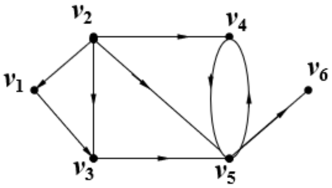
定义: 一个图是由点集 $V=\{v_j\}$ 和 V 中元素的无序对的一个集合 $E=\{e_k\}$ 构成的二元组，记为 $G=(V,E)$，其中 V 中的元素 v_j 叫做顶点，V 表示图 G 的点集合；E 中的元素 e_k 叫做边，E 表示图 G 的边集合。



- $V=\{v_1,v_2.v_3.v_4.v_5,v_6\}$
- $E=\{v_1,v_2.v_3.v_4.v_5,v_6\}$
- $e_1=(v_1,v_2)$, $e_2=(v_1,v_2)$, $e_3=(v_2,v_3)$, $e_4=(v_3,v_4)$, $e_5=(v_1,v_3)$, $e_6=(v_3,v_5)$, $e_7=(v_3,v_5)$, $e_8=(v_5,v_6)$, $e_9=(v_6,v_6)$, $e_{10}=(v_1,v_6)$

图与网络的基本概念

- 如果一个图是由点和边所构成的，则称其为无向图，记作 $G=(V,E)$，连接点的边记作 (v_i,v_j)，或者 (v_j,v_i)。
- 如果一个图是由点和弧所构成的，那么称它为有向图，记作 $D=(V,A)$，其中 V 表示有向图 D 的点集合，A 表示有向图 D 的弧集合。一条方向从 v_i 指向 v_j 的弧，记作 (v_i,v_j)。

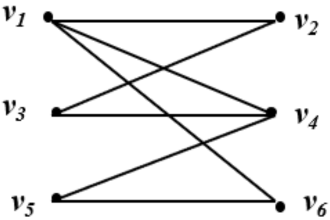


- $V=\{v_1,v_2.v_3.v_4.v_5,v_6\}$
- $A=\{(v_1,v_3),(v_2,v_1),(v_2,v_3),(v_2,v_5),(v_3,v_5),(v_4,v_5),(v_5,v_4),(v_5,v_6)\}$

图与网络的基本知识

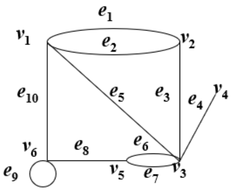
图与网络的基本概念

- 一条边的两个端点是相同的，那么称为这条边是环。如果两个端点之间有两条以上的边，那么称为它们为多重边。
- 一个无环，无多重边的图称为简单图，一个无环，有多重边的图称为多重图。
- 每一对顶点间都有边相连的无向简单图称为完全图。有向完全图则是指任意两个顶点之间有且仅有一条有向边的简单图。
- 图 $G=(V,E)$ 的点集 V 可以分为两个非空子集 X，Y，即 $X\cup Y=V$，$X\cap Y=\varnothing$，使得 E 中每条边的两个端点必有一个端点属于 X，另一个端点属于 Y，则称 G 为二部图(偶图)，有时记作 $G=(X,Y,E)$。



图与网络的基本概念

- 以点 v 为端点的边的个数称为点 v 的度（次），记作 $d(v)$。



- 例如图中 $d(v_1) = 4$，$d(v_6) = 4$（环计两度）
- 度为零的点称为弧立点，度为 1 的点称为悬挂点。悬挂点的关联边称为悬挂边。度为奇数的点称为奇点，度为偶数的点称为偶点。

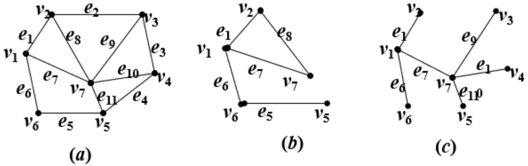
图与网络的基本概念

- 定理 1: 所有顶点度数之和等于所有边数的 2 倍。
- 定理 2: 在任一图中，奇点的个数必为偶数。
- 有向图中，以 v_i 为始点的边数称为点 v_i 的出次，用 $d^+(v_i)$ 表示；以 v_i 为终点的边数称为点 v_i 的入次，用 $d^-(v_i)$ 表示；v_i 点的出次和入次之和就是该点的次。所有顶点的入次之和等于所有顶点的出次之和。

图与网络的基本知识

图与网络的基本概念

- 图 $G=(V,E)$，若 E' 是 E 的子集，V' 是 V 的子集，且 E' 中的边仅与 V' 中的顶点相关联，则称 $G'=(V',E')$ 是 G 的一个子图。特别是，若 $V'=V$，则 G' 称为 G 的生成子图 (支撑子图)。



- 在实际应用中，给定一个图 $G=V,E$ 或有向图 $D=V,A$, 在 V 中指定两个点，一个称为始点（或发点)，记作 v_1 ，一个称为终点（或收点)，记作 v_n ，其余的点称为中间点。对每一条弧 $(v_i,v_j)\in A$，对应一个数 w_{ij} ，称为弧上的权。通常把这种赋权的图称为网络。

连通图

- 无向图 $G = (V, E)$，若图 G 中某些点与边的交替序列可以排成 $(v_{i0}, e_{i1}, v_{i1}, e_{i2}, \ldots, v_{ik-1}, e_{ik}, v_{ik})$ 的形式，且 $e_{it} = (v_{it-1}, v_{it})$ $(t = 1, \ldots, k)$，则称这个点边序列为连接 v_{i0} 与 v_{ik} 的一条链，链长为 k。点边列中没有重复的点和重复边者为初等链。
- 无向图 G 中，连结 v_{i0} 与 v_{ik} 的一条链，当 v_{i0} 与 v_{ik} 是同一个点时，称此链为圈。圈中既无重复点也无重复边者为初等圈。
- 对于有向图可以类似于无向图定义链和圈，初等链、圈，此时不考虑边的方向。而当链 (圈) 上的边方向相同时，称为道路 (回路)。对于无向图来说，道路与链、回路与圈意义相同。
- 一个图中任意两点间至少有一条链相连，则称此图为连通图。任何一个不连通图都可以分为若干个连通子图，每一个称为原图的一个分图。

图的矩阵表示

- 对于网络（赋权图）$G = V, E$, 其中边有权 (v_i, v_j), 构造矩阵 $A = (a_{ij})_{n\times n}$, 其中:

$$a_{ij} = \begin{cases} w_{ij} & (v_i, v_j) \in E \\ 0 & \text{其他} \end{cases}$$

  称矩阵 A 为网络 G 的**权矩阵**。
- 设图 $G = V, E$ 中顶点的个数为 n，构造一个矩阵 $A = (a_{ij})_{n\times n}$ ，其中:

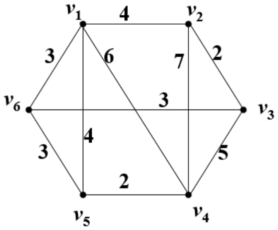
$$a_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases}$$

  称矩阵 A 为网络 G 的**邻接矩阵**。
- 当 G 为无向图时，邻接矩阵为对称矩阵。

图与网络的基本知识

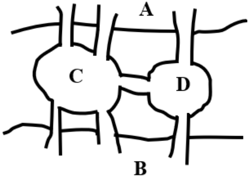
图的矩阵表示

- 试写出权矩阵和邻接矩阵



欧拉回路

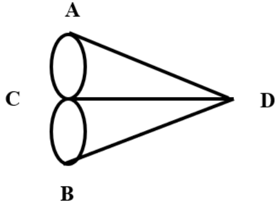
- 连通图 G 中，若存在一条道路，经过每边一次且仅一次，则称这条路为欧拉道路。若存在一条回路，经过每边一次且仅一次，则称这条回路为欧拉回路。具有欧拉回路的图称为欧拉图 (E 图)。



- 定理 3: 无向连通图 G 是欧拉图，当且仅当 G 中无奇点。
- 无向连通图 G 为欧拉图，当且仅当 G 的边集可划分为若干个初等回路。
- 无向连通图 G 有欧拉道路，当且仅当 G 中且有两个奇点。

欧拉回路

- 欧拉回路的算法: 从图 G 中的任一点 v_1 出发，找一个初等回路 c_1，再从途中去掉 c_1，在剩余的图中再找初等回路 c_2，一直做到图中所有的边都被包含在这些初等回路中，再把这些回路连续起来即得这个图的欧拉回路。



- 定理 4: 连通有向图 G 是欧拉图，当且仅当它每个顶点的出次等于入次。

中国邮递员问题

- 一个邮递员，负责某一地区的信件投递。他每天要从邮局出发，走遍该地区所有街道再返回邮局，问应如何安排送信的路线可以使所走的总路程最短？这个问题是我国管梅谷教授在 1962 年首先提出的。因此国际上通称为中国邮路问题。用图论的语言描述给定一个连通图 G，每边有非负权 $l(e)$，要求一条回路过每边至少一次，且满足总权最小。
- 定理 5: 已知图 $G^* = G + E_1$ 无奇点，则 $L(E_l) = \sum_{e \in E_l} l(e)$ 最小的充分必要条件为:
  - 每条边最多重复一次；
  - 对图 G 中每个初等圈来讲，重复边的长度和不超过圈长的一半。
- 奇偶点图上作业法

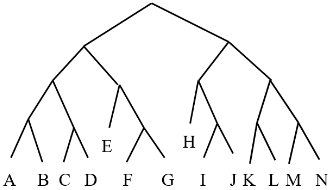
图与网络的基本知识

小结

- 图与网络的基本概念
  - 图、顶点、边、简单图、完全图
  - 顶点的次、奇点、偶点
  - 子图、生成子图
  - 权、网络
- 连通图
- 图的矩阵表示
- 欧拉回路与中国邮路问题

树的概念和性质

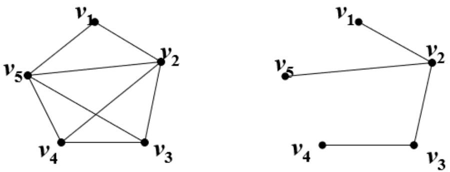
- 连通且不含圈的无向图称为树。树中次为 1 的点称为树叶，次大于 1 的点称为分枝点。



- 图 $T=(V,E)$，$|V|=n$，$|E|=m$，则下列关于树的说法是等价的。
  - T 是一个树。
  - T 无圈，且 $m=n-1$。
  - T 连通，且 $m=n-1$。
  - T 无圈，但每加一新边即得惟一一个圈。
  - T 连通，但任舍去一边就不连通。
  - T 中任意两点，有惟一链相连。

图的生成树

- 设图 $K=(V,E_1)$ 是图 $G=(V,E)$ 的一支撑子图，如果图 $K=(V,E_1)$ 是一个树, 那么称 K 是 G 的一个**生成树（支撑树）**，或简称为图 G 的树。图 G 中属于生成树的边称为**树枝**，不在生成树中的边称为**弦**。



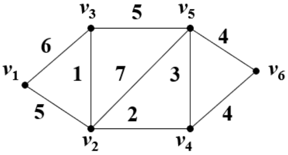
- **定理**: 一个图 G 有生成树的充要条件是 G 是连通图。

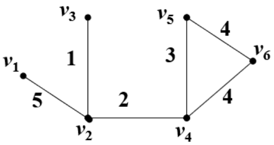
图的生成树

- 避圈法: 设在图中任取一条边 e_1，找一条与 e_1 不构成圈的边 e_2，再找一条与 $\{e_1 e_2\}$ 不构成圈的边 e_3。一般设已有 $\{e_1 e_2 \ldots, e_k\}$，找一条与 $\{e_1 e_2, \ldots e_k\}$ 中任何一些边不构成圈的边 e_{k+1}，重复这个过程，直到不能进行为止。

最小生成树

- 如果图 $T=(V,E_1)$ 是图 G 的一个生成树，那么称 E_1 上所有边的权的和为生成树 T 的权，记作 $S(T)$。如果图 G 的生成树 T^* 的权 $S(T^*)$，在 G 的所有生成树 T 中的权最小，即 $S(T^*)=\min_T S(T)$，那么称 T^* 是 G 的最小生成树。
- 某六个城市之间的道路网如图所示，要求沿着已知长度的道路联结六个城市的电话线网，使电话线的总长度最短。





- 根据破圈法和避圈法两种方式得到了图的两个不同的支撑树，由此可以看到连通图的支撑树不是唯一的。

根树及其应用

- 若一个有向图在不考虑边的方向时是一棵树，则称这个有向图为有向树。
- 有向树 T，恰有一个结点入次为 0，其余各点入次均为 1，则称 T 为根树 (又称外向树)。
- 在根树中，若每个顶点的出次小于或等于 m，称这棵树为m 叉树。若每个顶点的出次恰好等于 m 或零，则称这棵树为完全 m 叉树。当 $m = 2$ 时，称为二叉树、完全二叉树。

树

- 小结
  - 树
  - 生成树
    - 深探法
    - 广探法
  - 最小生成树
    - Kruskal 算法
    - 破圈法
  - 根树

最短路问题

问题描述

- 最短路问题是网络理论中应用最广泛的问题之一，例如设备更新、管道铺设、线路安排、厂区布局等。
- 设 $G=(V,E)$ 为连通图，图中各边 (v_i,v_j) 有权 l_{ij} （$l_{ij}=\infty$ 表示 v_i,v_j 之间没有边），v_s,v_t 为图中任意两点，求一条路 μ，使它为从 v_s 到 v_t 的所有路中总权最短。即:
$L(\mu)=\sum\limits_{(v_i,v_j)\in\mu} l_{ij}$ 最小。
- Dijkstra 算法是在 1959 年提出来的。目前公认，在所有的权 $w_{ij}\geq 0$ 时，这个算法是寻求最短路问题最好的算法。并且，这个算法实际上也给出了寻求从一个始定点 v_s 到任意一个点 v_j 的最短路。

Dijkstra 算法

- 给始点 v_s 以 P 标号 $P(v_s)=0$，这表示从 v_s 到 v_s 的最短距离为 0，其余节点均给 T 标号，$T(v_i)=+\infty\ (i=2,3,\ldots,n)$。
- 设节点 v_i 为刚得到 P 标号的点，考虑点 v_j，其中 $(v_i,v_j)\in E$，且 v_j 为 T 标号。对 v_j 的 T 标号进行如下修改:

$$T(v_j)=\min[T(v_j),P(v_j)+l_{ij}]$$

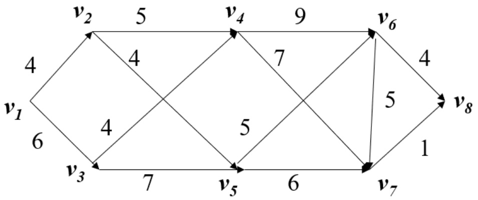
- 比较所有具有 T 标号的节点，把最小者改为 P 标号，即:

$$P(v_k)=\min[T(v_i)]$$

  当存在两个以上最小者时，可同时改为 P 标号。若全部节点均为 P 标号，则停止，否则用 v_k 代替 v_i，返回上一步。

最短路问题

- 求下图从 v_1 到 v_8 的最短路



- 首先给 v_1 以 P 标号，$P(v_1)=0$; 给其余所有点 T 标号, $T(v_i)=+\infty$ $(i=2,3,\ldots,8)$。
- $T(v_2)=\min[T(v_2),P(v_1)+l_{12}]=\min[+\infty,0+4]=4$
  $T(v_3)=\min[T(v_3),P(v_1)+l_{13}]=\min[+\infty,0+6]=6$
  比较所有 T 标号，$T(v_2)$ 最小，令 $P(v_2)=4$，并记录路径 (v_1,v_2)。
- $T(v_4)=\min[T(v_4),P(v_2)+l_{24}]=\min[+\infty,4+5]=9$
  $T(v_5)=\min[T(v_5),P(v_2)+l_{25}]=\min[+\infty,4+4]=8$
  比较所有 T 标号，$T(v_3)$ 最小，令 $P(v_3)=6$，并记录路径 (v_1,v_3)。

求下图从 v_1 到 v_8 的最短路

- $T(v_4)=\min[T(v_4),P(v_3)+l_{34}]=\min[9,4+9]=9$
  $T(v_5)=\min[T(v_5),P(v_3)+l_{35}]=\min[8,6+7]=8$
  比较所有 T 标号，$T(v_5)$ 最小，令 $P(v_5)=8$，并记录路径 (v_2,v_3)。
- $T(v_6)=\min[T(v_6),P(v_5)+l_{56}]=\min[+\infty,8+5]=13$
  $T(v_7)=\min[T(v_7),P(v_5)+l_{57}]=\min[+\infty,8+6]=14$
  比较所有 T 标号，$T(v_4)$ 最小，令 $P(v_4)=9$，并记录路径 (v_2,v_4)。
- $T(v_6)=\min[T(v_6),P(v_4)+l_{46}]=\min[13,9+9]=13$
  $T(v_7)=\min[T(v_7),P(v_4)+l_{47}]=\min[14,9+7]=14$
  比较所有 T 标号，$T(v_6)$ 最小，令 $P(v_6)=13$，并记录路径 (v_5,v_6)。
- $T(v_7)=\min[T(v_6),P(v_6)+l_{67}]=\min[14,13+5]=14$
  $T(v_8)=\min[T(v_8),P(v_6)+l_{68}]=\min[+\infty,13+4]=17$
  比较所有 T 标号，$T(v_7)$ 最小，令 $P(v_7)=14$，并记录路径 (v_7,v_8)。
- $T(v_8)=\min[T(v_8),P(v_7)+l_{78}]=\min[17,14+1]=15$
  因为只有一个 T 标号 $T(v_8)$ 最小，令 $P(v_8)=15$, 并记录路径 (v_7,v_8)，v_1 到 v_8 之最短路为: $v_1\to v_2\to v_5\to v_7\to v_8$

最短路问题

Floyd 算法

- 可直接求出网络中任意两点间的最短路。
- 令网路的权矩阵为 $D=(d_{ij})_{n\times n}$，l_{ij} 为 v_i 到 v_j 的距离

$$d_{ij}=\begin{cases} l_{ij} & \text{当}(v_i,v_j)\in E \\ \infty & \text{其他} \end{cases}$$

- 算法基本步骤
  - 输入权矩阵 $D^{(0)}=D$
  - 计算 $D^{(k)}=(d_{ij}^{(k)})_{n\times n}\ (k=1,2,\ldots,n)$，其中 $d_{ij}=\min[d_{ij}^{(k-1)},d_{ik}^{(k-1)}+d_{kj}^{k-1}]$
  - $D^{(n)}=(d_{ij}^{(n)})_{n\times n}$ 中元素 $d_{ij}^{(n)}$ 就是 v_i 到 v_j 的最短路长。

最短路问题

小结

- Dijkstra 算法
  - 求无负权网络最短路问题的最好方法
  - 指定两点间的最短路
  - 标号法
- Floyd 算法
  - 任意两点间的最短路

Q&A

Thank you!

感谢您的聆听和反馈