随机过程第二次大作业报告

漆耘含

无 63 班 2016011058

摘要：通过对问题的建模，采用 M/M/S/∞模型，用 MATLAB 对其进行仿真，并通过不同参数的设置，研究不同的选择边界。

一、问题引入及分析

你在同一地点要去银行和药店两个地点办事，先后顺序任意。银行到达顾客遵循参数为λ_1的泊松过程，银行共有$M_1 > 1$名柜员为顾客服务，每位柜员服务时间服从参数为μ_1的负指数分布；药店到达顾客遵循参数为λ_2的泊松过程，药店共有$M_2 > 1$名柜员为顾客服务，每位柜员服务时间服从参数为μ_1的负指数分布。当你到达该地点时，银行里有L_1人在排队,药店里有L_2人在排队。请分析决策：如果从节省时间的角度考虑，应该先到银行排队还是应该先到药店排队？

这个问题是排队论中的经典问题，两个地点顾客到达服从泊松分布，每一个柜员的服务时间也服从于泊松分布，并且要求每一个地点里面，柜台数都大于 1，因此采用排队论的经典模型 M/M/S/∞。

二、理论原理分析

1. M/M/1/∞模型

M/M/1/∞模型（单服务台模型）是最简单的一种情况，顾客到达服从参数为λ的泊松过程，柜台服务时间服从参数为μ的泊松过程，柜台数为 1，系统容量无限，排队长度无限，服务规则是先到先服务。

设$p_n = P\{N = n\}, n = 0,1,2,\dots\dots$为系统平稳后队列长度为 N 的概率分布，有：

$$p_1\mu = p_0\lambda$$

因此：

$$p_1 = \frac{\lambda}{\mu}p_0$$

同时，有：

$$p_0\lambda + p_2\mu = p_1(\lambda + \mu)$$

$$p_2 = \frac{\lambda}{\mu}p_1 = \left(\frac{\lambda}{\mu}\right)^2 p_0$$

以此类推，可以得到：

$$p_n = \left(\frac{\lambda}{\mu}\right)^n p_0$$

进行归一化，得到：

$$\sum_{n=0}^{\infty} p_n = \sum_{n=0}^{\infty}\left(\frac{\lambda}{\mu}\right)^n p_0 = 1$$

解方程，得到

$$p_0 = 1 - \frac{\lambda}{\mu}$$

由此便得到系统平稳状态后对长为 N 的概率分布。我们假设$\lambda < \mu$，即单位时间顾客到达速率小于柜台的服务效率，这样才能达到统计平衡。在概率分布中，$\frac{\lambda}{\mu}$为服务强度，记为ρ，即$\rho = \frac{\lambda}{\mu}$。下面推导系统的几个主要指标：

（1） 系统中平均顾客数（平均队长）

$$L_s = \sum_{n=0}^{\infty} np_n = \sum_{n=0}^{\infty} n(1-\rho)\rho^n$$

$$= \frac{\rho}{1-\rho} = \frac{\lambda}{\mu - \lambda}$$

（2） 系统中等待的平均顾客数（平均排队长）

$$L_q = \sum_{n=1}^{\infty}(n-1)p_n = \sum_{n=1}^{\infty} np_n - \sum_{n=1}^{\infty} p_n = L_s - (1 - p_0)$$

$$= L_s - \rho = \frac{\rho^2}{1-\rho} = \rho L_s$$

（3） 在系统中顾客平均逗留时间：

在 M/M/1 中，顾客在系统中逗留时间 T 服从参数为$\mu - \lambda$的负指数分布，即

$$f(t) = (\mu - \lambda)e^{-(\mu-\lambda)t}, t > 0$$

因此，平均逗留时间：

$$W = E[T] = \frac{1}{\mu - \lambda}$$

（4） 在队列中顾客平均等待时间：

因为逗留时间=等待时间T_q +服务时间 V，即$T = T_q + V$，故：

$$W = E(T_q) + E(V) = W_q + \frac{1}{\mu}$$

$$W_q = W - \frac{1}{\mu} = \frac{\rho}{\mu - \lambda} = \rho W$$

此外，还可以得到（时间与空间的关系）：

$$L = \lambda W$$

$$L_q = \lambda W_q$$

这两个公式常称为 Little 公式。

2. M/M/S/∞模型

和 M/M/1 的参数假设差不多，即顾客到达时间为参数为λ的指数分布，单台服务时间为参数为μ的指数分布，服务台数为 S，每一个服务台的服务效率是一样的，即$\mu_1 = \mu_2 = \mu_3 = \cdots = \mu$，系统容量无限，排队长度无限，服务规则为先到先服务。

设$p_n = P\{N = n\}, n = 0,1,2,\ldots\ldots$为系统平稳后队列长度为 N 的概率分布，有：

$$\lambda_n = \lambda, n = 0,1,2\ldots$$

系统的服务率：

$$\mu_n = \begin{cases} n\mu, & n = 1,2,3,\ldots,s \\ s\mu, & n = s, s+1,\ldots \end{cases}$$

记$\rho_s = \frac{\rho}{s} = \frac{\lambda}{s\mu}$，则当$\rho_s < 1$时，不至于越排越长，称$\rho_s$为系统的服务强度或服务机构的平均利用率。

通过公式推导，得：

$$p_n = \begin{cases} \dfrac{\rho^n}{n!} p_0, & n = 1,2,3,\ldots,s \\ \dfrac{\rho^n}{s!\, s^{n-s}} p_0, & n = s, s+1,\ldots \end{cases}$$

其中，

$$p_0 = \left[\sum_{n=2}^{s-1} \frac{\rho^n}{n!} + \frac{\rho^s}{s!\,(1-\rho_s)}\right]^{-1}$$

当$p_0 \geq s$的时候，顾客要等待，记这个等待的概率为：

$$c(\mathrm{s}, \rho) = \sum_{n=s}^{\infty} p_n = \frac{\rho^s}{s!\,(1-\rho_s)} p_0$$

这个公式称为 Erlang 等待公式

下面研究系统主要的几个指标：

（1） 平均排队长

$$L_q = \sum_{n=s+1}^{\infty} (n-s) p_n = \frac{p_0 \rho^s}{s!} \sum_{n=s+1}^{\infty} (n-s) \rho_s^{n-s}$$
$$= \frac{p_0 \rho^s \rho_s}{s!\,(1-\rho_s)^2}$$

（2） 正在接受服务的顾客的平均数

$$E(s) = \sum_{n=0}^{s-1} n p_n + s \sum_{n=s}^{\infty} p_n = \rho$$

（3） 平均队列长度 L=平均排队长+平均接受服务的顾客数

$$L = L_q + \rho$$

对多台服务系统，仍有 Little 公式：

$$W = \frac{L}{\lambda}, W_q = \frac{L_q}{\lambda} = W - \frac{1}{\mu}$$

三、实际仿真

1. 仿真模型的建立

本次大作业的基本模型采用的是 M/M/S/∞，即顾客到达银行/邮局服从负指数分布，银行/邮局中分别有M_1/M_2个柜台，每个柜台服务时长服从负指数分布，排队长度可以到无穷，并且没有顾客离开，具体参数如下表：

	银行	邮局
顾客到达参数	λ_1	λ_2
柜台数	M_1	M_2
柜台服务参数	μ_1	μ_2
初始排队人数	L_1	L_2

2. 程序编写解析

2.1 程序总流程

因为是对整个排队过程进行仿真，因此需要对整个排队过程进行模拟。假设先去地点 1，地点 1 中已经有L_1长度的队伍，顾客加入队尾，得到在地点 1 的总用时，即在地点 1 用时=排队时间+接受服务时间。得到在地点 1 的总用时T_1之后，需要计算在时间T_1中地点 2 的队伍长度变化，得到地点 2 的队伍长度，再计算出在地点 2 的用时T_2，则用户的总用时即为$\mathrm{T} = T_1 + T_2$。具体代码如下：

```
function time = count(miu_1,miu_2,lambda,L_1,L_2,M_1,M_2)
    time1 = get_time1(miu_1,L_1,M_1);%得到在地点1排队的时间T
    num2 = get_num2(lambda,miu_2,L_2,M_2,time1);%模拟在T时间内地点2队伍变化
    if num2 >= 0     %如果地点2的队列长度大于0
        time2 = get_time1(miu_2,num2,M_2);
    else             %如果地点2没有人排队
        time2 = exprnd(1/miu_2);
    end
    time = time1 + time2;%总用时为两个地点用时之和
end
```

如上图所示，get_time1 函数是给定服务速率μ、队列长度 L、柜台数 M，返回队列第 L+1 位顾客所用时间，即新到的顾客排队等待时间和服务时间之和；get_num2 函数是模拟在时间 time1 中，地点 2 队列变化情况，返回值为 T 时刻地点 2 的队列长度；如果队列长度大于 0，即当前有人在地点 2 排队，则调用 get_time1 函数进行计算在地点 2 所用时间 time2，如果队列长度小于 0，则说明当前没有人排队，即有柜台处于空闲阶段，因此顾客可以直接去接收服务，time2 即为接收服务的时间；最后总用时 time 就等于两个地点的时间之和（time1+time2）。下面对各个模块进行解析。

2.2 get_time1 函数

get_time1 函数是给定服务速率μ、队列长度 L、柜台数 M，返回队列第 L+1 位顾客所用时间。这里做了一个近似，即一共有 M 个柜台，因此等效为参数为$M\mu$的泊松分布，每一位顾客服务的时间服从参数为$M\mu$的负指数分布。

```
function time = get_time1(miu,L,M)
    num = M + L + 1;%当前地点1的总人数
    t = 0;          %时间
    for i = 1:1:num %对每一位顾客进行操作
        t = t + exprnd(1/miu/M);
    end
    time = t;
end
```

以上是一个近似，也是最后采用的方法，在最开始的时候，我自己编写了实际模拟在地点 1 排队过程的代码，思路如下：开一个 M*200 的数组，进行初始化（即正在服务的人所需要的时间），然后生成队列中每一个人所需要的服务时间，对队列中的第一个人，选择当前柜台所用时间之和最小的柜台，即对 M*200 的数组按行进行求和，然后队列中第一个人加入和最小值的那一行，一直到最后一个人（第 L+1 个人），最后对第 L+1 个人所在行进行求和，其值即为顾客

在该地点的用时，不过该方法在实际仿真的时候，结果不太理想，所以没有采用，代码如下：

```
function time = get_time(miu,L,M)
    serve = zeros(M,200);
    for i = 1:1:M %初始化
        serve(i,1) = exprnd(1/miu);
    end
    serve_L_t = exprnd(miu,1,L+1);%生成队列中每一个人所需要的服务时间
    flag = -1; %用于记录第L+1个人所在行号
    for i = 1:1:length(serve_L_t)
        serve_sum = sum(serve,2); %按行求和
        [m,min_index] = min(serve_sum); %求得行的最小值和序号
        add_index = sum(serve(min_index,:)>0);  %确定新加入元素的列序号
        serve(min_index,add_index+1) = serve_L_t(i);%将队列第一个人放入服务矩阵
        if i == length(serve_L_t)
            flag = min_index;
        end
    end
    sum_t = sum(serve,2);%按航求和
    time = sum_t(flag); %求得第L+1个人所用时间
end
```

2.3 get_num2 函数

get_num2 函数是模拟在时间 time1 中，地点 2 队列变化情况，返回值为 T 时刻地点 2 的队列长度。基本的思想如下：利用指数分布的无记忆性，即固定时刻 t，一位顾客结束服务离开的概率和新来一位顾客的概率和之前无关。顾客来的时间间隔是参数为λ的负指数分布，服务等效为$M\mu$的负指数分布（当队列长度为负值，则有柜台空闲，这时服务参数会减小），当下一位顾客到来需要时间小于下一位顾客结束服务时间，则队列长度+1，否则，队列长度-1。具体代码如下：

```
function num = get_num2(lambda,miu,L,M,time)
    t = 0;              %时间
    len = L;            %队列长度
    flag = 0;           %记录是新来顾客还是服务结束走一个顾客
    while(t < time)
        if len >= 0 %当队列长度大于0，说明所有柜台都在服务
            cus_t = exprnd(1/lambda);
            serve_t = exprnd(1/miu/M);
        elseif len >= -M && len < 0 %当队列长度为负数，即有柜台空闲，服务效率会变化
            cus_t = exprnd(1/lambda);
            serve_t = exprnd(1/miu/(M+len));%更新参数后的分布
        end
        seed = cus_t < serve_t;
        if seed == 1            %新来顾客所用时间小于服务所用时间时，队列长度+1
            flag = 1;
            len = len + 1;
            t = t + cus_t;
        else                    %%新来顾客所用时间大于服务所用时间时，队列长度+1
            flag = 2;
            len = len - 1;
            t = t + serve_t;
        end
    end
    %因为是while循环，因此会多+1或-1，因此需要修正
    if flag == 1
        len = len - 1;
    else
        len = len + 1;
    end
    num = len;
end
```

3. 仿真结果

3.1 仿真思路

仿真时，固定其他参数，研究一对参数的变化对决策边界的影响，比如研究在其他参数不变的情况下，队列长度L_1, L_2对决策边界的影响，为了更好地呈现出决策边界，在结果呈现的时候，用二值图的形式呈现出来。

3.2 队列长度 L 对决策边界的影响

设定参数：

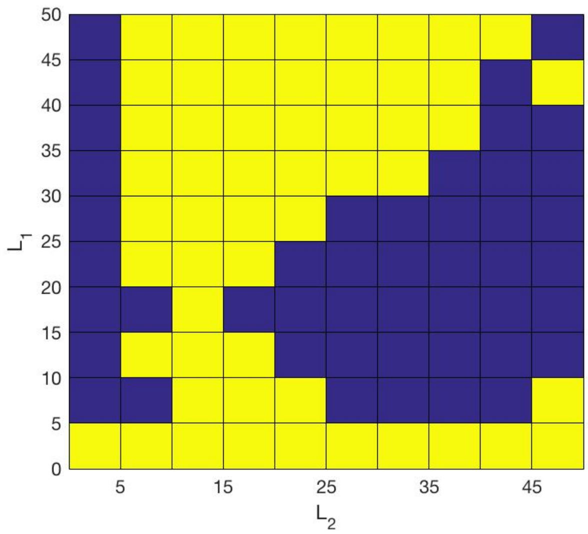
$$M_1 = M_2 = 3$$

$$\lambda_1 = \lambda_2 = 1$$

$$\mu_1 = \mu_2 = 0.4$$

$$L_1 = 0:5:50, L_2 = 0:5:50$$

即L_1和L_1在 0-50 之间变化，取样间隔为 5，结果如下：



其中，黄色代表先去地点 1，蓝色代表先去地点 2，由图可以看出，当L_1, L_2比较大的时候，$L_1 = L_2$是决策边界，固定L_1的时候，如果$L_1 < L_2$则先去L_2，如果$L_2 < L_1$则先去L_1，因为这时候的参数，是$\lambda < M \times \mu$，因此服务的速度是大于顾客到来的速度，选择排长的队伍，所用时间比较长，队列 1 缩短的长度也就更长；在队列比较短的时候，比如$L_1 \ll L_2$，则先去L_1，因为L_1太短，先排L_2的话，L_1的长度并不会减少很多就到 0，故此应选择去L_1。

下面看一下$\lambda > M \times \mu$的情况，参数如下：

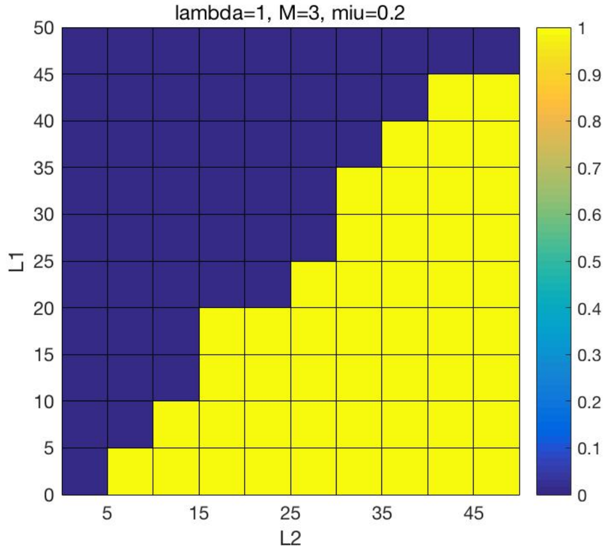
$$M_1 = M_2 = 3$$

$$\lambda_1 = \lambda_2 = 1$$

$$\mu_1 = \mu_2 = 0.2$$

$$L_1 = 0{:}5{:}50, L_2 = 0{:}5{:}50$$

结果如下：



可以看到，当$\lambda > M\times\mu$的时候，即服务速率小于顾客到达速率，队列是会一直增加的，因此当$L_1 < L_2$的时候，应该选择排L_1，如果排L_2，L_1的队伍会越来越长，得不偿失。

3.3 参数μ对决策边界的影响

参数如下：

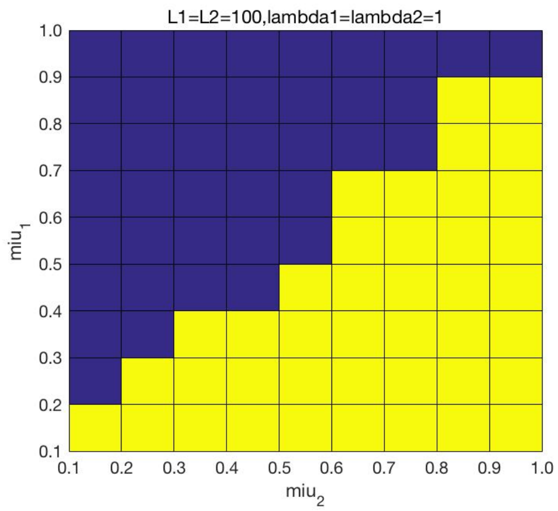
$$M_1 = M_2 = 3$$

$$\lambda_1 = \lambda_2 = 1$$

$$\mu_1 = \mu_2 = 0.1: 0.1: 1$$

$$L_1 = L_2 = 100$$

结果如下：



从上图看出，当初始排队长度相等，其余参数都相等的时候，$\mu_1 = \mu_2$为决策边界，当$\mu_1 < \mu_2$的时候，应该先去地点 1，因为地点 2 的服务速率更高，所以先去地点 1，地点 2 减少的人数更多，当$\mu_1 > \mu_2$的时候，应该先去地点 1，这在直观上是可以理解的。

3.4 参数λ对决策边界的影响

参数如下：

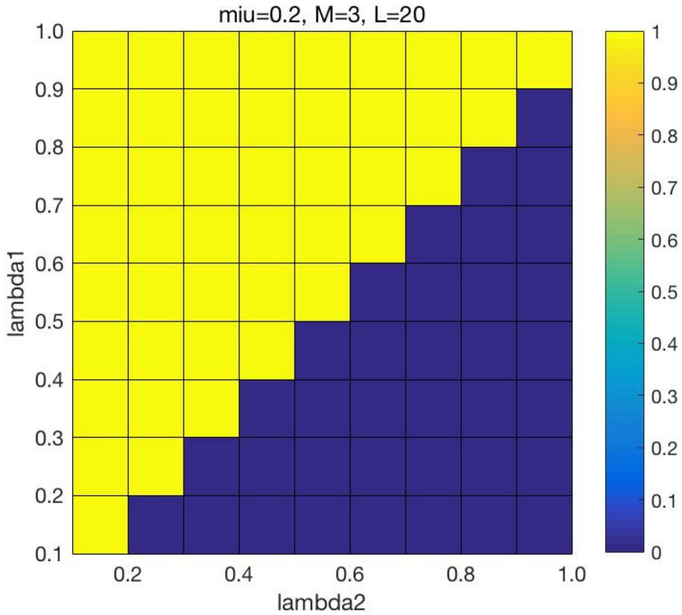
$$M_1 = M_2 = 3$$

$$\lambda_1 = \lambda_2 = 0.1 : 0.1 : 1$$

$$\mu_1 = \mu_2 = 0.2$$

$$L_1 = L_2 = 20$$

结果如下：



由上图可以看到，当其他参数固定的时候，当$\lambda_1 < \lambda_2$的时候，应该选择去地点 2，因为地点 1 顾客到的速率更小，增加的人数会更少；当$\lambda_1 > \lambda_2$的时候，因该先去地点 1，这在直观上也是可以理解的。

3.5 小节

因为该模型有很多个参数，每一组参数都会有不同的边界条件，这里通过控制变量法，对相同类型的参数进行研究，从而得到参数对边界选择的影响。

四、试验总结

本次试验是关于排队论的，课堂上只讲了 M/M/1 的情况，但大作业用到的是 M/M/S 的模型，因此通过查阅资料自学了该模型，并且在编写大作业仿真的时候，本来是想对整个排队过程进行详细的模拟，但后来发现难度比较大，因此就采取了一点近似的方法，但对边界选择的影响是不大的。

在本次大作业中，对排队论有了更深刻的认识，本模型只是采用了 M/M/S/∞，而实际上还有更多的模型，比如排队长度不是无穷的，当排队长度到了一定数量，会有顾客离开，不过限于时间的限制，没有编写该模型。

因为自己选课的原因，这学期一共写了 8 个大作业，有三个大作业都是在期末的时候，因此压力特别大，所以本次大作业中有一些比较粗糙的地方，还请谅解，在进行研究参数对选择边界的影响的时候，只研究了相同类型参数对选择边界的影响，而没有研究不同类型参数对选择边界的影响，大作业的完整性还差一点，但确实迫于其他大作业和期末考试的压力，没有去仿真，在此说一声抱歉。

最后，感谢老师和助教对本课程的付出，也感谢助教用心批改每一个小作业和大作业！