《Matrix》命题报告

余行江

长沙市长郡中学

2014.4.29

题目大意

- 给出一个$N \times N$的由非负实数组成的矩阵A

题目大意

- 给出一个$N \times N$的由非负实数组成的矩阵A
- 要求构造一个$N \times N$矩阵B，满足

题目大意

- 给出一个$N \times N$的由非负实数组成的矩阵A

- 要求构造一个$N \times N$矩阵B，满足
  - B的第i行的权值和+第j列的权值和$\geq A_{i,j}$
  - B内的元素值非负

题目大意

- 给出一个$N \times N$的由非负实数组成的矩阵A
- 要求构造一个$N \times N$矩阵B，满足
  - B的第i行的权值和+第j列的权值和$\geq A_{i,j}$
  - B内的元素值非负
- 最小化B的权值和

题目大意

- 给出一个$N \times N$的由非负实数组成的矩阵A
- 要求构造一个$N \times N$矩阵B，满足
  - B的第i行的权值和+第j列的权值和$\geq A_{i,j}$
  - B内的元素值非负
- 最小化B的权值和
- 只需要输出权值和，并不要求出详细的B

数据规模

- $N \leq 80$
- A内的所有实数的绝对值不超过1000，且保留3位小数

线性规划模型

- 本题有着很明显的线性规划模型

方法

- 令R_i表示矩阵B的第i行的权值和，C_j表示第j列的权值和

方法

- 令R_i表示矩阵B的第i行的权值和，C_j表示第j列的权值和

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \dots\dots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \dots\dots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

方法

- 令R_i表示矩阵B的第i行的权值和，C_j表示第j列的权值和

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \dots\dots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \dots\dots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

- 变量数 $O(n)$
- 约束数 $O(n^2)$

方法

- 令R_i表示矩阵B的第i行的权值和，C_j表示第j列的权值和

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \cdots\cdots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \cdots\cdots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

- 变量数 $O(n)$
- 约束数 $O(n^2)$

似乎挺不错的样子，不过……

方法

- 令R_i表示矩阵B的第i行的权值和，C_j表示第j列的权值和

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \dots\dots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \dots\dots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

- 变量数 $O(n)$
- 约束数 $O(n^2)$

似乎挺不错的样子，不过……
一定存在一个矩阵B，满足解得的R和C？

方法

- 令R_i表示矩阵B的第i行的权值和，C_j表示第j列的权值和

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \cdots\cdots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \cdots\cdots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

- 变量数 $O(n)$
- 约束数 $O(n^2)$

似乎挺不错的样子，不过……
一定存在一个矩阵B，满足解得的R和C？

Of course~

构造方法

- 任取一个R_i

构造方法

- 任取一个R_i
- 确定$B_{i,1}, B_{i,2}, \ldots, B_{i,n}$

构造方法

- 任取一个R_i

- 确定$B_{i,1}, B_{i,2}, \dots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \le C_j$

构造方法

- 任取一个R_i
- 确定$B_{i,1}, B_{i,2}, \ldots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \le C_j$
- 删去R_i

构造方法

- 任取一个R_i

- 确定$B_{i,1}, B_{i,2}, \dots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \le C_j$

- 删去R_i
  - 剩下的子问题的限制条件不变

构造方法

- 任取一个R_i

- 确定$B_{i,1}, B_{i,2}, \dots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \le C_j$

- 删去R_i
  - 剩下的子问题的限制条件不变

- 递归处理

构造方法

- 任取一个R_i
- 确定$B_{i,1}, B_{i,2}, \ldots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \le C_j$
- 删去R_i
  - 剩下的子问题的限制条件不变
- 递归处理

			1.5
			2.7
			3.0
0.4	5.5	1.3	

构造方法

- 任取一个R_i
- 确定$B_{i,1}, B_{i,2}, \dots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \le C_j$
- 删去R_i
  - 剩下的子问题的限制条件不变
- 递归处理

			1.5
			2.7
			3.0
0.4	5.5	1.3	

构造方法

- 任取一个R_i
- 确定$B_{i,1}, B_{i,2}, \ldots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \le C_j$
- 删去R_i
  - 剩下的子问题的限制条件不变
- 递归处理

0.1			1.4
0.2			2.5
0.1			2.9
0.4	5.5	1.3	

构造方法

- 任取一个R_i
- 确定$B_{i,1}, B_{i,2}, \dots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \le C_j$
- 删去R_i
  - 剩下的子问题的限制条件不变
- 递归处理

0.1			1.4
0.2			2.5
0.1			2.9
0.4	5.5	1.3	

构造方法

- 任取一个R_i
- 确定$B_{i,1}, B_{i,2}, \dots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \le C_j$
- 删去R_i
  - 剩下的子问题的限制条件不变
- 递归处理

0.1	**1.4**		**0**
0.2	2.5		0
0.1	1.6		1.3
0.4	5.5	1.3	

构造方法

- 任取一个R_i
- 确定$B_{i,1}, B_{i,2}, \ldots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \le C_j$
- 删去R_i
  - 剩下的子问题的限制条件不变
- 递归处理

0.1	1.4		0
0.2	2.5		0
0.1	1.6		1.3
0.4	5.5	1.3	

构造方法

- 任取一个R_i
- 确定$B_{i,1}, B_{i,2}, \dots, B_{i,n}$
  - 满足$\sum_{j=1}^{N} B_{i,j} = R_i$
  - $B_{i,j} \leq C_j$
- 删去R_i
  - 剩下的子问题的限制条件不变
- 递归处理

0.1	1.4	0	0
0.2	2.5	0	0
0.1	1.6	1.3	0
0.4	5.5	1.3	

分析

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \dots\dots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \dots\dots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

- 观察式子

分析

$$\min z=\sum_{i=1}^{N}R_i+\sum_{j=1}^{N}C_j$$

$$\begin{cases} R_1+C_1\ge A_{1,1} \\ R_1+C_1\ge A_{1,2} \\ \dots\dots \\ R_2+C_1\ge A_{2,1} \\ R_2+C_2\ge A_{2,2} \\ \dots\dots \\ R_N+C_N\ge A_{N,N} \\ \sum_{i=1}^{N}R_i-\sum_{j=1}^{N}C_j=0 \\ R_i\ge 0, C_i\ge 0 \end{cases}$$

- 观察式子
  - 很像一个二分图最大权匹配的顶标模型

分析

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \dots\dots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \dots\dots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

- 观察式子
  - 很像一个二分图最大权匹配的顶标模型
- 只有2点不同

分析

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \dots\dots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \dots\dots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

- 观察式子
  - 很像一个二分图最大权匹配的顶标模型
- 只有2点不同
  - 要求两边的顶标和相等

分析

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \ge A_{1,1} \\ R_1 + C_1 \ge A_{1,2} \\ \dots\dots \\ R_2 + C_1 \ge A_{2,1} \\ R_2 + C_2 \ge A_{2,2} \\ \dots\dots \\ R_N + C_N \ge A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \ge 0, C_i \ge 0 \end{cases}$$

- 观察式子
  - 很像一个二分图最大权匹配的顶标模型
- 只有2点不同
  - 要求两边的顶标和相等
  - 顶标必须≥ 0

分析

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \dots\dots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \dots\dots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

- 观察式子
  - 很像一个二分图最大权匹配的顶标模型
- 只有2点不同
  - 要求两边的顶标和相等
  - 顶标必须≥ 0
- 似乎难以继续做下去

分析

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \dots\dots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \dots\dots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

- 观察式子
  - 很像一个二分图最大权匹配的顶标模型
- 只有2点不同
  - 要求两边的顶标和相等
  - 顶标必须≥ 0
- 似乎难以继续做下去
  - 对偶原理

对偶原理

原问题：

- $\min C_1X_1 + C_2X_2 + \cdots + C_nX_n$
- 约束条件
  - $A_{i,1}X_1 + A_{i,2}X_2 + \cdots + A_{i,n}X_n \geq B_i$
  - $A_{i,1}X_1 + A_{i,2}X_2 + \cdots + A_{i,n}X_n \leq B_i$
  - $A_{i,1}X_1 + A_{i,2}X_2 + \cdots + A_{i,n}X_n = B_i$
  - $A_{i,1}X_1 + A_{i,2}X_2 + \cdots + A_{i,n}X_n$ 无限制
- 变量
  - $X_i \geq 0$
  - $X_i \leq 0$
  - X_i 无限制
  - $X_i = 0$

对偶问题：

- $\max B_1Y_1 + B_2Y_2 + \cdots + B_mY_m$
- 变量
  - $Y_i \geq 0$
  - $Y_i \leq 0$
  - Y_i 无限制
  - $Y_i = 0$
- 约束条件
  - $A_{1,i}Y_1 + A_{2,i}Y_2 + \cdots + A_{m,i}Y_m \leq C_i$
  - $A_{1,i}Y_1 + A_{2,i}Y_2 + \cdots + A_{m,i}Y_m \leq C_i$
  - $A_{1,i}Y_1 + A_{2,i}Y_2 + \cdots + A_{m,i}Y_m = C_i$
  - $A_{1,i}Y_1 + A_{2,i}Y_2 + \cdots + A_{m,i}Y_m$ 无限制

目标

原问题：

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \dots\dots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \dots\dots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

目标

原问题：

$$\min z = \sum_{i=1}^{N} R_i + \sum_{j=1}^{N} C_j$$

$$\begin{cases} R_1 + C_1 \geq A_{1,1} \\ R_1 + C_1 \geq A_{1,2} \\ \dots\dots \\ R_2 + C_1 \geq A_{2,1} \\ R_2 + C_2 \geq A_{2,2} \\ \dots\dots \\ R_N + C_N \geq A_{N,N} \\ \sum_{i=1}^{N} R_i - \sum_{j=1}^{N} C_j = 0 \\ R_i \geq 0, C_i \geq 0 \end{cases}$$

对偶问题：

$$\max z = 0 \times P + \sum_{i=1}^{N} \sum_{j=1}^{N} A_{i,j} \times Y_{i,j}$$

$$\begin{cases} Y_{1,1} + Y_{1,2} + \cdots + Y_{1,N} + P \leq 1 \\ Y_{2,1} + Y_{2,2} + \cdots + Y_{2,N} + P \leq 1 \\ \dots\dots \\ Y_{1,1} + Y_{2,1} + \cdots + Y_{N,1} - P \leq 1 \\ Y_{1,2} + Y_{2,2} + \cdots + Y_{N,2} - P \leq 1 \\ \dots\dots \\ Y_{N,1} + Y_{N,2} + \cdots + Y_{N,N} + P \leq 1 \\ Y_{1,N} + Y_{2,N} + \cdots + Y_{N,N} - P \leq 1 \\ Y_{i,j} \geq 0 \end{cases}$$

目标

- 二分图最大费用流的模型

目标

- 二分图最大费用流的模型
  - 一张带权二分图，每个点的初始点容量都为1

目标

- 二分图最大费用流的模型
  - 一张带权二分图，每个点的初始点容量都为1
  - 选定一个在$[-1,1]$内的实数P

目标

- 二分图最大费用流的模型
  - 一张带权二分图，每个点的初始点容量都为1
  - 选定一个在$[-1,1]$内的实数P
    - 一边的所有点的容量$+P$
    - 另一边的所有点的容量$-P$

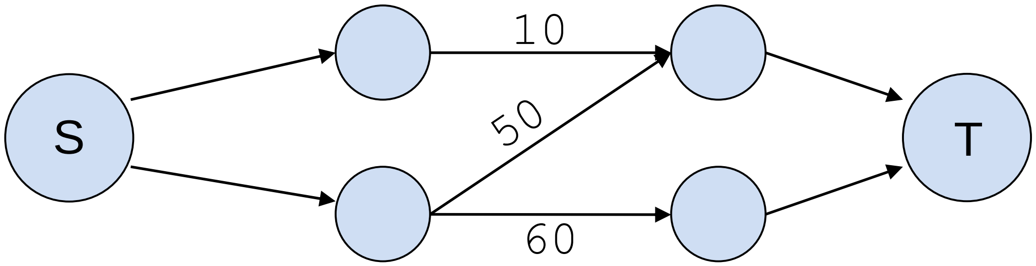
目标

- 二分图最大费用流的模型
  - 一张带权二分图，每个点的初始点容量都为1
  - 选定一个在$[-1,1]$内的实数P
    - 一边的所有点的容量$+P$
    - 另一边的所有点的容量$-P$
  - 求出P，最大化最大费用

目标

- 二分图最大费用流的模型
  - 一张带权二分图，每个点的初始点容量都为1
  - 选定一个在$[-1,1]$内的实数P
    - 一边的所有点的容量$+P$
    - 另一边的所有点的容量$-P$
  - 求出P，最大化最大费用

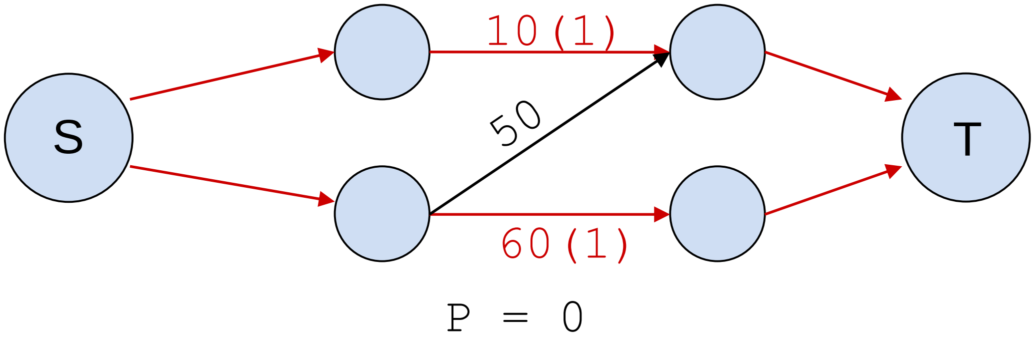
- For example...



目标

- 二分图最大费用流的模型
  - 一张带权二分图，每个点的初始点容量都为1
  - 选定一个在$[-1,1]$内的实数P
    - 一边的所有点的容量$+P$
    - 另一边的所有点的容量$-P$
  - 求出P，最大化最大费用

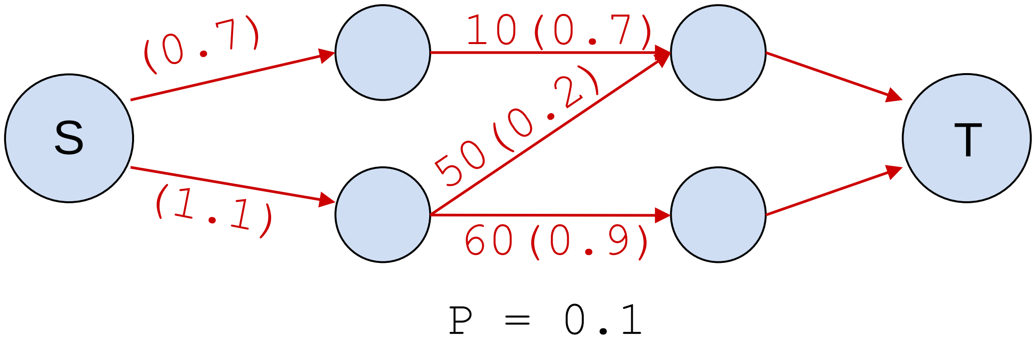
- For example...



目标

- 二分图最大费用流的模型
  - 一张带权二分图，每个点的初始点容量都为1
  - 选定一个在$[-1,1]$内的实数P
    - 一边的所有点的容量$+P$
    - 另一边的所有点的容量$-P$
  - 求出P，最大化最大费用

- For example...



目标

- 凭直觉...

目标

- 凭直觉...
  - 在$P \in [0,1]$与$P \in [-1,0]$内，最大费用是一个单峰函数

目标

- 凭直觉...
  - 在$P \in [0,1]$与$P \in [-1,0]$内，最大费用是一个单峰函数
  - 验证猜想？
    - 显然法...

目标

- 凭直觉...
  - 在$P \in [0,1]$与$P \in [-1,0]$内，最大费用是一个单峰函数
  - 验证猜想？
    - 显然法...
    - 打个表试一试

-1.000	-0.0000
-0.900	11.3472
-0.800	22.6319
-0.700	33.6295
-0.600	44.1110
-0.500	53.7035
-0.400	62.3809
-0.300	69.7991
-0.200	75.6002
-0.100	79.6134
-0.000	81.3465
0.100	79.5772
0.200	75.7720
0.300	69.8252
0.400	62.2517
0.500	53.4275
0.600	43.7136
0.700	33.3613
0.800	22.4175
0.900	11.2202
1.000	0.0000

目标

- 凭直觉...
  - 在$P \in [0,1]$与$P \in [-1,0]$内，最大费用是一个单峰函数
  - 验证猜想？
    - 显然法...
    - 打个表试一试
- 大量的打表实验，发现

-1.000	-0.0000
-0.900	11.3472
-0.800	22.6319
-0.700	33.6295
-0.600	44.1110
-0.500	53.7035
-0.400	62.3809
-0.300	69.7991
-0.200	75.6002
-0.100	79.6134
-0.000	81.3465
0.100	79.5772
0.200	75.7720
0.300	69.8252
0.400	62.2517
0.500	53.4275
0.600	43.7136
0.700	33.3613
0.800	22.4175
0.900	11.2202
1.000	0.0000

目标

- 凭直觉...
  - 在$P \in [0,1]$与$P \in [-1,0]$内，最大费用是一个单峰函数
  - 验证猜想？
    - 显然法...
    - 打个表试一试
- 大量的打表实验，发现
  - 严格的单峰函数

-1.000	-0.0000
-0.900	11.3472
-0.800	22.6319
-0.700	33.6295
-0.600	44.1110
-0.500	53.7035
-0.400	62.3809
-0.300	69.7991
-0.200	75.6002
-0.100	79.6134
-0.000	81.3465
0.100	79.5772
0.200	75.7720
0.300	69.8252
0.400	62.2517
0.500	53.4275
0.600	43.7136
0.700	33.3613
0.800	22.4175
0.900	11.2202
1.000	0.0000

目标

- 凭直觉...
  - 在$P \in [0,1]$与$P \in [-1,0]$内，最大费用是一个单峰函数
  - 验证猜想？
    - 显然法...
    - 打个表试一试

-1.000	-0.0000
-0.900	11.3472
-0.800	22.6319
-0.700	33.6295
-0.600	44.1110
-0.500	53.7035
-0.400	62.3809
-0.300	69.7991
-0.200	75.6002
-0.100	79.6134
-0.000	81.3465
0.100	79.5772
0.200	75.7720
0.300	69.8252
0.400	62.2517
0.500	53.4275
0.600	43.7136
0.700	33.3613
0.800	22.4175
0.900	11.2202
1.000	0.0000

- 大量的打表实验，发现
  - 严格的单峰函数
    - 可以三分

目标

- 凭直觉...
  - 在$P \in [0,1]$与$P \in [-1,0]$内，最大费用是一个单峰函数
  - 验证猜想？
    - 显然法...
    - 打个表试一试

-1.000	-0.0000
-0.900	11.3472
-0.800	22.6319
-0.700	33.6295
-0.600	44.1110
-0.500	53.7035
-0.400	62.3809
-0.300	69.7991
-0.200	75.6002
-0.100	79.6134
-0.000	81.3465
0.100	79.5772
0.200	75.7720
0.300	69.8252
0.400	62.2517
0.500	53.4275
0.600	43.7136
0.700	33.3613
0.800	22.4175
0.900	11.2202
1.000	0.0000

- 大量的打表实验，发现
  - 严格的单峰函数
    - 可以三分
  - 在$[-1,1]$内的单峰函数

目标

- 凭直觉...
  - 在$P \in [0,1]$与$P \in [-1,0]$内，最大费用是一个单峰函数
  - 验证猜想？
    - 显然法...
    - 打个表试一试
- 大量的打表实验，发现
  - 严格的单峰函数
    - 可以三分
  - 在$[-1,1]$内的单峰函数
    - 只用三分一次

-1.000	-0.0000
-0.900	11.3472
-0.800	22.6319
-0.700	33.6295
-0.600	44.1110
-0.500	53.7035
-0.400	62.3809
-0.300	69.7991
-0.200	75.6002
-0.100	79.6134
-0.000	81.3465
0.100	79.5772
0.200	75.7720
0.300	69.8252
0.400	62.2517
0.500	53.4275
0.600	43.7136
0.700	33.3613
0.800	22.4175
0.900	11.2202
1.000	0.0000

证明

- 注意到 $\frac{1+P}{1-P} \in (0, \infty)$

证明

- 注意到 $\frac{1+P}{1-P} \in (0, \infty)$
  - 不妨令$1 - P$为单位1，$1 + P$为K

证明

- 注意到$\frac{1+P}{1-P} \in (0, \infty)$
  - 不妨令$1-P$为单位1，$1+P$为K
  - K为任意正实数

证明

- 注意到 $\frac{1+P}{1-P} \in (0,\infty)$
  - 不妨令$1-P$为单位$1, 1+P$为K
  - K为任意正实数

- 对新图求最大费用流

证明

- 注意到 $\frac{1+P}{1-P} \in (0,\infty)$
  - 不妨令$1-P$为单位$1, 1+P$为K
  - K为任意正实数
- 对新图求最大费用流
  - 一边的点容量均为K，一边的点容量均为1

证明

- 注意到 $\frac{1+P}{1-P} \in (0, \infty)$
  - 不妨令$1-P$为单位$1, 1+P$为K
  - K为任意正实数
- 对新图求最大费用流
  - 一边的点容量均为K，一边的点容量均为1
  - 将新图的最大费用除以$\frac{(k+1)}{2}$，即为原图的答案

证明

- 注意到$\frac{1+P}{1-P} \in (0,\infty)$
  - 不妨令$1-P$为单位$1, 1+P$为K
  - K为任意正实数
- 对新图求最大费用流
  - 一边的点容量均为K，一边的点容量均为1
  - 将新图的最大费用除以$\frac{(k+1)}{2}$，即为原图的答案
- 设新图的最大费用为$MaxCost(k)$

证明

- 注意到$\frac{1+P}{1-P} \in (0, \infty)$
  - 不妨令$1-P$为单位1，$1+P$为K
  - K为任意正实数
- 对新图求最大费用流
  - 一边的点容量均为K，一边的点容量均为1
  - 将新图的最大费用除以$\frac{(k+1)}{2}$，即为原图的答案
- 设新图的最大费用为$MaxCost(k)$
  - 只需要证明$2 \times \frac{MaxCost(k)}{k+1}$为单峰函数

证明

- 二个显然的结论

证明

- 二个显然的结论
  - 当$K > N$时，$MaxCost(k) = MaxCost(n)$

证明

- 二个显然的结论
  - 当$K > N$时，$MaxCost(k) = MaxCost(n)$
  - $MaxCost''(k) < 0$

证明

- 二个显然的结论
  - 当$K > N$时，$MaxCost(k) = MaxCost(n)$
  - $MaxCost''(k) < 0$
    - $MaxCost(k)$在$k \in [0, N]$内为一个凸函数

证明

- 二个显然的结论
  - 当$K > N$时，$MaxCost(k) = MaxCost(n)$
  - $MaxCost''(k) < 0$
    - $MaxCost(k)$在$k \in [0, N]$内为一个凸函数
- 令 $f(k) = MaxCost(k), g(k) = \frac{f(k)}{k+1}$

证明

- 二个显然的结论
  - 当$K > N$时，$MaxCost(k) = MaxCost(n)$
  - $MaxCost''(k) < 0$
    - $MaxCost(k)$在$k \in [0, N]$内为一个凸函数

- 令 $f(k) = MaxCost(k), g(k) = \frac{f(k)}{k+1}$

- 对$g(k)$求导

证明

- 二个显然的结论
  - 当$K > N$时，$MaxCost(k) = MaxCost(n)$
  - $MaxCost''(k) < 0$
    - $MaxCost(k)$在$k \in [0, N]$内为一个凸函数

- 令 $f(k) = MaxCost(k), g(k) = \frac{f(k)}{k+1}$

- 对$g(k)$求导
  - $g'(k) = \frac{f'(k)}{k+1} - \frac{f(k)}{(k+1)^2}$

证明

- 二个显然的结论
  - 当$K > N$时，$MaxCost(k) = MaxCost(n)$
  - $MaxCost''(k) < 0$
    - $MaxCost(k)$在$k \in [0, N]$内为一个凸函数
- 令 $f(k) = MaxCost(k), g(k) = \frac{f(k)}{k+1}$
- 对$g(k)$求导
  - $g'(k) = \frac{f'(k)}{k+1} - \frac{f(k)}{(k+1)^2}$
  - 只需要证明至多存在一个k，使得$g'(k) = 0$

证明

- 转化式子

证明

- 转化式子
  - $f'(k) \times (k+1) = f(k)$

证明

- 转化式子
  - $f'(k) \times (k+1) \ = \ f(k)$
  - $\frac{f(k)}{f'(k)} - (k+1) = 0$

证明

- 转化式子
  - $f'(k) \times (k+1) \ = \ f(k)$
  - $\frac{f(k)}{f'(k)} - (k+1) = 0$
- 只需要证明该等式至多只有一个解即可

证明

- 转化式子
  - $f'(k) \times (k+1) = f(k)$
  - $\frac{f(k)}{f'(k)} - (k+1) = 0$
- 只需要证明该等式至多只有一个解即可
  - 左侧函数导数恒大于0或恒小于0

证明

- 转化式子
  - $f'(k) \times (k+1) \ = \ f(k)$
  - $\frac{f(k)}{f'(k)} - (k+1) = 0$
- 只需要证明该等式至多只有一个解即可
  - 左侧函数导数恒大于0或恒小于0
- 再次求导

证明

- 转化式子
  - $f'(k) \times (k+1) \ = \ f(k)$
  - $\frac{f(k)}{f'(k)} - (k+1) = 0$
- 只需要证明该等式至多只有一个解即可
  - 左侧函数导数恒大于0或恒小于0
- 再次求导
  - $\left(\frac{f(k)}{f'(k)} - (k+1)\right)'$

证明

- 转化式子
  - $f'(k)\times(k+1)\ =\ f(k)$
  - $\frac{f(k)}{f'(k)}-(k+1)=0$
- 只需要证明该等式至多只有一个解即可
  - 左侧函数导数恒大于0或恒小于0
- 再次求导
  - $\left(\frac{f(k)}{f'(k)}-(k+1)\right)'$

$$=\frac{[f'(k)\times f'(k)-f(k)\times f''(k)]}{[f'(k)]^2}-1$$

证明

- 转化式子
  - $f'(k) \times (k+1) \ = \ f(k)$
  - $\frac{f(k)}{f'(k)} - (k+1) = 0$
- 只需要证明该等式至多只有一个解即可
  - 左侧函数导数恒大于0或恒小于0
- 再次求导
  - $\left(\frac{f(k)}{f'(k)} - (k+1)\right)'$

$$= \frac{[f'(k) \times f'(k) - f(k) \times f''(k)]}{[f'(k)]^2} - 1$$

$$= \frac{-f(k) \times f''(k)}{[f'(k)]^2}$$

证明

- 转化式子
  - $f'(k) \times (k+1) = f(k)$
  - $\frac{f(k)}{f'(k)} - (k+1) = 0$
- 只需要证明该等式至多只有一个解即可
  - 左侧函数导数恒大于0或恒小于0
- 再次求导
  - $\left(\frac{f(k)}{f'(k)} - (k+1)\right)'$

$= \frac{[f'(k)\times f'(k) - f(k)\times f''(k)]}{[f'(k)]^2} - 1$

$= \frac{-f(k)\times f''(k)}{[f'(k)]^2}$

- 明显，在$k \in [0,n]$内
  - $f(k) > 0, f''(k) < 0, [f'(k)]^2 > 0$

证明

- 转化式子
  - $f'(k) \times (k+1) \;=\; f(k)$
  - $\frac{f(k)}{f'(k)} - (k+1) = 0$
- 只需要证明该等式至多只有一个解即可
  - 左侧函数导数恒大于0或恒小于0
- 再次求导
  - $\left(\frac{f(k)}{f'(k)} - (k+1)\right)'$

$$= \frac{[f'(k) \times f'(k) - f(k) \times f''(k)]}{[f'(k)]^2} - 1$$

$$= \frac{-f(k) \times f''(k)}{[f'(k)]^2}$$

- 明显，在$k \in [0,n]$内
  - $f(k) > 0, f''(k) < 0, [f'(k)]^2 > 0$
- 故$\frac{-f(k) \times f''(k)}{[f'(k)]^2}$恒大于0

算法

- 本题的模型是一张完全二分图

算法

- 本题的模型是一张完全二分图
  - $SPFA$效率太低

算法

- 本题的模型是一张完全二分图
  - $SPFA$效率太低

- 用张琨玮改进的费用流算法或原始对偶算法

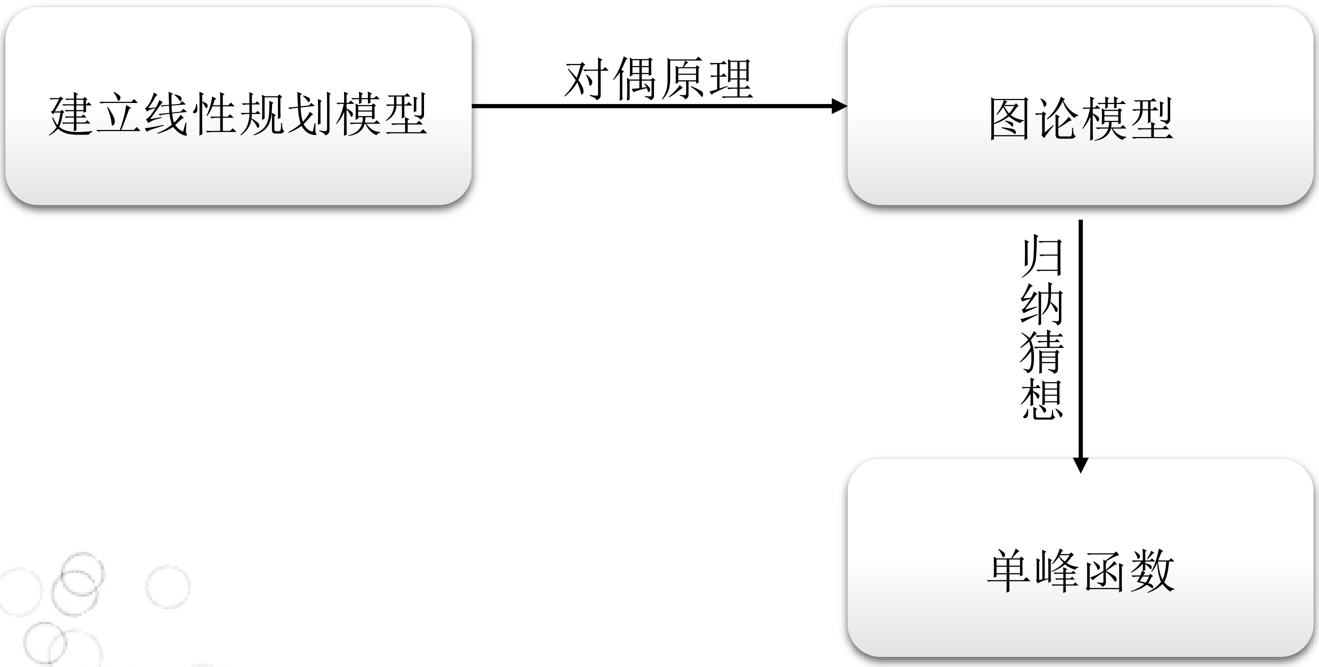
总结

建立线性规划模型

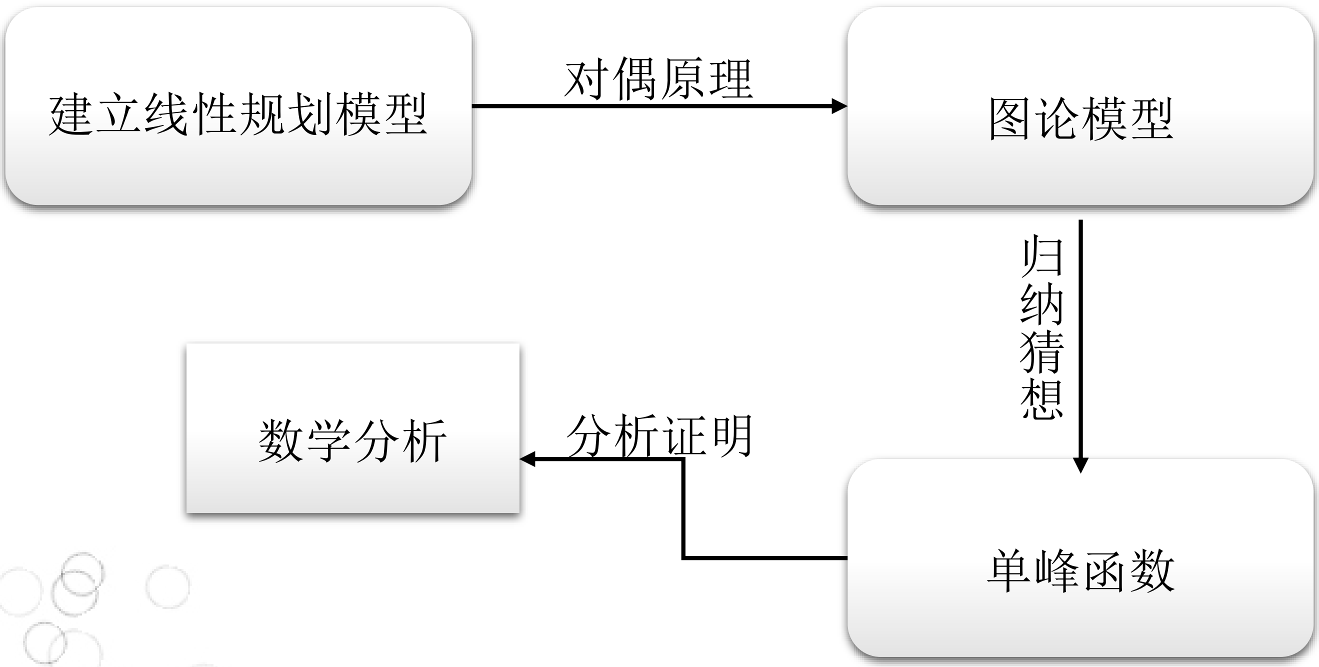
总结

建立线性规划模型 $\xrightarrow{\text{对偶原理}}$ 图论模型

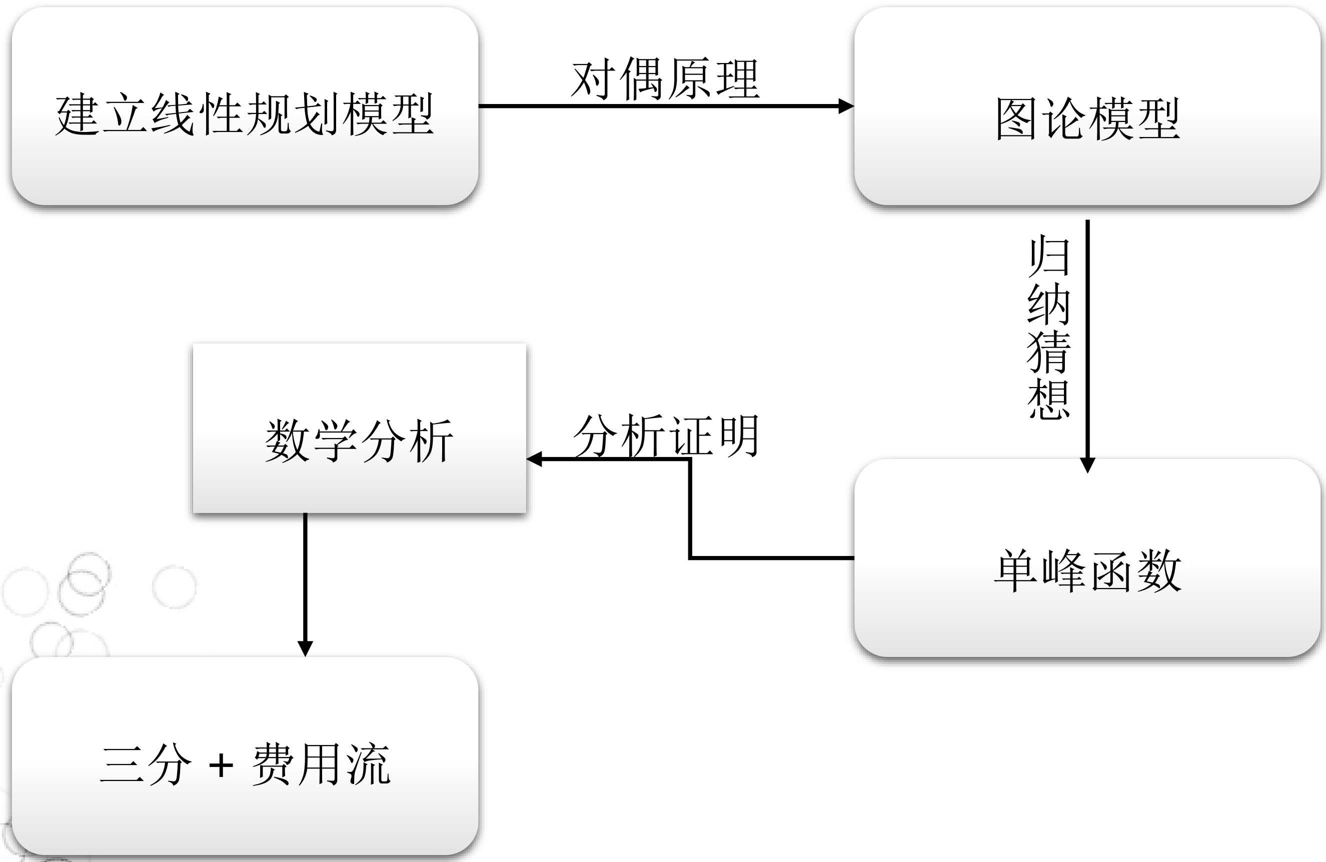
总结



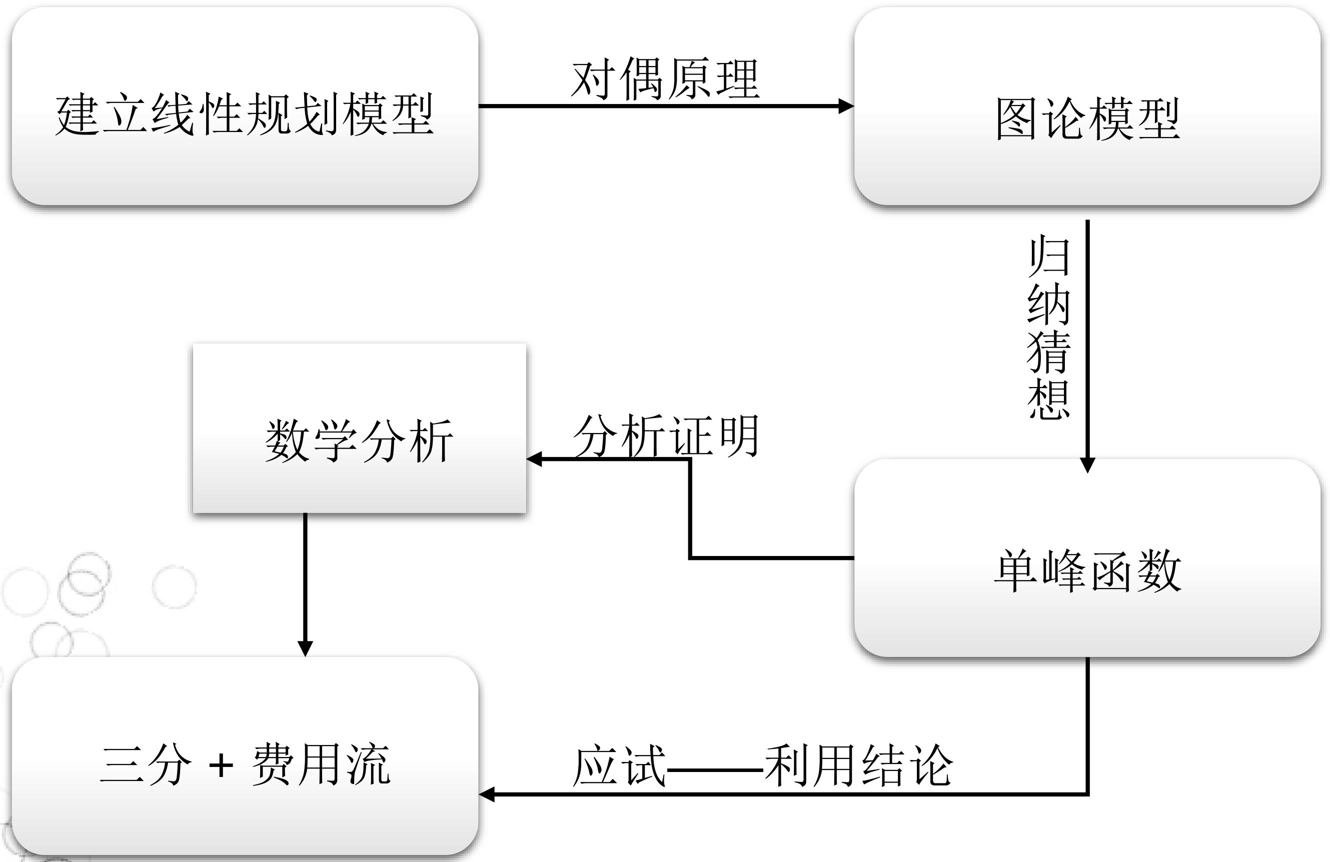
总结



总结



总结



考察

- 线性规划和网络流建模的基础知识

考察

- 线性规划和网络流建模的基础知识
- 对偶原理

考察

- 线性规划和网络流建模的基础知识
- 对偶原理
- 一些对网络流的优化

考察

- 线性规划和网络流建模的基础知识
- 对偶原理
- 一些对网络流的优化
- 简单的数学分析

感谢

- 感谢*CCF*提供学习交流的平台
- 感谢向期中教练的指导
- 感谢帮助我的同学们
- 感谢您认真的聆听

谢谢

单纯形

- 目标1的$n \leq 300$

单纯形

- 目标1的 $n \leq 300$
  - 肯定跑不过...

单纯形

- 目标1的 $n \leq 300$
  - 肯定跑不过...

- 目标2？

单纯形

- 目标1的 $n \leq 300$
  - 肯定跑不过...

- 目标2?

- 构造数据可以令单纯形 $10s+$

单纯形

- 目标1的 $n \le 300$
  - 肯定跑不过...

- 目标2？

- 构造数据可以令单纯形 $10s+$
  - 每行的元素值相同

单纯形

- 目标1的 $n \leq 300$
  - 肯定跑不过...

- 目标2?

- 构造数据可以令单纯形 $10s+$
  - 每行的元素值相同
  - 链表优化也需要 $4s+$

单纯形

- 目标1的 $n \le 300$
  - 肯定跑不过...
- 目标2？
- 构造数据可以令单纯形 $10s+$
  - 每行的元素值相同
  - 链表优化也需要 $4s+$
  - 再加优化！

单纯形

- 目标1的 $n \le 300$
  - 肯定跑不过...

- 目标2?

- 构造数据可以令单纯形 $10s+$
  - 每行的元素值相同
  - 链表优化也需要 $4s+$
  - 再加优化!
  - 奇葩优化我就没测试了...

Task0

- A内的所有元素值相等
  - 存在一个最优方案，使得B内的所有元素值也相等
  - 人工出解

Task0

- A内的所有元素值相等
  - 存在一个最优方案，使得B内的所有元素值也相等
  - 人工出解
- Just for fun! :)

方法1

- 令$B_{i,j}$表示矩阵B第i行第j列的元素值

方法1

- 令$B_{i,j}$表示矩阵B第i行第j列的元素值

$$\min z = \sum_{i=1}^{N}\sum_{j=1}^{N} B_{i,j}$$

$$\begin{cases}
\sum_{i=1}^{N} B_{i,1} + \sum_{j=1}^{N} B_{1,j} \geq A_{1,1} \\
\sum_{i=1}^{N} B_{i,2} + \sum_{j=1}^{N} B_{1,j} \geq A_{1,2} \\
\cdots\cdots \\
\sum_{i=1}^{N} B_{i,1} + \sum_{j=1}^{N} B_{2,j} \geq A_{2,1} \\
\sum_{i=1}^{N} B_{i,2} + \sum_{j=1}^{N} B_{2,j} \geq A_{2,2} \\
\cdots\cdots \\
\sum_{i=1}^{N} B_{i,N} + \sum_{j=1}^{N} B_{N,j} \geq A_{N,N}
\end{cases}$$

方法1

- 令$B_{i,j}$表示矩阵B第i行第j列的元素值

$$\min z = \sum_{i=1}^{N}\sum_{j=1}^{N} B_{i,j}$$

$$\begin{cases} \sum_{i=1}^{N} B_{i,1} + \sum_{j=1}^{N} B_{1,j} \geq A_{1,1} \\ \sum_{i=1}^{N} B_{i,2} + \sum_{j=1}^{N} B_{1,j} \geq A_{1,2} \\ \cdots\cdots \\ \sum_{i=1}^{N} B_{i,1} + \sum_{j=1}^{N} B_{2,j} \geq A_{2,1} \\ \sum_{i=1}^{N} B_{i,2} + \sum_{j=1}^{N} B_{2,j} \geq A_{2,2} \\ \cdots\cdots \\ \sum_{i=1}^{N} B_{i,N} + \sum_{j=1}^{N} B_{N,j} \geq A_{N,N} \end{cases}$$

- 变量数 $O(n^2)$
- 约束数 $O(n^2)$

方法1

- 令$B_{i,j}$表示矩阵B第i行第j列的元素值

$$\min z = \sum_{i=1}^{N}\sum_{j=1}^{N} B_{i,j}$$

$$\begin{cases} \sum_{i=1}^{N} B_{i,1} + \sum_{j=1}^{N} B_{1,j} \geq A_{1,1} \\ \sum_{i=1}^{N} B_{i,2} + \sum_{j=1}^{N} B_{1,j} \geq A_{1,2} \\ \cdots\cdots \\ \sum_{i=1}^{N} B_{i,1} + \sum_{j=1}^{N} B_{2,j} \geq A_{2,1} \\ \sum_{i=1}^{N} B_{i,2} + \sum_{j=1}^{N} B_{2,j} \geq A_{2,2} \\ \cdots\cdots \\ \sum_{i=1}^{N} B_{i,N} + \sum_{j=1}^{N} B_{N,j} \geq A_{N,N} \end{cases}$$

- 变量数 $O(n^2)$
- 约束数 $O(n^2)$

Too ugly...

Task1

- $N \leq 50$

Task1

- $N \leq 50$

- 线性规划求解

Task1

- $N \leq 50$
- 线性规划求解
  - 单纯形

Task1

- $N \leq 50$
- 线性规划求解
  - 单纯形
- 对于目标1，将每个R_i用$Ra_i - Rb_i$ 表示。

Task1

- $N \leq 50$

- 线性规划求解
  - 单纯形

- 对于目标1，将每个R_i用$Ra_i - Rb_i$ 表示。
  - $Ra_i, Rb_i \geq 0$

Task1

- $N \leq 50$

- 线性规划求解
  - 单纯形

- 对于目标1，将每个R_i用$Ra_i - Rb_i$ 表示。
  - $Ra_i, Rb_i \geq 0$
  - C同理

Task1

- $N \leq 50$
- 线性规划求解
  - 单纯形
- 对于目标1，将每个R_i用$Ra_i - Rb_i$ 表示。
  - $Ra_i, Rb_i \geq 0$
  - C同理
- 随机数据下，效率很不错

Task1

- $N \leq 50$
- 线性规划求解
  - 单纯形
- 对于目标1，将每个R_i用$Ra_i - Rb_i$ 表示。
  - $Ra_i, Rb_i \geq 0$
  - C同理
- 随机数据下，效率很不错
- 优化

Task1

- $N \leq 50$
- 线性规划求解
  - 单纯形
- 对于目标1，将每个R_i用$Ra_i - Rb_i$ 表示。
  - $Ra_i, Rb_i \geq 0$
  - C同理
- 随机数据下，效率很不错
- 优化
  - 变量数除以 2

Task1

- $N \leq 50$
- 线性规划求解
  - 单纯形
- 对于目标1，将每个R_i用$Ra_i - Rb_i$ 表示。
  - $Ra_i, Rb_i \geq 0$
  - C同理
- 随机数据下，效率很不错
- 优化
  - 变量数除以 2
  - 链表优化

- 原线性规划模型似乎不好继续做下去

目标1

- B内的元素值没有非负限制

目标1

- B内的元素值没有非负限制

- 先把式子写出来

目标1

$$\max z = 0 \times P + \sum_{i=1}^{N} \sum_{j=1}^{N} A_{i,j} \times Y_{i,j}$$

$$\begin{cases} Y_{1,1} + Y_{1,2} + \cdots + Y_{1,N} + P = 1 \\ Y_{2,1} + Y_{2,2} + \cdots + Y_{2,N} + P = 1 \\ \cdots\cdots \\ Y_{1,1} + Y_{2,1} + \cdots + Y_{N,1} - P = 1 \\ Y_{1,2} + Y_{2,2} + \cdots + Y_{N,2} - P = 1 \\ \cdots\cdots \\ Y_{N,1} + Y_{N,2} + \cdots + Y_{N,N} + P = 1 \\ Y_{1,N} + Y_{2,N} + \cdots + Y_{N,N} - P = 1 \\ Y_{i,j} \geq 0 \end{cases}$$

目标1

- 感觉很熟悉的东西

目标1

- 感觉很熟悉的东西
  - 去掉$p \rightarrow$ 二分图最大权匹配

目标1

- 感觉很熟悉的东西
  - 去掉$p \to$ 二分图最大权匹配
  - 所有的约束都是取等号

目标1

- 感觉很熟悉的东西
  - 去掉$p \rightarrow$ 二分图最大权匹配
  - 所有的约束都是取等号
    - 从S连出的流量和=向T连入的流量和

目标1

- 感觉很熟悉的东西
  - 去掉$p \rightarrow$ 二分图最大权匹配
  - 所有的约束都是取等号
    - 从S连出的流量和=向T连入的流量和
    - $P = 0$

目标1

- 感觉很熟悉的东西
  - 去掉$p \to$ 二分图最大权匹配
  - 所有的约束都是取等号
    - 从S连出的流量和=向T连入的流量和
    - $P = 0$
- 简单的费用流。

目标2

- 不妨再用目标1的方法对偶

Task2

- 数据随机

Task2

- 数据随机
- 对于第二问，$P = 0$就是最优解

Task2

- 数据随机
- 对于第二问, $P = 0$就是最优解
  - 10000 组数据，只有 50 组的 $|P| \in [10^{-8}\ 10^{-4}]$

目标2 - 2nd

- B的权值和

目标2 - 2nd

- B的权值和
  - 可以二分

目标2 - 2nd

- B的权值和
  - 可以二分
  - 判定答案$= K$时，是否可行

目标2 - 2nd

- B的权值和
  - 可以二分
  - 判定答案$= K$时，是否可行
- 对于原限制
  - $\sum_{i=1}^{N} R_i = \sum_{i=1}^{N} C_i$

目标2 - 2nd

- B的权值和
  - 可以二分
  - 判定答案$=K$时，是否可行
- 对于原限制
  - $\sum_{i=1}^{N} R_i = \sum_{i=1}^{N} C_i$
- 变为
  - $\sum_{i=1}^{N} R_i = K$
  - $\sum_{i=1}^{N} C_i = K$

目标2 - 2nd

- B的权值和
  - 可以二分
  - 判定答案$=K$时，是否可行
- 对于原限制
  - $\sum_{i=1}^{N} R_i = \sum_{i=1}^{N} C_i$
- 变为
  - $\sum_{i=1}^{N} R_i = K$
  - $\sum_{i=1}^{N} C_i = K$
- 再次对偶

目标2 - 2nd

$$\max z = K \times (A + B) + \sum_{i=1}^{N} \sum_{j=1}^{N} A_{i,j} \times Y_{i,j}$$

$$\begin{cases} Y_{1,1} + Y_{1,2} + \cdots + Y_{1,N} + A \leq 1 \\ Y_{2,1} + Y_{2,2} + \cdots + Y_{2,N} + A \leq 1 \\ \dots\dots \\ Y_{1,1} + Y_{2,1} + \cdots + Y_{N,1} + B \leq 1 \\ Y_{1,2} + Y_{2,2} + \cdots + Y_{N,2} + B \leq 1 \\ \dots\dots \\ Y_{N,1} + Y_{N,2} + \cdots + Y_{N,N} + A \leq 1 \\ Y_{1,N} + Y_{2,N} + \cdots + Y_{N,N} + B \leq 1 \\ Y_{i,j} \geq 0 \end{cases}$$

目标2 - 2nd

- 还是二分图

目标2 - 2nd

- 还是二分图
  - 可以用K的代价，给某一边的所有点的容量+1

目标2 - 2nd

- 还是二分图
  - 可以用K的代价，给某一边的所有点的容量+1
- 根据对偶原理的性质

目标2 - 2nd

- 还是二分图
  - 可以用K的代价，给某一边的所有点的容量+1
- 根据对偶原理的性质
  - 只要判定该线性规划是否有极大解即可

目标2 - 2nd

- 还是二分图
  - 可以用K的代价，给某一边的所有点的容量+1
- 根据对偶原理的性质
  - 只要判定该线性规划是否有极大解即可
  - 也就是最大收益趋近∞

目标2 - 2nd

- 明显有$A \to -\infty$并且$B \to -\infty$

目标2 - 2nd

- 明显有$A \to -\infty$并且$B \to -\infty$

- 当 A、B 足够小时

目标2 - 2nd

- 明显有$A \to -\infty$并且$B \to -\infty$

- 当 A、B 足够小时
  - A每减少ΔA, B 同时减少的 ΔB

目标2 - 2nd

- 明显有$A \to -\infty$并且$B \to -\infty$

- 当 A、B 足够小时
  - A每减少ΔA, B 同时减少的 ΔB
  - $\frac{\Delta A}{\Delta B}$趋近于一个定值

目标2 - 2nd

- 明显有$A \to -\infty$并且$B \to -\infty$
- 当 A、B 足够小时
  - A每减少ΔA, B 同时减少的 ΔB
  - $\frac{\Delta A}{\Delta B}$趋近于一个定值
  - $\frac{A}{B} = \frac{\Delta A}{\Delta B}$

目标2 - 2nd

- 明显有$A \to -\infty$并且$B \to -\infty$
- 当 A、B 足够小时
  - A每减少$\Delta A, B$ 同时减少的 ΔB
  - $\frac{\Delta A}{\Delta B}$趋近于一个定值
  - $\frac{A}{B} = \frac{\Delta A}{\Delta B}$
- 设$\Delta A \geq \Delta B$

目标2 - 2nd

- 明显有$A \to -\infty$并且$B \to -\infty$

- 当 A、B 足够小时
  - A每减少ΔA, B 同时减少的 ΔB
  - $\frac{\Delta A}{\Delta B}$趋近于一个定值
  - $\frac{A}{B} = \frac{\Delta A}{\Delta B}$

- 设$\Delta A \geq \Delta B$
  - $\Delta A = 1, \Delta B \in [0, 1]$

目标2 - 2nd

- 明显有$A \to -\infty$并且$B \to -\infty$
- 当 A、B 足够小时
  - A每减少ΔA, B 同时减少的 ΔB
  - $\frac{\Delta A}{\Delta B}$趋近于一个定值
  - $\frac{A}{B} = \frac{\Delta A}{\Delta B}$
- 设$\Delta A \geq \Delta B$
  - $\Delta A = 1, \Delta B \in [0,1]$
  - 三分ΔB
    - A, B 足够小，可以看做原二分图对ΔA 与 ΔB 无影响
    - A 边的所有点容量 $= \Delta A$, B 边的所有点容量 $= \Delta B$
    - 收益 $= MaxCost - (\Delta A + \Delta B) \times K$

目标2 - 2nd

- 明显有$A \to -\infty$并且$B \to -\infty$

- 当 A、B 足够小时
  - A每减少ΔA, B 同时减少的 ΔB
  - $\frac{\Delta A}{\Delta B}$趋近于一个定值
  - $\frac{A}{B} = \frac{\Delta A}{\Delta B}$

- 设$\Delta A \geq \Delta B$
  - $\Delta A = 1, \Delta B \in [0, 1]$
  - 三分ΔB
    - A, B 足够小，可以看做原二分图对ΔA 与 ΔB 无影响
    - A 边的所有点容量 $= \Delta A$, B 边的所有点容量 $= \Delta B$
    - 收益 $= MaxCost - (\Delta A + \Delta B) \times K$
  - 对于 $\Delta A \leq \Delta B$ 类似处理

目标2 - 2nd

- 相对于之前的三分，多了二分操作

目标2 - 2nd

- 相对于之前的三分，多了二分操作
- 勉强跑过了 $n = 50$ 的数据

目标2 - 2nd

- 相对于之前的三分，多了二分操作

- 勉强跑过了 $n = 50$ 的数据
  - T_T
  - 思想巧≠效率高

目标2

- 在回来看看之前的方法...