

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: $L_1 + L_2$

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: $1/2(1 - \rho)$

MM2: $(1 - \rho)$

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 $(L_1 + L_2)$ 要小得多。
EE6204

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 :

$$\begin{aligned}\rho &= \frac{\lambda}{\mu} && \text{利用率} \\ \pi_0 &= 1 - \rho && \text{空的概率 (the proportion of time each server is idle)} \\ \pi_k &= \rho^k (1 - \rho), \quad k \geq 1 && k \text{ 个顾客的概率} \\ L &= \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda} && \text{系统中的平均顾客数【带系统的都是L】 (the average number in the queue system)} \\ Q &= \frac{\rho^2}{1 - \rho} = \frac{\lambda^2}{\mu(\mu - \lambda)} && \text{排队的平均顾客数} \\ W &= \frac{1}{\mu(1 - \rho)} = \frac{1}{\mu - \lambda} && \text{顾客在系统的平均时间} \\ D &= W - \frac{1}{\mu} = \frac{\lambda}{\mu(\mu - \lambda)} && \text{排队的平均等待时间}\end{aligned}$$

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

$$\begin{aligned}\rho &= \frac{\lambda}{\mu} \\ \pi_0 &= \left(\sum_{k=0}^N \rho^k \right)^{-1} = \frac{1 - \rho}{1 - \rho^{N+1}} \\ \pi_k &= \rho^k \pi_0 = \frac{\rho^k (1 - \rho)}{1 - \rho^{N+1}}, \quad 0 \leq k \leq N \\ L &= \frac{\rho [1 - \rho^N - N \rho^N (1 - \rho)]}{(1 - \rho)(1 - \rho^{N+1})} \\ Q &= \frac{\rho^2 [1 - \rho^N - N \rho^{N-1} (1 - \rho)]}{(1 - \rho)(1 - \rho^{N+1})} \\ W &= \frac{1 - \rho^N - N \rho^N (1 - \rho)}{\mu(1 - \rho)(1 - \rho^{N+1})} \\ D &= \frac{\rho [1 - \rho^N - N \rho^{N-1} (1 - \rho)]}{\mu(1 - \rho)(1 - \rho^{N+1})}\end{aligned}$$

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

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

是如果k小于等于m-1,
就用 0乘以上面这一坨,
如果k大于等于m,
就用 0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 :

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

- 1. 写出目标函数 和 subject to
- 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^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， $b \div$ 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

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

		Destinations				
		1	2	...	n	Supply u_i
Sources	1	C_{11} x_{11}	C_{12} x_{12}	...	C_{1n} x_{1n}	S_1
	2	C_{21} x_{21}	C_{22} x_{22}	...	C_{2n} x_{2n}	S_2

	m	C_{m1} x_{m1}	C_{m2} x_{m2}	...	C_{mn} x_{mn}	S_m
Demand V_j		d_1	d_2	...	d_n	

	1	2	3	4	Supply	row diff.
1	10	5	1	5	10	4
2	2	5	7	6	15	3
3	12	2	6	13	25	4
Demand	8	20	12	10		
col. diff.	8	3	5	1		

col. with largest penalty

2. 求初始基本解

- 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

最大插入最小

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

	1	2	3	4	Supply	u_i
1	10 6	5 8	1 10	5 -3	10	-5
2	2 8	5 10	7 8	6 7	15	-7
3	12 3	2 20	6 2	13 3	25	0
Demand	8	20	12	10		
V_j	9	2	6	13		

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

				Supply
5	1	10	5	10
5	7		6	25
2	20	6	13	25
Demand	20	12	10	

任务分配问题

匈牙利方法

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*

目标函数的变化可以近似

为约束条件变化 B_i

1

1

乘以拉格朗日乘子 λ_i

的线性组合

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

Or

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

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:

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

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

- 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

$$\frac{\partial L}{\partial x_j} = 0 \quad (j = 1, 2, \dots, n) \text{ or } \nabla_x L = 0$$
$$\frac{\partial L}{\partial \lambda_i} = 0 \quad (i = 1, 2, \dots, m) \text{ or } \nabla_\lambda L = 0$$

5. 求出X*和 λ*

6. 选择非零的 λ 对应的h

$$h(X)$$

7. 约束条件对x求偏导

$$\nabla h(X)$$

8. 得到约束式子

$$\nabla h(X)^T \lambda$$

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

不等式拉格朗日法

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: $f(X)$

2. 写出题目要求，和约束，标准形式

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

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

(1) 对x求偏导

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

(2) 约束小于等于0

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

(3) μ x约束=0

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

(4) μ 大于等于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 μ_1 和 μ_2 分类讨论, 求出 X^* 和 μ^*

$$\mu_1 = \mu_2 = 0$$

$$\mu_1 = 0, \mu_2 > 0$$

$$\mu_1 > 0, \mu_2 = 0$$

$$\mu_1 > 0, \mu_2 > 0$$

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 $Y=[y_1 \ y_2 \ y_3]$ 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

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

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

$$P(X \leq x) = 1 - e^{-\lambda x}, \qquad x \geq 0$$

$$P(X > x) = e^{-\lambda x}$$

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

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

泊松随机变量公式

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

泊松随机变量

时间间隔[0 , t]内事件

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

泊松随机变量 - 均值

mean λt .

到达间隔时间

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

n个泊松过程的叠加

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

DTMC	CTMC
<p>1. $\{X_n : n \in N\}$ with state space $S = \{0, 1, \dots\}$</p>	<p>$\{X(t) : t \geq 0\}$ with state space $S = \{0, 1, \dots\}$</p>
<p>2. Markov property</p> $P(X_n = j \mid X_{n-1} = i, X_{n-2} = i_2, \dots, X_0 = i_n)$ $= P(X_n = j \mid X_{n-1} = i)$	$P(X(t) = j \mid X(s) = i, X(u) = x(u)), u \leq s \leq t$ $= P(X(t) = j \mid X(s) = i)$
<p>3. Sojourn time T_i of state i</p> <p>T_i is a geometric r.v. with mean</p> $E(T_i) = \frac{1}{1-p_{ii}}$	$T_i \sim EXP(\lambda_i)$ $P(T_i > x) = e^{-\lambda_i x}$

1. 定义

2. 马尔可夫性质

3. 状态*i*的逗留时间*T_i*

4.
转移概率

5.
转移概率矩阵

TPM

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

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

6.
状态概率

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

$p_0(n)$: n 步后, 变为 0 的概率

$p_1(n)$: n 步后, 变为 1 的概率

(n) : n 步后, 变为 0, 1, 2, 3... 的概率的集合

$p_0(t)$: t 时间后, 变为 0 的概率

$p_1(t)$: t 时间后, 变为 1 的概率

(t) : t 时间后, 变为 0, 1, 2, 3... 的概率的集合

7.
稳态概率

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

y_0 : 无穷 n 步后, 变为 0 的概率

y_1 : 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

平均生产效率 R

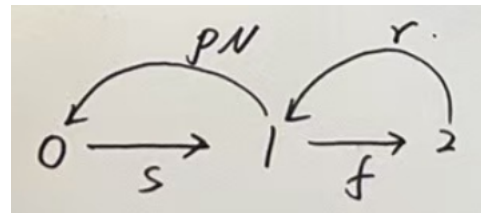
= 处于生产状态下的概率权重 \times 生产速率

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

$$= \pi_1 p$$



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

生产速率不同

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

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。64

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

决策树分析

冯·诺伊曼-摩根斯坦方法

奖励 r_i 的效用
utility of the reward r_i

期望效用
expected utility

确定性当量 $CE(L)$
certainty equivalent

风险溢价 $RP(L)$
risk premium

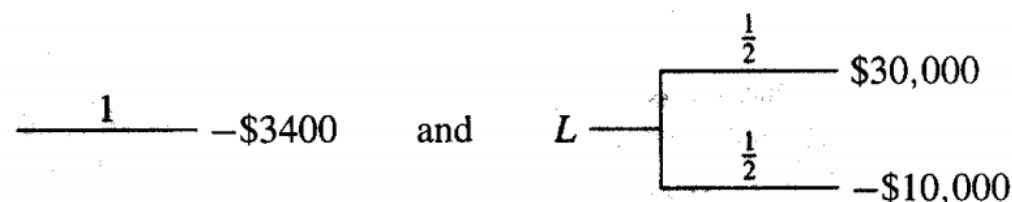
将风险纳入决策树分析

$$u(\text{least favorable outcome}) = 0$$

$$u(\text{most favorable outcome}) = 1$$

$$u(r_i) = q_i$$

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



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

$$RP(L) = EV(L) - CE(L), \text{ 确定性当量 } CE(L)$$

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 $RP(L) < 0$

将每个最终资产位置 x 替换为其效用 $u(x)$

信息的价值

$$EV(\text{after test}) - EV(\text{without test})$$

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