第一章

一、科学计算的基本概念

1. 科学计算：利用计算机完成…数学问题的计算
2. 科学研究：科学实验、理论研究、科学计算
3. 算法构造：了解、理解、选择、使用　基本手段：近似
　分析：分析、改进、创造、创新　核心问题：近似影响

二、误差分析方法

1. 分类 { 过失/疏忽误差 → 可避免
　非过失误差 → 模型、观测、截断、舍入
2. 绝对误差　$\varepsilon(x)=\overset{近}{x^*}-x$　误差限 $|\varepsilon(x)|=|x^*-x|\le\eta$
　相对……　$\varepsilon_r(x)=\dfrac{x^*-x}{\underset{精确}{x}}$　相对……　$|\varepsilon_r(x)|=\left|\dfrac{x^*-x}{x}\right|\le\eta_r$
　$\varepsilon(x)=x\cdot\varepsilon_r(x)$
　实际计算中取 $\varepsilon_r^*(x)=\dfrac{\varepsilon(x)}{x^*}=\dfrac{x^*-x}{x^*}$ 近似
3. 有效数字　$x^*=0.a_1a_2\cdots a_n\times10^m$
　x^*为n位数字的有效数　$|x^*-x|\le0.5\times10^{m-n}$
4. 误差的传播
　算法是稳定的：误差不增大
　避免：大小相近的同号数相减、乘数的绝对值很大，除数接近于零。

三、建模与仿真的相似性理论及方法

1. 相似：几何相似、运动相似、动力相似
　L 空间　U 时间　F 受力
　$k=\dfrac{M}{P}$.　$k_F=k_\rho k_L^2\cdot k_U^2$

第二章　数学建模方法及一般选取原则

一、概述

1. 分类 { 传统：机理分析(理论研究)、试验统计(科学实验)
　现代：层次分析、定性推理、系统辨识.
2. 原则
　确定：机理分析、直接相似、量纲、图解
　不确定：蒙特卡罗.
　社会问题：层次分析

二、层次分析法

1. 基本步骤
　(1) 建立层次结构模型
　(2) 构造成对比较矩阵
　(3) 层次单排序及一致性检验
　(4) 层次总排序及一致性检验
　(5) 计算得分并排序

$CI=\dfrac{\lambda_{max}-n}{n-1}$ 一致性指标
查RI 平均随机一致性指标
$CR=\dfrac{CI}{RI}<0.1$ 一致性比例

矩阵越一致，λ_{max} 接近 n	越小 $\lambda_{max}-n$

Z 目标层
A 准则层 ×6　1个 6×6
B 方案层 ×3　6个 3×3

一致矩阵直接归一
不一致归一 { 算术平均、几何平均
　特征值

2. CR>0.1 修正方法：往一致矩阵(各行列倍数关系)上修正
3. 优点：系统性、实用性、简洁性.
　缺点：局限性.

三、量纲分析法

1. 数&量　物理书中所有公式都是数的公式
　100 cm　物理量数值有量纲，无量纲量可以有单位(角度有单位)
2. 单位制 { 基本量类：$\tilde{l}$, $\tilde{m}$, $\tilde{t}$　导出量类：$\tilde{F}$
　基本单位：cm. g. s　导出单位：N
　$[f]=\dim f=LMT^{-2}=(\dim l)(\dim m)(\dim t)^{-2}$
3. 定理 { $C=A^\alpha B^\beta \Rightarrow \dim C=(\dim A)^\alpha(\dim B)^\beta$
　$C=A+B+\cdots \Rightarrow \dim C=\dim A=\dim B$
4. 量纲独立于单位
5. 瑞利法．设 $t=\lambda m^a l^b g^c$. $(\dim t)=(\dim m)^a(\dim l)^{b+c}(\dim$ …
6. π定理　设 $f(t,m,l,g)=0$.　$t^{y_1}m^{y_2}l^{y_3}g^{y_4}=\pi$
　$\begin{bmatrix}1&0&0&-2\\0&1&0&0\\0&0&1&1\end{bmatrix}\begin{bmatrix}y_1\\y_2\\y_3\\y_4\end{bmatrix}=0$　$\therefore y=\begin{bmatrix}2\\0\\-1\\1\end{bmatrix}$　$t^2m^0l^{-1}g^1=\pi$
　$t=\sqrt{\pi l/g}$
7. 优点：不用物理知识、去掉无关物理量.
　缺点：可能忽略无量纲量．用不了 sin. exp.

第三章　线性规划

一、线性规划及其概念

1. 思路 { 找出决策变量
　写出目标函数
　确定约束条件
2. 标准型 { 目标函数最大
　约束条件等式
　决策变量非负　注意无限制变量 $x_3=x_3'-x_3''$
　资源限量非负
2. 图解法
3. 基本可行解　$x=\begin{bmatrix}B^{-1}b\\0\end{bmatrix}$ 为基本解．$B^{-1}b\ge0\Rightarrow$ 基本可行解

$\max z=\underline{x_1+3x_2}$
　↳ 先画 $x_1=3x_2$ 等值线

4. 单纯形法

$A=[B\ N]\quad X^T=[X_B^T\ X_N^T]\quad C=[C_B\ C_N]$

$Z=CX=[C_B\ C_N]\begin{bmatrix}X_B\\X_N\end{bmatrix}=C_BX_B+C_NX_N$

$b=AX=[B\ N]\begin{bmatrix}X_B\\X_N\end{bmatrix}=BX_B+NX_N\Rightarrow(b=BX_B)$

$X_B=B^{-1}(b-NX_N)=B^{-1}b-B^{-1}NX_N$

$Z=C_BX_B+C_NX_N=C_B(B^{-1}b-B^{-1}NX_N)+C_NX_N$

$=C_BB^{-1}b+(C_N-C_BB^{-1}N)X_N\Rightarrow(Z=C_BB^{-1}b)$

步骤:

(1) 化标准型

(2) 找出可构造单位矩阵作为可行基. 建立单纯形表.

(3) 计算 $\boxed{\sigma_j=C_j-C_BP_j}$ 若 σ_j 所有 ≤ 0 则结束.

(4) 若 $\sigma_j>0$ 且 A_k 整 ≤ 0 则问题为无界解. 结束

(5) $\max\{\sigma_j\mid\sigma_j>0\}=\sigma_k$. X_k 换入变量

$\min\{b_i/a_{ik}\mid a_{ik}>0\}=b_l/a_{lk}$ 有限值最优 换出变量 X_{Bl}

(6) 迭代. 转换基变量

MATLAB ☆

$\max Z=4X_1+3X_2$

$$s.t.\begin{cases}2X_1+X_2\le 10\\X_1+X_2\le 8\\X_2\le 7\\X_1,X_2\ge 0\end{cases}\Rightarrow \min(-Z)=-4X_1-3X_2\quad f=[-4;-3]\quad A=\begin{bmatrix}2&1\\1&1\\0&1\end{bmatrix}\ b=\begin{bmatrix}10\\8\\7\end{bmatrix}$$

求解器:

```
clc, clear
c=[4;3]; b=[10;8;7]; a=[2,1;1,1;0,1];
lb=zeros(2,1);
[x,fval]=linprog(c,a,b,[],[],lb,[]);    % Aeq beq
y=-fval
```

基于问题: 选max → 进基变量

```
clc.clear.
prob=optimproblem('ObjectiveSense','max')
c=[4;3]; b=[10;8;7] a=[2,1;1,1;0,1] lb=zero(2,1);
x=optimvar('x',2,'LowerBound',0);      → 整数规划 x=('x',2,'Type','integer',...)
prob.Objective=c'*x;
prob.Constraints.con=a*x<=b;
[sol,fval,flage,out]=solve(prob)       % sol.x, y
```

5. 可以转化为线性规划的问题

① $|X_i|\rightarrow u_i+v_i\quad x_i\rightarrow u_i-v_i$

② $\min(\max(x_i-y_i))$ 令 $v_i=\max(x_i-y_i)$

第四章 整数规划

一. 整数规划问题

1. 背包问题 0-1

$\max Z=\sum_{i}^{n}C_iX_i$

$$s.t.\begin{cases}\sum_{i}^{n}a_iX_i\le b\\X_i=0\text{ or }1,\ i=1\cdots n\end{cases}$$

2. 旅行商问题 证明过程

$\min Z=\sum_i\sum_j C_{ij}X_{ij}$

$$s.t.\begin{cases}\sum_{i}^{n}X_{ij}=\sum_{j}X_{ij}=1\\u_i-u_j+nX_{ij}\le n-1.\\u_1=0.\ \boxed{1\le u_i\le n-1}\\X_{ij}=0\text{ or }1\end{cases}$$

3. 非线性约束条件的线性化

(1) 相互排斥的约束条件

车: $5X_1+4X_2\le 24$

船: $7X_1+3X_2\le 45$

$$\Rightarrow\begin{cases}5X_1+4X_2\le 24+yM\\7X_1+3X_2\le 45+(1-y)M\\y=0\text{ or }1\end{cases}$$

$\begin{cases}0-yM\le X_1\le 0+yM\\500-(1-y)M\le X_1\le 800+(1-y)M\end{cases}$

$X_1=0$ or $500\le X_1\le 800\Rightarrow 500y\le X_1\le 800y.\ y=0\text{ or }1$

二. 整数规划模型求解

1. MATLAB　　求 $\max(x+y)$. 随机生成x和y. 比较.

2. 蒙特卡罗法.

根据x的整数取值范围随机计算N个点

第五章 图与网络规划

一. 图的基本概念

两端一样

1. 有向图. 无向图. 弧(带方向). 环. 重边/平行边

2. 简单图(无环无重边). 完全图(任意两点均有边). 赋权图(网络)

3. 奇顶点(度为奇数的顶点) Σ奇顶点个数必为偶数

4. $\sum d(v)=2|E|$ 顶点度数之和是边数的2倍.

5. 子图若顶点数与母图相等 → 生成子图.

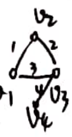
6. 迹: 各边相异　回路: 起点和终点重合

轨道: 各点相异　圈: 起终点重合+轨道

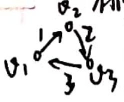
7. 连通图: 无孤立点. 非连通图中的孤立点→连通分支

强连通图: 双向连通

8. 关联矩阵



无向:
$$\begin{array}{c|cccc} & e_1 & e_2 & e_3 & e_4\\\hline v_1 & 1 & & 1 & \\ v_2 & 1 & 1 & & \\ v_3 & & 1 & 1 & 1\\ v_4 & & & & 1\end{array}$$



有向:
$$\begin{array}{c|ccc} & e_1 & e_2 & e_3\\\hline v_1 & 1 & & -1\\ v_2 & -1 & 1 & \\ v_3 & & -1 & 1\end{array}\quad\begin{cases}1\ \text{出}\\-1\rightarrow\text{入}\end{cases}$$

9. 邻接矩阵

无向 $\begin{cases}1\rightarrow\text{相邻}\\0\rightarrow\text{不相邻}\end{cases}$

有向 $\begin{cases}1\rightarrow v_i\rightarrow v_j\\0\rightarrow\text{无弧}\end{cases}$

赋权: 两点间权值

二. 最短路问题

1. Dijkstra (按自己的格式列表) $\begin{cases}\text{cost}\\\text{former}\end{cases}$

~~2. 最小支撑树问题~~

三.最小支撑树问题

1. 树的基本概念和算法

(1) 树:连通的无圈图

(2) 等价命题: 一个图的生成树通常不唯一

(i) 图G是树

(ii) 任意两个不同顶点之间存在唯一的路.

(iii) 连通,删除任一条边均不连通

(iv) 连通,且顶点数=边数+1

(v) 无圈,加一条边可得唯一的圈

(vi) 无圈,且点=边+1

2. 求解最小生成(支撑树)

(1) Kruskal 避圈 → 最小边排序 已有边集距其他点距离

(2) Prim 破圈 像dijkstra.

第六章 动态规划 解决多阶段决策过程最优化的一种方法

一.多阶段决策和动态规划

1. 多阶段决策过程:看作前后关联具有链状结构的多阶段过程. 状态 $\xrightarrow{}$ ①决策 $\xrightarrow{}$ 状态 $\xrightarrow{}$ ②决策 $\xrightarrow{}$ ……

2. 动态规划

(1) 阶段 k: 整个决策过程的自然划分.

(2) 状态 s_k: 每个阶段开始时所处的自然状况或客观条件. (无后效性) → 不会跨状态

(3) 决策 $u_k(s_k)$: 每一阶段状态确定后演变为哪个状态

(4) 策略 $\{u_k(s_k), u_{k+1}(s_{k+1}) \cdots\}$: 一组有序的决策.

(5) 状态转移方程: 第k阶段到第k+1阶段的状态转移规律.

(6) 指标函数和最优值函数

可分离性 动态递推关系

衡量的指标

↳ 采取最优值策略得到的指标函数值

二.动态规划的基本思路和步骤

1. 算法思想:分解. 状态转移. 每段最优从全局考虑.

2. Bellman最优化原理:一个最优策略的子策略最优.

$$f(B_1) = \min\begin{cases} d(B_2,C_1)+f_3(C_1) \\ d(B_2,C_2)+f_3(C_2) \\ d(B_2,C_3)+f_3(C_3) \end{cases} = \min\begin{cases} 2+1 \\ 3+3 \\ 1+4 \end{cases} = 3$$

3. 步骤(大题)

~~(1) 划分阶段k,依据时间、空间顺序.~~

~~(2) 正确选择状态变量 s_k.~~

~~(3) 确定决策变量 u_k 和~~

四.背包问题

1. 建模

$$\max f = \sum_{i}^{n} c_i x_i$$

$$\begin{cases} \sum_{i}^{n} w_i x_i \le a \\ x_i = \text{整数} \cdots \end{cases}$$

根据 s_k 选择 x_k.

(1) 状态划分: 带k种物品 → 第k个状态.

(2) 状态变量 s_k: 第k个阶段携带时,背包还可以携带的重量

(3) 决策变量 x_k: 第k个阶段携带第k种物品的件数

(4) 决策允许集合 $D_k(s_k) = \{x_k \mid 0 \le x_k \le s_k/w_k, x_k\text{为整数}\}$.

(5) 状态转移方程: $s_{k+1} = s_k - w_k x_k$

(6) 阶段指标 $v_k(s_k, x_k)$: 得到的价值 $v_k(s_k, x_k) = c_k x_k$

(7) 递推方程 $f_k(s_k) = \max\limits_{0 \le x_k \le \frac{s_k}{w_k}} \{v_k(s_k, x_k) + f_{k-1}(s_{k-1})\} = \{c_k x_k + f_{k-1}(s_k - w_k x_k)\}$

2. 过程明天看算法笔记再给吧

(以下来自北京大学算法课中提出的思路)

(1) 状态划分: 面对背包容量 s_k 带k种物品.

(2) 状态变量: 第k个阶段背包的余量 s_k

(3) 决策变量: 当前阶段是否携带第k种物品

(4) …允许集合: $x_k = 0$ or 1

(5) 状态转移方程: $s_k = s_{k-1} - w_k$.

(6) 阶段指标 $f_k(s_k)$ 和递推方程: $\boxed{f_k(s_k) = \max\{f_{k-1}(s_k), f_k(s_k - w_k) + c_k\}}$

k \ s	0	1	2	3	4	5	6	7	8	9	10
1	0	0	0	4	4	4	8	8	8	12	12
2	0	0	0	4	5	5	8	9	10	12	13
3	0	0	0	4	5	6	8	9	10	12	13

←

k	1	2	3
w_k	3	4	5
c_k	4	5	6.

$\therefore \max f = f_3(10) = 1$

$= 2\times4 + 1\times5$

$x_1 = 2.\ x_2 = 1.\ x_3 = 0$

五.旅行商问题

1. 建模.

$N_i = \{2, 3, \cdots, i-1, i+1, \cdots, n\}$ 从1到i的中间城市集合.

S: 从1到i所经中间城市集合. → 1 ⤴ i 一个过程

阶段划分: S中点个数. 状态变量 (i, S): 从1出发到i. 经过S中一次.

最优指标函数 $f_k(i, S)$: $1 \xrightarrow{k} i$ 的最短. 最优决策函数 $P_k(i, S)$.

$$\boxed{f_k(i, S) = \min_{j \in S}\{f_{k-1}(j, S\setminus\{j\}) + d_{ji}\}}$$

$f_0(i, \phi) = d_{1i}$.

↳ 记录最短的 $P_k(i, S) = j$.

j \ i	1	2	3	4
1	0	8	5	6
2	6	0	8	5
3	7	9	0	5
4	9	7	8	0

$$\min\ d_{ij} \cdot x_{ij}.$$

$$\text{s.t.}\begin{cases} \sum_{j}^{n} x_{ij} = \sum_{i}^{n} x_{ij} = 1 \\ u_i - u_j + n \cdot x_{ij} \le n-1 \\ u_1 = 0.\ 1 \le u_i \le n-1 \\ x_{ij} = 0 \text{ or } 1 \end{cases} \quad \begin{cases} j = 2, \cdots, n. \\ i = 1, \cdots n. \end{cases}$$

第7章 算法复杂性概述

启发 算法

一. 算法的基本概念

1. 算法：若干指令的有穷序列.
   性质：输入.输出.确定.有限
   以须明确量和时间
2. {算法：…指令有穷序列 / 程序：算法用某种程序语言的具体实现.} 所以不满足有限性
3. 分类：精确.启发式.近似.随机.
4. 评估算法的执行效率：经验(程序测)/理论
   (1) 经验法
   问题：依赖计算机.语言与编程.耗时.
   只能评估部分实例
   vs 理论法：不依赖x2. 节省时间. 任何实例可研究
   (2) 理论法
   评估时间.空间.

二. 算法的复杂性简述

1. $C=F(N,I,A) \rightarrow \begin{cases} T=T(N,I) \text{时间} \\ S=S(N,I) \text{空间} \end{cases}$
   复杂性 规模 输入 算法(隐含)
2. 下 $f(n)=\Omega(g(n))$ $f(n)\geq c\cdot g(n)$ $a\geq b$
   等 $f(n)=\theta(g(n))$ $f(n)=O(g(n))\ \&\ f(n)=\Omega(g(n))$
   上 $f(n)=O(g(n))$ $f(n)\leq c\cdot g(n)$ $a\leq b$
   $f(n)=o(g(n))$ $a<b$
   $f(n)=\omega(g(n))$ $a>b$
3. 复杂性渐近性态. 渐近表达式→略去低阶留下的主项
4. 大O表示法.

三. P类问题与NP类问题

1. 问题π 算法$O(n^k)$ → 存在多项式时间算法. (不能有指数)
2. P类问题
   (1) 确定性算法：每步只有一个确定的选择.
   (2) P类问题：$O(n^k)$时间运行确定性算法得到yes/no(确定的)答案.
   (3) 特点：重复运行结果一样
3. NP类问题
   (1) NP: $O(n^k)$运行一个非确定算法得到yes/no(确定的)答案.
   (2) 非确定性算法包括猜测阶段和验证阶段
   (3) NP: 存在一个确定性算法. 多项式时间检查验证猜测答案.

P NP

NP不是非P

第八章 优化算法复杂度分析 明天看. 不太看得懂

一. 局部谕O

1. ZO zero order oracle 零导数优化算法. $x_0\rightarrow f(x_0)$.
2. FO first order oracle 梯度法. $f(x_0)$及一阶导 $\partial f(x_0)/\nabla f(x_0)$
3. 2nd O second… 牛顿法. $f(x_0)$. 一阶导$\nabla f(x_0)$. 二阶导$\nabla^2 f(x_0)$
~~4. SFO stochastic first oo 随机梯度法~~
4. LO linear optimization oracle 条件梯度法. 线性规划的解 $\arg\min\langle x_0,x\rangle$

二. 停止算法.

1. 凸函数. 所求局部最优解就是全局最优解 f^*
   条件：① $f(x_k)-f^*\leq\epsilon$ (局部极小值) ② $\frac{f(x_k)-f^*}{f(x_k)}\leq\delta$ ③ $\|\nabla f(x_k)\|\leq\epsilon$ 导数=0.
   ④ $\|x_k-x^*\|\leq\epsilon$.
2. 非凸函数. 局部极小值 ≠ 全局最优解. Lipschitz → 利普希兹 比通常连续更强的光滑性条件
   ↳解梯度的范数与最优解误差.

三. 各种梯度

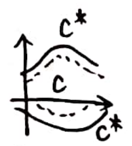
1. 条件梯度 $x_{k+1}=\arg\min\langle\nabla f(x_k),x\rangle$ 多面体x 通过求解线性子问题得到下一步
   子迭代复杂度低. 简单(不依赖步长). 解具有结构性.
2. 梯度法 $x_{k+1}=x_k-a\cdot\nabla f(x_k)$ (光滑).
3. 梯度投影法 {非光滑→次梯度g_k替代导数 / 有约束→可能落在约束集合外的点投影回来}
   L2范数 $x_{k+1}=\arg\min\frac{1}{2}\|x_k(x_k-\gamma_k g(x_k))\|^2$ $w(y)-[w(x)+\langle\nabla w(x),y-x\rangle]$
4. 镜像梯度法. $x_{k+1}=\arg\min\gamma_k\langle g(x_k),x\rangle+V(x_k,x)$
   投影L2范数→镜像欧几里德结构.
5. 近似点梯度法. 针对凸复合函数(非) $\min[g(x)+h(x)]$ h=0. 等于梯度法
   将x近似为欧式距离平方+约束f最小的u.
   把h用约束函数表示. 优化g之后投影到约束中.

第九章 近似算法

近似. 输入转换 A→B ↓问题求解 $a^*\leftarrow b^*$ 输出转换

一. 问题变换

1. 复杂性归约
   $\begin{cases} T(n)-O(t(n)) \\ T(n)+O(t(n)) \end{cases}$ $\xrightarrow{t(n)}\pi^{T(n)}$ 下界：$T(n)-O(t(n))$ 上界：$T(n)+O(t(n))$ 同阶时一样



二. NP完全问题

三. 近似算法性能

1. …性能比$\rho(n)$. $\max\left\{\frac{c}{c^*},\frac{c^*}{c}\right\}\leq\rho(n)$. 近似 原问题最优值 $\begin{cases} \text{最小} \frac{c}{c^*} & c\geq c^* \\ \text{最大} \frac{c^*}{c} & c^*\geq c \end{cases}$

四. 代表性近似算法

四. 代表性近似算法

1. 顶点覆盖问题

边集E1. 当边都没删完时：从E1中任取一条边加入cset 并删掉相关边.

→并非最优. 只是近似.

2. 旅行商问题 贪心策略. (1) →最近邻法.

每步选最近邻居.

(2) Prim

① 先生成最小生成树. ← 用kruskal算法

② 前序遍历(根左右) 确定顺序. 相连成哈密顿回路.

(3) kruskal避圈法→最近插入法.