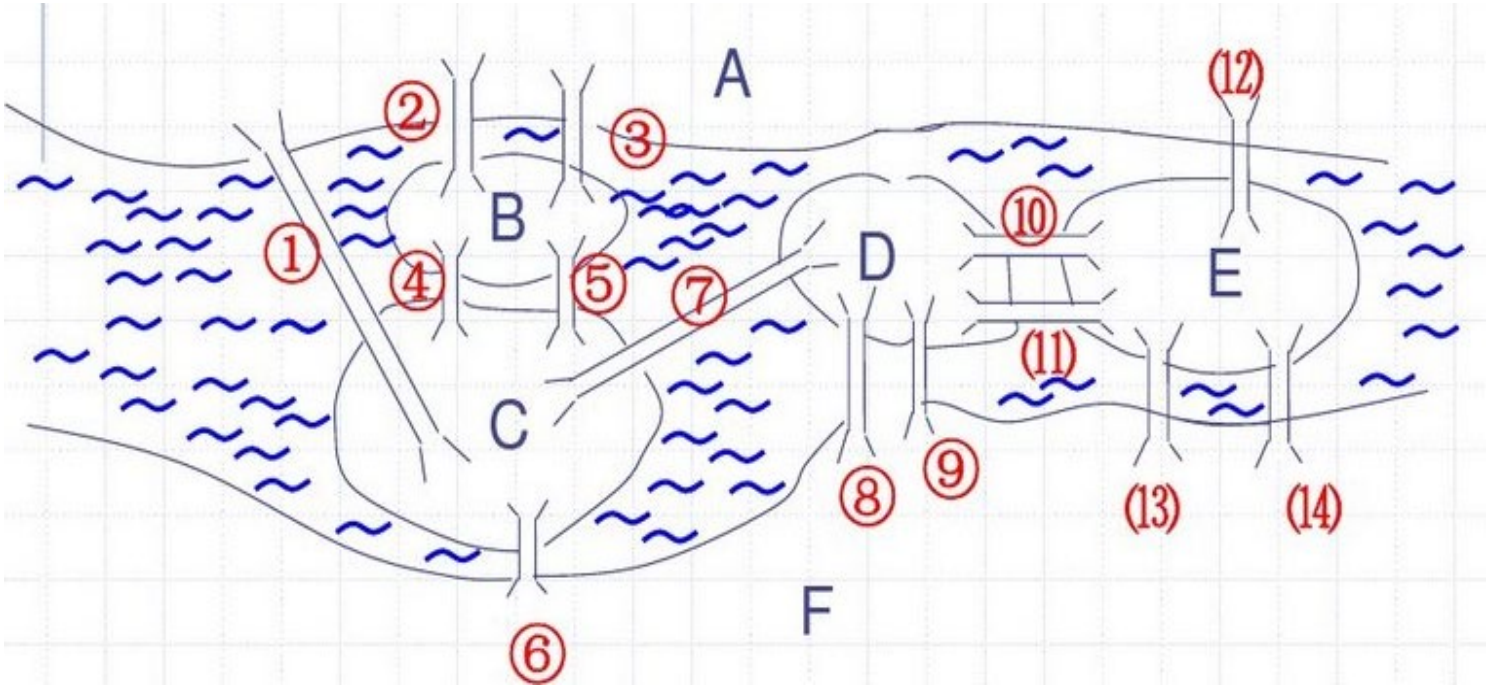
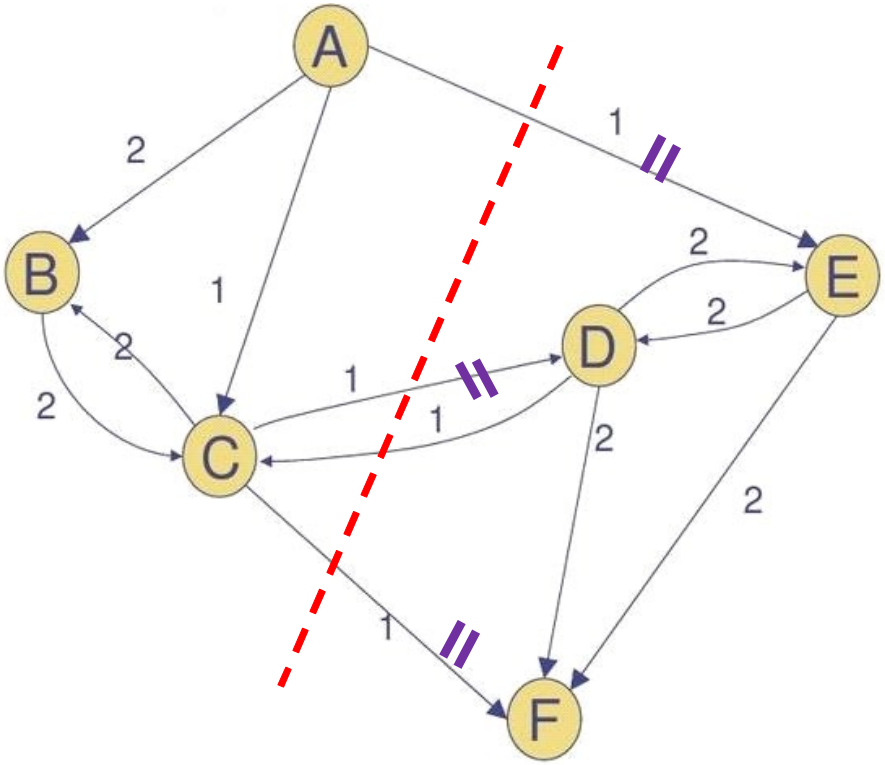
4.6 其它问题

图中A、B、C、D、E、F分别表示陆地和岛屿，1，2，…，14表示桥梁及其编号。河两岸分别互为敌对的双方部队占领，问至少应切几座桥梁（具体指出编号）才能达到阻止对方部队过河的目的。试用图论方法进行分析。



4.6 其它问题



最小割为 $\{AE, CD, CF\}$

4.6 其它问题

在美国职业棒球例行赛中，每个球队都要打162场比赛（对手包括但不限于同一分区里的其他队伍，和同一支队伍也往往会有多次交手），所胜场数最多者为该分区的冠军；如果出现并列第一，则用加赛决出冠军。在比赛过程中，如果发现某支球队无论如何都已经不可能以第一名或者并列第一名的成绩结束比赛，那么这支球队就提前被淘汰了（虽然它还要继续打下去）。

4.6 其它问题

Team	胜	负	剩余	纽约	巴尔的摩	波士顿	多伦多	底特律
纽约	75	59	28	0	3	8	7	3
巴尔的摩	72	62	28	3	0	2	7	4
波士顿	69	66	27	8	2	0	0	0
多伦多	60	75	27	7	7	0	0	0
底特律	49	86	27	3	4	0	0	0

该表是某次美国联盟东区比赛的结果。在该小组分区中，纽约队暂时排名第一，总共胜 75 场，负 59 场，剩余 28 场比赛没打，其中和巴尔的摩还有 3 场比赛，和波士顿还有 8 场比赛，和多伦多还有 7 场比赛，和底特律还有 3 场比赛（还有 7 场与不在此分区的其他队伍的比赛）。

4.6 其它问题

Team	胜	负	剩余	纽约	巴尔的摩	波士顿	多伦多	底特律
纽约	75	59	28	0	3	8	7	3
巴尔的摩	72	62	28	3	0	2	7	4
波士顿	69	66	27	8	2	0	0	0
多伦多	60	75	27	7	7	0	0	0
底特律	49	86	27	3	4	0	0	0

底特律暂时只有 49 场比赛获胜，剩余 27 场比赛没打。如果剩余的 27 场比赛全都获胜的话，是有希望超过纽约队的；即使只有其中 26 场比赛获胜，也有希望与纽约队战平，并在加赛中取胜。然而，根据表里的信息已经足以判断，其实底特律已经没有希望夺冠了，请你不妨来分析推导一下。

南京大学 NANJING UNIVERSITY 工程管理学院 SCHOOL OF MANAGEMENT & ENGINEERING

第七章 图与网络

5. 最小费用流问题

- 给定网络$G=(V,A,C,d)$和经过网络的流量$W(f)=v$，求流在网络上的最佳分布，使总费用最小。

$$\min \quad d(f)=\sum_{(i,j)\in A} d_{ij}f_{ij}$$

$$\sum_{(s,j)\in A} f_{sj}-\sum_{(j,s)\in A} f_{js}=v$$

$$\sum_{(t,j)\in A} f_{tj}-\sum_{(j,t)\in A} f_{jt}=-v$$

$$\sum_{(i,j)\in A} f_{ij}-\sum_{(j,i)\in A} f_{ji}=0, i\neq s,t$$

$$0\leq f_{ij}\leq C_{ij}$$

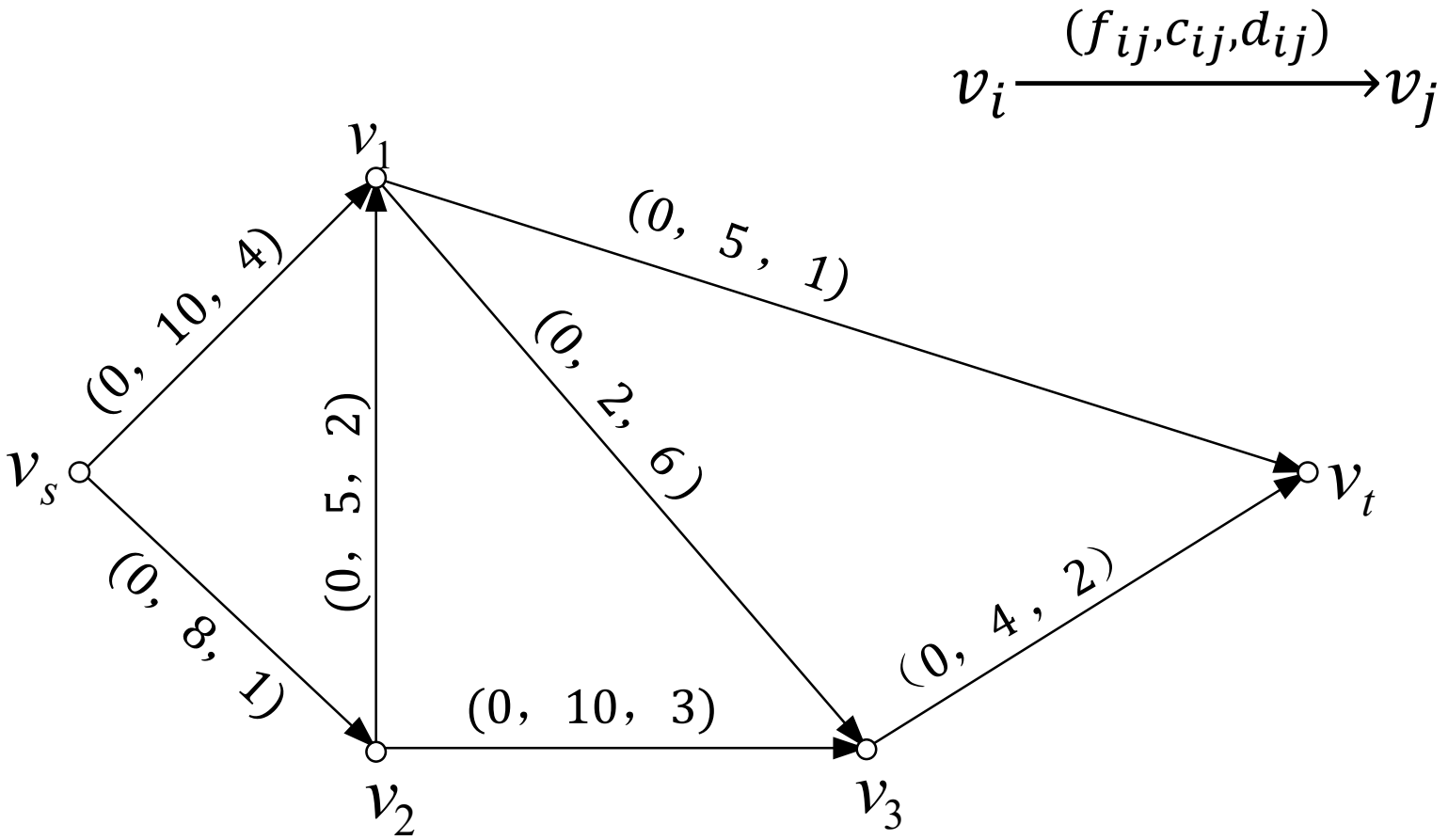
5.2 最小费用增广链

- 增广链费用，最小费用增广链。
- 对于最小费用可行流，沿最小费用增广链调整流，可使流增加，并保持流费用最小。
- 给定初始最小费用可行流，求最小费用增广链，若存在，则沿该增广链调整网络流，直到达到给定的网络流或不存在增广链为止，后一种情况为最小费用最大流。
- 若给定网络流超过最大流，则不可能实现。

5.2 最小费用增广链

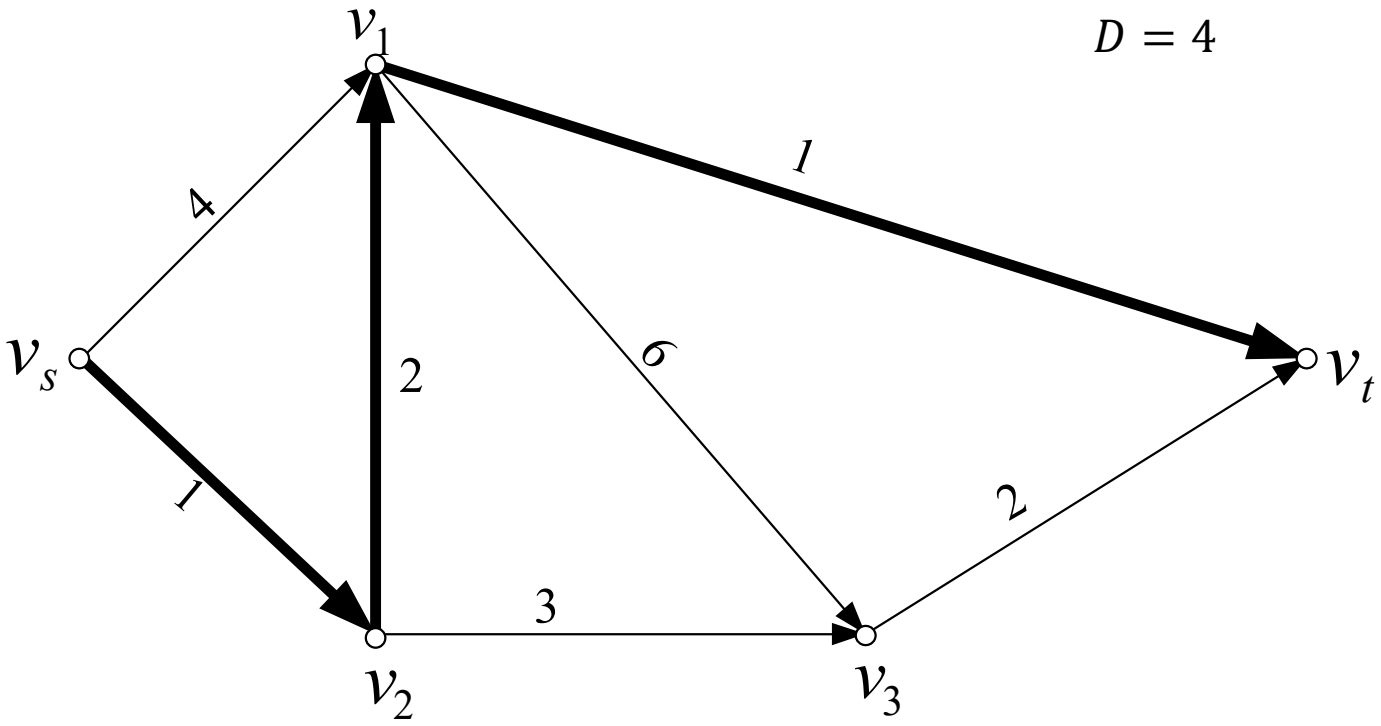
- 生成最小费用可行流的剩余网络：
  - 将饱和弧反向
  - 将非饱和非零流弧加一反向弧
  - 零流弧不变
  - 所有前向弧的权为该弧的费用，后向弧的权为该弧费用的相反数
- 剩余网络又叫长度网络
- 最小费用增广链对应剩余网络的最短路

5.3 最小费用流—例题



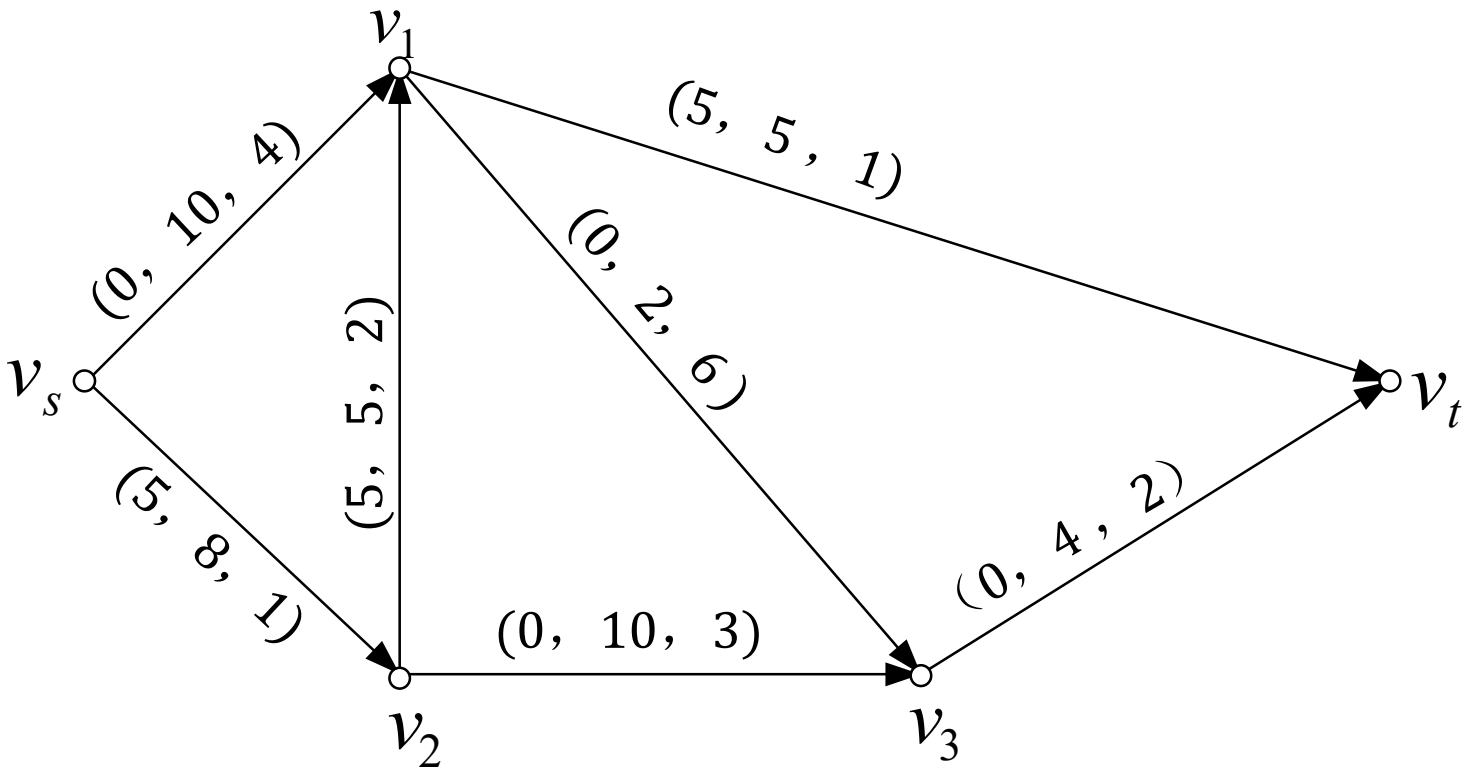
5.3 最小费用流—例题

第1次剩余网络最短路



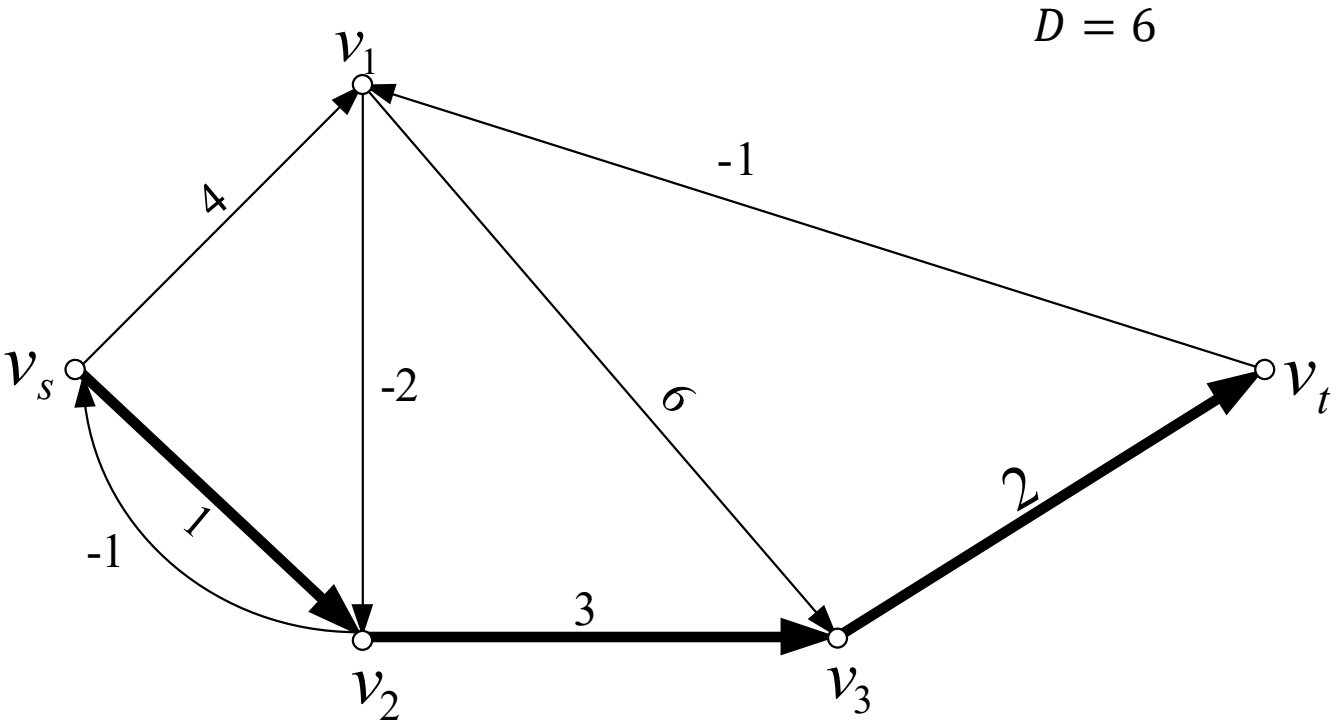
5.3 最小费用流—例题

第1次调整网络流



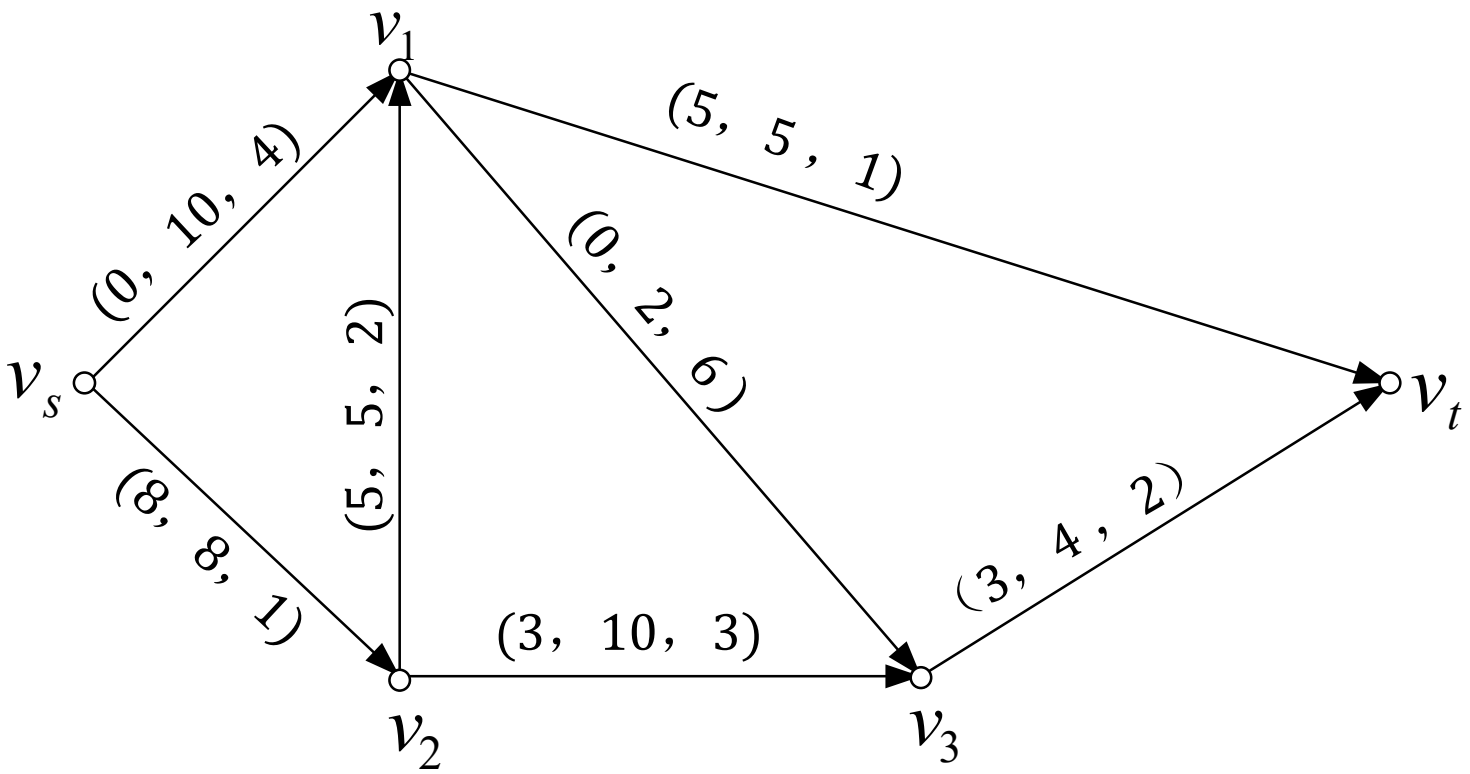
5.3 最小费用流—例题

第2次剩余网络最短路



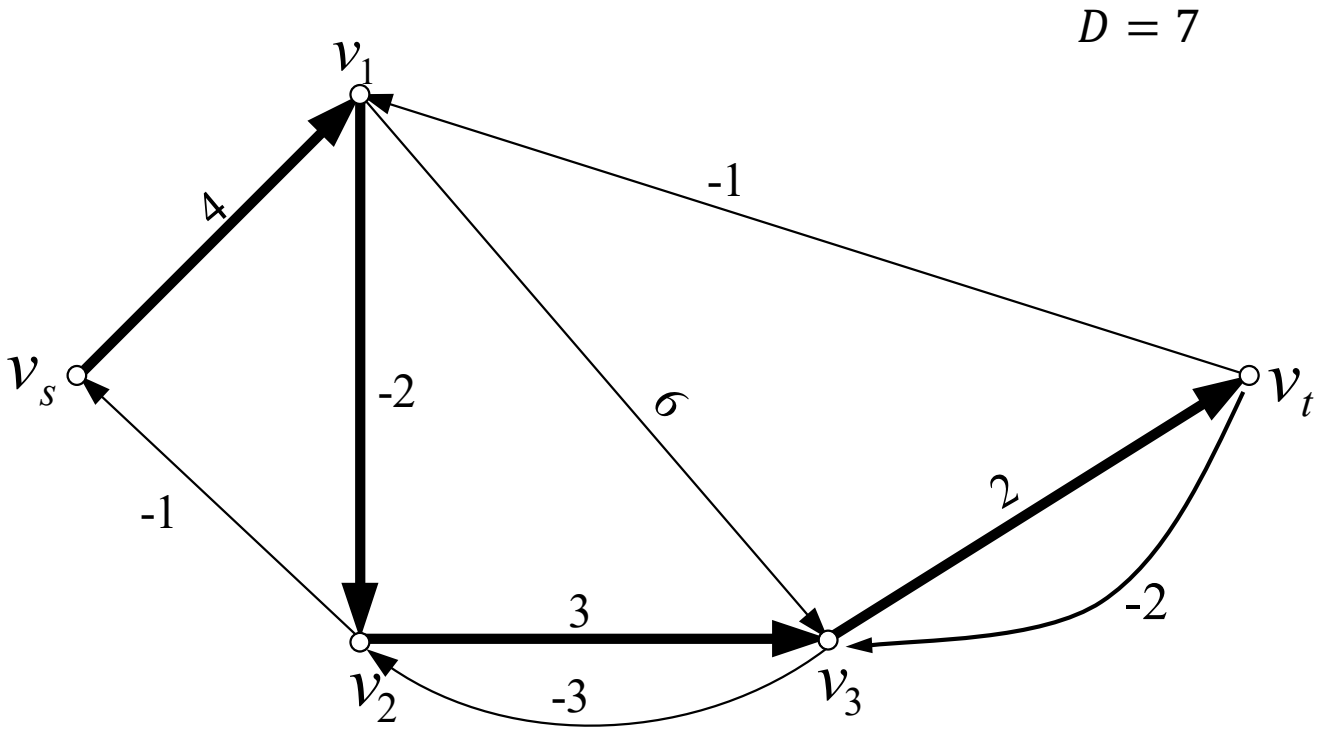
5.3 最小费用流—例题

第2次调整网络流



5.3 最小费用流—例题

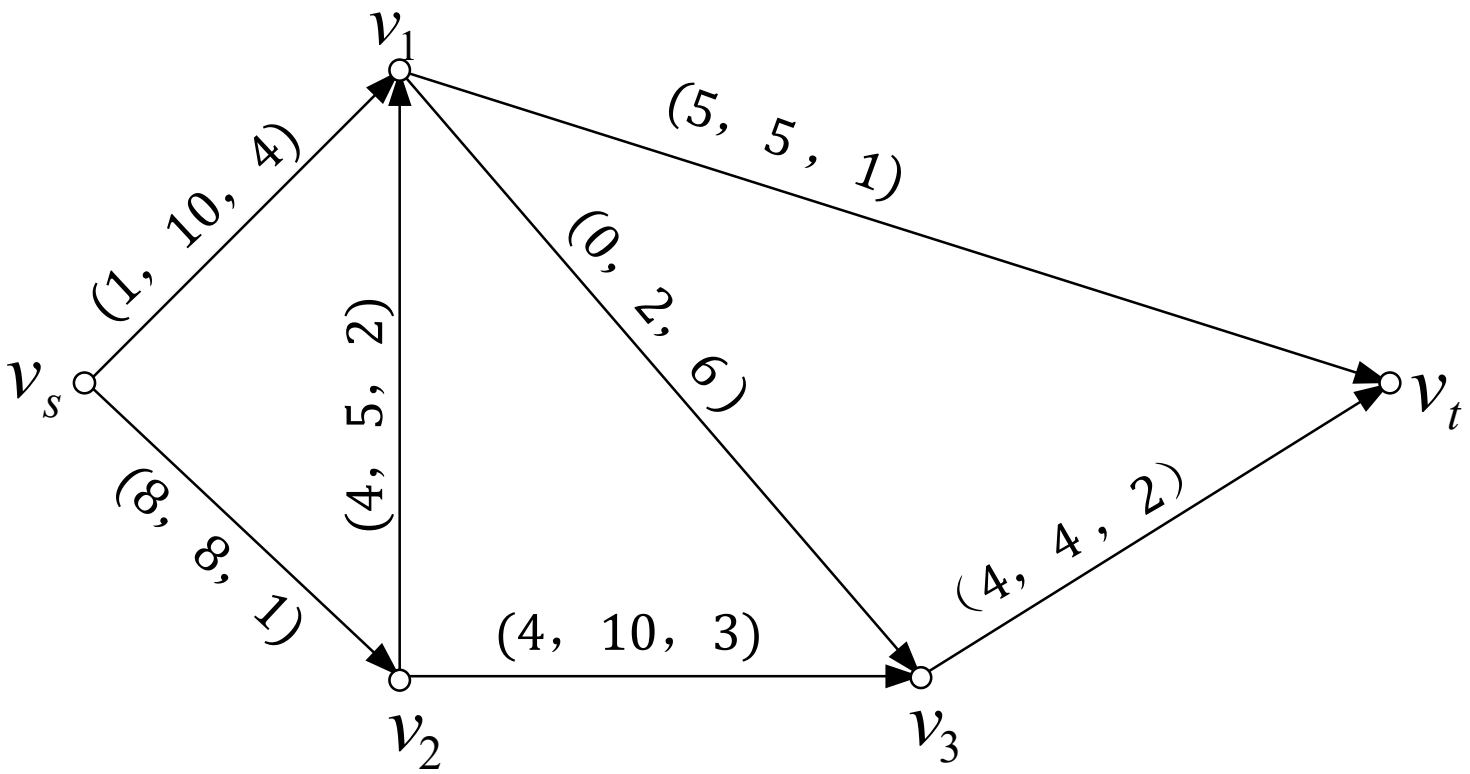
第3次剩余网络最短路



5.3 最小费用流—例题

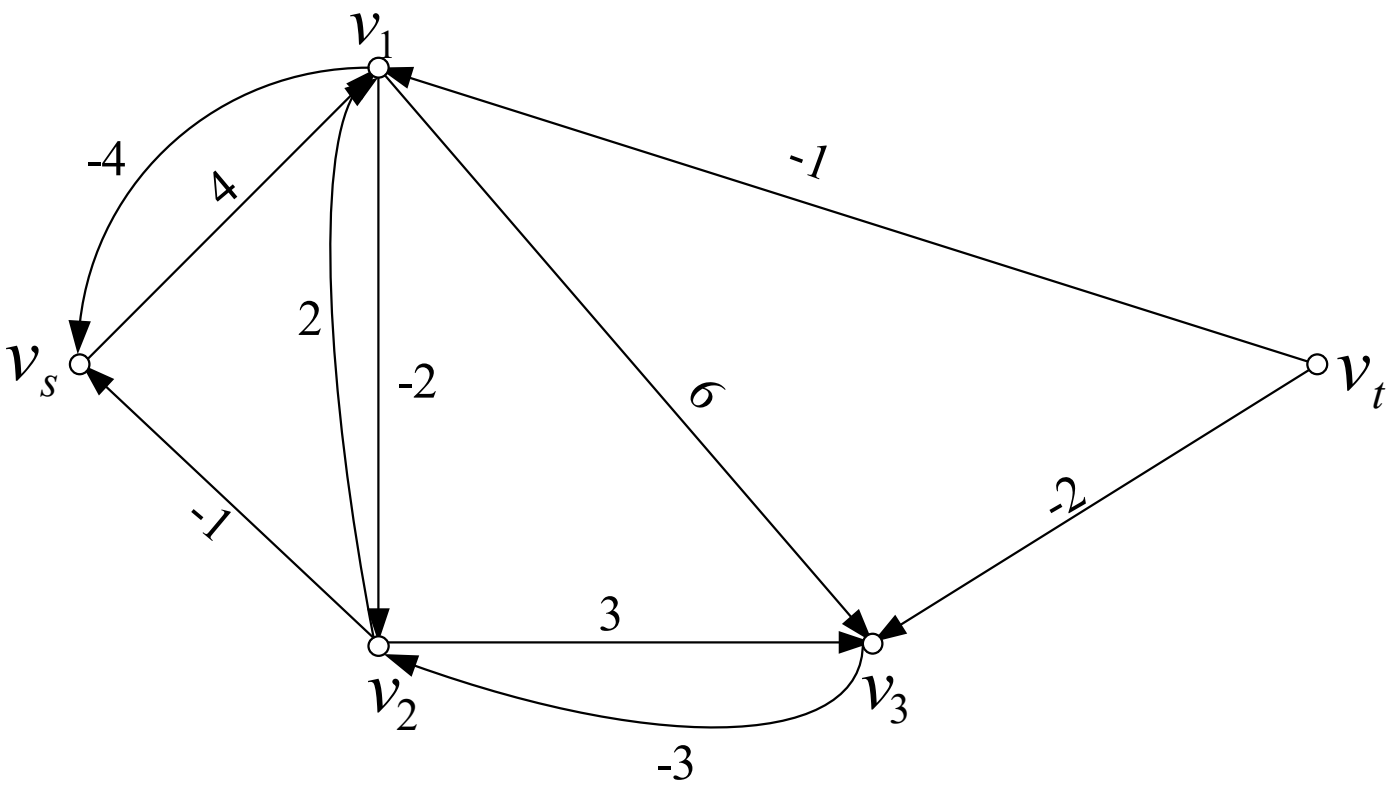
南京大学 NANJING UNIVERSITY 工程管理学院 SCHOOL OF MANAGEMENT & ENGINEERING

第3次调整网络流

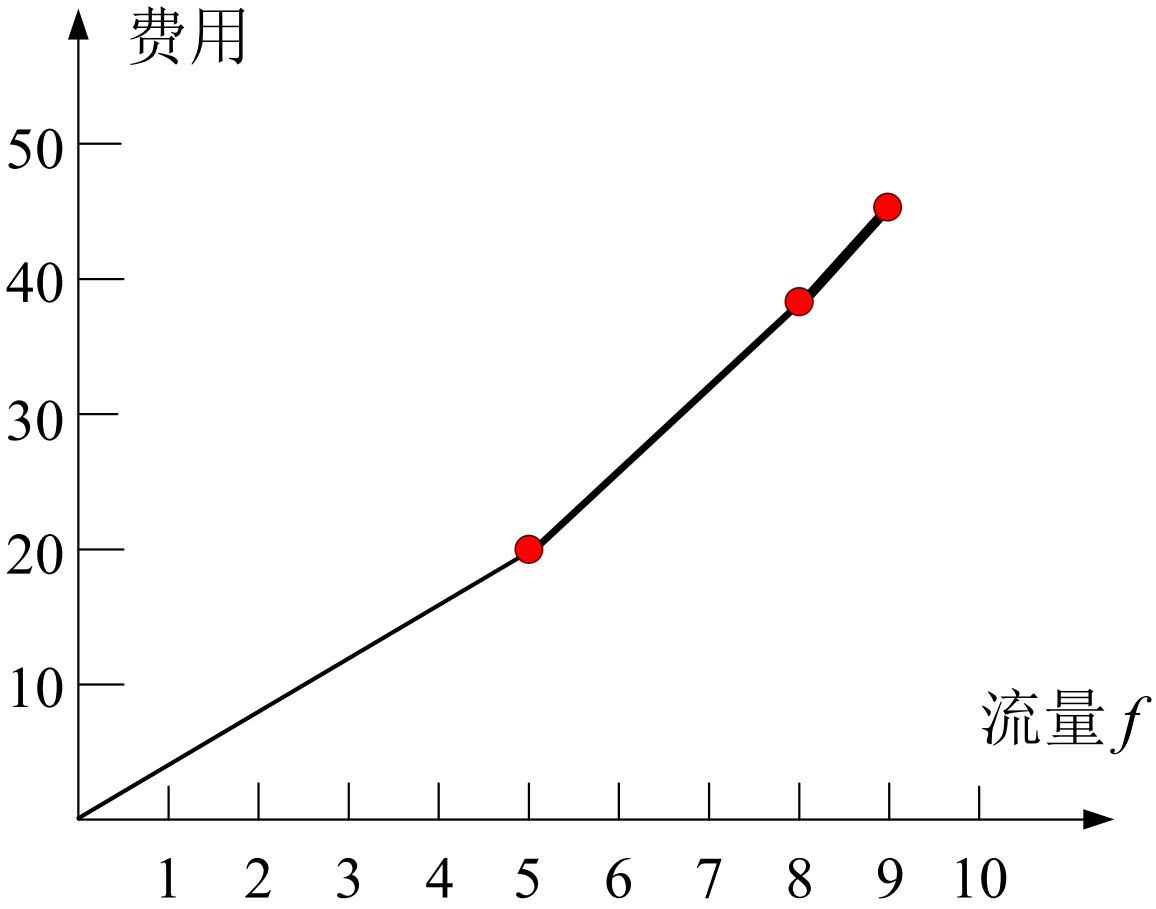


5.3 最小费用流—例题

剩余网络已不存在最短路



5.3 最小费用流—例题



一般形式的最小费用流问题

$$\min \quad c(\mathbf{x}) = \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$s.t. \sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = d_i, \forall i \in V$$

$d_i > 0$, supply node/source

$d_i < 0$, demand node/sink

$d_i = 0$, transshipment node

$$l_{ij} \le x_{ij} \le u_{ij} \rightarrow 0 \le x_{ij} \le u_{ij}$$

— 容量受限的转运问题

— 对于单源单汇的情形，寻找从 s 流到 t 的给定流量的最小费用流，是经典的最小费用流问题

令 $d_s = v,\ d_t = -v,\ d_i = 0$(当$i \neq s,t$) 即可。

最小费用流问题的特例

- **最短路问题**

  令所有弧的容量下界为 0，容量上界为 1。图中某个节点 s 的供需量为 1，某个节点 t 的供需量为 −1。再令所有弧的费用为“弧长”，则此时的最小费用流问题就是最短路问题。

- **运输问题**

  运输问题是研究比较早的最小费用流问题之一，早在1941年Hitchcock就进行了研究，因此运输问题又称为Hitchcock问题。

- **最大流问题**

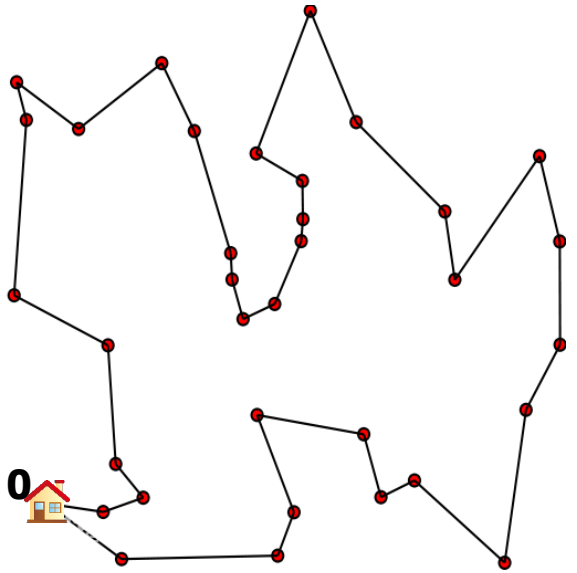
  设s为起点，t为终点，增加弧(t,s)，令$c_{ts}=-1,\ u_{ts}=+\infty$，而令所有其他弧上的费用为 0，所有节点上的供需量全为 0。

旅行售货商问题（TSP）

旅行售货商问题（Traveling Salesman Problem，TSP）

假设有一个旅行商人要从家出发拜访 n 个城市，最后返回家。他知道任意城市之间的距离。他要选择一条每个城市只拜访一次的最短路径。请问他应该按照怎样的次序拜访这些城市？

- 把家和 n 个城市看成节点，家用节点0表示，并用 V 表示 $n+1$ 个节点的集合
- 用 c_{ij} 表示从节点 i 到节点 j 的距离 $(i \neq j)$



决策变量：

$$x_{ij} = \begin{cases} 1 & \text{如果旅行商直接从城市}i\text{到城市}j \\ 0 & o.w. \end{cases}$$

旅行售货商问题（TSP）

旅行售货商问题（Traveling Salesman Problem，TSP）

假设有一个旅行商人要从家出发拜访 n 个城市，最后返回家。他知道任意城市之间的距离。他要选择一条每个城市只拜访一次的最短路径。请问他应该按照怎样的次序拜访这些城市？

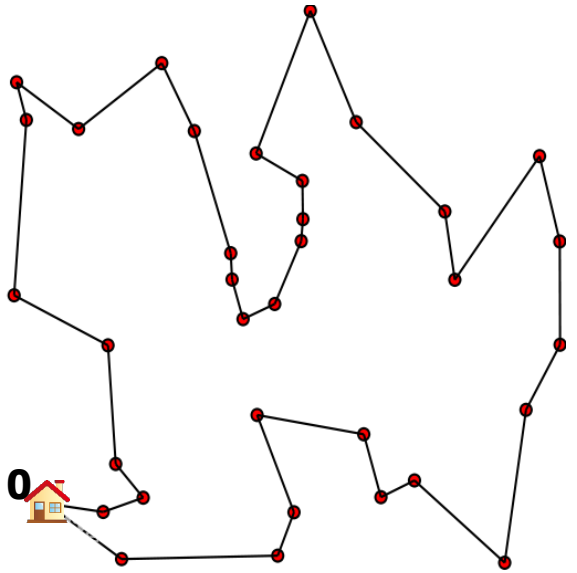
目标函数：

$$\min \sum_{i \in V} \sum_{j \in V: j \neq i} c_{ij} x_{ij}$$

约束条件：

离开城市i一次且仅一次：$\sum_{j \in V: j \neq i} x_{ij} = 1, \forall i \in V$

到达城市j一次且仅一次：$\sum_{i \in V: i \neq j} x_{ij} = 1, \forall j \in V$



旅行售货商问题（TSP）

旅行售货商问题（Traveling Salesman Problem，TSP）

假设有一个旅行商人要从家出发拜访 n 个城市，最后返回家。他知道任意城市之间的距离。他要选择一条每个城市只拜访一次的最短路径。请问他应该按照怎样的次序拜访这些城市？

子回路问题 (subtour)

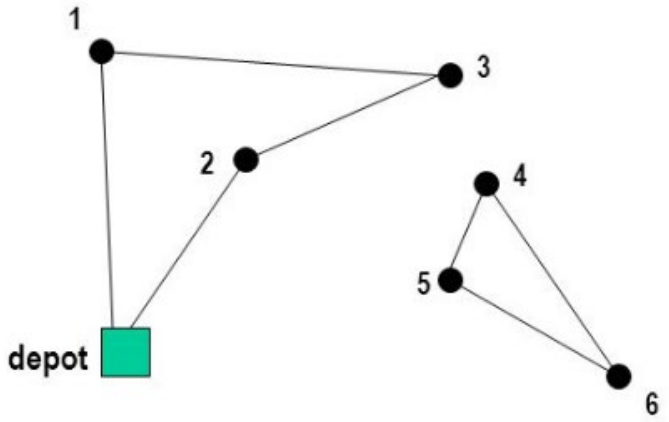
约束条件：

去除子回路 (subtour elimination)

两种常见的去除子回路方法：

（1）加入subtour elimination约束

（2）使用Miller-Tucker-Zemlin（MTZ）建模方法



旅行售货商问题（TSP）

旅行售货商问题（Traveling Salesman Problem，TSP）

假设有一个旅行商人要从家出发拜访 n 个城市，最后返回家。他知道任意城市之间的距离。他要选择一条每个城市只拜访一次的最短路径。请问他应该按照怎样的次序拜访这些城市？

约束条件：

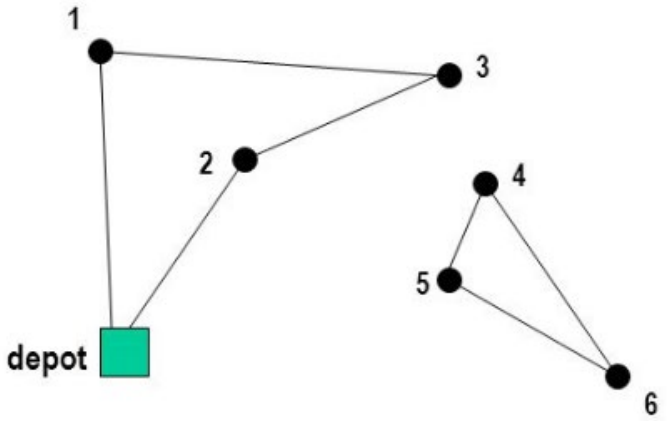
去除回路 (subtour elimination)

(1)

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1, \forall S \subset N = \{1, \ldots, n\}, S \neq \emptyset$$

or

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \forall S \subset N, 2 \leq |S| \leq n - 1$$



旅行售货商问题（TSP）

旅行售货商问题（Traveling Salesman Problem，TSP）

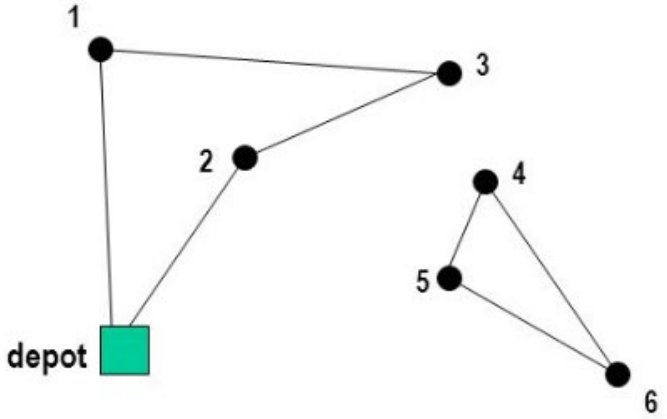
假设有一个旅行商人要从家出发拜访 n 个城市，最后返回家。他知道任意城市之间的距离。他要选择一条每个城市只拜访一次的最短路径。请问他应该按照怎样的次序拜访这些城市？

约束条件：

去除回路 (subtour elimination)

(2) 增加决策变量 u_i 表示访问城市 i 的顺序

$$u_j \geq 1 + u_i x_{ij}, \forall i \neq 0, j \neq 0$$



↷ 上面的约束条件是线性约束吗？

↷ 如果不是，如何线性化？

$$u_j \geq 1 + u_i + (x_{ij} - 1)n, \forall i \neq 0, j \neq 0$$

$$\begin{cases} u_j \geq 1 + u_i, if\ x_{ij} = 1, namely\ j\ is\ visited\ immediately\ after\ i; \\ u_j \geq 1 + u_i - n, if\ x_{ij} = 0, it\ is\ always\ satisfied\ because\ u_i \leq n. \end{cases}$$

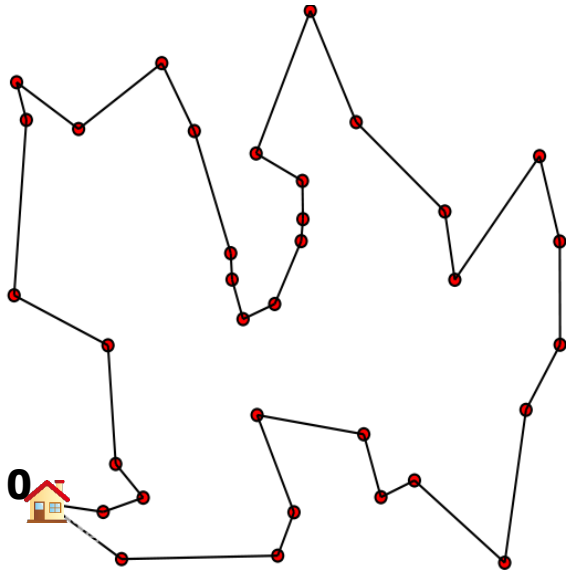
旅行售货商问题（TSP）

旅行售货商问题（Traveling Salesman Problem，TSP）

假设有一个旅行商人要从家出发拜访 n 个城市，最后返回家。他知道任意城市之间的距离。他要选择一条每个城市只拜访一次的最短路径。请问他应该按照怎样的次序拜访这些城市？

$$\min \sum_{i\in V}\sum_{j\in V:j\neq i} c_{ij}x_{ij}$$

$$s.t.\begin{cases} \sum_{j\in V:j\neq i} x_{ij} = 1, \forall i \in V, \\ \sum_{j\in V:j\neq i} x_{ij} = \sum_{j\in V:j\neq i} x_{ji}, \forall i \in V, \\ u_j \geq 1 + u_i + (x_{ij} - 1)n, \forall i \neq 0, j \neq 0, \\ x_{ij} \in \{0,1\}, \forall i \in V, j \in V, j \neq i, \\ u_0 = 0, u_i \geq 1, \forall i \in V, i \neq 0. \end{cases}$$



NP-Hard; Concorde TSP Solver http://www.math.uwaterloo.ca/tsp/concorde.html

车辆路径问题（VRP）

车辆路径问题（Vehicle Routing Problem，VRP）

假设有一定数量的客户，各自有不同数量的货物需求，配送中心向客户提供货物，由一个车队负责分送货物，组织适当的行车路线，目标是使得客户的需求得到满足，并能在一定的约束下，达到诸如路程最短、成本最小、耗费时间最少等目的。

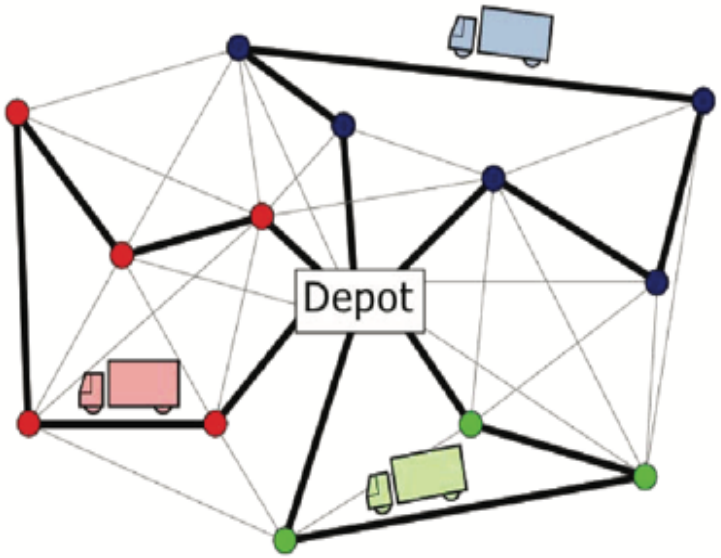
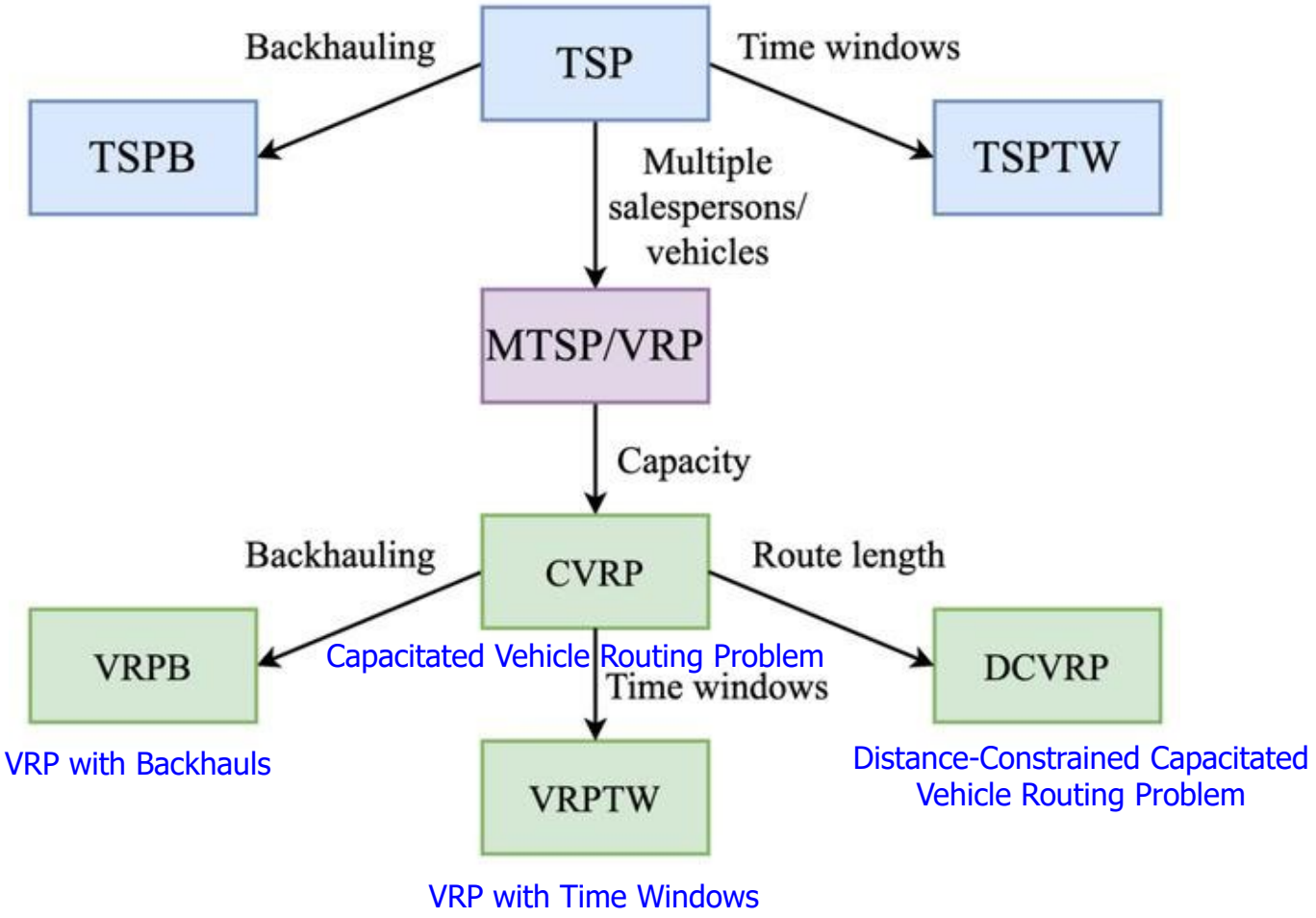


Figure: Vehicle routing problem

车辆路径问题（VRP）


TSP
Backhauling
Time windows
TSPB
TSPTW
Multiple salespersons/ vehicles
MTSP/VRP
Capacity
CVRP
Backhauling
Route length
VRPB
Capacitated Vehicle Routing Problem
Time windows
DCVRP
VRP with Backhauls
VRPTW
Distance-Constrained Capacitated Vehicle Routing Problem
VRP with Time Windows

南京大学 NANJING UNIVERSITY
工程管理学院 SCHOOL OF MANAGEMENT & ENGINEERING

第七章 图与网络

6. 网络单纯形
Network Simplex

一般形式的最小费用流问题

$$\min \quad c(\mathbf{x}) = \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$s.t. \sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = d_i, \forall i \in V$$

$d_i > 0$, supply node/source

$d_i < 0$, demand node/sink

$d_i = 0$, transshipment node

$$l_{ij} \le x_{ij} \le u_{ij} \rightarrow 0 \le x_{ij} \le u_{ij}$$

— 容量受限的转运问题

— 对于单源单汇的情形，寻找从 s 流到 t 的给定流量的最小费用流，是经典的最小费用流问题

令 $d_s = v,\ d_t = -v,\ d_i = 0$(当$i \neq s,t$) 即可。

容量无限的最小费用流问题

首先考虑容量无限的最小费用流问题，即假设所有弧上的容量下界为0，上界为无穷大。

$$\min \ \mathbf{c}^T\mathbf{x}$$

$$s.t. \quad \mathbf{Bx} = \mathbf{d}$$

$$\mathbf{x} \geq \mathbf{0}$$

这里包含 n 个等式约束，但其中只有 $n-1$ 个约束是独立的，因此可以任意去掉其中的某一个约束。

关联矩阵$\boldsymbol{B}$的秩为 $n-1$。

引理 关联矩阵B的列构成一组基的充要条件是它所对应的弧为支撑树。

证明：已假设流网络在不考虑方向时是连通的，所以$r(\mathbf{B})=n-1$。记$\mathbf{B}$的$n-1$列构成的子矩阵为$\mathbf{B}_1$。

必要性：若$\mathbf{B}_1$为一组基，则$r(\mathbf{B}_1)=n-1$。$\mathbf{B}_1$所对应的$n-1$条弧如果不连通，则至少含有一个圈，因此$r(\mathbf{B}_1)<n-1$，矛盾。因此$\mathbf{B}_1$所对应的$n-1$条弧一定是连通的，即这些弧构成一棵支撑树。

充分性：$\mathbf{B}_1$所对应的$n-1$条弧构成一棵支撑树，则$r(\mathbf{B}_1)=n-1$。因此$\mathbf{B}_1$是一组基。

基与支撑树

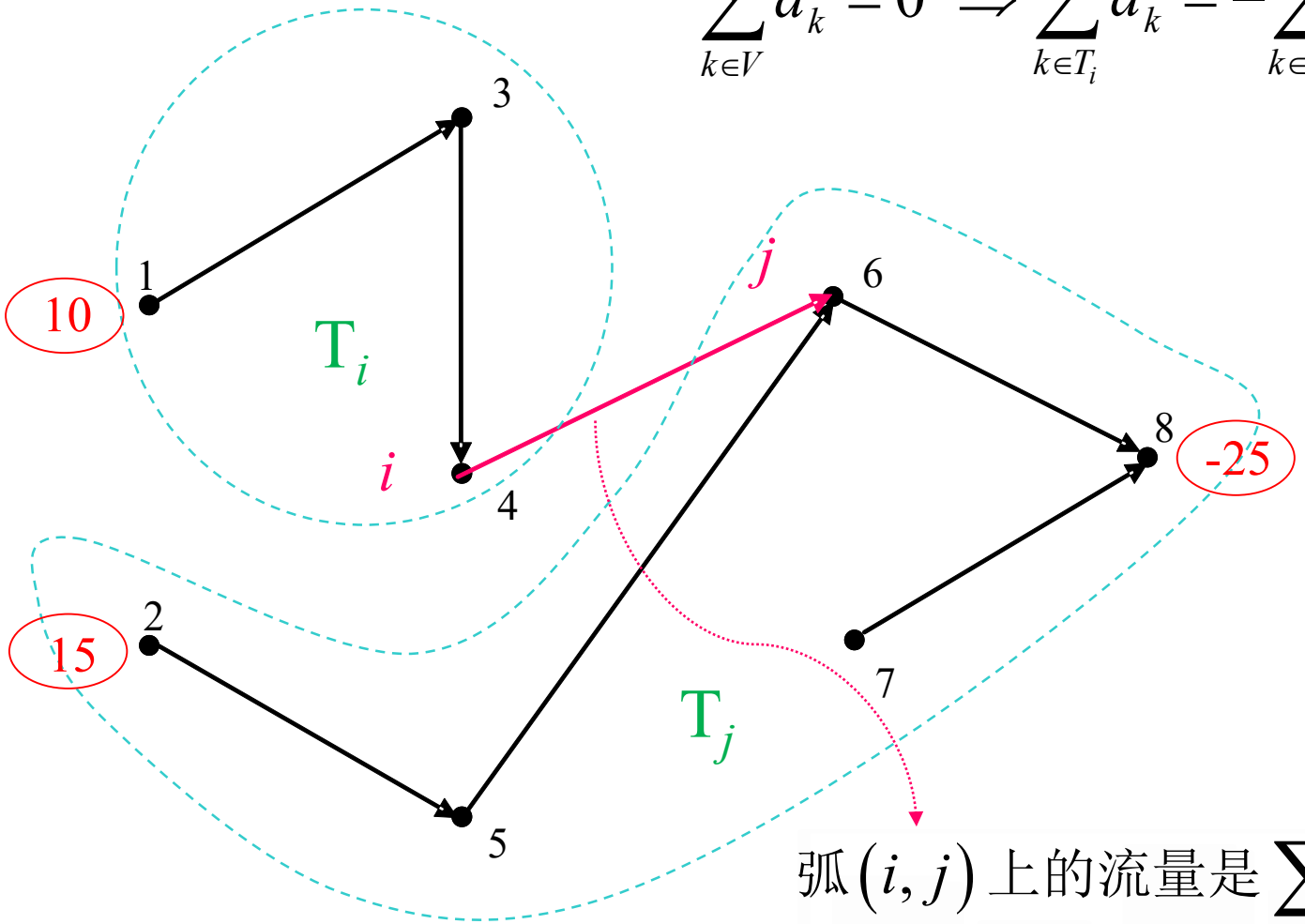
按照一般单纯形算法的表示方法，基变量的值为 $\mathbf{B}_1^{-1}\mathbf{d}$。这里把 $\mathbf{B}_1$ 所对应的弧集合（支撑树）用 T 表示，则

- 所有非树弧（非基弧）对应于非基变量，其上的流量为 0；
- 只有 T 中的弧（树弧，或称基弧）对应于基变量，其上的流量可以为正数（在退化的情况下也可能为 0）。

于是，确定基变量的值的问题在网络上等价于：当给定了支撑树 T 之后，确定树弧上的流量并使之满足流量守恒条件。

$\mathbf{Bx}=\mathbf{d}$ 有解的必要条件是 $\sum_{k\in V} d_k = 0$

$$\sum_{k\in V} d_k = 0 \Rightarrow \sum_{k\in T_i} d_k = -\sum_{k\in T_j} d_k$$



弧(i,j)上的流量是$\sum_{k\in T_i} d_k$

可行基/可行树

- 当弧(i,j)上的流量确定后，可以将(i,j)从 T 中删去（节点i上的供需量应当相应地进行修改），然后重复这一过程，直到所有树弧上的流量都得到确定为止。

- 但是，上述过程所确定的流 **x** 仍然不一定是可行的，即某些树弧上的流量可能为负数。我们把使得相应的流 **x** 为可行流的基 $\mathbf{B}_1$ 称为**可行基**，支撑树 T 称为**可行树**。

- 与单纯形算法只对基本可行解操作对应，网络单纯形算法只对可行树进行操作。

最优性检验

为了判断什么时候可行树所对应的流是最优流，算法仍然利用检验数进行判断。

检验数值为 $\mathbf{c}-\boldsymbol{\pi}\mathbf{B}$，这里 $\boldsymbol{\pi}$ 为对偶变量。

弧(i,j)对应的检验数：$c_{ij}^{\pi}=c_{ij}-\pi_i+\pi_j$

在容量无限这一特殊的问题中，互补松弛条件为：

$$\text{当}x_{ij}=0\text{时，}\ c_{ij}^{\pi}=c_{ij}-\pi_i+\pi_j\geq 0;$$

$$\text{当}x_{ij}>0\text{时，}\ c_{ij}^{\pi}=c_{ij}-\pi_i+\pi_j=0.$$

← 只有树弧满足

检验数计算

- 任意选定一个节点（通常称为“根”），令它的势为0；
- 然后利用 $c_{ij}^{\pi} = c_{ij} - \pi_i + \pi_j = 0$, 计算与它相邻的其它节点上的势，如此反复就可以方便地获得所有节点上的势。
- 如果如此确定的对偶变量 π_i 同时使得

$$\text{当} x_{ij} = 0 \text{时}, \quad c_{ij}^{\pi} = c_{ij} - \pi_i + \pi_j \geq 0;$$

也成立，则 **x** 就是原问题的最优解。

表示检验数非负
π_i 为对偶可行解

弧进基

- 如果 **x**不是最优解，则一定存在一条非树弧 (p,q)，使得

$$c_{pq}^{\pi} = c_{pq} - \pi_p + \pi_q < 0$$

此时，弧 (p,q)可以进入基。

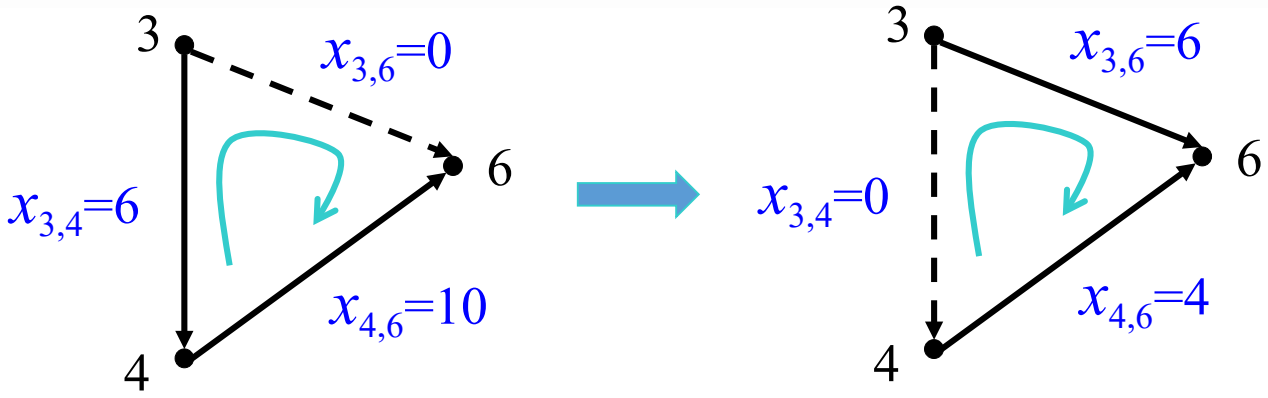
$T \cup \{(p,q)\}$一定含有一个唯一的圈W，把弧(p,q)的方向定义为W的方向。W的费用为

$$\begin{aligned} c(W) &= \sum_{(i,j)\in W^+} c_{ij} - \sum_{(i,j)\in W^-} c_{ij} \\ &= \sum_{(i,j)\in W^+} (c_{ij} - \pi_i + \pi_j) - \sum_{(i,j)\in W^-} (c_{ij} - \pi_i + \pi_j) \\ &= c_{pq}^{\pi} < 0 \end{aligned}$$

← W为负费用圈，沿W增广流量将使总费用下降

弧出基

- 如何找出原来基中的一条弧出基？
  - 令增广的流量等于W^-所有弧上当前流量中的最小值，而取到该最小值的弧出基。
  - 如果$W^- = \varnothing$，则原问题是无界的，即最小费用可以趋于负无穷大。



——求解容量无限的最小费用流问题

Step 0 获得一个初始的可行树 T 及其对应的基本可行解 $\mathbf{x}$。

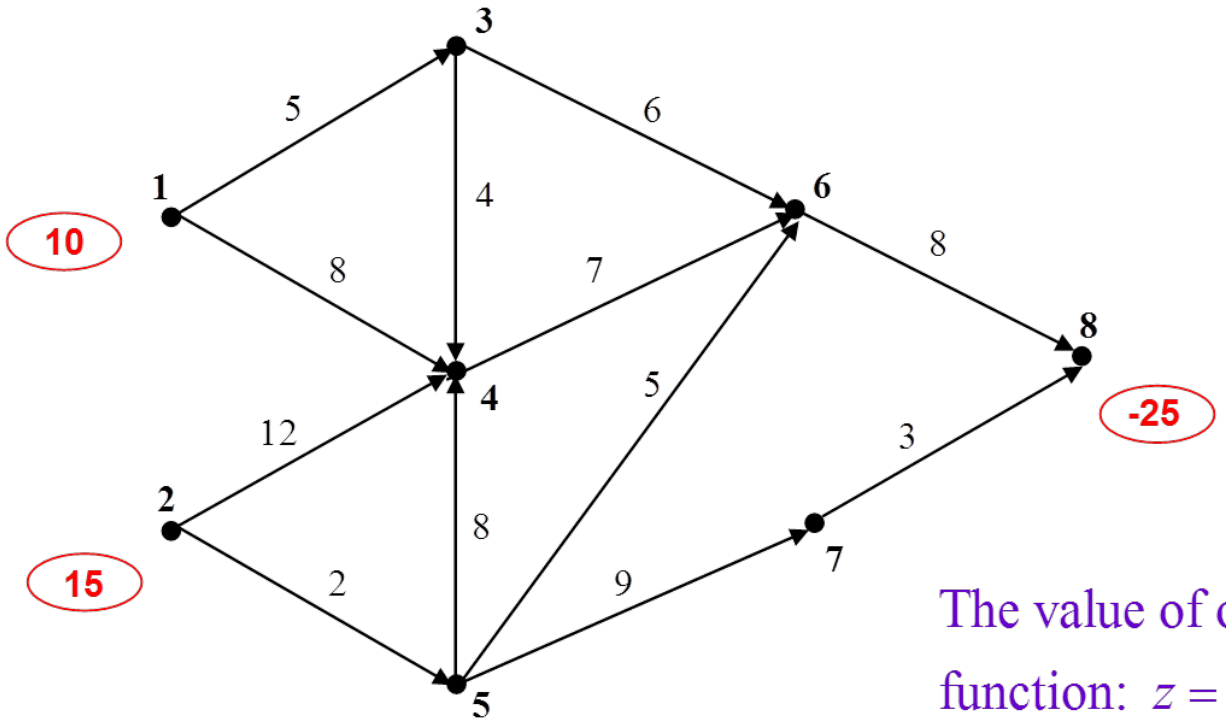
Step 1 计算对偶变量 $\boldsymbol{\pi}$。

Step 2 判断是否为最优解：若是，停止；否则选定一个进基变量（即选进基弧 (p,q)）。

Step 3 选定一个出基变量（即选出基弧），如果找不到这样的弧，则原问题无界，停止；否则转下一步。

Step 4 设 W 为 $T \cup \{(p,q)\}$ 所含的圈。沿 W 的正向增广流量，即修改 $\mathbf{x}$ 及对应的可行树 T，回到 Step 1。

网络单纯形 — 举例



The value of objective function: $z = 465$

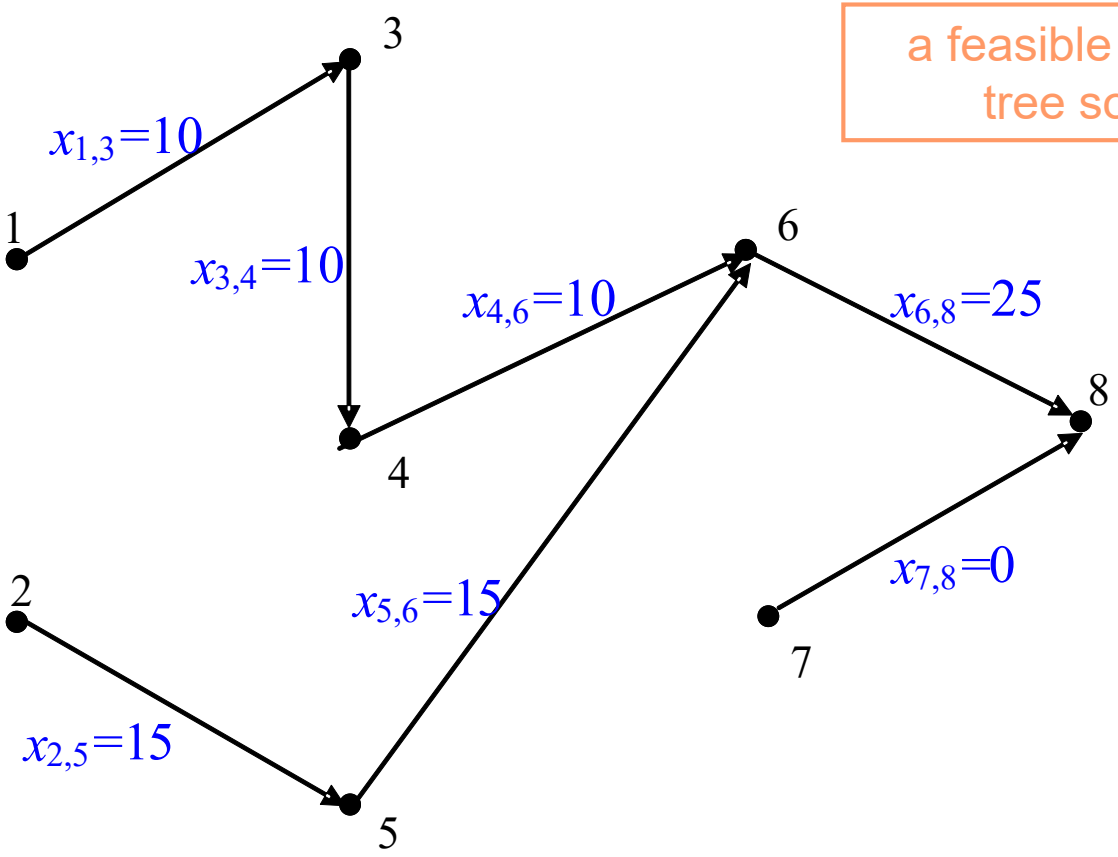
We initialize the method with the basic feasible solution

$$x_{1,3} = 10 \quad x_{3,4} = 10 \quad x_{4,6} = 10 \quad x_{6,8} = 25$$

$$x_{2,5} = 15 \quad x_{5,6} = 15 \quad x_{7,8} = 0$$

All other arcs are nonbasic, and the corresponding variables are zero.

网络单纯形 — 举例



a feasible spanning tree solution

网络单纯形 — 举例

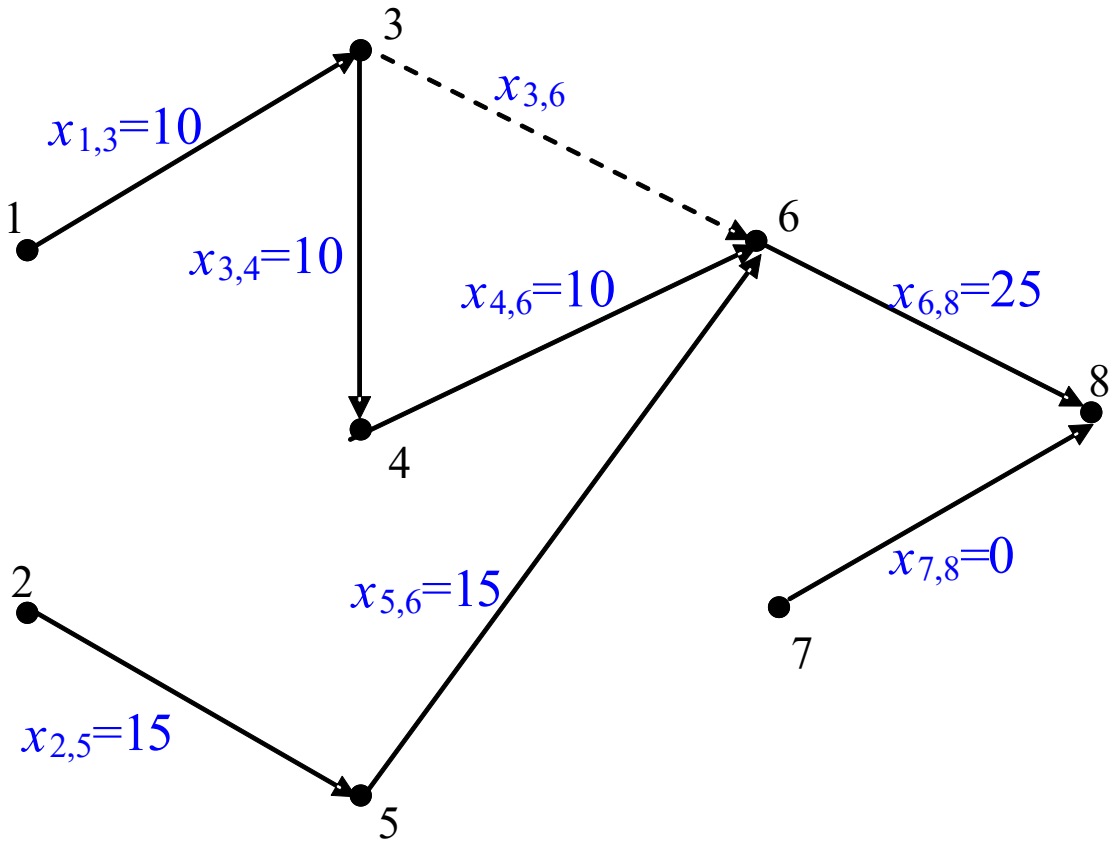
令 y_i 为对偶变量，计算各节点的势：

$$y_1 = 0$$
$$y_3 = y_1 - 5 = -5$$
$$y_4 = y_3 - 4 = -9$$
$$y_6 = y_4 - 7 = -16$$
$$y_8 = y_6 - 8 = -24$$
$$y_7 = y_8 + 3 = -21$$
$$y_5 = y_6 + 5 = -11$$
$$y_2 = y_5 + 2 = -9$$

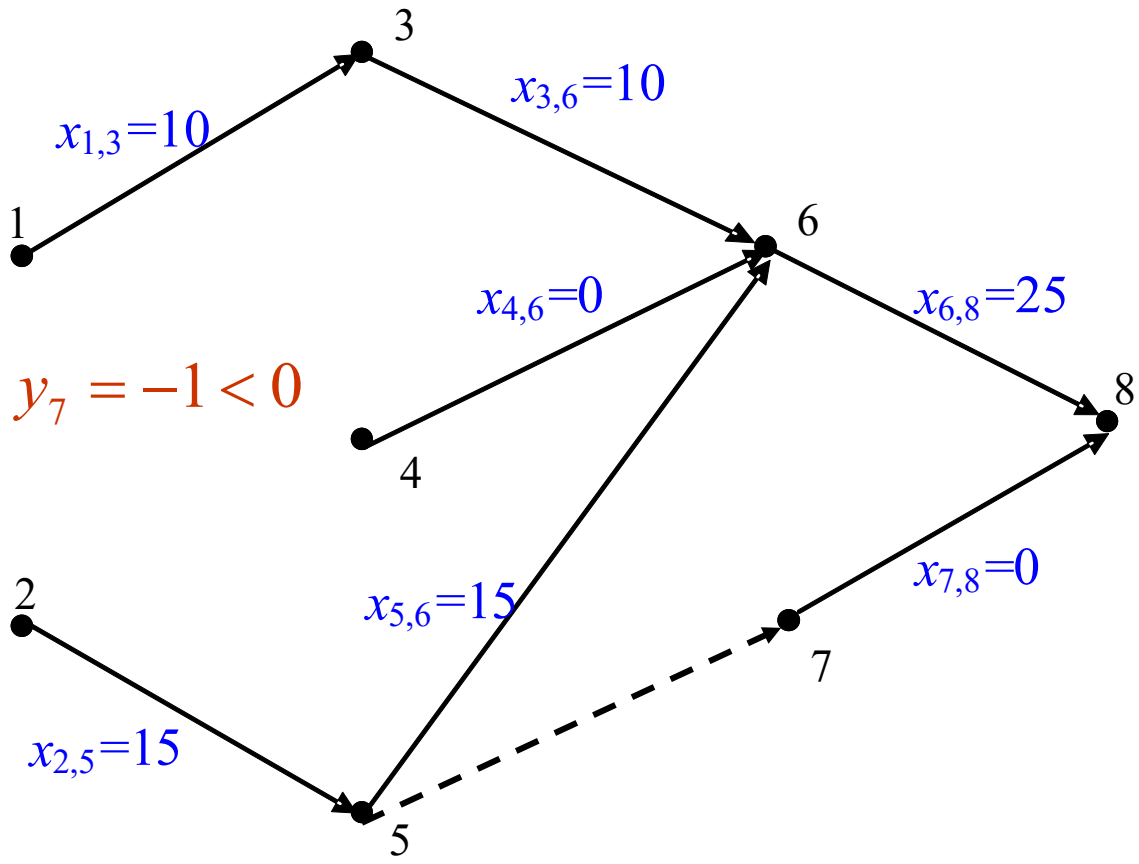
再计算各非树弧的检验数：

$$\hat{c}_{1,4} = 8 - y_1 + y_4 = -1 < 0$$
$$\hat{c}_{3,6} = 6 - y_3 + y_6 = -5 < 0$$
$$\hat{c}_{5,4} = 10 - y_5 + y_4 = 10$$
$$\hat{c}_{2,4} = 12 - y_2 + y_4 = 12$$
$$\hat{c}_{5,7} = 9 - y_5 + y_7 = -1 < 0$$

网络单纯形 — 举例



网络单纯形 — 举例

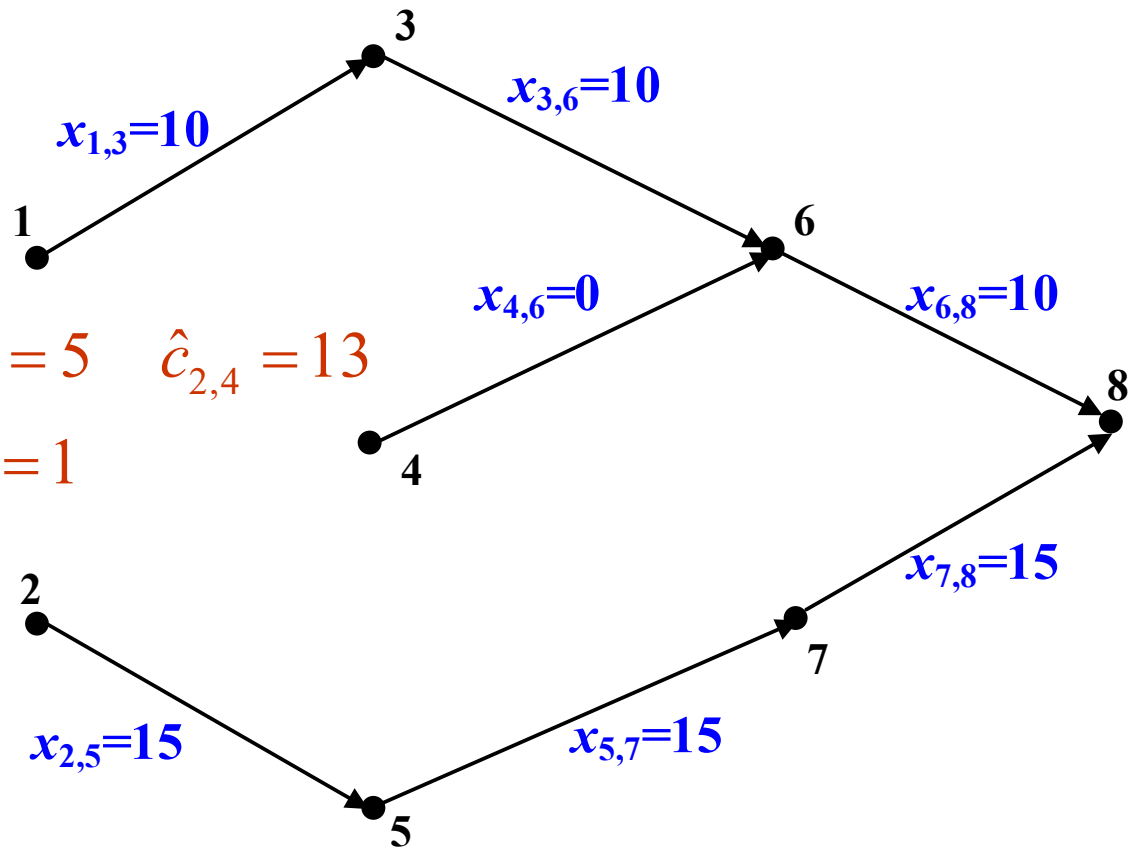


$$\hat{c}_{5,7} = 9 - y_5 + y_7 = -1 < 0$$

Setting $y_1 = 0$, the simplex multipliers are

$$y = \begin{pmatrix} 0 & -4 & -5 & -4 & -6 & -11 & -16 & -19 \end{pmatrix}^T$$

网络单纯形 — 举例



$\hat{c}_{1,4}=4 \quad \hat{c}_{3,4}=5 \quad \hat{c}_{2,4}=13$

$\hat{c}_{5,4}=11 \quad \hat{c}_{5,6}=1$

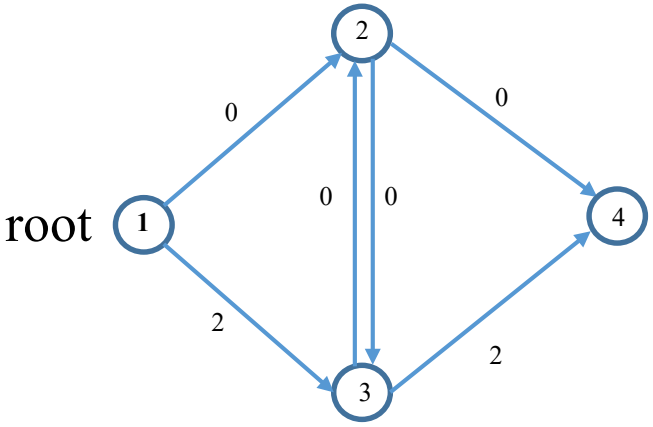
Setting $y_1=0$, the simplex multipliers are

$$y=\begin{pmatrix}0 & -5 & -5 & -4 & -7 & -11 & -16 & -19\end{pmatrix}^T$$

处理退化和循环的方法

定义(强可行树)——假定计算节点上的势时所选定的“根节点”是固定的。对于可行树 T 中的一条树弧 (i,j)，如果从根到 j 的通路过节点 i，则 (i,j) 称为远离根节点的弧(downward pointing arc)。如果 T 中的所有流量为 0 的弧都是远离根节点的弧，则称可行树 T 为**强可行树**。

如果网络单纯形算法中生成的所有可行树都是强可行树，则这些树互不相同。因此算法一定不会出现循环。



$$T_1=\{(1,3),(3,2),(3,4)\}$$

$$T_2=\{(1,3),(2,3),(3,4)\}$$

处理退化和循环的方法（续）

- 如何保证只对强可行树进行处理呢？
  - 要求初始的基本可行解对应于一棵强可行树。
  - 要求旋转变换只生成强可行树。

设旋转变换之前，强可行树为 T，(p,q)为进基弧，$T\cup\{(p,q)\}$包含的圈为 W。

假设$W^- \neq \varnothing$，令$\delta = \min\left\{x_{ij}\middle|(i,j)\in W^-\right\}$，$\overline{W} = \left\{(i,j)\in W^-\middle| x_{ij}=\delta\right\}$，则$\overline{W}$为所有可能的出基弧的集合。

记节点$\overline{p}$是 T中从根节点 r到节点 p的路与圈 W的第一个交点，并选取出基弧(k,l)为从 $\overline{p}$出发沿圈 W的正向前进时第一次所遇到的$\overline{W}$中的弧。由此得到的$\overline{T} = T\cup\{(p,q)\}\setminus\{(k,l)\}$仍然是强可行树。

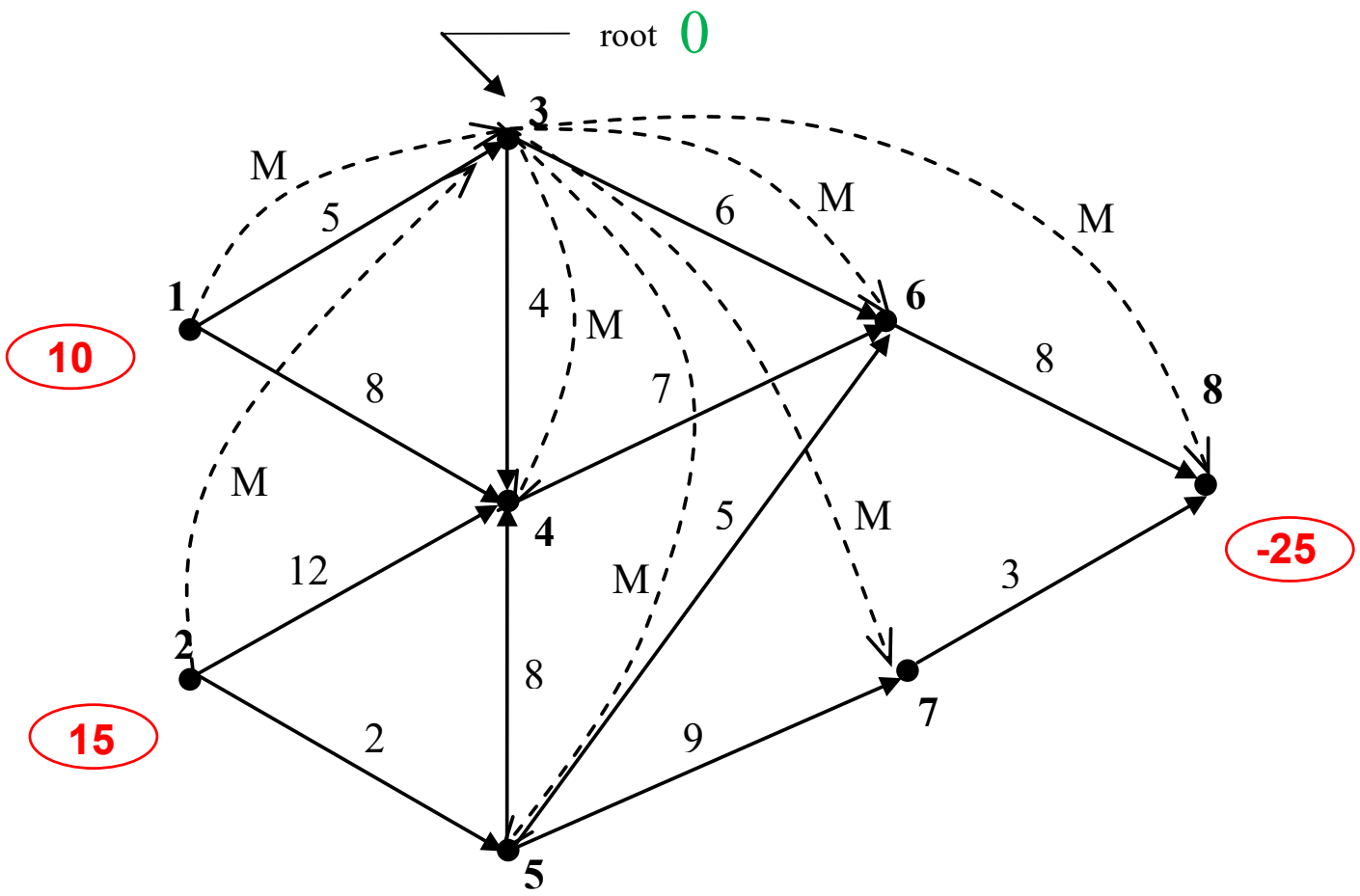
初始基本可行解

一般可以采用大-M法构造初始的强可行树。

- 首先在原问题的网络中加入一个人工节点 0，并假设其供需量为 $d_0 = 0$。
- 然后对原网络中的所有节点 i，按如下步骤加入人工弧：如果 $d_i > 0$，则加入人工弧 $(i,0)$；否则加入人工弧 $(0,i)$。
- 记所有人工弧的集合为 $T^{(0)}$，所有人工弧上的费用假定为一个充分大的正数 M。

—— 称原网络加入人工弧后的新网络为人工网络

网络单纯形 — 举例



人工网络最优解

$T^{(0)}$是人工网络的一棵强可行树，因此可以对人工网络上的最小费用流问题直接应用网络单纯形算法求解。

由于M是一个充分大的正数，因此当算法终止时，有3种情况：

1) 人工网络上的最小费用流问题有有界的最优解，且最优解中所有人工弧上的流量为0，则原问题也有有界的最优解，且人工网络上非人工弧上的流量正好就是原问题的最优解。

2) 人工网络上的最小费用流问题有有界的最优解，且最优解中某些人工弧上的流量不为0，则原问题是不可行的。

3) 人工网络上的最小费用流问题没有有界的最优解（即最优值趋向负无穷），则原问题也没有有界的最优解。

$$M > (n-1)C/2, \quad \text{where } C = \max\{|c_{ij}| : (i,j) \in A\}$$

容量有界的情形

网络单纯形算法可以直接推广到容量有界的最小费用流问题。

$$\min \ \mathbf{c}^T\mathbf{x}$$
$$s.t. \quad \mathbf{Bx} = \mathbf{d}$$
$$l_{ij} \le x_{ij} \le u_{ij}$$

此时，基可以用所有弧的一个划分 (T, L, U) 来表示，其中

- T 是一棵支撑树
- L 是非树弧中流量等于下界的弧的集合
- U 是非树弧中流量等于上界的弧的集合

容量有界的情形 — 初始可行解

定义——假定计算节点上的势时所选定的“根节点”是固定的。在可行树结构(T,L,U)中，如果树弧中所有流量等于下界的弧都是远离根节点的，并且树弧中所有流量等于上界的弧都不是远离根节点的，则(T,L,U)是强可行树结构。

为了从一个初始的强可行树结构开始迭代，仍采用大-M法构造人工网络。

- 首先增加人工节点0，并假设其供需量为$d_0=0$。
- 然后对原网络中的所有节点i，按如下步骤加入人工弧：如果$d_i>0$，则加入人工弧$(i,0)$；否则加入人工弧$(0,i)$。
- 记所有人工弧的集合为$T^{(0)}$，所有人工弧上的费用假定为一个充分大的正数M。

一般的网络单纯形算法

Step 0 获得一个初始的基本可行解 **x** 及对应的强可行树结构(T,L,U)。

Step 1 计算对偶变量 **π**。

Step 2 最优解判定。根据互补松弛条件，如果$\forall(i,j)\in L, c_{ij}^{\pi}\geq 0$；并且$\forall(i,j)\in U, c_{ij}^{\pi}\leq 0$，则已经达到最优解。否则选取一个不满足上述条件的弧(p,q)为进基弧。

Step 3 设$T\cup\{(p,q)\}$包含的圈为W。如果$(p,q)\in L$，我们把弧(p,q)的方向定义为W的正向。如果$(p,q)\in U$，我们把弧(p,q)的相反方向定义为W的正向。

一般的网络单纯形算法（续）

假设$W^- \neq \varnothing$，令

$$\delta = \min\left(\min\left\{x_{ij} - l_{ij} \middle| (i,j) \in W^-\right\}, \min\left\{u_{ij} - x_{ij} \middle| (i,j) \in W^+\right\}\right)$$

$$\overline{W} = \left\{(i,j) \in W^- \middle| x_{ij} - l_{ij} = \delta\right\} \cup \left\{(i,j) \in W^+ \middle| u_{ij} - x_{ij} = \delta\right\}$$

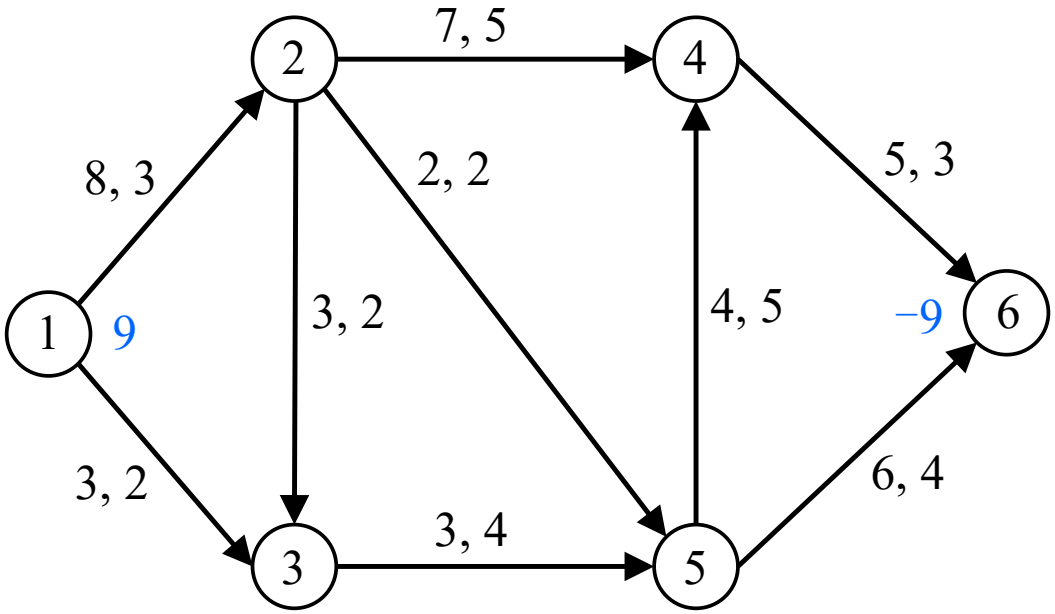
则$\overline{W}$为所有可能的出基弧的集合。

进一步，我们记节点$\overline{p}$是T中从根节点r到节点p的路与圈W的第一个交点，并选取出基弧(k,l)为从$\overline{p}$出发沿圈W的正向前进时第一次所遇到的$\overline{W}$中的弧。

Step 4 沿圈W的正向增广流量δ，即修改**x**及对应的强可行树结构(T,L,U)，回到 Step 1。

容量有界的情形 — 举例

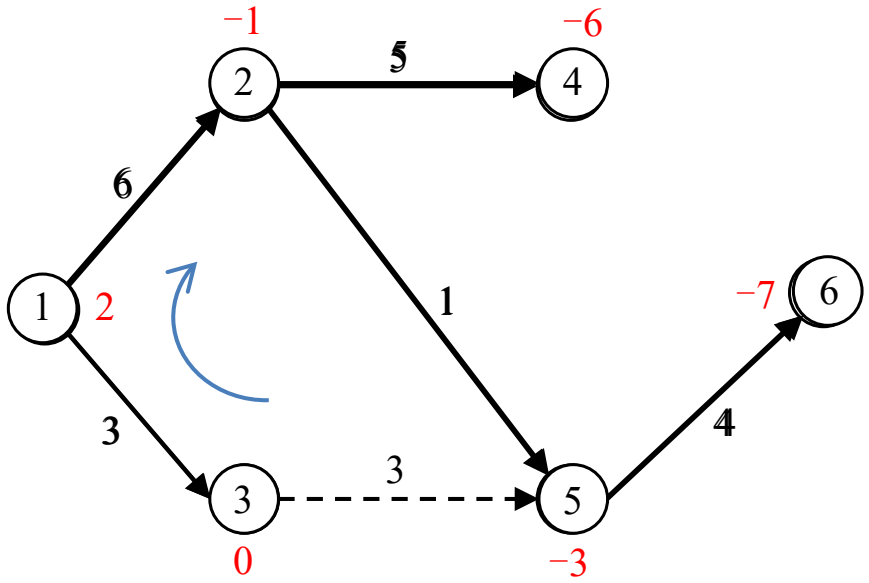
用网络单纯形算法计算如下图所示网络中的最小费用流（假设所有弧的下容量为 0，图中弧上的前一个数字表示弧的上容量，后一个数字表示弧上的单位费用；节点上的数字表示供需量）。



容量有界的情形 — 举例

假设已获得一个初始的可行树结构：

$T=\{(1,2),(1,3),(2,4),(2,5),(5,6)\}$; $L=\{(2,3),(5,4)\}$; $U=\{(3,5),(4,6)\}$



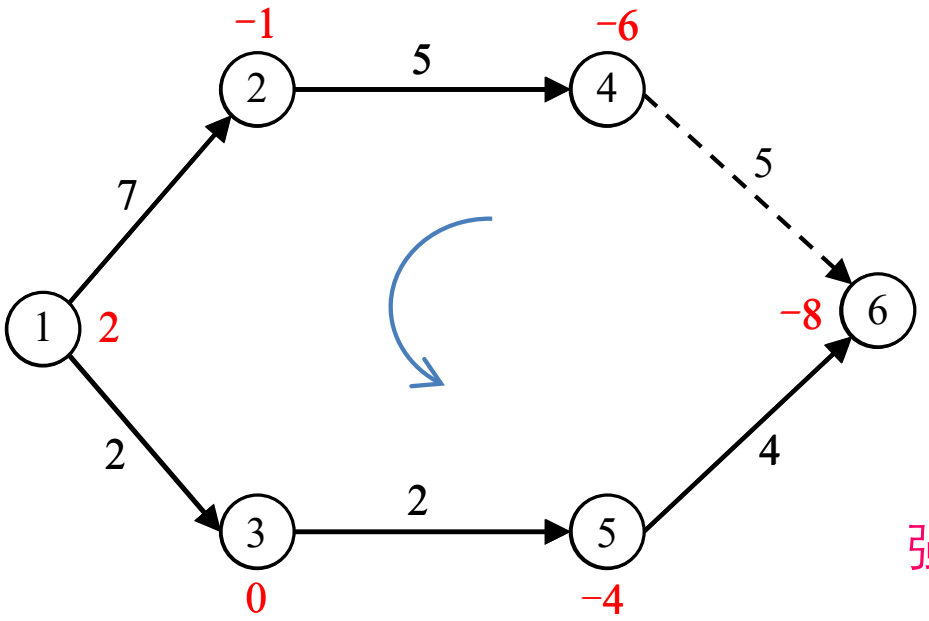
强可行树结构?

树弧中只有$(1,3)$上的流量是等于其上容量的，且$(1,3)$是面向根节点的，因此(T,L,U)是强可行树结构。

容量有界的情形 — 举例

此时，可行树结构为：

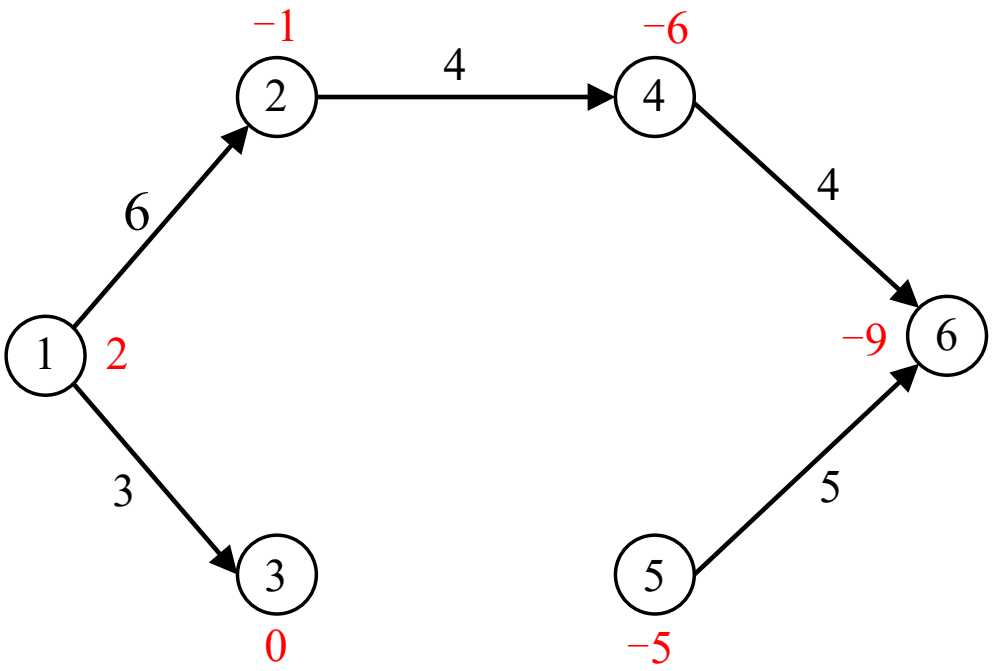
$$T=\{(1,2),(1,3),(2,4),(3,5),(5,6)\};\ L=\{(2,3),(5,4)\};\ U=\{(2,5),(4,6)\}$$



强可行树结构?

沿增广圈可以增广的最大流量为 1，即 $(3,5)$ 出基。

容量有界的情形 — 举例



上面介绍的网络单纯形算法Step3中选取的出基弧可能就是在Step2中选取的进基弧。这时，经过一次旋转，虽然T是不变的，但L, U会发生变化，算法过程中只对强可行树结构进行处理，因此可以避免出现循环。