第六章 最优化理论概要

第六章 最优化理论概要

简要历史

- 早期
  - Fermat, Lagrange：明确最优概念。
  - Newton, Gauss：最优找寻计算法。
- 1940s
  - George Dantzig上课迟到，将老师给出的二个统计学未解问题，当作家庭作业，形成simplex算法。
  - 1947, John von Neumann发展出duality理论。

单纯形

George Dantzig
Nov.8, 1914- May 13, 2005

对偶理论

John von Neumann
Dec. 28, 1903- Feb.8, 1957

数学规划（最优化）的目标

- The objective of mathematical programming is to select the best or optimal solution from the set of solutions that satisfy all of the restriction on resources, called feasible solutions.
- 二个关键语义
  - 从一组可行解中选择最好或最优的解;
  - 可行解是，满足所有资源约束的解。

[Ref]http://ceit.aut.ac.ir/~shiry/lecture

类型，依据决策变量及关系

- Linear Programming (LP)
  - Makes 4 assumptions: Linearity, Divisibility(实数), Certainty(有确定参数), and non negativity.
- Integer Programming (IP)
  - Assumes that the decision parameters must take on integer values.
- Non linear programming (NLP)
  - Assumes that the relationship in the objective function and/or constraint may be nonlinear.

建模与分析的一般过程

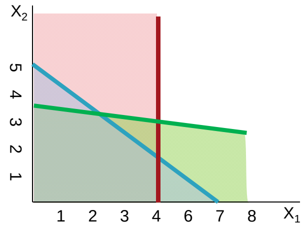
- Formulation（表述）
  - Defining the decision variables, objective function and constraints.
- Solution（求解）
  - Requires determining the optimal values of the decision variables and the objective function. For example by computer software.
- Interpretation（解释）
  - To interpret the results.

示例问题

- Assume that a firm produces alphas and omegas using labor, machine time, and finishing time.
- Profit for each alpha is 2.1USD and for each omega is 3.5USD.
  - Each alpha requires 10 labor hour and 2 hours of machine and 3 hours of finishing
  - Each Omega requires 14 labor hour and 2 hours of machine and no finishing.
  - They have 70 hours of labor, 70 hours of machine and 12 hours of finishing time each day.
- Determine how many alphas and omegas they should produce to maximize the daily profit?

线性规划问题的变形/形式化

- Model:
  - $MAX= 2.1 \cdot X_1 + 3.5 \cdot X_2$
  - $10 \cdot X_1 + 14 \cdot X_2 <= 70$
  - $2 \cdot X_1 + 20 \cdot X_2 <= 70$
  - $3 \cdot X_1 <= 12$
  - $X_1 >= 0$
  - $X_2 >= 0$

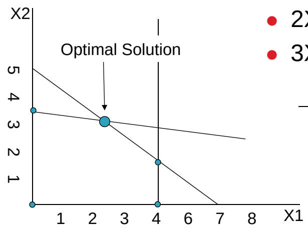


第六章 最优化理论概要

- 6.1 最优化问题类型
- 6.2 线性规划计算方法
- 6.3 选址问题
- 6.4 数独问题
- 6.5 网络单纯形法

图形法

- Model:
  - $MAX= 2.1 x_1 + 3.5X_2$
  - $10X_1+14X_2 <=70$
  - $2X_1+20X_2 <= 70$
  - $3X_1<=12$



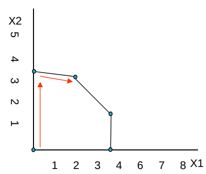
(X1,X2)	Obj. Func
(0,0)	0
(0,3.5)	12.25
(4,0)	8.4
(4, 2.14)	15.59
(2.44,3.26)	16.52

(多边形)角的坐标

- Corner points are where 2 or more lines intersect.
- The fundamental theorem of linear programming is that an optimal solution always lies at a corner point of the feasible region.
- There are 5 corner points in the problem.

单纯形 (Simplex)法

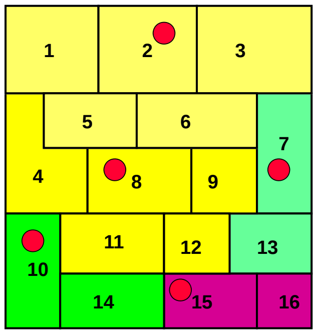
- 查找角点直到目标函数值不能增长为止。
  - 沿X2轴，到达 (0,3.5)；
  - 再沿X1，增大目标函数值，(2.44,3.26)；
  - 再沿X1，目标函数值减少，所以停止；
  - 算法只涉及3个点。
- 变量数较多时，有利于缩小查找空间。



- Model:
  - MAX= 2.1 x1+ 3.5X2
  - 10X1+14X2 <=70
  - 2X1+20X2 <= 70
  - 3X1<=12

第六章 最优化理论概要

选址覆盖的目标



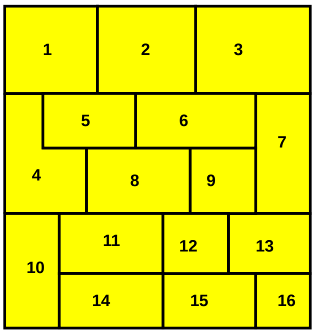
规则

16个区域，或直接，或间隔一个区域，取到灭火器

目标

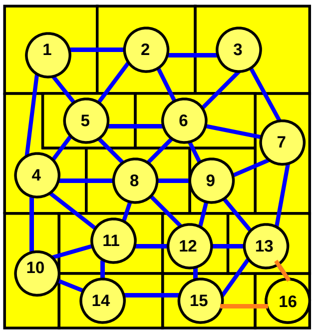
所部署的灭火器，数量最少

形式化/模型



Set	Covers
1	1, 2, 4, 5
2	1, 2, 3, 5, 6
3	2, 3, 6, 7
⋮	⋮
16	13, 15, 16

图覆盖问题



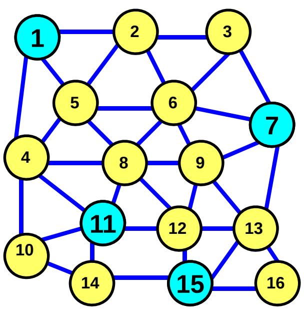
节点表示区域

节点相邻表示区域相邻

节点16覆盖节点13、15和16

求节点集的最小子集，覆盖所有节点

整数规划



$x_j = 1$ if node j is selected
$x_j = 0$ otherwise

Minimize $x_1 + x_2 + \dots + x_{16}$

s.t. $x_1 + x_2 + x_4 + x_5 \geq 1$

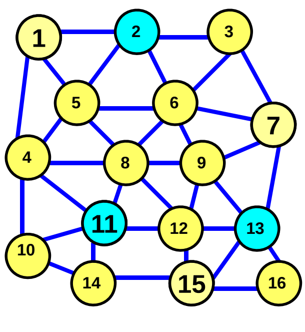
$x_1 + x_2 + x_3 + x_5 + x_6 \geq 1$

⋮ ⋮

$x_{13} + x_{15} + x_{16} \geq 1$

$x_j \in \{0, 1\}$ for each j.

求解结果



$x_j = 1$ if node j is selected
$x_j = 0$ otherwise

Minimize $x_1 + x_2 + \dots + x_{16}$

s.t. $x_1 + x_2 + x_4 + x_5 \geq 1$

$x_1 + x_2 + x_3 + x_5 + x_6 \geq 1$

⋮ ⋮

$x_{13} + x_{15} + x_{16} \geq 1$

$x_j \in \{0, 1\}$ for each j.

第六章 最优化理论概要

Sudoku(数独)

```
+-------+-------+-------+
| 5 3 . | . 7 . | . . . |
| 6 . . | 1 9 5 | . . . |
| . 9 8 | . . . | . 6 . |
+-------+-------+-------+
| 8 . . | . 6 . | . . 3 |
| 4 . . | 8 . 3 | . . 1 |
| 7 . . | . 2 . | . . 6 |
+-------+-------+-------+
| . 6 . | . . . | 2 8 . |
| . . . | 4 1 9 | . . 5 |
| . . . | . 8 . | . 7 9 |
+-------+-------+-------+
```

行：
0~9，各占一格

区：
0~9，各占一格

列：
0~9，各占一格

直接排除法

```
+-------+-------+-------+
| 5 3 . | . 7 . | . . . |
| 6 . . | 1 9 5 | . . . |
| . 9 8 | . . . | . 6 . |
+-------+-------+-------+
| 8 . . | . 6 . | . . 3 |
| 4 . . | 8 . 3 | . . 1 |
| 7 . . | . 2 . | . . 6 |
+-------+-------+-------+
| . 6 . | . . . | 2 8 . |
| . . . | 4 1 9 | . . 5 |
| . . . | . 8 . | . 7 9 |
+-------+-------+-------+
```

区：
仅有一格可填5

简接排除法

```
+-------+-------+-------+
| 5 3 . | . 7 . | . . . |
| 6 . . | 1 9 5 | . . . |
| . 9 8 | . . . | 5 6 . |
+-------+-------+-------+
| 8 . . | . 6 . | . . 3 |
| 4 . . | 8 . 3 | . . 1 |
| 7 . . | . 2 . | . . 6 |
+-------+-------+-------+
| . 6 . | . . . | 2 8 . |
| . . . | 4 1 9 | . . 5 |
| . . . | . 8 . | . 7 9 |
+-------+-------+-------+
```

区：
有二格可填 7

简接排除法

```
+-------+-------+-------+
| 5 3 . | . 7 . | . . . |
| 6 . . | 1 9 5 | . . . |
| . 9 8 | . . . | 5 6 . |
+-------+-------+-------+
| 8 . . | . 6 . | . . 3 |
| 4 . . | 8 . 3 | 7 . 1 |
| 7 . . | . 2 . | . . 6 |
+-------+-------+-------+
| . 6 . | . . . | 2 8 . |
| . . . | 4 1 9 | . . 5 |
| . . . | . 8 . | . 7 9 |
+-------+-------+-------+
```

区：
仅一格可填 7

GLPK/GMPL coding

变量定义

```
var x{i in 1..9, j in 1..9, k in 1..9}, binary; /* [i,j]为k的变量 */
```

初始条件

```
s.t. fa{i in 1..9, j in 1..9, k in 1..9: givens[i,j] != 0}:
  x[i,j,k] = (if givens[i,j] = k then 1 else 0); /*"givens"已给*/
```

约束条件

```
s.t. fb{i in 1..9, j in 1..9}: sum{k in 1..9} x[i,j,k] = 1;/*单值*/
s.t. fc{i in 1..9, k in 1..9}: sum{j in 1..9} x[i,j,k] = 1;/*行*/
s.t. fd{j in 1..9, k in 1..9}: sum{i in 1..9} x[i,j,k] = 1;/*列*/
s.t. fe{I in 1..9 by 3, J in 1..9 by 3, k in 1..9}:
     sum{i in I..I+2, j in J..J+2} x[i,j,k] = 1;         /*区唯一*/
```

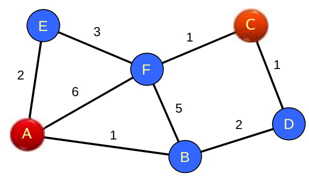
经0.340秒计算的结果

```
+-------+-------+-------+
| 5 3 4 | 6 7 8 | 9 1 2 |
| 6 7 2 | 1 9 5 | 3 4 8 |
| 1 9 8 | 3 4 2 | 5 6 7 |
+-------+-------+-------+
| 8 5 9 | 7 6 1 | 4 2 3 |
| 4 2 6 | 8 5 3 | 7 9 1 |
| 7 1 3 | 9 2 4 | 8 5 6 |
+-------+-------+-------+
| 9 6 1 | 5 3 7 | 2 8 4 |
| 2 8 7 | 4 1 9 | 6 3 5 |
| 3 4 5 | 2 8 6 | 1 7 9 |
+-------+-------+-------+
```

第六章 最优化理论概要

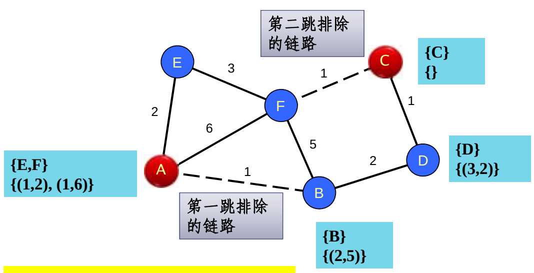
最小成本网络流问题
Minimum Cost Network Flow
Min-Max
MCF
Network Simplex(网络单形)

Dijkstra求解跳数最短路由



[Prob]
剩余资源有限，如何求最短路由？

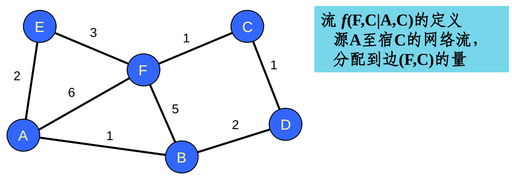
容量约束：bw>=2



[Prob]
如何求(A,C)最大容量的最短路径？

最优化模型的定义

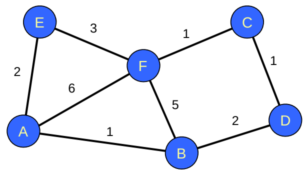
- $G(V, E)$，$n=|V|=5$, $m=|E|=8*2=16$
- 成本：$w(u,v)$；容量：$c(u,v)$, 边$(u,v) \in E$
  - *eg.*, $c(A,E) = 2$, $c(A,F) = 6$, …
- 流：$f(u,v|s,t)$，点 $s,t \in V$, 边$(u,v) \in E$.
- 总成本最小：min sum$\{(u,v), (s,t)\}$ $f(u,v|s,t)\cdot w(u,v)$



约束条件

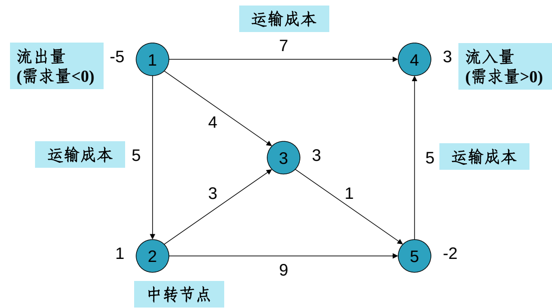
f(F,C|A,C):
源A至宿C的网络流，
分配到边(F,C)的量

- 容量：sum{(s,t)} $f(u,v|s,t)$<=$c(u,v)$, for each 边(u,v) ∈E
- 对称：$f(u,v|s,t)$ = - $f(v,u|s,t)$
- 守恒：sum{v, (s,t)} $f(u,v|s,t)$ = 0， for each $u \neq s, t$
- 源点：sum{v, (s,t)} $f(s,v|s,t)$ = $d(s)$, for each 边(s,v) ∈E
- 目标：sum{v, (s,t)} $f(v,t|s,t)$ = $d(t)$, for each 边(v,t) ∈E

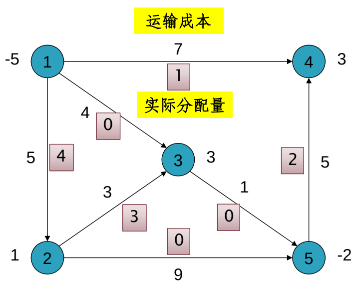


如果取消容量约束
则等同于最短路径问题

货品流的有向图示例

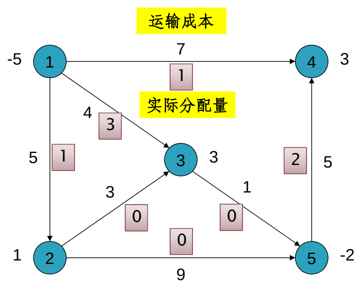


流分配的可行解



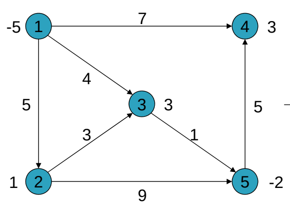
总成本:
7 +
5*4 + 2*5 +
3*3
= 46

流分配的最优解



总成本:
7 +
5 + 3*4 + 2*5
= 34

图的代数表示



容量：行向量

$$\mathbf{c} = [5 \quad 4 \quad 7 \quad 3 \quad 9 \quad 1 \quad 5]$$

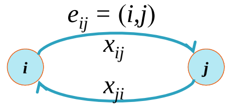
$$\mathbf{x} = \begin{bmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{23} \\ x_{25} \\ x_{35} \\ x_{54} \end{bmatrix}$$

分配量：列向量

$$\mathbf{b} = [-5 \quad 1 \quad 3 \quad 3 \quad -2]$$

需求量：列向量

LP公式表述



$e_{ij} = (i,j)$

x_{ij}

x_{ji}

$e_{ji} = (j,i)$

$a_{i,(i,j)} = -1$

$a_{(j,i),i} = +1$

$$\text{minimize } \mathbf{cx} = \sum_{ij} c_{ij} x_{ij}$$

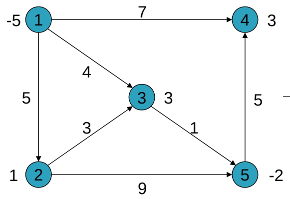
subject to

$$(ij) \qquad x_{ij} \geq 0$$

$$(i) \quad \sum_{ji} x_{ji} - \sum_{ij} x_{ij} = b_i$$

$$\sum_i b_i = 0$$

关联矩阵



$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c}
\begin{array}{ccccccc} 12 & 13 & 14 & 23 & 25 & 35 & 54 \end{array} \\
\begin{bmatrix}
-1 & -1 & -1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & -1 & -1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & -1
\end{bmatrix}
\end{array}
$$

LP表示式的简化

minimize $\mathbf{cx}$

subject to

$(ij)\quad x_{ij} \geq 0$

$\mathbf{Ax} = \mathbf{b}$

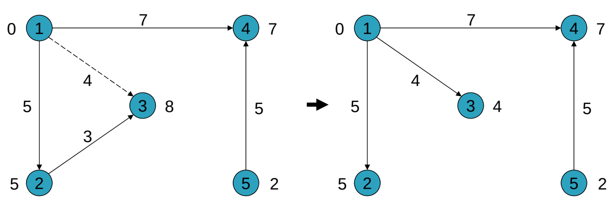
$\sum_i b_i = 0$

网络流的
网络单纯形法求解

解题思路

- 以一个初始生成树为起点
  - 该生成树包含网络的所有节点
- 可行树解 $\boldsymbol{x}$ 与生成树 $\boldsymbol{T}$ 相关联，且
  - $x_{ij} = 0$ if (i,j) 不是树 $\boldsymbol{T}$ 的边
- 通过查找所有可行树解，得到最优

说明示例，源于节点1的流



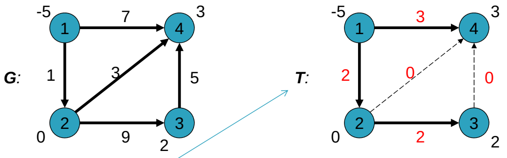
1）节点1，累计成本为0;
2）节点2，累计成本为5;
3）节点3，累计成本为8;
4）节点4，累计成本为7;
5）节点5，累计成本为7 − 5 = 2。

1）选择边 (1, 4)，替代 (2, 3);
2）节点3，累计成本减为4。

代数符号说明

- 从树 $\boldsymbol{T}$ 开始，求可行解 $\boldsymbol{x}$
  - 向量 $\boldsymbol{x}$，包括每条边上所分配流的大小
- step 1中，对每个节点计算
  - $y_i + c_{ij} = y_j$, for each $(i,j) \in \boldsymbol{T}$.
  - 向量 $\boldsymbol{y}$，流经节点的成本
- $\boldsymbol{c}$ 为成本向量, $\boldsymbol{b}$ 为需求向量, $\boldsymbol{A}$ 为关联矩阵

第二个例子



$$\mathbf{y}\ [0\ \ 1\ \ 10\ \ 7] \qquad \mathbf{A}\ \begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{ccccc} 12 & 14 & 23 & 24 & 34 \end{array} \\ \begin{bmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array} \qquad \mathbf{x}\ \begin{bmatrix} 2\\3\\2\\0\\0 \end{bmatrix} \qquad \mathbf{b}\ \begin{bmatrix} -5\\0\\2\\3 \end{bmatrix}$$

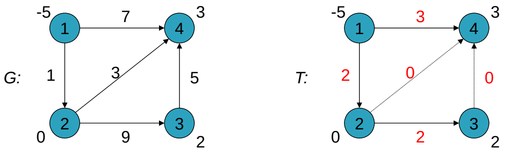
$$\mathbf{c}\ [1\ \ 7\ \ 9\ \ 3\ \ 5]$$

选择一条添加边(Step 1)

- 定义： c’ = c – yA.
- c’ 为边成本与所分配流的差，显然 c = c’ + yA

- If $ij \in T$ then $c'_{ij} = c_{ij} + y_i - y_j = 0$.
- If $ij \notin T$ 且 if $c'_{ij} < 0$， then ij 为候选边.
- Also if $ij \notin T$ then $x_{ij} = 0$,
- 综合以上关系，有：
- c’x = 0 ($\forall$ ij, either $c'_{ij} = 0$ or $x_{ij} = 0$).

更新示例



$$\mathbf{y}\ [0\ \ 1\ \ 10\ \ 7] \qquad \mathbf{A}\ \begin{array}{c} \\ 1\\2\\3\\4 \end{array}\begin{array}{c} \begin{array}{ccccc} 12 & 14 & 23 & 24 & 34 \end{array} \\ \begin{bmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array} \qquad \mathbf{x}\ \begin{bmatrix} 2\\3\\2\\0\\0 \end{bmatrix} \qquad \mathbf{b}\ \begin{bmatrix} -5\\0\\2\\3 \end{bmatrix}$$

$$\mathbf{c}\ [1\ \ 7\ \ 9\ \ 3\ \ 5] \qquad \mathbf{c}'\ [0\ \ 0\ \ 0\ \ -3\ \ 8]$$

代数表示(Step 1)

- 对于可行解 x’ (Ax’ = b, x’ ≥ 0), 其成本为:
  - cx’ = (c’ + yA)x’ (c’ = c – yA)
  - = c’x’ + yAx’
  - = c’x’ + yb. (Ax’ = b)
- 对于 x, 其成本为：
  - cx = c’x + yb = yb. (c’x = 0)
- 将 yb 用 cx 替代，有：
  - cx’ = c’x’ + cx (1)
- 所以，if c’x’ < 0 ， x’ 是比 x 更好的解.

代数表示(Step 2)

- step 2中，找到 $e = uv$， 满足
- $y_u + c_{uv} < y_v$ (i.e. $c'_{uv} < 0$).

- 如果无此边，then **c’** ≥ **0** and so **c’x’** ≥ **0**.
- 由 cx’ = c’x’ + cx 喻示对于所有可行解**x’**，都有
- **cx’** ≥ **cx**, 即**x**为最优

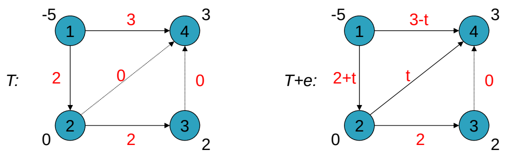
- 如果有此边，e, 将其加入到树 $\boldsymbol{T}$.

代数表示(Step 3)

- step 3中, $T + e$ 存在一个圈.
- 依照e的方向，将圈中的边分为前向和反向边.
- 设置:

$$\bar{x}_{ij} = \begin{cases} x_{ij} + t & \text{if } ij \text{ is a forward arc,} \\ x_{ij} - t & \text{if } ij \text{ is a reverse arc,} \\ x_{ij} & \text{if } ij \text{ is not on the cycle.} \end{cases}$$

图示(Step 3)



$$
\begin{array}{c} c' \\ \begin{bmatrix} 0 & 0 & 0 & -3 & 8 \end{bmatrix} \end{array}
\qquad
\begin{array}{c} x \\ \begin{bmatrix} 2 \\ 3 \\ 2 \\ 0 \\ 0 \end{bmatrix} \end{array}
\qquad
\begin{array}{c} x' \\ \begin{bmatrix} 2+t \\ 3-t \\ 2 \\ 0+t \\ 0 \end{bmatrix} \end{array}
$$

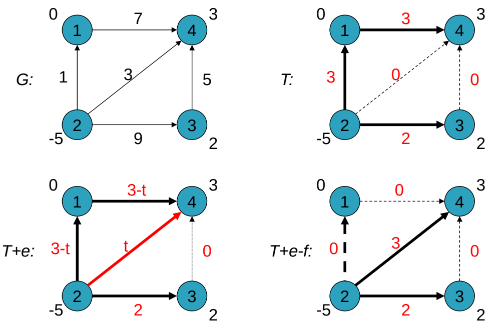
代数表示(Step 3)

- 因$\pm t$相互抵消, **Ax’** = **Ax** = b.
- 所以，合适的 t，满足 **x’** ≥ 0, 则 **x’** 是可行解.
- 因只有 e 满足 $c'_{ij} \neq 0$ 且 $x'_{ij} \neq 0$, 所以:
- **c’x’** = $c'_e x'_e = c'_e t$.
- 替代到**cx’** 的表达，得：**cx’** = **cx** + $c'_e t$.
- 问题变换为选择 t，使用 **x’** 为可行解，并减少**cx’**.

代数表示(Step 3)

- 如前, **cx’** = **cx** + $c'_e t$.
- 由于 $c'_e < 0$, 减少 **cx’**, 就是增长t.
- 为保证 **x’** 为可行解 (i.e. **x’** ≥ 0), 需要找到反向边 f，该边具有最小值 x_f, 得到 $t = x_f$.
- 可行解 **x’**，有 $x'_f = 0$, 因此f为移除边.
- 移除边 f，正好解除了 $T + e$ 中圈，因此，
  - $T + e - f$ 对应于新可行解 **x’**.

第三个示例求解过程

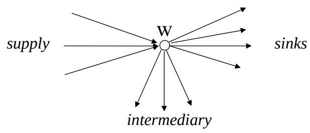


网络单纯形法的初始树

初始树，即第一个树

- 如果存在一节点 w:
  - 所有源都有边连接到 w
  - w有边连接到所有的宿和中间节点

  则初始树以w为根



人工边，增广图

- 如果不存在这类节点 w, 人为增加边或节点
- 并为边（ij）关联惩罚函数 p_{ij} :
  - $p_{ij} = 0$ for original arcs
  - $p_{ij} = 1$ for artificial arcs
- 初始树求解，变换为附加问题:
    - min sum $p_{ij}x_{ij}$

第六章 最优化理论概要思考题

针对以下整数规划：

Maximize: $5X_1+8X_2$

s.t $6X_1+5X_2<=30$

$9X_1+4X_2<=36$

$X_1+2X_2<=10$

采用图形法求最优解。