**Содержание**

Введение

1.Общая характеристика СМО

2.Исходные данные для создания СМО

3. Построение алгоритма имитационной модели. Машинная программа модели

3.1 Построение алгоритма имитационной модели

3.2 Машинная программа модели

4. Описание функционирования математической модели

5. Анализ результатов испытаний

Заключение

Список использованных источников

# Введение

модель имитационная система массового обслуживания

*Система массового обслуживания* – совокупность последовательно связанных между собой входящих потоков требований на обслуживание (машин, самолетов, пользователей и т.д.), накопителей, очередей, каналов обслуживания (станций техобслуживания, аэродромов, ЭВМ и т.д.) и выходящих потоков требований после обслуживания.

*Имитационная СМО* – модель, отражающая поведение системы и изменения ее состояния во времени при заданных потоках требований, поступающих на входы системы.

Выходными параметрами являются величины, характеризующие свойства системы – качество ее функционирования, - например такие, как [3]:

* коэффициенты использования каналов обслуживания;
* максимальная и средняя длина очередей в системе;
* время нахождения требований в очередях и каналах обслуживания.

Модели массового обслуживания находят широкое применение при исследовании надежности технических систем, организации их эксплуатации и использования по назначению, а также при анализе и синтезе автоматизированных систем управления. СМО – это предприятие, выполняющее заказы; станок, обрабатывающий детали, компьютер, решающий задачи; магазин, обслуживающий покупателей, и т.д. Имитационное моделирование дает возможность исследовать большие системы и сложные ситуации во всей их полноте, а также реально невоспроизводимые ситуации и объединять в процессе моделирования формальные и неформальные методы исследования. Под имитационной моделью понимают алгоритмическое описание со всей доступной для исследования полнотой изучаемой системы процесса ее функционирования.

# Общая характеристика СМО

Основными признакамиреальной системы, позволяющими рассматривать ее как своеобразную СМО, являются:

* наличие объектов, нуждающихся в случайные моменты времени в обслуживании; эти объекты порождают так называемый входящий поток заявок на обслуживание;
* наличие объектов, которые производят обслуживание и называются обслуживающими приборами (каналами);
* возникновение задержек в обслуживании (образование очереди).

В качестве своеобразных СМО могут рассматриваться:

* системы связи и ремонта;
* пункты технического обслуживания;
* вычислительные центры и отдельные ЭВМ: автоматизированные производственные цехи, поточные линии;
* транспортные системы;
* системы материального обеспечения.

***Обслуживающий прибор (канал)*** – материальный объект или совокупность объектов, одновременно участвующих в обслуживании заявки. В каждый момент времени прибор может обслуживать только одну заявку.

СМО бывают:

* с отказами – если все каналы заняты – заявка покидает систему;
* с ожиданием – заявка, поступившая в систему, становится в очередь;

СМО с конечной очередью характеризуется тем, что при поступлении очередной заявки возможно:

* заявка немедленно принимается на обслуживание, если в системе находится k-заявок и k < n, где n – число каналов;
* заявка становится в очередь, если n ≤ k < n + m;
* заявка получает отказ и покидает систему, если k = n + m.

Следовательно, в любой момент времени система может находиться в одном из состояний.

# Исходные данные для создания СМО

В рассматриваемом примере СМО представим как парикмахерскую. Из Таблицы исходных данных видно, что в данной системе работают три парикмахера, интенсивность поступления клиентов 7 чел/час, время работы парикмахерской 7 часов. Максимальная длина очереди 4 человека. Закон распределения времени обслуживания – экспоненциальный, это означает, что интервалы между заявками могут быть как очень короткими, так и очень длинными. [4]

Таблица исходных данных

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Траб | λ | n | m |  | ЗР | ε |
| 7 | 7 | 3 | 4 | 0,5 | exp | 0,1 |

*Траб* – время работы СМО, час;

*λ* – интенсивность поступления заявок, ед./час;

*n* – число обслуживающих каналов, ед.;

*m* – максимальная длина очереди, ед.;

 - среднее время обслуживания, час;

*ЗР* – закон распределения времени обслуживания, exp – экспоненциальный;

*ε* – погрешность вычислений.

Существует некоторый поток заявок (клиентов), объектов, нуждающихся в обслуживании. Также в наличии обслуживающие приборы (парикмахеры), каналы, осуществляющие обслуживание, и задержка в обслуживании (наличие очереди).

# Построение алгоритма имитационной модели. Машинная программа модели

Построение алгоритма решения и нахождения решения с его помощью относится к компетенции специалистов-математиков и программистов.

Имеется большое количество разработанных алгоритмов для различных моделей. Во многих случаях удается свести решение к готовому алгоритму. В противном случае приходится строить новый.

# *Построение алгоритма имитационной модели*

При управлении техническими системами известны условия протекания процессов в системе и предусмотрены способы нормализации поведения системы в зависимости от возмущений. Поэтому процесс управления может быть алгоритмизирован и, следовательно, автоматизирован до конца.

Для создания машинной программы СМО необходимо построить алгоритм ее выполнения Рис. 3.1.1.

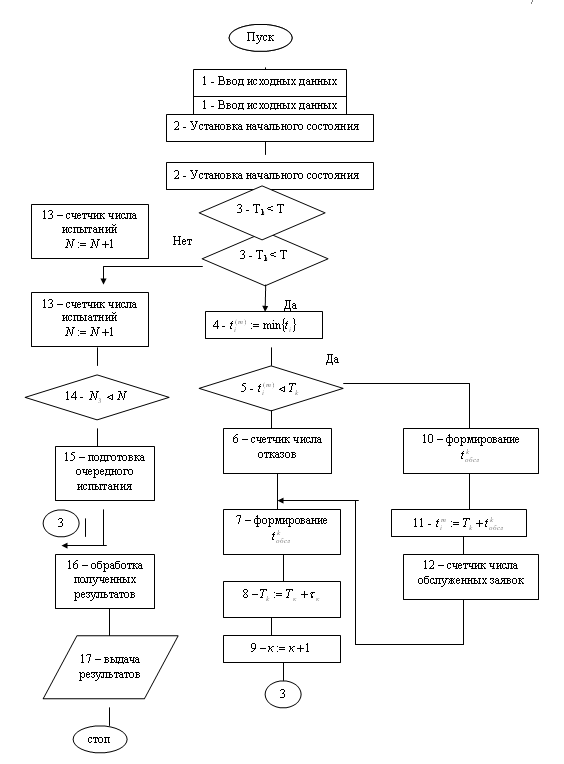


Рис.3.1.1 [1]

Оператор 1 – осуществляет ввод исходной информации: число каналов, параметры законов распределения потока заявок, время работы системы, заданное число испытаний;

оператор 2 – устанавливает начальное состояние системы Т1=0, t1= t2= t3=0 и k=0;

оператор 3 – определяет принадлежит ли k-я заявка заданному интервалу времени Т;

оператор 4 – сравнивает между собой моменты освобождения каналов системы и выбирает канал, освободившийся раньше всех;

оператор 5 – сравнивает момент освобождения выбранного канала  с моментов поступления k-й заявки;

оператор 6 – счетчик числа отказов, после каждого отказа показание счетчика увеличивается на единицу;

оператор 7 – формируется поток заявок, определяет интервал времени между двумя последовательными заявками;

оператор 8 – формирует момент времени поступления следующей заявки;

оператор 9 – формирует номер очередной заявки;

оператор 10 – определяет время обслуживания k-й заявки;

оператор 11 – вычисляет время освобождения  канала с номером m;

оператор 12 – счетчик числа обслуженных заявок, после каждой обслуженной заявки показание счетчика увеличивается на единицу;

оператор 13 – счетчик числа испытаний;

оператор 14 – проверяет, получено ли уже заданное число испытаний, если  выполняется, управление передается оператору 15;

оператор 15 – подготовка следующего испытания;

оператор 16 – осуществляет статистическую обработку полученных результатов и вычисляет требуемые показатели эффективности функционирования системы за время Т.

# *Машинная программа модели*

Программа имитационной модели построена на языке Perl.

open(OUT,">output.txt");

print OUT "СИСТЕМНОЕ МОДЕЛИРОВАНИЕ\n\n";

if (not !$ARGV[0])

{

print OUT "Имя файла данных: $ARGV[0]\n\n";

if (-e $ARGV[0])

{

open(DATA,"<$ARGV[0]");

while (<DATA>)

{

$\_=~s/\n//gi;

$\_=~s/\t//gi;

($\_,$comments)=split("#",$\_);

($q1,$q2)=split("=", $\_);

if ($q1 eq 'T') {print OUT "время работы СМО, час [T]: $q2\n"; $T=$q2;}

if ($q1 eq 'L') {print OUT "интенсивность поступления заявок, ед./час [L]: $q2\n"; $L=$q2;}

if ($q1 eq 'N') {print OUT "число обслуживающих каналов, ед. [N]: $q2\n"; $N=$q2;}

if ($q1 eq 'M') {print OUT "максимальная длина очереди, ед. [M]: $q2\n"; $M=$q2;}

if ($q1 eq 'ZR') {print OUT "закон распределения времени обслуживания(exp/evenly) [ZR]: $q2\n"; $ZR=$q2;}

if ($q1 eq 'E') {print OUT "погрешность вычислений [E]: $q2\n"; $E=$q2;}

if ($q1 eq 'TO'){

print OUT "среднее время обслуживания [TO]: ";

if ($ZR eq 'evenly')

{

$q4 = 0;

while (length($q2) > 0)

{

($TO[$q4], $q2)=split(";", $q2);

$q4 = $q4 + 1;

}

$q6=1/$q4\*100;

for ($q5=0;$q5<$q4;$q5++) # $q4 кол-во вариантов времени обслуживания

{

if ($q5>0) {print OUT " ; ";}

print OUT "$TO[$q5] ($q6%)";

}

}

if ($ZR eq 'exp')

{

$TO = $q2;

print OUT " $TO";

}

print OUT "\n";

}

}

close(DATA);

print OUT "\nНеобходимое кол-во экспериментов F=(T\*T)/(E\*E\*L\*L) при погрешности вычислений E=$E : ";

$F=int(($T\*$T)/($L\*$L\*$E\*$E)) + 1; # округляем до целого и прибавляем единицу

print OUT "$F \n\n\n";

print OUT "z - номер заявки\n";

print OUT "vp - время прихода заявки\n";

print OUT "vno - время начала обслуживания заявки\n";

print OUT "vko - время конца обслуживания заявки\n";

print OUT "kanal - номер обслуживающего канала\n";

print OUT "vob - время обслуживания заявки\n";

print OUT "nvo - номер в очереди, 0 - очереди нет\n";

print OUT "voo - время ожидания обслуживания (сколько стоит в очереди)\n\n\n\n";

$srkanal = 0;

$skz = 0; # среднее кол-во заявок

$sotkaz = 0; # среднее кол-во отказов в обслуживании

$svpr = 0; # среднее время простоя

$ssvo = 0; # среднее время ожидания

$smaxvoo = 0; # среднее максимальное время ожидания

$sr\_och = 0; # общее время ожидания, если умножить на 1, то получится площадь очереди.

$svrobsl = 0; # общее время обслуживания за все прогоны

for ($q7=1; $q7<=$F; $q7++) # выполняем необходимое кол-во прогонов модели

{

print OUT "\n\n эксперимент $q7 \n\n";

$o0=0; # обнуление счетчика отказов

$t0=0; # обнуление времени

$n0=0; # обнуление счетсика заявок

for ($q8 = 1; $q8 <= $N; $q8++) {$N[$q8] = 0;} # обнуления состояния каналов

for ($q8 = 1; $q8 <= $M; $q8++) {$och[$q8] = 0;} # обнуление очереди

while ($t0 < $T)

{

$r=int(rand(100001))/100000;

$tt = int(-(1/$L)\*log($r)\*1000000)/1000000;

$t0 = $t0 + $tt;

if ($t0 <= $T)

{

$n0 = $n0 + 1;

# начало проверки состояния очереди

$m0=0; # обнуление счетчика очереди

for ($o=1; $o<=$M; $o++)

{

if ($och[$o] < $t0)

{

$och[$o] = 0;

}

else

{

$m0 = $m0 + 1;

}

}

# конец проверки состояния очереди

$vp[$n0] = $t0; # время прихода заявки

# определение времени обслуживания

if ($ZR eq 'evenly') # для РАВНОМЕРНОГО РАСПРЕДЕЛЕНИЯ ВРЕМЕНИ ОБСЛУЖИВАНИЯ

{

$r=int(rand(100001))/100000;

for ($q5=0;$q5<$q4;$q5++)

{

$dn = $q5\*1/$q4;

$dk = ($q5+1)\*1/$q4;

if (($r > $dn) and ($r <= $dk))

{

$vob[$n0] = $TO[$q5];

}

}

}

if ($ZR eq 'exp') # для ЭКСПОНЕНЦИАЛЬНОГО РАСПРЕДЕЛЕНИЯ ВРЕМЕНИ ОБСЛУЖИВАНИЯ

{

$r=int(rand(100001))/100000;

$vob[$n0] = int(-$TO\*log(1-$r)\*1000)/1000;

$ww=log(1-$r);

$www=-1/$TO;

}

# конец определения времени обслуживания

# выбор канала обслуживания

$kanal[$n0] = 0;

for ($q8 = 1; $q8 <= $N; $q8++)

{

if (($N[$q8] <= $vp[$n0]) and ($kanal[$n0] == 0))

{

$vno[$n0] = $vp[$n0];

$N[$q8]=$vno[$n0]+$vob[$n0];

$kanal[$n0] = $q8;

}

}

if ($kanal[$n0] == 0) # формирование очереди

{

$m1 = $m0 + 1;

if ($m1 <= $M) # проверка длины очереди

{

$nvo[$n0] = "$m0->$m1"; # очередь для этой заявки

# > выбор канала для заявки поставленной в очередь

$kanal[$n0] = 1; # выбор первого канала для сравнения с остальными

for ($q8 = 1; $q8 <= $N; $q8++)

{

if ($N[$q8] < $N[$kanal[$n0]])

{

$kanal[$n0] = $q8;

}

}

$vno[$n0] = $N[$kanal[$n0]];

$och[$m1] = $vno[$n0]; # время очереди

$voo[$n0] = int(($N[$kanal[$n0]] - $vp[$n0])\*1000000)/1000000;

$vko[$n0] = $vno[$n0] + $vob[$n0];

$N[$kanal[$n0]] = $vko[$n0]; # сдвиг времени по данному каналу

}

else

{

$vno[$n0] = -1; # отказ в обслуживании

$vob[$n0] = 0;

$nvo[$n0] = "$m0->$m0";

$kanal[$n0] = '-';

}

}

else

{

$voo[$n0] = 0; # для данной заявки время ожидания = 0

$nvo[$n0] = 0; # для данной заявки очереди нет

$vko[$n0] = $vno[$n0] + $vob[$n0];

}

# конец выбора канала обслуживания

}

}

# ВЫВОД РЕЗУЛЬТАТОВ

$ppp = 0;

$svoo = 0;

$svob = 0;

$rabot = 0;

$otkaz = 0;

$maxvoo = $voo[1];

$minvoo = $voo[1];

$sr\_voo = 0;

$skanal[$q7] = 0;

print OUT "z | vp | vno | vko | kanal | nvo | vob | voo\n";

for ($z = 1; $z<=$n0; $z++)

{

if ($vp[$z] <= $T)

{

$ppp = $ppp + 1;

$svoo = $svoo + $voo[$z]; # считает время ожидания

$svob = $svob + $vob[$z]; # считает время обслуживания, чтобы вычитать время простоя

if ($maxvoo < $voo[$z]) {$maxvoo = $voo[$z];}

if ($minvoo > $voo[$z]) {$minvoo = $voo[$z];}

if ($vno[$z] == -1) {$otkaz = $otkaz + 1;}

if (not $kanal[$z] eq '-') {$skanal[$q7] = $skanal[$q7] + $kanal[$z];}

$aa="$z"; $aaa=length($aa); for ($aaaa=1; $aaaa < 4-$aaa; $aaaa++) {$aa="$aa ";} $z=$aa;

$aa="$vp[$z]"; $aaa=length($aa); for ($aaaa=1; $aaaa < 9-$aaa; $aaaa++) {$aa="$aa ";} $vp=$aa;

$aa="$vno[$z]"; $aaa=length($aa); for ($aaaa=1; $aaaa < 11-$aaa; $aaaa++) {$aa="$aa ";} $vno=$aa;

$aa="$vko[$z]"; $aaa=length($aa); for ($aaaa=1; $aaaa < 11-$aaa; $aaaa++) {$aa="$aa ";} $vko=$aa;

$aa="$kanal[$z]"; $aaa=length($aa); for ($aaaa=1; $aaaa < 7-$aaa; $aaaa++) {$aa="$aa ";} $kanal=$aa;

$aa="$nvo[$z]"; $aaa=length($aa); for ($aaaa=1; $aaaa < 6-$aaa; $aaaa++) {$aa="$aa ";} $nvo=$aa;

$aa="$vob[$z]"; $aaa=length($aa); for ($aaaa=1; $aaaa < 7-$aaa; $aaaa++) {$aa="$aa ";} $vob=$aa;

$aa="$voo[$z]"; $aaa=length($aa); for ($aaaa=1; $aaaa < 9-$aaa; $aaaa++) {$aa="$aa "; $sr\_voo = $sr\_voo + $voo[$z];} $voo=$aa;

print OUT "$z| $vp| $vno| $vko| $kanal| $nvo| $vob| $voo\n";

$vp[$z] = '-';

$vno[$z] = '-';

$vko[$z] = '-';

$kanal[$z] = '-';

$nvo[$z] = '-';

$vob[$z] = '-';

$voo[$z] = '-';

}

}

$srkanal = $srkanal + $skanal[$q7] / $ppp;

print OUT "\n\nОкончание обслуживания каждым каналом:\n";

$max\_T = $N[1]; # время работы СМО

for ($q8 = 1; $q8 <= $N; $q8++)

{

print OUT "канал $q8 : $N[$q8]\n";

if ($N[$q8] < $T) {$rabot = $rabot + $T;} else {$rabot = $rabot + $N[$q8];}

if ($max\_T < $N[$q8]) {$max\_T = $N[$q8];}

}

$vrp = $rabot - $svob;

$svrobsl = $svrobsl + $svob;

$sr\_och = $sr\_och + $sr\_dl\_och;

$skz = $skz + $ppp;

$sotkaz = $sotkaz + $otkaz;

$svpr = $svpr + $vrp;

$smaxvoo = $smaxvoo + $maxvoo;

$vrp = int($vrp \* 100) / 100;

$rabot = int($rabot \* 100) / 100;

print OUT "\n\n\nСуммарное время простоя на $N каналах обслуживания за общее время $rabot часов, час: $vrp\n";

$svo = int(($svoo / $ppp)\*1000000)/1000000;

print OUT "Минимальное время ожидания: $minvoo\n";

print OUT "Максимальное время ожидания: $maxvoo\n";

print OUT "Среднее время ожидания: $svo\n";

$potkaz = int($otkaz/$ppp\*10000)/100;

print OUT "Количество отказов: $otkaz, $potkaz%\n";

$sr\_dl\_och = int($sr\_voo / $max\_T \* 100) / 100;

print OUT "Средняя длина очереди: $sr\_dl\_och\n\n";

$ssvo = $ssvo + $svo;

$ppp = 0;

$otkaz = 0;

$rabot = 0;

$svob = 0;

$svo = 0;

$manvoo = 0;

$maxvoo = 0;

}

$skz = int($skz / $F\*1)/1;

$sotkaz = int($sotkaz / $F\*100)/100;

$svpr = int($svpr / $F\*100)/100;

$ssvo = int($ssvo / $F\*100)/100;

$smaxvoo = int($smaxvoo / $F\*100)/100;

$sr\_och = int($sr\_och / $F \* 100) / 100;

print OUT "\n\nОТЧЕТ:\n\n";

print OUT "Среднее кол-во заявок за рабочий период: $skz\n";

$spotkaz = int($sotkaz/$skz\*10000)/100;

print OUT "Среднее кол-во отказов: $sotkaz, $spotkaz%\n";

$vodsl = 100 - $spotkaz;

print OUT "Вероятность обслуживания: $vodsl%\n";

$ops = 1 - $spotkaz / 100;

print OUT "Относительная пропускная способность: $ops\n";

$aps = int($ops \* $L\*100) / 100;

print OUT "Абсолютная пропускная способность [ед./час]: $aps\n";

print OUT "Среднее время простоя на $N каналах обслуживания за период $T часов, час: $svpr\n";

$vprost = int($svpr / $T / $N \* 10000) / 100;

print OUT "Вероятность простоя СМО: $vprost%\n";

$kz = 100 - $vprost;

print OUT "Коэффициент загузки СМО: $kz%\n";

$srkanal = int($srkanal / $F \* 100) / 100;

print OUT "Среднее число занятых каналов: $srkanal из $N\n";

print OUT "Среднее время ожидания: $ssvo\n";

$svprzay = int(($ssvo + $svrobsl / $F / $skz) \* 100) / 100;

print OUT "Среднее время пребывания заявки в СМО (ожидание + обслуживание): $svprzay\n";

print OUT "Среднее максимальное время ожидания: $smaxvoo\n";

print OUT "Средняя длина очереди: $sr\_och\n\n";

}

else

{

print "файл данных не найден\n";

}

}

else

{

print "файл данных не задан\n";

}

close(OUT);

# Описание функционирования математической модели

Приведем формулы, необходимые для вычислений:

1. 

расчет времени поступления заявок в СМО, где *λ* – интенсивность потока заявок,

*R* - случайная величина, равномерно распределенная в интервале (0,1),

*Тk-1* – время поступления предыдущей заявки;

1.  - вычисляет время обслуживания канала m;
2.  - определяет интервал времени между двумя последовательными заявками (между k-й и k+1);
3.  - формирует момент поступления следующей заявки;
4.  - расчет среднего времени ожидания;

Под характеристиками СМО с ограничением на количество заявок в очереди будем понимать величины, по которым можно оценивать эффективность работы СМО и выбирать лучший из нескольких вариантов СМО.

Величины эффективности:

*Ротк* – вероятность отказа, показывает, какая доля всех поступающих заявок не обслуживается системой из-за занятости ее каналов или большого количества заявок в очереди (для СМО без ограничений на очередь *Ротк = 0*);

*Ротк=kотк/k,* 4.1

где *kотк –* количество заявок, получивших отказ, *k –* общее количество заявок, поступивших в СМО

*Робсл –* вероятность обслуживания (относительная пропускная прособность), показывает, какая доля всех поступающих заявок обслуживается системой,

*Робсл=1-Ротк* 4.2

(для СМО без отказов *Робсл = 1*);

*U –* коэффициент загрузки СМО, показывает, какую часть от общего времени своей работы СМО выполняет обслуживание заявок; желательно, чтобы коэффициент загрузки СМО имел значение на уровне 0,75–0,85. Значения *U* < 0,75 указывают, что СМО простаивает значительную часть времени, т.е. используется нерационально. Значения U > 0,85 указывают на перегрузку СМО.

*U* = ρ(1 – *Pотк*), 4.3

где ρ - нагрузка на СМО:

 *где*  4.4

Величина *ρ* представляет собой отношение интенсивности потока заявок к интенсивности, с которой СМО может их обслуживать. Для СМО с ограничениями на очередь и без очереди возможны любые значения ρ, так как в таких СМО часть заявок получает отказ, т.е. не допускается в СМО.

q – среднее число заявок в очереди (средняя длина очереди);

S – среднее число заявок на обслуживании (в каналах), или среднее число занятых каналов;

*S* = *mU* 4.5

k – среднее число заявок в СМО, т.е. на обслуживании и в очереди;

*k* = *q* + *S* 4.6

w – среднее время пребывания заявки в очереди (среднее время ожидания обслуживания); формула Литтла:

4.7



t – среднее время пребывания заявки в СМО, т.е. в очереди и на обслуживании;

или 4.8 - 4.9



γ – пропускная способность (среднее количество заявок, обслуживаемых в единицу времени); эта величина представляет интерес с точки зрения стороны, осуществляющей эксплуатацию СМО. Обычно желательна максимизация этой величины, особенно в случаях, когда обслуживание каждой заявки обеспечивает получение определенной прибыли.

γ = μ*S или* γ = λ(1 – *Pотк*). 4.10

 - абсолютная пропускная способность.

Величины *U* и *S* характеризуют степень загрузки СМО. Эти величины представляют интерес с точки зрения стороны, осуществляющей эксплуатацию СМО.

Например, если в качестве СМО рассматривается предприятие, выполняющее некоторые заказы, то эти величины представляют интерес для владельцев предприятия.

Величины *Pотк, Pобсл, w* и *t* характеризуют качество обслуживания заявок.

Они представляют интерес с точки зрения пользователей СМО. Желательна минимизация значений *Pотк, w , t* и максимизация *Pобсл.*

Величины *q* и *k* обычно используются в качестве вспомогательных для расчета других характеристик СМО.

Формулы (4.1)–(4.10) могут применяться для расчета характеристик *любых* разомкнутых СМО, независимо от количества каналов, потока заявок, закона распределения времени обслуживания и т.д. [4]

Обозначения:

время работы СМО, час [T]: 7

интенсивность поступления заявок, ед./час [L]: 7

число обслуживающих каналов, ед. [N]: 3

максимальная длина очереди, ед. [M]: 4

закон распределения времени обслуживания (exp/evenly) [ZR]: exp

среднее время обслуживания [TO]: 0,5

погрешность вычислений [E]: 0,1

количество прогонов модели

[5].

В связи с большим объемом данных по реализации 100 прогонов, приведу результаты одного в Таблице 4.1

Таблица 4.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | время прихода заявки | время  начала обслуживания | время  конца обслуживания | канал | номер в очереди | время обслуживания заявки | время ожидания (в очереди) |
|  | 0.13423 | 0.13423 | 1.07323 | 1 | 0 | 0.939 | 0 |
|  | 0.172969 | 0.172969 | 0.177969 | 2 | 0 | 0.005 | 0 |
|  | 0.372996 | 0.372996 | 0.498996 | 2 | 0 | 0.126 | 0 |
|  | 0.395133 | 0.395133 | 1.477133 | 3 | 0 | 1.082 | 0 |
|  | 0.454734 | 0.498996 | 0.708996 | 2 | 0->1 | 0.21 | 0.044261 |
|  | 1.0321 | 1.0321 | 1.0741 | 2 | 0 | 0.042 | 0 |
|  | 1.192161 | 1.192161 | 1.804161 | 1 | 0 | 0.612 | 0 |
|  | 1.304736 | 1.304736 | 1.508736 | 2 | 0 | 0.204 | 0 |
|  | 1.423904 | 1.477133 | 1.500133 | 3 | 0->1 | 0.023 | 0.053228 |
|  | 1.498956 | 1.500133 | 1.501133 | 3 | 0->1 | 0.001 | 0.001176 |
|  | 1.583731 | 1.583731 | 1.738731 | 2 | 0 | 0.155 | 0 |
|  | 1.72184 | 1.72184 | 2.37884 | 3 | 0 | 0.657 | 0 |
|  | 1.768943 | 1.768943 | 2.605943 | 2 | 0 | 0.837 | 0 |
|  | 1.929808 | 1.929808 | 1.941808 | 1 | 0 | 0.012 | 0 |
|  | 1.949207 | 1.949207 | 4.358207 | 1 | 0 | 2.409 | 0 |
|  | 2.020496 | 2.37884 | 2.70784 | 3 | 0->1 | 0.329 | 0.358344 |
|  | 2.199114 | 2.605943 | 2.791943 | 2 | 1->2 | 0.186 | 0.406828 |
|  | 2.401371 | 2.70784 | 4.36184 | 3 | 1->2 | 1.654 | 0.306469 |
|  | 2.666255 | 2.791943 | 2.986943 | 2 | 1->2 | 0.195 | 0.125687 |
|  | 2.728184 | 2.986943 | 3.338943 | 2 | 1->2 | 0.352 | 0.258758 |
|  | 3.364248 | 3.364248 | 3.379248 | 2 | 0 | 0.015 | 0 |
|  | 3.450507 | 3.450507 | 3.584507 | 2 | 0 | 0.134 | 0 |
|  | 3.798883 | 3.798883 | 4.041883 | 2 | 0 | 0.243 | 0 |
|  | 3.870281 | 4.041883 | 4.080883 | 2 | 0->1 | 0.039 | 0.171602 |
|  | 4.028639 | 4.080883 | 5.240883 | 2 | 1->2 | 1.16 | 0.052243 |
|  | 4.074847 | 4.358207 | 4.730207 | 1 | 1->2 | 0.372 | 0.28336 |
|  | 4.316383 | 4.36184 | 5.63084 | 3 | 1->2 | 1.269 | 0.045457 |
|  | 4.465872 | 4.730207 | 4.902207 | 1 | 0->1 | 0.172 | 0.264335 |
|  | 4.494469 | 4.902207 | 5.309207 | 1 | 1->2 | 0.407 | 0.407737 |
|  | 4.528788 | 5.240883 | 5.272883 | 2 | 2->3 | 0.032 | 0.712094 |
|  | 4.536596 | 5.272883 | 6.252883 | 2 | 3->4 | 0.98 | 0.736286 |
|  | 4.565434 | -1 | - | - | 4->4 | 0 | - |
|  | 4.580016 | -1 | - | - | 4->4 | 0 | - |
|  | 4.644491 | -1 | - | - | 4->4 | 0 | - |
|  | 4.944335 | 5.309207 | 5.662207 | 1 | 2->3 | 0.353 | 0.364871 |
|  | 5.064146 | 5.63084 | 5.90084 | 3 | 2->3 | 0.27 | 0.566694 |
|  | 5.117229 | 5.662207 | 5.743207 | 1 | 2->3 | 0.081 | 0.544977 |
|  | 5.201751 | 5.743207 | 6.242207 | 1 | 2->3 | 0.499 | 0.541455 |
|  | 5.525887 | 5.90084 | 6.12684 | 3 | 1->2 | 0.226 | 0.374952 |
|  | 5.5837 | 6.12684 | 6.68084 | 3 | 2->3 | 0.554 | 0.543139 |
|  | 5.595149 | 6.242207 | 7.331207 | 1 | 2->3 | 1.089 | 0.647057 |
|  | 5.626051 | 6.252883 | 6.354883 | 2 | 2->3 | 0.102 | 0.626831 |
|  | 5.741963 | 6.354883 | 6.773883 | 2 | 2->3 | 0.419 | 0.612919 |
|  | 5.790596 | 6.68084 | 7.11084 | 3 | 2->3 | 0.43 | 0.890244 |
|  | 6.176534 | 6.773883 | 7.871883 | 2 | 1->2 | 1.098 | 0.597348 |
|  | 6.310764 | 7.11084 | 8.88584 | 3 | 2->3 | 1.775 | 0.800075 |
|  | 6.407596 | 7.331207 | 8.878207 | 1 | 2->3 | 1.547 | 0.92361 |
|  | 6.592344 | 7.871883 | 8.082883 | 2 | 2->3 | 0.211 | 1.279538 |
|  | 6.687681 | 8.082883 | 9.866883 | 2 | 2->3 | 1.784 | 1.395202 |
|  | 6.702902 | 8.878207 | 8.940207 | 1 | 2->3 | 0.062 | 2.175304 |
|  | 6.910557 | 8.88584 | 9.59184 | 3 | 1->2 | 0.706 | 1.975282 |

Окончание обслуживания каждым каналом:

канал 1: 8.940207

канал 2: 9.866883

канал 3: 9.59184

Суммарное время простоя на 3 каналах: 2.33993000000001 час за общее время обслуживания 28.39893 часов,

минимальное время ожидания: 0

максимальное время ожидания: 2.175304

среднее время ожидания: 0.374262

количество отказов: 3, 588%

# **Анализ результатов испытаний**

Средние значения по 100 прогонам:

Среднее количество заявок за рабочий период: 49

Среднее количество отказов: 0.8, 1.63%

Вероятность обслуживания: 98.37%

Относительная пропускная способность: 0.9837

Абсолютная пропускная способность [ед./час]: 6.88

Среднее время простоя на 3 каналах 2.55ч за период обслуживания 7 часов

Вероятность простоя СМО: 12.14%

Коэффициент загрузки СМО: 87.86%

Среднее число занятых каналов: 1.94 из 3

Среднее время ожидания: 0.88

Среднее время пребывания заявки в СМО (ожидание + обслуживание): 1.38

Среднее максимальное время ожидания: 2.13

Средняя длина очереди: 0.49

По коэффициенту загрузки можно судить о качестве загрузки СМО. Используя формулы 4.1 - 4.4 и таблицу 4.1, получим значение 87.86%

Коэффициент загрузки равен 0,8786 и находится в промежутке больше 0,85. Это значит, что СМО перегружена.

Если рассматривать данную СМО с целью получения прибыли, то по формулам 4.2, 4.10 и с помощью таблицы 4.1 получим значение пропускной способности 6,88. Для получения прибыли важна ее максимизация.

# Заключение

По мере усложнения производственных процессов, развития науки, проникновения в тайны функционирования и развития живых организмов появились задачи, которые не решались с помощью традиционных математических методов и в которых все больше место стал занимать собственно процесс постановки задачи, возросла роль эвристических методов, усложнился эксперимент, доказывающий адекватность формальной математической модели.

В области применения имитационного моделирования лежат задачи моделирования биологических систем, военные, экономические, социальные. Что позволяет решать проблемы различного характера и большого объема.

В данной курсовой мы рассмотрели примитивную задачу о поступлении заявок (клиентов) в канал (парикмахерскую), убедились в эффективности модели.

# Список использованных источников

1. Голик Е.С. Системное моделирование. Ч.1. Имитационное моделирование. Факторный эксперимент: учебно-методический комплекс (учебное пособие)/Е.С. Голик, О.В. Афанасьева. – СПб: СЗТУ, 2007. – 211 с.
2. Голик Е.С. Математические методы системного анализа и теории приятия решений. Ч. II: Учебное пособие. – СПб: СЗТУ, 2005, - 102 с.
3. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем. – М.: ДМК Пресс, 2004. – 320 с.: ил. (Серия «Проектирование»).
4. Оптимизация решений на основе методов и моделей мат. программирования: Учеб. пособие по курсу «Систем. анализ и исслед. операций» для студ. спец. «Автоматизир. системы обраб. информ.» дневн. и дистанц. форм обуч. / С.С. Смородинский, Н.В. Батин. – Мн.: БГУИР, 2003. – 136 с.: ил.
5. Конспект лекций