МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Вычислительная техника»

Дисциплина «Исследование операций»

**Лабораторная работа №3.**

**Система массового обслуживания**

**многоканальная модель с ожиданием (очередью)**

Выполнил:

студент группы ИВТАПбд-31

Кондратьев П. С.

Проверил:

Фролов В. А.

Ульяновск, 2018

**Система массового обслуживания**

В качестве показателей эффективности СМО с ожиданием, кроме уже известных показателей — абсолютной A

и относительной Q

пропускной способности, вероятности отказа Potk

, среднего числа занятых каналов к (для многоканальной системы) будем рассматривать также следующие:

1) Lsist — среднее число заявок в системе;

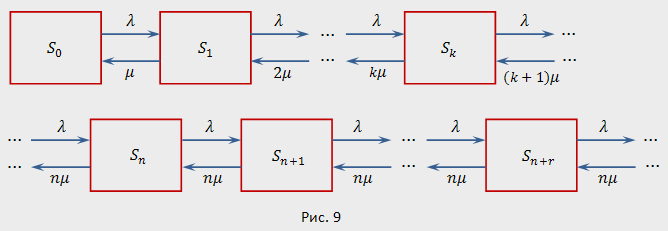
2) Tsist — среднее время пребывания заявки в системе;

3) Loch — среднее число заявок в очереди (длина очереди);

4) Toch — среднее время пребывания заявки в очереди;

5) Pzan — вероятность того, что канал занят (степень загрузки канала)..

Граф состояний системы показан на рис. 9. Обратим внимание на то, что в отличие от предыдущей СМО, интенсивность потока обслуживании (переводящего систему из одного состояния в другое справа налево) не остается постоянной, а по мере увеличения числа заявок в СМО от 0 до n увеличивается от величины μ до nμ , так как соответственно увеличивается число каналов обслуживания. При числе заявок в СМО большем, чем n , интенсивность потока обслуживании сохраняется равной nμ



**Пример Задачи:**

В универсаме к узлу расчета поступает поток покупателей с интенсивностью λ=81λ=81чел. в час. Средняя продолжительность обслуживания контролером-кассиром одного покупателя t¯ob.=2t¯ob.=2 мин. Определить:

**а.** Минимальное количество контролеров-кассиров nminnmin, при котором очередь не будет расти до бесконечности, и соответствующие характеристики обслуживания при n=nminn=nmin.

**б.** Оптимальное количествоnopt.nopt.контролеров-кассиров, при котором относительная величина затрат Cotn.Cotn., связанная с издержками на содержание каналов обслуживания и с пребыванием в очереди покупателей, задаваемая, например, как Cotn.=nλ+3Toch.Cotn.=nλ+3Toch, будет минимальна, и сравнить характеристики обслуживания при n=nminn=nmin

и n=nopt.n=nopt.

.**в.** Вероятность того, что в очереди будет не более трех покупателей.

***Решение.***а. По условию

λ=81λ=81(1/ч) =8160=1,35=8160=1,35(1/мин.).

ρ=λμ=λt¯ob.=1,35⋅2=2,7ρ=λμ=λt¯ob.=1,35⋅2=2,7

. Очередь не будет возрастать до бесконечности при условии ρn<1ρn<1, т.е. при n>ρ=2,7n>ρ=2,7

. Таким образом, минимальное количество контролеров-кассиров nmin=3 nmin=3

Найдем характеристики обслуживания СМО при n=3n=3.

Вероятность того, что в узле расчета отсутствуют покупатели

p0=(1+2,7+2,722!+2,733!+2,743!(3−2,7))−1=0,025p0=(1+2,7+2,722!+2,733!+2,743!(3−2,7))−1=0,025 т.е в средне 2,5% времени контролеры-кассиры будут простаивать.

Вероятность того, что в узле расчета будет очередь

Poch.=2,743!⋅(3−2,7)⋅0,025=0,735.Poch.=2,743!⋅(3−2,7)⋅0,025=0,735.

Среднее число покупателей, находящихся в очереди

Loch.=2,743⋅3!⋅(1−2,7/3)2⋅0,025=7,35.Loch.=2,743⋅3!⋅(1−2,7/3)2⋅0,025=7,35.

Среднее время ожидания в очереди

Toch.=7,351,35=5,44Toch.=7,351,35=5,44(мин).

Среднее число покупателей в узле расчета

Lsist.=7,35+2,7=10,05.Lsist.=7,35+2,7=10,05.

Среднее время нахождения покупателей в узле расчета

Tsist.=10,051,35≈7,44Tsist.=10,051,35≈7,44(мин).

Среднее число контролеров-кассиров, занятых обслуживанием покупателей, k¯¯¯=2,7k¯=2,7.

Коэффициент (доля) занятых обслуживанием контролеров-кассиров k¯otn.=ρn=2,73=0,9k¯otn.=ρn=2,73=0,9.

Абсолютная пропускная способность узла расчета

A=1,35A=1,35 (1/мин), или 81 (1/ч), т.е. 81 покупатель в час.

Анализ характеристик обслуживания свидетельствует о значительной перегрузке узла расчета при наличии трех контролеров-кассиров.

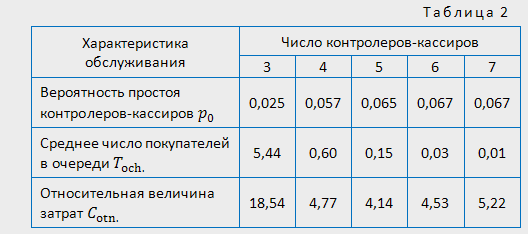
**б.** Относительная величина затрат при

n=3n=3

Cotn.=nλ+3Toch.=31,35+3⋅5,44=5,44.Cotn.=nλ+3Toch.=31,35+3⋅5,44=5,44.

Рассчитаем относительную величину затрат при других значениях

nn (табл. 2).



Как видно из табл. 2, минимальные затраты получены при

n=nopt.=5n=nopt.=5

контролерах-кассирах.

Определим характеристики обслуживания узла расчета при

n=nopt.=5n=nopt.=5

. Получим

Poch.=0,091; Loch.=0,198; Toch.=0,146; Lsist.=2,9; Tsist.=2,15; k¯¯¯=2,7; k3=0,54.Poch.=0,091; Loch.=0,198; Toch.=0,146; Lsist.=2,9; Tsist.=2,15; k¯=2,7; k3=0,54.

Как видим, при n=5n=5 по сравнению с n=3n=3 существенно уменьшились вероятность возникновения очереди

Poch.Poch., длина очереди

Loch.Loch.и среднее время пребывания в очереди

Toch.Toch., и соответственно среднее число покупателей

Lsist.Lsist.и среднее время нахождения в узле расчета

Tsist.Tsist., а также доля занятых обслуживанием контролеров

k3k3

. Но среднее число занятых обслуживанием контролеров-кассиров

k¯¯¯k¯и абсолютная пропускная способность узла расчета

AA естественно не изменились.

**в.** Вероятность того, что в очереди будет не более 3 покупателей, определится как

P{r⩽3}=p1+p2+p3+p4+p5+p5+1+p5+2+p5+3=1−Poch.+p5+1+p5+2+p5+3,P{r⩽3}=p1+p2+p3+p4+p5+p5+1+p5+2+p5+3=1−Poch.+p5+1+p5+2+p5+3,

n=5:n=5:

P{r⩽3}=1−2,765!(5−2,3)⋅0,065+2,765⋅5!⋅0,065+2,7752⋅5!⋅0,065+2,7853⋅5!⋅0,065=0,986.P{r⩽3}=1−2,765!(5−2,3)⋅0,065+2,765⋅5!⋅0,065+2,7752⋅5!⋅0,065+2,7853⋅5!⋅0,065=0,986.

(Заметим, что в случае n=3n=3 контролеров-кассиров та же вероятность существенно меньше:

P{r⩽3}=0,464P{r⩽3}=0,464

).

**Исходный код**

let λ = 81 / 60, // поток покупателей с интенсивностью в час

t\_ob = 2, // Средняя продолжительность обслуживания контролером-кассиром одного покупателя в мин

p,

p\_och,

l\_och, //среднее число заявок в очереди (длина очереди)

t\_och, //среднее время пребывания заявки в очереди

l\_sist, //среднее число заявок в системе

t\_sist, //среднее время пребывания заявки в системе

k, //Среднее число контролеров-кассиров, занятых обслуживанием покупателей

k\_otn, //Коэффициент (доля) занятых обслуживанием контролеров-кассиров

c\_otn,

i,

n; //Очередь не будет возрастать до бесконечности при условии ρ/n<1

function factorial(n) {

return n ? n \* factorial(n - 1) : 1;

}

let F = function(p, n) {

let fp0 = 1; // Вероятность того, что в узле расчета отсутствуют покупатели

for (let i = 0; i <= n; i++) {

fp0 += Math.pow(p, i) / factorial(i);

}

fp0 += Math.pow(p, n + 1) / (factorial(n) \* (n - p));

return fp0 = Math.pow(fp0, -1);

};

let Step\_one = function() {

console.log(`\nМинимальное количество контролеров-кассиров nmin, при котором очередь не будет расти до бесконечности, и соответствующие характеристики обслуживания при n = nmin\n`);

p = λ \* t\_ob;

console.log(`-> Очередь не будет возрастать до бесконечности при условии ρ/n<1, где p = ${p}`);

n = Math.round(p);

console.log(`-> Таким образом, минимальное количество контролеров-кассиров nmin = ${n}`);

let p0 = F(p, n);

console.log(`-> Вероятность того, что в узле расчета отсутствуют покупатели = ${parseFloat(p0.toFixed(3))} т.е. в среднем ${parseFloat((p0 \* 100).toFixed(1))}% времени контролеры-кассиры будут простаивать.`);

p\_och = (Math.pow(p, n + 1) / (factorial(n) \* (n - p))) \* p0;

console.log(`-> Вероятность того, что в узле расчета будет очередь = ${parseFloat(p\_och.toFixed(3))}`);

l\_och = ((Math.pow(p, n + 1) \* p0) / (n \* factorial(n))) \* Math.pow(1 - (p / n), -2);

console.log(`-> Среднее число покупателей, находящихся в очереди = ${parseFloat(l\_och.toFixed(2))}`);

t\_och = (1 / λ) \* l\_och;

console.log(`-> Среднее время ожидания в очереди = ${parseFloat(t\_och.toFixed(2))} (мин)`);

l\_sist = l\_och + p;

console.log(`-> Среднее число покупателей в узле расчета = ${parseFloat(l\_sist.toFixed(2))}`);

t\_sist = (1 / λ) \* l\_sist;

console.log(`-> Среднее время нахождения покупателей в узле расчета = ${parseFloat(t\_sist.toFixed(2))} (мин)`);

k = p;

console.log(`-> Среднее число контролеров-кассиров, занятых обслуживанием покупателей = ${k}`);

k\_otn = p / n;

console.log(`-> Коэффициент (доля) занятых обслуживанием контролеров-кассиров = ${parseFloat(k\_otn.toFixed(2))}`);

console.log(`-> Абсолютная пропускная способность узла расчета A = ${λ} (1/мин), или ${λ \* 60} (1/ч), т.е. ${λ \* 60} покупатель в час.`);

};

let Step\_two = function() {

let temp, temp\_p, min\_c\_otn;

i = n;

console.log(`\nОптимальное количество n\_opt контролеров-кассиров, при котором относительная величина затрат C\_otn, связанная с издержками на содержание каналов обслуживания и с пребыванием в очереди покупателей, задаваемая, например, как C\_otn = n / λ + 3T\_och, будет минимальна, и сравнить характеристики обслуживания при n = nmin и n = n\_opt\n`);

do {

temp\_p = F(p, i);

console.log(`-> Вероятность простоя контройлеров-кассиров p0 при n = ${i} равна ${parseFloat(temp\_p.toFixed(2))}`);

t\_och = (1 / λ) \* ((Math.pow(p, i + 1) \* temp\_p) / (i \* factorial(i))) \* Math.pow(1 - (p / i), -2);

console.log(`-> Среднее число покупателей в очереди T\_och при n = ${i} равна ${parseFloat(t\_och.toFixed(2))}`);

temp = (i / λ) + (3 \* t\_och);

console.log(`-> Относительная величина затрас C\_otn при n = ${i} равна ${parseFloat(temp.toFixed(2))}\n`);

i++;

temp\_p = F(p, i);

console.log(`-> Вероятность простоя контройлеров-кассиров p0 при n = ${i} равна ${parseFloat(temp\_p.toFixed(2))}`);

t\_och = (1 / λ) \* ((Math.pow(p, i + 1) \* temp\_p) / (i \* factorial(i))) \* Math.pow(1 - (p / i), -2);

console.log(`-> Среднее число покупателей в очереди T\_och при n = ${i} равна ${parseFloat(t\_och.toFixed(2))}`);

c\_otn = (i / λ) + (3 \* t\_och);

console.log(`-> Относительная величина затрас C\_otn при n = ${i} равна ${parseFloat(c\_otn.toFixed(2))}\n`);

} while(c\_otn < temp)

i--;

console.log(`Как видно из вычислениий минимальные затраты получены при n = n\_opt = ${i} контролерах-кассирах.`);

console.log(`Определим характеристики обслуживания узла расчета при n = n\_opt = ${i} Получим`);

p\_och = (Math.pow(p, i + 1) / (factorial(i) \* (i - p))) \* F(p, i);

console.log(`-> Вероятность того, что в узле расчета будет очередь = ${parseFloat(p\_och.toFixed(3))}`);

l\_och = ((Math.pow(p, i + 1) \* F(p, i)) / (i \* factorial(i))) \* Math.pow(1 - (p / i), -2);

console.log(`-> Среднее число покупателей, находящихся в очереди = ${parseFloat(l\_och.toFixed(2))}`);

t\_och = (1 / λ) \* l\_och;

console.log(`-> Среднее время ожидания в очереди = ${parseFloat(t\_och.toFixed(2))} (мин)`);

l\_sist = l\_och + p;

console.log(`-> Среднее число покупателей в узле расчета = ${parseFloat(l\_sist.toFixed(2))}`);

t\_sist = (1 / λ) \* l\_sist;

console.log(`-> Среднее время нахождения покупателей в узле расчета = ${parseFloat(t\_sist.toFixed(2))} (мин)`);

k = p;

console.log(`-> Среднее число контролеров-кассиров, занятых обслуживанием покупателей = ${k}`);

stroke\_k = F(p, i);

console.log(`-> Среднее число контролеров-кассиров, занятых обслуживанием покупателей = ${parseFloat(stroke\_k.toFixed(2))}`);

console.log(`Как видим, при n = ${i} по сравнению с n = ${Math.round(λ \* t\_ob)} существенно уменьшились вероятность возникновения очереди P\_och, длина очереди L\_och и среднее время пребывания в очереди Toch, и соответственно среднее число покупателей L\_sist и среднее время нахождения в узле расчета T\_sist , а также доля занятых обслуживанием контролеров k3. Но среднее число занятых обслуживанием контролеров-кассиров stroke\_k и абсолютная пропускная способность узла расчета A естественно не изменились.`);

};

let Step\_three = function() {

console.log(`\nВероятность того, что в очереди будет не более 3 покупателей, определится как\n`);

let P\_r, P\_rn, index = n;

P\_rn = 1 - ((Math.pow(p, index + 1) / (factorial(index) \* (index - p))) \* F(p, index)) + ((Math.pow(p, index + 1) / (factorial(index) \* index)) \* F(p, index)) + (Math.pow(p, index + 2) / (factorial(index) \* Math.pow(index, 2)) \* F(p, index)) + (Math.pow(p, index + 3) / (factorial(index) \* Math.pow(index, 3)) \* F(p, index));

console.log(`-> В случае n = ${index} вероятность = P{r ⩽ ${index}} = ${parseFloat(P\_rn.toFixed(3))}.`);

index = i;

P\_r = 1 - ((Math.pow(p, index + 1) / (factorial(index) \* (index - p))) \* F(p, index)) + ((Math.pow(p, index + 1) / (factorial(index) \* index)) \* F(p, index)) + (Math.pow(p, index + 2) / (factorial(index) \* Math.pow(index, 2)) \* F(p, index)) + (Math.pow(p, index + 3) / (factorial(index) \* Math.pow(index, 3)) \* F(p, index));

console.log(`-> В случае n = ${index} вероятность = P{r ⩽ ${index}} = ${parseFloat(P\_r.toFixed(3))}.`);

console.log(`(Заметим, что в случае n = ${n} контролеров-кассиров та же вероятность существенно меньше: P{r ⩽ ${n}} = ${parseFloat(P\_rn.toFixed(3))}).`);

};

let main = function() {

console.log(`\n---------------------------------------------------------------------------------------------------------------------------------------------------`);

console.log(`Задание а`);

Step\_one();

console.log(`\n---------------------------------------------------------------------------------------------------------------------------------------------------`);

console.log(`Задание б`);

Step\_two();

console.log(`\n---------------------------------------------------------------------------------------------------------------------------------------------------`);

console.log(`Задание с`);

Step\_three();

console.log(`\n---------------------------------------------------------------------------------------------------------------------------------------------------`);

};

main();