

В. М. Кичак, д. т. н., проф.; **М. В. Васильская**

РАЗРАБОТКА ОБОБЩЕННОЙ ИМИТАЦИОННОЙ МОДЕЛИ СИСТЕМЫ МОБИЛЬНОЙ СВЯЗИ НА БАЗЕ АГРЕГАТНОГО ПОДХОДА

Предложено и обосновано построение обобщенной, открытой для модификаций рабочей математической модели многоканальной системы массового обслуживания с произвольными дисциплинами обслуживания. Функциональные подсистемы имитационной модели построены как агрегаты. Целевое назначение модели – моделирование процессов в системах мобильной связи.

Ключевые слова: агрегат, агрегатная система, массовое обслуживание, имитационная модель, рабочая модель, распределение вероятностей, неопределенности, векторизация, многоканальная система.

Постановка проблемы. Системы мобильной связи сегодня являются не только весомым сектором индустрии и технологий, но и неотъемлемой составляющей всех организационно-производственных систем. Поэтому, безусловно, актуальными являются задачи прогнозирования и планирования развития систем мобильной связи. Высокие темпы инноваций, высокие уровни неопределенности, отсутствие статистики обуславливают необходимость использования математических моделей для функционирования таких систем. Одна из важных задач в моделировании систем мобильной связи – получение функций влияния внешних и внутренних факторов на эффективность систем мобильной связи. Существующие методы расчета очередей и пропускной способности систем мобильной связи требуют знания теоретических частотных распределений для интервалов и объемов поступления заявок (задач) и длительности времени обслуживания.

Постановка задачи. Объединяем этапы разработки математической модели и программы моделирования в разработку рабочих моделей. Рабочая модель – это продукт современных пакетов для моделирования вообще и специализированных пакетов для моделирования систем мобильной связи. Рабочая модель является фактически интерфейсом пользователя с программными модулями пакета, написанными на языке C++. Визуальное представление рабочей модели достаточно точно отвечает стандартам научных, в частности, математических публикаций.

Обоснование метода определения функций влияния

В работе выбирается базовая модель функционирования системы мобильной связи, как многоканальная система массового обслуживания (СМО) с произвольными и переменными дисциплинами обслуживания. В рамках известных аналитических моделей СМО невозможно удовлетворительно отобразить специфические особенности и сложность систем мобильной связи. Выбран путь разработки имитационной модели, которая отвечала бы требованиям вычислительной эффективности, модульности и пригодности к модификациям. В качестве средств реализации модели выбраны доступные математические пакеты Matlab, Mathcad, VisSim.

Сегодня эффективным подходом к имитационному моделированию сложных многоканальных систем является разбиение модели СМО на агрегаты и построение агрегатных моделей. Агрегат является обобщением элемента из теории динамических систем. Суть обобщения в том, что в составе агрегата могут быть непрерывные статические и динамические модели, импульсные, релейные, логико-динамические модели, модели конечных автоматов и модели неопределенности и случайных возмущений. В частности, это могут быть классические вероятностные модели и модели на базе нечеткой логики. Агрегат имеет входы и выходы, выступает как универсальный преобразователь информации.

Модель сложной системы, собранная из агрегатов – агрегатная модель системы,

получается методами декомпозиции и агрегации. В средах математических пакетов агрегат можно определить как функцию пользователя, что берет векторы параметров агрегата и входных переменных и возвращает вектор выходных переменных. Также можно эффективно реализовать функции оптимизации, поиска нулевых корней и состояний равновесия.

После определения и реализации агрегатов и агрегатных моделей можно оперировать с ними как с обычными аналитическими функциями (например, $\sin()$, $\arcsin()$, $\ln()$) – интегрировать их, искать экстремумы и нулевые точки, с определенными ограничениями находить производные. То есть, когда имитационная агрегатная модель построена, она позволяет просто находить функциональные зависимости показателей эффективности.

Кроме декомпозиции модели на элементы-агрегаты, применяем декомпозицию сложной имитационной модели во времени: создаем модель не одномоментно, а шаг за шагом. На каждом шагу существующая работоспособная модель незначительно усложняется, уточняется и проходит всестороннее тестирование. На модели также выполняются исследования – согласно целевому назначению. В практическом аспекте пошаговое создание модели позволяет избегать ошибок и, главное, позволяет получать новые знания о свойствах объекта моделирования, корректировать направления развития имитационной модели.

При разработке очередной модели, предыдущая модель остается как средство контроля более сложных моделей и как упрощенная модель для упрощенных задач. На базе разработанной модели могут создаваться имитационные модели конкретных систем мобильной связи.

Определение базовых понятий

Агрегатные модели. В литературе по моделированию не существует единых стандартных классификаций моделей. Не существует также единственного определения агрегатных моделей и агрегатных систем, поэтому уточняем и конкретизируем эти определения относительно объектов исследования и моделирования – систем мобильной связи. Рассмотрим класс сложных систем, которые являются конструкциями из агрегатов и имеют следующие свойства: существует такое, обычно не единственное разбиение системы на элементы, при котором каждый полученный элемент является агрегатом.

Определение агрегата в теории систем. Наиболее общее определение агрегата: заданы T – множество моментов времени, множества произвольной природы X, U, B, Z . Элементы указанных множеств имеют такие интерпретации: $t \in T$ – момент времени; $x \in X$ – входной сигнал; $u \in U$ – сигнал управления; $y \in Y$ – выходной сигнал; $z \in Z$ – состояние агрегата. Состояния, входы, выходы, управление являются функциями времени $z(t), x(t), u(t)$ и $y(t)$.

Агрегатом называется объект $\langle T, X, U, B, Z, H, G \rangle$, где H, G – операторы, могут быть случайными, нечеткими. Выходами операторов переходов и выходов H и G являются функции $z(t)$ и $y(t)$. Отличие агрегатов от конечных автоматов и обычных динамических систем в структуре этих операторов.

Операторы переходов и выходов. В направлении «агрегатные имитационные модели» существуют два подхода – абстрактный, где рассматриваются классы операторов с «особенными состояниями» с искусственными примерами без содержательных интерпретаций, и подход, где главная цель – создание эффективных программ имитационного моделирования для исследований и встраивания в автоматизированные системы управления. Последовательность этапов исследования в последнем подходе – создание работоспособных моделей, проведение интенсивных исследований на модели, потом теоретическое обобщение для создания очередного поколения имитационных моделей. Ни в реальных объектах, ни в моделях не должно быть «особенных состояний» временной или окончательной неработоспособности. Выбираем именно такой подход.

Процесс функционирования агрегата. Агрегат является математической схемой достаточно общего вида, частными случаями которой являются функции алгебры логики, релейно-контактные схемы, конечные автоматы, все классы систем массового обслуживания,

динамические и логико-динамические системы, которые описываются обычными дифференциальными уравнениями, и некоторые другие объекты. С точки зрения моделирования агрегат выступает как достаточно универсальный преобразователь, он воспринимает входные сигналы, которые управляют, и выдает исходные сигналы. В данной работе набор математических агрегатов расширяется за счет имитационных моделей сложных случайных событий и моделей нечеткой логики.

Агрегативные системы. Рассмотрим класс сложных систем, которые являются конструкциями из агрегатов, которые владеют следующим свойством: существует такая (в общем случае неоднозначная) разбивка системы на элементы, при которой каждый полученный элемент является агрегатом. Такого рода сложные системы будем называть агрегативными или А-системы. Желательно, но не обязательно, в реальной системе поданной агрегативной моделью иметь специализированные агрегаты для обмена информацией с внешним окружением. Соответственно, вся информация, циркулирующая в А-системы, делится на внешнюю и внутреннюю.

Для каждого агрегата описываются условия передачи информации – опоздания, передача, по запросу, имеются шумы и возмущения. Относительно структуры агрегатов, то все они описываются теорией графов. Свойства А-системы определяются не только свойствами агрегатов, но также и ее структурой. Нас связки между агрегатами интересуют только в аспекте замены определенных структур эквивалентными агрегатами. Естественно выделить последовательные, параллельные сочетания и циклы – обратные связки. Всегда возможно заменить такие структуры эквивалентными агрегатами. Это утверждение легко проверить на уровне программной реализации – объединением программных модулей для агрегатов, которые составляют структуры. Теоретическое обоснование утверждения возможно, но сопряжено с большим объемом процедур формализации сценариев функционирования агрегатов – опозданий, изменения состояний, возмущений и т.д.

Построим базовую агрегативную модель СМО, как интерпретацию теоретических положений.

Выбор технологии реализации рабочей модели

Выбираем технологию разработки базовой модели, максимально согласованную с технологией разработки программного обеспечения, – выбираем для описания СМО подход на базе "вектора состояния" – выбираем структуру вектора состояния, которая является компромиссом между удобством, точностью описания системы и размерностью, – выполняем иерархическую декомпозицию модели на малые функциональные модули, каждый модуль тщательным образом тестируем, прежде чем включить в главную программу, – модули имитации поведения формируем как операторы превращения текущего состояния в следующий; – при переходе от одномерных систем к многомерным используем векторизование вычислений [3].

В порядке обобщения такая последовательность моделей:

- $x_{k+1} = A \cdot x_k + B \cdot u_k$ – линейная система, объекты – векторы состояния;
- $Mu_{k+1} = Mu_k + Nast(Mu_k, dMu_k)$ – нелинейная система, объекты – матрицы состояния;
- $Mu_{k+1} = Op(M_k, U_k, P)$ – нелинейная система, объекты – произвольные матричные структуры, U_k, P – управление и параметры.

Произвольная матричная структура – матрица, элементы которой произвольные матричные структуры: скаляры, матрицы с числовыми и символьными переменными. В имитационные модели систем мобильной связи необходимо включать модели поведения пользователей. Как пример векторизации рассмотрим модуль нечеткого выбора с обучением, который берет скалярные параметры определенного элемента и возвращает скалярный параметр – вероятность выбора. Сначала делаем скалярный модуль.

$$dmu(vyb, bolv, nvz) = \begin{cases} kys \leftarrow vyb \cdot nvz \\ qq \leftarrow vyb \cdot bolv + rnd(kys) - 0.5 \cdot kys \\ qq \end{cases} \quad (1)$$

Применяем модуль (1) к обработке массива матричных объектов – массивам параметров этих потребителей (статистика использования услуг мобильной связи для каждого пользователя), на выходе тоже получаем массив выходных данных:

$$dMU = \overrightarrow{dmu(Vyb, Bolv, Nvz)} \quad (2)$$

Разработка агрегатных модулей СМО

Ставим две рабочие цели разработки: создание открытой библиотеки модулей для разных дисциплин обслуживания потока требований обслуживания, создания открытой библиотеки модулей имитации неопределенности. Пригодные для современных условий алгоритмы обслуживания должны быть простыми – "естественными", децентрализующими, отказоустойчивыми, адаптируемыми к статистике входящего потока задач.

Модуль "выбор распределений вероятностей". Модуль базируется на функциях, которые генерируют массивы чисел с заданным распределением вероятностей. Предусмотрена параметрическая настройка этих функций на статистику реальных входных данных. На рис. 1 подан пример реализации библиотеки неопределенности.

$kk := 3 \quad Fvx(kk) := \begin{cases} rbinom(1, 8, 0.9) & \text{if } kk = 1 \\ rpois(x, \lambda) & \text{if } kk = 2 \\ rnorm(1, 6, 1) & \text{if } kk = 3 \end{cases}$	$\begin{aligned} Fvx(1) &\rightarrow rbinom(1, 8, .9) \\ Fvx(2) &\rightarrow rpois(x, \lambda) \\ Fvx(3) &\rightarrow rnorm(1, 6, 1) \end{aligned}$
---	---

Рис. 1. Библиотека неопределенности (часть)

Модуль имитации неопределенности. На рис. 2 пример модуля имитации потока заявок на обслуживание – три случайных параметра: количество, объемы (длительности) и приоритеты обслуживания.

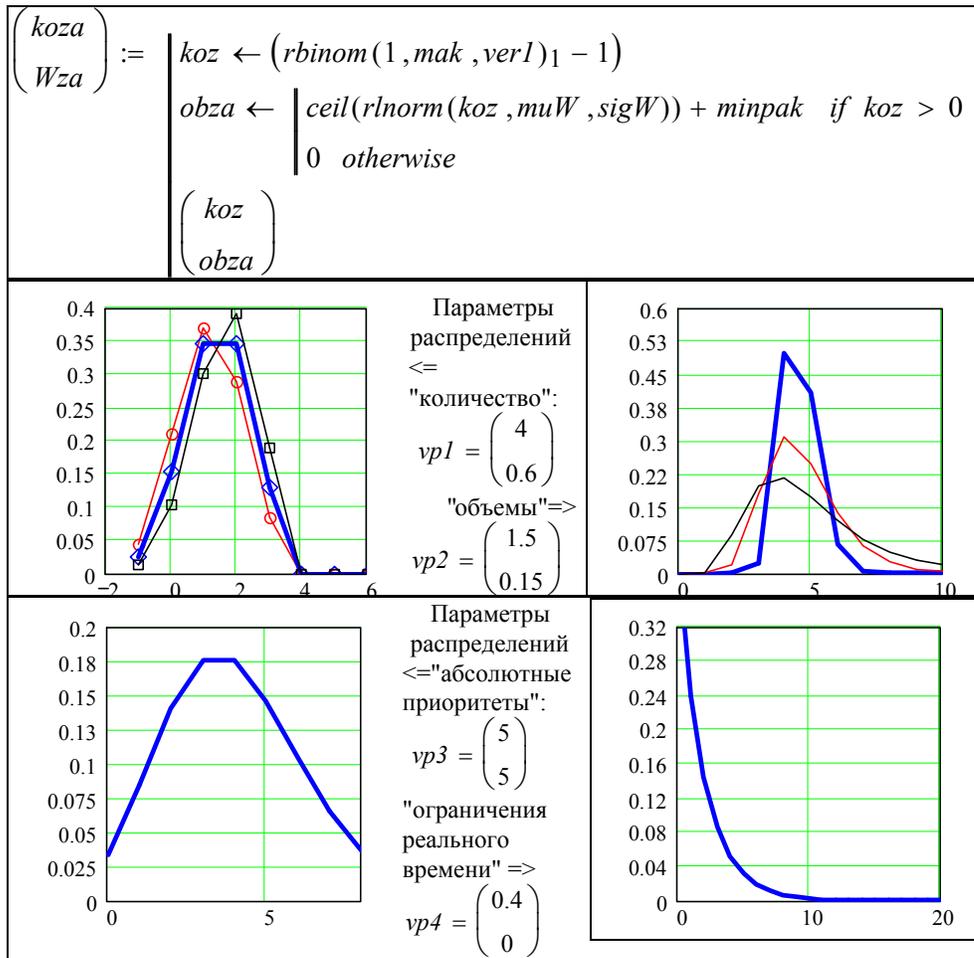


Рис. 2. Модуль имитации потока заявок на обслуживание

В классической литературе по СМО даются детальные классификации по типам распределений вероятностей входных потоков задач и длительностей обработки. Причина этого в том, что проводился поиск аналитических решений индивидуально для каждого вида распределения. Сделанный модуль генерации входного потока задач может быть настроен на произвольные распределения вероятностей.

Модули формирования состояния очереди и приоритетов

Радикально решим проблему выбора оптимального управления приоритетами задач в очереди: сформируем приоритет как взвешенную сумму от приоритетов, определенных по разным алгоритмам, которые обычно используют в вычислительных системах [4].

$$Ps_i = \sum_{p=1}^{Np} \alpha_p \cdot prior_{i,p}, \quad (3)$$

где α_p – взвешивающий коэффициент по p -тому показателю приоритета $prior_{i,p}$ – приоритет i -ой задачи по p -тому показателю приоритета. Структура данных "очередь с приоритетами" состоит из входных данных и вычисляемых данных – приоритетов. Подаем очередь матрицей, где:

- первый столбец – индексы "натуральной очереди";
- второй столбец – объемы задач в натуральном порядке;
- третий столбец – ожидание в очереди (шагов, тактов);

- четвертый – ограничение реального времени (максимальное опоздание обработки);
- пятый – **абсолютные приоритеты**, которые интерпретируются как важность задачи.

В столбцах 6, 7, 8 расположены **относительные приоритеты**, которые вычисляются и могут изменяться в течение ожидания в очереди:

- 6 – приоритет размера задачи: чем меньше размер задачи, тем выше приоритет;
- 7 – приоритет ожидания в очереди: чем более длинное ожидание, тем выше приоритет;
- 8 – приоритет реального времени: чем меньше времени до момента обработки заявки, тем выше приоритет задачи.

В последнем столбце 9 – суммарный взвешенный приоритет, который вычисляется с помощью (3).

Взвешенная свёртка (3) имеет еще одно назначение: обнулив все, кроме одного, коэффициенты, можем реализовать и исследовать «в чистом виде» СМО с одним определенным приоритетом. Наличие "библиотеки приоритетов" образует "базу знаний" для построения адаптивных систем.

Динамика приоритетов. Определенные данные в матрице $Ps\theta$ зависят от других или изменяются во времени согласно определенным механизмам. Классифицируем эти изменения [5]:

- 1) исчезновение строк очереди в результате обработки задач согласно приоритетам;
- 2) появление новых строк при поступлении новых задач с соответствующими приоритетами;
- 3) переисчисление приоритетов на каждом шаге процесса: увеличение ожидания в очереди, увеличение приоритетов для задач реального времени.

В итоге нам нужно получить оператор перехода между текущим и следующим состояниями структуры данных "очередь с приоритетами". Сначала сделаем модули, которые описывают превращение отдельных фрагментов структуры данных – столбцов, строк.

Модуль вычисления приоритетов по размерам задач берет второй столбец $Ps\theta^{(2)}$ (объемы задач) структуры данных, а возвращает нормируемый вектор приоритетов.

Модуль вычисления среднего взвешенного приоритета берет столбцы 5 – 8 матрицы $Ps\theta$, а возвращает средний взвешенный вектор приоритетов, который размещается в последнем столбце матрицы $Ps\theta$.

Результатом работы рассмотренных выше модулей является формирование столбцов 6 – 9. В результате формируется структура, удобная для программной реализации широкого класса алгоритмов управления процессом обработки потока задач.

Модуль обработки очереди. Агрегатная модель обработки очереди, кроме функции определения приоритетов и выбора дежурной задачи для обработки, должна с каждым шагом обработки упорядочивать матрицу состояния очереди. Это можно сделать по-разному: обнулить обработанные строки, или обрезать, или игнорировать, сортировать за разными переменными, а затем сортировать назад. Обычно будет выбран наихудший вариант, если не идти шагами "модуль менее семи строк" и не проверять каждый шаг. Испытано несколько альтернатив и обосновано экспериментами последнюю альтернативу. Лингвистическая модель процесса пополнения и обработки очереди:

- на каждом шаге поступают новые задачи (выход модуля имитации потока задач);
- на каждом шаге обрабатывается очередь задач в объеме пропускной способности.

Возможны ситуации обработки очереди:

- 1) текущая очередь = 0; \implies следующая очередь = 0;
- 2) текущая очередь \leq пропускной способности \implies следующая очередь = 0;
- 3) пропускная способность < объем первой задачи \implies остаток первой задачи в следующем шаге еще будет в обработке;

Конечная цель разработки программы моделирования – сделать такую структуру и такой оператор (функцию пользователя), чтобы динамика системы описывалась таким разностным

уравнением превращения состояния системы:

$$ss^{(k+1)} = Y_{pur}P(ss^{(k)}), \quad (4)$$

$$\text{или } ss^{(k+1)} = Y_{pur}P1\left(ss^{(k)}, u\left(ss^{(k)}\right)\right). \quad (5)$$

Модуль обработки очереди берет пропускную способность СМО и вектор сортированной за выбранными приоритетами очереди задач, а возвращает новый вектор очереди (это отображено в определениях $ss3$, $S3$ – определение соответствующего компонента структуры состояния ВС в главной рабочей модели). Словесное описание процесса обработки очереди:

- делаем "заглушку" для случая нулевой очереди: $nulka \leftarrow (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$;
- определяем количество заявок $Okol$ и объем $Odlin$ очереди;
- сортируем матрицу состояния очереди по выбранному приоритету (переменная u);
- формируем условие "очередь не нулевая".

На рис. 3 представлен текст модуля, который реализует это словесное описание.

$smHP(props, ss3, u) :=$	$nulka \leftarrow (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$ $Okol \leftarrow rows(ss3)$ $srti \leftarrow csort(ss3, u)$ $Odlin \leftarrow \sum_{i=1}^{Okol} srti_{i,2}$ $yslo \leftarrow Odlin > props$ $S3 \leftarrow \begin{cases} sjelo(props, srti, u) & \text{if } yslo \\ nulka & \text{otherwise} \end{cases}$ $S3$
--------------------------	---

Рис. 3. Агрегатная модель «модуль обработки очереди»

Аналогично были разработаны агрегатные модели для следующих функциональных модулей.

Модуль пополнения очереди, которая принимает в очередь новые задачи, вычисляет значение относительных приоритетов задач, вычисляет динамические показатели задач очереди: ожидание в очереди, критическое время для задач реального времени.

Модуль динамики приоритетов, которая отслеживает и вычисляет переменные относительные приоритеты.

Модуль "оператор перехода для СМО с приоритетным обслуживанием". Этот модуль использует предыдущие модули-агрегаты, схема связей которых дана на рис. 4. Словесное описание процесса перехода от предыдущего состояния в следующий:

- определяется результат обработки предыдущей очереди $S30 \leftarrow smH(props, ss3)$;
- на каждом шаге обрабатывается очередь задач в объеме пропускной способности;
- определяются длина $S4 \leftarrow length(S30)$ и объем $S5 \leftarrow mean(S30) \cdot S4$ очереди после обработки;
- с помощью модуля "пополнения очереди": $BY \leftarrow koWza(vp1, vp2)$ определяется количество $S1 \leftarrow BY_1$ и объем $S2 \leftarrow BY_2$ новых задач, которые пришли в течение шага процесса;
- формируется очередь на следующий шаг $S31 \leftarrow stack(S30, S2)$ (новые задачи ставятся в

"хвост" очереди);

– формируется выход – следующий вектор состояния вычислительной системы.

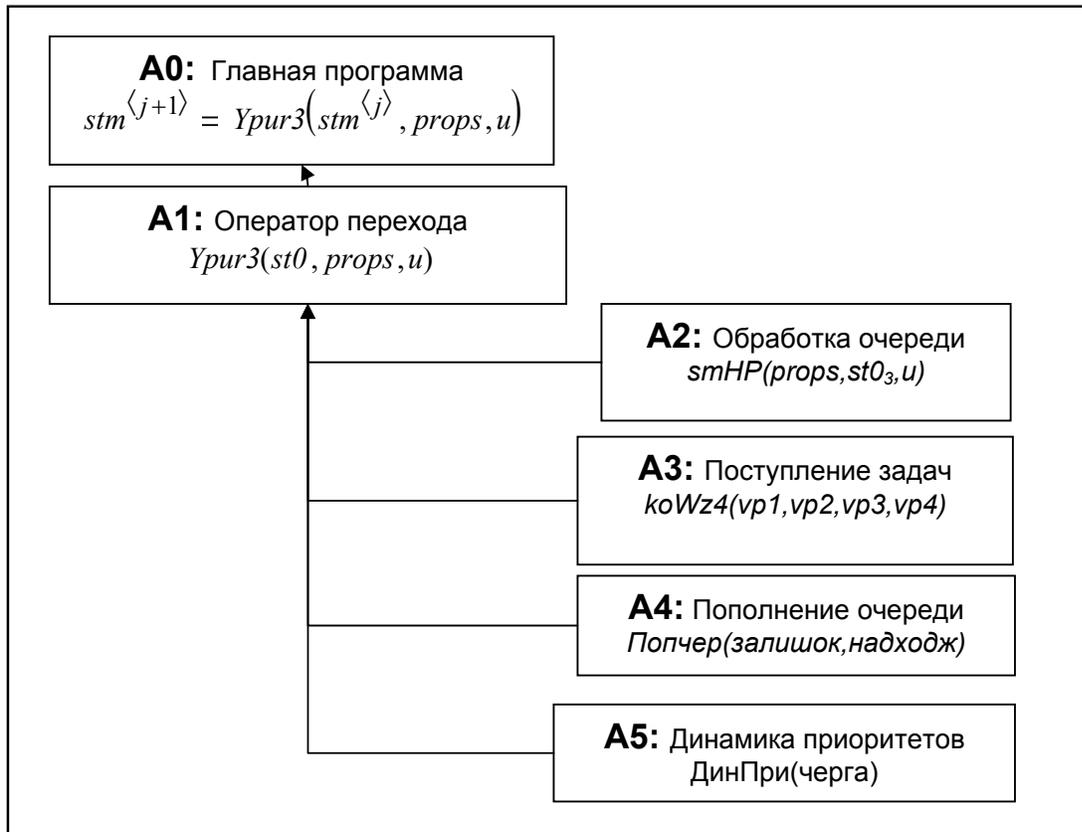


Рис. 4. Структура агрегативной модели «система массового обслуживания с произвольными приоритетами и произвольными распределениями параметров обслуживания»

Для данной задачи существовала альтернатива – сделанная при помощи традиционной технологии программа моделирования. Она не была доказана из-за сложности логики работы СМО в целом.

Агрегативная имитационная модель была доведена до работоспособного, состояния и на ней проведен большой объем исследований. На рис. 5 и 6 даны два примера моделирования процессов обслуживания с разными приоритетами.

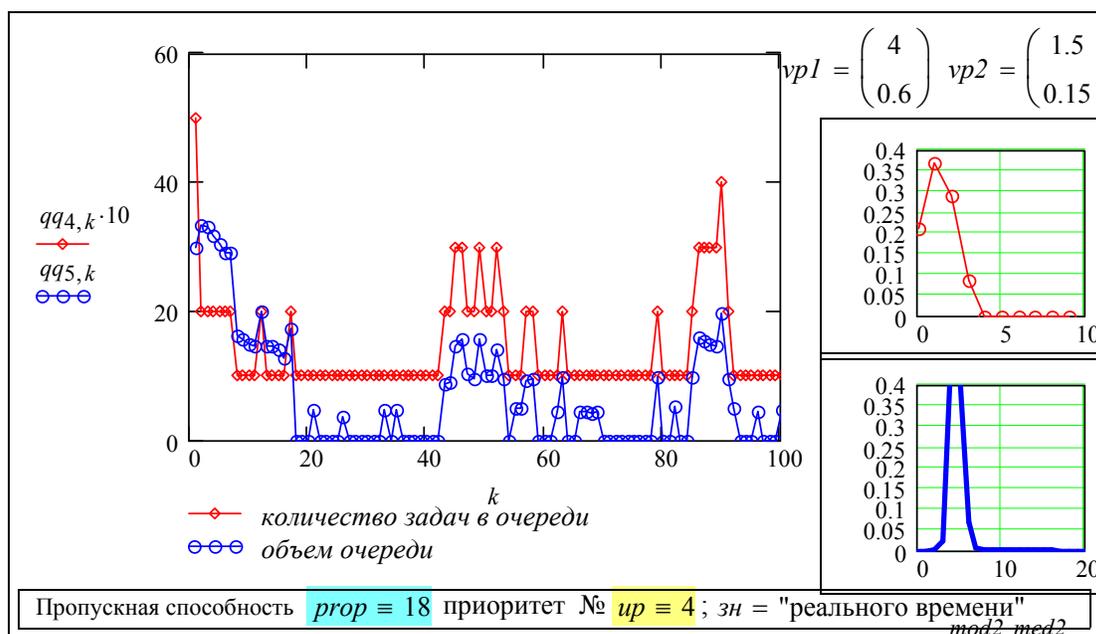


Рис. 5 Пример результатов моделирования. Процесс для приоритета обслуживания «реального времени»

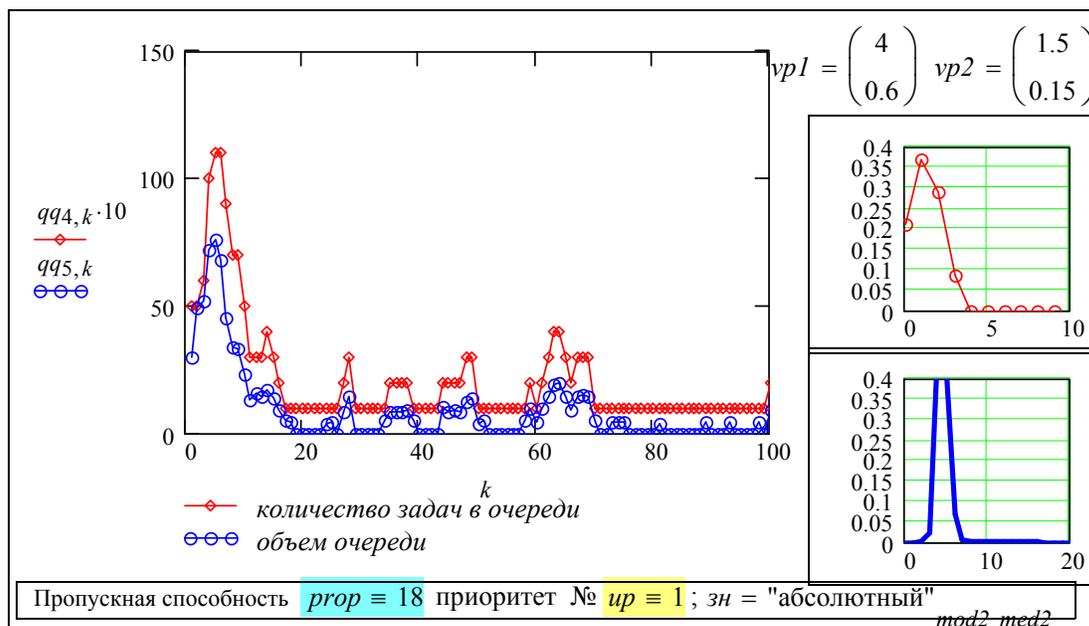


Рис. 6. Пример результатов моделирования. Процесс для приоритета обслуживания «абсолютный»

Практическое значение разработки модели и проведенных исследований – определение направлений сбора данных в реальных системах мобильной связи для настройки и модификации данной имитационной модели, создания модели-предиктора, для реальных систем.

Теоретическое значение – определение направлений теоретического обоснования обнаруженных свойств системы массового обслуживания, обобщения теоретических моделей систем массового обслуживания. Перспективным направлением является получение и анализ статистики виртуальной реальности и получения функций влияния.

СПИСОК ЛИТЕРАТУРЫ

1. Нэгл Томас Т. Стратегия и тактика ценообразования. Руководство для принятия решений, приносящих прибыль / Томас Т. Нэгл. – М.: Питер, 2001. – 375 с.
2. Kelly K. New Rules for the New Economy. 10 radical strategies for a connected world. – Penguin books, 1999. – 180 p.
3. Форрестер Дж. Основы кибернетики предприятия / Дж. Форрестер. – М.: Прогресс, 1971. – 340 с.
4. Боровська Т. М. Основи теорії управління та дослідження операцій. Навчальний посібник / Т. М. Боровська, І. С. Колесник, В. А. Северілов. – Вінниця: УНІВЕРСУМ-Вінниця, 2008. – 242 с. – ISBN 978-966-641-275-4.
5. Моделювання та оптимізація у менеджменті: Навчальний посібник / Т. М. Боровська, В. А. Северілов, С. П. Бадьора, І. С. Колесник. – Вінниця: УНІВЕРСУМ-Вінниця, 2009. – 145 с. – ISBN 978-966-641-287-7.

Кичак Василий Мартынович – д. т. н., профессор кафедры телекоммуникационных систем и телевидения; институт радиотехники, связи и приборостроения.

Васильская Майя Валериевна – аспирант кафедры телекоммуникационных систем и телевидения; институт радиотехники, связи и приборостроения.
Винницкий национальный технический университет.