

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ

Варанкин Игорь Игоревич

Выпускная квалификационная работа бакалавра

**Нестационарная система обслуживания с
конечным источником заявок с абсолютными
приоритетами**

Направление 010400

Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Еремин А. С.

Санкт-Петербург

2016

Содержание

Введение	3
Постановка задачи	5
Обзор литературы	6
Глава 1. Аналитическая модель НСО.....	8
1.1. Модель НСО	8
1.2. Модель НСО с абсолютными приоритетами.....	12
Глава 2. Описание алгоритма.....	16
2.1. Формирование списка возможных состояний.....	16
2.2. Заполнение матрицы коэффициентов	20
2.3. Реализация решения	21
2.4. Реализация имитационного моделирования	23
Глава 3. Практический пример	25
.....	
Заключение	33
Список литературы	34

Введение

Теория массового обслуживания (ТМО) является относительно новым разделом теории вероятности. Целью изучения ТМО является выбор структуры системы обслуживания и процесса обслуживания на основе изучения потоков требований на обслуживание, поступающих в систему и выходящие из неё, длительности ожидания и длины очередей.

Большинство авторов используют модели ТМО в предположении, что очередь заявок бесконечна, существует стационарный режим. Это позволяет оценить эффективность системы обслуживания на длительном интервале времени, получить характеристики «в среднем». Какой в среднем процент заявок будет теряться, какая средняя длительность ожидания обслуживания, какая в среднем нагрузка на один канал и т.д.

В то же время большой практический и теоретический интерес представляют модели нестационарных систем обслуживания (НСО). Изучению таких систем посвящено сравнительно мало работ.

Под моделью НСО понимается модель системы обслуживания с переменными во времени вероятностями состояний, каждое из которых определяет вероятность числа находящихся в НСО заявок и числа получивших обслуживание заявок. Число заявок, поступающих на обслуживание в рассматриваемых моделях, конечно. Стационарный режим не выделяется и основным предметом исследования является переходный процесс. Соответственно и важными характеристиками системы являются совсем другие: например среднее время обработки всех или некоторого числа заявок или вероятность не уложиться в требуемые сроки, вероятность превышения критической длины очереди, вероятность отказа в обслуживании критического числа заявок и т.д. Для иллюстрации важности моделирования НСО можно рассмотреть работу приемной комиссии университета. Заявления на поступление, приходят концентрированно в течение небольшого

промежутка времени. Кроме того нас не интересует сколько из них мы можем потерять ведь важно обработать их все. Поэтому одной из важных характеристик будет являться количество людей (каналов обслуживания) необходимых для обработки заявлений в срок.

Имеется большое число реальных ситуаций, когда в потоке требований могут содержаться требования нескольких типов, причем требования первого типа обслуживаются вне всякой очереди, если только в очереди нет требований того же типа. По отношению к требованиям третьего и следующих типов правом преимущества пользуются требования второго типа и т.д. В качестве хорошего примера можно привести обслуживание на телеграфе, где срочные телеграммы передаются раньше обычных, даже когда простые телеграммы сданы ранее срочных.

Далее будут рассматриваться системы с так называемыми абсолютными приоритетами, в которых обслуживание заявки меньшего приоритета в случае прихода более «важной» заявки прерывается, и возобновляется лишь после обработки всех более приоритетных заявок.

Вероятности нахождения системы в том или ином состоянии есть функции от времени, которые являются решением обратного уравнения Чепмена – Колмогорова. Однако его составление даже для одноканальной НСО с абсолютными приоритетами представляется проблематичным.

В данной работе представлен алгоритм формирования нижнетреугольной матрицы коэффициентов линейной стационарной системы обыкновенных дифференциальных уравнений (ОДУ) для вышеуказанных систем без вывода общего уравнения системы. Для проверки корректности составления матрицы, проводится сравнение решения системы ОДУ с указанной матрицей коэффициентов, и результатами имитационного моделирования. Так же приведены графики, отображающие наиболее интересные характеристики исследуемой системы.

Постановка задачи

Рассматривается одноканальная нестационарная система массового обслуживания с абсолютными приоритетами. На вход системы поступают N заявок K типов (приоритетов) ($K \leq N$). Известно сколько заявок каждого типа может прийти. Распределения временных интервалов между моментами поступления заявок описывается экспоненциальными законами с интенсивностями, зависящими от номера типа заявок. Закон распределения времён обслуживания – экспоненциальный с интенсивностями, так же зависящими от номера типа заявок. Вычислительную систему представим в виде одноканальной нестационарной системы обслуживания без потерь.

Необходимо сформулировать правила формирования списка состояний, а так же способ генерации нижнетреугольной матрицы коэффициентов системы однородных дифференциальных уравнений, описывающих вышеуказанную систему.

Обзор литературы

Изучение нестационарных систем обслуживания на практике, как впрочем и всей теории массового обслуживания, невозможно без должного освоения теоретической литературы.

В книге Дж. Кемени и Дж. Снелла [6] изложены основы теории цепей Маркова. Основные теоремы даны в простой и понятной форме, не требующей глубоких математических познаний.

В последнем разделе учебника Е. С. Венцель [4], даются основные определения и теоремы ТМО. Автор учебника ставил задачу изложить предмет наиболее просто и наглядно. Книга снабжена большим количеством примеров, в ряде случаев расчетного характера, в которых применение излагаемых методов иллюстрируется на конкретном практическом материале и доводится до численного результата.

Для более подробного и целостного изучения ТМО следует обратиться к труду А. Я. Хинчина [10], а так же к работе Л. Клейнрока [7].

В книге Б. В. Гнеденко, И. Н. Коваленко [5] помимо рассмотрения задач теории массового обслуживания в целом и методов их решений, уделено внимание интересующим нас нестационарным системам обслуживания.

В учебнике В. Ф. Матвеева и В. Г. Ушакова [8] рассматриваются приоритетные НСО, в частности системы с абсолютными приоритетами.

Методическое издание В. П. Бубнова и В. И. Сафонова [2] содержит рассмотрение основных моделей НСО с различными типами поступления и обработки заявок. Приводятся диаграммы переходов, а так же характеристики этих моделей.

Статья В. П. Бубнова, А. С. Еремина и С. А. Сергеева [1] содержит описание численно-аналитического метода решения уравнения Чепмена-

Колмогорова. Показано преимущество данного метода над “классическим” методом Рунге-Кутты четвертого порядка с постоянным шагом, а так же методом ode45 пакета Matlab, используемым при решении вышеуказанных уравнений.

Глава 1. Аналитическая модель НСО

1.1. Модель НСО

Рассмотрим систему обслуживания, на вход которой последовательно поступает N заявок. Распределения временных интервалов между моментами поступления заявок описываются экспоненциальными законами с интенсивностями, зависящими от номера заявки, $\{ \lambda_1, \lambda_2, \dots, \lambda_N \}$ соответственно. Закон распределения времени обслуживания – экспоненциальный с интенсивностями, зависящими от номера заявок, $\{ \mu_1, \mu_2, \dots, \mu_N \}$ соответственно. Представим такую систему обслуживания вложенной марковской цепью с дискретным множеством состояний и непрерывным временем без потерь. [2]

Марковская цепь – процесс, в котором для каждого момента времени вероятность любого состояния системы в будущем зависит только от состояния системы в настоящий момент времени и не зависит от того, каким образом система пришла в это состояние. [4]

Система обслуживания без потерь – система, в которой заявки поступающие, в момент обработки какой либо другой заявки, не теряются. [7]

Состояния системы в каждый момент времени будем характеризовать числом находящихся в системе запросов i ($i = 0, N$) и числом обработанных запросов j ($i = 0, N-i$). Вероятности пребывания системы в этих состояниях будем обозначать $P_{i,j}(t)$. В таком представлении система будет иметь суммарное число состояний равное $N_c = (N+1)(N+2)/2$.

Все состояния невозвратные, иначе говоря, из них можно выйти, но вернуться уже нельзя. Конечное состояние – поглощающее, то есть, попав в него, выйти уже нельзя. Процесс однородный, не эргодический, для него не существует стационарного режима.

Однородный процесс – процесс, при котором в любой фиксированный интервал времени с вероятностью 1 система находится в одном из состояний. [5]

Эргодический процесс – процесс, в котором из каждого состояния можно попасть в другое за конечное число шагов. [6]

Под переходом из одного состояния в другое понимается поступление новой заявки на обслуживание, либо обработка заявки, которая в исходном состоянии обрабатывалась.

Диаграмма переходов между состояниями системы представлена на рис.1 и подробно рассмотрена в [9].

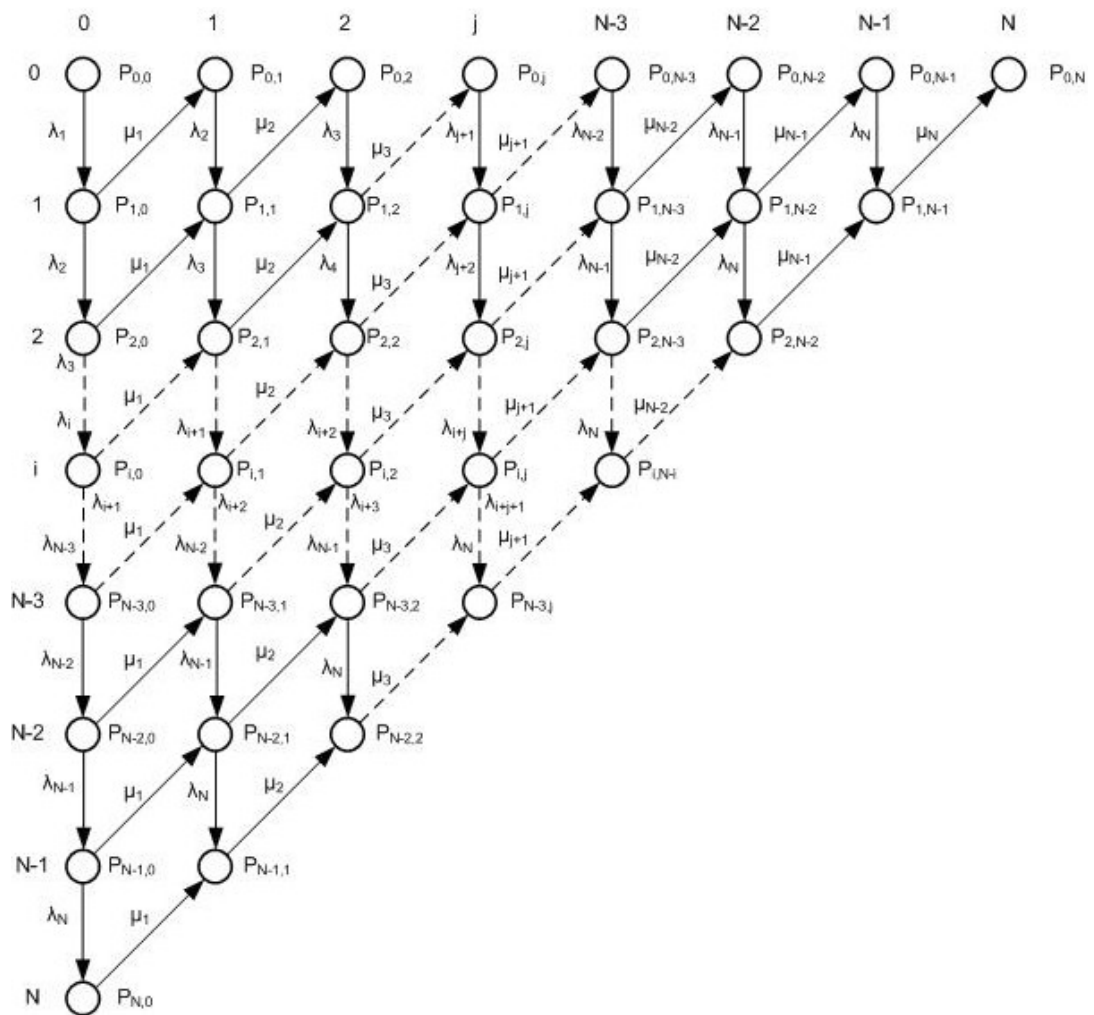


Рис. 1. Диаграмма переходов между состояниями НСО

Для того чтобы определить значения вероятностей нахождения системы в каждом из состояний необходимо решать относительно $P_{i,j}(t)$ систему дифференциальных уравнений i,j -тое уравнение которой в общем виде можно записать в виде

$$\begin{aligned} \frac{dP_{i,j}(t)}{dt} = & H(i)(P_{i-1,j}(t)\lambda_{i+j} - P_{i,j}(t)\mu_{j+1}) + \dot{c} \\ & + H(j)P_{i+1,j-1}(t)\mu_j - H(N_c - i - j)P_{i,j}(t)\lambda_{i+j+1} \end{aligned} \quad (1)$$

$$j = 0, N, \quad i = 0, N-j$$

где $H(k)$ – функция Хевисайда:

$$H(k) = \begin{cases} 1, & \text{если } k > 0 \\ 0, & \text{если } k \leq 0 \end{cases}$$

В качестве начальных условий выберем нахождение в состоянии $(0,0)$, то есть $P_{i,j}(0) = 1 - H(i+j)$. Для каждого момента времени должно

выполняться условие вида $\sum_{i=0}^N \sum_{j=0}^{N-i} P_{i,j}(t) = 1$. [3]

Рассмотренная выше линейная однородная система уравнений (1) представляет собой обратное уравнение Чепмена–Колмогорова и может быть представлена в матричной форме:

$$\dot{P}(t) = AP(t) \quad (2)$$

где $P(t)$ – вектор неизвестных функций размерности N_c , а A – квадратная матрица.

Для системы (2) существует явное аналитическое решение, если только известны собственные числа матрицы A . Очевидным является тот факт, что в случае треугольного вида матрицы A (для определенности – нижнетреугольного), её собственные числа выписаны в явном виде на диагонали. [1]

Если пронумеровать состояния по возрастанию числа обработанных заявок, а внутри этих групп по возрастанию числа поступивших заявок, то ни одно из состояний не будет иметь зависимости от последующих. Соответственно, полученная матрица будет нижнетреугольной. На графе это будет выглядеть как нумерация сверху вниз с перемещением по столбцам слева направо. [1]

Рассмотрим простой пример заполнения матрицы A . Пусть может прийти 2 заявки с интенсивностями распределения времени (ИРВ) $\{ \lambda_1, \lambda_2 \}$ соответственно. ИРВ обработки возьмем равными $\{ \mu_1, \mu_2 \}$ соответственно. Количество состояний $N_c = 6$. Тогда матрица A будет заполняться следующим образом. Диагональные элементы $A[i,i]$, $i = 0, N_c$ заполняются суммой интенсивностей выхода из i -го состояния, со знаком минус. Недиагональные элементы $A[i,j]$, $i = 0, N_c - 1$, $j = 0, N_c - i$ заполняются ИРВ перехода из j -го состояния в i -ое. Матрица A в этом маленьком примере выглядит так:

$-\lambda_1$	0	0	0	0	0
	$-\lambda_2$				
λ_1	μ_1	0	0	0	0
0	λ_2	$-\mu_1$	0	0	0
0	μ_1	0	$-\lambda_1$	0	0
0	0	μ_1	λ_1	$-\mu_2$	0
0	0	0	0	μ_2	0

Значение вероятности нахождения в НСО i заявок в каждый момент времени определяется из формулы:

$$P_i(t) = \sum_{j=0}^{N-i} P_{i,j}(t)$$

Значение вероятности обслуживания j заявок в каждый момент времени определяется из:

$$P_j(t) = \sum_{i=0}^{N-j} P_{i,j}(t)$$

Значение вероятности обслуживания не менее q запросов может быть определено из выражения:

$$P_q(t) = \sum_{i=q}^N P_j(t)$$

1.2. Модель НСО с абсолютными приоритетами

От рассмотрения простейшей одноканальной модели нестационарной системы обслуживания перейдем к одноканальной нестационарной системе с абсолютными приоритетами.

На вход системы поступают N заявок K типов (приоритетов) ($K \leq N$). Известно сколько заявок каждого типа может прийти. Распределения временных интервалов между моментами поступления заявок описывается экспоненциальными законами с интенсивностями, зависящими от номера типа заявок. Закон распределения времён обслуживания – экспоненциальный с интенсивностями, так же зависящими от номера типа заявок. Вычислительную систему представим в виде одноканальной нестационарной системы обслуживания без потерь.

Введем некоторые обозначения:

- n_1, n_2, \dots, n_k – максимальное число заявок каждого приоритета
($N = \sum_{z=1}^k n_z$)
- i_1, i_2, \dots, i_k – число поступивших в систему заявок соответствующего приоритета (обработанных и не обработанных) $i_1 = 0, n_1, \dots, i_k = 0, n_k$,
- j_1, j_2, \dots, j_k – число обработанных заявок соответствующего приоритета

$$j_1 = 0, n_1, \dots, j_k = 0, n_k,$$

- $\lambda_1, \lambda_2, \dots, \lambda_k$ – интенсивности поступления заявок соответствующего типа
- $\mu_1, \mu_2, \dots, \mu_k$ – интенсивности обработки заявок соответствующего типа

Таким образом, НСО с K приоритетами описывается K векторами (n_1, λ_1, μ_1) , (n_2, λ_2, μ_2) , ..., (n_k, λ_k, μ_k) . А любое состояние описывается вектором $(i_1, j_1, i_2, j_2, \dots, i_k, j_k)$.

Граф переходов между состояниями НСО с абсолютными приоритетами в общем случае является многомерным, и к тому же не обладает свойством планарности, а значит, не может быть изображен на плоскости без пересечения ребер.

Вероятности нахождения системы в том или ином состоянии есть функции от времени, которые являются решением стационарной системы однородных дифференциальных уравнений:

$$\dot{P}(t) = AP(t)$$

где $P(t)$ – вектор функция, i -й компонент которой есть вероятность нахождения в i -ом состоянии в момент времени t , а A – квадратная матрица. В матрице A каждый недиагональный элемент $A[i, j]$ содержит интенсивность перехода из j -го состояния в i -ое, а диагональные $A[i, i]$ – сумму интенсивностей обработки i -ой заявки, со знаком минус. В качестве начальных данных выберем нахождение системы в начальный момент времени в состоянии $(0, 0, 0, 0, \dots, 0, 0)$, то есть $P_0 = P(0) = (1, 0, \dots, 0)$.

Стоит подчеркнуть основную особенность НСО с абсолютными приоритетами. Мы не можем приступить к обработке и даже продолжить обработку заявки с определенным приоритетом, если в очереди находится заявка более высокого приоритета. К примеру, рассмотрим систему, в которую может прийти три заявки первого приоритета и две второго. Пусть система находится в состоянии $(2, 1, 1, 0)$, то есть пришло две заявки первого

типа и одна из них уже обработана, а так же в очереди находится одна заявка второго типа.

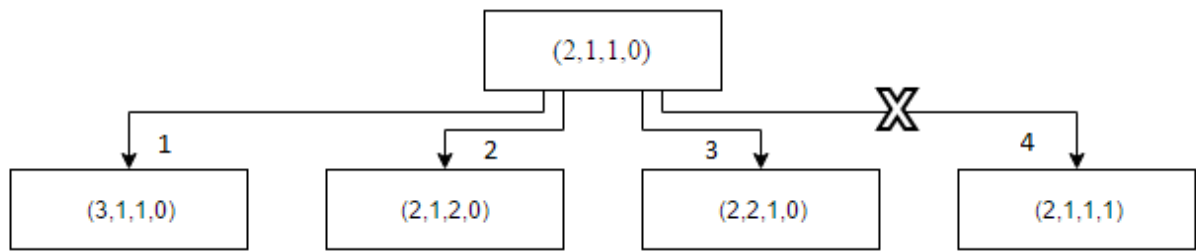


Рис.2 Возможные переходы из состояния $(2,1,1,0)$

Как можно увидеть из рис.2 из состояния $(2,1,1,0)$ можно получить заявку первого типа или второго типа, чему соответствуют первый и второй переходы на рис.2. Так же мы можем обработать заявку первого приоритета и перейти в состояние $(2,2,1,0)$, однако переход из $(2,1,1,0)$ в $(2,1,1,1)$ невозможен, так как в очереди есть еще необработанная заявка более высокого приоритета.

Стоит отметить также, что возобновление обработки заявки, обработка которой была прервана приходом более приоритетной, начинается заново, то есть без учета того, сколько она обрабатывалась до этого.

Поскольку система не имеет возвратных состояний, можно упорядочить состояния таким образом, чтобы матрица A была треугольной. В таком случае собственные числа будут находиться на её диагонали. Найти аналитическое решение системы в явном виде, зная собственные числа, не составит большого труда.

Однако использовать стандартные методы решения систем дифференциальных уравнений с большими матрицами коэффициентов, даже зная собственные числа, не всегда рационально.

Основными недостатками традиционных численных методов являются накапливаемая на каждом шаге интегрирования погрешность и время работы

алгоритма. Причем, вычислительная погрешность может привести и к отсутствию физического смысла получаемого решения. Поэтому рекомендуется использовать метод, описанный в статье [1].

Глава 2. Описание алгоритма

В данном разделе будет описан алгоритм, принимающий на вход параметры нестационарной системы обслуживания с абсолютными приоритетами, и выдающий на выход нумерованный список возможных

состояний системы, а так же нижнетреугольную матрицу коэффициентов обратного уравнения Чепмена-Колмогорова.

Алгоритм реализован на объектно-ориентированном языке C#.

Программа принимает на вход данные вида:

$$n_1 : \lambda_1 ; \mu_1$$

$$n_2 : \lambda_2 ; \mu_2$$

...

$$n_k : \lambda_k ; \mu_k$$

2.1. Формирование списка всех возможных состояний

ИРВ поступления и обработки заявок нас интересовать пока не будут. Для формирования списка состояний нам потребуется лишь количество заявок каждого типа.

Основной принцип алгоритма заключается в том, что все возможные состояния разбиваются на уровни. Каждый уровень фактически есть множество состояний, с одинаковым числом переходов. Любое состояние можно представить в виде $(i_1, j_1, i_2, j_2, \dots, i_k, j_k)$. Таким образом, номер уровня для любого из состояний может быть найден как:

$$L(i_1, j_1, i_2, j_2, \dots, i_k, j_k) = \sum_{x=1}^k \sum_{y=0}^{n_x} (i_y + j_y)$$

Номер последнего уровня находится из формулы:

$$lastL = 2 * \sum_{x=1}^k n_x$$

Очевидно, что нулевой и последний уровни будут состоять лишь из одного состояния $(0,0, \dots, 0,0)$, $(n_1, n_1, n_2, n_2, \dots, n_k, n_k)$ соответственно.

Алгоритм последовательно генерирует каждый уровень, начиная с нулевого. Каждая итерация получает на вход уровень. По окончании итерации производится очистка полученного уровня от одинаковых состояний, оставляя состояния лишь в единственном экземпляре. Итерации продолжаются, пока не будет сформирован последний уровень. Рассмотрим процесс формирования уровней подробнее.

Начало итерации.

Исследуем каждое состояние уровня.

Просматриваем каждый четный (0-ую позицию считаем четной) элемент вектора $(i_1, j_1, i_2, j_2, \dots, i_k, j_k)$, другими словами смотрим на элементы i . Если $i_x < n_x$, $x=1, k$ то в генерируемый уровень добавляем новое состояние, отличающееся от исходного, лишь тем, что i_x увеличен на 1. То есть мы добавляем состояние, в котором пришла еще одна заявка x -го приоритета.

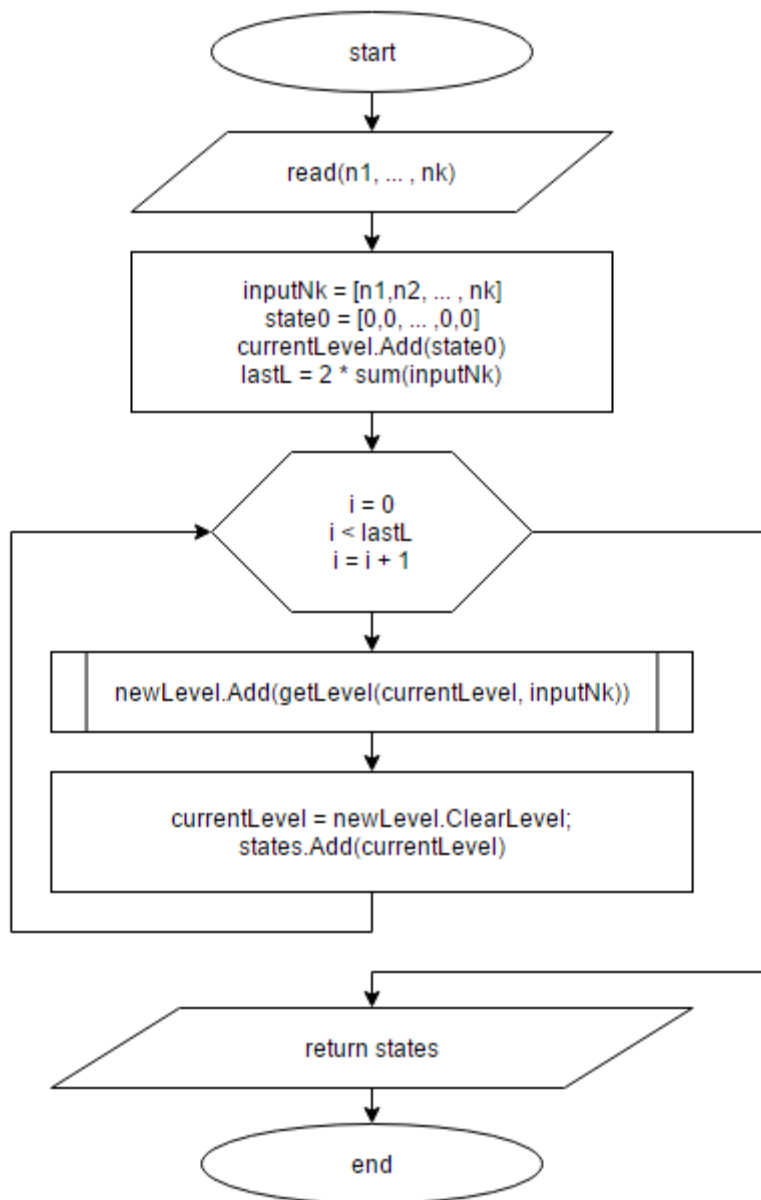
Далее рассматриваем каждый нечетный элемент вектора $(i_1, j_1, i_2, j_2, \dots, i_k, j_k)$, то есть элементы j . Если $j_x < i_x$, $x=1, k$, то в генерируемый уровень добавляем новое состояние, отличающееся от исходного лишь тем, что j_x увеличен на 1. Если мы создаем такое состояние, то прекращаем рассмотрение нечетных элементов вектора. Это позволит нам не рассматривать обработки системой заявок, при наличии в очереди заявки с более высоким приоритетом.

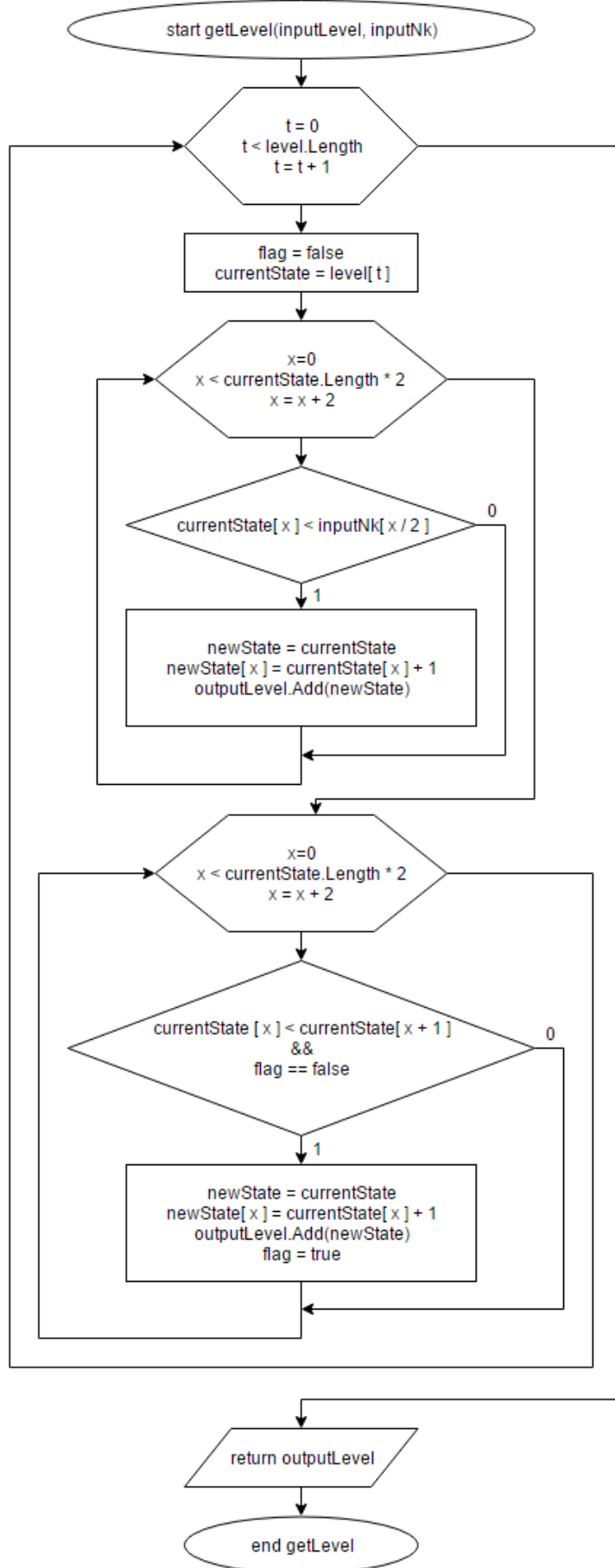
Очищаем полученный уровень от дубликатов.

Конец итерации.

В итоговый список состояний добавляем сформированный уровень.

Переход между состояниями внутри уровня невозможен. Из любого уровня (кроме последнего) система может перейти только в следующий уровень. Если пронумеровать состояния, в том порядке, в котором они добавлялись итоговый список состояний, то получится что состояния с большим номером не зависят от состояний с меньшим. Это гарантирует нам нижнетреугольный вид матрицы коэффициентов обратного уравнения Чепмена-Колмогорова. Ниже представлены блок-схемы алгоритма.





$X[i]$ возвращает i -й элемент структуры X .

Метод $X.Length$ возвращает количество элементов в X .

Метод $X.Add(Y)$ добавляет в конец структуры X структуру Y .

Метод $sum(X)$ возвращает сумму элементов вектора X .

Метод $X.ClearLevel$ – удаляет дубликаты из структуры X .

2.2. Заполнение матрицы коэффициентов

Для начала создадим квадратную матрицу из нулей размерностью $N_c \times N_c$, где N_c число состояний, которое мы получим после их генерации. Так же имеем список всех состояний. Структура хранения всех состояний выглядит следующим образом: список содержащий множество уровней от нулевого до последнего. Каждый уровень представляется списком состояний. Состояния представлены в виде векторов вида $(i_1, j_1, i_2, j_2, \dots, i_k, j_k)$.

Каждому состоянию присваиваем номер, равный его порядковому, при расположении состояний в порядке возрастания уровней

Так же нам потребуется максимальное число заявок каждого приоритета $\{n_1, n_2, \dots, n_k\}$, и ИРВ поступления и обработки заявок каждого типа $\{\lambda_1, \lambda_2, \dots, \lambda_k\}$ и $\{\mu_1, \mu_2, \dots, \mu_k\}$.

Нумеруем состояния, в том порядке, в котором они добавлялись итоговый список состояний.

Алгоритм заполнения матрицы A просматривает все уровни, и каждое из состояний внутри уровней. Рассмотрение одного состояния будем называть одной итерацией.

Начало итерации.

Находим номер рассматриваемого состояния $index1$.

Просматриваем каждый четный (0-ую позицию считаем четной) элемент вектора $(i_1, j_1, i_2, j_2, \dots, i_k, j_k)$, другими словами смотрим на i -е элементы. Если $i_x < n_x$, $x=1, k$, то находим номер состояния в следующем уровне, отличающегося от исходного, лишь увеличенным на единицу i_x . Номер найденного состояния $index2$. Это состояние, в которое мы можем перейти, получив заявку x -го приоритета. Далее в матрице A к значению ячейки $[index2, index1]$ прибавляем значение λ_x , а из значения ячейки $[index1, index1]$ вычитаем λ_x .

Далее рассматриваем каждый нечетный элемент вектора $(i_1, j_1, i_2, j_2, \dots, i_k, j_k)$, то есть j -е элементы. Если $j_x < i_x$, $x=1, k$, то находим номер состояния в следующем уровне, отличающегося от исходного, лишь увеличенным на единицу j_x . Номер найденного состояния $index2$. Если мы находим такое состояние, то прекращаем рассмотрение нечетных элементов вектора. Это состояние, в которое мы можем попасть, обработав заявку x -го приоритета. В матрице A к значению ячейки $[index2, index1]$ прибавляем значение μ_x , а из значения ячейки $[index1, index1]$ вычитаем μ_x .

Конец итерации.

2.3. Реализация решения

Рассматриваем обратное уравнение Чепмена – Колмогорова вида:

$$\dot{P}(t) = AP(t) \quad (2)$$

где $P(t)$ – вектор функция, i -й компонент которой есть вероятность нахождения в i -ом состоянии в момент времени t , а A – найденная квадратная матрица.

Решение системы было найдено методом матричной экспоненты.

В качестве начальных данных выберем вектор нахождения в состоянии $(0,0,0,0, \dots, 0,0)$, то есть в нулевом состоянии:

$$P_0 = P(0) = (1, 0, \dots, 0)$$

Решение уравнения (2) можно выписать в явном виде:

$$P(t) = e^{tA} P_0$$

Матричная экспонента e^{tA} находится как сумма ряда:

$$e^{tA} = E + \sum_{k=1}^{\infty} \frac{t^k A^k}{k!}$$

где E – единичная матрица.

Суммирование заканчивается если очередное k -е слагаемое по норме мало:

$$\left\| \frac{t^k A^k}{k!} \right\| < \varepsilon$$

В качестве нормы матрицы выбрана норма Фробениуса:

$$\|Q\| = \sqrt{\sum_{i,j} Q[i,j]^2}$$

Находим решение системы (2) во временном отрезке $[0, \max T]$ с шагом step . Матрица решения Solve будет иметь размерность $N_c \times [\max T / \text{step}]$, где каждый столбец $\text{Solve}[i,j]$, $i=1, N_c-1$, для любого фиксированного j содержит вероятности нахождения системы в состояниях i в момент времени $j \times \text{step}$.

2.4. Реализация имитационного моделирования

При имитационном моделировании алгоритм, реализующий модель, воспроизводит процесс функционирования системы во времени. Имитируется процесс получения и обработки заявок, с сохранением логической структуры объекта и последовательности протекания процесса во времени. Это позволяет по исходным данным получить сведения о состоянии процесса в определенные моменты времени.

Результаты, полученные имитационной моделью, являются реализацией случайных величин и функций, поэтому для нахождения характеристик процесса функциональной системы производится его многократное воспроизведение с последующей статистической обработкой.

Пусть моделирование проводится N_s раз. Назовем один этап имитационного моделирования малой имитацией.

Каждая малая имитация начинается из нулевого состояния и выполняется, пока система не окажется в последнем состоянии либо будет достигнут временной предел $\max T$. Изначально время имитации $\text{Time} = 0$. Матрица малой имитации smallImit имеет размерность $N_c \times \lceil \max T / \text{step} \rceil$.

$\lceil q \rceil$ – округление q до целого в большую сторону.

$\lfloor q \rfloor$ – округление q до целого в меньшую сторону.

Рассмотрим один шаг малой имитации.

Предположим, система находится в состоянии с номером r . Формируем множество состояний, в которое мы можем перейти из r -го состояния. Это множество будет состоять из состояний, в которое система может перейти, получив заявку каждого приоритета, или обработав наиболее приоритетную заявку. Каждому из возможных переходов соответствует ИРВ, с которой мы можем в него попасть. Генерируем экспоненциально распределенную

псевдослучайную величину - время перехода в каждое из состояний (параметром распределения служат ИРВ переходов).

Выбираем состояние, в которое можем перейти за наименьшее время minTime

В матрице имитации элементам $\text{smallImit}[p,x]$ присвоить 1, где

$$x = \lceil \text{maxT} / \text{step} \rceil, \lfloor (\text{Time} + \text{minTime}) / \text{step} \rfloor$$

К суммарному времени имитации Time прибавляем minTime .

Конец шага малой имитации.

По окончании малой имитации элементы $\text{smallImit}[i,j]$ равные 1 будут говорить о том, что в момент времени $j \times \text{step}$ система находилась в состоянии i .

Проведя N_3 малых имитаций, сложим все матрицы этих имитаций, и разделим каждый элемент этой суммы на N_3 . В результате получим матрицу Imit . Смысл матрицы Imit такой же, как и у матрицы Solve .

Анализ матриц Imit и Solve позволяет нам судить о правильности или не правильности работы алгоритма нумерации состояний и формировании матрицы коэффициентов обратного уравнения Чепмена-Колмогорова для НСО с абсолютными приоритетами.

Глава 3. Практический пример

Рассмотрим конкретный пример НСО с абсолютными приоритетами.

На вход программы подаются следующие данные:

$$2:0.5;0.3$$

$$5:0.4;0.25$$

$$3:0.6;0.5$$

Это интерпретируется так:

$$n_1=2 \quad , \quad n_2=5 \quad , \quad n_3=3$$

$$\lambda_1=0.5 \quad , \quad \mu_1=i \quad 0.3$$

$$\lambda_2=0.4 \quad , \quad \mu_2=i \quad 0.25$$

$$\lambda_3=0.6 \quad , \quad \mu_3=i \quad 0.5$$

Суммарное число заявок $N = 10$. Число состояний после их генерации $N_c=1260$. Матрица A является нижнетреугольной и имеет размерность 1260×1260 .

Найдем матрицу решения обратного уравнения Чепмена-Колмогорова на отрезке $[0,70]$ шагом $step=0.5$.

Имитационное моделирование выполним на том же отрезке, то есть с параметром $maxT=70$ и $step = 0.5$.

В результате работы программы получаем две матрицы одинаковой размерности 1260×141 . Первая матрица – матрица решения уравнения Чепмена – Колмогорова, а вторая – результатов имитационного моделирования. i -я строка этих матриц содержит значения вероятностей

нахождения в i -ом состоянии на протяжении всего временного интервала с выбранным шагом.

На основании этих матриц с помощью пакета прикладных программ MATLAB были построены графики интересующих нас характеристик рассматриваемой системы.

На рис. 2 изображены кривые вероятностей пребывания системы в последнем состоянии на рассматриваемом промежутке времени. Имитации проводились 200 раз. Сплошная кривая – кривая имитации, пунктирная – решения. Максимальная разница между значениями на указанном промежутке $\max\text{Diff} = 0.0339$.

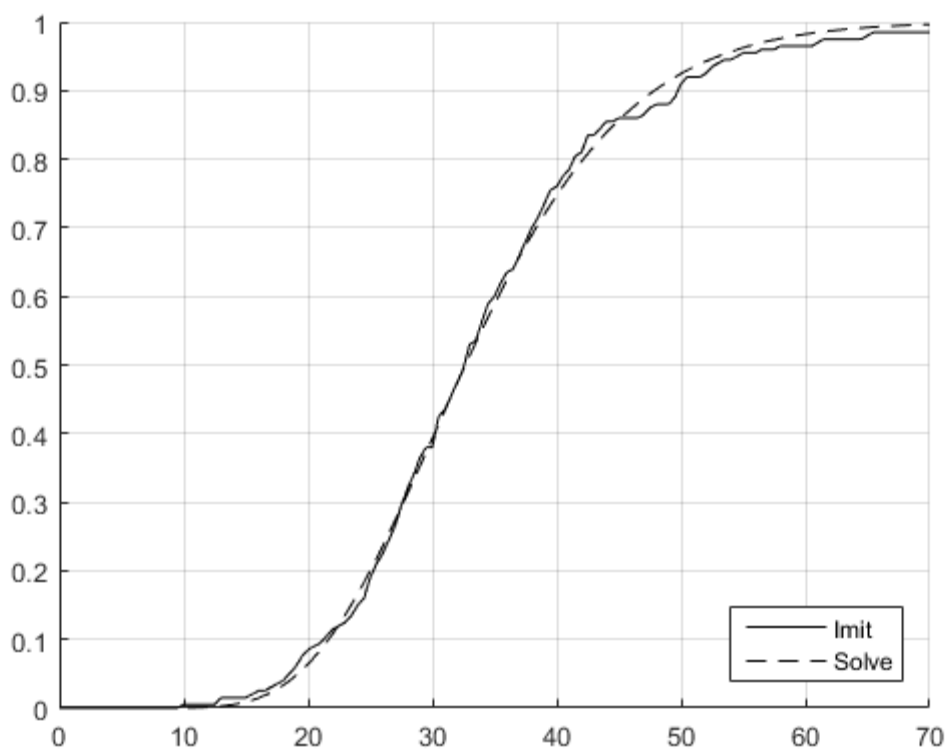


Рис.2

Кривые вероятностей нахождения системы в последнем состоянии уже при 1000 имитаций на исследуемом промежутке времени изображены на Рис. 3. $\maxDiff = 0.0144$.

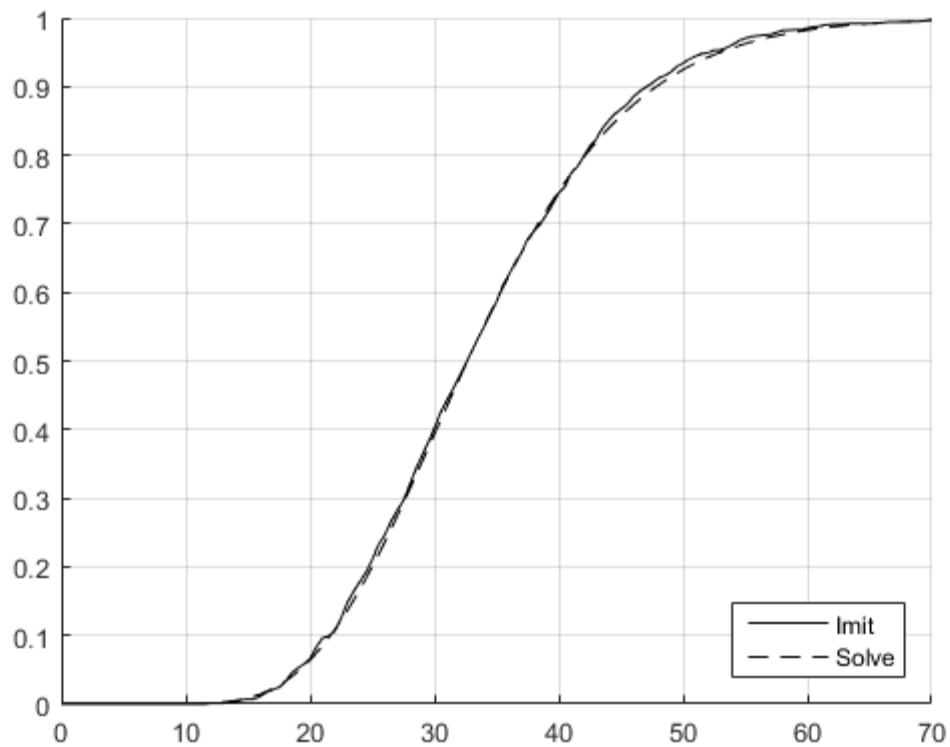


Рис. 3

А на рис. 4 изображены кривые вероятностей нахождения в последнем состоянии при 100 000 имитаций. $\maxDiff=0.0015$. Визуально графики сливаются.

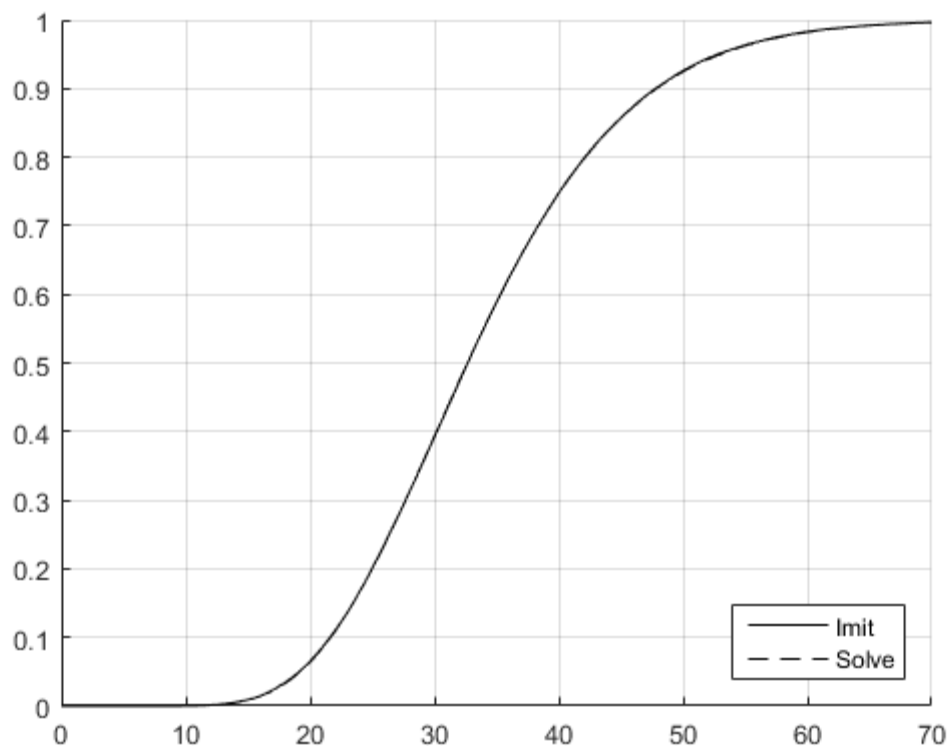


Рис. 4

Не трудно заметить тенденцию к уменьшению разницы между решением и результатом имитации при увеличении количества имитаций.

Дальнейшие графики приводятся для числа имитаций равного 100 000.

График вероятностей того, что система находится в свободном состоянии, приводится на рис. 5. Свободное состояние системы – состояние, при котором в системе отсутствуют необработанные заявки.

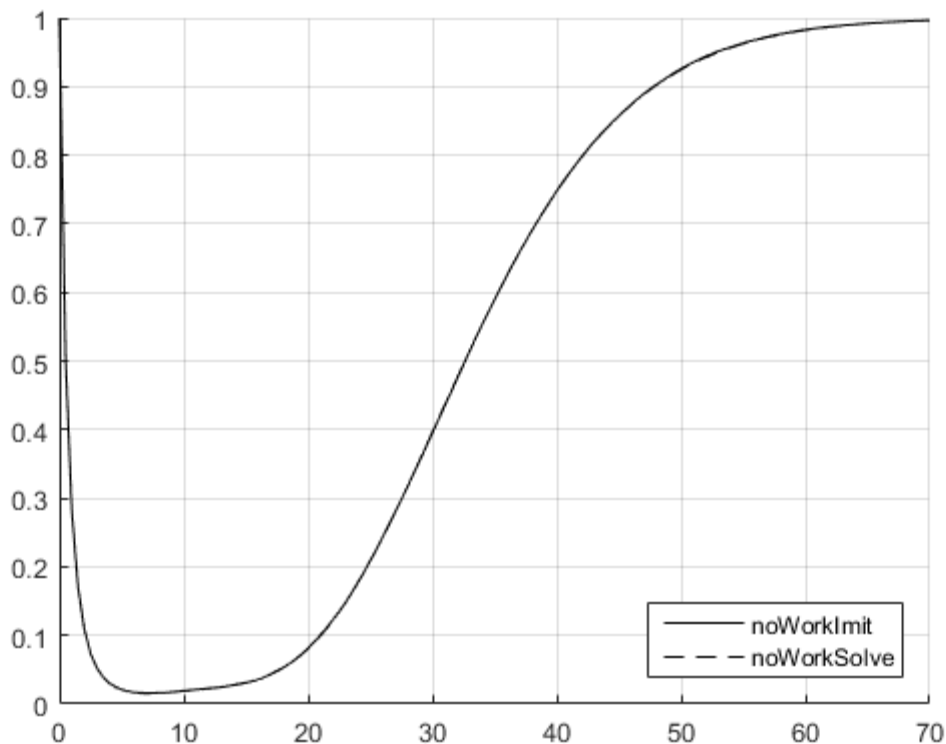


Рис. 5

На рисунках 6, 7 и 8 приведены графики вероятности того что в системе обработаны все заявки первого, второго и третьего типов соответственно. Сплошной линией, как и прежде, обозначается кривая, полученная в результате имитационного моделирования, а пунктирной – в результате решения. Визуально кривые практически не различимы, и максимальная разница между значениями на рассматриваемом промежутке указана ниже:

$\max\text{Diff} = 0.0034$ для рис. 6.

$\max\text{Diff} = 0.0021$ для рис. 7.

$\max\text{Diff} = 0.0014$ для рис. 8.

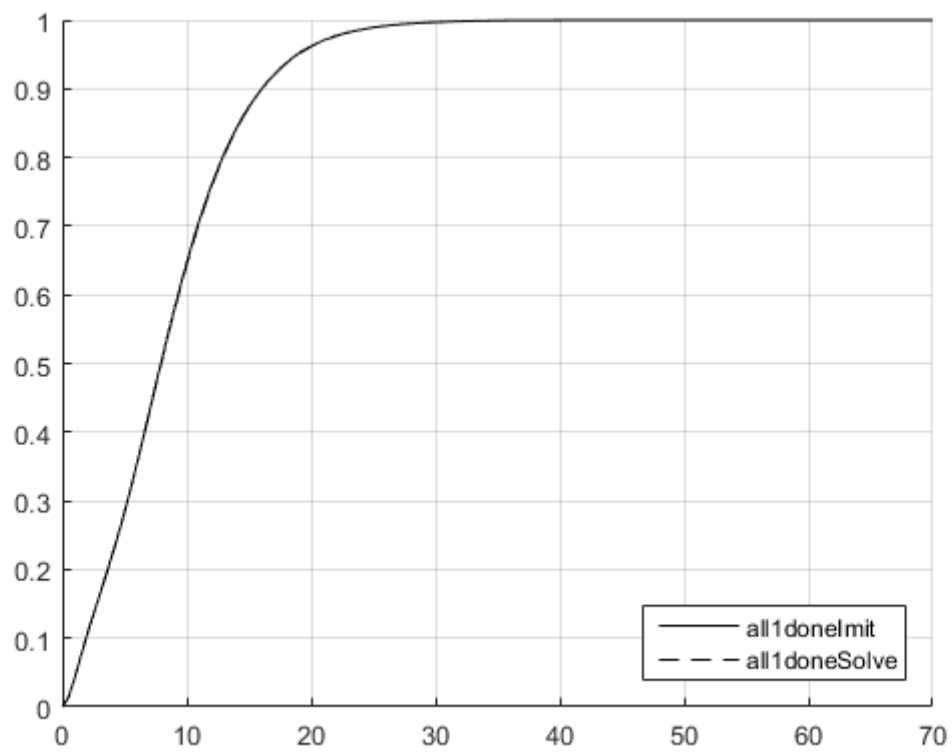


Рис. 6

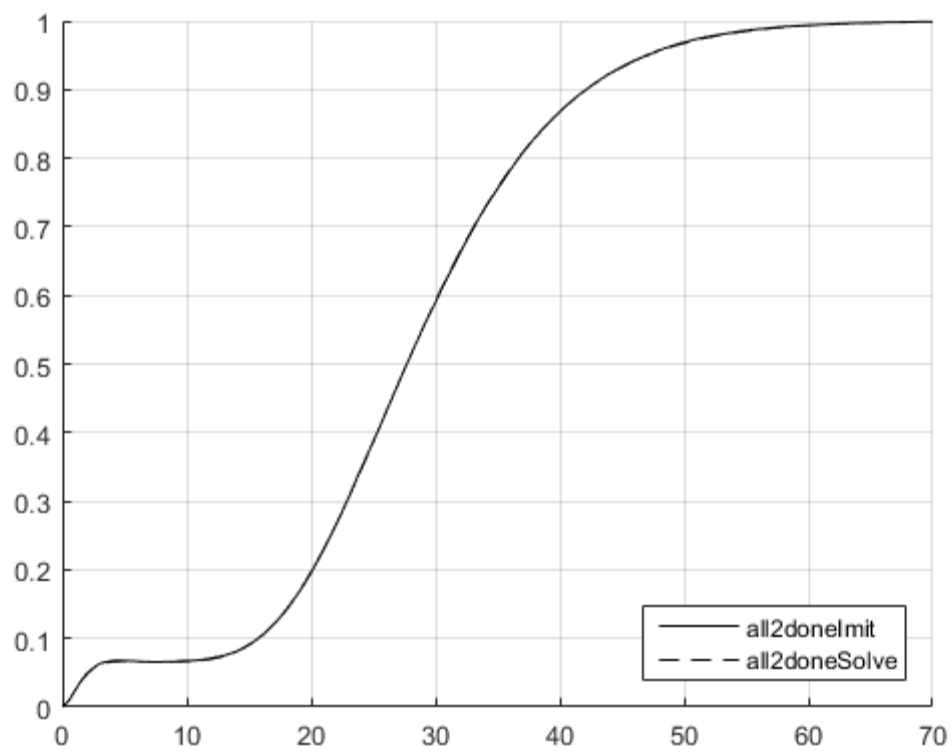


Рис. 7

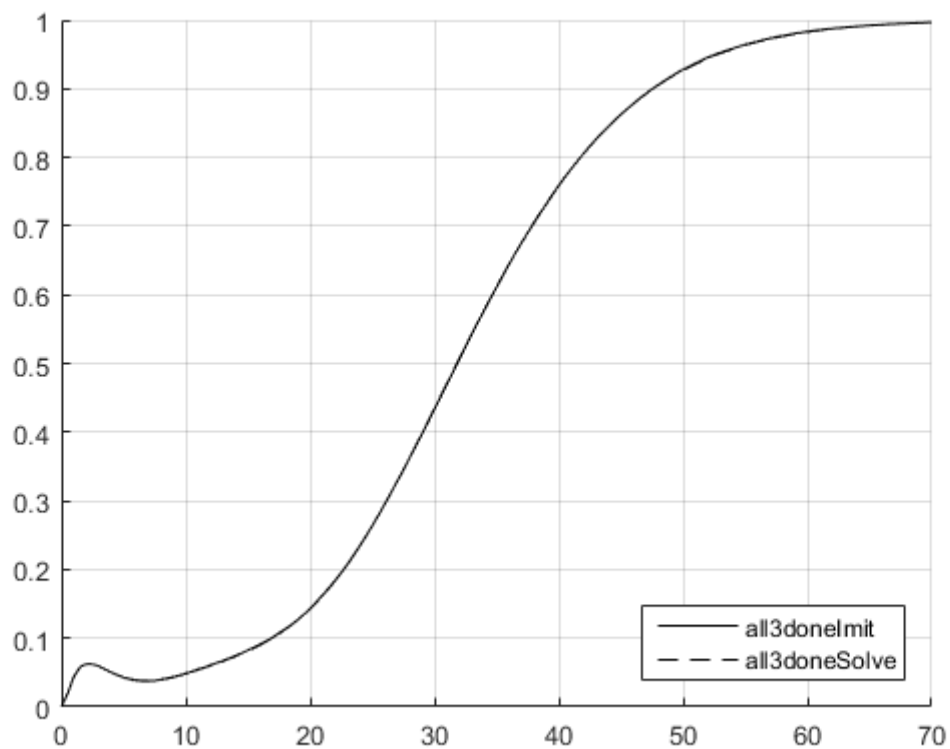


Рис. 8

Малые значения различия между кривыми, полученными с использования матрицы решения, и матрицы имитации, позволяют говорить о правильности выбора метода нумерации состояний и верном способе заполнения матрицы коэффициентов.

На рис. 9 приведены кривые вероятности того, что в системе находится в очереди более чем 2, 3 и 4 заявки. График построен основываясь на решении системы. Сплошной линией указана кривая, соответствующая вероятности нахождения в системе более 2-х необработанных заявок в очереди, пунктирной – более 3-х, штрихпунктирной – более 4-х.

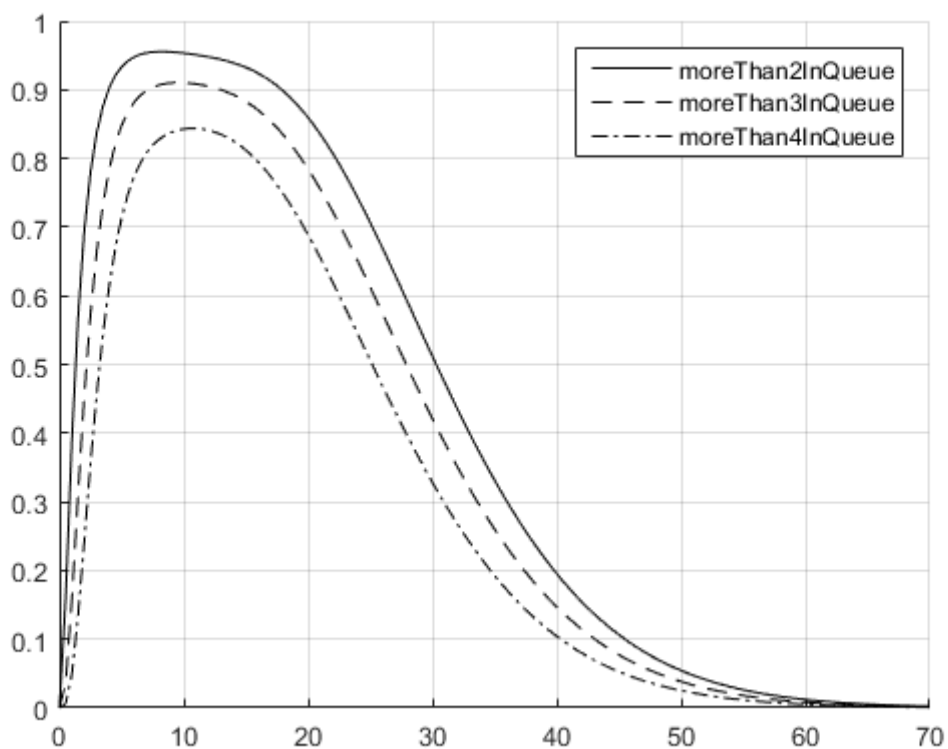


Рис. 9

Заключение

В данной работе описан алгоритм формирования списка состояний для НСО с абсолютными приоритетами таким образом, чтобы матрица A коэффициентов уравнения Чепмена – Колмогорова, описывающего данную систему, имела нижнетреугольный вид.

Правильность выполнения поставленной задачи показана, путем сравнения решения уравнения с результатами имитационного моделирования.

Для решения уравнения Чепмена – Колмогорова был реализован метод матричной экспоненты, имеющий очевидные недостатки. Одним из них является накапливающаяся величина погрешности округления, растущая вместе с увеличением рассматриваемого отрезка времени. Так же, в связи с разреженностью матрицы A , выбранный метод выполняет множество ненужных операций, что значительно увеличивает время работы программы. Чтобы максимально избежать описанных проблем, планируется использовать численно-аналитический метод, описанный в статье [1].

В дальнейшем планируется так же рассмотрение НСО с абсолютными приоритетами с несколькими каналами обслуживания, а так же с распределениями временных интервалов между моментами поступления и обработки заявок подчиняющимся гиперэкспоненциальному распределению, а так же распределениям Эрланга и Кокса.

Рост количества входных заявок увеличивает рассматриваемую систему на порядки, делая её крайне неудобной для исследования. В связи с этим возникает потребность в объединении состояний в группы по некоторому принципу, тем самым аппроксимируя исходную громоздкую систему. Выбор таких принципов так же входит в планы дальнейшего рассмотрения НСО с приоритетами.

Список литературы

1. Бубнов В. П., Еремин А. С., Сергеев С. А. Особенности программной реализации численно-аналитического метода расчета моделей нестационарных систем обслуживания // Труды СПИИРАН 2015. Вып. 38. С. 218-232.
2. Бубнов В. П., Сафонов В.И. Разработка динамических моделей нестационарных систем обслуживания. СПб.: Издательство «Лань». 1999. 64 с.
3. Бубнов В. П., Тырва А. В., Еремин А. С. Комплекс моделей нестационарных систем обслуживания с распределениями фазового типа // Труды СПИИРАН. 2014. Вып. 37. С. 61-71.
4. Вентцель Е.С. Теория вероятностей: Учебник для вузов. 6-е изд. стер. М.: Высш. шк., 1999. 576 с.
5. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания. Изд. 5-е, испр. М.: Издательство ЛКИ, 2011. 400с.
6. Кемени Д. Дж., Снелл Дж. Л. Конечные цепи Маркова. М.: Наука, 1970, 271 с.
7. Клейнрок Л. Теория массового обслуживания. Пер. с англ./ Пер. И. И. Грушко; ред. В. И. Нейман. М.:Машиностроение, 1979. 432с.
8. Матвеев В. Ф., Ушаков В. Г. Системы массового обслуживания. М.: Изд-во МГУ, 1984. 240 с.
9. Сафонов В. И. Моделирование систем обслуживания с изменяющейся интенсивностью поступления запросов. Деп. в войсковой части 11520 № В 216 // УПИМ. 1987. Вып. 3, сер, Б.
10. Хинчин А. Я. Работы по математической теории массового обслуживания / Под редакцией Б. В. Гнеденко. М.: Физматгиз, 1963. 236 с.