当存在两个并行的队列，一起组成系统时两个MM1

排队系统中的平均顾客数
the average number in the queue system

两个MM1：L1+L2

MM2：L

每个服务器空闲的时间比例（平均）
the proportion of time each server is idle (on average)

两个MM1：1/2(Π0+Π0)

MM2：（Π0+1/2 Π1）

我们看到了将单服务器队列合并为双服务器队列的优势。

第二个系统的 L 显然比第一个系统的 (L1+L2) 要小得多。

第二个系统中的服务器平均没有更多的工作
（没有额外的顾客需要服务），
但我们避免了一个服务器前有顾客排队，
而另一个服务器空闲的情况。

这减少了系统中的平均等待时间(W)。

Appendix A

M/M/1 Queue with Arrival Rate λ and Service Rate μ:

马尔可夫到达间隔时间
泊松过程
到达间隔时间 λ

马尔可夫服务时间
指数分布
平均服务率 μ
状态转移率 μ

单个服务器

利用率
$$\rho = \frac{\lambda}{\mu}$$

空的概率（the proportion of time each server is idle
$$\pi_0 = 1-\rho$$

k个顾客的概率
$$\pi_k = \rho^k(1-\rho), \qquad k \geq 1$$

系统中的平均顾客数【带系统的都是L】
(the average number in the queue system
$$L = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}$$

排队的平均顾客数
$$Q = \frac{\rho^2}{1-\rho} = \frac{\lambda^2}{\mu(\mu-\lambda)}$$

顾客在系统的平均时间
$$W = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu-\lambda}$$

排队的平均等待时间
$$D = D = W - \frac{1}{\mu} = \frac{\lambda}{\mu(\mu-\lambda)}$$

M/M/1/N Queue with Arrival Rate λ and Service Rate μ:

最多可以容纳（N -1）个客户满员不给进

单人理发店：
一把理发椅
和两把供等待的顾客使用的椅子
N=1+2=3

$$\rho = \frac{\lambda}{\mu}$$

$$\pi_0 = \left(\sum_{k=0}^{N}\rho^k\right)^{-1} = \frac{1-\rho}{1-\rho^{N+1}}$$

$$\pi_k = \rho^k\pi_0 = \frac{\rho^k(1-\rho)}{1-\rho^{N+1}}, \qquad 0 \leq k \leq N$$

$$L = \frac{\rho[1-\rho^N - N\rho^N(1-\rho)]}{(1-\rho)(1-\rho^{N+1})}$$

$$Q = \frac{\rho^2[1-\rho^N - N\rho^{N-1}(1-\rho)]}{(1-\rho)(1-\rho^{N+1})}$$

$$W = \frac{1-\rho^N - N\rho^N(1-\rho)}{\mu(1-\rho)(1-\rho^{N+1})}$$

$$D = \frac{\rho[1-\rho^N - N\rho^{N-1}(1-\rho)]}{\mu(1-\rho)(1-\rho^{N+1})}$$

M/M/m Queue with Arrival Rate λ and Service Rate μ: 银行叫号，m个柜台

$$
\begin{aligned}
\rho &= \frac{\lambda}{m\mu} \\
\pi_0 &= \left[\frac{(m\rho)^m}{m!(1-\rho)} + \sum_{k=0}^{m-1}\frac{(m\rho)^k}{k!}\right]^{-1} \\
\pi_k &= \pi_0 \begin{cases} \dfrac{(m\rho)^k}{k!}, & 0 \le k \le m-1 \\[2ex] \dfrac{m^m\rho^k}{m!}, & k \ge m \end{cases} \\
L &= \frac{\rho(m\rho)^m\pi_0}{m!(1-\rho)^2} + \frac{\lambda}{\mu} \\
Q &= \sum_{k=m}^{\infty}(k-m)\pi_k = \frac{\rho(m\rho)^m\pi_0}{m!(1-\rho)^2} \\
W &= \frac{L}{\lambda} = \frac{\rho(m\rho)^m\pi_0}{m!\lambda(1-\rho)^2} + \frac{1}{\mu} \\
D &= W - \frac{1}{\mu} = \frac{\rho(m\rho)^m\pi_0}{m!\lambda(1-\rho)^2}
\end{aligned}
$$

是如果k小于等于m-1，
就用Π0乘以上面这一坨，
如果k大于等于m，
就用Π0乘以下面这一坨

系统中的平均顾客数
【相比m个 MM1，减少】

M^b/M/1 Queue with Arrival Rate λ and Service Rate μ: 团购，一次来b个人
只能检票口就一个

$$
\begin{aligned}
\rho &= \frac{b\lambda}{\mu} \\
\pi_0 &= 1-\rho \\
\pi_k &= \begin{cases} \left(\dfrac{\lambda+\mu}{\mu}\right)^{k-1}\dfrac{\lambda}{\mu}\pi_0 & 1 \le k \le b \\[2ex] \dfrac{\lambda+\mu}{\mu}\pi_{k-1} - \dfrac{\lambda}{\mu}\pi_{k-b-1} & k \ge b+1 \end{cases} \\
L &= \frac{\rho(1+b)}{2(1-\rho)} \\
Q &= L - \rho = \frac{\rho(b-1+2\rho)}{2(1-\rho)} \\
W &= \frac{L}{\lambda b} = \frac{1+b}{2\mu(1-\rho)} \\
D &= W - \frac{1}{\mu} = \frac{b+2\rho-1}{2\mu(1-\rho)}
\end{aligned}
$$

SIMPLEX方法 3 3

1.写出目标函数 和 subject to

人工变量

小于：松弛变量

大于：剩余变量

等于和大于：人工变量

无界：两变量相减

右侧b负数：变号

2.最小值矩阵

X_0不包括surplus变量

	x^T	
x_0	A	B
	$C^T - C_0^T A$	$-C_0^T B$

两阶段法：
分两行
常数行
和M行

最大值矩阵

	x^T	
x_0	A	B
	$-C^T + C_0^T A$	$C_0^T B$

3.底部行，最大负数，工作列

4.工作列的，大于零的，系数，b÷系数，最小比率

5.主元素1，其余0，换基本变量集

6.最后一行中没有负数

7.最优解是左右对应

8.目标函数的最优值，
对于最大化程序来说是最后一行和最后一列的数字，
但是对于最小化程序来说是这个数字的负数

5.2若M行非负，则判断

若人工变量x仍然是基本变量

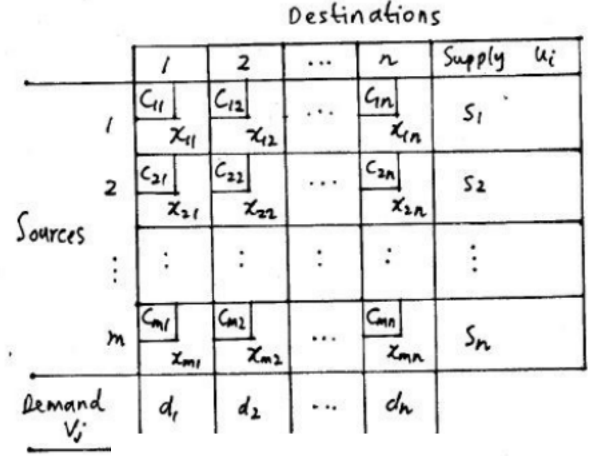
则无解

若左边没有人工变量

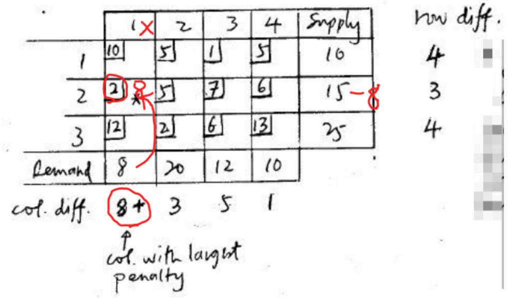
则删去M行 和 人工变量列

运输模型　2

1.画出 运输模型图，写出总运费
左来源
上目的地
虚拟源：运输成本被视为零



2.求初始基本解

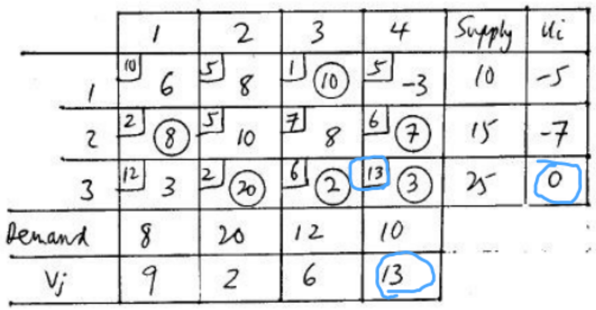


最大插入最小

第一步：每行找出最小的两个相减得到row diff
第二步：每列出最小的两个相减得到col diff
第三步：找到最大的值row和col diff中最大的值
第四步：找到其对应的行或者列的最小代价值，标记上*
第五步：将供给和需求的最小值填进去*，然后供给和需求都减去最小值，并删除为0的行

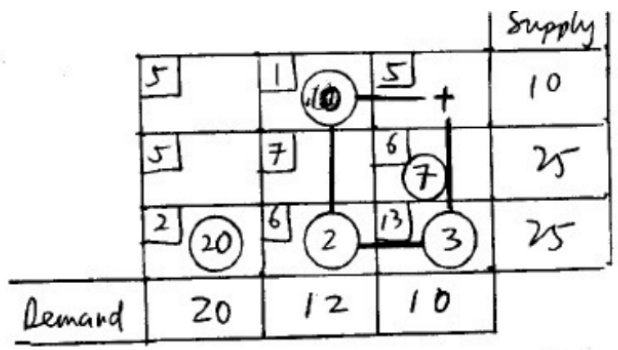
3.测试最优

1.开始先找基本元素（圆圈）最多的行或者列，设为0
2.算圆圈cost=u+v
3.空白数为cost-u-v
4.如果空白出现负数，则非最优



4.获得最优解

1.找到最负的元素
2.用圆圈构成loop
3.从起始位置沿着环标上1234等
4.找到2468等偶数位置最小值
5.奇数位置加，偶数位置减
6.减为0的取消圆圈，负数部分加圆圈



任务分配问题

匈牙利方法 2 1

Hungarian method

1.用M替换-，-代表不能做的工作

虚拟设置一个工作，代价为0

2.找到每行最小元素

3.所有减它

4.每列最小元素

5.所有减它

6.用最少数量的线盖住所有的0

7.所有非盖住的数字减去最小数a

8.两条线交点处的元素加最小数a

9.给四个人分配cost为0的工作

10.分配完后0变为0*

正定性测试

3

positive definite

positive semi-definite

negative-definite

negative semi-definite

indefinite

正定性测试（Test for Positive-Definiteness）

为了测试一个矩阵 A 是否正定，使用主子式（principal minors）的方法。假设 $A = [a_{ij}]$ 是一个 $n \times n$ 对称矩阵，定义如下的行列式：

$$A_1 = |a_{11}|, A_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, A_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}, \ldots, A_n = |A|$$

- **正定矩阵的条件**：如果 $A_1 > 0, A_2 > 0, \ldots, A_n > 0$，则矩阵 A 为正定矩阵。
- **正半定矩阵（positive semi-definite）**：如果 $A_1 \geq 0, A_2 \geq 0, \ldots, A_n \geq 0$，则矩阵 A 为正半定矩阵。
- **负定矩阵的条件**：如果 $A_1 < 0, A_2 > 0, A_3 < 0, \ldots$ 交替符号，则矩阵 A 为负定矩阵。

PPT45

等式拉格朗日法 3 1

1.判断NLP：目标函数可能不是线性函数，或者某些约束可能不是线性约束。

2.写出题目要求，和约束，标准形式

3.写出拉格朗日函数 L

$$L(X,\lambda) = f(X) + \sum_{i=1}^{m} \lambda_i h_i(X)$$

4.对x求偏导

对λ求偏导

$$\frac{\partial L}{\partial x_j} = 0\ (j = 1, 2, \ldots, n) \text{ or } \nabla_x L = 0$$

$$\frac{\partial L}{\partial \lambda_i} = 0\ (j = 1, 2, \ldots, m) \text{ or } \nabla_\lambda L = 0$$

5.求出X*和λ*

6.选择非零的λ对应的h-列向量 $h(X)$

7.约束条件对x求偏导 $\nabla h(X)$

8.得到约束式子，y列向量 $\nabla h(X)^T Y$

9.写出Hessian矩阵，证明正定

10.求Y^T H T大于零，

$$\nabla_{xx}^2 L = \begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 \partial x_2} \\ \frac{\partial^2 L}{\partial x_2 \partial x_1} & \frac{\partial^2 L}{\partial x_2^2} \end{bmatrix}$$

任意的y不等于0

就是最小值

验证答案有没有算错，把x和λ带回等式中，查看是不是为0

不等式拉格朗日法 2 1

1.判断NLP：目标函数可能不是线性函数，

或者某些约束可能不是线性约束。

Minimize: $f(X)$

Subject to: $g_j(X) \le 0,\ j = 1, 2, \ldots, p$

2.写出题目要求，和约束，标准形式

3.写出拉格朗日函数 L

$$L(X, \mu) = f(X) + \sum_{j=1}^{p} \mu_j g_j(X)$$

4.写出KT定理(充分条件)

（1）对x求偏导

（2）约束小于等于0

（3） μx约束=0

（4） μ大于等于0

(1) $\nabla_x L(X^*, \mu^*) = 0$

(2) $g(X^*) \le 0$

(3) $\mu_j^* g_j(X^*) = 0,\ j = 1, 2, \ldots, p$

(4) $\mu^* \ge 0$

(5) $Y^T \nabla_{xx}^2 L(X^*, \mu^*) Y > 0$ on $M' = \{Y: \nabla g_j(X^*)^T Y = 0 \text{ for all } j \in J\}$ where $J = \{j: g_j(X^*) = 0, \mu_j^* > 0\}$.

5.按照μ1 和μ2 分类讨论,求出X*和μ*

μ1 = μ2 =0

6.21-S1-Q2

According to PPT 45, $\nabla_{xx}^2 L(x,\mu)$ is a positive definite ⇒ the function L is a convex function.

According to PPT 45, A linear function is also convex

⇒ $g_1 = x_1 - 2$, $g_2 = x_2 - 2$ are linea function

⇒ So g_1 g_2 are also convex

According to PPT 46. KT sufficient theorem, f(x) is convex, the inequality constrains $g_i(x)$ are all convex functions and equality constraints $h_i(x)$ be linear. If the exists a solution $(X^*\ \lambda^*\ \mu^*)$ that satisfies the K-T conditions, then X^* is an optimal solution to the NLP problem.

So $X^* = (-\frac{1}{4}, -\frac{1}{4})$ is a mininum point

μ1 = 0， μ2 > 0

μ1 > 0， μ2 = 0

μ1 > 0， μ2 > 0

6.选择所有μ大于的，对应的g，

$g_j(X^*)$

7.让g=0对x求偏导，

$\nabla g_j(X^*)^T$

8.乘以Y=[y1 y2 y3]得到Y的式子

$\nabla g_j(X^*)^T Y = 0$

9.写出Hessian矩阵，证明正定

$$[\nabla^2 f(X)]_{ij} = \frac{\partial^2 f(X)}{\partial x_i \partial x_j}$$

10.求Y^T H T大于零，就是最小值

$$\nabla^2 f(x_1, x_2) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}$$

KT充分定理 2 1

K-T充分定理

设f(X)为convex，
不等式约束g (X)为convex，
等式约束h(X)为linear。

如果存在满足K-T条件的解(X*， a *， μ*)，则X*是NLP问题的最优解。

不等式 线性约束 是convex

灵敏度分析

$L_\lambda(X, \lambda) = f(X) + \lambda^T(AX - B)$

B+ΔB

2

1

目标函数的变换=约束条件B变化 x 拉格朗日乘子

Δz=-ΔBi λ i

$$f(\bar{X}) \approx f(X^*) - \sum_{i=1}^{m} \Delta B_i \lambda_i^*$$

Or

$$\Delta f(X_*) = f(\bar{X}) - f(X^*) \approx - \sum_{i=1}^{m} \Delta B_i \lambda_i^*$$

随机过程公式总结

条件概率公式

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

几何随机变量

$$P(X = k) = (1-p)^{k-1}p.$$

几何随机变量 - 均值

$$E(X) = \frac{1}{p}$$

马尔可夫属性
无记忆属性

$$P(X = m + n \mid X > m) = P(X = n)$$

指数随机变量

$$X \sim EXP(\lambda)$$

指数随机变量 - 累积分布函数

$$P(X \le x) = 1 - e^{-\lambda x}, \qquad x \ge 0$$
$$P(X > x) = e^{-\lambda x}$$

指数随机变量 - 均值

$$E(X) = \frac{1}{\lambda}$$

指数随机变量 - 概率密度函数

$$f_X(x) = \begin{cases} \lambda e^{-\lambda x}, & x > 0 \\ 0, & x \le 0 \end{cases}$$

泊松随机变量公式 2 1

$$P(X(t) = k) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}, \qquad k = 0, 1, 2, \cdots$$

泊松随机变量

$$X(t) \sim P(\lambda t)$$

时间间隔[0，t]内事件

泊松随机变量 - 均值

$$\textbf{mean } \lambda t.$$

到达间隔时间

$$T \sim EXP(\lambda)$$

n个泊松过程的叠加

$$\sum_{i=1}^{n} \lambda_i$$

定义

Definition A *discrete time Markov chain (DTMC)* is a discrete time stochastic process $\{X_n : n \in N\}$ with countable state space S, such that the *Markov property* holds

离散时间马尔可夫链（DTMC）是一个离散时间随机过程，具有可计数的状态空间S，使得马尔可夫链的马尔可夫性成立

1 1 $$P(X_n = j \mid X_{n-1} = i, \boxed{X_{n-2} = i_2, X_{n-3} = i_3, \cdots, X_0 = i_n})$$

条件概率

$$= P(X_n = j \mid X_{n-1} = i). \qquad (15)$$

明天赚到的钱之和今天赚到的钱有关
与昨天及之前赚到的钱无关

	DTMC	CTMC
1.状态空间	1. $\{X_n : n \in N\}$ with state space $S = \{0, 1, \cdots\}$	$\{X(t) : t \geq 0\}$ with state space $S = \{0, 1, \cdots\}$
2.马尔可夫性质	**2. Markov property** $P(X_n = j \mid X_{n-1} = i, X_{n-2} = i_2, \cdots, X_0 = i_n)$ $= P(X_n = j \mid X_{n-1} = i)$	$P(X(t) = j \mid X(s) = i, X(u) = x(u)), \quad u \leq s \leq t$ $= P(X(t) = j \mid X(s) = i)$
3.状态i的逗留时间Ti 3	**3. Sojourn time T_i of state i** T_i is a geometric r.v. with mean $E(T_i) = \frac{1}{1-p_{ii}}$ 几何 随机变量	$T_i \sim EXP(\lambda_i)$ 指数 随机变量 $P(T_i > x) = e^{-\lambda_i x}$ E(Ti) = 1/ λ
	状态i的逗留时间Ti的期望	状态i的逗留时间Ti大于x的概率

DTMC	CTMC
4. Transition probability	
$p_{ij}(n) = P(X_n = j \mid X_0 = i)$	$p_{ij}(t) = P(X(t) = j \mid X(0) = i)$
$\sum_j p_{ij}(n) = 1$	$\sum_j p_{ij}(t) = 1$
5. Transition probability matrix	
$P(n) = [p_{ij}(n)]$	$H(t) = [p_{ij}(t)]$
On-step TPM, $P = [p_{ij}]$	
$P(m+n) = P(m)P(n)$	$H(s+t) = H(s)H(t)$
$P(n) = P^n$	$H(t) = \exp(Qt)$
	$= I + Qt + Q^2 \frac{t^2}{2!} + Q^3 \frac{t^3}{3!} + \cdots$

4.
转移概率

5.
转移概率矩阵

TPM

1

状态转移矩阵P = 从状态i变为j的概率，要求行和为1

过渡率矩阵

TRM

1

DTMC	CTMC
	Transition rate matrix $Q = [q_{ij}]$ 过渡率矩阵 q_{ij} **is the rate at which the CTMC moves from state i to state j** gij是CTMC从状态i移动到状态j的速率 $\sum_j q_{ij} = 0$ **i.e., sum of each row of Q is zero**

$$q_{ij} = \lim_{h \to 0} \frac{p_{ij}(h) - \delta_{ij}}{h}$$

状态转换图

不会标出自己到自己的概率qii，只能用行和为0来解出qii

6.
状态概率

DTMC	CTMC
6. State probability	
$p_j(n) = P(X_n = j)$	$p_j(t) = P(X(t) = j)$
$\sum_j p_j(n) = 1$	$\sum_j p_j(t) = 1$
$\pi(n) = [p_0(n) \;\; p_1(n) \;\; p_2(n) \;\; \cdots \;]$	$\pi(t) = [p_0(t) \;\; p_1(t) \;\; p_2(t) \;\; \cdots \;]$
$\pi(n)$ is the pmf of $X(n)$	$\pi(t)$ is the pmf of $X(t)$
$\pi(n) = \pi(0)P(n) = \pi(0)P^n$	$\pi(t) = \pi(0)H(t) = \pi(0)\exp(Qt)$

p0(n): n步后，变为 0 的概率

p1(n): n步后，变为 1 的概率

Π（n）：n步后，变为 0，1，2，3... 的概率的集合

p0(t): t时间后，变为 0 的概率

p1(t): t时间后，变为 1 的概率

Π（t）：t时间后，变为 0，1，2，3... 的概率的集合

7. 稳态概率

2

DTMC	CTMC
7. Steady state probability	
$Y = \lim_{n\to\infty} \pi(n) = [y_0 \quad y_1 \quad y_2 \quad \cdots\ ]$	$\pi = \lim_{t\to\infty} \pi(t) = [\pi_0 \quad \pi_1 \quad \pi_2 \quad \cdots\ ]$
$y_j = \lim_{n\to\infty} p_j(n) = \lim_{n\to\infty} P(X_n = j)$	$\pi_j = \lim_{t\to\infty} p_j(t) = \lim_{t\to\infty} P(X(t) = j)$
To calculate Y :	**To calculate π :**
$\begin{cases} Y = YP \\ \sum_j y_j = 1, \quad 0 \le y_j \le 1 \end{cases}$ 再怎么用状态转移矩阵（TPM）都不会变；所有状态之和概率要为1	$\begin{cases} \pi Q = 0 \quad \text{(rate balance equations)} \\ \sum_j \pi_j = 1, \quad 0 \le \pi_j \le 1 \end{cases}$

法2：输出=输入

y0：无穷n步后，变为 0 的概率

y1：无穷n步后，变为 1 的概率

Y：无穷n步后，变为 0，1，2，3... 的概率的集合

法2：输出=输入

平均生产效率 R

= 处于生产状态下的概率权重 x 生产速率

1　　　1

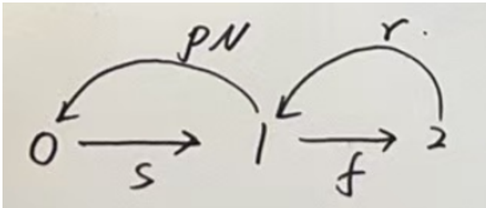
平均产量R

Average production rate R

$= $ no. of parts produced per hour 每小时生产的零件数量

$= \pi_1 p$

R'=Π1' pN



$$R' = \pi_1' \cdot pN$$

生产速率不同

稳态可用性

$$\text{Steady-state availability} = \left(1 + \frac{f}{r}\right)^{-1} \tag{71}$$

稳态可用性

Steady-state availability is the probability that the system is functioning in a productive way.

稳态可用性是系统以生产方式运行的可能性。

平均完工时间

平均完工时间

Mean completion time

$$= \left(1 + \frac{f}{r}\right) \frac{1}{p}$$

$$= \frac{\text{mean processing time}}{\text{steady-state availability}}$$

平均处理时间

稳态可用性

层次分析法
Analytic Hierarchy Process　　2

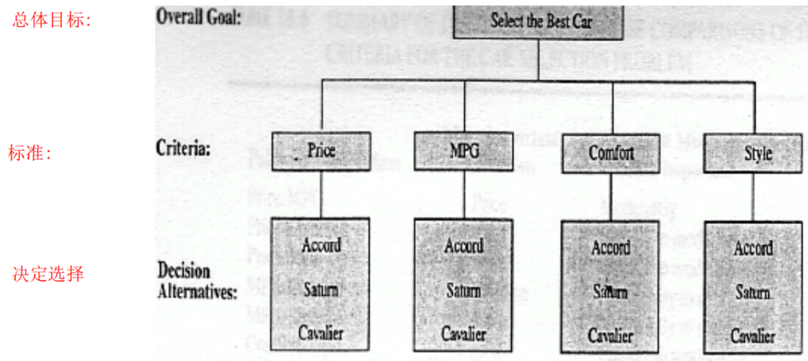
比较标准

0.画出AHP树，总体目标，标准，决策选项

Overall Goal

Critetia

Decision Alternatives



1.写出每个标准，两两比较，求列和

得到两两比较矩阵：

重要在前，当横坐标

表C: Payne女士对汽车选择问题的4个标准的两两比较总结

Table C: Summary of Ms. Payne's pairwise comparisons of the 4 criteria for the car selection problem

Pairwise Comparison	**More Important Criterion**	**How Much More Important**	**Numerical Rating**
Price-MPG	*Price*	*Moderately*	*3*
Price-Comfort	*Price*	*Equally to moderately*	*2*
Price-Style	*Price*	*Equally to moderately*	*2*
MPG-Comfort	*Comfort*	*Moderately to strongly*	*4*
MPG-Style	*Style*	*Moderately to strongly*	*4*
Comfort-Style	*Style*	*Equally to moderately*	*2*

Step 1. Sum the values in each column.

	Price	MPG	Comfort	Style
Price	1	3	2	2
MPG	1/3	1	1/4	1/4
Comfort	1/2	4	1	1/2
Style	1/2	4	2	1
Sum	2.333	12.0	5.25	3.75

2.每个标准除以列和

得到两两比较矩阵归一化

例题中保留3位小数

Step 2. Divide each element of the matrix by its column total.

	Price	MPG	Comfort	Style
Price	0.429	0.250	0.381	0.533
MPG	0.143	0.083	0.048	0.067
Comfort	0.214	0.333	0.190	0.133
Style	0.214	0.333	0.381	0.267

normalized column

3.求行平均

得到每个 标准的优先级

Step 3. Average each row to determine the priority of each criterion

	Price	MPG	Comfort	Style	Priority/Average
Price	0.429	0.250	0.381	0.533	0.398
MPG	0.143	0.083	0.048	0.067	0.085
Comfort	0.214	0.333	0.190	0.133	0.218
Style	0.214	0.333	0.381	0.267	0.299

priority vector

查找一致性比率　1　1

Step 1 Multiply each value in the first column of the pairwise comparison matrix by the priority of the first item; multiply each value in the second column of the pairwise comparison matrix by the priority of the second item; continue this process for all columns of the pairwise comparison matrix.
步骤1将两两比较矩阵第一列中的每个值乘以第一项的优先级；将两两比较矩阵第二列中的每个值乘以第二项的优先级；对成对的所有列继续此过程

Sum the values, across the rows to obtain a vector of values labeled "weighted sum." The computation for the car selection problem is as follows: ***criteria pairwise comparison matrix * priority vector***
将这些值跨行求和，以获得标记为"加权和"的值向量。汽车选择问题的计算如下：标准两两比较矩阵*优先级向量

1.标准两两比较矩阵 独特运算 标准优先级

非矩阵乘法

列 分别乘以 单个，再求和

得到 加权和列向量

$$\begin{bmatrix} 1 & 3 & 2 & 2 \\ 1/3 & 1 & 1/4 & 1/4 \\ 1/2 & 4 & 1 & 1/2 \\ 1/2 & 4 & 2 & 1 \end{bmatrix} \begin{bmatrix} 0.398 \\ 0.085 \\ 0.218 \\ 0.299 \end{bmatrix}$$

$$= 0.398 \begin{bmatrix} 1 \\ 1/3 \\ 1/2 \\ 1/2 \end{bmatrix} + 0.085 \begin{bmatrix} 3 \\ 1 \\ 4 \\ 4 \end{bmatrix} + 0.218 \begin{bmatrix} 2 \\ 1/4 \\ 1 \\ 2 \end{bmatrix} + 0.299 \begin{bmatrix} 2 \\ 1/4 \\ 1/2 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.398 \\ 0.133 \\ 0.199 \\ 0.199 \end{bmatrix} + \begin{bmatrix} 0.255 \\ 0.085 \\ 0.340 \\ 0.340 \end{bmatrix} + \begin{bmatrix} 0.436 \\ 0.054 \\ 0.218 \\ 0.436 \end{bmatrix} + \begin{bmatrix} 0.598 \\ 0.075 \\ 0.149 \\ 0.299 \end{bmatrix} = \begin{bmatrix} 1.687 \\ 0.347 \\ 0.907 \\ 1.274 \end{bmatrix}$$

2．加权和 / 标准优先级

得到 λ

Step 2 Divide the elements of the weighted sum vector obtained in step 1 by the corresponding priority for each criterion.
步骤2　将步骤1得到的加权和向量的元素除以每个准则对应的优先级。

Price	*1.687/0.398*	=	*4.236*
MPG	*0.347/0.085*	=	*4.077*
Comfort	*0.907/0.218*	=	*4.163*
Style	*1.274/0.299*	=	*4.262*

3.求平均值得到λmax

步骤3计算步骤2中找到的值的平均值；
这个平均值记为λMax

Step 3 Compute the average of the values found in step 2; this average is denoted λ_{max}.

λ_{max} = (4.236 + 4.077 + 4.163 + 4.264)/4 = 4.185

4.consistency index = λmax-n / n-1

步骤4计算一致性指数CI，如下所示。
这里n = 被比较的项目数量。

Step 4 Compute the ***consistency index*** (CI) as follows

Here, *n* = no. of items being compared.

$$\text{C.I.} = \frac{\lambda_{\max} - n}{n-1}.$$

5.consistency ratio = CI/RI

Step 5 Compute the ***consistency ratio***, which is defined as

步骤5计算一致性比，定义为

CR = CI / RI

where RI is the consistency index of a randomly generated pairwise comparison matrix.

其中RI是一个随机生成的配对的一致性指标

The value of RI depends on the number *n* of items being compared and is given as follows:

RI的值取决于被比较项目的数量，给出如下：

n	3	4	5	6	7	8
RI	0.58	0.90	1.12	1.24	1.32	1.41

6.CR < 0.1 acceptable

CR = 0.0616/0.90 = 0.068 < **0.1** [Acceptable consistency]

（可接受的一致性）

AHP：比较选项

1.每个标准下，比较产品

Price

	Accord	Saturn	Cavalier
Accord	1	1/3	1/4
Saturn	3	1	1/2
Cavalier	4	2	1

2.每个标准下

两两比较矩阵

产品 除以列和

得到两两比较矩阵归一化

MPG

	Accord	Saturn	Cavalier
Accord	1	1/4	1/6
Saturn	4	1	1/3
Cavalier	6	3	1

3.求行平均

得到每个产品，在该标准下的优先级

Comfort

	Accord	Saturn	Cavalier
Accord	1	2	8
Saturn	1/2	1	6
Cavalier	1/8	1/6	1

每个标准对汽车的偏好

Style

	Accord	Saturn	Cavalier
Accord	1	1/3	4
Saturn	3	1	7
Cavalier	1/4	1/7	1

4.组成4个产品与4个标准优先级的矩阵

Table F: Priorities for each car using each criterion

表F：使用每个标准的每辆车的优先级

	Criterion			
	Price	MPG	Comfort	Style
Accord	0.123	0.087	**0.593**	0.265
Saturn	0.320	0.274	0.341	**0.656**
Cavalier	**0.557**	**0.639**	0.065	0.080

整体优先级排序

5.计算复合优先级

4个产品与4个标准优先级的矩阵 * 标准优先级列向

得到每个车的优先级

The **composite priority** is calculated as follows: 复合优先级计算公式如下：

Level 3 priority matrix *Level 2 priorities*

$$\begin{array}{c} \\ 车1 \\ 车2 \\ 车3 \end{array} \begin{array}{c} \begin{array}{cccc} 标准1 & 标准2 & 标准3 & 标准4 \end{array} \\ \begin{bmatrix} 0.123 & 0.087 & 0.593 & 0.265 \\ 0.320 & 0.274 & 0.341 & 0.656 \\ 0.557 & 0.639 & 0.065 & 0.080 \end{bmatrix} \end{array} \text{x} \begin{bmatrix} 0.398 \\ 0.085 \\ 0.218 \\ 0.299 \end{bmatrix} \begin{array}{c} 标准1 \\ 标准2 \\ 标准3 \\ 标准4 \end{array}$$

$$= \begin{bmatrix} .123(.398)+.087(.085)+.593(.218)+.265(.299) \\ .320(.398)+.274(.085)+.341(.218)+.656(.299) \\ .557(.398)+.639(.085)+.065(.218)+.080(.299) \end{bmatrix}$$

$$= \begin{bmatrix} 0.265 \\ 0.421 \\ 0.314 \end{bmatrix}$$

6.优先级排序

得到决策结果

Ranking these priorities, we have **the AHP ranking of the decision alternatives**

对这些优先级进行排序，我们有决策选择的AHP排序

	Car	Priority
1.	*Saturn*	***0.421***
2.	*Cavalier*	***0.314***
3.	*Accord*	***0.265***

决策树分析

u(least favorable outcome) = 0

冯 · 诺伊曼-摩根斯坦方法

u(most favorable outcome) = 1

奖励ri的效用
utility of the reward ri

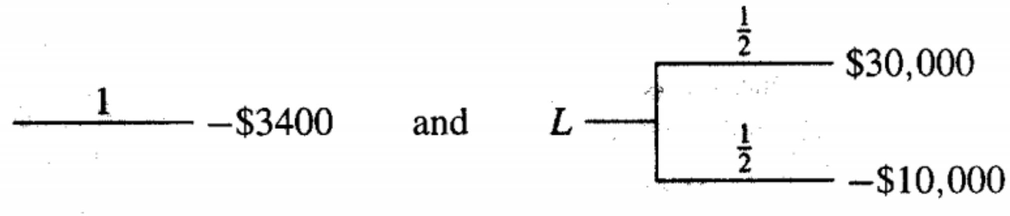
$$u(r_i) = q_i$$

期望效用
expected utility

$$E\,(U \text{ for } L) = \sum_{i=1}^{n} p_i u(r_i)$$

确定性当量CE(L)
certainty equivalent

1



$CE(L) = -\$3400.$

风险溢价RP(L)
risk premium

$RP(L) = EV(L) - CE(L)$, 确定性当量CE(L)

EV(L)是彩票结果的期望值

Risk-averse if $RP(L) > 0$

Risk-neutral if $RP(L) = 0$

Risk-seeking if $RP(L) < 0$

将风险纳入决策树分析

将每个最终资产位置x替换为其效用u(x)

信息的价值 1

EV(after test) – EV(without test)

EVSI = EVWSI – EVWOI

样本信息期望值 (EVSI)
Expected Value of Sample Information (EVSI)

信息的价值

样本信息 无成本 期望值 (EVWSI)
Expected Value with Sample Information(EVWSI).

测试无成本

原始信息期望值 (EVWOI)
Expected Value with Original Information(EVWOI).

测试不可用