# 第四次第八周小班讨论（个人资料）

图结构研讨

中国邮递员问题

湖南大學 HUNAN UNIVERSITY

# 前言 PREFACE

图的应用问题是指在实际问题中，利用图论的相关概念和算法来解决具体的问题。图论作为一门数学分支，不仅有理论研究，还有广泛的实际应用。以下是一些图的应用领域：

网络和通信：

1. 计算机网络、社交网络、电信网络等都可以用图模型来表示和分析。
2. 图的最短路径算法用于路由和导航系统。
3. 图的最大流算法用于网络流量控制。

运输和物流：

中国邮递员问题（Route Inspection Problem）就是一个图的应用问题，邮递员需要找到最短路径来投递邮件。

物流中的路径规划、货物运输等问题也可以用图来建模和求解。

电路设计：

电路中的逻辑门、电路板、芯片等可以用图来表示。

图的遍历和连通性算法用于电路测试和故障诊断。

社交网络分析：

社交网络中的人际关系、信息传播、影响力分析等问题可以用图来研究。

图的中心性度量用于找出社交网络中的关键人物。

生物学和化学：

蛋白质相互作用网络、基因调控网络、分子结构等都可以用图来描述。

图的匹配算法用于蛋白质相互作用预测。

1. 城市规划和交通：

城市道路、地铁线路、交通流量等都可以用图来建模。

图的着色算法用于地图着色和交通信号灯优化。

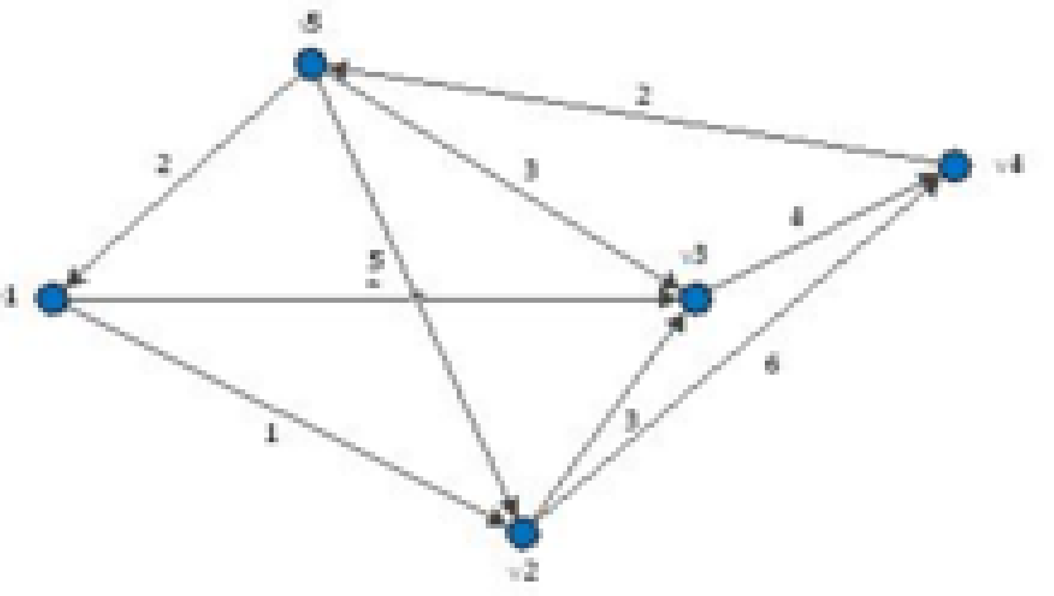
# 目录

第 一 部 分

# 概述

标题一 概 述

# 中国邮递员问题



G

中国邮递员问题（也称路线检查问题，Route Inspection Problem）是一个图论问题。此问题为在一个连通的无向图中找到一最短的封闭路径，且此路径需通过所有边至少一次。现实意义中，中国邮递员问题就是在一个已知的地区，邮差要设法找到一条最短路径，走过此地区所有的街道，且最后要回到出发点。

中国邮递员问题由管梅谷教授在1960年提出，而美国国家标准和技术研究院（NIST）的 Alan Goldman 首先将此问题命名为中国邮递员问题。

湖南大学 HUNAN UNIVERSITY

# 中国邮递员问题的分类

- 有向图：

  无向图的中国邮递员问题是容易解决的，是P问题

- 无向图：

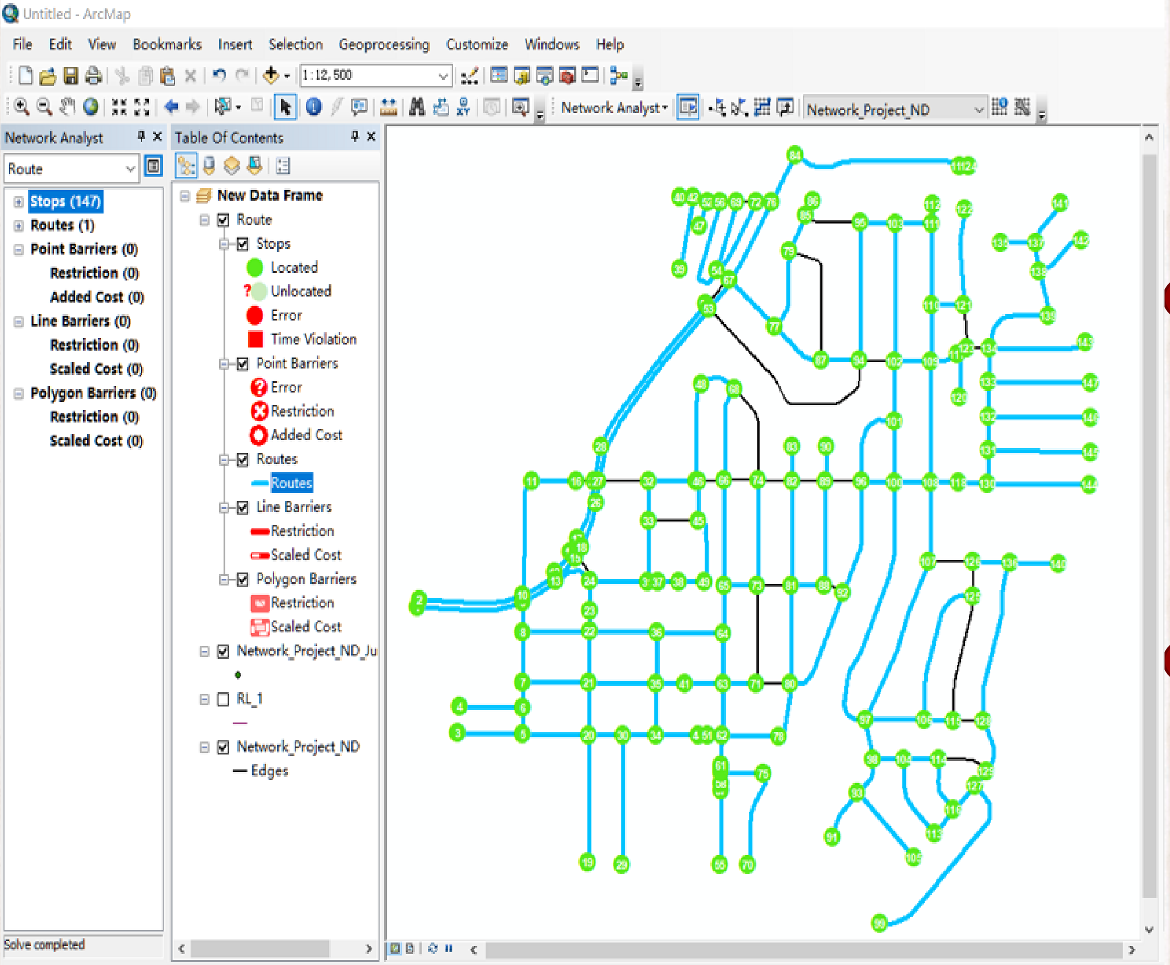
  有向图的中国邮递员问题是NP完全问题

湖南大學
HUNAN UNIVERSITY

## 第二部分

# 算法思想

# 标题二 算法思想




- 如果图 G 是欧拉图：

  任一定点出发的欧拉回路都是最优投递路线。

- 如果图 G 恰有两个奇点 x 和 y

  存在一条 x 到 y 的欧拉路径，因此，由这条欧拉路径与 x 到 y 最短路相加，即是所求的最优投递路线。

- 如果连通图 G 既不是欧拉图也没有欧拉路径

  我们需要构建多重欧拉图。

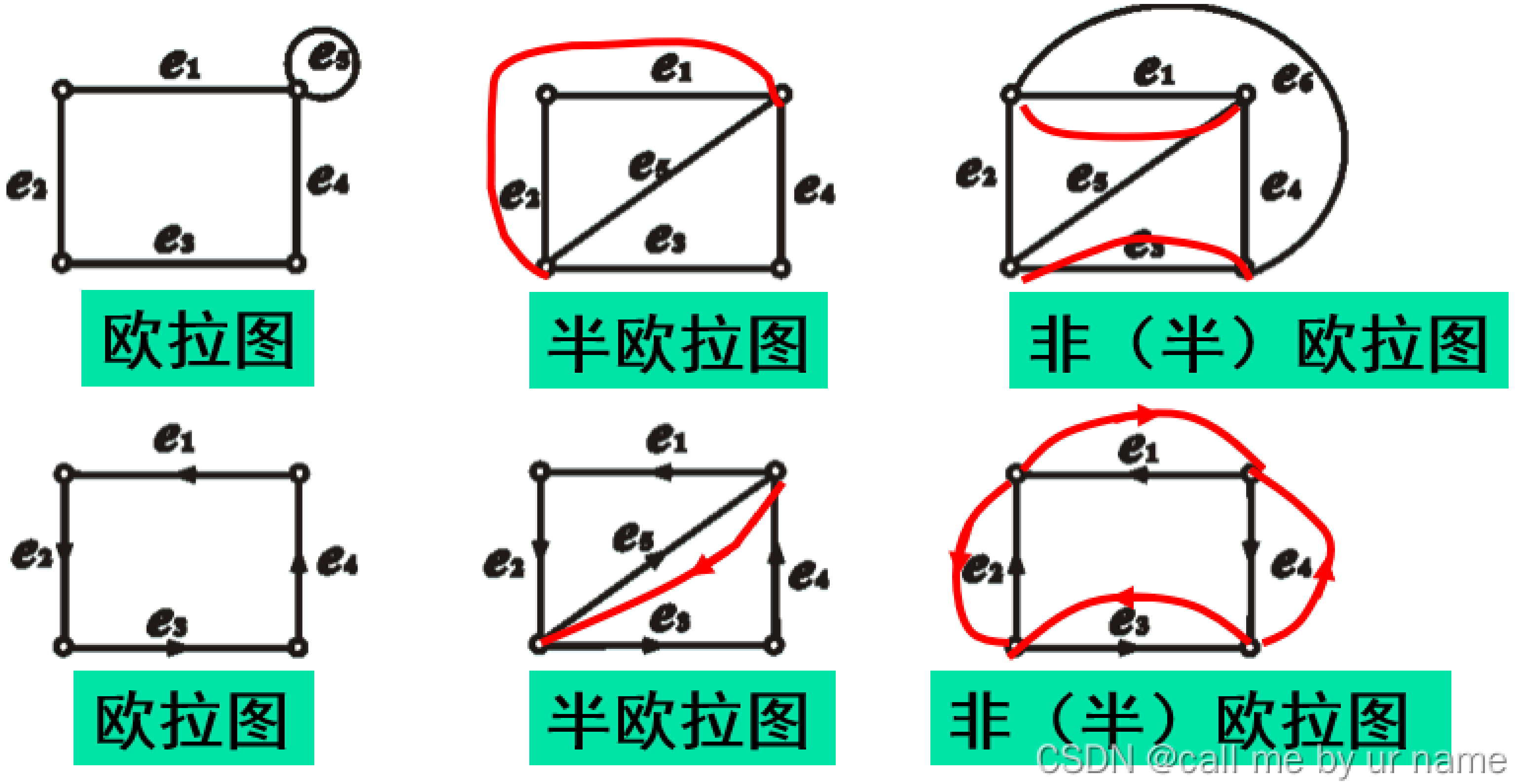
## 多重欧拉图的构建

**思 想**

若图 G 不是欧拉图也没有欧拉路径，则图 G 有多个奇点，邮递员必定要折返。

① 此时，图 G 一定有偶数个奇点，则这些奇点可以任意两两配对。任取两个奇点 G 和 G，则必有一条 G–G 路。
把 G–G 路上的每条边改为二重边，
则 G 和 G 变为偶点， G–G 路上的其他结点的度数均增加 2，其奇偶性不变。不断重复此过程，直到图 G 中所有顶点变成偶点，从而得到多重欧拉图 G′。

② 在 G′ 中，若邻点 G和 G 之间连接的边数多于 2，则可去掉其中的偶数条多重边，最后剩下连接 G 和 G 的边仅有 1 或 2 条边，这样得到的图 G′ 仍是欧拉图。



## 最优解的充要条件

具 体 表 示

图 $G'$ 中，重复的边就是邮递员要折返的道路。于是最优解满足：

① 邮递员在每条道路最多折返一次；② 邮递员总选择长度较短的边进行折返。

**定理：** 设 $P$ 是加权连通图 $G$ 中一条包含 $G$ 的所有边至少一次的回路，则 $P$ 最优（即具有最小长度）的充要条件是：

① $P$ 中没有二重以上的边。

② 在 $G$ 的每个基本回路 $C$ 中，重复边集 $E$ 的长度之和不超过所在基本回路的长度的一半，即 $\boxed{w(E) \leq \frac{1}{2}w(C)}$。

# 奇偶点图上作业法

## 具 体 表 示

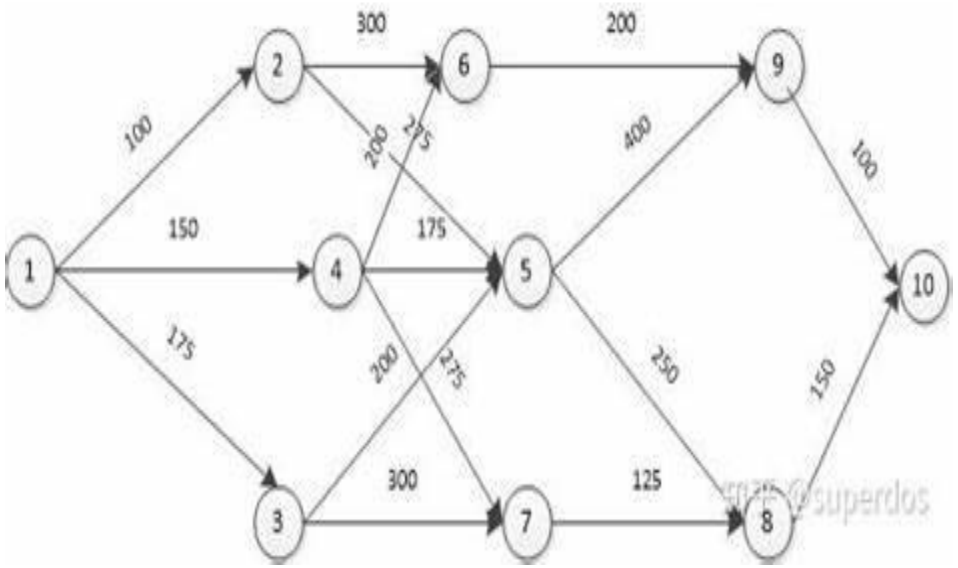
根据上面的讨论及定理，我们可以设计出求带权非欧拉连通图 G 的最优环游的算法，即**奇偶点图上作业法。**

①把 $G$ 中所有奇点配对，将每对奇点之间的一条路上的每边改为二重边，得到一个新图 $G_1$，新图 $G_1$ 中没有奇点，即 $G_1$ 为多重欧拉图。

②若 $G_1$ 中每一对顶点之间有多于 2 条边连接，则去掉其中的偶数条边，直到每一对相邻顶点至多由 2 条边连接，得到图 $G_2$。

③检查 $G_2$ 的每一个回路 $C$，若 $C$ 上重复边的权和超过此圈权和的一半，则把其中的**重边**改为**单边**，**单边**改为**重边**。直到所有回路都复合要求，得到图 $G_3$。

④ $G_3$ 为对应 $G$ 的欧拉回路，即为最优解。



湖南大學
HUNAN UNIVERSITY

第 三 部 分

# 求解过程

**举例**

求图 G 的最优环游

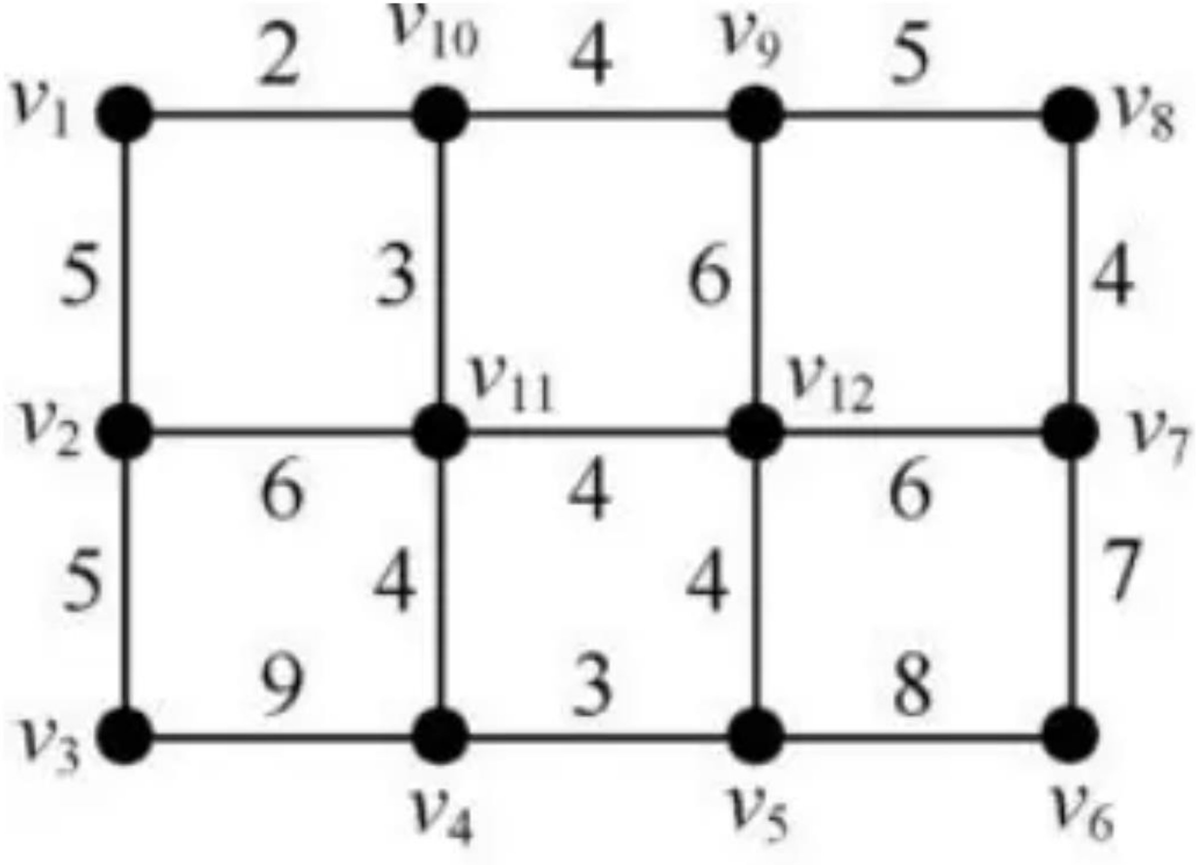



图 G

# 整体过程

① 配对

② 添加重边，去掉奇点

③ 删除偶数条重边

④ 调整回路中的权重问题，

调整单边和重边

# 具体过程

1 配对　图 $G$ 中有 6 个奇点 $v_2, v_4, v_5, v_7, v_9, v_{10}$，把它们配成三对：$v_2$ 与 $v_5$，$v_4$ 与 $v_7$，$v_9$ 与 $v_{10}$。

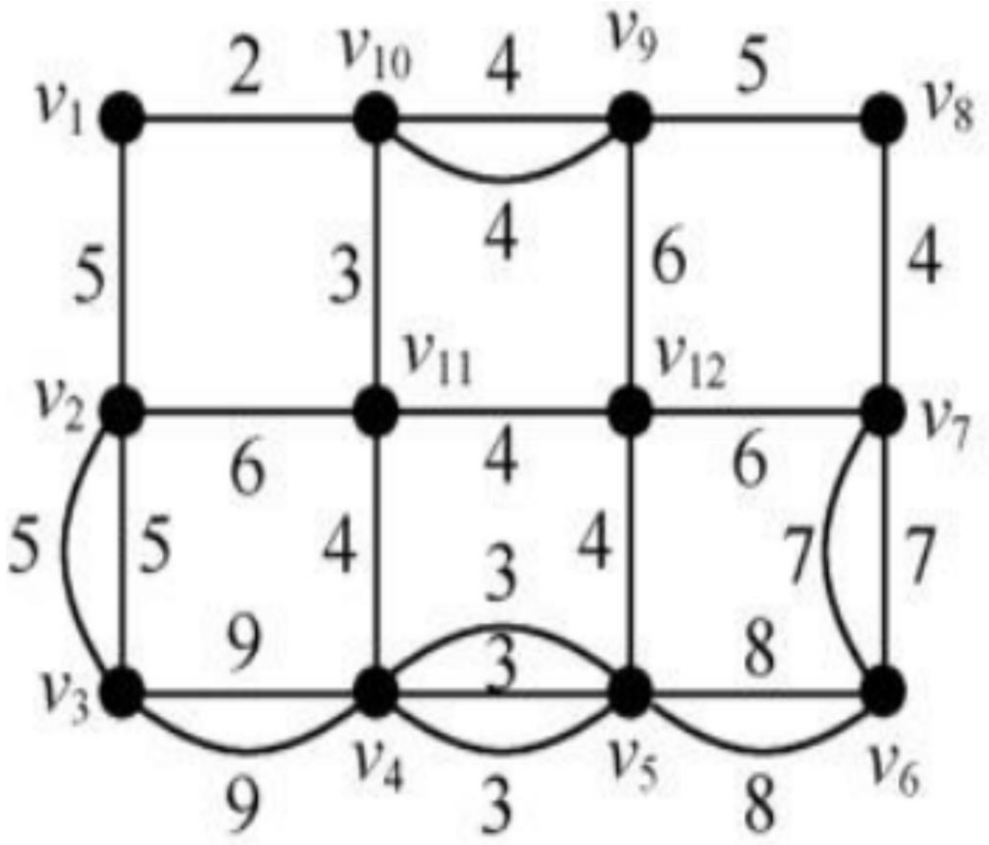
2 添加重边，去掉奇点

在图 $G$ 中，取一条连接 $v_2$ 与 $v_5$ 的路 $v_2v_3v_4v_5$，把边 $(v_2, v_3)$，$(v_3, v_4)$，$(v_4, v_5)$ 作为重复边加入图中；再取 $v_4$ 与 $v_7$ 之间一条路 $v_4v_5v_6v_7$，把边 $(v_4, v_5)$，$(v_5, v_6)$，$(v_6, v_7)$ 作为重复边加入图中；最后在 $v_9$ 和 $v_{10}$ 之间加一条重复边 $(v_9, v_{10})$，则图中没有奇点，是一个欧拉图。

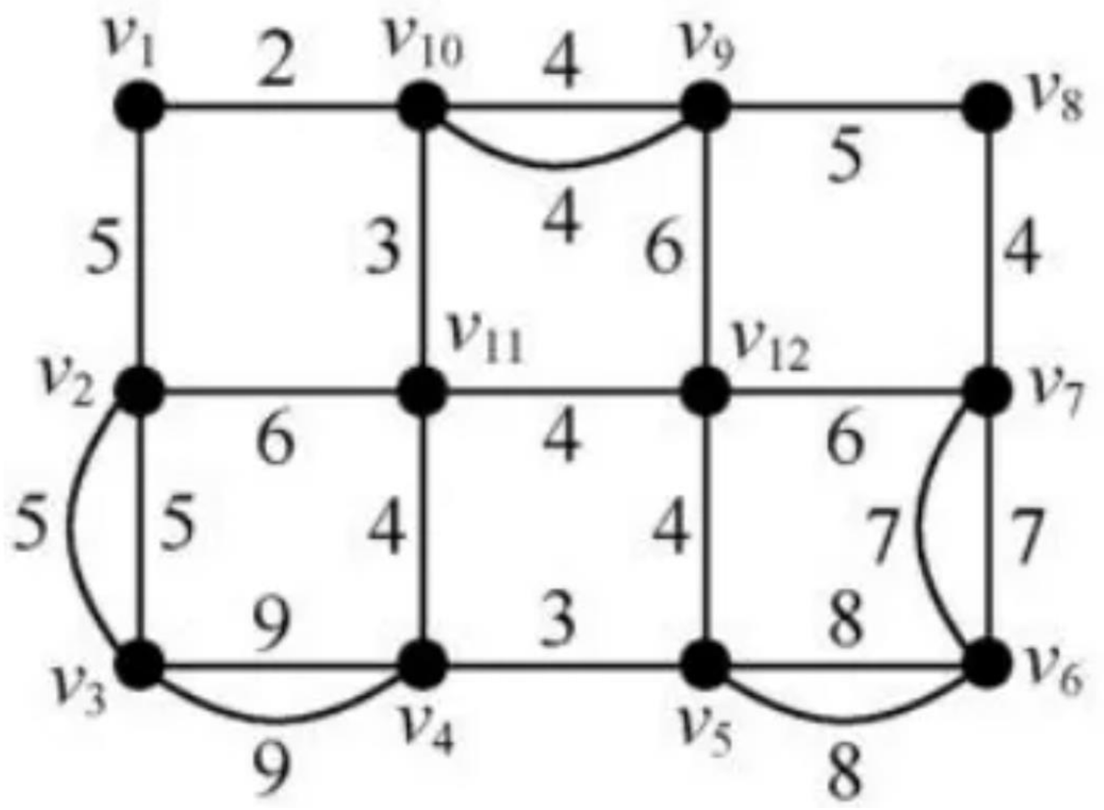
3 删除偶数条重边

注意到，顶点 $v_4$ 与 $v_5$ 之间有 3 条重边，去掉其中 2 条，该图仍是一个欧拉图。

# 具体过程




配对之后的图



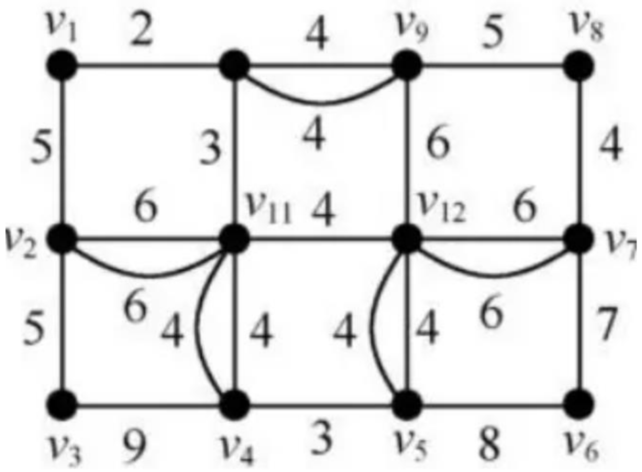

删除偶数条重边之后的图

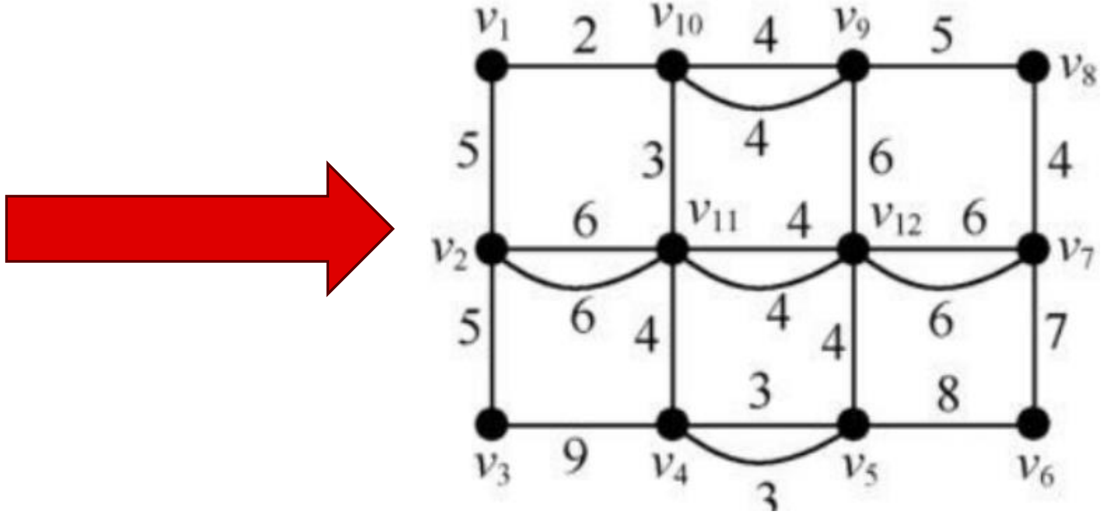
# 具体过程

4 调整回路中的权重问题，
调整单边和重边

1 回路 $v_2v_3v_4v_{11}v_2$ 的总权为 24，重复边的权和为 14，大于该圈总权的一半，于是去掉 $(v_2,v_3)$ 和 $(v_3,v_4)$ 的重复边，加入 $(v_2,v_{11})$ 和 $(v_4,v_{11})$ 的重复边

2 回路 $v_5v_6v_7v_{12}v_5$ 的总权为 25，而重复边权和为 15，去掉 $(v_5,v_6)$ 和 $(v_6,v_7)$ 的重复边，加入 $(v_5,v_{12})$ 和 $(v_7,v_{12})$ 的重复边。



湖南大学
HUNAN UNIVERSITY

第四部分

# 算法具体步骤和性能分析

## •针对有向图

一个无向图是欧拉图的充要条件是，所有顶点的度数都是偶数，即所有点都是偶点。而有向图是欧拉图的充要条件也是所有顶点都是偶点，但有向图中偶点的定义为出度=入度，则奇点的定义便理应为出度≠入度，下给出求简单有向联通图的转化为欧拉图的最小权匹配模型。

设$G=(V,E)$为一简单有向联通图，记$R_i=di^+-d_i^-$，其中$di^+$为顶点$v_i$的出度，$di^-$为顶点$v_i$的入度。建立顶点集$V^+=\{v_i|v_i\in V,R_i>0\}$以及$V^-=\{v_i|v_i\in V,R_i<0\}$，设其长度分别为$n^+$与$n^+$。构建二分图$B=(V^+,V^-,E^{'})$，其中$S=T=V_1$，$E^{'}$的构建参照无向图，亦由$floyd$算法得到最短距离矩阵决定。

引入变量x i j x_{ij}x ij ，表示从顶点v j v_jv j 到v i v_iv i 新增的链数，建立数学规划模型如下：

$$min\ \ w(e_{ij}^{'})x_{ij}$$

$$s.t.\begin{cases}\sum_{i=1}^{n}x_{ij}=-R_j, & j=1,2,...,n^-\\ \sum_{j=1}^{n}x_{ij}=R_i, & i=1,2,...,n^+\\ x_{ij}\in N, & i=1,2,...,n^+;j=1,2,...,n^-\end{cases}$$

•针对无向图

设$G = (V, E)$为一简单无向联通图，$D$为使用$floyd$算法求得的图的最短路程矩阵，$P$为对应的路径矩阵。设$V_1$为图$G$的奇点集，由图论基础知识可证明一简单图的奇点个数为偶数，记其为$n$。构建二分图$B = (S, T, E^{'})$，其中$S = T = V_1$，$E^{'}$的构建如下：

$$w(e_{ij}^{'}) = \begin{cases} D_{S_i T_j}^{'}, & \text{if } S_i \neq T_j \\ \infty, & \text{if } S_i = T_j \end{cases}$$

求该二分图的最小权匹配，引入决策变量$x_{ij} = 0$，$1$来表示$S_i$与$T_j$的匹配关系，若$x_{ij} = 1$则表示与匹配反之则不匹配，可建立数学规划模型如下：

$$min \ \ w(e_{ij}^{'})x_{ij}$$

$$s.t. \begin{cases} \sum_{i=1}^{n} x_{ij} = 1, & j = 1, 2, ..., n \\ \sum_{j=1}^{n} x_{ij} = 1, & i = 1, 2, ..., n \\ x_{ij} = x_{ji}, & i = 1, 2, ..., n; j = 1, 2, ..., n \\ x_{ij} \in \{0, 1\}, & i = 1, 2, ..., n; j = 1, 2, ..., n \end{cases}$$

## 代 码 实 现　最小权匹配的实现

```
function [D,path,min1,path1]=floydf(a,start,terminal)
% --------------------
% a 表示权值矩阵 start 起点 terminal 终点
% ---------------------
% D 表示任意两点间最小路径权值 path 表示路径矩阵
% min1 表示 start 到 terminal 的最小路径权值
% path1 表示 start 到 terminal 的最小路径
% ---------------------
D=a; %赋初值，最小路径权值矩阵开始等于初始的权值矩阵
n=size(D,1);% 顶点数目
path=zeros(n,n); %初始的路径矩阵，全部置为 0
%修改路径矩阵，若两点 ij 之间有边，路径矩阵相应位置置为 j
for i=1:n
for j=1:n
if D(i,j)~=inf
path(i,j)=j;
end
end
end
%插入顶点计算最小路径
for k=1:n % 最多插入 n 个顶点
for i=1:n
for j=1:n
if D(i,k)+D(k,j)<D(i,j) %插入 k 后，得到的两个路径之和比原来的路径权值小
D(i,j)=D(i,k)+D(k,j); %修改 ij 的路径权值
path(i,j)=path(i,k); %在 ij 路径矩阵中插入 k
```

floyd 算法

```
if nargin==3 %如果输入参数是 3 个
min1=D(start,terminal);%直接从最小路径权值矩阵中读出 start 到 terminal 的最小路径权值
%下面是构造从 start 到 terminal 的最小路径
m(1)=start;%起点
i=1;%最小路径中顶点序号
path1=[ ]; %开始路径为空
while path(m(i),terminal)~=terminal %表示如果 m(i)和 terminal 之间还有插入点
k=i+1; %最小路径顶点序号更新（加一个）
m(k)=path(m(i),terminal); %m(i)和 terminal 之间的插入点是 start 到 terminal 最小路径的第 k 个顶点
i=i+1; %序号更新
end
m(i+1)=terminal;%start 到 terminal 最小路径的最后一个顶点是 terminal
path1=m;%生成的最小路径 m 复制给路径 path1
end
```

## 代 码 实 现 最小权匹配的实现

## 最小权匹配算法

```
function [linked,F,PA]=minweightmatch(E)
%--------------------------
% E 为图的邻接矩阵
%--------------------------
% linked 为相互配对的奇点
% F 为新增的路径权值
% PA 为配对奇点之间的最短路径
%--------------------------
% 找出所有奇点
L=length(E);
JD=mod(sum(E>0 & E<inf),2);
p=find(JD==1);
% 弗洛伊德算法求奇点间最短路
[D,path]=floydf(E);
% 构建奇点间二分图
BinV=inf*ones(length(p),length(p));
for i=1:length(p)
for j=1:length(p)
if i~=j
BinV(i,j)=D(p(i),p(j));
end
end
end
% 令无穷大权值为 9999
BinV(BinV==inf)=9999;
LL=length(BinV);
solution=[];
% 整数规划求解最小权匹配
LL=length(BinV);
Aeq=[];
for i=1:LL
```

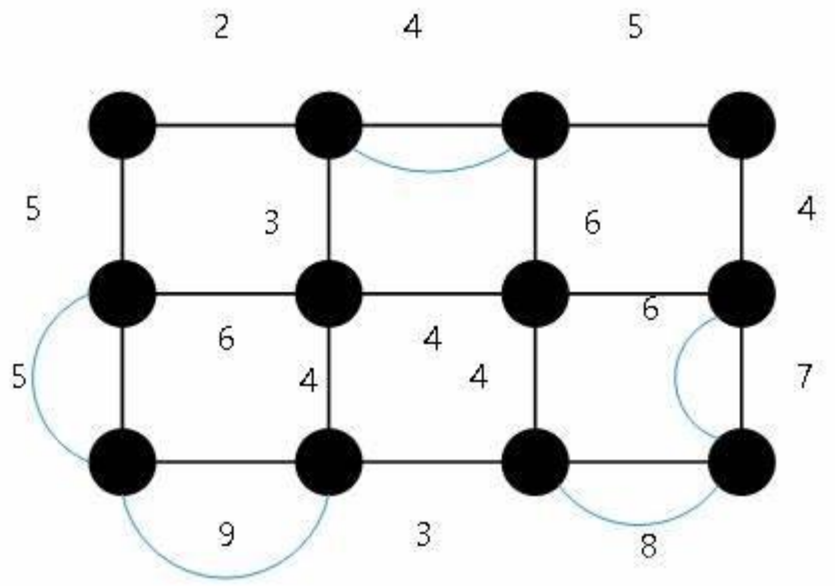
```
Aeq=[Aeq;zeros(1,(i-1)*LL) ones(1,LL) zeros(1,(LL-i)*LL)];
end
for i=1:LL
temp=[zeros(i-1,LL);ones(1,LL);zeros(LL-i,LL)];
temp=reshape(temp,[1,LL*LL]);
Aeq=[Aeq;temp];
end
beq=ones(2*LL,1);
for i=1:LL
for j=i+1:LL
tt=zeros(LL,LL);
tt(i,j)=1;tt(j,i)=-1;
tt=reshape(tt,[1,LL*LL]);
Aeq=[Aeq;tt];
beq=[beq;0];
end
end
lb=zeros(LL*LL,1);ub=ones(LL*LL,1);
options = optimoptions('intlinprog');
options.TolInteger=0.001;
[x,fval]=intlinprog(BinV(:),LL*LL,[],[],Aeq,beq,lb,ub,options);
x=reshape(x,[LL,LL]);
for i=1:LL
for j=i+1:LL
if x(i,j)==1;
solution=[solution;p(i) p(j)];
```

```
kp=x;kp(x~=0 & x~=1)=0;
kkp=find(sum(kp)==0);
p=p(kkp);
for i=1:2:length(p)
solution=[solution;p(i) p(i+1)] ;
end
% 构造配对奇点矩阵
linked=solution;
% 新增路径总权值
F=fval/2;
% 配对奇点最短路
[m,n]=size(linked);
PA(1).path=[];
for i=1:m
temp=[linked(i,1)];
while(temp(end)~=linked(i,2))
temp=[temp path(temp(end),linked(i,2))];
end
PA(i).path=temp;
end
```

## •算法性能分析

### 最小权匹配算法

作为一种解决二分图最大权匹配问题的一种经典算法。它的目标是在一个带权无向图中找到一组边，使得这些边的权重之和最大，且每个顶点至多与一条边相连。




**算法复杂度**:

•Kuhn-Munkres 算法的时间复杂度为 O(｜V｜3)，其中 ｜V｜ 是图中的顶点数。

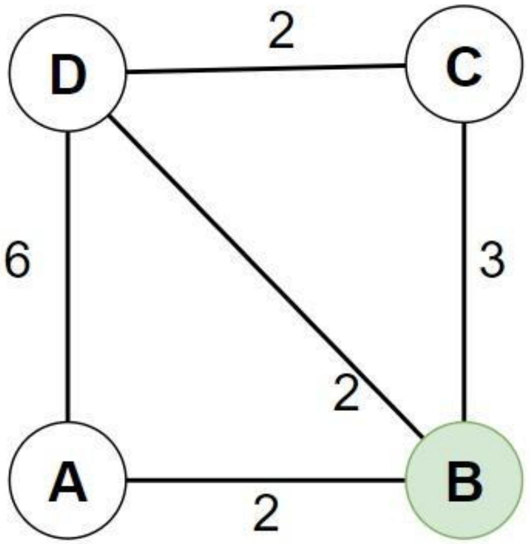
•这个算法适用于解决二分图的最大权匹配问题，有时也被称为**分配问题**。

# •算法性能分析

## Floyd 算法

**Floyd算法**，也被称为**弗洛伊德算法**，是解决任意两点之间最短路径的一种经典算法。它可以处理有向图或带有正负权重（但没有负权重环）的最短路径问题。

时间复杂度分析：
•Floyd算法的时间复杂度为O(n3)，其中n是图中的顶点数。
•这个算法适用于解决图中所有点对之间的最短路径问题。




| | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 2 | -1 | 6 |
| B | 2 | 0 | 3 | 2 |
| C | -1 | 3 | 0 | 2 |
| D | 6 | 2 | 2 | 0 |



优点：

1. 可以找到所有顶点对之间的最短路径长度，而不仅仅是单个源点到其他顶点的最短路径。
2. 相对于其他单源最短路径算法（如Dijkstra算法），Floyd算法在处理大规模稀疏图时具有较好的性能表现。

# 谢 谢 观 看

thank you for watching

参考文献

管梅谷. 奇偶点图上作业法[J]. 数学学报，1960(3):263-266.

郭心月. 运筹学(第四版)[M]. 清华大学出版社，2012.

知乎.Floyd算法详解 通俗易懂,2020.