数学建模笔记

数学建模

姓 名 Soleil

学号 lalalala~

学院 信电学院

2024 年 10 月 17 日

数学建模

摘 要

1 绪论

2 Google 中的数学模型

2.1 搜索引擎（search engine）

2.1.1 概念

> 一种用于帮助用户查找因特网信息的技术工具。其工作原理是先以一定的策略在因特网中搜索，发现信息，然后对信息进行理解、提取。组织和处理，最后为用户提供信息和服务

2.1.2 网页重要度 (模型的第一种解释)

为了排序网页，我们首先肯定需要定量地描述网页的重要度。于是，我们给出以下网页重要度的原则与假设：

1. 传递性：
   重要度大的网页链接到网页 A 时，它对 A 重要度的贡献大于重要度小的网页 **某网页对其他网页重要度的贡献之和等于它自身的重要度**
2. 等效性：网页对它所链接的每个网页的重要度贡献相等**某网页对其他网页的重要度贡献与它所链接的网页数量成反比**
3. 叠加性：链接到网页 A 的网页越多，A 越重要**网页 A 的重要度是所有链接到它的网页的重要度贡献之和**
4. 无关性：网页链接其它网页的多少，与其本身的重要度无关

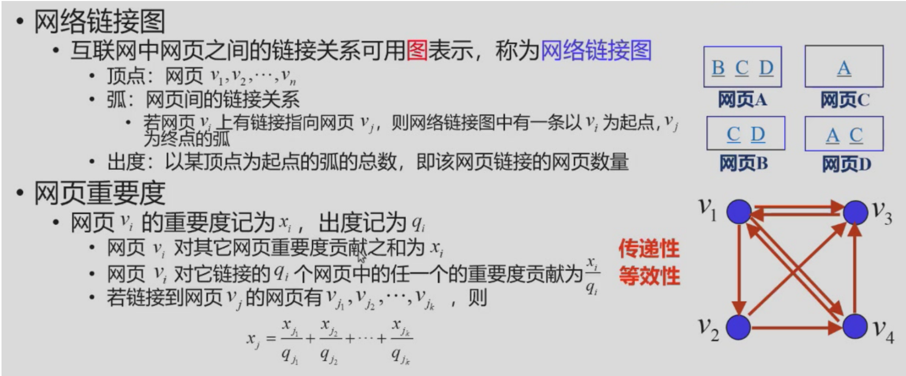


图 1: 网页重要度

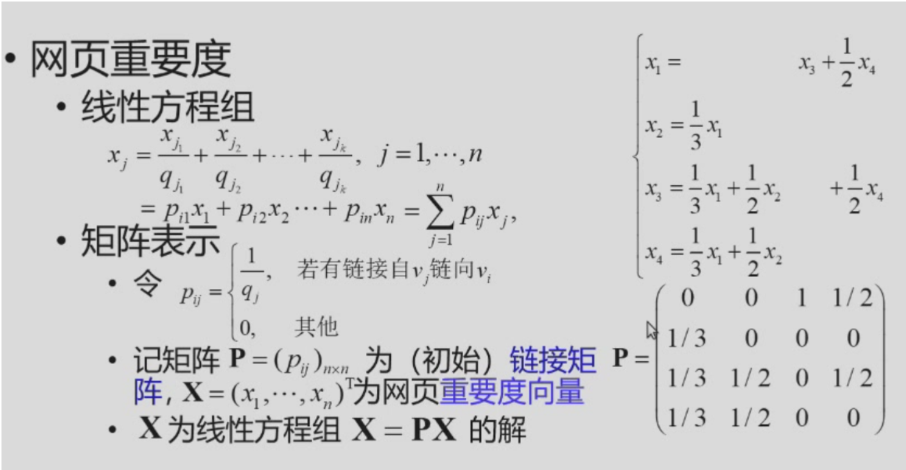


图 2: 初步矩阵化

最终化为如下方程

$$\mathbf{x} = \mathbf{P}\mathbf{x}$$

但是，考虑一个问题：**这个线性方程组是否一定有非零解**

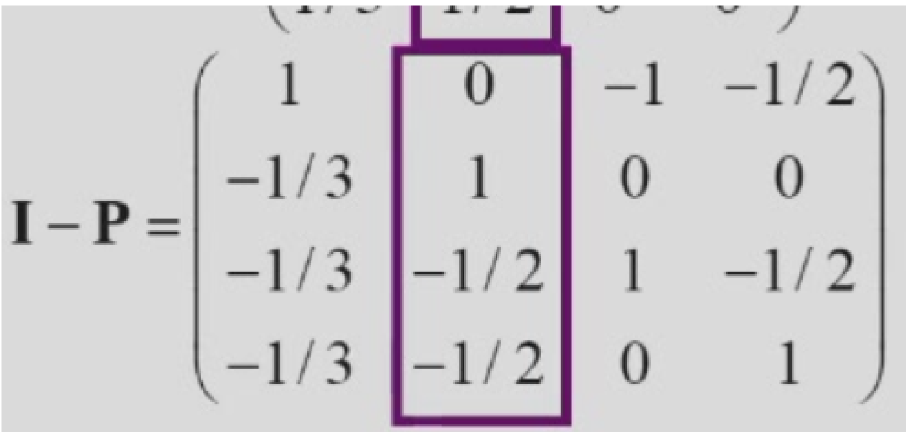


图 3: $I - P$

是的，因为行向量组是线性无关的。**可是，矩阵运算的时间复杂度**大致为 $O(n^3)$，还是很慢的

不妨换一种思路，$\mathbf{x} = \mathbf{P}\mathbf{x}$ 亦是 $\mathbf{P}$ 矩阵是否有一个特征值为 1，那接下来我们引入随机矩阵的概念

2.2 随机矩阵

2.2.1 概念

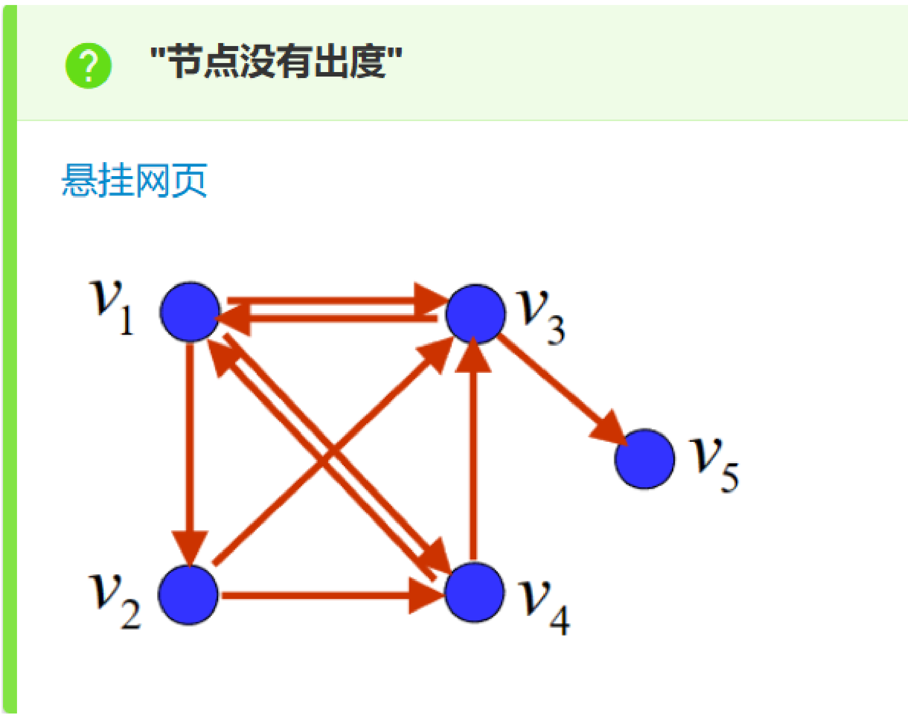
2.2.2 (未补) 随机矩阵的性质

各行（列）元素之和均为 1 的非负方阵称为行（列）随机矩阵

各行和各列元素之和均为 1 的非负方阵称为双随机矩阵

2.3 修正解

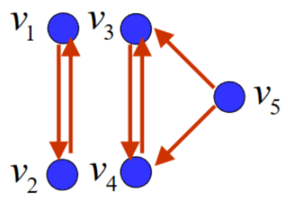
2.3.1 连接矩阵



2.3.2 特征值为 1 的特征向量不唯一

"这个线性方程组有多个解"

若$\overline{\mathbf{P}}$有两个属于特征值1的线性无关的特征向量，我们就无法得到唯一的网页重要度向量$\mathbf{x}$。如下图。



$$\mathbf{P}=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1/2 \\ 0 & 0 & 1 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

若 $\mathbf{P}$ 有两个属于特征值 1 的线性无关的特向量, 我们就无法得到唯一的网页重要度向量。

于是，我们对 $\mathbf{P}$ 进行修正，使得 $\mathbf{P}$ 成为**完全正矩阵**，即 $\mathbf{P}$ 的所有方阵子的行列式都大于 0。

我们先给出修正的方法:

$$\mathbf{P}=\alpha\mathbf{P}+(1-\alpha)\frac{1}{n}\mathbf{1}\mathbf{1}^T$$

其中，α 为修正系数，$\alpha=0.85$。$\mathbf{P}$ 是完全正矩阵与列随机矩阵的结合。

证明 P 是列随机矩阵

$$\mathbf{1}^T\mathbf{P}=\alpha\mathbf{1}^T\mathbf{P}+(1-\alpha)\mathbf{1}^T\frac{1}{n}\mathbf{1}\mathbf{1}^T=\alpha\mathbf{1}^T+(1-\alpha)\mathbf{1}^T=\mathbf{1}^T$$

因此，$\mathbf{P}$ 是列随机矩阵。

2.3.3 (未补) PageRank 中的矩阵求解

3 运筹学

3.1 优化理论与算法 (起源)

- Fermat 定理（1637）
- Lagrange 乘子法（1788）

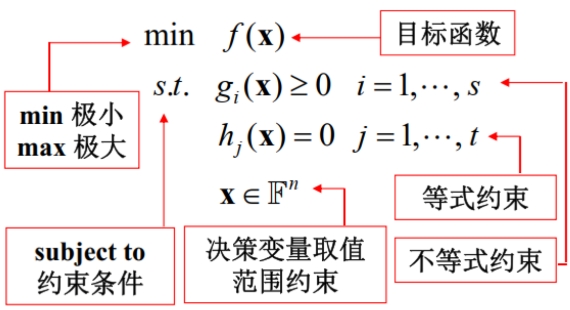
- Newton 法（1665）
- Cauchy 最速下降法（1847）

运筹学主要分支

- 数学规划（Mathematical Programming）
  - 线性规划（Linear Programming）
  - 非线性规划（Nonlinear Programming）
  - 整数规划（Integer Programming）
  - 多目标规划（Multiobjective Programming）
- 组合优化（Combinatorial Optimization）
- 随机运筹
  - 排队论（Queuing Theory）
  - 可靠性理论（Reliability Theory）
  - 库存论（Inventory theory）
- 博弈论（Game Theory）与决策理论 (Decision Theory)

3.2 运筹与统计–数学规划

定义 3.1. 数学规划：若干个变量在满足一些等式或不等式限制条件下，使一个或多个目标函数取得最大值或最小值



思考：为什么要把等式约束和不等式约束分开？

答：在优化理论中等式约束和不等式约束性质不尽相同，且一般避免“≠”

定义 3.2. 相关概念理解 *(感觉对于以下概念不定自明)*:

- 可行域

- 可行解
- 最优解
- 最优值

- 可行域
  $$S=\{\mathbf{x}\in\mathbb{F}^n \mid g_i(\mathbf{x})\geq 0, i=1,\cdots,s, h_j(\mathbf{x})=0, j=1,\cdots,t\}$$
- 可行解 $\mathbf{x}\in S$
- 最优解
  - $\mathbf{x}^*\in S$，且对任意 $\mathbf{x}\in S, f(\mathbf{x}^*)\leq f(\mathbf{x})$
- 最优值 $f(\mathbf{x}^*)$

按函数性质分类

- 线性规划（Linear Proramming, LP）
  - 目标函数为线性函数，约束条件为线性等式或不等式
- 非线性规划（Nonlinear Programming, NLP）
  - 目标函数为非线性函数，或至少有一个约束条件为非线性等式或不等式
    * 二次规划（Quadratic Programming, QP）：目标函数为二次函数，约束条件为线性等式或不等式
    * 带二次约束的二次规划（Quadratically Constrained Quadratic Program, QCQP）：目标函数为二次函数，约束条件为线性或二次等式或不等式
    * 线性分式规划（Linear Fractional Programming）：目标函数为两个线性函数的商，约束条件为线性等式或不等式

$$\begin{aligned}\min\quad & f(\mathbf{x})\\ s.t.\quad & g_i(\mathbf{x})\geq 0 \quad i=1,\cdots,s\\ & h_j(\mathbf{x})=0 \quad j=1,\cdots,t\\ & \mathbf{x}\in\mathbb{R}^n\end{aligned}$$

$$\begin{aligned}\min\quad & \sum_{i=1}^n c_i x_i\\ s.t.\quad & \sum_{j=1}^n a_{ij}x_j=b_i, i=1,\cdots,m\\ & x_1,\cdots,x_n\geq 0\end{aligned}
\qquad
\begin{aligned}\min\quad & \mathbf{cx}\\ s.t.\quad & \mathbf{Ax}=\mathbf{b}\\ & \mathbf{x}\geq\mathbf{0}\end{aligned}$$

$$\begin{aligned}\min\quad & \mathbf{x}^{\mathrm{T}}\mathbf{Ax}\\ s.t.\quad & \mathbf{Ax}=\mathbf{b}\\ & \mathbf{x}\geq\mathbf{0}\end{aligned}
\qquad
\begin{aligned}\min\quad & \frac{\mathbf{cx}+d}{\mathbf{ux}+v}\\ s.t.\quad & \mathbf{Ax}=\mathbf{b}\\ & \mathbf{x}\geq\mathbf{0}\end{aligned}$$

按变量性质分类

整数规划（Integer Programming, IP）：至少有一个决策变量限定取整数值

- 整数决策变量意义
  - 用于表示只能取离散值的对象的数量
  - 用于表示约束条件之间的逻辑关系或复杂的函数形式
  - 用于表示非数值的优化或可行性问题
- 特殊整数规划
  - 部分决策变量取整数值的数学规划特称为混合整数规划（Mixed Integer Programming, MIP）
  - 0-1 规划：决策变量仅取值 0 或 1 的数学规划

按约束条件

- 无约束优化
- 约束优化

3.3 数学规划建模

- 数学规划模型是问题要求和限制的真实反映
  - 数学规划模型的最优解（可行解）与问题最优解（可行解）是否一致或对应
  - 是否遗漏问题的隐含约束、决策变量的必然要求、多组决策变量间的联系等约束条件
- 数学规划模型应符合数学规划的内容规范和形式要求
  - 要素完整、变量指标运用准确。逻辑关系、集合运算等一般不在数学规划中出现
- 问题可能存在多个数学规划描述，需根据实际情况进行选择和不断完善
  - 复杂目标函数和约束条件的简化，**0-1 变量的灵活运用**
  - 可行域约简、数学规划的重构、分解与松弛

3.3.1 数学规划建模的适用范围

- 具有简单最优算法或可转化为已知多项式时间可解问题，不需运用数学规划求解
- 最优解不具必要性，或求解时间要求高于精度要求等问题，不宜盲目运用数学规划求解
- 建立数学规划模型困难，因实例规模或问题结构等原因使求解不具现实可能性的问题，不能直接运用数学规划求解

3.3.2 食谱问题（实质上感觉是多元线性规划）

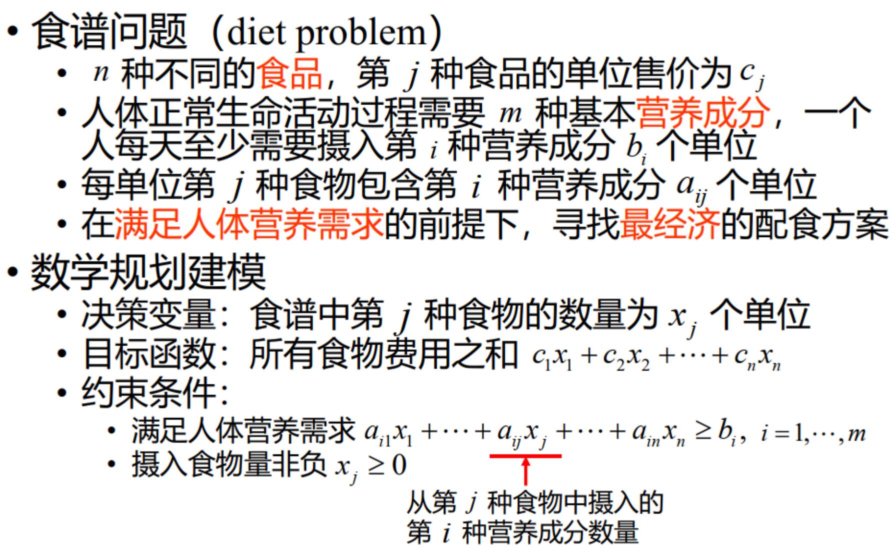


图 4: 食谱问题

mathmatic 求解方式：

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

$$约束为：\mathbf{Ax}=\mathbf{b}$$

注：此处 $\mathbf{Ax}=\mathbf{b}$ 也不一定是等号，代表一种约束

3.3.3 运输问题

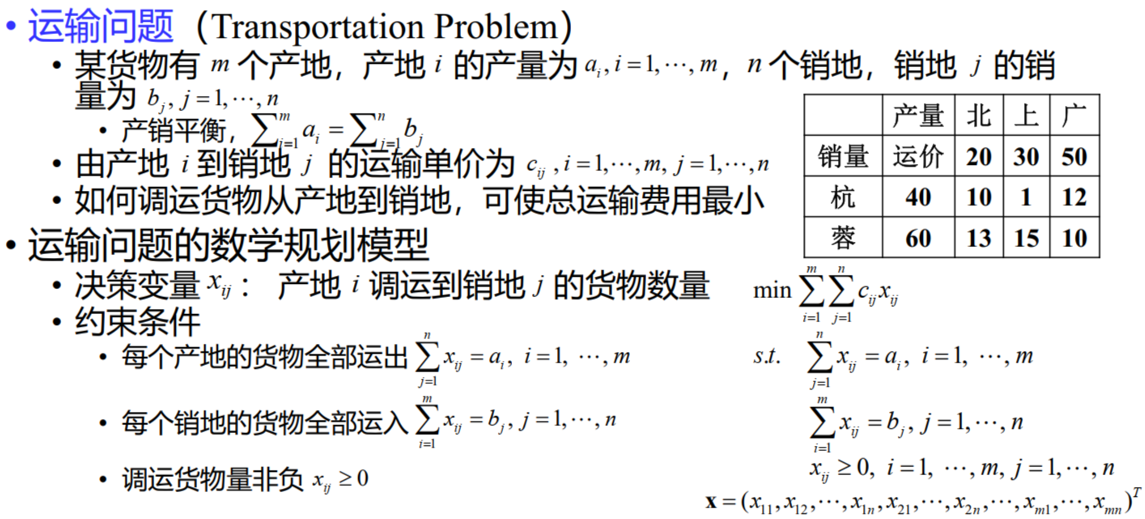


图 5: 运输问题

3.3.4 数独

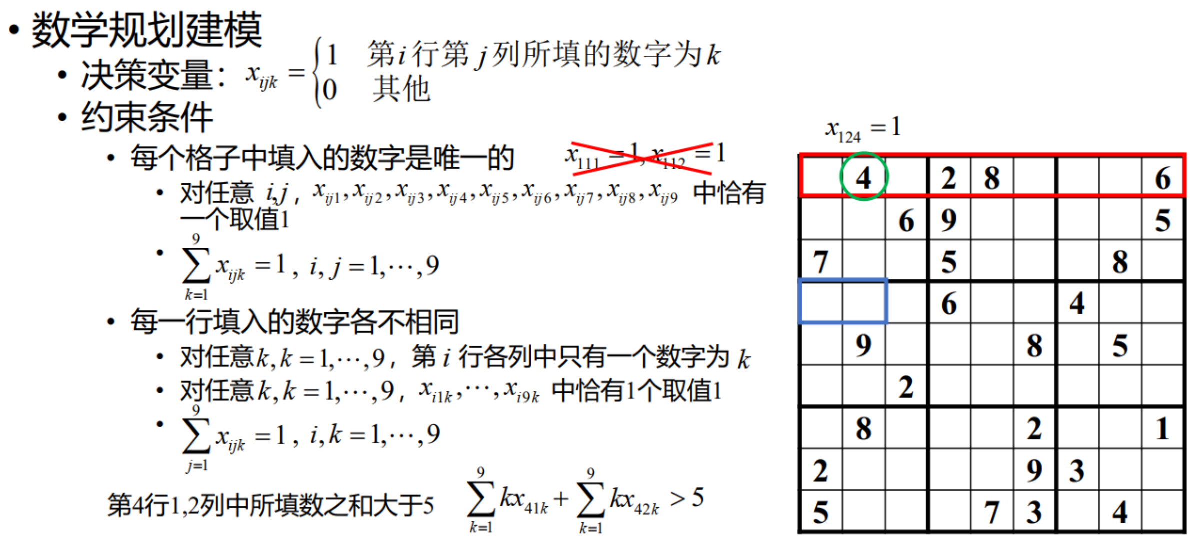


图 6: 数独

- **数学规划建模** 数学规划可以没有目标函数
  - s.t. $\sum_{k=1}^{9} x_{ijk} = 1,\ i, j = 1, \cdots, 9$ 每个格子中填入的数字唯一

    $\sum_{j=1}^{9} x_{ijk} = 1,\ i, k = 1, \cdots, 9$ 每行各列中的数各不相同

    $\sum_{i=1}^{9} x_{ijk} = 1,\ j, k = 1, \cdots, 9$ 每列各行中的数各不相同

    $\sum_{i=3s+1}^{3s+3} \sum_{j=3t+1}^{3t+3} x_{ijk} = 1,\ s, t = 0, 1, 2, k = 1, \cdots, 9$ 每个小方块中的数各不相同

    $x_{ijk} = 1,\ (i, j, k) \in P$ P 由预置的数字及其位置确定，

    $x_{ijk} = 0, 1,\ i, j, k = 1, \cdots, 9$ 如 $x_{124} = 1$，$x_{142} = 1$ 等

思考：为什么要以三角标的形式来写？

答：如果采用如下形式，无法不使用不等号来表示各个元素各不相同

- 决策变量：第 i 行第 j 列所填的数字
  - $x_{ij} = k$：第 i 行第 j 列所填的数字为 k

$\sum_{k=1}^{9} k x_{ijk}$ **的值为在这个格子里填的数字**

不好的写法约等于（不正确）：

决策变量（实际上就是变量）**不能作为下标或者求和符号上的上下界来使用，不能出现逻辑关系（if..then...），不能出现集合及相互关系、个数等逻辑/运算等**要脱离编程思维，如下：

错误示范：

$$x_{41j_1} = 1 \quad \text{且} \quad x_{41j_2} = 1$$

$$j_1 + j_2 \leq 0$$

因此决定进行升维，使用 0，1 变量作为决策变量的取值进行操作，可操作性也会加强

《0，1 变量的使用技巧》（重要重要很重要）

另外提及扩展：**数回**，maybe 考试可能会涉及

3.3.5 下料问题

问题背景

现有 W 米长的钢管若干。生产某产品需长为 w_i 米的短管 b_i 根，$i = 1, 2, \cdots, n$。如何截取能使材料最省？

我们构造数学规划模型：

决策变量：x_{ji} 表示第 j 根钢管截取第 i 种短管的数量，$i = 1,2,\cdots,k$，$j = 1,2,\cdots,n$。$(n=\sum\limits_{i=1}^{k} b_i)$

约束条件：

1. 每根钢管截取的短管总长度不超过钢管长度 W，即 $\sum\limits_{i=1}^{n} w_i x_{ji} \leq W$，$j=1,2,\cdots,n$
2. 每种短管截取的数量不低于需求量，即 $\sum\limits_{j=1}^{n} x_{ji} \geq b_i$，$i=1,2,\cdots,k$

我们的目标是使得截取的钢管最少，即 $\min n$。但这个 n 是出现在求和号上的，在线代的数学规划中我们往往不会直接去求解这个 n，而是将其转化。

我们构造一个 0-1 变量 y_j，表示第 j 根钢管是否被截取，即 $y_j=1$ 表示第 j 根钢管被截取，$y_j=0$ 表示第 j 根钢管未被截取，$j=1,2,\cdots,n$。

所以我们的目标函数可以写成 $\min\sum\limits_{j=1}^{n} y_j$。

但现在有一个问题：y_j 与 x_{ji} 之间的关系是什么呢?

$$\exists i, x_{ji}>0 \to y_j = 1 \Leftrightarrow \sum_{i=1}^{k} x_{ji} > 0 \to y_j = 1$$

约束条件 1 可以写成

$$\sum_{i=1}^{k} w_i x_{ji} \leq W y_j$$

，

$$j=1,2,\cdots,n$$

所以我们有：

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} y_j \\
s.t. \quad & \sum_{i=1}^{k} w_i x_{ji} \leq W y_j \\
& \sum_{j=1}^{n} x_{ji} \geq b_i \\
& x_{ji} \geq 0, y_j \in \{0,1\} \\
& \text{当} y_j = 1 \to \exists i, x_{ji} > 0 \\
& \text{当} \sum_{i=1}^{k} = 0\text{时，有} y_i = 0\text{(给定目标下，最优解自动满足左式)}
\end{aligned}
$$

另一种解法

• 决策变量 z_i：按第 i 种方式截取的原料的数量，$i=1,\cdots,7$

长度	需量	可能截取方案						
7米	200	2	1	1	0	0	0	0
5米	150	0	1	0	3	2	1	0
4米	100	0	0	2	0	1	2	3
15米	?	1	2	3	4	5	6	7

$$
\begin{aligned}
\min\quad & z_1+z_2+z_3+z_4+z_5+z_6+z_7 \\
s.t.\quad & 2z_1+z_2+z_3 \geq 200 \\
& z_2+3z_4+2z_5+z_6 \geq 150 \\
& 2z_3+z_5+2z_6+3z_7 \geq 100 \\
& z_i \geq 0 \text{ 且 } z_i \text{ 为整数}, \quad i=1,2,\cdots,7.
\end{aligned}
$$

若有多种钢管或短管类型过多，可行方案数量庞大，得到所有可行方案也是困难的

图 7: 另一种解法

思考：为什么是大于等于号

3.3.6 选址问题

问题背景

平面上有 n 个点，求一个面积最小的圆，使得这 n 个点都在圆内。

解决方法

记第 j 个点的坐标为 (x_j,y_j)，$j=1,2,\cdots,n$。我们给出一个带二次约束的二次规划模型：

决策变量：圆心坐标 (x_0,y_0)，圆半径 r

目标函数：$\min \quad r^2$

约束条件：$(x_j-x_0)^2+(y_j-y_0)^2 \leq r^2$，$j=1,2,\cdots,n$

这是一个二次约束规划问题，比较难写，我们可以将其转化成更简单的形式——————**引入新的决策变量**：

决策变量：改为 $\lambda = r^2-(x_0^2+y_0^2)$

目标函数：改为 $\min \quad \lambda+(x_0^2+y_0^2)$

约束条件：改为 $\lambda+2x_0x_j+2y_0y_j \geq x_j^2+y_j^2$，$j=1,2,\cdots,n$

3.3.7 时间分配问题

问题背景

有 T 天时间可用于安排复习 n 门课程，每天只能复习一门课程，每门课程至少复习一天。用 t 天时间复习第 j 门课程可使该门课程提高 p_{jt} 分。如何制定复习计划可使所有课程提高的总分尽可能大?

解决方法

决策变量：x_{jt} 表示第 j 门课程是否复习 t 天，$j=1,2,\cdots,n$，$t=1,2,\cdots,T$

$$x_{jt}=\begin{cases}1, & \text{第 } j \text{ 门课程复习 } t \text{ 天}\\ 0, & \text{其他}\end{cases}$$

目标函数：$\max\sum\limits_{j=1}^{n}\sum\limits_{t=1}^{T}p_{jt}x_{jt}$

约束条件：$\sum\limits_{t=1}^{T}x_{jt}=1$；$\sum\limits_{j=1}^{n}\sum\limits_{t=1}^{T}tx_{jt}\leq T$

3.4 相对复杂的数学规划问题 1——赛程编制

- Symmetry and separation(对称性和分离性)
- Breaks(出现连续的客场或主场比赛–这是我们不愿看到的)
- The carry-over effect

关于 n 支队伍的赛程模式

对于 n（偶数）支队伍的赛程，用形如 HAH…HA，长度为 $n-1$（奇数）的字符串表示每支队伍的主客场安排，称为**模式（pattern）**：

- 任何两支队伍的模式互不相同。

单循环赛程

n 支队伍的单循环赛程，全程所有队伍总 break 数至少为 $n-2$。

说明：

因为任意两支队伍的模式互不相同，而只有 HAHA…HAH 和 AHAH…AHA 两种模式没有 break，其它模式的 break 数至少为 1，所以总 break 数至少为 $n-2$。

镜像双循环赛程

n 支队伍的镜像双循环赛程，全程所有队伍总 break 数至少为 $3n-6$。

1. 若半程没有 break,则全程也没有 break,这样的队伍至多有两支(HAHAH-AHAHA，AHAHA-HAHAH)。
2. 若半程有且仅有一个 break，由于模式字符串长度为奇数，在前后半程之间有一个 break（H **A A** H **A - A H H** A H)，break 数为 3。
3. 若半程有至少两个 break，全程 break 数至少为 4。

因此，总 break 数至少为 $3(n-2)$。

n（偶数）支队伍的镜像赛程中的 double-round break

此时赛程分成多个阶段，我们**只考虑每个阶段中的 double-round break, 而不考虑各个阶段边界处的 break**

因为队伍为偶数支，所以每支队伍的模式字符串长度为奇数。

- 若半程没有 break,则全程也没有 break,这样的队伍至多有两支(HAHAH-AHAHA，AHAHA-HAHAH)。
- 若半程至少有 1 个 break，全程至少有 2 个 double-round break（H A | A H | **A - A** | **H H** | A H)，根据奇数长度的模式字符串，我们可以得到：
  - 前后半程之间若有 break，必为 double-round break。
  - 若前半程的 break 不为 double-round，后半程的 break 必为 double-round。

所以总 double-round break 数至少为 $2(n-2)$。

其他方案

	1	2	3	…	9	10	11	12	…	17	18	
镜像（mirror）	1	2	3	…	9	1	2	3	…	8	9	意、德
法制（French）	1	2	3	…	9	2	3	4	…	9	1	法、俄
英制（English）	1	2	3	…	9	9	1	2	…	7	8	奥
逆向（Inverted）	1	2	3	…	9	9	8	7	…	2	1	瑞士

图 8: 其他方案

3.4.1 数学规划求解

我们来到实际问题：有 10 支队伍，每阶段两场比赛

决策变量：$x_{ijk}=\begin{cases}1, & 第k轮第i支队伍在主场对阵第j支队伍\\ 0, & 其他\end{cases}$

约束条件

1. 每轮各队恰有一场比赛：
$$\sum_{i=1}^{10}(x_{ijk}+x_{jik})=1,\quad j=1,2,\cdots,10,\quad k=1,2,\cdots,18$$
2. 任意两队在前后半程各交手一次：
$$\sum_{k=1}^{18}x_{ijk}=1,\quad i,j=1,2,\cdots,10,\quad i\neq j$$
3. 任意两队之间的两场比赛中每队均有一个主场：
$$\sum_{k=1}^{9}(x_{ijk}+x_{jik})=1,\quad \sum_{k=9}^{18}(x_{ijk}+x_{jik})=1,\quad i,j=1,2,\cdots,10,\quad i\neq j$$
4. 任一队不连续与种子队（用 I_s 表示）对阵：
$$\sum_{j\in I_s}\left(x_{ijk}+x_{jik}+x_{i,j,k+1}+x_{j,i,k+1}\right)\leq 1,\quad i\in I\setminus I_s,\quad k=1,\cdots,18$$

均衡各阶段主场次数

每支队伍各阶段先主后客（或先客后主）的次数尽可能均衡：

定义辅助变量：
$$y_{il}=\begin{cases}1, & 第l阶段第i支队伍先主后客\\ 0, & 第l阶段第i支队伍先客后主\end{cases}$$

x_{ijk} 与 y_{il} 之间的关系：
$$y_{il}=1\Leftrightarrow\sum_{j=1}^{10}x_{i,j,2l-1}=1且\sum_{j=1}^{10}x_{j,i,2l}=1$$

改写为以下形式：
$$\begin{cases}\sum\limits_{j=1}^{10}\left(x_{i,j,2l-1}+x_{j,i,2l}\right)\leq 1+y_{il}\\ y_{il}\leq\sum\limits_{j=1}^{10}x_{i,j,2l-1}\\ y_{il}\leq\sum\limits_{j=1}^{10}x_{j,i,2l}\end{cases}$$

每支队伍先主后客的总次数尽可能均衡时，满足以下条件：
$$4\leq\sum_{l=1}^{9}y_{il}\leq 5,\quad i=1,2,\cdots,10$$

各阶段连续客场次数尽可能少

在法制双循环赛制中，满足以下对称条件：

$$x_{i,j,1}=x_{j,i,18},\quad x_{i,j,k}=x_{j,i,k+8},\quad k=2,3,\cdots,9,\quad i,j=1,2,\cdots,10,\quad i\neq j$$

定义辅助变量：

$$w_{il}=\begin{cases}1, & \text{第}l\text{阶段第}i\text{支队伍两场比赛均为客场}\\ 0, & \text{其他}\end{cases}$$

$x_{i,j,k}$ 与 w_{il} 之间的关系：

$$w_{il}=1\Leftrightarrow\sum_{j=1}^{10}x_{j,i,2l-1}=1\text{且}\sum_{j=1}^{10}x_{j,i,2l}=1$$

改写为以下形式：

$$\begin{cases}\sum\limits_{j=1}^{10}(x_{j,i,2l-1}+x_{j,i,2l})\leq 1+w_{il}\\ w_{il}\leq\sum\limits_{j=1}^{10}x_{j,i,2l-1}\\ w_{il}\leq\sum\limits_{j=1}^{10}x_{j,i,2l}\end{cases}$$

目标函数

目标函数为：

$$\min\sum_{i=1}^{10}\sum_{l=1}^{9}w_{il}$$

3.5 支持向量机

我们拟将一数据集 S 分为 C_1,C_2 两类。每个数据有 n 个特征，用 n 维实向量表示。我们有训练集 $S'=\{\mathbf{x}_1,\cdots,\mathbf{x}_m\}$，其中 $\mathbf{x}_i\in\mathbb{R}^n$，并记

$$y_i=\begin{cases}1, & \mathbf{x}_i\in C_1\\ -1, & \mathbf{x}_i\in C_2\end{cases}$$

假设训练集可线性分离，即存在超平面 $\mathbf{w}\cdot\mathbf{x}+b=0$，使得对于所有 i，有 $y_i(\mathbf{w}\cdot\mathbf{x}_i+b)>0$。

超平面

设 $\mathbf{w}$ 为 n 维实向量，b 为实数，$\mathbf{w}\cdot\mathbf{x}+b=0$ 称为 $\mathbb{R}^n$ 中的超平面。

- $\mathbb{R}^n$ 中的点 $\mathbf{x}$ 到超平面 $\mathbf{w}\cdot\mathbf{x}+b=0$ 的距离为 $\frac{|\mathbf{w}\cdot\mathbf{x}+b|}{\|\mathbf{w}\|}$。- 若 $\mathbf{w}\cdot\mathbf{w}=1$，则距离为 $|\mathbf{w}\cdot\mathbf{x}+b|$。

我们现在要做的是找到一个超平面，使得**它到两类数据的距离的最小值达到最大（判别效果最好）**。

数学规划问题

根据上述描述，可以得到如下的数学规划问题：

$$\begin{aligned} &\max \min_{i=1,\cdots,m} |\mathbf{w}\cdot\mathbf{x}_i+b| \\ \text{s.t.}\quad & y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 0,\quad i=1,\cdots,m \\ & \|\mathbf{w}\|=1 \end{aligned}$$

此目标函数含绝对值与极小值，约束含二次函数，计算较为复杂。为了简化，我们去掉绝对值，得到如下的数学规划问题：

$$\begin{aligned} &\max \min_{i=1,\cdots,m} y_i(\mathbf{w}\cdot\mathbf{x}_i+b) \\ \text{s.t.}\quad & \|\mathbf{w}\|=1 \end{aligned}$$

- 若（I）有解，（I）与（II）等价
  - （I）的可行域包含在（II）的可行域内
  - （II）的最优解在（I）的可行域内
    - 由（I）有解，存在 $(\mathbf{w},b)$，满足 $\mathbf{w}\cdot\mathbf{w}=1$ 与 $y_i(\mathbf{w}\cdot\mathbf{x}_i+b)>0,\ i=1,\cdots,m$。$(\mathbf{w},b)$ 也是（II）的一组可行解，目标值 $\min\limits_{i=1,\cdots,m} y_i(\mathbf{w}\cdot\mathbf{x}_i+b)>0$
    - （II）的最优解 $(\mathbf{w}^*,b^*)$ 的目标值 $\min\limits_{i=1,\cdots,m} y_i(\mathbf{w}^*\cdot\mathbf{x}_i+b^*)\geq\min\limits_{i=1,\cdots,m} y_i(\mathbf{w}\cdot\mathbf{x}_i+b)>0$，即 $(\mathbf{w}^*,b^*)$ 满足 $y_i(\mathbf{w}^*\cdot\mathbf{x}_i+b^*)>0,\ i=1,\cdots,m$，$(\mathbf{w}^*,b^*)$ 是（I）的一组可行解
  - 在（I）的可行域内，对相同决策变量，（I）的目标值与（II）的目标值相等
    - 由 $y_i=\pm1$，若 $y_i(\mathbf{w}\cdot\mathbf{x}_i+b)>0$，则 $y_i(\mathbf{w}\cdot\mathbf{x}_i+b)=|\mathbf{w}\cdot\mathbf{x}_i+b|$

进一步简化

虽然去掉了绝对值，但该目标函数仍然含有极小值，且约束依然是二次的。我们可以将其进一步简化为以下数学规划问题：

$$\begin{aligned} &\min \|\mathbf{w}\| \\ \text{s.t.}\quad & y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1,\quad i=1,\cdots,m \end{aligned}$$

- 若 $(\mathbf{w}_0, b_0)$ 是（III）的最优解，则 $\left(\frac{\mathbf{w}_0}{\sqrt{\mathbf{w}_0\cdot\mathbf{w}_0}}, \frac{b_0}{\sqrt{\mathbf{w}_0\cdot\mathbf{w}_0}}\right)$ 是（II）的最优解
    - 设 $(\mathbf{w}^*, b^*)$ 是（II）的最优解，$\mathbf{w}^*\cdot\mathbf{w}^*=1$，最优值 $\gamma^*=\min_{i=1,\cdots,m} y_i(\mathbf{w}^*\cdot\mathbf{x}_i+b^*)$
    - $y_i(\mathbf{w}^*\cdot\mathbf{x}_i+b^*)\ge\gamma^*$，$i=1,\cdots,m$，即 $y_i\left(\frac{\mathbf{w}^*}{\gamma^*}\cdot\mathbf{x}_i+\frac{b^*}{\gamma^*}\right)\ge 1$，$i=1,\cdots,m$，$\left(\frac{\mathbf{w}^*}{\gamma^*}, \frac{b^*}{\gamma^*}\right)$ 是（III）的可行解
    - 由于 $(\mathbf{w}_0, b_0)$ 是（III）的最优解，$\sqrt{\mathbf{w}_0\cdot\mathbf{w}_0}\le\sqrt{\frac{\mathbf{w}^*}{\gamma^*}\cdot\frac{\mathbf{w}^*}{\gamma^*}}=\frac{1}{\gamma^*}$
    - $y_i\left(\frac{\mathbf{w}_0}{\sqrt{\mathbf{w}_0\cdot\mathbf{w}_0}}\cdot\mathbf{x}_i+\frac{b_0}{\sqrt{\mathbf{w}_0\cdot\mathbf{w}_0}}\right)\ge y_i\left(\gamma^*\mathbf{w}_0\cdot\mathbf{x}_i+\gamma^* b_0\right)=\gamma^* y_i\left(\mathbf{w}_0\cdot\mathbf{x}_i+b_0\right)\ge\gamma^*$，$i=1,\cdots,m$，故 $\left(\frac{\mathbf{w}_0}{\sqrt{\mathbf{w}_0\cdot\mathbf{w}_0}}, \frac{b_0}{\sqrt{\mathbf{w}_0\cdot\mathbf{w}_0}}\right)$ 的目标值不小于 $(\mathbf{w}^*, b^*)$ 的目标值，也是（II）的最优解

3.6 矩阵形式线性规划

$$
\begin{aligned}
&\min \mathbf{c}^\top\mathbf{x}\\
\text{s.t.}\quad &\mathbf{A}\mathbf{x}=\mathbf{b},\\
&\mathbf{x}\ge\mathbf{0}
\end{aligned}
$$

不妨设系数矩阵 $\mathbf{A}=(a_{ij})_{m\times n}$ 为行满秩矩阵，即 $\mathrm{rank}(\mathbf{A})=m$。

- 若 $\mathrm{rank}(\mathbf{A})<m$，则存在冗余约束。
- 不妨设 $\mathbf{A}$ 的前 m 列线性无关。
- 重排 $\mathbf{A}$ 的列，并相应调整决策变量的顺序。

3.6.1 标准型

- 目标函数极小化。
- 决策变量取非负值：若 x_j 无限制，则 $x_j=x_j^+-x_j^-$，且 $x_j^+, x_j^-\ge 0$。
- 所有约束均为等式约束：

$$\mathbf{A}\mathbf{x}\le\mathbf{b}\Leftrightarrow\mathbf{A}\mathbf{x}+\mathbf{y}=\mathbf{b}, \mathbf{y}\ge 0$$

$$\mathbf{A}\mathbf{x}\ge\mathbf{b}\Leftrightarrow\mathbf{A}\mathbf{x}-\mathbf{y}=\mathbf{b}, \mathbf{y}\ge 0$$

- 等式约束右端均为非负常数。

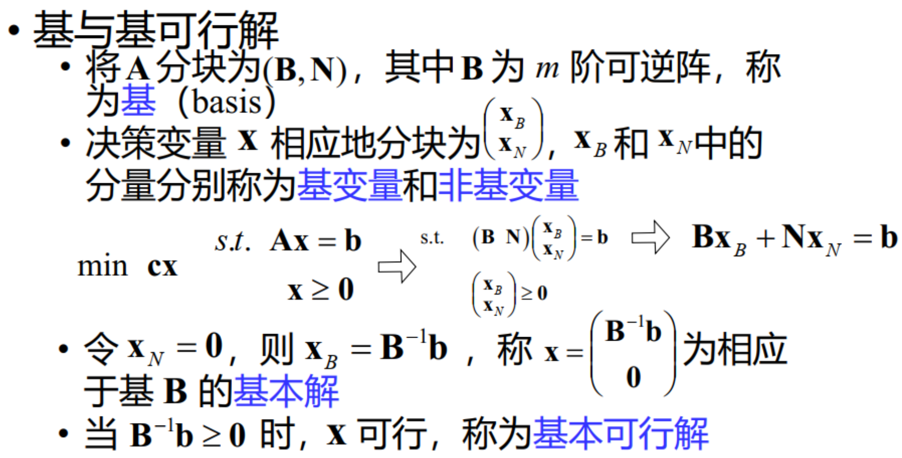
3.6.2 最优解的类型

线性规划有四种最优解的类型：

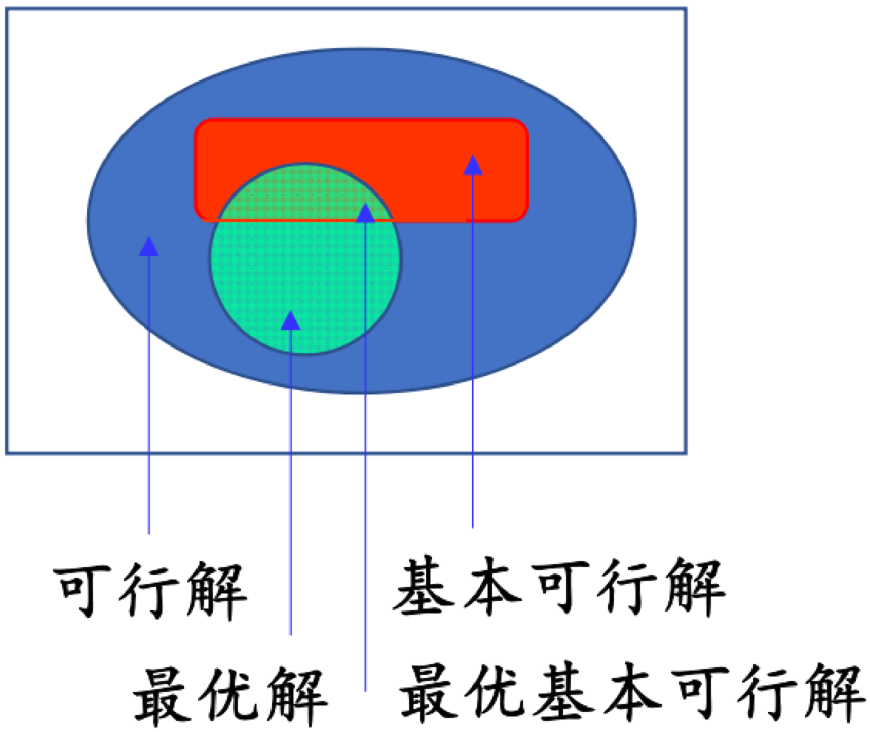
- 唯一最优解，

- 无穷多最优解，
- 无可行解，
- 无界解（有可行解，但最优值无下界）。

3.6.3 基与基可行解



3.6.4 线性规划基本定理



- 若线性规划有可行解，必有基本可行解。
- 若线性规划有最优解，必有最优基本可行解。

- 线性规划最优解，只需在所有基本可行解中去寻找。

3.7 求解算法

3.7.1 单纯形法

从一个基本可行解转到另一个基本可行解，并使目标值下降。迭代有限次，找到最优解或判断最优值无界。

注意：单纯形法是指数时间算法。存在含 m 个变量、m 个约束的线性规划，单纯形法需要进行 $2^m - 1$ 次迭代。

实践表明, 对多数线性规划问题, 单纯形法迭代次数为变量和约束数的多项式函数。

3.7.2 多项式时间算法

1979 年，Khachiyan 给出了求解线性规划的第一个多项式时间算法——**椭球算法**。

1984 年，Karmarkar 给出了实际效果更好的多项式时间算法——**内点法**，产生了深远的影响。

3.8 整数线性规划

3.8.1 分枝定界法

分枝定界法是求解整数规划最常用的算法，算法思想可用于其它离散优化问题的求解。它是一种指数时间算法。

3.8.2 分枝定界法的基本思想

- 用线性规划求解松弛问题，得到松弛问题的最优解。
  - 若最优解为整数解，则该整数解为整数规划的最优解。
  - 若最优解不是整数解，则将松弛问题的可行域分为两个子可行域，分别求解这两个子问题。
- 重复上述过程，直到找到整数解或判断无整数解。

例子：

IP: min $-30x_1 - 36x_2$, s.t.

$$\begin{cases} x_1 + x_2 \le 6 \\ 5x_1 + 9x_2 \le 45 \\ x_1, x_2 \ge 0, x_1, x_2 \in \mathbb{Z} \end{cases}$$

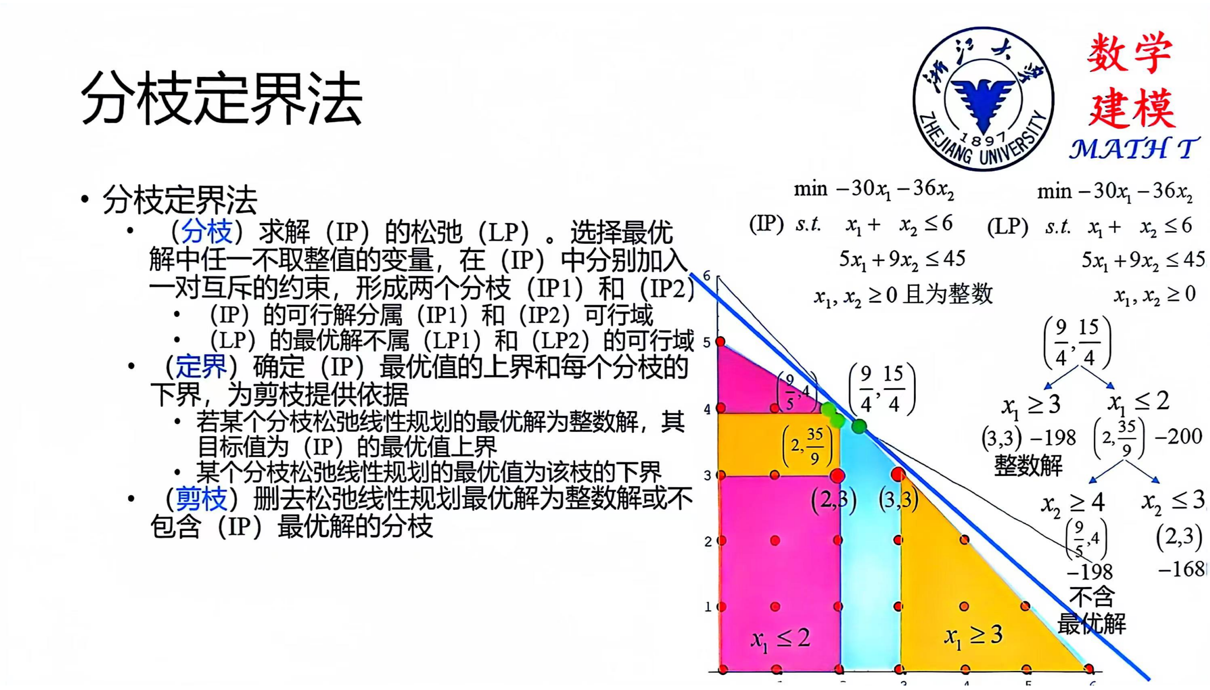
我们先求解松弛问题：

LP: $\min -30x_1 - 36x_2$, s.t.

$$
\begin{cases}
x_1 + x_2 \le 6 \\
5x_1 + 9x_2 \le 45 \\
x_1, x_2 \ge 0
\end{cases}
$$

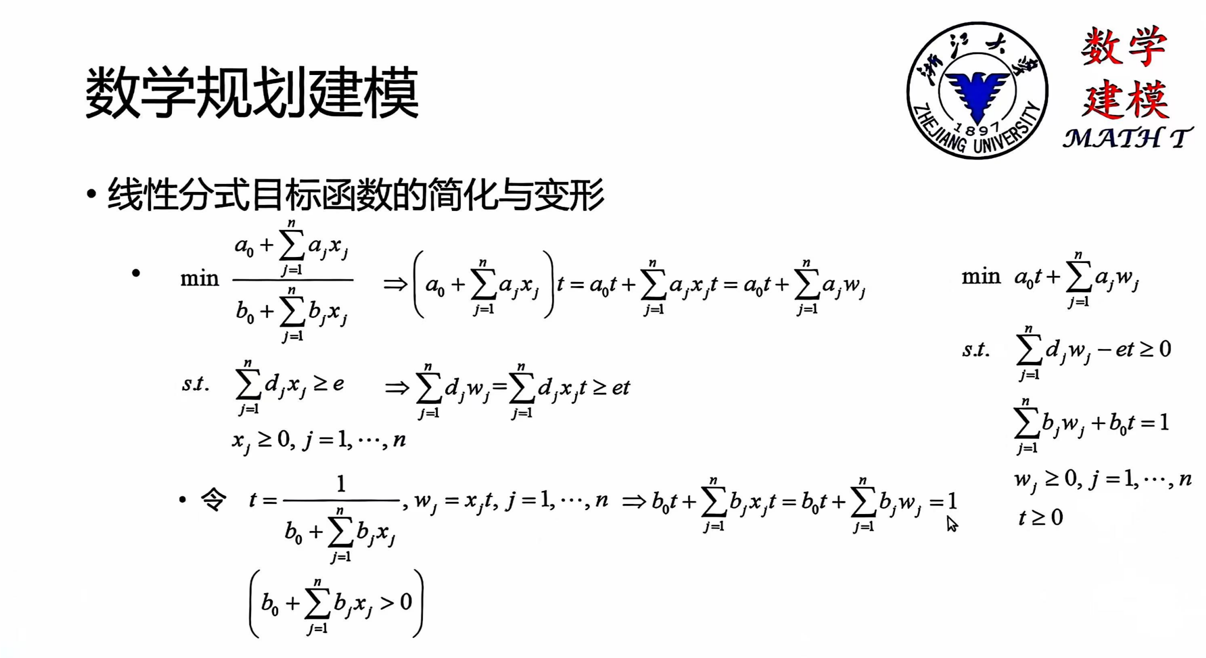
画图易得，松弛问题的最优解为 $(x_1, x_2) = \left(\frac{9}{4}, \frac{15}{4}\right)$，目标函数值为 -202.5。

由于最优解不是整数解，我们将松弛问题的可行域分为两个子可行域：



遍历所有分支，重复以上过程。所有分支均被删去之后，(IP) 的最优解包含在历次求得的整数解中

3.9 线性分式目标函数的简化与变形



3.10 0-1 变量的应用

分段函数

$$f(x)=\begin{cases}0+bx, & x=0\\ a+bx, & x\ge 1\end{cases}$$

定义辅助变量

$$y=\begin{cases}0, & x=0\\ 1, & x\ge 1\end{cases}$$

即 $x\ge 1 \iff y=1$。

则:

$$f(x)=F(x,y)=ay+bx$$

并且有以下约束条件:

$$x\ge y$$

$$y=1 \implies x\ge 1$$

$$My \geq x$$

其中 M 为一个很大的数。在许多实际问题中，决策变量存在明确的上界或下界。

逻辑关系

1. 若条件 1 成立，则条件 2 成立：

$$y_i = \begin{cases} 1, & \text{条件 1 成立} \\ 0, & \text{其他} \end{cases}$$

   且要满足 $y_2 \geq y_1$

2. 条件 1 和条件 2 至少有一个成立：

$$y_1 + y_2 \geq 1$$

3. 条件 1 和条件 2 至少有一个成立时，条件 3, 4, 5 中至少有一个成立：

$$z = \begin{cases} 1, & \text{条件 1, 2 中至少一个成立} \\ 0, & \text{其他} \end{cases}$$

3-4 段分段函数如何解决呢？————可能考试会考!!!

09 **考试考过**

布尔运算符

通常，指示变量 $z \in \{0,1\}$ 表示布尔值（真/假）。布尔运算符可以通过线性不等式来实现。表中假设所有变量均为二元变量。

布尔运算	线性表示
$z = x$ 或y	$x \leq z, y \leq z, z \leq x + y$
$z = x$ 且y	$x \geq z, y \geq z, z \geq x + y - 1$
$z =$ 非x	$z = 1 - x$
$x \implies y$	$x \leq y$
最多一个 $z_1, \ldots, z_n$ 为 1	$\sum_i z_i \leq 1$
恰好一个 $z_1, \ldots, z_n$ 为 1	$\sum_i z_i = 1$
至少一个 $z_1, \ldots, z_n$ 为 1	$\sum_i z_i \geq 1$
最多 k 个 $z_1, \ldots, z_n$ 为 1	$\sum_i z_i \leq k$

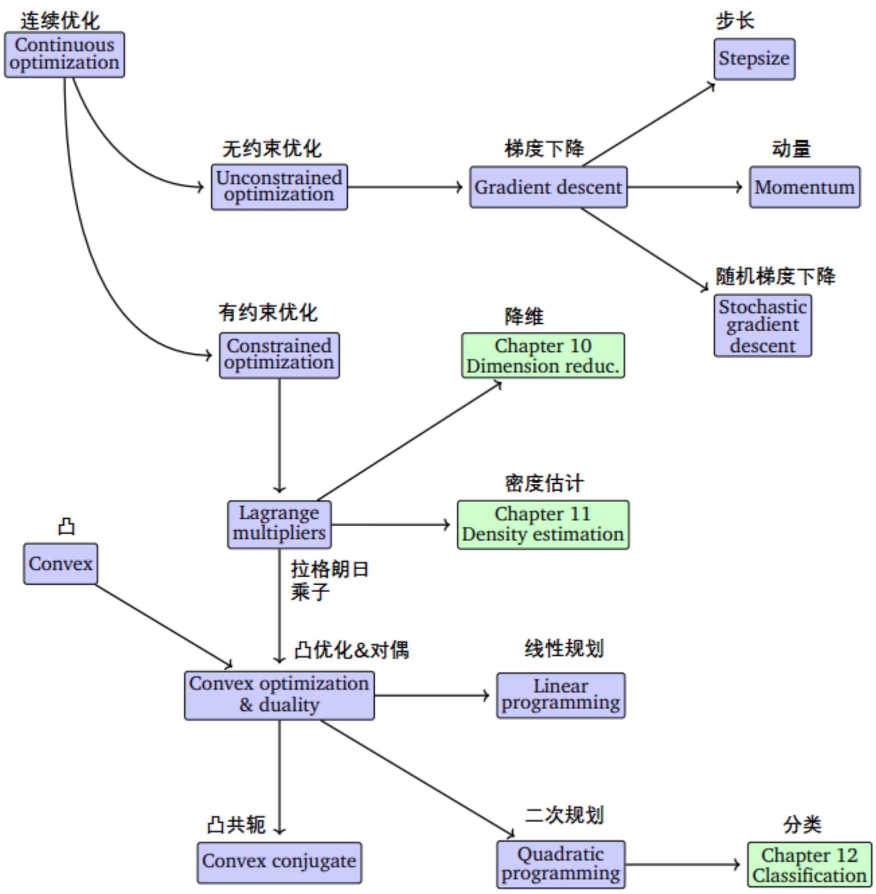
表 1: 布尔运算符及其线性不等式表示

3.11 组合优化

3.11.1 组合优化的基本概念

组合优化是应用于离散对象的，从有限多个可行解中找出使某个目标函数达到最优的解的优化问题

组合优化是离散数学（Discrete Mathematics）与最优化的交叉学科分支



3.11.2 组合优化与连续优化

背包问题

$$
\begin{aligned}
\max\ & \sum_{j=1}^{n} p_j x_j \\
s.t.\ & \sum_{j=1}^{n} w_j x_j \le C \\
& x_j = 0,1,\ j = 1,\cdots,n \\
& \frac{p_1}{w_1} \ge \frac{p_2}{w_2} \ge \cdots \ge \frac{p_n}{w_n}
\end{aligned}
$$

$$
\begin{aligned}
& x_j^* = 1,\ j = 1,2,\cdots,l-1, \\
& x_l^* = \frac{1}{w_l}\left(C - \sum_{j=1}^{l-1} w_j\right), \\
& x_j^* = 0,\ j = l+1,\cdots,n
\end{aligned}
$$

$$
l = \min\left\{ k \,\middle|\, \sum_{i=1}^{k} w_i > C \right\}
$$

图 9: 规划公式

对于离散背包问题，
现有 n 件物品，物品 j 的价值为 p_j，大小为 w_j。物品不可切割，每个物品只能被选择一次。目标是选择物品的一个子集，使得总价值最大化，且总大小不超过背包容量 W。

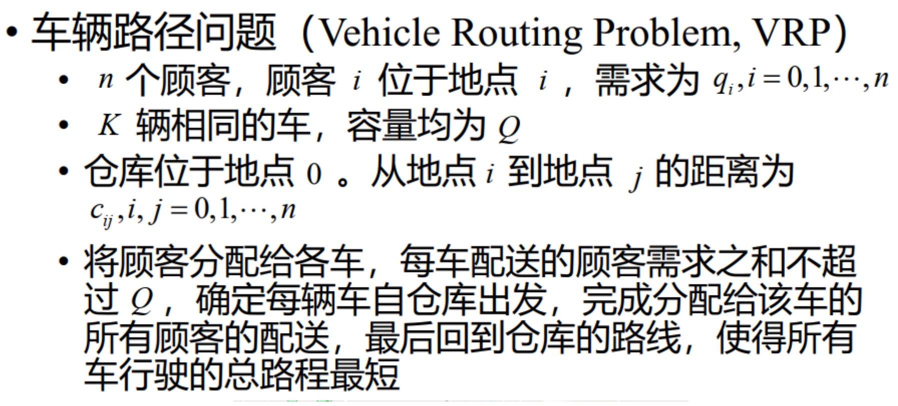
TSP 问题（旅行商问题）

- 一推销员想在若干个城市中推销自己的产品。计划从某个城市出发，经过每个城市恰好一次，最后回到出发的城市
- 城市之间距离已知
- 如何选择环游路线，使推销员走的路程最短

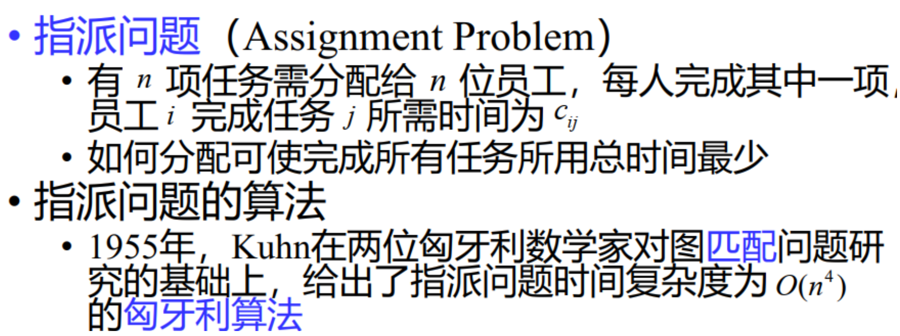
可行解（环游）

每一条环游路线由 n 段两个城市之间的旅行路线连接而成，对应于 1,2,...,n 的一个圆周排列

车辆路径问题



指派问题



3.12 算法与计算复杂性

3.12.1 $\mathcal{P}$ vs.$\mathcal{NP}$ 问题

$\mathcal{P}$ 和 $\mathcal{NP}$ 问题的通俗解释：

- $\mathcal{P}$：有（确定性）多项式时间算法的问题
- $\mathcal{NP}$：有非确定性多项式时间算法的问题

确定性算法是一种特殊的非确定性算法，故 $P \subseteq NP$

确定性算法： 设 A 是求解问题 B 的一个解算法，在算法的整个执行过程中，每一步都能得到一个确定的解，这样的算法就是确定性算法。

非确定性算法： 设 A 是求解问题 B 的一个解算法，它将问题分解成两部分，分别为猜测阶段和验证阶段，其中：

1. **猜测阶段**：在这个阶段，对问题的一个特定的输入实例产生一个任意字符串 y，在算法的每一次运行时，y 的值可能不同，因此，猜测以一种非确定的形式工作。

2. **验证阶段**：在这个阶段，用一个确定性算法（有时间限制）验证：

   (a) 检查在猜测阶段产生的 y 是否是合适的形式，如果不是，则算法停止并得到 **no**；

   (b) 如果 y 是合适的形式，则验证它是否是问题的解，如果是，则算法停止并得到 **yes**，否则算法停止并得到 **no**。它是验证所猜测的解的正确性。

P 类问题：能在多项式时间内可解的问题。

NP 类问题：在多项式时间内“可验证”的问题。也就是说，不能判定这个问题到底有没有解，而是猜出一个解来在多项式时间内证明这个解是否正确。即该问题的猜测过程是不确定的，而对其某一个解的验证则能够在多项式时间内完成。P 类问题属于 NP 问题，但 NP 类问题不一定属于 P 类问题。

$\mathcal{P}$ vs. $\mathcal{NP}$ 问题是指 $\mathcal{P}=\mathcal{NP}$ 和 $\mathcal{P}\subset\mathcal{NP}$ 中何者成立

3.12.2 $\mathcal{NP}$-完全与 $\mathcal{NP}$-难问题

- $\mathcal{NP}-\mathcal{C}$：$\mathcal{NP}$ 类中最难的问题
  - 若一个 $\mathcal{NP}-\mathcal{C}$ 问题有多项式时间算法，所有 $\mathcal{NP}$ 问题都有多项式时间算法。
- $\mathcal{NP}-\text{hard}$：不比 $\mathcal{NP}-\mathcal{C}$ 问题容易的问题

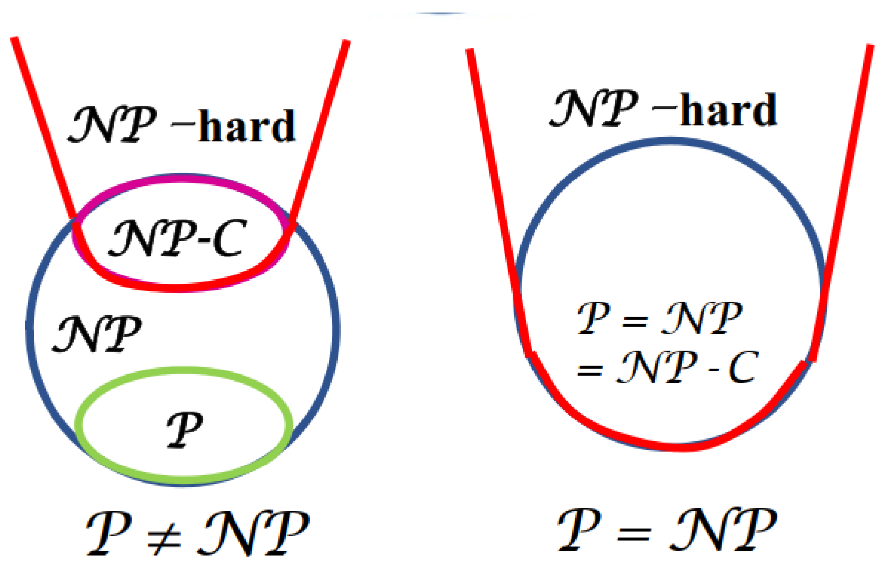
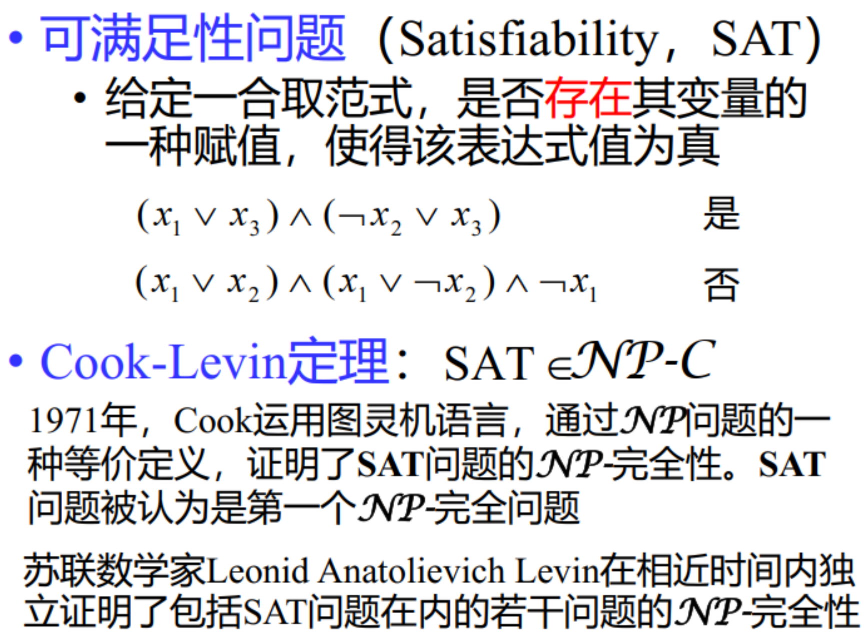


图 10: $\mathcal{NP}$–*hard* 可以是 $\mathcal{NP}$ 问题，也可以不是 $\mathcal{NP}$ 问题

概念辨析

- $\mathcal{NP}$ 问题没有多项式时间算法的问题 (×)
- 背包问题是一个 $\mathcal{NP}$ 问题，因此难解 (×)(错误原因在于没有因果关系)

3.12.3 SAT 问题（Satisfiability Problem）



4 图论模型

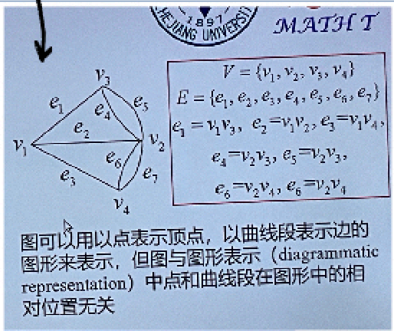
24 图论模型

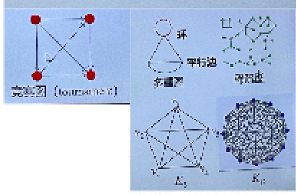
图的基本概念

图

图是一个有序二元组 $G=(V,E)$，其中 V 是**顶点（vertex）集合**，E 是**边（edge）的集合**。E **中每条边** e **与** V **中两个顶点关联（incident）**。

无向图是一种特殊的有向图





- 若与边关联的两个顶点有序，则称图为**有向图（digraph）**，否则称为**无向图**

度：无向图 G 中与顶点 v 关联的边的数目称为 v 的度，记为 $d(v)(deg_G(v))$

- 图的所有顶点的度的最大值与最小值分别称为最大度和最小度，记为 $\Delta(G)$ 和 $\delta(G)$
- （握手定理）所有顶点的度之和等于边数的两倍，即 $\sum_{v\in V}d(v)=2|E|$

简单图

A B 平行边

两端点相同的边称为**环（loop）**，两端点分别相同的多条边称为**平行边**（parallel edges）

既没有环，也没有平行边的图称为**简单图（simple graph）**。不是简单图的图称为**多重图（multigraph）**

完全图

任何两个不同顶点之间都有边相连的简单图称为**完全图（complete graph）**

- n 个顶点的完全图记为 K_n，K_n的边数为 $\frac{n(n-1)}{2}$

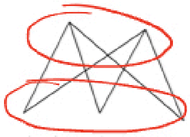
简单图的顶点数与边数

- 若 $G=(V,E)$ 为简单图，则边数的上界为 $\frac{|V|(|V|-1)}{2}$，下界为 $|V|-1$
- 边数接近上界的称为**稠密图（dense graph）**，边数远离上界的称为**稀疏图（sparse graph）**

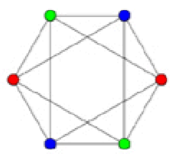
二部图与连通

若是二部图，

- 二部图
  - 若图的顶点集可以划分为两个非空集合 X 和 Y，使得图中任一条边的两个端点分属 X,Y 两个集合，则称该图为二部图（bipartite graph），记为 $G=(X\cup Y,E)$
    - X 中所有顶点与 Y 中所有顶点都有边相连的二部图称为完全二部图
  - G 是二部图当且仅当 G 中不存在奇圈 → G里不存在长度为奇数的圈。
- 连通
  - 若无向图中两顶点 u,v 之间有路相连，则称 u,v 连通（connected）
  - 无向图中任意两顶点均连通的图称为连通图（connect graph）



完全二部图 $K_{2,3}$



完全3部图 $K_{2,2,2}$

奇圈的找法：深度优先搜索。

子图

- 子图
  - 图 $G'=(V',E')$ 称为图 $G=(V,E)$ 的一个子图（subgraph），若 $V'\subseteq V$，$E'\subseteq E$，且 G 中边的关联关系在 G' 中保持不变
    - 生成子图：$V'=V$
    - 导出子图：$G(V')$，$G\setminus V'$，$G(E')$，$G\setminus E'$

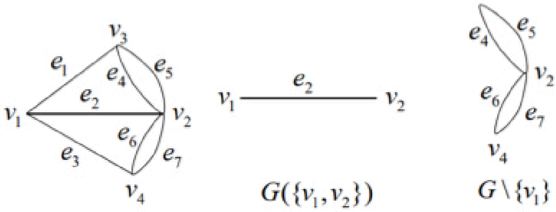


图 1

路①途径：可以重复走

顶点和边交替出现的序列 $v_{i_0}e_{i_1}v_{i_1}e_{i_2}\cdots e_{i_k}v_{i_k}$，且序列中与每条边相邻的两个顶点为该边的两个端点，称为连接顶点 v_{i_0} 和 v_{i_k} 的**途径（walk）**

- 若图为简单图，可省略途径中边的符号

1/7/24, 12:38 PM ALL-SUM

经过边互不相同的途径称为**迹**（trail）

经过的边不相同⇒经过的顶点可以相同

迹不一定是路，但路一定是迹

- 起点和终点相同的迹称为闭迹

经过顶点互不相同的途径称为**路**（path）

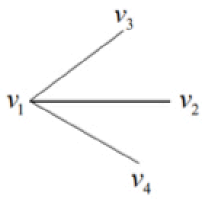
- 起点和终点相同，其余顶点互不相同，也不与起点和终点相同的途径称为**圈**（cycle）

路的长度

- 边赋权图中一条路所含边的权之和称为它的长度

树

- 树
  - 连通的无圈图称为树（tree）
  - 图 G 的为树的生成子图为称生成树（spanning tree）



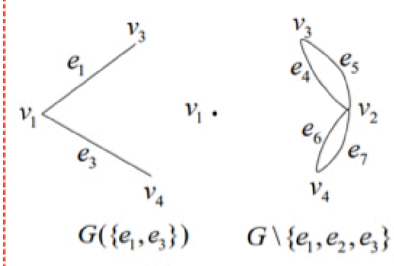


图 2

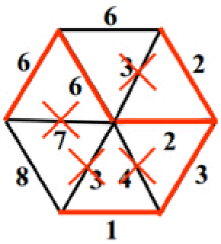
最小生成树

最小生成树（MST）是赋权图所有生成树中总权和最少的生成树

####### Kruskal 算法 {#kruskal-算法}

单调不增 ☆

- Kruskal 算法
  - 将连通图 $G=(V,E)$ 的所有边按权非降顺序排列
  - $F=\varnothing$，$j=1$
  - 若图 $T=(V,F\cup\{e_j\})$不含圈，则令 $F=F\cup\{e_j\}$
  - 若$|F|=|V|-1$，终止，$T=(V,F)$即为最小生成树。否则，返回上一步



最短路

Bellman–Ford算法：无负权圈图上某个顶点到其余顶点的最短路

Dijkstra算法：非负权图上某个顶点到其余顶点的最短路

- Dijkstra算法目前可在 $\Theta(|E|+|V|\log|V|)$时间内实现

Floyd–Warshall算法：无负权圈图上所有点对之间的最短路

最短连接

给定Euclidean平面上 n 个点，如何用总长度最短的若干条线段将它们连接起来？

5 latex 代码样例

定理 5.1. *asdfasdf*

引理 5.2. *adsf*

推论 5.3. *asdf*

命题 5.4. *adf*

定义 5.5. *adsf*

例 5.6. *asdfasfd*

证明. adsfasfd □

This is a colored box.

This is a colored box with a wider text.

这是一个简单的 fcolorbox 框。

asdfas[1]

Code Listing 1: Python example

```
def hello_world():
    print("Hello,world!")
```

参考文献

[1] John Smith and Jane Doe. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2019:1–13, 2019.