В.А. Михеев, В.М. Черненький, П.Н. Шкатов

ПРОЕКТИРОВАНИЕ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

МЕТОДЫ И АЛГОРИТМЫ РАСЧЕТА

Под редакцией В.М. Черненького

Рекомендовано Федеральным учебно-методическим объединением
в системе высшего образования
по укрупненной группе специальностей и направлений
подготовки 09.00.00 «Информатика и вычислительная техника»
в качестве учебного пособия для студентов, обучающихся
по основным образовательным программам высшего образования
по направлению подготовки магистров 09.04.01
«Информатика и вычислительная техника»

Москва
Радиотехника
2017

УДК 658.012.011.56
ББК 65.050.2
М 69

Р е ц е н з е н т ы :
А.Н. Данчул *– докт. техн. наук, профессор;*
А.В. Остроух *– докт. техн. наук, профессор*

Михеев В.А., Черненький В.М., Шкатов П.Н.

М 69 Проектирование корпоративных информационных систем. Методы и алгоритмы расчета. Учебное пособие / Под ред. *В.М. Черненького*. – М.: Радиотехника, 2017. – 176 с.

ISBN 978-5-93108-163-2

Представлены методы формализации постановки задач проектирования сложных корпоративных информационно-вычислительных систем; методы построения стохастических моделей для расчета и характеристик, методы агрегирования и декомпозиции. Подробно рассмотрены примеры, а также даны контрольные задачи для самостоятельного решения.

Для студентов вузов, обучающихся по магистерской программе «Проектирование корпоративных информационных систем», преподавателей вузов, а также специалистов, работающих в области разработки информационных систем.

УДК 658.012.011.56
ББК 65.050.2

ISBN 978-5-93108-163-2

ОГЛАВЛЕНИЕ

ОТ РЕДАКТОРА

Укрупнение производств, создание сетевых структур, возникновение корпораций потребовали создание информационных технологий по разработке комплексных АСУ. Если ранее делался упор на разработку АСУ предприятий, то сейчас – на создание корпоративных многофункциональных интегрированных АСУ. Проектирование последних требует использования математических моделей высокой размерности, что неизменно приводит к созданию методик агрегирования показателей и характеристик, построению систем иерархических моделей и вложенных описаний.

Предлагаемое учебное пособие является основополагающим для новой магистерской программы «Проектирование корпоративных информационных систем» в рамках направления подготовки 09.04.01 «Информатика и вычислительная техника».

Задача пособия – задать вектор обучения по программе «Проектирование корпоративных информационных систем» в целом. Для этого *в пособие включены разделы*, содержащие:

- описание объекта проектирования в виде многофункциональной информационной системы интегрированных структур;
- изложение основ концепции проектирования на основе методов исследования операций (классификация параметров, построение критериев, свертки показателей, композиция, декомпозиция);
- классификация методов моделирования на основе моделей описания процессов;
- изложение методов теории массового обслуживания для расчета характеристик сложных информационных систем.

Большая часть материалов пособия посвящена методам исследования с использованием теории массового обслуживания,

как наиболее сложным в изложении и усвоении студентами. Поэтому особое внимание уделено систематизации методов в их взаимосвязи с задачами проектирования.

Предлагается магистерская программа, включающая такие дисциплины как «Архитектура корпоративных систем. Исследование операций», «Теория массового обслуживания», «Методы описания процессов», «Имитационное моделирование», «Статистический анализ», «Технологии проектирования корпоративных информационных систем».

В каждом разделе пособия содержатся примеры использования рассматриваемых методов, а также предлагаются задачи для самостоятельной проработки. Кроме того, разработаны конкретные задачи проектирования корпоративных информационных систем и их решение с помощью прикладных пакетов позволяет обучать студента умению применять предлагаемые методы, а также прививать ему навыки решения конкретных задач проектирования.

Учебное пособие соответствует требованиям Федерального государственного образовательного стандарта подготовки магистров по направлению 09.04.01 «Информатика и вычислительная техника». Виды профессиональной деятельности, к которым готовятся выпускники вузов, освоившие эту программу, определены как научно-исследовательская и проектная.

Выпускник по окончании освоения магистерской программы должен быть готов

1. Решать профессиональные задачи:

- Разработка математических моделей исследуемых процессов и изделий.
- Разработка методик проектирования новых процессов и изделий.
- Концептуальное проектирование сложных изделий, включая программные комплексы, с использованием средств автоматизации проектирования.
- Разработка и реализация проектов по интеграции информационных систем в соответствии с методиками и стандартами информационной поддержки изделий.

2. Обладать общекультурными компетенциями:

- способностью совершенствовать и развивать свой интеллектуальный и общекультурный уровень;

- способностью заниматься научными исследованиями;
- использовать на практике умения и навыки в организации исследовательских и проектных работ.

3. Обладать общепрофессиональными компетенциями:

- культурой мышления;
- способностью выстраивать логику рассуждений и высказываний, основанную на интерпретации данных, интегрированных из разных областей науки и техники;
- выносить суждения на основании неполных данных.

4. Обладать профессиональными компетенциями:

- знанием методов научных исследований и владение навыками их проведения;
- способностью проектировать распределенные информационные системы, их компоненты и протоколы их взаимодействия;
- способностью выбирать методы и разрабатывать алгоритмы решения задач проектирования объектов автоматизации.

В.М. Черненький

ВВЕДЕНИЕ

Современное развитие экономики привело к созданию крупных научно-производственных интегрированных структур (концерны, холдинги, корпорации), которые объединяют предприятия смежных секторов промышленности, что позволяет *интегрировать научно-технический, производственный и экономический потенциал этих предприятий*. Условием интеграции является создание эффективной системы сбора, накопления, обработки, хранения и передачи больших объемов разнообразной информации. Такая система дает возможность объединить в единое информационное пространство различные программно-технические комплексы, средства и системы предприятий, и обеспечить удовлетворение ее информационных потребностей при разработке, производстве и реализации гражданской продукции и продукции двойного назначения. Таким образом, непрерывно возрастающие требования современных приложений обработки информации, необходимость построения единого информационного пространства ИВС определяют необходимость создания высокоэффективных *корпоративных информационных систем интегрированных структур*, которые, в свою очередь, строятся на основе объединения территориально распределенных локальных вычислительных сетей (ЛВС) предприятий.

Современные корпоративные информационные системы (КИС)[*] являются сложными системами, для которых на всех эта-

[*] В пособии термин *информационные вычислительные системы* (ИВС) будет использоваться ко всему классу систем обработки информации; термин *автоматизированные системы управления* (АСУ) к классу ИВС с включением функций поддержки управленческих решений; термин *корпоративные информационные системы* (КИС) к классу АСУ, ориентированных на поддержку функционирования крупных интегрированных структур.

пах жизненного цикла (при обосновании технических требований, проектировании, испытаниях, эксплуатации, модернизации и развитии, анализе опыта создания и эксплуатации) учитываются параметры и взаимодействие вычислительных средств, системы передачи данных, математического и информационного обеспечения, периферийного оборудования, иерархически организованных групп персонала, обеспечивающих, наряду с программно-техническими средствами, надлежащее выполнение задач, возложенных на систему, а также параметры групп пользователей, направляющих запросы на решение задач, в которых используются разнообразные ресурсы системы.

Анализ современных КИС показывает, что их развитие идет в следующих направлениях:

- Объединение в единое информационное пространство всех участников бизнес-процессов.
- Повышение уровня предъявляемых требований.
- Расширение перечня функций КИС, технологий, методов и средств, их реализующих.
- Автоматизация бизнес-процессов.
- Обеспечение информационной поддержки продукции на всех этапах жизненного цикла.
- Поддержание необходимого уровня доступности критически важных информационных сервисов и минимизация времени простоя.
- Обеспечение требуемого уровня защищенности, а также адекватное реагирование на возникающие угрозы и попытки несанкционированного доступа к ресурсам.

В то же время, существующие информационные системы обладают целым рядом недостатков, основными из которых являются их узкая специализация, отсутствие гибкости и адаптации к изменению требований пользователей и прикладных процессов, а также низкая эффективность использования сетевых ресурсов.

КИС должна обеспечить:

- Интеграцию управления бизнес-процессами, проектно-конструкторскими разработками и технологическими процессами производства выпускаемой продукции.

- Полное удовлетворение в реальном масштабе времени информационных потребностей всех предприятий КИС.
- Высокую гибкость и адаптируемость КИС как к изменению уровня требований пользователей к качеству обслуживания, так и к появлению новых услуг, требования которых еще четко не определены.
- Повышение эффективности использования сетевых ресурсов.
- Снижение затрат на проектирование, строительство и эксплуатацию.

Существуют две важные проблемы, которые необходимо учитывать при создании таких систем:

- Разработка КИС невозможна вне системного подхода, без глубокого анализа общих принципов построения и методов системного подхода и принятия решений при построении КИС.
- Разработка КИС невозможна без знания методов и технологий ее формального описания и оценки эффективности как системы в целом, так и прикладных процессов, реализуемых ею.

Для успешного решения этих проблем в первую очередь необходимо уметь количественно оценивать параметры производительности и надежности как системы в целом, так и ее составных частей. Оценка параметров, характеризующих указанные свойства системы, как и при анализе других сложных систем, должна осуществляться на основе сочетания расчетов, производимых с помощью программных средств общего назначения, а также использования программно реализованных аналитических моделей и имитационных моделей [21, 22].

Концептуальной основой проектирования сложных систем являются *методы агрегирования и декомпозиции*.

При исследовании, использующем агрегирование и декомпозицию, неизбежно введение упрощающих допущений, аппроксимации и приближенных оценок. Снижение точности оценок при этом компенсируется тем, что исследование моделей, использующих агрегирование и введение обобщенных параметров, приводит к лучшему пониманию процессов в системе в целом.

В главе 1 «Математические модели в задачах проектирования корпоративных систем» сначала в сжатой форме излагаются концептуальные основы создания многофункциональных инфор-

мационных систем интегрированных научно-производственных структур, формулируются основные задачи и принципы, которые необходимо учитывать при их построении.

Далее на основании идей теории исследования операций предлагается методология формализованной постановки задач, возникающих в практике построения сложных информационных систем, позволяющая использовать в качестве инструмента анализа математические модели. Излагается схема двух вариантов организации последовательности решения этих задач, использующих принцип декомпозиции.

В последующих главах рассмотрены модели описания процессов функционирования информационных систем, представлены инженерные методики расчета параметров производительности, эффективности и надежности сложных информационных систем, в которых используются различные типы стохастических моделей. Изложение идет, начиная от простых марковских моделей (одномерные процессы размножения-гибели), для анализа стационарного состояния которых используются прямые расчетные алгоритмы, с переходом к более сложным моделям, для которых предлагаются разновидности методов агрегирования и квазиэквивалентного укрупнения состояний модели, с помощью которых сложная модель представляется в виде совокупности одномерных субмоделей.

При описании важных в практике проектирования систем обслуживания с приоритетами (полумарковские модели) предлагаются методы формализованного описания широкого класса приоритетных дисциплин, позволяющие в явном виде получить расчетные соотношения выходных параметров. Рассмотрены также более сложные, многоресурсные модели (стохастические сети и многоуровневые модели). Предложены алгоритмы исследования информационных систем на основе использования стохастических моделей.

ГЛАВА 1

МАТЕМАТИЧЕСКИЕ МОДЕЛИ В ЗАДАЧАХ ПРОЕКТИРОВАНИЯ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

1.1. Функциональные свойства КИС

Формирование корпоративных научно-производственных интегрированных структур (КИС), осуществляющих координацию и реализацию важных проектов (программ), оптимизацию и повышение концентрации производства, совершенствование корпоративного управления и уровня менеджмента является естественным этапом современного развития передовых отраслей промышленности и экономики.

Такие научно-производственные структуры (концерны, холдинги, корпорации) объединяют предприятия смежных секторов промышленности, что позволяет интегрировать научно-технический, производственный и экономический потенциал этих предприятий. *Условие интеграции* – создание эффективной системы сбора, накопления, обработки, хранения и передачи больших объемов разнообразной информации.

> Такая *корпоративная информационная система* интегрированных структур (КИС) позволяет объединить в единое информационное пространство различные программно-технические комплексы, средства и системы предприятий и обеспечить удовлетворение ее информационных потребностей при разработке, производстве и реализации новых образцов техники, продукции двойного назначения, военной и гражданской продукции.

Методология построения КИС заключается в организации процесса ее построения и обеспечении управления этим процессом для того, чтобы гарантировать выполнение требований как к самой КИС, так и к характеристикам процесса ее разработки [7, 22].

Основные задачи при построении КИС

- Обеспечение создания систем, отвечающих поставленным целям и предъявляемым требованиям.
- Гарантия создания КИС с заданными параметрами в течение заданного времени в рамках стоимостных ограничений.
- Простота сопровождения, модификации и расширения КИС с целью обеспечения ее соответствия изменяющимся условиям работы.
- Обеспечение создания КИС, отвечающих требованиям открытости, переносимости и масштабируемости.
- Возможность использования в создаваемой КИС разработанных ранее средств информационных технологий (программного обеспечения, баз данных, средств вычислительной техники, телекоммуникаций).

Основные принципы,
которые необходимо учитывать
при создании КИС, применительно к КИС

- *Принцип целеобусловленности* основывается на задании цели (множества целей) и механизма достижения цели, количественно характеризующего в каждый момент времени степень соответствия поведения КИС заданной цели. Цель (множество целей) задает система верхнего уровня по отношению к исследуемой КИС.

Применительно к КИС системой верхнего уровня выступает КИС. В связи с этим в качестве основной меры достижения заданной цели (целей) КИС должны выступать критерии эффективности интегрированных структур.

- *Принцип системного (комплексного) подхода* основывается на системном анализе как КИС, так и внешней среды. Это означает, что должны быть определены цели и критерии эффективности и осуществлена структуризация, определяющая весь комплекс вопросов, которые следует решить. Решение комплекса вопросов создаваемой КИС ИС при этом должно

наилучшим образом соответствовать установленным целям и критериям. В этот комплекс должны быть включены вопросы не только технического, но также организационного и экономического характера.

- *Принцип целостности*, в соответствии с которым КИС, как единое целое, должна обладать конечным множеством особых, системных свойств, которых нет у ее подсистем при любом способе декомпозиции. КИС, как сложная система, не может быть сведена к простой совокупности подсистем и, исследуя каждую из подсистем в отдельности, нельзя оценить все свойства КИС в целом.
- *Принцип открытости* предполагает включение новых информационных ресурсов в КИС. Применение этого принципа позволит обеспечить дальнейшее совершенствование, наращивание возможностей КИС и ее интеграцию с другими информационными системами.
- *Принцип этапности* предполагает установление определенной последовательности проведения работ по созданию КИС ИС. Применение этого принципа позволит упорядочить работы и рационально распределить временные, материальные и финансовые ресурсы, необходимые для создания КИС.
- *Принцип адаптивности* означает возможность наращивания и изменения функциональных возможностей в соответствии с вновь выявившимися потребностями или структурными изменениями без дополнительной переработки общего программного обеспечения, специального программного обеспечения и технических средств.
- *Принцип автономности* определяет, что из состава КИС ОПК могут быть выделены относительно самостоятельные объекты (подсистемы), обладающие системозначимыми свойствами.

Принципы целостности и автономности в совокупности позволяют найти способ декомпозиции КИС, способствующий снижению сложности ее анализа и синтеза.

- *Принцип иерархичности* означает, что КИС должна представляться в иерархической форме и рассматриваться как самостоятельная система и как часть (подсистема) другой,

большей системы, в которую она входит. Данный принцип полностью соответствует иерархической структуре КИС и требует ее рассмотрения в тесной связи с КИС и взаимодействующими информационными системами. Из данного принципа вытекает важное следствие, что при оценке и оптимизации КИС, как самостоятельной системы, наряду с внешними критериями может свободно выбираться совокупность внутренних критериев, согласованных с внешними.

- *Принцип непрерывного развития* связан с появлением изменений и инноваций во внешней и внутренней среде КИС, что влечет за собой появление новых задач и решений. В целях быстрого реагирования на изменения и инновации, они должны обладать возможностями автоматизации программирования и перекомпоновки базы данных и информационных файлов. Создаваемые комплексы рабочих программ должны строиться таким образом, чтобы в случае необходимости можно было просто менять не только отдельные программы, но и критерии, по которым ведется управление.
- *Принцип преемственности* предполагает рациональное использование существующих технических средств и систем автоматизации, информационных ресурсов, научного и методического задела при создании КИС. Применение этого принципа позволит уменьшить материальные затраты, сократить сроки ее создания.
- *Принцип доступности* предполагает возможность получения необходимой информации в требуемом виде независимо от ее местоположения. Применение этого принципа позволит обеспечить оперативное и полное предоставление данных в соответствии с информационными потребностями КИС.
- *Принцип защищенности* предполагает возможность противостояния несанкционированному доступу к информационным ресурсам. Применение этого принципа позволит обеспечить информационную безопасность КИС.
- *Принцип управляемости* предполагает, что КИС должна быть управляемой, несмотря на сложность ее поведения и способность к самоорганизации. Для обеспечения управляемости должен реализовываться механизм управления в виде

различных способов применения при автоматизации процессов, функций и задач интегрированных структур, возложенных на КИС.

- *Принцип моделируемости* предполагает, что КИС может быть описана конечным множеством моделей. Этот принцип дает возможность исследовать определенное свойство или группу свойств при помощи одной или нескольких упрощенных моделей.

Анализ приведенных принципов определяет необходимость использования различных математических моделей на разных этапах проектирования и жизненного цикла КИС ИС.

1.2. Формализация постановки задач проектирования КИС

Процесс проектирования включает в себя постановку и решение задач в определенной последовательности.

Эти задачи назовем *частными задачами проектирования* (ЧЗП).

Каждая ЧЗП может быть сформулирована как задача определения некоторого набора параметров – *конструктивных параметров*.

К конструктивным параметрам могут относиться

- Технические параметры элементов информационных вычислительных систем (ИВС).
- Параметры, характеризующие состав, структуру или организацию работы системы.

Выбор одного из альтернативных вариантов построения системы может быть также сформулирован как задача определения значения некоторого конструктивного параметра (номер варианта). Так или иначе, в любой ЧЗП речь идет о выборе точки $X = (X_1, X_2, \ldots, X_n)$ из некоторого множества $\chi (X \in \chi)$, которое назовем *пространством конструктивных параметров*. Пространство χ описывается совокупностью ограничений, которым должны удовлетворять компоненты вектора X. Эти ограничения

имеют самую разнообразную «природу возникновения» и различный вид.

Можно выделить ограничения, накладываемые на отдельные компоненты вектора X:

$$X_i^H \le X_i \le X_i^B,\ i \in I_1; \tag{1.1}$$

$$X_i \in \left\{X_{i_1}, X_{i_2}, \ldots, X_{i_k}, \ldots, X_{i_{ki}}\right\},\ i \in I_2; \tag{1.2}$$

$$I_1 \bigcup I_2 = \left\{\overline{1,n}\right\},\ I_1 \bigcap I_2 = \varnothing, \tag{1.3}$$

где X_i^H, X_i^B – соответственно, нижняя и верхняя границы области изменения *i*-го непрерывного конструктивного параметра; X_{i_k} – *k*-е значение *i*-го дискретного конструктивного параметра, $k = \overline{1,K_i}$; K_i – количество возможных значений *i*-го дискретного конструктивного параметра.

Множество значений параметра X_i, описываемое ограничениями (1.1) или (1.2), назовем *областью изменения параметра* χ_i, $X_i \in \chi_i$. Тогда пространство конструктивных параметров представляет собой подмножество декартова произведения множеств χ_i:

$$\chi \subseteq \chi^* = \prod_{i=1}^{n} \chi_i, \tag{1.4}$$

где $X \in \chi$; $X_i \in \chi_i, i = \overline{1,n}$.

Выделение подмножества χ из множества χ^* связано с тем, что на компоненты вектора X кроме (1.1) и (1.2) наложен ряд других ограничений, например

$$\varphi\left(X_1, X_2, \ldots, X_n\right) \ge 0 \tag{1.5}$$

или

$$\psi\left(X_1, X_2, \ldots, X_n\right) = 0. \tag{1.6}$$

Ограничения (1.5) могут быть связаны с лимитами на какие-либо ресурсы (финансовые, трудовые, энергетические и т.п.),

ограничения (1.6) могут выражать собой некоторые законы сохранения (так называемые балансовые ограничения). Наконец, указанные ограничения могут определяться требованиями к конструкции или проекту (*функциональные* или *критериальные ограничения*).

Таким образом, *решить ЧЗП* – это значит определить некоторую точку X в пространстве χ конструктивных параметров. Каждая точка X – это некоторое техническое решение (некоторый альтернативный вариант). Для выбора варианта надо задавать на множестве χ отношение предпочтения, т.е. некоторым образом упорядочить χ. Это можно сделать, в частности, формулируя целевую функцию, или критерий эффективности ЧЗП, на основе совокупности *технических требований* (ТТ), имеющих отношение к данной ЧЗП. Это требования представляют собой ограничения, наложенные на выходные параметры [2] объекта в данной ЧЗП.

Обозначим через $V=(\upsilon_1,\upsilon_2,\ldots,\upsilon_m)$ – вектор выходных параметров данной ЧЗП.

Для задач проектирования АСУ *к выходным параметрам* относят параметры производительности, надежности, достоверности, а также технико-экономические параметры. Выходные параметры следует рассматривать в контексте решаемой ЧЗП. В частности, иерархическая взаимосвязь ЧЗП в общем процессе проектирования сложной системы приводит к тому, что параметры, относящиеся к *конструктивным* в ЧЗП данного уровня проектирования, могут стать *выходными* для ЧЗП следующего (более низкого) уровня.

Заданные соотношения между выходными параметрами и техническими требованиями называют *условиями работоспособности* [2]. Условия работоспособности могут задаваться следующим образом:

$$\upsilon_j \le \widehat{\upsilon_j},\ j \in J_1; \tag{1.7}$$

$$\upsilon_j \ge \widehat{\upsilon_j},\ j \in J_2; \tag{1.8}$$

$$\upsilon_j = \upsilon_j^{\text{ном}} \pm \Delta_j,\ j \in J_3; \tag{1.9}$$

$$J_1 \bigcup J_2 \bigcup J_3 = \overline{1,m}, \; J_\alpha \bigcap J_\beta = \varnothing \quad \text{при } \alpha \neq \beta; \tag{1.10}$$

$$\alpha, \beta \in \{1,2,3\}.$$

Условие вида (1.9) представляет собой совокупность условий вида (1.7) и (1.8):

$$\begin{aligned} &\upsilon_j \le \upsilon_j^{\text{ном}} + \Delta_j = \hat{\upsilon}_j^B, \\ &\upsilon_j \ge \upsilon_j^{\text{ном}} - \Delta_j = \hat{\upsilon}_j^H, \; j \in J_3. \end{aligned} \tag{1.11}$$

Проектировщик пытается добиться выполнения условий работоспособности соответствующим выбором $X \in \chi$. Его задача усложняется тем, что выходные параметры вектора V зависят не только от конструктивных параметров X, находящихся в его распоряжении, но и от некоторого вектора *неуправляемых параметров* Y:

$$V = V(X, Y),$$

где $Y = (y_1, y_2, \ldots, y_r)$, $Y \in \mathcal{Y}$.

Неуправляемые параметры некоторой ЧЗП – параметры среды, в которой работает проектируемая система (например, характеристики внешних потоков информации, абонентов), а также либо такие параметры, которые выбраны в ходе решения предыдущих ЧЗП, или такие, которые будут выбираться в последующих ЧЗП. В практике проектирования сложных систем для любой ЧЗП наиболее типичным является случай, когда одни компоненты вектора Y фиксированы, другие случайны, третьи неопределенны.

Формирование целевой функции для некоторой ЧЗП

Одна из естественных целей деятельности проектировщика, решающего ЧЗП, состоит в том, чтобы выбранное техническое решение наилучшим образом удовлетворяло условиям работоспособности во всем диапазоне изменения внешних параметров при выполнении всех качественных требований ТЗ [2]. Качественные требования ТЗ в формализованном описании ЧЗП, как правило, трансформируются в ограничения, описывающие про-

странство χ, или реализуются на заключительном этапе отбраковки и выбора решения, осуществляемого непосредственно конструктором.

Чтобы выразить математически свойство «наилучшим образом удовлетворять условиям работоспособности», введем функции $z_j\left(\upsilon_j\right)$ в виде

$$z_j\left(\upsilon_j\right)=1-\frac{\upsilon_j^*}{\hat{\upsilon}_j^*},\ j\in\left\{\overline{1,m}\right\}, \tag{1.12}$$

где

$$\upsilon_j^*=\upsilon_{jo}-\upsilon_j\ ; \tag{1.13}$$

$$\hat{\upsilon}_j^*=\upsilon_{jo}-\widehat{\upsilon_j}\ . \tag{1.14}$$

Здесь υ_{jo} – некоторое «рекордное» значение υ_j, которое достигается в допустимой области решений и для которого можно положить $z_j=I$.

Для выходных параметров υ_j, которые следует уменьшать ($j\in J_1$), можно взять

$$\upsilon_{jo}=\min_{X\in\chi}\min_{Y\in Y}\upsilon_j\left(X,Y\right), \tag{1.15}$$

а для параметров υ_j, которые желательно увеличивать $\left(j\in J_2\right)$, можно взять

$$\upsilon_{jo}=\max_{X\in\chi}\max_{Y\in Y}\upsilon_j\left(X,Y\right). \tag{1.16}$$

Для параметров υ_j, которые в результате проектирования должны попасть в допустимые границы $\left(j\in J_3\right)$, естественно взять

$$\upsilon_{jo}=\upsilon_j^{\text{ном}}. \tag{1.17}$$

В качестве значений υ_{jo} можно, не решая экстремальных задач (1.15) и (1.16), взять оценки, полученные либо опросом

экспертов (конструкторов), либо на основании анализа систем-аналогов.

Рассмотрим свойства функций $z_j(\upsilon_j)$, определяемых соотношениями (1.12)–(1.17). Легко видеть, что $z_j(\upsilon_j)$– линейная функция выходного параметра, причем для $\upsilon_j = \upsilon_{jo}$ $z_j = 1$, а для $\upsilon_j = \widehat{\upsilon_j}$ $z_j = 0$. При этом для $j \in J_1 \upsilon_{jo} \langle \hat{\upsilon}_j$, а для $j \in J_2$ $\upsilon_{jo} \rangle \hat{\upsilon}_j$. Отсюда следует, что при выполнении условий (1.7) и (1.8) $z_j(\upsilon_j) > 0$, а когда эти условия не выполняются, то $z_j(\upsilon_j) < 0$.

Аналогично обстоит дело с выполнением ограничений (1.9). Эти свойства позволяют трактовать функцию $z_j(\upsilon_j)$ как запас в выполнении ограничений, описывающих условия работоспособности по *j*-му выходному параметру. Значения функции

$$W^*(X,Y) = \min_{1 \le j \le m} \alpha_j z_j\left(\upsilon_j(X,Y)\right) \tag{1.18}$$

можно рассматривать как запас в выполнении всей совокупности ограничений (1.7)–(1.9), причем цель проектировщика – увеличивать этот запас.

Числовые коэффициенты α_j отражают неодинаковую важность отдельных выходных параметров ($\alpha_j > 0$, меньшие значения α_j соответствуют большей важности выходного параметра υ_j).

На рис. 1.1 изображены функции запаса для выходных параметров разных категорий.

Когда компоненты вектора *Y* фиксированы, целевую функцию для ЧЗП можно сформулировать в виде

$$W(X) = \min_{1 \le j \le m} \alpha_j z_j\left(\upsilon_j\left(X, Y^{\Phi}\right)\right). \tag{1.19}$$

Цель проектировщика, решающего ЧЗП на основе наилучшего удовлетворения условиям работоспособности, – найти вектор X^o, обеспечивающий

$$W\left(X^o\right) = \max_{X \in \chi} W(X). \tag{1.20}$$

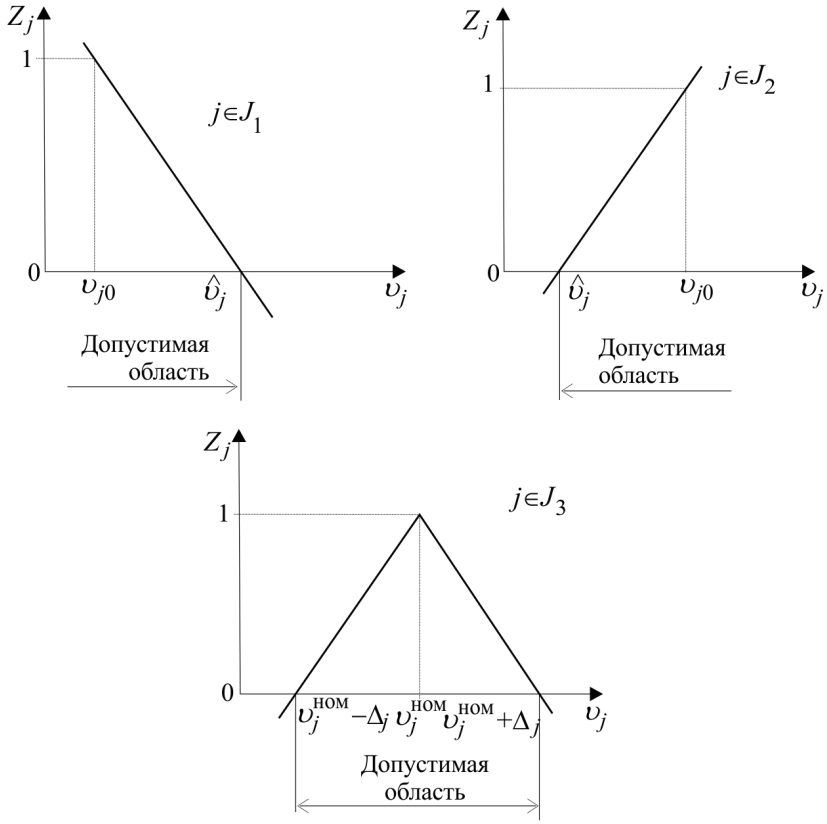


Рис. 1.1. Функции запаса
для выходных параметров разных категорий

Иногда в практике проектирования решаются *задачи оптимизации по какому-то одному выходному параметру* υ_{j1} (например, стоимости) *при выполнении ограничений* ТЗ *по остальным параметрам*. Указанные задачи входят как частный случай в постановку (1.19), (1.20). При этом надо положить $\alpha_{j1} = 1;\ \ \alpha_j = M$ для $j \in \left\{\overline{1,m}\right\}$, $j \neq j_1$, где M – некоторое большое число ($M \gg 1$).

Также в практике проектирования вместо цели *наилучшим образом удовлетворить условиям работоспособности задаются*

более скромной – удовлетворить условиям работоспособности с некоторым запасом γ, что соответствует целевой функции вида

$$W(X)=\begin{cases}1, & \text{если } \min\limits_{1\le j\le m} z_j \ge \gamma,\\ 0, & \text{иначе.}\end{cases} \tag{1.21}$$

В условиях автоматизированного проектирования, когда оценка значений выходных параметров производится посредством математических моделей $V=V(X,Y)$, надо учитывать неизбежный разброс параметров Y (например, исходных данных) и наличие погрешностей моделей.

В связи с этим критерий (1.21) можно использовать при условии, что функции запаса определены с некоторой поправкой на разброс параметров, например, в таком виде:

$$z_j = 1-\frac{\upsilon_j^*}{\hat{\upsilon}_j^*}-\frac{\delta_j}{\left|\hat{\upsilon}_j^*\right|},\ j\in\overline{\{1,m\}}, \tag{1.22}$$

где δ_j – оценка разброса выходного параметра υ_j в результате неточности математической модели и разброса *Y* в области *У*.

Когда компоненты вектора *Y* случайны, вместо $W(X)$ можно брать функцию

$$W(X)=\overline{W}^*(X,Y), \tag{1.23}$$

где $\overline{W}^*(X,Y)$ – усреднение по *Y* функции W^*, определяемой соотношением (1.18).

Возможен другой подход, при котором целевая функция берется в виде

$$W(X)=\min_{1\le j\le m}\alpha_j\overline{z_j}(X), \tag{1.24}$$

где $\overline{z_j}(X)$ – усреднение по *Y* функции z_j.

Вычисление $W(X)$, выполненное в соответствии с (1.23) или (1.24), существенно более трудоемко по сравнению с вычислениями, выполненными по (1.19), поскольку усреднение функ-

ции W^* или z_j по Y необходимо производить для каждого значения вектора X.

При линейной зависимости z_j от Y можно производить усреднение Y, а потом брать $\overline{z_j}(X)=z_j\left(\upsilon_j\left(X,\overline{Y}\right)\right)$, но в общем случае так делать нельзя.

Когда компоненты вектора Y не определены, то, в соответствии с принципом гарантированного результата, в качестве целевой функции $W(X)$ следует брать

$$W(X)=\min_{Y\in\mathcal{Y}}\min_{1\le j\le m}\alpha_j z_j\left(\upsilon_j(X,Y)\right). \qquad (1.25)$$

Цель по-прежнему состоит в поиске значения X^o, обеспечивающего решение задачи (1.20).

Анализ выражений (1.25) и (1.20) показывает, что трудность решения задачи выбора оптимального значения вектора конструктивных параметров резко возрастает по сравнению с детерминированным случаем, поскольку даже для расчета одного значения функции $W(X)$ необходимо сначала решить другую экстремальную задачу – нахождения «наихудшего» сочетания неконтролируемых факторов Y в точке (X, Y) декартова произведения χ и $\mathcal{Y}$. Это вынуждает заменять оптимизационную задачу (1.20)–(1.25) другой задачей: берется некоторое фиксированное значение Y в области $\mathcal{Y}$ и для него решается задача (1.19), (1.22), (1.20). Для найденного оптимального значения X^o осуществляется проверка условий работоспособности (статистический анализ) в области $\mathcal{Y}$. Если результат этой проверки признается удовлетворительным, то решение X^o оставляют, если же результат не признается удовлетворительным, то производят корректировку вектора X^o, причем результаты статистического анализа часто подсказывают направление, в котором следует изменять компоненты вектора X^o.

Приведенная схема принятия решения, конечно, упрощенная. На самом деле неопределенностей в процессе принятия решений в каждой ЧЗП существенно больше. В частности, не затра-

гивался вопрос о непротиворечивости требований ТЗ, неопределенности значений $\hat{\upsilon}_j, \alpha_j, \delta_j$, неопределенности выбора той или иной свертки по Y критерия W^*. Последовательное снижение этих неопределенностей может быть осуществлено непосредственно в процессе анализа значений выходных параметров V в специальным образом выбранных точках пространства $\chi \times Y$ (в режиме диалога конструктора с ЭВМ).

Когда математические модели применяются в оптимизационных процедурах при проектировании сложных систем, возникает проблема большой размерности пространства конструктивных параметров и сложности зависимостей между конструктивными и выходными параметрами. Для решения этой проблемы могут использоваться методы декомпозиции задач проектирования, декомпозиции математических моделей, агрегирования параметров, выделение «существенных» параметров. Здесь также можно использовать оптимизацию в пространстве конструктивных параметров на основе вспомогательных, относительно легко вычисляемых функционалов и т.д.

1.3. Декомпозиция задач проектирования КИС на основе многоуровневого подхода

Рассмотрим упрощенную формализованную постановку задачи проектирования КИС на основе *двухуровневого подхода*, когда на верхнем уровне решается задача общесистемной проработки, а на нижнем – задача выбора параметров подсистем КИС [9].

Пусть вектор конструктивных параметров в задаче общесистемной проработки имеет вид

$$XS = \left(XS_1, XS_2, \ldots, XS_{ns} \right), \; XS \in \chi s,$$

где XS_i – *i*-й общесистемный параметр; *ns* – число общесистемных параметров.

К *общесистемным параметрам* относятся параметры в наибольшей степени влияющие на эффективность КИС в целом и ее свойства как системы обработки информации, а также и некоторые обобщенные (агрегированные) параметры подсистем.

Назовем *матрицей конструктивных параметров подсистем* матрицу $XP = \left(XP_{kj} \right)$, $k \in \overline{1, M}$, $j \in \overline{1, N}$ (здесь M – число подсистем; N – максимальное число конструктивных параметров, описывающих подсистему).

Обозначим $XP_k \in \chi p_k$, где χp_k – пространство конструктивных параметров k-й подсистемы, $XP \in \chi p$, здесь

$$\chi p \subseteq \chi p^* = \prod_{k=1}^{M} \chi p_k .$$

Компоненты вектора XP_k (k-й строки матрицы XP) несут детальную информацию по k-й подсистеме.

Для простоты изложения предположим, что внешние условия, влияющие на функционирование КИС, известны и фиксированы (случай $Y = Y^{\Phi}$). Тогда критерий эффективности функционирования КИС является функцией только конструктивных параметров. Моделью общей задачи проектирования КИС может служить система соотношений

$$W\left(XS, XP \right) \to \max_{\substack{XS \subseteq XS^* \\ XP \subseteq XP^*}} ; \tag{1.26}$$

$$\varphi\left(XS, XP \right) \geq 0; \tag{1.27}$$

$$\psi\left(XS, XP \right) = 0 , \tag{1.28}$$

где φ и ψ – вектор-функции.

Специфика решения задачи (1.26)–(1.28) состоит в том, что в практике проектирования выбор параметров XS, XP осуществляется не одновременно, а последовательно во времени: сначала выбирают параметры XS, а затем разные коллективы проектировщиков занимаются выбором XP. Такой подход представляет собой декомпозицию задачи (1.26)–(1.28), осуществляемую в практике проектирования.

Представим задачу (1.26)–(1.28) в виде двухуровневой системы задач. На верхнем уровне (ВУ) решается задача

$$W_0\left(XS, \widetilde{XP}\right) \to \max_{XS \subseteq XS^*}; \tag{1.29}$$

$$\varphi_0\left(XS, \widetilde{XP}\right) \geq 0; \tag{1.30}$$

$$\psi_0\left(XS, \widetilde{XP}\right) = 0. \tag{1.31}$$

В задаче (1.29)–(1.31) не рассматриваются детальные параметры подсистем XP, что можно представить как принятия некоторых фиксированных значений $XP = \widetilde{XP}$. При этом математические модели для расчета W_0, φ_0, ψ_0 могут быть приближенными, подсистемы представляются в них основными интегральными характеристиками или параметрами-агрегатами (коэффициенты в различных приближенных формулах, зависимостью которых от параметров XP в задаче ВУ пренебрегаем). Решая (1.29)–(1.31) при фиксированных $\widetilde{XP}$, получим $XS = XS^o$ – оптимальное решение задачи ВУ.

На нижнем уровне (НУ) решаются задачи оптимизации подсистем при фиксированных общесистемных параметрах КИС $XS = XS^o$.

Задачи НУ

$$W_k = \left(XS^o, \widetilde{XP}_1, \ldots, \widetilde{XP}_{k-1}, XP_k, \widetilde{XP}_{k+1}, \ldots, \widetilde{XP}_N\right) \to \max_{XP_k \subseteq \chi p_k}; \tag{1.32}$$

$$\varphi_k\left(XS^o, \widetilde{XP}_1, \ldots, \widetilde{XP}_{k-1}, XP_k, \widetilde{XP}_{k+1}, \ldots, \widetilde{XP}_N\right) \geq 0; \tag{1.33}$$

$$\psi_k\left(XS^o, \widetilde{XP}_1, \ldots, \widetilde{XP}_{k-1}, XP_k, \widetilde{XP}_{k+1}, \ldots, \widetilde{XP}_N\right) = 0,\ k \in \left\{\overline{1,N}\right\}, \tag{1.34}$$

где W_k – локальные критерии эффективности задач НУ.

При этом, в зависимости от степени разработанности математических моделей для подсистем, возможно решение некоторых задач НУ на основе математико-вычислительных средств, тогда как для решения других задач возможно или целесообразно лишь поочередное рассмотрение ограниченного числа вариантов, задаваемых проектировщиком. Проектирование КИС, по существу и организационно, – процесс последовательных приближе-

ний, заключающийся в поочередном решении задач ВУ и НУ с нарастающей степенью подробности. Уменьшить число итераций можно при выполнении условий согласованности и непротиворечивости задач НУ по отношению к задаче ВУ.

Непротиворечивость задач (1.29)–(1.31) и (1.32)–(1.34) означает, что оптимизация XP_k улучшает глобальный критерий W_0:

$$\begin{aligned} &W_0\left(XS^o, \widetilde{XP}_1, \widetilde{XP}_2, \ldots, \widetilde{XP}_N\right) \le \\ &\le W_0(XS^o, \widetilde{XP}_1, \ldots, \widetilde{XP}_{k-1}, XP_k^o, \widetilde{XP}_{k+1}, \ldots, \widetilde{XP}_N), \end{aligned} \tag{1.35}$$

где XP_k^o – оптимальное решение задачи (1.32)–(1.34).

Согласованность задач (1.29)–(1.31) и (1.32)–(1.34) означает, что оптимизация XS^o не выводит общесистемные параметры из допустимой области (1.30)–(1.31):

$$\begin{aligned} &\varphi_0(XS^o, \widetilde{XP}_1, \ldots, \widetilde{XP}_{k-1}, XP_k^o, \widetilde{XP}_{k+1}, \ldots, \widetilde{XP}_N) \ge 0; \\ &\psi_0(XS^o, \widetilde{XP}_1, \ldots, \widetilde{XP}_{k-1}, XP_k^o, \widetilde{XP}_{k+1}, \ldots, \widetilde{XP}_N) = 0. \end{aligned} \tag{1.36}$$

При расширении допустимой области (при согласованности задач ВУ и НУ) значение глобального критерия может быть улучшено, если вторично решать задачу ВУ после решения задачи НУ.

Обеспечить выполнение условий непротиворечивости и согласованности задач ВУ и НУ можно за счет локальных критериев W_k. В соответствии с подходом, предлагаемым в [3] для проектирования сложных систем на примере проектирования морских судов, можно в качестве критерия W_0 задачи ВУ использовать взятые со знаком минус *приведенные затраты* – затраты, отнесенные к производительности – по системе в целом, а для задач НУ – приведенные затраты по подсистемам. Это обеспечит выполнение требования непротиворечивости.

Для обеспечения согласованности предлагается использовать два способа.

Способ 1. Включить в число ограничений для каждой задачи НУ *директивные константы*, запрещающие выводить XS за пределы допустимой области.

Способ 2. Ввести в локальные критерии дополнительные слагаемые, учитывающие изменение глобального критерия вследствие изменения границ допустимой области задачи НУ, которое происходит в результате решения задачи НУ.

1.4. Декомпозиция задач проектирования КИС на основе вспомогательных критериев

Рассмотрим схему декомпозиции, в которой используется разделение конструктивных параметров на «существенные» и «несущественные» и воплощается идея выбора существенных параметров на основе некоторых вспомогательных критериев, или «агрегированных характеристик».

В исходной постановке надо выбрать вектор конструктивных параметров $X=(X_1,X_2,\ldots,X_n)$, обеспечив при этом выполнение требований ТЗ на выходные параметры V.

Для простоты изложения будем считать, что внешние условия, выражаемые значениями компонент вектора неуправляемых параметров Y, известны и фиксированы: $Y=Y^{\phi}$, т.е. $V=V(X)$.

Решение задачи выбора вектора X из условия максимизации целевой функции $W(X)$, являющейся некоторой сверткой функций $V(X)$, сложно, поскольку размерность вектора X велика и сложна зависимость $V(X)$. Исходную задачу представим таким образом:

$$W(V(X))\to\max_{X\in\chi}. \tag{1.37}$$

С точки зрения главного конструктора, свойства КИС как системы обработки информации характеризуются набором относительно небольшого числа параметров $F_j, j=\overline{1,p}$. Эти параметры, как правило, являются *агрегатами*, т.е. некоторыми функциями конструктивных параметров КИС:

$$F_j=F_j(X_1,X_2,\ldots,X_n),\ j=\overline{1,p}. \tag{1.38}$$

Поэтому F_j называют *агрегированными характеристиками* (АХ) *системы* [5].

В реальных условиях часто количество АХ обычно не превосходит десятка. АХ в совокупности определяют качество системы. При этом важно, чтобы их расчет был достаточно простым. Это необходимо для указанного расчета с помощью математических моделей, входящих в САПР, в течение нескольких секунд машинного времени. Простота такого расчета возникает вследствие того, что АХ с точностью, необходимой главному конструктору, зависят, как правило, лишь от небольшого числа «существенных» параметров. Поэтому конструктивные параметры можно разбить на две группы:

$$X = \left(X^+, X^- \right),$$

где X^+ – вектор существенных параметров (его размерность относительно невелика (обычно это десятки переменных)); X^- – вектор всех остальных переменных (его размерность – тысячи переменных).

Тогда зависимости $F_j(X)$ имеют вид

$$F_j(X) = F_j\left(X^+, \varepsilon X^- \right),\ j = \overline{1, p}\ , \tag{1.39}$$

где ε – некоторый малый параметр такой, что на уровне общесистемной обработки можно положить

$$F_j\left(X^+, \varepsilon X^- \right) \cong F_j\left(X^+, 0 \right),\ j = \overline{1, p}\ . \tag{1.40}$$

Далее вместо X^+ будем писать X.

> Таким образом, главный конструктор сосредотачивает свое внимание на выборе существенных параметров, оставляя определение остальных параметров на последующие этапы проектирования, в частности, на проектирование подсистем.

Выбор существенных переменных происходит в два этапа. Предположим, что критерий $W\left(V(X)\right)$ (назовем его *глобаль-*

ным критерием) удовлетворяет условию монотонности по F_j, $j=\overline{1,p}$: $\forall X', X'' \in \chi$ из $F_j(X') > F_j(X'')$, $j=\overline{1,p}$ следует $W(V(X')) > W(V(X''))$.

Условие монотонности, в частности, выполнено, если критерий представлен в виде

$$W(V(X)) = \Phi(F_1(X), F_2(X), \ldots, F_p(X)), \tag{1.41}$$

где $\Phi(F)$ определена на множестве значений G вектор-функции $F(X) = (F_1(X), F_2(X), \ldots, F_p(X))$ и монотонна по F, т.е. $\forall F', F'' \in G$ из $F_j' \geq F_j'', j=\overline{1,p}$ следует $\Phi(F') \geq \Phi(F'')$.

Обозначим через $\Pi(G)$ множество эффективных (оптимальных по Парето) векторов F из G, а через $\Pi(X)$ – множество векторов $X \in \chi$, для которых $F(X) \in \Pi(G)$. Вектор $F \in G$ называется эффективным, если $\forall F' \in G$ из $F_j' \geq F_j$, $j=\overline{1,p}$ следует $F' = F$. *Эффективные векторы* – векторы, которые невозможно улучшить одновременно по всем компонентам.

На **этапе 1** декомпозиции отыскиваются векторы $X \in \Pi(X)$ (варианты построения системы, которые невозможно улучшить одновременно по всем АХ), а на этапе 2 задача максимизации функции $W(V(X))$ решается уже не на всем множестве вариантов χ, а на множестве $\Pi(X)$:

$$W(V(X)) \to \max_{X \in \Pi(X)}. \tag{1.42}$$

Этап 1 не зависит от критерия $W(V(X))$ и позволяет, как правило, произвести резкое сокращение числа рассматриваемых вариантов. Его успешное решение на ЭВМ определяется простотой расчета АХ как $F(X)$.

На **этапе 2**, если критерий представлен в виде (1.41), вместо задачи (1.42) решают задачу

$$\Phi(F) \to \max_{F \in \Pi(G)}. \tag{1.43}$$

В задаче (1.43) по сравнению с задачей (1.42) можно сократить объем вычислений, если учесть возможное совпадение значений $F(X)$ для различных $X \in \Pi(X)$.

Декомпозиция задачи (1.37) возможна и без выполнения требований монотонности. На практике, как правило, декомпозиция будет неформальной. Проблема выбора вектора АХ должна решаться с учетом опыта проектирования КИС.

Практическая реализация схемы декомпозиции

Обозначим

$$WW(X,\Lambda) = \max_{1 \le j \le p} \lambda_j \frac{F_j^o - F_j(X)}{F_j^o}, \tag{1.44}$$

где

$$F_j^o = \max_{X \in \chi} F_j(X), \quad j = \overline{1,p}. \tag{1.45}$$

Вспомогательный критерий $WW(X,\Lambda)$ является сверткой АХ $F_j(X)$, $j = \overline{1,p}$. Выражение (1.44) представляет собой меру отклонения совокупности АХ некоторого технического варианта X реализации системы от несуществующего идеального варианта, «рекордного по всем АХ». Эта мера отклонения зависит от векторного параметра $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_p)$, который называется *вектором концепции системы* и учитывает неодинаковую относительную важность отдельных АХ. Будем задавать значения вектора Λ из некоторой области, которая, например, описывается совокупностью ограничений $\lambda_j > 0$, $j = \overline{1,p}$, $\sum_{j=1}^{p} \lambda_j = 1$.

Одновременно будем решать задачу

$$WW(X,\Lambda) \to \min_{X \in \chi}. \tag{1.46}$$

Тогда можно найти $X^o(\Lambda)$ – представителя «концепции Λ» (технический вариант, оптимальный с точки зрения данной концепции).

Задачу (1.46) можно решить в силу простоты расчета АХ и относительно небольшой размерности вектора существенных параметров X. Так как процедура решения задачи (1.46) каждому Λ ставит в соответствие X^o, то глобальный критерий $W\left(V\left(X^o(\Lambda)\right)\right)$ на множестве значений $X^o(\Lambda)$ становится функцией Λ, что позволяет решать задачу

$$W\left(V\left(X^o(\Lambda)\right)\right) \to \max_{\Lambda}. \qquad (1.47)$$

В этой задаче – более сложный критерий по сравнению с задачей (1.46), но меньшая размерность пространства концепций $(p < n)$.

Таким образом, наилучший вариант виден не сразу, а в два этапа.

Этап 1. Формируется пространство векторов концепции $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_p)$, где значения вектора Λ берутся из области, описываемой совокупностью ограничений $\lambda_j > 0, \quad j = \overline{1, p}$, $\sum_{j=1}^{p} \lambda_j = 1.$

Для каждого Λ решается задача (1.46) – отыскивается вариант $X^o(\Lambda)$ (представитель концепции Λ).

Этап 2. Среди всех найденных представителей концепции выбирается вариант, реализующий решение задачи (1.47) (в сущности, отыскивается наилучшая концепция).

На самом деле последовательность выполнения этапов 1 и 2 не обязательно должна быть линейной (этап 2 после выполнения всего этапа 1).

Решение задачи (1.46) для каждого значения Λ может быть «вложено» в алгоритм решения задачи (1.47), т.е. этап 2 – это алгоритм верхнего уровня, обращающийся к алгоритму решения задачи (1.46) этапа 1 в соответствии со своей логикой.

Например, поиск оптимального решения задачи (1.47) может осуществляться в режиме диалога с ЭВМ, на основе зондирования пространства Λ конечным числом точек и анализ этих точек

конструктором-исследователем. На основе анализа производится отбор одной или нескольких перспективных точек и последующее зондирование более узких областей пространства Λ вокруг этих точек, новый анализ и т.д.

ГЛАВА 2

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ СТОХАСТИЧЕСКИХ МОДЕЛЕЙ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

2.1. Алгоритмическая модель процесса функционирования КИС

В основе методов моделирования, как аналитических, так имитационных, лежит понятие процесса *функционирования дискретных систем*. Дадим определение процесса в том виде, который будет использоваться в дальнейшем изложении. В основу положим метод описания структуры процессов функционирования дискретных систем.

Под *функционированием дискретной системы* понимается процесс изменения ее состояния во времени. Система имеет высокую размерность, разделяется на множество объектов, различным способом связанных между собой, руководствуется сложными алгоритмами, описывающими переход из одного ее состояния в другое.

Всю совокупность параметров дискретной системы, определяющих процесс функционирования или участвующих в нем, назовем *параметрическим множеством системы*

$$Q = \{q_i\}_{i=1}^{n},$$

где q_i – некоторый параметр.

Каждый параметр q_i принимает множество значений, обозначаемое в дальнейшем как $\sigma(q_i)$. Определим *пространство состояний системы S* или S_Q как декартово произведение:

$$S_Q = \prod_{q_i \in Q} \sigma(q_i),$$

а *процесс* как

$$Z =< S,\ T,\ F >,$$

где S – пространство состояний системы, определенное ранее; T – упорядоченное по возрастанию множество моментов времени изменения состояний системы; F – график процесса, определяемый как отображение $T \to S$, причем это отображение должно быть функциональным (однозначным).

Определим объект O_l как составную часть системы: $O_l \subset Q$. Если задан процесс Z_Q в системе, то процесс в объекте O_l может быть определен как проекция процесса в системе Z_Q на подпространство S_{O_l}:

$$Z_{O_l} = \underset{S_{O_l}}{\Pi p} Z_Q .$$

Поскольку в общем случае задание процесса в виде единого оператора затруднительно либо невозможно, то предлагается задавать оператор H в виде некоторой алгоритмической структуры. Для этого каждой i-й точке дискретного процесса (момент времени изменения состояния t_i) поставим в соответствие некоторый оператор h_i^c, описывающий вычисление только одной i-й точки процесса Z. В силу этого условия будем называть этот оператор *элементарным оператором*.

Таким образом, если график процесса содержит n точек, то необходимо задать линейную последовательность элементарных операторов $h_1^c, h_2^c ..., h_i^c, ..., h_n^c$, называемую *треком*.

Введем новый элемент модели – *инициатор*, полагая, что это объект, обладающий следующими фундаментальными свойствами:

независимость – может существовать самостоятельно без операторов;

динамичность – имеет возможность перемещаться от оператора к оператору; попадание инициатора на оператор будем называть *сцеплением инициатора с элементарным оператором*;

инициативность – в момент сцепления инициатора с оператором происходит выполнение (инициирование) элементарного оператора, что соответствует вычислению нового состояния процесса. В дальнейшем будем полагать, что выполнение элементарного оператора происходит мгновенно.

Предлагаемая модель описания процесса функционирования КИС предполагает, что моменты сцепления инициатора с элементарными операторами определяют сами элементарные операторы. С этой целью введем в состав элементарного оператора $h_i^{\text{с}}$ оператор $h_i^{\text{у}}$, определяющий условие, при выполнении которого инициатор покидает оператор $h_i^{\text{с}}$ и сцепляется со следующим оператором $h_{i+1}^{\text{с}}$.

Варианты задания такого условия:

а) указание момента времени сцепления инициатора с оператором $h_{i+1}^{\text{с}}$;

б) определение логического условия, при выполнении которого инициатор сцепляется с оператором $h_{i+1}^{\text{с}}$;

в) комбинированная форма, включающая варианты а и б.

Таким образом, можно определить:

h_i^t – оператор временного условия продвижения инициатора (соответствует варианту а);

$h_i^{\text{л}}$ – оператор логического условия продвижения инициатора (соответствует варианту б);

$h_i^{t,\text{л}}$ – оператор комбинированного условия продвижения инициатора (соответствует варианту в).

Расширим понятие элементарного оператора, добавив к нему помимо оператора $h_i^{\text{с}}$ оператор $h_i^{\text{у}}$.

В результате можно определить понятие *алгоритмической модели процесса* (в дальнейшем АМП) в виде «тройки»:

$$\text{АМП} = \left\langle \left\{h_i\right\}_{i=1}^{n}, \beta, I \right\rangle,$$

где $\left\{h_i\right\}_{i=1}^{n}$ – множество элементарных операторов; β – линейный порядок на $\left\{h_i\right\}_{i=1}^{n}$; I – инициатор.

> АМП содержит *один и только один инициатор*, который соответствует одному конкретному процессу. В этом смысле инициатор является представителем процесса, при его потери либо отсутствии развитие процесса прекращается.

Структура

Пусть задан некоторый трек TR. В реальных приложениях трек содержит достаточно много элементарных операторов, выполняющих одни и те же операции над аргументами. *Операторы эквивалентны*, если при одних и тех же значениях аргументов они вычисляют одинаковые результаты. Это свойство трека позволяет задать отношение эквивалентности на множестве $\left\{h_i\right\}_{i=1}^{n}$ элементарных операторов трека TR.

Назовем *структурой* свертку трека TR по отношению эквивалентности элементарных операторов.

> Таким образом, при заданном треке и отношении эквивалентности на нем всегда может быть построена структура.

Обратная операция – построение трека при заданной структуре – неопределенна. С тем, чтобы операцию построения трека из структуры сделать однозначной, введем еще один тип элементарного оператора – *навигационный оператор*, определяемый так же как и элементарный оператор, однако в результате его выполнения определяется тот элементарный оператор в структуре, который должен выполняться следующим. Выполнение навигационного оператора инициируется *инициатором*.

> Использование структуры по сравнению с треком позволяет значительно снизить размерность описания процесса. Однако необходимо иметь

в виду, что процесс определен только в случае задания трека, поэтому структура есть лишь способ более компактного описания трека, генерация самого трека остается необходимой операцией.

Если на треке элементарных операторов указать используемые этими операторами параметры и их взаимосвязи, то получим *операторно-параметрическую схему*. Такие схемы дают наглядную картину взаимодействия параметров в ходе реализации процесса.

Подобные процессы

Рассмотрим случай задания двух близких по описанию процессов Z_1 и Z_2. В обоих треках используются одинаковые элементарные операторы, но они взаимодействуют с разными параметрами как входными, так и выходными. Желательно найти способ объединения описаний таких процессов.

Для решения поставленной задачи дополним определение инициатора, добавив к его фундаментальным свойствам возможность включать в себя параметры. Таким образом, инициатор, наряду с фундаментальными свойствами, приобретает некоторое «тело» в виде совокупности параметров. Назовем эту совокупность параметров *локальной средой процесса* и в дальнейшем будем считать, что «тело» инициатора представляет собой *ссылку на локальную среду*. Тогда можно предложить схему свертки описаний двух процессов в одно общее описание.

Таким образом, удается снизить размерность описания множества процессов, введя отношение подобия процессов.

Для описания совокупности подобных процессов достаточно иметь одно объединенное описание трека или структуры и множество одинаково структурированных локальных сред, привязанных к инициаторам.

Ресурсы, конфликты на ресурсах

Процессы Z_i в системе Q развиваются параллельно. Это значит, что они изменяют значения параметров системы в течение одного и того же интервала времени. Достаточно типичны

ситуации когда по логике функционирования системы накладываются ограничения на изменение некоторых параметров несколькими процессами одновременно в течение заданного либо обусловленного интервала времени. Это возможно лишь для пересекающихся объектов. Ограничения развития процесса, накладываемые другими процессами, назовем *конфликтными ситуациями*, или *конфликтами.*

Общую область параметров для пересекающихся процессов назовем *ресурсом*.

> Конфликт возможен только при наличии ресурса и при условии *одновременного* обращения процессов к нему. В основе способов разрешения конфликтов лежит утверждение об обязательном разделении доступа процессов к ресурсу во времени.

Блок

Совокупность ряда операторов и связанных с ним параметров назовем *блоком*.

Агрегат

На практике часто возникает необходимость описывать процесс функционирования некоторой машины по преобразованию значений параметров в соответствии с заданным циклическим алгоритмом. Такой процесс можно описать с помощью блока общего вида, в котором существует один инициатор, а трек циклически замкнут. Блок, в котором развивается один единственный циклический процесс, будем называть *агрегатом* или *А-блоком*.

> Таким образом, агрегат содержит единственный инициатор и трек элементарных операторов, замкнутый внутри блока. Обмен между агрегатом и другими блоками возможен исключительно посредством параметров.

Процессор

Блок, предназначенный для генерации процессов, инициаторы которых являются внешними по отношению к блоку, называется *процессором* или *П-блоком*. Инициаторы, поступившие извне, сцепляются с блоком, порождая процессы, и затем покидают его. Поскольку процессор генерирует множество одновременно

протекающих процессов, в нем используются исключительно объединенные элементарные операторы, а инициаторы должны содержать локальные среды.

> Таким образом, процессор представляет собой описание произвольной структуры, содержащей объединенные операторы. Процессы порождаются в этом блоке лишь при поступлении в него извне инициаторов, содержащих локальные среды, т.е. процессор порождает параллельно протекающие во времени подобные процессы.

Контроллер

Блок типа агрегат. Как было показано выше, не имеет возможности взаимодействовать с внешними инициаторами. С тем, чтобы снять это ограничение, введем над инициаторами операции пассивизации и активизации. *Операция пассивизации* переводит инициатор в класс обычных параметров. *Операция активизации*, обычный параметр переводит в класс инициаторов.

Если агрегат содержит операторы, выполняющие указанные операции, то такой агрегат назовем *контроллером* или К-блоком.

Контроллер представляет собой агрегат, выполняющий операции над внешними инициаторами в соответствии с собственным алгоритмом функционирования. *Операции над инициаторами* суть операции над процессами. Таким образом, контроллер исполняет роль управляющего звена в некоторой системе процессов.

Классификация схем описания процессов в системах

Агрегативная схема

Агрегативная схема описания функционирования системы предполагает использование только А-блоков. Как показано выше, в такой схеме взаимодействие может осуществляться лишь через параметры. Пример агрегативных моделей рассмотрен в [21], где показано, что каждый А-блок в такой модели может представляться некоторым конечным автоматом. Если функционирование системы может быть описано совокупностью конечных автоматов, взаимодействующих между собой через множество входных и выходных параметров, то применение агрегативных схем представляется наиболее рациональным.

Процессная схема

Процессная схема описания функционирования системы предполагает использование только П-блоков. Взаимодействие в процессных схемах осуществляется как через параметры, так и через локальные среды процессов. Процессный подход наиболее эффективен, когда имеем дело с множеством явно выраженных локальных процессов, например при описании биологических, экономических, социальных и т.п. систем. Процессный подход реализован в языках Simula, GPSS и др.

Потоковая схема

Если в блочной схеме общего вида ограничиваться использованием А-блоков с простыми алгоритмами, а саму схему изобразить в виде сети А-блоков, где линии связи соответствуют движению инициаторов, то получим потоковую схему. Такая схема отражает движение потоков инициаторов между блоками.

Сети массового обслуживания

Сети массового обслуживания являются особым видом потоковых схем, где все процессы пересечены на ресурсах. Разрешение конфликтных ситуаций в этих схемах выполняется с помощью контроллеров, которые называются *системами массового обслуживания* (СМО) – объединение К-блока и одного или нескольких А-блоков (К-блок реализует дисциплину обслуживания требований, а А-блоки – процесс изменения параметров ресурса).

> Таким образом, формализация функционирования систем в виде сети массового обслуживания возникает в том случае, если в схемах общего вида отсутствуют независимые процессоры, а все процессы пересечены на ресурсах. Поток требований на каждый контроллер есть не что иное, как поток инициаторов процессов, а каждое отдельное требование, будучи инициатором, определяет процесс.

Изложенный способ описания процессов, опирающийся на алгоритмическую модель, позволяет провести анализ особенностей такого описания, определить способы сокраще-

ния размерности описаний как по длине треков, так и по разнообразию типов процессов.

Показано, что алгоритмическая модель позволяет объединить такие разные подходы, как агрегативные и процессные; каким образом использование алгоритмической модели позволяет сформировать потоковые схемы, в частности, использование сетей массового обслуживания.

Опираясь на алгоритмическую модель и ее свойства, могут быть предложены псевдоязыки описания структуры процессов и их взаимодействия. Все это совместно позволяет обосновать правила построения квазипараллельного процесса, лежащего в основе имитационных методов моделирования.

2.2. Параметры производительности КИС

Оценка производительности КИС и ее составных частей необходима для целенаправленного повышения эффективности и качества процесса проектирования на этапах разработки системы и ее эксплуатации. При этом производительность оценивается с целью обоснования и определения наилучших вариантов построения, усовершенствования и оперативного управления функционированием. Деятельность по оценке производительности и целенаправленному ее повышению называется *исследованием оценки производительности*.

В зависимости от специфики задач, решаемых сложной КИС, показатели производительности системы могут отличаться. *Показателями производительности системы* являются параметры (индексы) производительности, характеризующие всю систему или какие-либо ее составные части.

Параметры производительности КИС

- Производительность системы – количество унифицированных проектов, выполняемых в плановую единицу времени.
- Цикл проектирования.
- Характеристики производительности узкого места процесса проектирования.
- Процессорное время, необходимое для выполнения всего комплекса работ по проектированию.

- Время работы периферийных устройств.
- Загрузка элементов системы и т.п.

Параметры производительности КИС в значительной мере определяются составом и организацией совокупности технических средств и общесистемного программного обеспечения, называемой *программно-техническим комплексом*.

Программно-технический комплекс должен обеспечивать

- Производительность, достаточную для решения всех проектных задач.
- Возможность оперативного взаимодействия пользователей с ресурсами системы в процессе решения задач.
- Приемлемое для пользователя время реакции системы на его запросы.
- Высокую надежность функционирования.
- Открытость комплекса для реконфигурации и дальнейшего развития.

В связи с этим необходимы методы и средства, позволяющие оценивать параметры, на основании которых можно судить о соответствии КИС поставленным выше требованиям. Такая оценка должна давать ответы на вопросы качественного характера (что, на что и как влияет), иметь достаточную разрешающую способность, а также оперативность и вычислительную эффективность, позволяющие выполнять исследование в приемлемые сроки.

Рассмотрим некоторые ситуации, в которых может оказаться полезным использование математических моделей.

Пример 2.1. Провести обоснование и выбор варианта объединения нескольких двухуровневых КИС, установленных в различных подразделениях проектной организации, в единую локальную сеть. Предлагается несколько альтернативных вариантов организации.

Для выбора конкретного варианта необходимо провести исследование зависимости пропускной способности сети, измеряемой количеством данных, которые сеть может передать от одного подключенного к ней устройства к другому в единицу времени, от нагрузки на сеть.

Необходимо оценить вероятности безошибочной передачи данных через сеть, времени реакции КИС на запрос с учетом передачи данных через сеть для каждого из альтернативных вариантов архитектуры сетевой КИС.

Пример 2.2. Исследовать возможности модернизации эксплуатируемой КИС без наращивания ее вычислительных мощностей.

Подразумевается исследование способов повышения характеристик производительности, оперативности и надежности КИС за счет внутренних резервов: изменения распределения задач (функциональных пакетов) системы по машинам в многомашинной конфигурации, по разделам оперативной памяти внутри ЭВМ, за счет изменения приоритетов, размещения информации на внешних носителях (магнитных дисках), изменения варианта подключения накопителей на магнитных дисках (НМД) и устройств управления внешними устройствами (УВУ) к каналам (процессорам ввода-вывода) и т.п.

Пример 2.3. Исследовать выполнение требований технического задания по организации работы Единого центра эксплуатации (ЕЦЭ) телекоммуникационной и информационно-технологической инфраструктуры, оперативного мониторинга состояния, а также обеспечения информационной безопасности некоторой специализированной интегрированной системы обработки данных.

ЕЦЭ – сложная человеко-машинная АСУ, включающая в себя вычислительные средства, систему передачи данных, математическое и информационное обеспечение, периферийное оборудование и иерархически организованные группы персонала, обеспечивающие, наряду с программно-техническими средствами, надлежащее выполнение задач, возложенных на систему.

Основные структурные компоненты ЕЦЭ

1. Центр управления мониторинга состояния и эксплуатации подсистемы обеспечения информационной безопасности.

2. Центр управления мониторинга состояния и эксплуатации системы в целом.

3. Диспетчерский центр эксплуатации системы.

4. Центр управления мониторинга состояния и эксплуатации прикладными сервисами системы.

5. Центр обучения, поддержки внедрения, проведения испытаний и хранения документации.

6. Центр управления мониторинга состояния и эксплуатации центра обработки данных системы.

Анализ требований ТЗ к указанным выше центрам и их структуры показывает, что для оценки параметров, входящих в ограничения по ТЗ, в результате математического моделирования, необходима разработка стохастических моделей теории массового обслуживания (ТМО) и теории надежности.

Стохастические модели – многоресурсные системы массового обслуживания (СМО) достаточно сложной структуры, допускающие, тем не менее, декомпозиционные методы расчета параметров на основе агрегированных представлений отдельных подсистем. В последующих разделах рассмотрим эти методы более подробно.

Все вопросы типа «что будет, если...» или «как сделать, чтобы...» полезно прорабатывать до принятия тех или иных решений, связанных с реконструкцией, на основе предварительного анализа, осуществляемого с помощью математических моделей.

Параметры производительности и оперативности КИС

- Время на решение заданного набора задач (для систем, работающих в режиме пакетной обработки); время реакции системы (для систем, работающих в режиме «запрос/ответ» или диалога, в системах РМВ).
- Абсолютная и относительная пропускная способность.
- Длины очередей к совместно используемым ресурсам.
- Аналогичные параметры отдельных составных частей системы (статистические характеристики некоторых случайных величин: моменты функций распределения (ФР) этих случайных величин – математическое ожидание, дисперсия, иногда моменты более высокого порядка и др.

Далее, в основном, будут рассматриваться методы оценки и расчета таких параметров как время реакции и пропускная способность системы.

Время реакции системы на запрос – среднее время от момента нажатия пользователем клавиши «Ввод» на клавиатуре персонального компьютера до момента появления на экране первого символа ответа.

Пропускная способность системы – среднее число задач, которые система способна решить в единицу времени.

2.3. Компоненты формализованной схемы стохастической модели КИС

Проанализируем основные составляющие, из которых складывается время реакции системы на запрос пользователя.

Во-первых, таких составляющих довольно много и они связаны с обработкой запроса разнообразными ресурсами вычислительной системы.

Во-вторых, наряду с величинами, определяемыми только техническими характеристиками устройств (быстродействие процессоров, скорость записи-чтения информации с диска) и характеристиками инициированной данным запросом задачи, время реакции системы включает задержки, связанные с ожиданием освобождения ресурсов, занятых в это время обработкой запросов других пользователей. Значение этих задержек зависит от количества пользователей в системе, порядка просмотра очередей и других факторов, связанных с взаимодействием запросов и ресурсов.

Таким образом, *при анализе времени задержки возникают две основные* ***проблемы***.

1. Правильно представить последовательность прохождения запроса в системе, не упустив при этом никакие важные компоненты и не увязнув в несущественных деталях.

2. Оценить задержки, связанные с конкуренцией запросов к ресурсам.

Эти проблемы решаются в процессе построения и исследования математической модели.

Для оценки указанных параметров надо учитывать, что КИС имеет в своем составе некоторую совокупность программных и аппаратных ресурсов. При этом работа указанных ресурсов по обслуживанию запросов совмещается во времени, что уменьшает простои ресурсов и повышает производительность, но при этом к ресурсам образуются очереди, а это увеличивает время решения каждой отдельно взятой задачи.

> Для определения временных характеристик обработки запросов в системе надо уметь оценивать отрезки времени занятости ресурсов и отрезки времени ожидания освобождения каждого ресурса.

Математическая модель может появиться как следствие четкого формального описания процессов обработки информации в системе с необходимой степенью детализации, т.е. в результате формализации.

Этапы выполнения формализации

- *Содержательное описание* – включает в себя сведения о выходных и конструктивных параметрах системы, ее структуре, особенностях работы каждого элемента (ресурса), характере взаимодействия между ресурсами, а также постановку прикладной задачи (анализа, исследования, проектирования), определяющей цели моделирования исследуемой системы, а также исходные данные, необходимые для исследования.
- *Формализованная схема* – более или менее сложная система массового обслуживания. Из теории (ТМО) известно большое число типов таких систем, причем их классификация может осуществляться по разным признакам, например по особенностям структуры, потоков заявок, параметров обслуживания заявок отдельными ресурсами, дисциплинами просмотра и обслуживания очередей и т.д.
- *Математическая модель технического объекта* – система математических объектов (чисел, переменных, матриц, множеств и т.п.) и отношений между ними, отражающая некоторые свойства технического объекта (**определение И.П. Норенкова**) [6].

Количественно свойства технического объекта выражаются через параметры, среди которых выделяют выходные (характеризующие свойства проектируемой или исследуемой системы), а также внутренние и внешние параметры, от которых зависят выходные параметры.

В ряде случаев удобно подразделять параметры, от которых зависят выходные параметры, на конструктивные (контролируемые) и неуправляемые (неконтролируемые).

К *конструктивным параметрам* относятся параметры, подлежащие выбору, изменению в результате проводимого исследования.

Неуправляемые параметры – параметры, от которых зависят выходные параметры, но которые нельзя произвольно выбирать или изменять.

Любое разделение параметров или на выходные, внутренние и внешние, или выходные, конструктивные и неуправляемые в значительной мере условно и меняется при изменении задач исследования или проектирования (такое изменение всегда имеет место в реальных условиях). Этими терминами будем пользоваться для пояснения принципов построения моделей и требований, предъявляемых к ним.

В дальнейшем будут рассматриваться модели, относящиеся к функциональным аналитическим, или алгоритмическим моделям, которые представляют собой либо явные выражения выходных параметров как функций внутренних и внешних параметров, либо алгоритмы их получения.

Далее рассматриваются способы получения соотношений

$$V = V(U, Z) \quad \text{или} \quad V = V(X, Y), \tag{2.1}$$

где $V = (V_1, V_2, \ldots, V_m)$ – вектор выходных параметров; $X = (X_1, X_2, \ldots, X_n)$ – вектор конструктивных параметров; $Y = (Y_1, Y_2, \ldots, Y_r)$ – вектор неуправляемых параметров; U, Z – соответственно векторы внутренних и внешних параметров.

В **примере 2.1** конструктивные параметры – параметры, описывающие альтернативные варианты организации сетевой КИС, выходные параметры – это пропускная способность сети, время реакции КИС с учетом передачи данных через сеть и т.д.

В **примере 2.2** конструктивные параметры – параметры, описывающие распределение задач КИС по ЭВМ, задач внутри ЭВМ по разделам основной памяти (ОП), приоритетов, размещения информации на внешних носителях и т.д.

Базисные и интерфейсные модели

Время реакции системы на запрос включает в себя задержки в очередях, связанные со случайным характером поступления запросов в систему, случайными маршрутами прохождения запросов внутри множества ресурсов системы, случайными временами их обработки отдельными ресурсами.

Математический аппарат, позволяющий строить модели, связанные с процессами образования очередей, разрабатывается в теории массового обслуживания. Модели, изучаемые в рамках этой теории, называются *системами массового обслуживания* (СМО). В результате четкого формального описания процессов обработки запросов ресурсами системы, проводимого с необходимой степенью детализации, получают СМО того или иного класса.

Методы теории массового обслуживания позволяют вывести соотношения, связывающие значения выходных параметров (например, времени реакции или пропускной способности) с некоторыми параметрами, обобщенно описывающими элементы СМО или ее структуру (моментами функций распределения длительности обслуживания, интенсивностями входных потоков заявок, вероятностями перехода между узлами стохастической сети и т.д.). Такие параметры называют *обобщенными параметрами модели*. Получаемые соотношения называются *базисными моделями* и они имеют вид

$$V = V(\Lambda),$$

где Λ – вектор обобщенных параметров [18].

Для получения математической модели, т.е. зависимостей вида (2.1), необходимо иметь соотношения, связывающие обобщенные параметры с реальными техническими параметрами (см. справочники, результаты измерений на действующих системах, анализ действий пользователя за экраном дисплея, анализ структуры программ и т.п.). Эти соотношения называются *интерфейсными моделями* и имеют вид

$$\Lambda = \Lambda(X, Y).$$

Разделение математической модели на базисную и интерфейсную позволяет использовать одни и те же базисные модели

для разных задач исследования и проектирования, осуществляя настройку на соответствующую задачу посредством изменения интерфейсных моделей. Она позволяет строить и программно реализовывать некоторые обобщенные модели, описывающие достаточно широкие классы формализованных схем.

В отдельных случаях представление математической модели в виде совокупности базисной и интерфейсной моделей может служить основой декомпозиции и снижения размерности задач многовариантного анализа или оптимизационных задач выбора параметров. С применением такого подхода строятся математические модели функциональной надежности, определяющие параметры производительности КИС с учетом ненадежности компонентов системы.

> Примеры построения математических моделей КИС, как совокупности базисных и интерфейсных моделей, приведены в [9]. Рассматриваемая в 6.3 модель процесса проектирования, в сущности, является интерфейсной моделью, позволяющей получить параметры модели САПР в целом.

Поскольку выходные параметры типа характеристик производительности или времени реакции системы или ее составных частей, как было сказано выше, зависят от многих случайных факторов, их количественная оценка производится на основе методологии теории массового обслуживания.

В соответствии с этой теорией ***в формализованном представлении исследуемой системы всегда присутствуют следующие компоненты***.

- Потоки заявок на использование ресурсов системы.
- Обслуживающие аппараты, отображающие программные и/или аппаратные ресурсы (иногда встречаются так называемые «составные ресурсы»), с помощью которых обрабатываются заявки.
- Очереди к обслуживающим аппаратам.

Необходимо уметь параметризовать каждый из этих компонентов.

2.4. Потоки событий в схемах стохастических моделей КИС

Кроме термина «*потоки заявок*» на использование ресурсов, при описании процессов, протекающих в системах обслуживания с дискретными состояниями и непрерывным временем, часто используется термин «*потоки событий*».

Поток событий – последовательность однородных событий, следующих одно за другим в какие-то случайные моменты времени.

Поток событий – последовательность моментов: 1) окончание обслуживания в каком-то ресурсе системы (обслуживающем аппарате); 2) выход из строя какого-то компонента системы; 3) завершение ремонта или профилактического обслуживания какого-то ресурса и т.д.

Важно то, что математическое описание потоков заявок и потоков событий одинаково. Для описания используется либо неотрицательная целочисленная случайная величина $\eta(T, T+t)$, зависящая от непрерывного параметра t (времени) и называемая *количеством событий в фиксированном интервале* $(T, T+t)$, либо неотрицательная случайная величина τ, называемая *длительностью интервала времени между соседними событиями*. Эти две случайные величины взаимосвязаны и, в зависимости от конкретной задачи, удобно пользоваться либо той, либо другой из них.

При рассмотрении процессов, протекающих в системе с дискретными состояниями и непрерывным временем, часто бывает удобно представлять себе процесс так, как будто переходы системы из состояния в состояние происходят под действием каких-то потоков событий (поток вызовов, поток неисправностей, поток заявок на обслуживание, поток моментов окончания обслуживания на какой-то фазе и т.д.). Поэтому необходимо рассмотреть подробнее потоки событий и их свойства.

Потоки событий и их свойства

Будем изображать поток событий последовательностью точек на оси времени. Положение каждой точки на оси абсцисс случайно.

Регулярный поток событий – события следуют одно за другим через строго определенные промежутки времени. Такой поток сравнительно редко встречается на практике, но представляет определенный интерес как предельный случай.

Случайный поток событий – момент наступления событий и промежутки времени между ними случайны. Такой поток событий встречается на практике чаще.

Рассмотрим отдельные классы потоков событий, наиболее часто используемых в анализе систем массового обслуживания.

Простейший поток. Введем ряд определений. *Стационарный поток событий* – вероятность попадания того или иного числа событий на участок времени $(T, T+t)$ длиной t зависит только от длины участка t и не зависит от того, где именно на оси времени расположен этот участок.

Это выражается соотношением

$$P_k(T, T+t) = \Pr\{\eta(T, T+t) = k\} = f(k, t). \qquad (2.2)$$

Ординарный поток событий – вероятность попадания на элементарный участок двух или более событий пренебрежимо мала по сравнению с вероятностью попадания одного события или не попадания ни одного.

Это выражается соотношением

$$P_{>1}(dt) = \Pr\{\eta(dt) > 1\} = o(dt) \qquad (2.3)$$

Или, иными словами,

$$\lim_{dt \to 0} \frac{P_{>1}(dt)}{dt} = 0.$$

(В связи с тем, что $P_0(dt) + P_1(dt) + P_{>1}(dt) = 1$)

Поток событий без последействия – для любых непересекающихся участков времени число событий, попадающих на один из них, не зависит от того, сколько событий попало на другой (или другие, если рассматривается больше двух участков).

Это выражается соотношением

$$\Pr\{\eta(T, T+t) = k\} = \Pr\{\eta(T, T+t) = k \mid \eta(T - t_1, T) = n\}$$

для $\forall T, t, t_1, k, n$. (2.4)

Свойства потоков событий. *Стационарность потока событий* означает его однородность по времени, т.е. вероятностные характеристики такого потока не должны меняться в зависимости от времени.

Интенсивность потока событий – среднее число событий в единицу времени – для стационарного потока должна оставаться постоянной. Это, разумеется, не значит, что фактическое число событий, появляющихся в единицу времени, постоянно – поток может иметь местные сгущения и разрежения.

> Для стационарного потока эти сгущения и разрежения не носят закономерный характер, а среднее число событий, попадающих на единичный участок времени, остается постоянным для всего рассматриваемого периода.

На практике часто встречаются потоки событий, которые (по крайней мере, на ограниченном участке времени) могут рассматриваться как стационарные. Например, поток заявок, поступающих в какую-то электронную справочную систему (например, справочную систему аэропорта), на интервале от 12 до 13 ч может считаться стационарным. Тот же поток в течение целых суток уже не будет стационарным (ночью интенсивность потока вызовов гораздо меньше, чем днем). Также обстоит дело и с большинством *физических процессов*, которые называют *стационарными*. В действительности они стационарны только на ограниченном участке времени, а распространение этого участка до бесконечности – лишь удобный прием, применяемый в целях упрощения исследования.

Отсутствие последействия в потоке означает, что события, образующие поток, появляются в последовательные моменты времени независимо друг от друга. Например, поток запросов в электронную справочную систему от независимых пользователей, можно считать потоком без последействия, потому что причины, обусловившие поступления запроса от отдельного пользователя именно в данный момент, а не в другой, как правило, не связаны с аналогичными причинами для других пользователей. Если такая зависимость появляется, условие отсутствия последействия оказывается нарушенным.

Ординарность потока означает, что события в потоке приходят поодиночке, а не группами.

Как можно рассчитывать характеристики неординарных потоков (например, при групповом поступлении в случайные моменты времени в систему заявок с фиксированным или случайным числом заявок в группе) будет рассмотрено ниже.

Рассмотрим поток событий, обладающий всеми тремя свойствами: стационарный, без последействия, ординарный. Такой поток называется *простейшим* (*стационарным пуассоновским*) *потоком*. Название «простейший» связано с тем, что математическое описание событий, связанных с простейшими потоками, оказывается наиболее простым.

> Самый простой, на первый взгляд, регулярный поток со строго постоянными интервалами между событиями отнюдь не является «простейшим» в вышеназванном смысле слова: он обладает ярко выраженным последействием, так как моменты появления событий связаны между собой жесткой функциональной зависимостью. Именно из-за этого последействия анализ процессов, связанных с регулярными потоками, оказывается, как правило, труднее по сравнению с простейшими потоками.

Простейший поток выполняет среди других потоков особую роль. Оказывается, при суперпозиции (взаимном наложении) достаточно большого числа потоков, обладающих последействием (но стационарных и ординарных), образуется суммарный поток, который можно аппроксимировать простейшим потоком и тем точнее, чем большее число потоков объединяется.

Если поток событий не имеет последействия, ординарен, но не стационарен, он называется *нестационарным пуассоновским потоком*. В таком потоке интенсивность λ (среднее число событий в единицу времени) зависит от времени ($\lambda = \lambda(t)$), тогда как для простейшего потока $\lambda = \text{const}$.

Простейший поток событий называют *пуассоновским потоком*, так как он тесно связан с известным распределением Пуассона.

Можно доказать, что число событий потока, обладающего свойствами стационарности, ординарности и отсутствия после-

действия, попадающих на любой участок длины $t(\eta(t))$, распределено по закону Пуассона:

$$P_k = \Pr\{\eta(T,T+t)=k\} = \Pr\{\eta(t)=k\} = \frac{(\lambda t)^k}{k!}e^{-\lambda t}$$

для $k = 0,1,2,\ldots$. (2.5)

Можно показать и обратное – если число событий потока, попадающих на любой участок длины $t(\eta(t))$, распределено по закону Пуассона, то поток обладает свойствами стационарности, ординарности и отсутствия последействия.

Действительно, если

$$P_k = \Pr\{\eta(T,T+t)=k\} = \Pr\{\eta(t)=k\} = \frac{(\lambda t)^k}{k!}e^{-\lambda t},$$

то $\eta(T,T+t)$ не зависит от T (стационарность).

Далее покажем ординарность, используя известное разложение в ряд функции e^{-x} при малых x:

$$e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \ldots = 1 - x + o(x).$$

Если

$$P_k = \Pr\{\eta(t)=k\} = \frac{(\lambda t)^k}{k!}e^{-\lambda t},$$

то

$$P_0(dt) = \frac{(\lambda dt)^0}{0!}e^{-\lambda dt} = e^{-\lambda dt} = 1 - \lambda dt + o(dt),$$

$$P_1(dt) = \frac{(\lambda dt)^1}{1!}e^{-\lambda dt} = \lambda dt\left(1 - \lambda dt + o(dt)\right) = \lambda dt + o(dt).$$

Тогда

$$P_0(dt) + P_1(dt) = 1 + o(dt).$$

Отсюда

$$P_{>1}(dt) = \Pr\{\eta(dt) > 1\} = o(dt).$$

Впоследствии можно самостоятельно вывести, что математическое ожидание числа событий $\eta(t)$ простейшего потока на участке длины $t\left(M\left[\eta(t)\right]\right)$ равно λt, а значит интенсивность этого потока

$$\lim_{t\to 1}\frac{M\left[\eta(t)\right]}{t} = \lambda\,. \qquad (2.6)$$

Интенсивность λ пуассоновского потока часто называют *параметром потока*.

Функция распределения случайной величины τ – интервал между соседними событиями в простейшем потоке

$$F_\tau(t) = Pr\{\tau < t\} = 1 - Pr\{\tau > t\} = 1 - P_0(t) = 1 - e^{-\lambda t} \text{ для } t > 0$$

и

$$F_\tau(t) = 0 \text{ для } t < 0\,.$$

Таким образом, в простейшем потоке событий длина интервалов между событиями распределена экспоненциально.

Поток событий Эрланга. Важным для практики потоком событий является поток, образующийся в результате «просеивания» простейшего потока.

Рассмотрим поток событий, получающийся из простейшего потока, в котором сохраняется каждое второе событие, а первое отбрасывается. Длина интервалов между событиями в этом потоке представляет собой сумму двух независимых случайных величин, имеющих одинаковое экспоненциальное распределение. Распределение суммы двух независимых случайных величин, имеющих одинаковое экспоненциальное распределение, называется *распределением Эрланга второго порядка*. Если случайная величина является суммой не двух, а k независимых случайных величин, имеющих одинаковое экспоненциальное распределение,

то она имеет распределение Эрланга k-го порядка. Соответствующий поток называется потоком Эрланга k-го порядка.

Можно достаточно просто получить характеристики (моменты) распределения Эрланга k-го порядка.

Поток Эрланга k-го порядка получается, например, на выходе программы, на входе которой стоит непустая очередь заявок на ее выполнение, а сама программа состоит из k последовательно выполняемых подпрограмм, длительность каждой из которых случайна (зависит от исходных данных) и имеет экспоненциальное распределение с одинаковым математическим ожиданием. Событием в этом потоке является момент окончания обработки программой очередной заявки.

Замечательным свойством распределения Эрланга k-го порядка является то, что с его помощью можно аппроксимировать неизвестные распределения случайных величин с коэффициентами вариации от нуля до единицы по двум первым моментам распределения.

Распределение Эрланга 1-го порядка – экспоненциальное распределение, а с ростом k получаем распределения все более «регулярных» случайных величин. Это дает возможность анализировать *немарковские модели обслуживания*, т.е. модели, потоки событий в которых не обладают свойством отсутствия последействия.

2.5. Преобразование Лапласа–Стилтьеса. Производящие функции

При расчете характеристик случайных величин (СВ), кроме функций распределения, в теории массового обслуживания используются *преобразование Лапласа–Стилтьеса* (для неотрицательных непрерывных и непрерывно-дискретных СВ) и *производящие функции* (для неотрицательных целочисленных СВ). Эти инструменты удобно использовать для инженерных расчетов моментов функций распределения таких СВ, которые являются суммами фиксированного или случайного числа СВ, что нередко встречается в стохастических моделях обслуживания.

Интеграл Стилтьеса

Интегралом Стилтьеса от функции $\varphi(x)$ по функции $F(x)$ называется

$$\int_a^b \varphi(x)\,dF(x) = \lim_{\max(x_i - x_{i-1})\to 0} \sum_{i=1}^{n} \varphi(\omega_i)\left[F(x_i) - F(x_{i-1})\right],\ x_{i-1} \le \omega_i < x_i. \tag{2.7}$$

Свойства интеграла Стилтьеса

1. Если функция $F(x)$ дифференцируема на $[a,b]$, то

$$\int_a^b \varphi(x)\,dF(x) = \int_a^b \varphi(x)\frac{dF(x)}{dx}dx. \tag{2.8}$$

2. Если $F(x)$ имеет на отрезке $[a,b]$ конечное число точек разрыва $x_0, x_1, \ldots, x_n$ со скачками $F(x_i^+) - F(x_i^-) = h_i$, $x_i \in [a,b]$, то

$$\int_a^b \varphi(x)\,dF(x) = \sum_{i=1}^{n} \varphi(x_i)h_i + \sum_{i=0}^{n} \int_{x_i}^{x_i+1} \varphi(x)\frac{dF(x)}{dx}dx, \tag{2.9}$$

где $a = x_0$, $b = x_{n+1}$.

3.

$$\int_a^b \varphi(x)\,d(\alpha F_1(x) + \beta F_2(x)) = \alpha \int_a^b \varphi(x)\,dF_1(x) + \beta \int_a^b \varphi(x)\,dF_2(x). \tag{2.10}$$

4. $$\int_a^b \varphi(x)\,dF(x) = \varphi(x)F(x)\Big|_a^b - \int_a^b F(x)\,d\varphi(x). \tag{2.11}$$

Остальные свойства интеграла Стилтьеса совпадают с соответствующими свойствами интеграла Римана.

Преобразование Лапласа–Стилтьеса

Пусть имеется неотрицательная случайная величина (НСВ) ξ с функцией распределения (ФР) $F(x) = Pr\{\xi < x\}$. Тогда преобразованием Лапласа–Стилтьеса ФР $F(x)$ называется интеграл вида $F_\xi^*(s) = \int_0^\infty e^{-sx}dF(x)$, или

$$G(s)=\int_0^{\infty} e^{-sx}dF(x). \tag{2.12}$$

Свойства Лапласа–Стилтьеса

1. $G(0)=1$. (2.13)

2. $m_n=(-1)^n\dfrac{d^nG(s)}{ds^n}\Big|s=0$, (2.14)

где $m_n=\int_0^{\infty}x^n dF(x)$ – момент *n*-го порядка ФР $F(x)$.

В частности,

$M[\xi]=-G'(0)$;

$D[\xi]=G''(0)-\left[G'(0)\right]^2$.

3. Если $\xi=\sum_{i-1}^{n}\xi_i$, где ξ_i – независимые НСВ, то

$$F_{\xi}^{*}(s)=\prod_{i=1}^{n}F_{\xi_i}^{*}(s). \tag{2.15}$$

В частности, если ξ_i распределены одинаково, то

$$F_{\xi}^{*}(s)=\left[F_{\xi_i}^{*}(s)\right]^n.$$

Производящие функции

Пусть ξ – неотрицательная целочисленная СВ, принимающая значения $0,1,2,\ldots,n,\ldots$ с вероятностями $p_0,p_1,\ldots,p_n,\ldots$.

Тогда *производящей функцией распределения СВ* называется

$$P_{\xi}(z)=\sum_{n=0}^{\infty}p_n z^n. \tag{2.16}$$

Свойства производящей функции

1. $P_\xi(1)=1$. (2.17)

2. $M_\xi = P'_\xi(z)|_{z=1}$. (2.18)

3. $D_\xi = P''_\xi(z)\Big|_{z=1} + P'_\xi(z)\Big|_{z=1} - \left[P'_\xi(z)\right]^2\Big|_{z=1}$. (2.19)

4. $p_n = \frac{1}{n!}\frac{d^n P_\xi(z)}{dz^n}\Big|_{z=0}$. (2.20)

5. Если $\xi = \sum_{i=1}^{n}\xi_i$, где ξ_i – независимые целочисленные НСВ, то

$$P_\xi(z) = \prod_{i=1}^{n} P_{\xi_i}(z). \quad (2.21)$$

В частности, если ξ_i распределены одинаково, то $P_\xi(z) = \left[P_{\xi_i}(z)\right]^n$.

6. Если $\xi = \sum_{i=1}^{\nu}\xi_i$, где ν – целочисленная НСВ, ξ_i – независимые, одинаково распределенные целочисленные НСВ, то

$$P_\xi(z) = P_\nu\left[P_{\xi_i}(z)\right]. \quad (2.22)$$

Следствия. Дифференцируя (2.22) нужное число раз и полагая $z=1$, получим:

$$M_\xi = M_\nu * M_{\xi_i}, \quad (2.23)$$

$$D_\xi = D_\nu[M_{\xi_i}]^2 + D_{\xi_i} M_\nu. \quad (2.24)$$

Если снять ограничение целочисленности, наложенное на величины ξ_i, то вместо (2.22) имеет место соотношение

$$G_\xi(s) = P_\nu\left[G_{\xi_i}(s)\right]. \quad (2.25)$$

Соотношения для математического ожидания и дисперсии остаются прежними.

Пример 2.4 (характеристики пуассоновского потока)

На вход системы поступает пуассоновский поток заявок интенсивности λ.

Обозначив νt число заявок, поступающих за время $(0,t)$, найдем ПФ случайной величины νt, ее математическое ожидание и дисперсию.

Если входной поток событий обладает свойствами стационарности, ординарности и отсутствия последействия, то его называют *пуассоновским потоком*, так как распределение количества событий, произошедших в течение фиксированного интервала времени длины t, (т.е. случайной величины νt), имеет распределение Пуассона:

$$P(n) = \Pr\{\nu t = n\} = \frac{(a)^n}{n!} e^{-a},\ n = 0,1,2,\ldots,$$

где $a = \lambda t$, λ – интенсивность потока (среднее число событий в единицу времени).

Р е ш е н и е

$$P_{\nu t}(z) = \sum_0^\infty p_n z^n = \sum_0^\infty \frac{(\lambda t)^n}{n!} e^{-\lambda t} z^n = e^{-\lambda t} \sum_0^\infty \frac{(\lambda t)^n}{n!} z^n = e^{-\lambda t(1-z)}.$$

По (2.18) получим:

$$M_{\nu t} = \lambda t e^{-\lambda t(1-z)}\,|_{z=1} = \lambda t,$$

По (2.19) получим:

$$D_{\nu t} = (\lambda t)^2 + \lambda t - (\lambda t)^2 = \lambda t.$$

Пример 2.5 (случайное прореживание потока)

На вход системы поступает пуассоновский поток заявок интенсивности λ. Каждая заявка с вероятностью p принимается и с вероятностью $q = 1 - p$ получает отказ.

Найти характеристики потока принятых заявок.

Р е ш е н и е

Пусть ξ_i – СВ, принимающая значения 0 и 1 с вероятностью $p_0 = 1 - p$ и $p_1 = p$. Если обозначить через $\nu(t)$ – число заявок на

входе системы за время *t*, то $\xi(t)$ – число принятых заявок за время *t* можно записать в виде $\xi(t)=\sum_{i=1}^{\nu(t)}\xi_i$.

Тогда по (2.22)

$$P_{\xi(t)}(z)=P_{\nu t}\left[P_{\xi_i}(z)\right].$$

По (2.16)

$$P_{\xi_i}(z)=1-p+pz.$$

Из решения примера 2.4

$$P_{\nu t}(z)=e^{-\lambda t(1-z)}.$$

Тогда

$$P_{\xi(t)}(z)=e^{\lambda t(1-p+pz-1)}.$$

Окончательно,

$$P_{\xi(t)}(z)=e^{-p\lambda t(1-z)}.$$

Таким образом, поток принятых заявок представляет собой поток Пуассона с параметром $p\lambda$.

Пример 2.6 (неординарный поток – групповое поступление заявок)

В вычислительную систему (ВС) поступает поток входных сообщений. Прием сообщений в систему происходит в некоторые «моменты вызова», последовательность которых образует пуассоновский поток с параметром λ. Число сообщений на входе системы в каждый из моментов вызова i – СВ ξ_i с известным математическим ожиданием $M[\xi_i]$ и дисперсией $D[\xi_i]$.

Определить математическое ожидание и дисперсию числа сообщений, поступивших на вход системы за время $(0,t)$.

Р е ш е н и е

Если $\xi(t)$ – число сообщений, поступивших на вход ВС за время $(0,t)$, а νt – число моментов вызова на интервале $(0,t)$, то

$$(t)=\sum_{i=1}^{\nu t}\xi_i,$$

и по формуле (2.22)

$$P_{\xi(t)}(z) = P_{\nu t}\left[P_{\xi_i}(z)\right].$$

Поскольку моменты вызова образуют пуассоновский поток, то

$$P_{\nu t}(z) = e^{-\lambda t(1-z)} \text{ и } M[\nu t] = D[\nu t] = \lambda t .$$

Тогда в силу (2.22)

$$P_{\xi(t)}(z) = e^{\lambda t(P_{\xi_i}(z)-1)} .$$

В соответствии с (2.23) и (2.24) получим

$$M[\xi(t)] = \lambda t\, M[\xi_i].$$

$$D[\xi(t)] = \lambda t\left(M[\xi_i]^2 + D[\xi_i]\right).$$

Пример 2.7. На этапе i обработки заявки программой (в вершине i графа, фрагмент которого показан на рис. 2.1) производится контроль исходных данных, вычисляемых на предыдущих этапах обработки. В случае обнаружения неправильности данных, производится их корректировка (в вершине k графа) и последующий возврат заявки на этап i. Вероятность успешного прохождения контроля равна p_1 вероятность успешного прохождения контроля после корректировки равна p_2 ($p_2 \geq p_1$).

Определить математическое ожидание $M[\nu_i]$ и дисперсию $D[\nu_i]$ случайного числа прохождений ν_i заявки вершины i графа, в которой производится контроль (рис. 2.1).

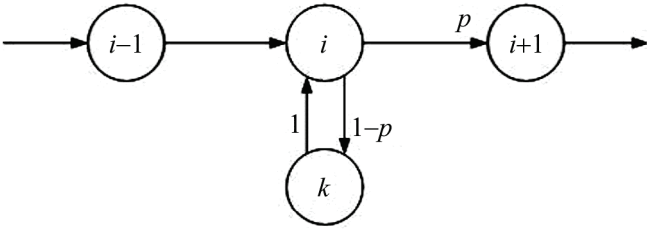


Рис. 2.1. Фрагмент схемы обработки заявки программой (i – номер этапа обработки)

Р е ш е н и е

Рассмотрим случай

$p_1 = p_2 = p$, $\nu_i \in \{1, 2, \ldots, n, \ldots\}$.

Обозначим

$p(n) = \Pr\{\nu_i = n\}$.

Тогда

$p(n) = p(1-p)^{n-1}$, $n = 1, 2, \ldots$.

Производящая функция СВ ν_i имеет вид

$$P_{\nu_i}(z) = \sum_{i=1}^{\infty} p_z z^n = \sum_{0}^{\infty} p(1-p)^{n-1} z^n = \frac{pz}{1-(1-p)z}.$$

В силу (2.18) получим

$M[\nu_i] = 1/p$.

В силу (2.19) получим

$D[\nu_i] = (1-p)/p^2$.

Решение при $p_2 > p_1$ рекомендуется аналогичным способом получить самостоятельно.

Задачи для самостоятельного решения

Задача 2.1 (экспоненциальное распределение)

Неотрицательная СВ τ распределена экспоненциально с параметром λ.

Найти ПЛС распределения СВ τ, а также $M[\tau]$ и $D[\tau]$.

Задача 2.2 (распределение Эрланга)

Обработка сообщения в специализированной ВС осуществляется K последовательными программами. Длительность работы каждой программы представляет собой СВ, распределенную экспоненциально со средним значением $T = 1/\lambda$.

Найти ПЛС распределения длительности обработки сообщения в ВС, его математическое ожидание и дисперсию.

Задача 2.3

Доказать, что сумма двух величин, распределенных по закону Пуассона с параметрами α_1 и α_2, распределена по закону Пуассона.

Указание: использовать производящие функции.

Задача 2.4

На вход специализированной системы обработки данных поступает поток сообщений, формируемый следующим образом. Имеется N абонентов, каждый из которых формирует сообщение, предназначенное для обработки системой. Каждое сообщение проходит первичный контроль, в результате которого оно либо принимается, либо отправляется абоненту на исправление и доработку. Пусть:

1) прием каждого сообщения происходит с вероятностью p (собранные вместе пачки сообщений, прошедших первичный контроль, поступают на вход системы);

2) моменты прихода пачек сообщений на вход системы образуют пуассоновский поток событий с параметром λ.

Построить производящую функцию распределения числа сообщений, поступающих в систему за время $(0,t)$, и определить математическое ожидание и дисперсию этого числа.

Задача 2.5

Решить задачу 2.4, считая, что в формировании пачки сообщений участвует не фиксированное, а случайное число ν абонентов, имеющее математическое ожидание $M[\nu]$ и дисперсию D_ν.

2.6. Марковские процессы. Уравнения Колмогорова

Для получения характеристик производительности и времени реакции системы КИС на запросы используются стохастические модели ТМО. События в таких системах – это моменты изменения состояния элементов системы.

> Если все потоки событий, изменяющих состояния системы, пуассоновские, то систему называют *марковской*, так как изменение ее состояния во времени можно описать с помощью *марковского процесса.*

С состоянием системы в момент t связывается некоторая случайная величина (СВ) $\xi(t)$.

> Поскольку $\xi(t)$ есть функция времени, то это – случайный процесс.

Обычно *указанная СВ* либо число заявок в системе, либо вектор, отдельные компоненты которого суть числа заявок на каждой фазе (если модель многофазная), либо числа заявок каждого типа (если в системе неоднородные заявки, например в приоритетной системе обслуживания).

Для системы, в которой учитываются отказы устройств, одна компонента вектора $\xi(t)$ может представлять собой число требований в системе, а другая – число исправных устройств. Пространство возможных значений $\xi(t)$ – *пространство состояний системы* – обычно дискретно:

$$\xi(t) \in S = \{S_0, S_1, \ldots, S_n, \ldots\}.$$

Случайный процесс $\xi(t)$ называется *марковским процессом*, если

$$\Pr\{\xi(t_{n+1}) = S_j | \xi(t_n) = S_i, \xi(t_{n-1}) = S_k, \ldots, \xi(t_1) = S_r\} =$$
$$= \Pr\{\xi(t_{n+1}) = S_j | \xi(t_n) = S_i\} = p_{ij}(t_n, t_{n+1}). \quad (2.26)$$

Если $p_{ij}(t_n, t_{n+1}) = f(t_{n+1} - t_n)$, т.е. если эти вероятности перехода зависят только от разности моментов времени, а не от их расположения на оси времени, то *марковский процесс называется однородным*.

Однородный марковский процесс $\xi(t)$ с дискретным множеством состояний S определяется *распределением начальных состояний*

$$P_n(0) = \Pr\{\xi(0) = S_n\}, S_n \in S \quad (2.27)$$

и *интенсивностями переходов*

$$\lambda_{ij} = \lim_{h \to 0} \frac{\Pr\{\xi(t+h) = S_j | \xi(t) = S_i\}}{h}, \text{ где } i \neq j. \quad (2.28)$$

Однородный марковский процесс $\xi(t)$ называется *регулярным*, если

$$\Pr\{\xi(t+h) = S_i | \xi(t) = S_i\} = 1 - q_i h + 0(h), \; h \to 0, \quad (2.29)$$

$$\Pr\{\xi(t+h)=S_j|\xi(t)=S_i\}=\lambda_{ij}h+0(h),\ \ h\to 0,\ i\neq j\ . \qquad (2.30)$$

Обозначим

$$P_{ij}(t)=\Pr\{\xi(t)=S_j|\xi(0)=S_i\}\ . \qquad (2.31)$$

Функции $P_{ij}(t)$ удовлетворяют системе *прямых уравнений Колмогорова:*

$$\frac{dP_{ij}(t)}{dt}=-q_jP_{ij}(t)+\sum_{k\neq j}\lambda_{kj}P_{ik}(t),\ i,j,k\in\{0,1,2,\ldots\}\ . \qquad (2.32)$$

При любом фиксированном i указанные функции определяются как единственное непрерывное при $t\geq 0$ решение этой системы, удовлетворяющее начальному условию

$$P_{ij}(0)=\begin{cases}1 & \text{при } i=j,\\ 0 & \text{иначе.}\end{cases} \qquad (2.33)$$

Марковский процесс называется эргодическим, если существует предел

$$\lim(P_{ij}(t)_{t\to\infty})=P_j\ , \qquad (2.34)$$

где

$$P_j\geq 0,\ j=0,1,2,\ldots;\ \ \sum_{j=0}^{\infty}P_j=1\ . \qquad (2.35)$$

Распределение $\{P_j\}$ называют *стационарным распределением процесса* $\xi(t)$.

Выходные параметры модели, такие как пропускная способность, время реакции и т.д., вычисляются как функции компонентов стационарного распределения.

> Получение уравнений Колмогорова основано на рассмотрении всех возможных путей перехода системы из начального состояния i в фиксированное состояние j через все возможные промежуточные состояния и использовании при этом формулы полной вероятности и записанных выше свойств регулярного марковского процесса.

Рассмотрим на примере простой модели ВС вывод уравнений Колмогорова.

Пример 2.8. Модель системы с несколькими терминалами и одной центральной ЭВМ – одна из первых математических моделей ВС [9].

На входе системы имеются N терминалов, на каждом из которых работает пользователь. Пользователи посылают в систему запросы на решение своих задач и ожидают ответа ЭВМ.

ЭВМ решает задачи пользователей в порядке поступления запросов.

1. Пусть время решения задачи произвольного пользователя – случайная величина $t_{\text{реш}}$, распределенная экспоненциально со средним значением $\mathrm{M}\left[t_{\text{реш}}\right] = 1/\mu$, тогда ФР этой случайной величины

$$B(t) = \Pr\left\{t_{\text{реш}} < t\right\} = 1 - e^{-\mu t}, \quad t \geq 0. \tag{2.36}$$

2. Пусть интервалы между моментом получения ответа на запрос и моментом посылки нового запроса (время обдумывания) – независимые случайные величины $t_{\text{обд}}$, распределенные экспоненциально со средним значением $\mathrm{M}\left[t_{\text{обд}}\right] = 1/\lambda$.

Тогда ФР этой случайной величины:

$$A(t) = \Pr\left\{t_{\text{обд}} < t\right\} = 1 - e^{-\lambda t}, \quad t \geq 0. \tag{2.37}$$

Математическая модель должна обеспечить вычисление времени реакции системы на запрос и средней производительности системы в зависимости от других параметров (N, λ, μ).

Структура модели представлена на рис. 2.2.

3. Пусть $\xi(t)$ – число запросов в системе (в очереди или ЭВМ) в момент *t*. Тогда $\xi(t) \in \{0,1,\ldots,N\}$.

4. Пусть в начальный момент времени ЭВМ свободна, т.е. $\xi(0) = 0$. Обозначим

$$P_n(t) = P_{on}(t) = \Pr\{\xi(t) = n | \xi(0) = 0\}. \tag{2.38}$$

Используя формулу полной вероятности, запишем

$$P_0(t+dt) = P_0(t)P_{00}(dt) + P_1(t)P_{10}(dt) + P_2(t)P_{20}(dt) + \\ + \ldots + P_N(t)P_{N0}(dt), \tag{2.39}$$

где $P_{k0}(dt) = \Pr\{\xi(t+dt) = 0 | \xi(t) = k\}, k = 0,1,\ldots,N$.

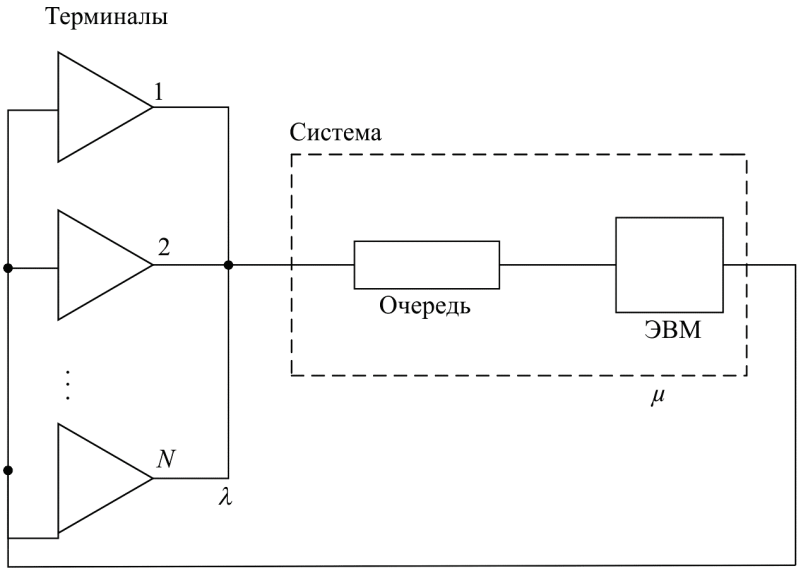


Рис. 2.2. Структура модели примера 2.8

Тогда

$$P_{00}(dt)=\left[\Pr\{t_{\text{обд}}>dt\}\right]^{N}=e^{-N\lambda dt}=1-N\lambda dt+o(dt),$$

$$P_{10}(dt)=\left[\Pr\{t_{\text{обд}}>dt\}\right]^{N-1}\Pr\{t_{\text{реш}}<dt\}=\mu dt+o(dt),$$

$$P_{20}(dt)=\left[\Pr\{t_{\text{обд}}>dt\}\right]^{N-2}\left[\Pr\{t_{\text{реш}}<dt\}\right]^{2}=o(dt).$$

И вообще $P_{k0}(dt)=o(dt)$ для всех $k\geq 2$.

Подставим выражения для $P_{k0}(dt)$ в соотношение (2.39), перенесем в левую часть $P_0(t)$, разделим обе части уравнения на dt и перейдем к пределу при $dt\to 0$. В результате получим

$$\frac{dP_0(t)}{dt}=-N\lambda P_0(t)+\mu P_1(t).$$

Аналогично можно вывести остальные уравнения:

$$\frac{dP_1(t)}{dt}=-N\lambda P_0(t)-\left(\mu+(N-1)\lambda\right)P_1(t)+\mu P_2(t),$$

...

$$\frac{dP_N(t)}{dt}=\lambda P_{N-1}(t)-\mu P_N(t). \qquad (2.40)$$

Это и есть система уравнений Колмогорова. Добавляя к ней условие нормировки

$$\sum_{n=0}^{N} P_n(t) = 1$$

и решая эту систему дифференциальных уравнений с учетом начальных условий $P_0(0) = 1$, $P_n(t) = 0$ для $n = 1,2,\ldots,N$, найдем функции $P_n(t)$ для $n = 0,1,2,\ldots,N$.

Обычно уравнения Колмогорова не выводят рассмотренным выше способом, а записывают формально, исходя из структуры *графа переходов марковской системы*. Вершины графа соответствуют состояниям системы, стрелки показывают направления переходов. Вес над стрелками соответствует интенсивности переходов λ_{ij}, выражаемых соотношениями (2.28).

Для составления уравнений Колмогорова на основе графа переходов используют такое формальное **правило**:

В левой части каждого уравнения стоит производная вероятности состояния, а в правой части находится столько слагаемых, сколько стрелок связано с этим состоянием. Если стрелка направлена из состояния, то соответствующий член имеет знак минус, если в состояние, то знак плюс. Каждый член равен интенсивности перехода, отвечающей данной стрелке, умноженной на вероятность того состояния, из которого выходит стрелка.

Граф переходов для рассматриваемой системы изображен на рис. 2.3.

Нетрудно видеть, что уравнения, составленные по графу в соответствии с приведенным правилом, совпадают с уравнениями, выведенными строго, т.е. (2.40).

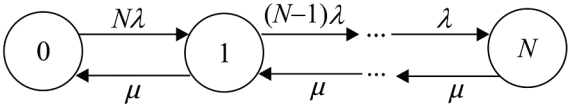


Рис. 2.3. Граф переходов модели примера 2.8

Для получения выходных параметров систем обслуживания часто бывает достаточно вычислить значения *стационарных вероятностей* P_n:

$$P_n = \lim_{t\to\infty} P_n(t), \quad n = 0,1,2,\ldots,N.$$

Вероятность P_n равна средней доле времени, проведенного системой в состоянии S_n.

Для нахождения системы уравнений относительно компонент стационарного распределения в системе уравнений (2.40) надо положить левые части равными нулю. В результате имеем систему линейных алгебраических уравнений (СЛАУ):

$$\begin{aligned} &N\lambda P_0 = \mu P_1;\\ &\left(\mu + (N-1)\lambda\right)P_1 = N\lambda P_0 + \mu P_2;\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ &\lambda P_{N-1} = \mu P_N. \end{aligned} \qquad (2.41)$$

Эта система может быть получена и непосредственно из графа по правилу «что входит, то и выходит» (*принцип сохранения потока*). Решение системы (2.41) с учетом уравнения нормировки

$$\sum_{n=0}^{N} P_n = 1$$

легко получить в явном виде последовательным выражением вероятностей P_n для $n = 1,2,\ldots,N$ через P_0 и подстановкой их в уравнение нормировки.

В результате получим

$$\begin{aligned} &P_0 = \left(1 + N\rho + N(N-1)\rho^2 + \ldots + N!\rho^N\right)^{-1};\\ &P_1 = N\rho P_0;\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ &P_N = N!\rho^N P_0, \end{aligned} \qquad (2.42)$$

где $\rho = \lambda/\mu$ («загрузка» системы).

Теперь можно найти среднее число запросов в системе $N_{\text{ср}}$ и среднюю производительность системы $\mu_{\text{ср}}$:

$$N_{\text{ср}} = \sum_{n=1}^{N} nP_n , \tag{2.43}$$

$$\mu_{\text{ср}} = \mu\left(1 - P_0\right). \tag{2.44}$$

Для нахождения математического ожидания (среднего значения) времени реакции $M[t_{\text{р}}]$ воспользуемся соотношением

$$\frac{N_{\text{ср}}}{N} = \frac{M[t_{\text{р}}]}{M[t_{\text{р}}] + 1/\lambda}. \tag{2.45}$$

В основу (2.45) положено такое соображение: *в замкнутой системе обслуживания число запросов в отдельных частях системы пропорционально времени, которое проводят в этих частях запросы*. Из (2.45)

$$M[t_{\text{р}}] = \frac{N_{\text{ср}}}{\lambda\left(N - N_{\text{ср}}\right)}. \tag{2.46}$$

С помощью аналогичного рассуждения можно получить более удобное выражение для $M[t_{\text{р}}]$, не требующее вычисления $N_{\text{ср}}$:

$$M[t_{\text{р}}] = \frac{N}{\mu(1 - P_0)} - \frac{1}{\lambda}. \tag{2.47}$$

Полученная система соотношений (2.42)–(2.47) может рассматриваться как базисная модель оценки характеристик производительности данной системы. Входящий в эту модель параметр λ является усредненной характеристикой пользователей и их действий: $M[t_{\text{обд}}] = 1/\lambda$ – пользователи решают разные задачи, требующие разное время обдумывания, ввод данных и т.д. Параметр μ является функцией технических характеристик ЭВМ и решаемых в системе задач пользователей. Эта связь должна быть установлена с помощью соотношений, называемых *интерфейсными моделями* (или *соотношениями настройки базисных*

моделей). В простейшем случае, когда время ввода/вывода информации по каждой задаче и время обмена с внешними базами данных мало по сравнению со временем ее решения в процессоре (это характерно для научно-технических задач не слишком большого объема), можно принять

$$\frac{1}{\mu} = M[T_{\text{реш}}] = \frac{n_{\text{пр}}}{V_{\text{пр}}}, \tag{2.48}$$

где $M[T_{\text{реш}}]$ – среднее время решения задачи в процессоре, с; $n_{\text{пр}}$ – среднее число операций, выполняемых процессором, на одну задачу, оп.; $V_{\text{пр}}$ – среднее быстродействие процессора, оп./с.

Процесс размножения-гибели

Марковский процесс, с помощью которого была построена математическая модель в примере 2.8, является представителем важного класса марковских процессов – *процессов размножения-гибели* (ПРГ) – процессов, граф переходов которых имеет такой вид, как на рис. 2.4. Для ПРГ решение СЛАУ относительно стационарных вероятностей состояния записывается в явном виде.

Каждое состояние в графе связано с двумя соседними, каждое из двух крайних состояний связано с одним соседним состоянием (рис. 2.4).

Для ПРГ решение СЛАУ относительно стационарных вероятностей состояния имеет вид

$$\begin{aligned}
P_0 &= \left(1 + \frac{\lambda_0}{\mu_1} + \frac{\lambda_0}{\mu_1}\frac{\lambda_1}{\mu_2} + \ldots + \prod_{k=1}^{n}\frac{\lambda_{(k-1)}}{\mu_k}\right)^{-1}; \\
P_1 &= P_0\frac{\lambda_0}{\mu_1}; \\
P_2 &= P_1\frac{\lambda_1}{\mu_2}; \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
P_n &= P_{n-1}\frac{\lambda_{n-1}}{\mu_n}.
\end{aligned} \tag{2.49}$$

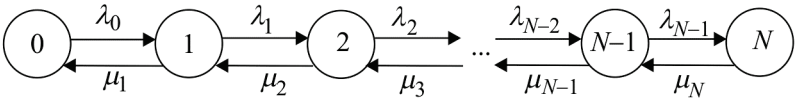


Рис. 2.4. Граф переходов ПРГ

К сожалению, стохастические модели, которые описывают взаимодействие компонентов КИС, обычно имеют более сложную структуру, чем ПРГ, несмотря на то, что характеристики потоков событий в них часто допускают аппроксимацию пуассоновскими потоками, и, следовательно, эти модели относятся к классу марковских моделей, для которых можно записать уравнения Колмогорова. Для инженерных расчетов выходных параметров в марковских моделях более сложной структуры используются различные методы агрегирования.

ГЛАВА 3

МЕТОДЫ АГРЕГИРОВАНИЯ В АЛГОРИТМАХ РАСЧЕТА МАРКОВСКИХ МОДЕЛЕЙ

3.1. Метод агрегирования с использованием параметров связи

Рассмотрим примеры структурно более сложных марковских моделей, а на их основе – методы повышения вычислительной эффективности моделей, в результате чего становится возможным проведение многовариантных расчетов [18].

Пример 3.1. Рассмотрим модель, которая была актуальна в начале 1970-х гг., накануне появления персональных компьютеров. В настоящее время она полезна с точки зрения методологии анализа структурно более сложных мар-ковских моделей. Модель использовалась как инструмент анализа ядра комплекса технических средств (КТС) специализированной автоматизированной системы обработки данных, позволяющий определить характеристики КТС в зависимости от структурных и технических параметров, входящих в КТС элементов. Упрощенная структура модели показана на рис. 3.1.

В модели отдельно показана подсистема обработки (процессорная фаза), включающая в себя несколько параллельных процессоров, и подсистема обмена (канальная фаза), включающая в себя селекторные каналы и накопители на магнитных дисках (НМД). Пусть каждый запрос пользователя требует решения задачи в процессоре и обмена информацией между ОП и НМД, осуществляемый посредством селекторного канала.

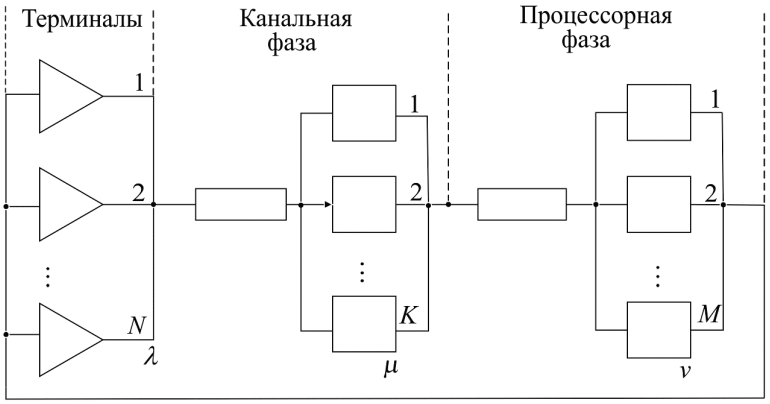


Рис. 3.1. Структура модели примера 3.1

В зависимости от конкретного варианта подключения НМД к селекторным каналам и распределения задач пользователей по НМД получаются различные формализованные схемы подсистемы обмена. Рассмотрим один из вариантов, состоящий в том, что имеется общая очередь заявок к совокупности селекторных каналов, из которой заявки выбираются в порядке их поступления.

Пусть система содержит N терминалов, K селекторных каналов и M процессоров.

Если принять допущение, что время решения задачи в процессоре, время обмена информацией между оперативной и внешней памятью и время обдумывания задачи пользователем (время между соседними запросами одного пользователя) представляют собой независимые СВ, имеющие экспоненциальные распределения, то изменение состояния системы во времени может быть описано марковским случайным процессом. В такой модели среднее время реакции $M[t_p]$ является функцией параметров $N, K, M, T_{\text{обд}}, T_{\text{обм}}, T_{\text{реш}}$.

При этом первые моменты ФР указанных случайных величин равны соответственно:

$$T_{\text{обд}} = 1/\lambda\ ,\ T_{\text{обм}} = 1/\mu\ ,\ T_{\text{реш}} = 1/\nu\ .$$

В качестве состояния системы в момент t можно взять вектор $\xi(t) = (\xi_1(t), \xi_2(t))$, где $\xi_1(t)$ – число пользователей в момент t, осуществляющих обдумывание своей задачи (число заявок на фазе терминалов), а $\xi_2(t)$ – число задач в момент t, осуществляющих

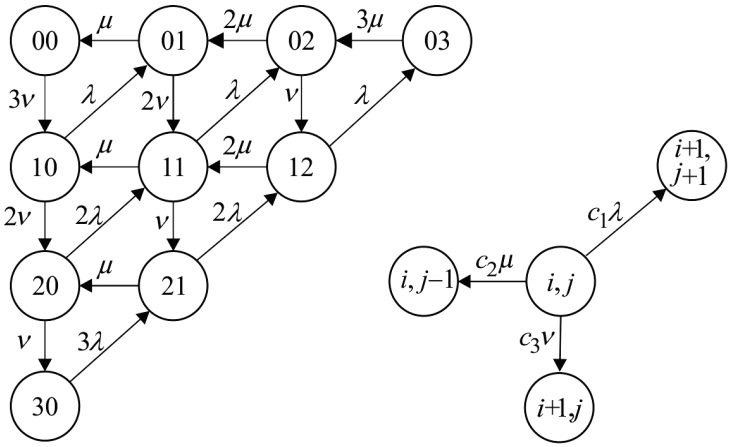
обмен или ожидающих обмен между оперативной и внешней памятью (число заявок на фазе каналов обмена).

Тогда число заявок на фазе процессоров $N-\left(\xi_1(t)+\xi_2(t)\right)$.

Граф переходов модели для частного случая $N=3,\ K=3,\ M=2$ и структура фрагмента графа переходов (показаны только исходящие дуги) для общего случая изображены на рис. 3.2, где

$$\xi_1(t)=i;\ \xi_2(t)=j;\ c_1=i;$$

$$c_2=\min\{K,j\};\ c_3=\min\left\{M,\left[N-(i+j)\right]\right\}.$$



а) *б*)

Рис. 3.2. Граф переходов (*а*) и структура фрагмента графа (*б*)

Исходя из структуры графа переходов, в соответствии с общим правилом можно записать систему линейных алгебраических уравнений (СЛАУ) относительно компонент стационарного распределения вероятностей состояния системы. Написать решение этой системы в явном виде не удается, поэтому в процедуре расчета выходных параметров, осуществляемого с помощью ЭВМ, был предусмотрен блок формирования матрицы коэффициентов системы уравнений и обращение к стандартной программе решения СЛАУ методом Гаусса. Имея значения P_{ij} компонент стационарного распределения, можно вычислить среднее число заявок на терминальной фазе $N_{\text{терм}}$, а также среднее время реакции системы $M[t_p]$:

$$N_{\text{терм}} = \sum_{i=1}^{N} i \sum_{j=0}^{N-i} P_{ij}\,; \tag{3.1}$$

$$M[t_p] = \frac{N - N_{\text{терм}}}{\lambda N_{\text{терм}}}\,. \tag{3.2}$$

СЛАУ относительно стационарных вероятностей состояния, а также соотношения (3.1) и (3.2) могут рассматриваться как базисная модель оценки характеристик производительности КТС.

Входящие в эту модель параметры λ, μ, ν являются функциями технических характеристик процессоров, селекторных каналов и накопителей на магнитных дисках, а также характеристик решаемых в системе задач пользователей. Эта связь устанавливается с помощью интерфейсных моделей.

Агрегирование модели

Использование описанной процедуры анализа, т.е. формирование матрицы коэффициентов, решение СЛАУ и расчет по формулам (3.1) и (3.2) для одной точки пространства параметров (анализа одного варианта) в многовариантных задачах анализа может оказаться затруднительным для задач большой размерности.

Действительно, уже при $N = 30$ имеем систему с числом уравнений порядка 500. Формирование коэффициентов для такой системы уравнений требует порядка десяти тысяч операций. Алгоритм Гаусса для такой системы имеет вычислительную сложность порядка 10^8 операций. Для построения кривых, характеризующих зависимости выходных параметров от конструктивных параметров системы (с учетом интерфейсных моделей) надо проделать анализ для числа вариантов порядка 10^4–10^5. Нетрудно посчитать, что это займет даже при использовании ЭВМ с быстродействием 10^8 оп./с порядка нескольких часов непрерывной работы.

И это всего лишь очень упрощенная модель анализа программно-технического комплекса.

С развитием аппаратных средств это время будет уменьшаться, но и размерности решаемых задач при этом возрастают.

По мере роста производительности вычислительных средств, используемых для анализа, увеличиваются потребности в более адекватном представлении взаимодействия элементов, входящих в состав модели, т.е. растет число компонентов и связей в модели, что обгоняет рост производительности технических средств анализа. Нужны изменения методики инженерного анализа. Аналогичная ситуация и в других инженерных отраслях [5].

Декомпозиция лежит в основе проектирования любых сложных систем – начиная от зданий, мостов, туннелей, паровозов и автомобилей и кончая самолетами, ракетами, космическими аппаратами и системами управления, энергетическими системами.

Представим один из методов декомпозиции на примере анализа рассмотренной марковской модели. Идея метода состоит в том, чтобы анализ сложной по структуре модели производить по частям, с помощью совокупности частично укрупненных («агрегированных») моделей. В каждой модели подробно представлена некоторая часть системы, а влияние остальных частей отражается некоторым обобщенным параметром (параметром связи). В итоге получается система уравнений, описывающих изменение состояния каждой из агрегированных моделей. Эти уравнения приходится решать совместно, так как в качестве переменных они содержат и параметры связи.

Рассмотрим вариант реализации метода агрегирования на примере модели примера 3.1, представленной на рис. 3.1, с использованием параметров связи.

Пример 3.2. Получить для модели, структура которой представлена на рис. 3.1, расчетные соотношения для выходных параметров в явном виде (не обращаясь к стандартной программе решения системы линейных алгебраических уравнений). Для этого рассмотрим две агрегированные модели, АМ1 и АМ2.

Модель АМ1. В агрегированной модели АМ1 канальная и процессорная фаза представлены укрупненно – одним агрегированным узлом – ОА с очередью, который описывается обобщенным параметром μ_{eff} – средней производительностью в обработке запросов, зависящей от числа заявок в этом узле. В качестве состояния системы в момент времени t берется $\xi(t)$ – число запросов в очереди или на обслуживании.

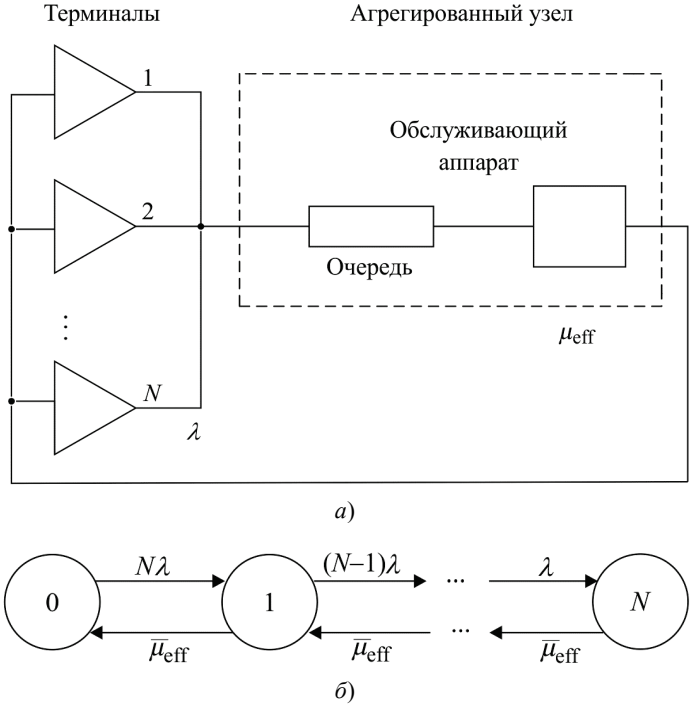


Рис. 3.3. Структура модели АМ1 (*а*) и граф переходов (*б*)

Структура модели АМ1 и граф переходов изображены на рис. 3.3,*а* и *б* соответственно.

Поскольку процесс $\xi(t)$ относится к классу ПРГ, решение системы уравнений относительно стационарных вероятностей записывается в явном виде:

$$
\begin{aligned}
P_0 &= \left(1+\frac{N\lambda}{\mu_{\text{eff}}}+\frac{N(N-1)\lambda^2}{\mu_{\text{eff}}^2}+\ldots+N!\left(\frac{\lambda}{\mu_{\text{eff}}}\right)^N\right)^{-1},\\
P_1 &= P_0\frac{N\lambda}{\mu_{\text{eff}}},\\
P_2 &= P_1\frac{(N-1)\lambda}{\mu_{\text{eff}}},\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
P_N &= P_{N-1}\frac{\lambda}{\mu_{\text{eff}}},
\end{aligned}
\qquad (3.3)
$$

$$N_{\text{ср}} = \sum_{n=1}^{N} nP_n , \tag{3.4}$$

$$M\left[t_p\right] = \frac{N_{\text{ср}}}{\lambda\left(N - N_{\text{ср}}\right)}, \tag{3.5}$$

$$\mu_{\text{ср}} = \mu_{\text{eff}}\left(1 - P_0\right). \tag{3.6}$$

Для расчета $N_{\text{ср}}$, $M\left[t_p\right]$ и $\mu_{\text{ср}}$ по (3.4)–(3.6) кроме N и λ необходимо иметь значение параметра связи μ_{eff}. Для его нахождения используется вторая агрегированная модель АМ2.

Модель АМ2. В модели АМ2 укрупненно представлена фаза терминалов. Ее влияние отражается с помощью обобщенного параметра $N_{\text{ср}}$, представляющего собой среднее число заявок в системе «процессоры-каналы» (в модели АМ2 постоянно циркулируют $N_{\text{ср}}$ заявок).

В качестве состояния в момент t берется $\eta(t)$ – число заявок на процессорной фазе. Структура модели АМ2 и граф переходов изображены на рис. 3.4,*а* и *б*.

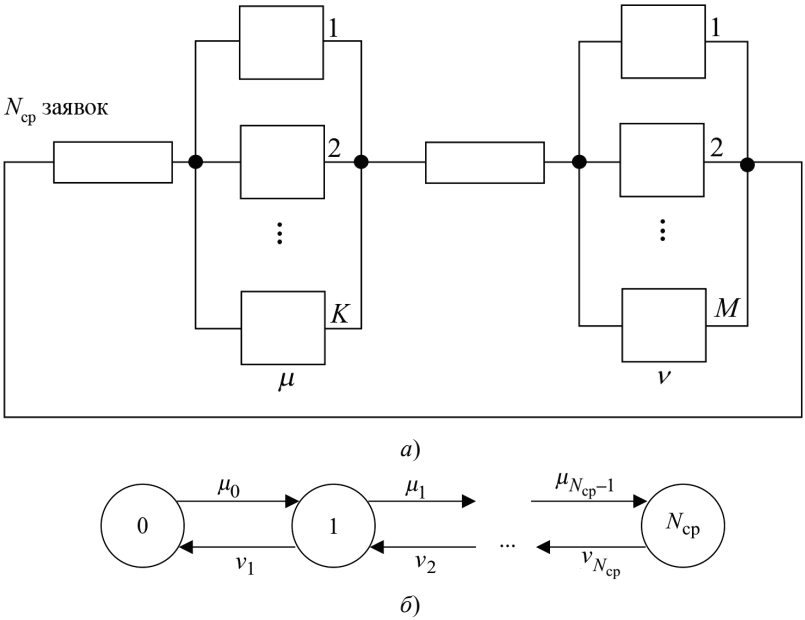


Рис. 3.4. Структура модели АМ2 (*а*) и граф переходов (*б*)

Обозначим

$$\pi_n(t) = \Pr\{\eta(t) = n\},\ n = 0,1,\ldots,N_{\text{ср}}. \tag{3.7}$$

Расчетные соотношения

$$\mu_n = \mu \min\{K, N_{\text{ср}} - n\},\ n = 0,1,\ldots,N_{\text{ср}} - 1, \tag{3.8}$$

$$\nu_n = \nu \min\{M, n\},\ n = 1,2,\ldots,N_{\text{ср}}. \tag{3.9}$$

Если обозначить через π_n компоненту стационарного распределения вероятностей числа заявок на процессорной фазе, то

$$\begin{aligned}
\pi_0 &= \left(1 + \frac{\mu_0}{\nu_1} + \frac{\mu_0\mu_1}{\nu_1\nu_2} + \ldots\right)^{-1}, \\
\pi_1 &= \pi_0 \frac{\mu_0}{\nu_1}, \\
\pi_2 &= \pi_1 \frac{\mu_1}{\nu_2}, \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\pi_{N_{\text{ср}}} &= \pi_{N_{\text{ср}}-1} \frac{\mu_{N_{\text{ср}}-1}}{\nu_{N_{\text{ср}}}},
\end{aligned} \tag{3.10}$$

$$\mu_{\text{eff}} = \sum_{n=1}^{N_{\text{ср}}} \pi_n \nu_n. \tag{3.11}$$

Для анализа АМ2, кроме K, M, μ и ν, надо знать $N_{\text{ср}}$.

Таким образом, система уравнений (3.3)–(3.11) должна решаться совместно. Совокупность соотношений (3.3), (3.4), в сущности, представляет собой уравнение

$$N_{\text{ср}} = f(\mu_{\text{eff}}). \tag{3.12}$$

Совокупность соотношений (3.8)–(3.11), в сущности, представляет собой уравнение

$$\mu_{\text{eff}} = g(N_{\text{ср}}). \tag{3.13}$$

Подставляя (3.13) в (3.12), получим

$$F(N_{\text{ср}}) = 0, \tag{3.14}$$

где

$$F(N_{\text{ср}}) = N_{\text{ср}} - f\left[g\left(N_{\text{ср}}\right)\right]. \qquad (3.15)$$

Уравнение (3.14) можно решать, например, *методом половинного деления* или *методом перебора с переменным шагом*, взяв в качестве отрезка, на котором находится корень, отрезок $[0, N]$.

При $N = 30$ для решения уравнения (3.14) методом половинного деления потребуется пять раз вычислить значение функции $F(N)$, что составляет порядка 10^3–10^4 операций, а это соответствует долям секунды (время одновариантного анализа) даже при использовании средних ЭВМ 1970-х гг. Поэтому такая декомпозиция легко допускает построение зависимостей выходных характеристик от параметров системы (*многовариантный анализ*).

В рассмотренном варианте метода декомпозиции параметрами связи между агрегированными моделями АМ1 и АМ2 являются μ_{eff} и $N_{\text{ср}}$.

3.2. Метод агрегирования на основе принципа эквивалентности потоков

Поскольку метод декомпозиции, основанный на агрегированном представлении отдельных составных частей исходной модели, изначально обладает методической погрешностью, важным этапом исследования систем с использованием агрегированных моделей является оценка этой погрешности, ее анализ и поиск методов ее уменьшения.

Для оценки методической погрешности производят вычисление значений выходных параметров (в примере 3.1 $N_{\text{терм}}$ и $M\left[t_p\right]$) в различных точках пространства параметров модели с помощью исходной «точной» модели и системы агрегированных моделей. Кроме того, используют гораздо менее трудоемкие многовариантные расчеты с варьированием исходных параметров, применяя изложенную процедуру декомпозиции. Обнаружилось, что, в силу целочисленности параметра $N_{\text{ср}}$, используемого для анализа модели АМ2, изложенная выше процедура декомпозиции

и решения на ее основе уравнения (3.15) может иметь в определенных диапазонах исходных данных заметную методическую погрешность, связанную с дискретностью этого параметра связи*.

На основании таких исследований для эффективной реализации метода агрегирования в Отраслевой лаборатории вычислительных систем (кафедра ИУ-5 МГТУ им. Н.Э. Баумана) был сформулирован *принцип эквивалентности потоков*. При его использовании часть системы заменяется агрегированным узлом – обслуживающим аппаратом (ОА) с очередью. Интенсивность обслуживания в таком ОА (параметр связи) зависит от числа заявок в узле. Принцип эквивалентности потоков состоит в том, что агрегированный узел при любом числе заявок в нем должен обеспечивать такой же поток во внешнюю сеть (такую же пропускную способность), как заменяемая им подсистема. Этот принцип в анализе марковских стохастических моделей напоминает теорему об эквивалентном генераторе тока (теорема Нортона) в анализе электрических цепей.

Пример 3.3. Получить для модели, представленной на рис. 3.1, расчетные соотношения для выходных параметров с использованием принципа эквивалентности потоков.

Рассмотрим две агрегированные модели, АЭМ1 и АЭМ2.

Модель АЭМ1. Структура модели АЭМ1 такая же, как у рассмотренной выше модели АМ1 (рис. 3.3,*а*). Отличие состоит в том, что интенсивность обслуживания в агрегированном узле (параметр связи) не постоянная (μ_{eff}, зависящая от среднего количества заявок $N_{\text{ср}}$ в узле), а вектор $\{\mu_e(n)\}$, каждая компонента которого зависит от числа заявок n в агрегированном узле.

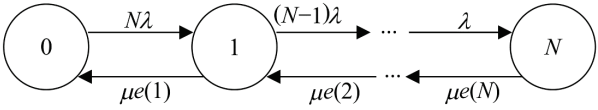


Рис. 3.5. Граф переходов модели АЭМ1

---

* Примеры таких расчетов в среде Microsoft office 2010 (основной инструмент расчетов – электронные таблицы Excel) с анализом результатов) приведены в Приложении.

Поскольку процесс $\xi(t)$ относится к классу ПРГ, решение системы уравнений относительно стационарных вероятностей записывается в явном виде:

$$P_0 = \left(1 + \frac{N\lambda}{\mu_e(1)} + \frac{N(N-1)\lambda^2}{\mu_e(1)\mu_e(2)} + \ldots + \frac{N!\lambda^N}{\prod_{n=1}^{N}\mu_e(n)} \right)^{-1},$$

$$P_1 = P_0 \frac{N\lambda}{\mu_e(1)},$$

$$P_2 = P_1 \frac{(N-1)\lambda}{\mu_e(2)},$$

...

$$P_N = P_{N-1} \frac{\lambda}{\mu_e(N)}, \qquad (3.16)$$

$$N_{\text{ср}} = \sum_{n=1}^{N} n P_n, \qquad (3.17)$$

$$M\left[t_p\right] = \frac{N_{\text{ср}}}{N - N_{\text{ср}}} \frac{1}{\lambda}, \qquad (3.18)$$

$$\mu_{\text{ср}} = \sum_{n=1}^{N} \mu_e(n) P_n. \qquad (3.19)$$

Следует отметить, что для расчета $N_{\text{ср}}$, $M\left[t_p\right]$ и $\mu_{\text{ср}}$ по формулам (3.16)–(3.19), кроме N и λ, необходимо иметь значения компонент вектора параметра связи $\mu_e(n)$ $(n=1,2,\ldots,N)$. Для его нахождения используется вторая агрегированная модель АЭМ2.

Модель АЭМ2. В модели АЭМ2 укрупненно представлена фаза терминалов. Ее влияние отражается с помощью обобщенного параметра *m*, представляющего собой число заявок в системе «процессоры-каналы». В качестве состояния в момент времени t берется число заявок на процессорной фазе. Структура модели АЭМ2 совпадает со структурой модели АМ2 (рис. 3.4,*а*), но она рассматривается не с фиксированным в ней числом заявок $N_{\text{ср}}$, а анализируется последовательно N раз, с циркулирующим в ней числом заявок m, где $m=1,2,\ldots,N$.

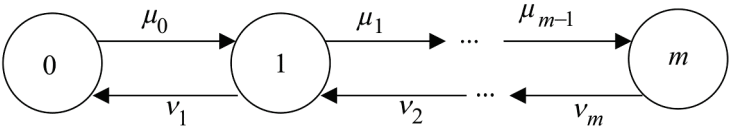


Рис. 3.6. Граф переходов модели АЭМ2

Граф переходов (в АЭМ2 циркулирует m заявок) изображен на рис. 3.6.

Расчетные соотношения таковы. Для каждого значения $m = \overline{1, N}$ рассчитываются последовательно:

$$\mu_n = \mu \min\{K, m-n\},\ n = 0,1,\ldots,m-1; \tag{3.20}$$

$$\nu_n = \nu \min\{M, n\}, n = 1,2,\ldots,m. \tag{3.21}$$

Если обозначить через π_n – компоненту стационарного распределения вероятностей числа заявок на процессорной фазе, то

$$\pi_0 = \left(1 + \frac{\mu_0}{\nu_1} + \frac{\mu_0\mu_1}{\nu_1\nu_2} + \ldots\right)^{-1},$$

$$\pi_1 = \pi_0 \frac{\mu_0}{\nu_1},$$

$$\pi_2 = \pi_1 \frac{\mu_1}{\nu_2}, \tag{3.22}$$

...

$$\pi_m = \pi_{m-1} \frac{\mu_{m-1}}{\nu_m};$$

$$\mu_e(m) = \sum_{n=0}^{m-1} \mu(n)\pi(n). \tag{3.23}$$

После N расчетов значений $\mu_e(m)$ по формулам (3.22), (3.23) полученные значения подставляются в формулы (3.16) расчета стационарных вероятностей состояний модели АЭМ1. После этого рассчитываются выходные параметры системы по формулам (3.17)–(3.19).

3.3. Методика последовательного агрегирования

Процедура агрегирования (частичного укрупнения) марковских моделей может использоваться в анализе одной системы многократно, при этом анализируемая модель последовательно упрощается.

Пример 3.4. Получить в явном виде расчетные соотношения для выходных параметров стохастической модели, структура которой показана на рис. 3.7.

От рассмотренной ранее модели она отличается подсистемой обмена. В ней очереди к селекторным каналам раздельные, и для каждого канала средние времена обмена различные. Такая модель может использоваться, например для случая, когда системные программы и общая база данных записаны на магнитных дисках, соединенных с ОП через первый канал, а прикладные программы пользователей или относящиеся к ним наборы данных – на дисках, соединенных с ОП через другой канал.

Для получения значений вероятности обращения к каждому каналу и значений среднего времени обмена нужна разработка интерфейсных моделей, учитывающих частоты обращения к тем или иным данным на дисках, параметры быстродействия НМД, размеры блоков данных, которыми осуществляется обмен, и т.д.

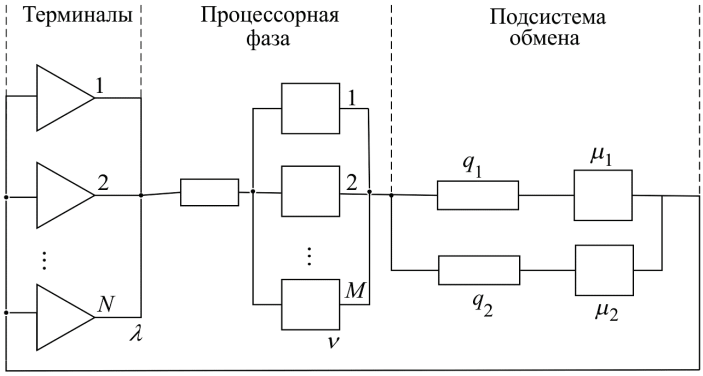


Рис. 3.7. Структура модели с раздельными каналами обмена

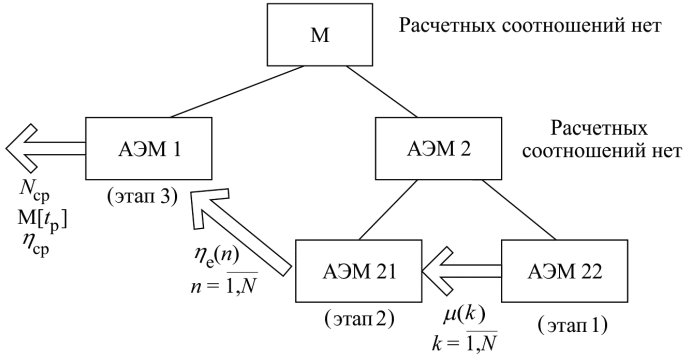


Рис. 3.8. Общая структура взаимодействия агрегированных моделей

Для описания состояния такой системы придется использовать вектор с тремя компонентами, например, число заявок на фазе терминалов, число заявок на фазе процессоров, число заявок в первом канале обмена и очереди к нему (тогда однозначно определяется и число заявок во втором канале обмена и очереди к нему).

Трехмерный граф переходов этой системы – это трехмерный аналог процесса размножения–гибели, но его затруднительно как изобразить с интенсивностями переходов из состояния в состояние, так и написать аналитические выражения для компонент стационарного распределения вероятностей состояния, являющихся решением соответствующей системы линейных алгебраических уравнений. Поэтому для расчета последовательно используем принцип эквивалентности потоков, заменяя отдельные части модели агрегированными узлами. Общая структура представления модели системы через совокупность агрегированных моделей представлена на рис. 3.8.

Модель АЭМ1. Агрегированная модель АЭМ1 и ее граф переходов показаны на рис. 3.9,*а*,*б*. В ней агрегированный узел «*эквивалентно*» представляет совокупность процессорной фазы и подсистемы обмена. Термин «эквивалентно», в соответствии с принципом эквивалентности потоков, означает, что при любом числе заявок n в агрегированном узле он обеспечивает такую же интенсивность поступления заявок в фазу терминалов $\left(\eta_e(n)\right)$ как заменяемая им подсистема.

За состояние системы принимаем количество заявок в агрегированном узле. Расчетные соотношения для компонент стационарного распределения и получаемых из них характеристик системы нетрудно записать в явном виде:

$$P_0 = \left(1 + \frac{N\lambda}{\eta_e(1)} + \frac{N(N-1)\lambda^2}{\eta_e(1)\eta_e(2)} + \ldots + \frac{N!\lambda^N}{\prod_{n=1}^{N}\eta_e(n)} \right)^{-1},$$

$$P_1 = P_0 \frac{N\lambda}{\eta_e(1)},$$

$$P_2 = P_1 \frac{(N-1)\lambda}{\eta_e(2)}, \qquad (3.24)$$

.....

$$P_N = P_{N-1} \frac{\lambda}{\eta_e(N)};$$

$$N_{\text{ср}} = \sum_{n=1}^{N} n P_n ; \qquad (3.25)$$

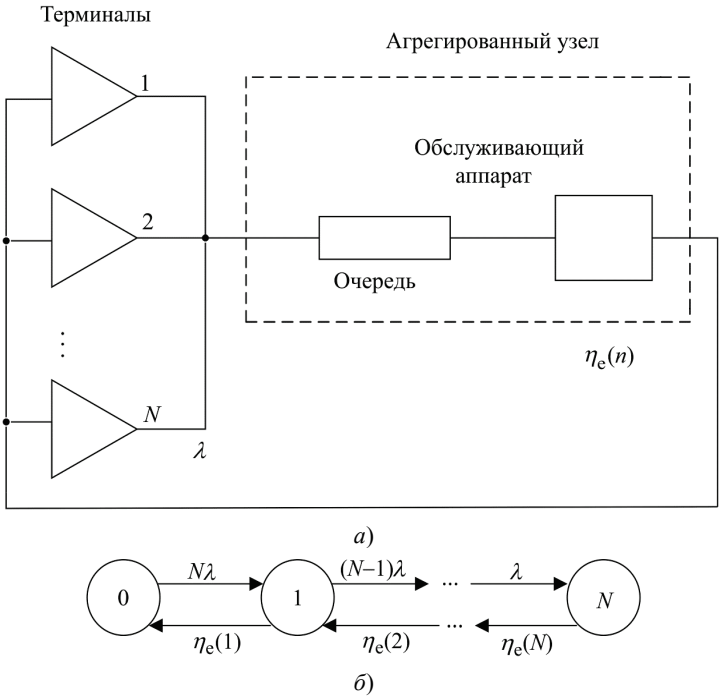


Рис. 3.9. Структура агрегированной модели АЭМ1 (*а*) и граф переходов (*б*)

$$M\left[t_p\right]=\frac{N_{\text{ср}}}{\lambda\left(N-N_{\text{ср}}\right)};\tag{3.26}$$

$$\eta_{\text{ср}}=\sum_{n=1}^{N}\eta_e\left(n\right)P_n\,.\tag{3.27}$$

Но для расчета по этим формулам необходимы значения интенсивностей поступления заявок в фазу терминалов $\eta_e\left(n\right)$. Они могут быть получены в соответствии с принципом эквивалентности потоков из анализа подсистемы процессоры/каналы, заменяемой агрегированным узлом.

Модель АЭМ2. Подсистема процессоры/каналы представлена моделью АЭМ2, структура которой показана на рис. 3.10. В ней агрегировано отражена фаза терминалов через параметр n – число заявок в модели АЭМ2. Надо для каждого значения $n=1,2,\ldots,N$ провести расчет агрегированной модели АЭМ2. Результатом расчета должны служить значения эквивалентных интенсивностей поступления заявок в фазу терминалов $\eta_e\left(n\right)$, которые позволят рассчитать стационарные вероятности состояния в агрегированной модели АЭМ1.

Состояние модели АЭМ2 описывается вектором с двумя компонентами – число заявок на фазе процессоров и число заявок в первом канале обмена и очереди к нему. Для получения расчетных соотношений в явном виде снова используем агрегирование на основе принципа эквивалентности потоков.

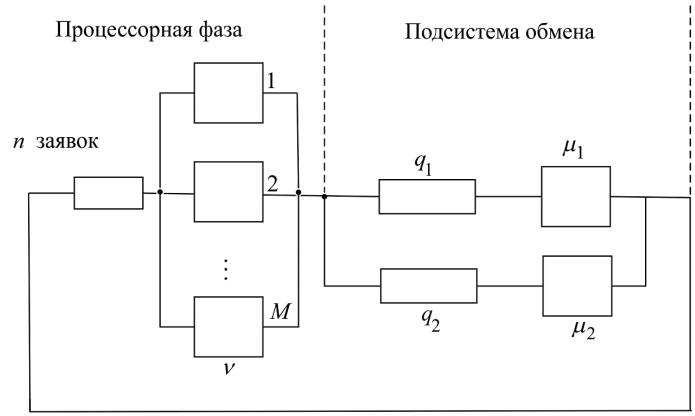


Рис. 3.10. Структура модели АЭМ2

Модель АЭМ21. В модели АЭМ21 агрегированно представлена фаза каналов. Структура модели АЭМ21 и граф переходов показаны на рис. 3.11,*а* и *б*.

За состояние системы принимается $\zeta(t)$ – число заявок на процессорной фазе с возможными значениями i $(i=0,1,\ldots,n)$. Тогда в состоянии i в агрегированной системе обмена находится $k=n-i$ заявок. Интенсивность обслуживания заявок в процессорной фазе

$$\nu(i)=\nu\min\{M,\, i\}\text{, где } i=1,2,\ldots,n\,. \tag{3.28}$$

Интенсивность обслуживания заявок в агрегированной системе обмена, в зависимости от количества заявок $\mu(k)$ $(k=0,1,\ldots,n)$, определяется на основании принципа эквивалентности потоков из анализа модели АЭМ22, рассмотренной ниже.

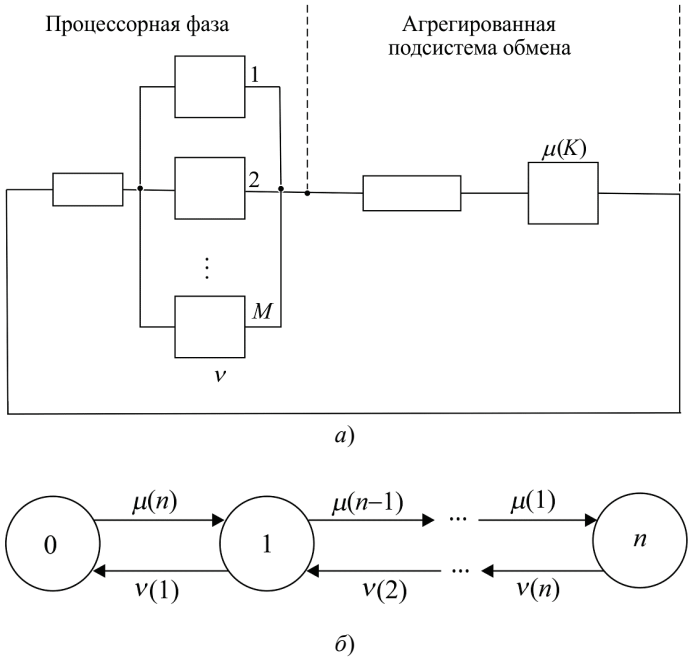


Рис. 3.11. Структура модели АЭМ21 (*а*) и граф переходов (*б*)

Обозначим $r_i = \lim_{t\to\infty} \Pr\{\zeta(t) = i\}$. Тогда остальные расчетные соотношения модели АЭМ21 имеют вид:

$$r_0 = \left(1 + \frac{\mu(n)}{\nu(1)} + \frac{\mu(n)\mu(n-1)}{\nu(1)\nu(2)} + \ldots + \prod_{i=1}^{n} \frac{\mu(n-i+1)}{\nu(i)}\right)^{-1},$$

$$r_1 = r_0 \frac{\mu(n)}{\nu(1)},$$

$$r_2 = r_1 \frac{\mu(n-1)}{\nu(2)}, \qquad (3.29)$$

.....

$$r_N = r_{N-1} \frac{\mu(1)}{\nu(n)}.$$

Эквивалентная интенсивность $\eta_e(n)$ подсистемы процессоры/каналы

$$\eta_e(n) = \sum_{i=1}^{n} \nu(i) r_i \text{ , где } n = 1,2,\ldots,N. \qquad (3.30)$$

Для расчета по (3.29)–(3.30) нужны эквивалентные интенсивности $\mu(k)$, $k = 1,2,\ldots,n$ агрегированной подсистемы обмена, которые рассчитываются на основании принципа эквивалентности потоков в модели АЭМ22.

Модель АЭМ22. Структура модели представлена на рис. 3.12,*а* (*k* – число заявок на фазе обмена).

Если за состояние системы принять $\chi(t)$ – число заявок во втором канале обмена, то граф переходов имеет вид, показанный на рис. 3.12,*б*.

Обозначим

$S_j = \lim_{t\to\infty} \Pr\{\chi(t) = j\}$ и $u = q_2\mu_1 / q_1\mu_2$.

Тогда

$$S_0 = \left(1 + u + u^2 + \ldots + u^k\right)^{-1};$$

$$S_1 = S_0 u;$$

$$S_2 = S_1 u; \qquad (3.31)$$

.....

$$S_k = S_{k-1} u.$$

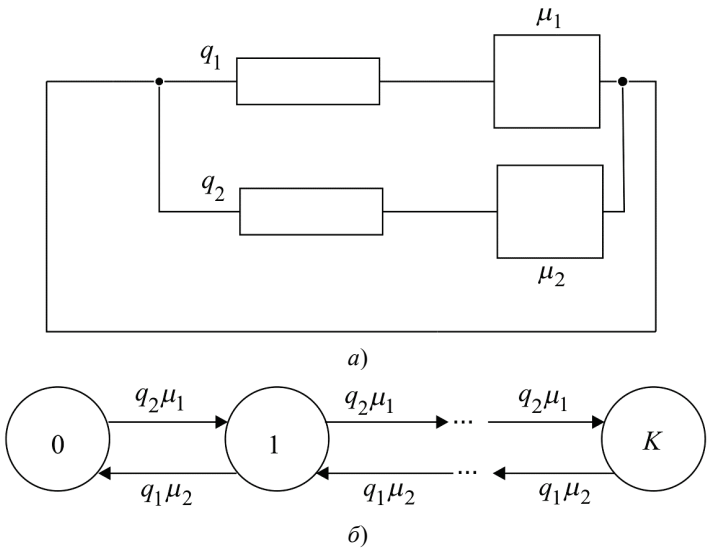


Рис. 3.12. Структура модели АЭМ22 (*а*) и граф переходов (*б*)

Пропускная способность модели АЭМ22 (эквивалентная интенсивность переходов агрегированной подсистемы обмена, когда в ней находится k заявок), выражается соотношением

$$\mu(k) = \mu_2(1 - S_0) + \mu_1(1 - S_k). \tag{3.32}$$

Таким образом, *последовательность расчета модели системы с раздельными очередями к каналам такова*.

Этап 1. Для каждого значения $k = 1, 2, \ldots, N$ производится расчет по формулам (3.31) и (3.32) значений $\mu(k)$.

Этап 2. Для каждого значения $n = 1, 2, \ldots, N$ производится расчет по формулам (3.28)–(3.30) значений $\eta_e(n)$.

Этап 3. Производится расчет по формулам (3.24)–(3.27) значений выходных характеристик основной модели: $N_{\text{ср}}, M\left[t_p\right], \eta_{\text{ср}}$.

3.4. Метод укрупнения состояний марковских моделей

Рассмотренный выше метод агрегированного представления отдельных фрагментов структуры самой модели с использованием принципа эквивалентности потоков строится на основе схемы взаимодействия заявок и ресурсов. Но в анализе марковских моделей используется также *укрупнение структуры графа переходов марковского процесса*, описывающего состояние модели. Такое укрупнение представляет собой объединение в одно макросостояние всех состояний, обладающих некоторым признаком, и запись уравнений относительно вероятностей макросостояний.

В теории марковских процессов рассматриваются условия, которым должна удовлетворять структура графа переходов для того, чтобы процесс допускал эквивалентное укрупнение. В этом случае система линейных алгебраических уравнений относительно вероятностей стационарных состояний, построенная на основе графа переходов, просто допускает сложение некоторой группы уравнений с вынесением за скобки общих множителей (интенсивностей перехода), в результате которого в скобках получается сумма вероятностей некоторой группы состояний. Эта сумма объявляется вероятностью макросостояния, включающего в себя состояния, входящие в группу. На практике применяется и *квазиэквивалентное укрупнение*, при котором условия эквивалентного укрупнения выполняются приближенно. В этом случае между вероятностями макросостояний, вычисленными на основании исходного графа (без укрупнения), и вероятностями макросостояний, вычисленными на основе укрупнения состояний, имеет место приближенное равенство, причем можно построить процедуры, дающие оценку погрешностей выходных параметров модели, вызванных квазиэквивалентным укрупнением.

Укрупнение состояний моделей применяется, в частности, при анализе моделей с неоднородными заявками (например, в случае приоритетных дисциплин, прерывающих обслуживание, или моделей, описывающих совместное использование заявкой разнородных ресурсов, а также моделей функциональной надежности, учитывающих при оценке характеристик производительности отказы и восстановления компонентов системы.

Рассмотрим на примерах применение методов эквивалентного и квазиэквивалентного укрупнения структуры графа переходов марковского процесса, описывающего состояние модели.

Пример 3.5. Предлагается провести анализ модели из примера 2.8 (см. гл. 2, 2.5), структура которой показана на рис. 2.2, *с учетом возможных отказов и восстановлений ЭВМ*.

Допустим, что интервалы безотказной работы и времена ремонта ЭВМ – независимые случайные величины, имеющие экспоненциальные распределения; их средние значения, соответственно, $T_0 = 1/\alpha$ и $T_в = 1/\beta$.

Состояние системы в момент времени t описывается вектором $\xi(t) = (\xi_1(t), \xi_2(t))$, где $\xi_1(t) \in \{0,1\}$ – число неисправных ЭВМ, $\xi_2(t) \in \overline{\{0, N\}}$ – число задач в очереди и в ЭВМ.

Граф переходов марковского процесса $\xi(t)$ имеет структуру, показанную на рис. 3.13,*а*.

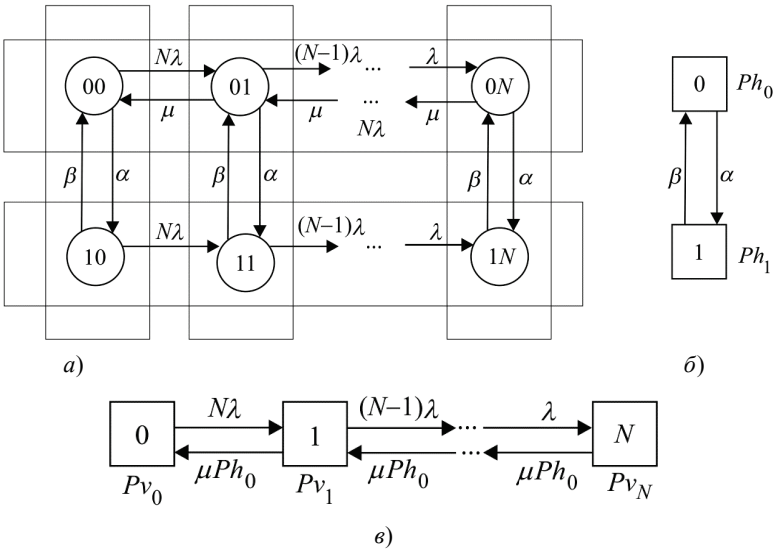


Рис. 3.13. Графы переходов марковского процесса:
а – граф исходного процесса; *б* – граф процесса с эквивалентным укрупнением состояний; *в* – граф процесса с квазиэквивалентным укрупнением состояний

Для системы уравнений, соответствующей этому графу переходов, написать решение в явном виде не удается. Конечно, для анализа системы можно воспользоваться стандартными программами решения систем линейных алгебраических равнений. Но часто бывает удобно иметь хотя и приближенные, но простые алгебраические соотношения, связывающие выходные параметры с внутренними параметрами базисной модели, для предварительной прикидки, оценки вариантов и получения зависимостей качественного характера. Для получения таких соотношений проведем укрупнение состояний графа переходов процесса $\xi(t)$. Обозначим

$$Ph_0 = \sum_{j=0}^{N} P_{0j}\,,\ Ph_1 = \sum_{j=0}^{N} P_{1j}\,.$$

Символ h – от слова horizontal (горизонтально) – укрупнение по горизонтали.

Запишем уравнения Колмогорова для стационарного распределения. Сложим уравнения, соответствующие состояниям верхнего ряда графа переходов, а затем уравнения, соответствующие состояниям нижнего ряда.

Сокращая одинаковые элементы в левой и правой частях уравнений и вынося за скобки общие множители, получим в итоге два уравнения относительно вероятностей Ph_0 и Ph_1, соответствующих укрупненному графу, изображенному на рис. 3.13,*б*. Граф получен из исходного графа объединением в макросостояния состояний верхнего и нижнего ряда (укрупнение «по горизонтали»). В данном случае укрупнение состояний является эквивалентным, так как оно соответствует эквивалентному преобразованию исходной системы уравнений относительно стационарных вероятностей состояния.

Укрупненный граф соответствует процессу размножения-гибели и тем самым позволяет легко найти *коэффициент готовности системы* (средняя доля времени, соответствующая исправному состоянию):

$$K_{\text{г}} = Ph_0 = \left(1 + \frac{\alpha}{\beta}\right)^{-1} = \frac{\beta}{\alpha + \beta} = \frac{T_{\text{о}}}{T_{\text{о}} + T_{\text{в}}}\,.$$

Чтобы вывести простые формулы для времени реакции $M\left[t_p\right]$ и средней производительности системы $\mu_{\text{ср}}$, можно попы-

таться укрупнить граф «по вертикали», объединив состояния каждого столбца. Обозначим $Pv_j = P_{0j} + P_{1j}$, $j = 0,1,2,\ldots,N$. Однако, если сложим соответствующие уравнения Колмогорова для исходного графа, то обнаружим, что записанные уравнения, кроме стационарных вероятностей макросостояний Pv_j, содержат вероятность исходных состояний P_{ij}:

$$Pv_0 N\lambda = Pv_1\mu\left(1 - \frac{P_{11}}{Pv_1}\right),$$

$$Pv_1\left((N-1)\lambda + \mu\left(1 - \frac{P_{11}}{Pv_1}\right)\right) = Pv_0 N\lambda + Pv_2\mu\left(1 - \frac{P_{12}}{Pv_2}\right).$$

Подобным же образом выглядят остальные уравнения. Таким образом, укрупнения достичь не удалось.

Укрупненный граф можно получить при следующем допущении:

$$\frac{P_{11}}{Pv_1} = \frac{P_{12}}{Pv_2} = \ldots = \frac{P_{1N}}{Pv_N} = Ph_1. \qquad (3.33)$$

Однако соотношения (3.33) выполняются лишь приближенно, поэтому укрупнение графа является квазиэквивалентным. В результате получается система уравнений, соответствующая укрупненному графу, изображенному на рис. 3.13,*в*.

Для рассматриваемой модели квазиэквивалентное укрупнение состояний графа соответствует замене исходной модели системы с ненадежной ЭВМ системой с абсолютно надежной ЭВМ, имеющей производительность $\mu_г = K_г\mu$, т.е. сниженную с μ до $K_г\mu$. Поэтому выходные параметры можно теперь рассчитывать по формулам (2.42)–(2.47), заменив в них параметр μ на $\mu_г = K_г\mu$.

Обобщим изложенную в примере 3.5 процедуру квазиэквивалентного укрупнения.

Пример 3.6. Получить расчетные соотношения для выходных параметров модели двухпроцессорной системы с отказами и восстановлением процессоров.

Структура модели представлена на рис. 3.14.

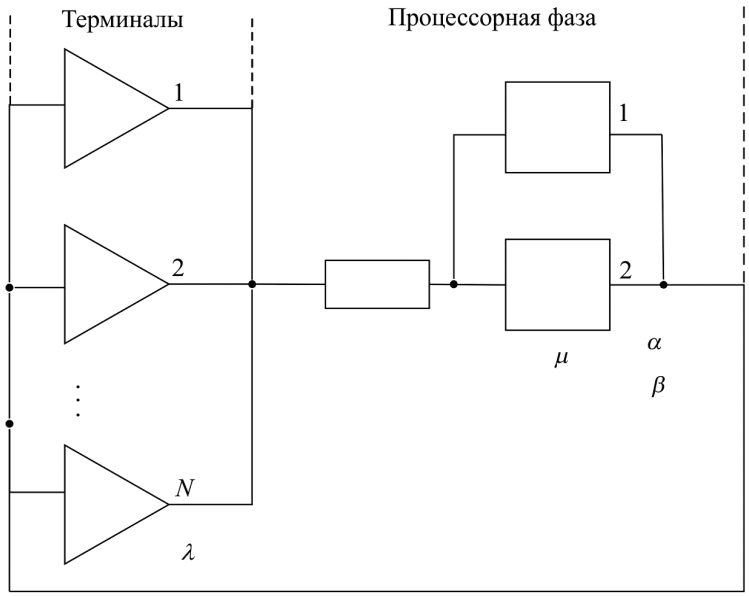


Рис. 3.14. Структура модели двухпроцессорной системы с отказами/восстановлением

В модели имеется N терминалов, за которыми работают пользователи, а обработку заявок от пользователей осуществляет двухпроцессорная система с общей очередью заявок, обслуживаемых в порядке поступления. Время решения задачи произвольного пользователя – случайная величина $t_{\text{реш}}$, имеющая экспоненциальную функцию распределения со средним значением $M\left[t_{\text{реш}}\right]=1/\mu$. Интервалы между моментом получения ответа на запрос и моментом посылки нового запроса («времена обдумывания») – независимые случайные величины $t_{\text{обд}}$, распределенные экспоненциально, со средним значением $M\left[t_{\text{обд}}\right]=1/\lambda$.

Требуется получить характеристики производительности системы с учетом возможных отказов и восстановлений процессоров.

Допустим, что интервалы безотказной работы и времена восстановления процессоров – независимые случайные величины, имеющие экспоненциальные распределения, их средние значения, соответственно, $T_{\text{о}}=1/\alpha$ и $T_{\text{в}}=1/\beta$.

Восстановление процессоров осуществляет одна ремонтная бригада, это означает, что в случае отказа второго процессора в момент ремонта первого процессора, второй процессор ожидает момента окончания ремонта первого, после чего начинается его ремонт.

Состояние системы в момент времени t описывается вектором $\xi(t)=(\xi_1(t),\xi_2(t))$, где $\xi_1(t)$ – число неисправных процессоров в момент времени t, $\xi_1(t)\in\{0,1,2\}$, $\xi_2(t)$ – число задач в момент t, обрабатываемых в процессорах или ожидающих в очереди начала обработки.

При введенных допущениях о распределениях случайных величин, вызывающих изменения состояния системы вектор $\xi(t)$ представляет собой марковский процесс. Исходный граф переходов этого процесса имеет структуру, показанную на рис. 3.15,*а*, а на рис. 3.15,*б* и *в* изображены графы переходов макросостояний, полученных в результате объединения в группы состояний исходного марковского процесса.

Обозначим, как и в примере 3.5, вероятность макросостояний, укрупненных «по горизонтали», через Ph_i:

$$Ph_i=\sum_{j=1}^{N}P_{ij}\,, \tag{3.34}$$

где $i=0,1,2$.

Запишем уравнения Колмогорова, соответствующие стационарному режиму (производные в левой части уравнений равны нулю), для всех состояний исходного процесса.

При сложении уравнения для состояний верхнего ряда, сокращении членов, входящих в правую часть уравнений с разными знаками, и вынесении общих множителей за скобки, получим уравнение

$$Ph_0 2\alpha = Ph_1\beta\,. \tag{3.35}$$

Если внимательно посмотреть на структуру графа исходного процесса (рис. 3.15,*а*) и вспомнить формальное правило записи уравнений Колмогорова, исходя из структуры графа переходов, становится очевидным, что если сложить уравнения, записанные для состояний верхнего ряда, то в них сократятся члены с интенсивностями перехода, соответствующими горизонтальным дугам, потому что они в одно уравнение входят со знаком плюс, а в дру-

гое уравнение из этой же группы со знаком минус. Остаются только члены с интенсивностями перехода, соответствующими вертикальным дугам, а эти интенсивности одинаковы для всех состояний горизонтального ряда, поэтому они выносятся за скобки. Из этого следует, что уравнение (3.35) можно записать непосредственно из структуры графа исходного процесса. В уравнение (3.35) входят только вероятности макросостояний (групп состояний, имеющих одинаковый первый индекс – число неисправных процессоров).

Совершенно аналогично записываются уравнения для макросостояний среднего и нижнего ряда:

$$Ph_1(\alpha+\beta) = Ph_0 2\alpha + Ph_2\beta; \tag{3.36}$$

$$Ph_2\beta = Ph_1\alpha. \tag{3.37}$$

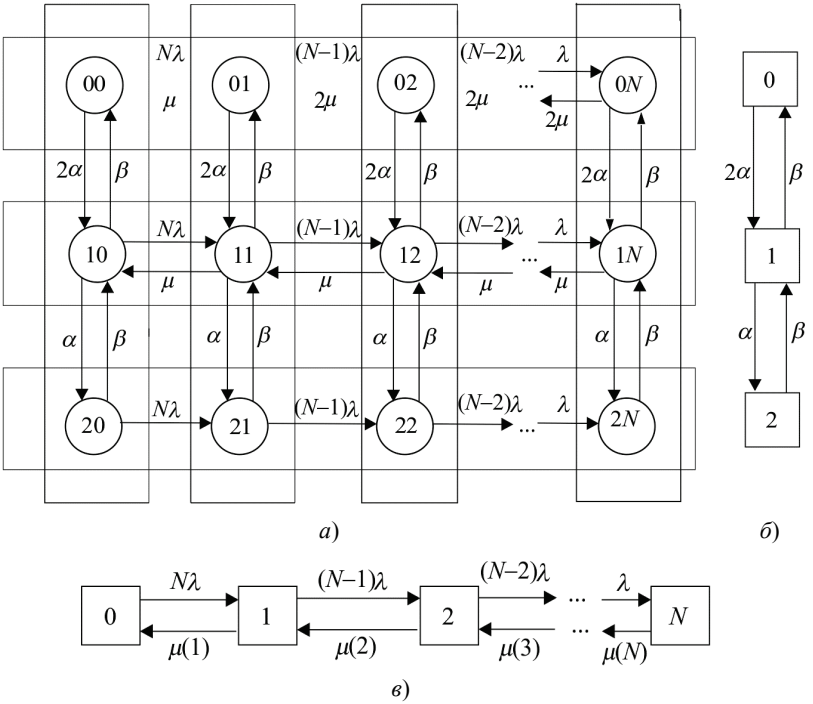


Рис. 3.15. Графы переходов марковского процесса:
а – граф исходного процесса; *б* – граф процесса
с эквивалентным укрупнением состояний; *в* – граф процесса
с квазиэквивалентным укрупнением состояний

Сопоставив уравнения (3.35)–(3.37) со структурой графа на рис. 3.15,*б*, видим, что они полностью соответствуют друг другу.

Таким образом, укрупнение состояний исходного графа «по горизонтали», т.е. объединение состояний с одинаковым первым индексом (числом неисправных процессоров), получилось эквивалентным, что означает: если решить СЛАУ для вероятностей стационарных состояний исходного графа и потом сложить вероятности состояний с одинаковым первым индексом, то получится тот же результат, как при решении СЛАУ для графа рис. 3.15,*б*, которое записывается в явном виде просто по формулам для процесса размножения/гибели.

Попытаемся проделать аналогичную процедуру с группами уравнений «по вертикали», объединив в макросостояния состояния с одинаковым вторым индексом (числом задач в процессорной фазе). Обозначим при этом

$$Pv_j = \sum_{i=0}^{2} P_{ij} \text{, где } j = 0,1,\ldots,N \text{ (символ } v \text{ от слова vertical).}$$

Оказывается, если принять допущение, что

$$P_{ij} = Ph_i Pv_j \text{ для } \forall i = 0,1,2 \text{ и } j = 1,2,\ldots,N, \tag{3.38}$$

то в этом случае граф, изображенный на рис. 3.15,*в*, соответствует исходному графу переходов, если

$$\mu(1) = \mu Ph_0 + \mu Ph_1, \tag{3.39}$$

$$\mu(2) = \mu(3) = \ldots = \mu(N) = 2\mu Ph_0 + \mu Ph_1. \tag{3.40}$$

Выходные параметры модели выражаются теперь соотношениями

$$N_{\text{ср}} = \sum_{j=1}^{N} jPv_j, \tag{3.41}$$

$$M\left[t_p\right] = \frac{N_{\text{ср}}}{\lambda\left(N - N_{\text{ср}}\right)}. \tag{3.42}$$

Поскольку справедливость допущения (3.38) не доказана, укрупнение графа на основании этого допущения следует считать квазиэквивалентным. По-видимому, стоит исследовать экспериментально, задав диапазоны изменения параметров λ, μ и N, какую ошибку вносит это допущение в расчеты P_{ij}, и проверить, насколько существенна в результате величина расхождения значе-

ний $N_{\text{ср}}$ и $M\left[t_p\right]$, полученных на основе квазиэквивалентного укрупнения, по сравнению с точным расчетом, требующим решения СЛАУ для вероятностей стационарных состояний исходного графа.

Пример 3.7. Рассчитать с помощью квазиэквивалентного укрупнения состояний характеристики надежности резервированной ИВС, содержащей устройства двух типов, в которой ремонт осуществляется специализированными бригадами ремонтников, оказывающими помощь друг другу.

Пусть система содержит *N* процессоров и *M* однотипных периферийных устройств (ПУ). Среднее время наработки на отказ процессора $1/\alpha$, периферийного устройства $1/\gamma$; среднее время восстановления процессора $1/\beta$, ПУ – $1/\delta$. Имеются три ремонтные бригады, две из которых закреплены за ПУ, одна – за процессорами. Если бригада свободна (для нее нет неисправного устройства данного типа), то она может ремонтировать устройство другого типа. Интенсивность ремонта устройства бригадой, оказывающей помощь, составляет 80% от интенсивности ремонта устройства основной бригадой. Одна бригада ремонтирует только одно устройство (взаимопомощь используется только в случае, когда имеется соответствующее число отказавших устройств). Математическая модель должна оценивать коэффициенты готовности процессорной и периферийной части и системы в целом в зависимости от $N, M, \alpha, \beta, \gamma, \delta$.

Состояние системы в момент времени *t* с точки зрения исправности устройств описывается вектором $\xi(t)=\left(\xi_1(t),\xi_2(t)\right)$, где $\xi_1(t)$ – число исправных процессоров в момент времени *t*, $\xi_2(t)$ – число исправных ПУ в момент времени *t*. Если интервалы безотказной работы процессора и ПУ, а также длительности их восстановления аппроксимировать экспоненциальными распределениями, то $\xi(t)$ представляет собой марковский процесс, для которого нетрудно построить граф переходов.

Объединим в одно макросостояние i, $i \in \overline{\{0, N\}}$ все вершины графа, у которых фиксирован первый компонент $(\xi_1(t)=i)$. Макросостояние *i* соответствует совокупности состояний системы, в которых исправны *i* процессоров.

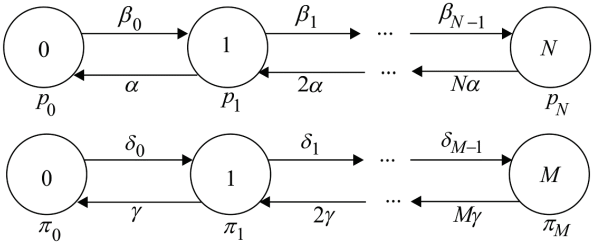


Рис. 3.16. Графы макросостояний

Стационарную вероятность макросостояния *i* обозначим

$$P_i = \sum_{j=0}^{M} p_{ij},\ i = \overline{0,N}\ .$$

Объединим теперь в макросостояние $j, j \in \overline{\{0,M\}}$ все вершины графа, у которых фиксирован второй компонент $\left(\xi_2(t) = j\right)$, т.е. в состояния, в которых исправны *j* ПУ.

Стационарная вероятность макросостояния *j*

$$\pi_j = \sum_{i=0}^{N} p_{ij},\ j = \overline{0,M}\ .$$

Полученные графы макросостояний изображены на рис. 3.16,*а* и *б*.

Условие укрупнения состояний графа переходов марковского процесса $\xi(t)$ имеет вид $p_{ij} = P_i \pi_j$, $i = \overline{0,N}, j = \overline{0,M}$. Поскольку точное выполнение этих условий не доказано, укрупнение следует считать квазиэквивалентным, и полученные далее выражения для выходных параметров модели могут иметь методическую погрешность.

Выражения для интенсивностей переходов графов макросостояний нетрудно получить, минуя этап построения графа на основе следующих рассуждений.

Величина δ_j, $j = \overline{0,M-1}$ представляет собой интенсивность ремонта ПУ в состояниях, когда исправны j ПУ (следовательно, неисправны $M-j$ ПУ). В ремонте заняты одна или две бригады, закрепленные за ПУ, а третья бригада в случае, когда исправны все N процессоров. Если $j = M-1$, то неисправно одно ПУ, поэтому здесь интенсивность ремонта $\delta_j = \delta$.

Если $j = M-2$, неисправны два ПУ, то интенсивность ремонта $\delta_j = 2\delta$.

Если $j < M-2$, т.е. число неисправных ПУ больше двух, то интенсивность ремонта составляет $\delta_j = 2\delta + 0,8\delta P_N$, $j = \overline{0, M-3}$.

Аналогично можно получить формулы для β_i:

$$\begin{aligned}
&\beta_{N-1} = \beta,\\
&\beta_{N-2} = \beta + 0,8\beta\left(\pi_{M-1} + \pi_M\right),\\
&\beta_i = \beta + 0,8\beta\pi_{M-1} + 1,6\beta\pi_M,\ i = \overline{0, N-3}.
\end{aligned} \tag{3.43}$$

Расчетные соотношения для стационарных вероятностей макросостояний:

$$P_0 = \left(1 + \frac{\beta_0}{\alpha} + \frac{\beta_0\beta_1}{2\alpha^2} + \ldots + \frac{1}{N!\alpha^N}\prod_{i=0}^{N-1}\beta_i\right)^{-1},$$

$$P_1 = P_0\frac{\beta_0}{\alpha}, \tag{3.44}$$

.....

$$\pi_0 = \left(1 + \frac{\delta_0}{\gamma} + \frac{\delta_0\delta_1}{2\gamma^2} + \ldots + \frac{1}{M!\gamma^M}\prod_{i=0}^{M-1}\delta_j\right)^{-1},$$

$$\pi_1 = \pi_0\frac{\delta_0}{\gamma}, \tag{3.45}$$

.....

Особенность приведенных соотношений состоит в том, что правые части уравнений (3.44) для расчета P_i содержат переменные π_{M-1} и π_M, а правые части уравнений (3.45) для расчета π_j содержат переменную P_N. Это означает, что систему уравнений (3.44)–(3.45) приходится решать совместно.

Для расчета можно использовать итерационную процедуру, например, в следующем виде: задавшись $P_N = 1$, рассчитать π_j, $j = \overline{0, M}$ по (3.45); определить интенсивность переходов β_i, $i = \overline{0, N-1}$, и по (3.44) рассчитать значения P_i, $i = \overline{0, N}$.

Используя полученное значение P_N, вновь рассчитать интенсивность перехода и значения π_j и P_i.

Критерием окончания процесса итераций может быть малость величины

$$\max\left\{\sum_{i=0}^{N}\left|P_i^{(r+1)}-P_i^{(r)}\right|,\ \sum_{j=0}^{M}\left|\pi_j^{(r+1)}-\pi_j^{(r)}\right|\right\},$$

где верхние индексы показывают номер итерации.

Искомые коэффициенты готовности определяются соотношениями

$$K_{\text{г}}^{\text{ПР}}=1-P_0;\ K_{\text{г}}^{\text{ПУ}}=1-\pi_0;\ K_{\text{г}}=K_{\text{г}}^{\text{ПР}}\cdot K_{\text{г}}^{\text{ПУ}}. \qquad (3.46)$$

Написать расчетные соотношения для $M\left[t_{p1}\right]$ и $M\left[t_{p2}\right]$.

Задача 3.1. Разработать математическую модель, позволяющую оценивать математическое ожидание времени реакции многопроцессорной КИС на запросы двух категорий пользователей в следующих условиях. Имеются N_1 терминалов пользователей категории 1 и N_2 терминалов пользователей категории 2. В системе имеются *M* процессоров, каждый из которых может решать задачи любой из двух групп пользователей.

Пользователи группы 1 имеют абсолютный приоритет в обслуживании своих запросов.

Интенсивность потоков запросов на решение задач пользователей каждой группы, соответственно,

$\lambda_1=1/T_{\text{обд1}}$ и $\lambda_2=1/T_{\text{обд2}}$.

Среднее время решения задач пользователей каждой группы, соответственно,

$T_{\text{реш1}}=1/\mu_1$ и $T_{\text{реш2}}=1/\mu_2$.

В допущении, что все потоки событий, изменяющих состояние системы, пуассоновские, написать расчетные соотношения для среднего времени реакции системы на запросы каждой категории пользователей $M\left[t_{p1}\right]$ и $M\left[t_{p2}\right]$.

ГЛАВА 4

МОДЕЛИ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ ОБСЛУЖИВАНИЯ С ПРИОРИТЕТАМИ

4.1. Разновидности приоритетных дисциплин

Когда при фиксированных вычислительных мощностях требуется улучшить характеристики обслуживания некоторых запросов, последним предоставляется преимущество перед другими категориями запросов в отношении порядка выбора из очереди в момент освобождения ресурса или в момент прихода запроса в систему. Это осуществляется введением той или иной разновидности обслуживания с приоритетами. В данном случае каждой заявке ставится в соответствие некоторое число – ее приоритет, который определяет очередность обслуживания заявки.

Приоритетные дисциплины характерны для любых операционных систем, осуществляющих управление мультипрограммным решением задач на ЭВМ. Среди приоритетных дисциплин выделяют статические и динамические.

Статические приоритетные дисциплины

Эти дисциплины характеризуются тем, что в них порядок обслуживания определяется заранее и не зависит от состояния системы.

Среди статических наиболее часто выделяют относительные и абсолютные приоритетные дисциплины.

Относительные приоритеты предоставляют преимущества отдельным заявкам в момент освобождения ресурса: очередь с номером j (приоритетный класс j) просматривается только в том случае, если нет ожидающих запросов в очереди $i, i < j$. Относительные приоритеты характерны для работы селекторных каналов.

Абсолютные приоритеты, кроме преимуществ в отношении порядка просмотра очередей, предоставляют запросам более высокого приоритета право прерывания запросов низкого приоритета: с дообслуживанием прерванных запросов; с возобновлением обслуживания прерванных запросов; с потерей прерванных запросов. Последние две разновидности соответствуют, например, отказам и сбоям в работе систем, т.е. отказы могут интерпретироваться как требования более высокого приоритета, а поиск и устранения неисправности – как обслуживание этого требования. Абсолютные приоритеты характерны для работы центрального процессора.

Могут быть *приоритеты, занимающие промежуточное положение между абсолютными и относительными*, например когда прерывание обслуживания может происходить в некоторых фиксированных точках программ, либо программы имеют зоны запрещения прерываний.

Могут быть *смешанные приоритеты* – это в том случае, когда в системе действуют разные правила прерывания для требований разных приоритетных классов. Например, запросы к ресурсу распределены в пять приоритетных очередей: запросы очереди 1 обслуживаются по правилу абсолютных приоритетов; запросы очереди 2 имеют право прерывания обслуживания запросов последней очереди 5; требования очередей 3 и 4 не имеют права прерывания.

Динамические приоритетные дисциплины

Эти дисциплины характеризуются тем, что в них порядок обслуживания изменяется в зависимости от длины очереди, времени ожидания требований в очереди, оставшегося времени обслуживания и т.д.

> Специфические приоритетные дисциплины используются в режиме разделения времени [1].

4.2. Метод Кобхэма

Для анализа приоритетных дисциплин используется ряд методов, отличающихся степенью точности и сложности.

Метод Кобхэма, допускающий использование в различных декомпозиционных схемах анализ сложных систем с многими ресурсами. Этот метод, также называемый *методом меченой заявки*, состоит в том, что фиксируется заявка некоторого класса, и прослеживается цепь событий, происходящих от момента поступления этой заявки в очередь до момента выхода из нее. В результате получаются соотношения для среднего значения времени ожидания рассматриваемой заявки и времени пребывания ее в системе обслуживания (времени реакции). Метод Кобхэма применяется для анализа ряда приоритетных дисциплин.

Если все потоки запросов пуассоновские, а длительность обслуживания всех категорий запросов в системе с приоритетным обслуживанием является случайной величиной с экспоненциальными функциями распределения (ФР), либо если осуществляется экспоненциальная аппроксимация неизвестных ФР по первым моментам, то для анализа таких систем можно использовать обычный аппарат: уравнения Колмогорова и методы агрегирования и эквивалентного или квазиэквивалентного укрупнения состояний.

Если же ФР длительностей обслуживания заведомо отличаются от экспоненциальных (коэффициенты вариации распределений значительно меньше единицы), и надо проводить анализ с учетом двух моментов ФР длительностей обслуживания, то используется *метод вложенных цепей Маркова* или *метод Кобхэма* (метод меченой заявки).

С помощью этих методов получены формулы для расчета среднего значения времени ожидания начала обслуживания и времени реакции системы на запрос для каждой категории запросов для различных приоритетных дисциплин обслуживания в одноканальных системах.

Структура системы показана на рис. 4.1, где входные потоки запросов (заявок) пуассоновские с параметрами λ_i, $i = \overline{1,n}$, обслуживание – с произвольными ФР, у которых известны два пер-

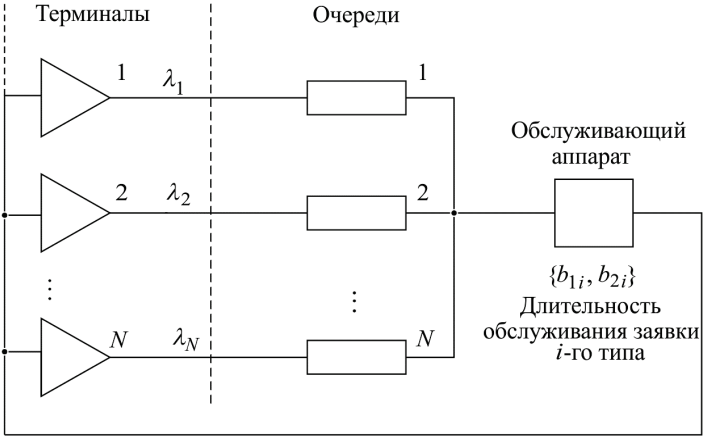


Рис. 4.1. Структура одноканальной системы с приоритетным обслуживанием заявок

вых момента. $B_i(t) = \Pr\{t_{\text{обсл}} < t\}$ – ФР длительности обслуживания заявок i-го типа (i-го приоритета).

$$b_{1i} = \int_0^\infty t dB_i(t);\ b_{2i} = \int_0^\infty t^2 dB_i(t),\ i = \overline{1,n}\ .$$

Относительные приоритеты

Рассмотрим вывод методом меченой заявки расчетных соотношений для выходных параметров модели в случае относительных приоритетов (без прерывания обслуживания).

В момент поступления в систему заявки приоритета i система может быть либо свободной, либо иметь заявку на обслуживании и $N_1, N_2, \ldots, N_n$ заявок в очередях $1,2,\ldots,n$, где N_j – случайные величины.

В этом случае при поступлении в систему заявки i-го типа эта заявка поступает на обслуживание, если система свободна, или становится в конец очереди i-го приоритета. В момент окончания обслуживания просматриваются очереди, начиная с номера 1, и при наличии в просматриваемой очереди заявок на об-

служивание принимается на обслуживание заявка, стоящая в очереди первой.

При этом среднее время реакции системы на заявку *i*-го приоритета

$$M\left[t_{pi}\right]=M\left[t_{\text{ож}i}\right]+b_{1i},$$

$$t_{\text{ож}i}=t_{\text{д}}+\sum_{j=1}^{i}\sum_{kj=1}^{N_j}t_{\text{обсл}\,kj}+\sum_{j=1}^{i-1}\sum_{kj=N_j+1}^{N_j+N_j^*}t_{\text{обсл}\,kj}, \tag{4.1}$$

где $t_{\text{д}}$ – время дообслуживания текущей заявки (если система свободна, это время равно нулю); N_j – число заявок (СВ), находящихся в очереди с номером j ($j \le i$) в момент поступления заявки *i*-го типа, $j=\overline{1,i}$; kj – номер заявки в *j*-й очереди; N_j^* – число заявок *j*-го типа (СВ), поступивших в очереди (выше приоритетом) за время ожидания обслуживания *i*-й заявки, $j=\overline{1,i-1}$.

Переходя к математическому ожиданию в левой и правой частях соотношения (4.1) и используя свойство линейности математического ожидания, получим

$$W_i=M[t_{\text{д}}]+\sum_{j=1}^{i}M\left[\sum_{kj=1}^{N_j}t_{\text{обсл}\,kj}\right]+\sum_{j=1}^{i-1}M\left[\sum_{kj=N_j+1}^{N_j+N_j^*}t_{\text{обсл}\,kj}\right],$$

где $W_i=M\left[t_{\text{ож}i}\right]$ – среднее время пребывания в очереди заявки типа *i* (среднее время ожидания обслуживания заявки типа *i*).

Далее на основании свойства математического ожидания суммы случайного числа одинаково распределенных случайных величин, и формулы Литтла определим

$$M\left[\sum_{kj=1}^{N_j}t_{\text{обсл}\,kj}\right]=M\left[N_j\right]\cdot M\left[t_{\text{обсл}\,kj}\right]=M\left[N_j\right]b_{1j}=\lambda_j W_j\,b_{1j},$$

где W_{j} – среднее время ожидания заявки типа j; $M\left[N_j\right]$ – среднее число заявок в очереди приоритета *j*, поступивших в систему за время ожидания заявки приоритета *j*.

Следовательно,

$$M\left[N_j\right]=\lambda_j W_j,\ j=\overline{1,i}.$$

Аналогично,

$$M\left[\sum_{kj=N_j+1}^{N_j+N_j^*} t_{\text{обсл}\,kj}\right]=M\left[N_j^*\right]M\left[t_{\text{обсл}\,kj}\right]=M\left[N_j^*\right]b_{1j}=\lambda_j W_i\, b_{1j},$$

где W_i – среднее время пребывания в очереди заявки типа i; $M\left[N_j^*\right]$ – среднее число заявок в очереди приоритета j, поступивших в систему за время ожидания рассматриваемой заявки (типа i), поэтому

$$M\left[N_j^*\right]=\lambda_j W_i,\ j=\overline{1,i-1}.$$

В результате

$$W_i=M\left[t_{\text{д}}\right]+\sum_{j=1}^{i}\lambda_j b_{1j}W_j+W_i\sum_{j=1}^{i-1}\lambda_j b_{1j}.$$

Обозначим $\rho_j=\lambda_j b_{1j}$, $\sigma_i=\sum_{j=1}^{i}\rho_j$, $i=\overline{1,n}$; $\sigma_0=0$, тогда

$$W_i=M\left[t_{\text{д}}\right]+\sum_{j=1}^{i}\rho_j W_j+W_i\sum_{j=1}^{i-1}\rho_j;$$

$$W_i=M\left[t_{\text{д}}\right]+\sum_{j=1}^{i}\rho_j W_j+W_i\sigma_{i-1};$$

$$W_i=\frac{M\left[t_{\text{д}}\right]+\sum_{j=1}^{i}\rho_j W_j}{1-\sigma_{i-1}},\ i=\overline{1,n}.$$

Методом математической индукции докажем, что предыдущее рекуррентное соотношение имеет следующее решение:

$$W_i=\frac{M\left[t_{\partial}\right]}{\left(1-\sigma_{i-1}\right)\left(1-\sigma_i\right)},\ i=\overline{1,n}.$$

При $i=1$

$$W_1 = M\left[t_д\right] + \rho_1 W_1 + 0 = M\left[t_д\right] + \rho_1 W_1;$$

$$W_1 = \frac{M\left[t_д\right]}{(1-\rho_1)}.$$

При $i=2$

$$W_2 = M\left[t_д\right] + \rho_1 W_1 + \rho_2 W_2 + \rho_1 W_2,\ \rho_1 = \sigma_1;$$

$$W_2(1-\rho_1-\rho_2) = M\left[t_д\right] + \rho_1 W_1,\ \rho_1 + \rho_2 = \sigma_2;$$

$$W_2(1-\sigma_2) = M\left[t_д\right] + \frac{\rho_1 M\left[t_д\right]}{(1-\rho_1)};$$

$$W_2(1-\sigma_2) = \frac{M\left[t_д\right](\rho_1 + 1 - \rho_1)}{(1-\rho_1)};$$

$$W_2 = \frac{M\left[t_д\right]}{(1-\sigma_1)(1-\sigma_2)}.$$

Допустим, что $W_j = \dfrac{M\left[t_д\right]}{(1-\sigma_{j-1})(1-\sigma_j)}$ верно для $\forall j = \overline{1,m}$.

Тогда для $j = m+1$

$$W_{m+1} = M\left[t_д\right] + \sum_{j=1}^{m+1} \rho_j W_j + W_{m+1}\sigma_m;$$

$$W_{m+1} = M\left[t_д\right] + \rho_1 W_1 + \rho_2 W_2 + \ldots + \rho_m W_m + \rho_{m+1} W_{m+1} + W_{m+1}\sigma_m;$$

$$W_{m+1}(1-\rho_{m+1}-\sigma_m) = M\left[t_д\right] + \sum_{j=1}^{m} \rho_j W_j;$$

$$\rho_{m+1} + \sigma_m = \sigma_{m+1};$$

$$W_{m+1}(1-\sigma_{m+1}) = M\left[t_д\right] + M\left[t_д\right] \sum_{j=1}^{m} \frac{\rho_j}{(1-\sigma_{j-1})(1-\sigma_j)};$$

$$W_{m+1}\left(1-\sigma_{m+1}\right)=M\left[t_{\text{д}}\right]\left(1+\frac{\rho_1}{\left(1-\sigma_1\right)}+\frac{\rho_2}{\left(1-\sigma_1\right)\left(1-\sigma_2\right)}+\ldots+\right.$$

$$\left.+\frac{\rho_{m-1}}{\left(1-\sigma_{m-2}\right)\left(1-\sigma_{m-1}\right)}+\frac{\rho_m}{\left(1-\sigma_{m-1}\right)\left(1-\sigma_m\right)}\right).$$

Складывая последовательно по два слагаемых в правой части, в итоге получим

$$W_{m+1}\left(1-\sigma_{m+1}\right)=M\left[t_{\text{д}}\right]\left(\frac{1}{\left(1-\sigma_m\right)}\right);$$

$$W_{m+1}=\frac{M\left[t_{\text{д}}\right]}{\left(1-\sigma_m\right)\left(1-\sigma_{m+1}\right)}.$$

Таким образом, доказали, что

$$W_i=\frac{M\left[t_{\text{д}}\right]}{\left(1-\sigma_{i-1}\right)\left(1-\sigma_i\right)},\ i=\overline{1,n}.$$

Среднее время дообслуживания заявки типа j определяется соотношением $M\left[t_{\text{д}j}\right]=\frac{b_{2j}}{2b_{1j}}$ [10]. Поэтому

$$M\left[t_{\text{д}}\right]=\sum_{j=1}^{n}\frac{\rho j b_{2j}}{2b_{1j}}=\frac{1}{2}\sum_{j=1}^{n}\lambda_j b_{2j}.$$

В окончательном виде запишем:
среднее время ожидания в очереди для заявки типа i:

$$W_i=\frac{\sum_{j=1}^{n}\lambda_j b_{2j}}{2\left(1-\sigma_{i-1}\right)\left(1-\sigma_i\right)},\ i=\overline{1,n};\qquad(4.2)$$

время реакции системы для заявки типа i

$$M\left[t_{pi}\right]=W_i+b_{1i},\qquad(4.3)$$

где $i=\overline{1,n}$.

Абсолютные приоритеты

Рассуждая таким же образом, как и для относительных приоритетов, для абсолютных приоритетов получим

$$W_i = \frac{\sum_{j=1}^{i} \lambda_j b_{2j}}{2(1-\sigma_{i-1})(1-\sigma_i)}, \quad i = \overline{1,n}. \tag{4.4}$$

$$M[t_{\text{р}i}] = W_i + M\left[t^*_{\text{обсл}i}\right] = W_i + \frac{b_{1i}}{(1-\sigma_{i-1})}, \quad i = \overline{1,n}, \tag{4.5}$$

где $M\left[t^*_{\text{обсл}i}\right]$ – среднее время обслуживания заявки типа i с учетом прерываний (время завершения обслуживания).

Полученные соотношения справедливы только тогда, когда $\sigma_i < 1$ (суммарная загрузка системы обслуживанием заявок типов 1, 2, …, i не превышает единицы).

Таким образом, в системе может быть, например, $\sigma_3 < 1$, а $\sigma_4 > 1$. Это означает, что заявки потоков 1, 2, 3 обслуживаются в стационарном режиме (существует конечное среднее время ожидания этих заявок), а число заявок из остальных потоков в очередях, начиная с четвертой, неограниченно растет.

Реально в любой системе стационарный режим существует либо за счет конечного числа заявок в источниках, либо за счет ограниченного числа мест для ожидания в очередях. Если же имеет место временное увеличение нагрузки (увеличение интенсивности входных потоков в «час пик» или уменьшение интенсивности обслуживания вследствие отказов) такое, что $\sigma_n > 1$, то разделение заявок на приоритетные классы приводит к тому, что более приоритетные заявки обслуживаются в стационарном режиме (для некоторых $j = 1, 2, \ldots, m$ будет $\sigma_j < 1$), а остальные потоки скапливаются в очередях, ожидая снижения пика нагрузки.

Смешанные приоритеты

Систему со смешанными приоритетами можно описать посредством матрицы прерываний $A = (a_{ij})$, $i, j = \overline{1,n}$, в которой $a_{ij} = 1$ в случае, если заявка приоритета j имеет право преры-

вать обслуживание заявки приоритета i, и $a_{ij} = 0$ в противном случае.

Для относительных приоритетов $a_{ij} = 0$ для $i, j = \overline{1,n}$.

Для абсолютных приоритетов $a_{ij} = 1$ для $i > j$, $a_{ij} = 0$ для $i \le j$.

Для любых смешанных приоритетов естественно установить $a_{ij} = 0$ для $i \le j$ (иначе будет противоречие между правилами прерывания и порядком просмотра очередей).

Среднее время ожидания и время реакции для случая смешанных приоритетов

$$W_i = \frac{\sum_{j=1}^{n} \left(1 - a_{ji}\right) \lambda_j b_{2j}}{2\left(1-\sigma_{i-1}\right)\left(1-\sigma_i\right)},\ i = \overline{1,n}; \tag{4.6}$$

$$M[t_{pi}] = W_i + M\left[t^*_{\text{обсл}i}\right]. \tag{4.7}$$

Для среднего времени обслуживания требований приоритета i с учетом прерываний можно указать две граничные оценки:

$$\frac{b_{1i}}{\left(1 - \sum_{j=1}^{n} a_{ij}\rho_j\right)} \le M\left[t^*_{\text{обсл}i}\right] \le \frac{b_{1i}}{\left(1-\sigma_{i-1}\right)},\ i = \overline{1,n}. \tag{4.8}$$

Нижняя (левая) граница – учтено обслуживание заявок, имеющих право прервать обслуживание рассматриваемой заявки приоритета *i*.

Верхняя (правая) граница – учтены все заявки более высокого приоритета, в том числе и те, которые не имеют права прерывания рассматриваемой заявки.

Пример. Допустим, в системе обслуживаются три приоритетных очереди заявок, где заявки очереди (приоритета) 1 имеют право прерывания обслуживания заявок очереди 3, а заявки очереди 2 не имеют права прерывания обслуживания этих заявок. Когда началось обслуживание заявки приоритета 3, в системе не было за-

явок в более приоритетных очередях 1 и 2. Если за время обслуживания этой заявки пришли заявки приоритета 2, но при этом не было заявок приоритета 1, то эти заявки будут обслуживаться после окончания обслуживания заявки приоритета 3 и не повлияют на время завершения ее обслуживания. Но если заявка приоритета 1 пришла после начала обслуживания заявки приоритета 3, то она прервет обслуживание этой заявки, а после окончания обслуживания заявки приоритета 1 на обслуживание попадут ожидающие в очереди заявки приоритета 2, и только потом будет продолжаться обслуживание заявки приоритета 3.

Реально при реализации обслуживания приоритетных очередей со смешанными приоритетами возможны различные варианты порядка просмотра очередей в момент освобождения обслуживающего аппарата. В частности, прерванная заявка может быть поставлена в отдельную очередь, которая просматривается ранее очередей более высокого приоритета, но не имеющих права ее прерывания. В этом случае для оценки времени завершения ее обслуживания можно брать левую границу формулы (4.8).

Обобщенные смешанные приоритеты

Эта приоритетная дисциплина включает в себя как частные случаи все предыдущие дисциплины, а также обслуживание заявок в порядке поступления.

Рассмотрим модель, в которой n типов заявок распределяются в m очередей ($n > m$), обслуживаемых в соответствии с дисциплиной смешанных приоритетов. Условия прерывания обслуживания описываются матрицей $A = (a_{ij}),\ i, j = \overline{1,m}$, состоящей из нулей и единиц.

Распределение заявок по очередям (классам приоритетов) описывается двумя способами.

Способ 1. Задается вектор $P = (p_1, \ldots, p_n)$, где p_i – номер приоритета, присвоенный заявке типа $i = \overline{1,n},\ p_i \in \{\overline{1,m}\}$.

Способ 2. Задается матрица $Z = (z_{ij}),\ i = \overline{1,n},\ \ j = \overline{1,m}$, где $z_{ij} = 1$ в случае, если заявка типа i распределена в очередь j, и $z_{ij} = 0$ в противном случае. При этом должно соблюдаться усло-

вие $\sum_{j=1}^{m} z_{ij} = 1,\ i = \overline{1,n}$ (каждый тип заявок распределяется в некоторую очередь).

В зависимости от способа описания распределения заявок по очередям изменяется вид формул, определяющих время реакции системы для заявок каждого класса.

Для способа 1 расчетные соотношения имеют вид

$$W_i = \frac{\sum_{k=1}^{n} \lambda_k b_{2k} \left(1 - a_{p_k p_i}\right)}{2\left(1 - \sum_{k \in \alpha_i} \rho_k\right)\left(1 - \sum_{k \in \alpha_i^*} \rho_k\right)},\ i = \overline{1,n}, \qquad (4.9)$$

где $\alpha_i = \left\{k : k \in \left\{\overline{1,N}\right\},\ \rho_k < \rho_i\right\}$; $\alpha_i^* = \left\{k : k \in \left\{\overline{1,N}\right\},\ \rho_k \le \rho_i\right\}$, т.е. α_i – множество типов заявок, имеющих более высокий приоритет по сравнению с заявками типа i; α_i^* – множество типов заявок, имеющих приоритет не ниже, чем заявки типа i.

Нижняя и верхняя границы для среднего времени обслуживания заявки типа i (с учетом прерываний):

$$\frac{b_{1i}}{\left(1 - \sum_{k=1}^{n} \rho_k a_{p_i p_k}\right)} \le M\left[t_{\text{обсл}i}\right] \le \frac{b_{1i}}{\left(1 - \sum_{k \in \alpha_i} \rho_k\right)},\ i = \overline{1,n}. \qquad (4.10)$$

Время реакции системы для заявки типа i, как и прежде,

$$M[t_{pi}] = W_i + M\left[t^*_{\text{обсл}i}\right],\ i = \overline{1,n}. \qquad (4.11)$$

Расчетные соотношения, определяющие время реакции для заявок каждого типа, для второго способа описания распределения приоритетов предлагаем читателю написать самостоятельно.

Приоритеты без конкуренции однотипных заявок

Рассмотрим случай дисциплины с относительными или абсолютными приоритетами, когда поступающая в систему обслуживания заявка типа i не может застать в очереди или на обслуживании заявки того же типа. Такую модель приходится рассматривать при анализе сложной системы обслуживания с блокиров-

ками на основе многоуровневого подхода. При таком подходе часть сети внутри блокированного контура заменяется одной СМО, в которой время обслуживания заявки равно времени ее пребывания в блокированной части сети, «вложенной» в данную СМО. В этом случае очередь заявок рассматриваемого класса образуется перед блокированной частью сети, а внутри нее рассматриваемая заявка встречает только заявки других типов.

Аналогично соотношению (4.1) можно записать

$$t_{\text{ож}i} = t_{\text{д}} + \sum_{j=1}^{i-1}\sum_{k=1}^{N_j} t_{\text{обсл}\,kj} + \sum_{j=1}^{i-1}\sum_{k=N_j+1}^{N_j+N_j^*} t_{\text{обсл}\,kj}\,.$$

Переходя к математическому ожиданию, получим

$$W_i = M\left[t_{\text{д}i}\right] + \sum_{j=1}^{i-1}\rho_j W_j + W_i\sum_{j=1}^{i-1}\rho_j$$

или

$$W_i = \frac{M\left[t_{\text{д}i}\right] + \sum_{j=1}^{i-1}\rho_j W_j}{1-\sigma_{i-1}}, \quad i=\overline{1,n}\,. \tag{4.12}$$

Математическое ожидание времени дообслуживания

для *относительных приоритетов*

$$M\left[t_{\text{д}i}\right] = \frac{1}{2}\sum_{\substack{j=1\\ j\neq i}}^{n}\lambda_j b_{2j}\,, \tag{4.13}$$

для *абсолютных приоритетов*

$$M\left[t_{\text{д}i}\right] = \frac{1}{2}\sum_{j=1}^{i-1}\lambda_j b_{2j}\,. \tag{4.14}$$

Рекуррентное соотношение (4.12) позволяет последовательно рассчитывать W_1, W_2, ...,W_n для каждого типа заявок.

Для задания вторых моментов b_{2j} часто удобно использовать коэффициент вариации функций распределения длительности обслуживания

$$b_{2j} = b_{1j}^2 + \sigma_{bj}^2 = b_{1j}^2\left(1 + \nu_j^2\right),$$

где σ_{bj} – среднеквадратическое отклонение длительности обслуживания заявки j-го типа, а $\nu_j = \sigma_{bj} / b_{1j}$ – коэффициент вариации распределения длительности обслуживания заявки j-го типа.

Время реакции системы для заявки типа i для случаев относительных и абсолютных приоритетов определяется соответственно формулами (4.3) и (4.5).

Обобщенные приоритеты без конкуренции однотипных заявок

В этом случае заявки разных типов могут иметь одинаковые приоритеты (n типов заявок распределяются в m приоритетных очередей), но заявка каждого типа не застает в очереди или на обслуживании заявки того же типа. Приоритетные очереди обслуживаются по правилу относительных или абсолютных приоритетов.

Нетрудно показать, что в этом случае

$$W_i = M\left[t_{\text{д}i}\right] + \sum_{\substack{j=1 \\ j\neq i \\ p_j \le p_i}}^{n} \rho_j W_j + W_i \sum_{\substack{j=1 \\ p_j < p_i}} \rho_j, \quad i = \overline{1,n}, \tag{4.15}$$

где p_j, p_i – номер приоритетных очередей заявок типа j и типа i, а $M\left[t_{\text{д}i}\right]$ для относительных приоритетов определяется формулой (4.13) и для абсолютных приоритетов имеет вид

$$M\left[t_{\text{о}i}\right] = \frac{1}{2}\sum_{\substack{j=1 \\ j\neq i \\ p_j \le p_i}}^{n} \lambda_j b_{2j}, \quad i = \overline{1,n}.$$

Систему уравнений (4.15) не удается решить в явном виде или записать в виде рекуррентного соотношения, позволяющего последовательно вычислять W_i для i от 1 до n.

Анализ структуры системы уравнений (4.15) показывает что она представляет собой совокупность подсистем, которые можно решать рекуррентно. Число таких подсистем равно m, т.е. равно

числу приоритетных очередей, а число уравнений в каждой подсистеме – числу классов заявок, распределенных в данную очередь.

Обозначим $\beta_j = \{i : p_i = j\}$, т.е. β_j – множество типов заявок, распределенных в очередь j; $n_j = |\beta_j|$ – число элементов множества β_j. Тогда первые n_1 уравнений (4.15) содержат n_1 неизвестных W_i таких, что $i \in \beta_1$. Решим эту подсистему и подставим полученные значения W_i в остальные уравнения. Тогда можно решить подсистему из следующих n_2 уравнений, содержащих n_2 неизвестных W таких, что $i \in \beta_2$.

Эффективность квазирекуррентной процедуры решения системы (4.15) тем выше, чем ближе значение m к значению n, т.е. чем меньшее число типов заявок попадает в одну и ту же очередь.

> Процедуру удается упростить и свести к рекуррентной, если заявки, распределяемые в одну очередь, являются однородными (имеют близкие значения моментов распределения длительности обслуживания и интенсивности входных потоков).

Для анализа приоритетных систем сложной структуры, включающих в себя несколько ресурсов, метод Кобхэма применяется обычно в сочетании с некоторой схемой декомпозиции, при которой тем или иным способом выделяются элементы декомпозиции (субмодели), определяется интерфейс между субмоделями, т.е. состав информации, описывающий взаимодействие субмоделей, рассчитываются значения интерфейсных переменных, выходных параметров субмоделей и затем с их помощью вычисляются выходные параметры модели в целом. В такой схеме метод Кобхэма используется для расчета характеристик субмоделей.

ГЛАВА 5

МНОГОРЕСУРСНЫЕ МОДЕЛИ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

5.1. Формализованное представление многоресурсных КИС

Инструментальной базой больших автоматизированных систем обработки информации и управления является комплекс технических средств (КТС), системное и прикладное программное обеспечение (ПО), включающие в себя многочисленные и разнообразные компоненты, находящиеся между собой в сложном взаимодействии в процессе обеспечения работы пользователей, а также обширные базы данных, состав которых определяется назначением системы.

Рабочие места пользователей, терминалы, линии связи, персональные компьютеры, объединенные в локальную сеть, серверы, связные процессоры, основная и внешняя память, центральные процессоры и специализированные процессоры ввода-вывода, разнообразные устройства ввода-вывода, программные модули, наборы данных – эти и многие другие элементы КТС и ПО АСОИУ с точки зрения обеспечения выполнения задач, возложенных на систему, являются *ресурсами*.

Когда большое число задач пользователей параллельно обрабатывается с помощью разнообразных ресурсов, возникают многочисленные *конфликтные ситуации*, связанные с одновременным наличием нескольких запросов на использование одного

ресурса. Конфликтные ситуации приводят к задержкам, влияющим в конечном итоге на эффективность работы системы. Эти задержки желательно уметь оценивать на этапе выбора основных технических решений будущей системы, при испытаниях, модернизации, анализе возможностей развития.

Особенности моделирования КИС

Рассмотрим задачу организации использования многих ресурсов, включающую рассмотрение последовательности запросов на эти ресурсы, создаваемой заданиями (задачами) по мере их прохождения через систему.

Широкая номенклатура технических и программных средств, сложная организация их взаимодействия, разнородность одновременно выполняемых системой заданий приводят к необходимости анализа моделей стохастической сетевой структуры. Дополнительные трудности возникают в связи с тем, что наряду с разделением ресурсов между задачами имеет место совместное использование одной задачей нескольких разнородных ресурсов. Так, при обработке одной задачи ей одновременно должны принадлежать терминал, область оперативной памяти, общий нерентабельный программный модуль (в частности, СУБД) и т.д. При осуществлении обмена информацией между оперативной памятью и НМД одновременно участвуют селекторный канал, УВУ и НМД. Подобные перечни можно продолжить.

В *одноуровневом представлении взаимодействия ресурсов* такую ситуацию либо отображают, вводя в структуру модели различного рода блокировки и логические условия, либо обходят, выделяя среди нескольких совместно используемых ресурсов один – критический в плане совместной реализации нескольких информационных процессов. Другие совместно используемые ресурсы при описании взаимодействия информационных процессов либо вообще не учитываются, либо учитываются косвенно путем введения некоторых ограничений (например, ограниченность памяти как ресурса часто учитывается заданием предельно допустимого уровня мультипрограммирования; совместная работа канала, УВУ и НМД отображается с помощью одного обслуживающего аппарата и т.д.).

При *многоуровневом представлении взаимодействия ресурсов* одновременное занятие заявкой нескольких ресурсов отображается на основе концепции вложенных процессов [11], расширяющей возможности применения аналитических методов исследования благодаря тому, что в моделях отдельных уровней отсутствуют сложные логические условия и блокировки.

Формализованное описание взаимодействия компонентов КИС

В основе формализации лежат понятия ресурса и заявки. Несмотря на трудность их формального определения, интуитивно представляется следующее.

Ресурс – любой аппаратный или программный компонент системы, на входе которого могут возникать конфликты, приводящие к образованию очередей и временных задержек.

Заявка (*запрос*) *на использование ресурса* – некоторое обобщенное понятие, физический смысл которого зависит от уровня и области исследования системы, ее структуры, способа организации, режима функционирования.

Например, функции ЭВМ могут заключаться в выполнении единиц работы, организованных в виде заданий. В этом случае задание может быть минимально-различимым элементом, которым ограничивается степень подробности исследования. Каждое задание включает одну или несколько программ, и выполнение этих программ, а также операции, производимые ими над определенными наборами данных, образуют другой уровень рассмотрения системы. При описании работы на каждом из этих уровней применяется понятие *заявки на использование ресурсов*, имеющее разную физическую сущность.

Заявкой для пакетного режима, исследуемого на уровне прикладных программ или уровне устройств, может быть задание или задача.

Заявка для режимов диалога и *запрос–ответ* – это сообщение, запрос, транзакция.

Заявкой для уровня, на котором исследуются конфликты в многопроцессорной вычислительной сети (ВС) при обращении к общим модулям оперативной памяти со стороны процессоров и

каналов ввода–вывода, будет единичное обращение к ОП (выбор команды либо операнда, запись слова и т.д.).

В формализованное описание взаимодействия ресурсов системы входит построение некоторого процесса обработки заявки ресурсами или иначе «информационного процесса, определяемого формальным образом в виде последовательности этапов передачи и обработки информации на средствах сети ЭВМ, инициируемой при реализации заявки абонента на выполнение запрашиваемых от сети информационно-вычислительных работ» [12].

> Совокупность взаимодействующих информационных процессов определяет функционирование системы, а оценка качества ее функционирования сводится к оценке совокупности характеристик процессов с учетом их взаимодействия.

В общем случае процесс обработки заявки состоит из этапов, на каждом из которых ей одновременно требуется один или несколько ресурсов. Для формального описания процессов используется понятие *трека процесса* – множества ресурсов, упорядоченного в соответствии с последовательностью этапов обработки заявки. Трек процесса может быть либо детерминированным, либо случайным.

Если *трек процесса детерминированный*, то задается непосредственно

$$TR = W_0, W_1, \ldots, W_i, \ldots, W_m,$$

где W_0 – источник заявки (процесса); W_i – *i*-й ресурс системы.

Если *трек процесса случайный*, то он задается как неупорядоченное множество, а конкретная последовательность этапов обработки задается с помощью матрицы вероятностей перехода $R = (r_{ij})$, $i, j = \overline{0, M}$, где r_{ij} – вероятность перехода заявки после окончания ее обработки ресурсом *i* на обработку ресурсом *j*.

Информация о ресурсах, содержащихся в треке процесса обработки заявки

- Параметры функций распределения длительности обслуживания заявки данным ресурсом (как правило, математическое ожидание и дисперсия) в изоляции от других заявок.

- Алгоритм функционирования (дисциплина обслуживания заявок или правила, определяющие порядок предоставления ресурса конкурирующим заявкам).
- Приоритет рассматриваемой заявки при обслуживании ее в данном ресурсе.

Если перечисленные компоненты треков процессов всех заявок определены (главная трудность обычно заключается в определении параметров функций распределения длительностей обслуживания заявок – для этого применяются измерения с помощью специальных программных и аппаратных мониторов, а также расчеты с помощью интерфейсных моделей), задача анализа сводится к расчету некоторой стохастической сети, рассматриваемой в 5.2.

Если же параметры времени обслуживания для некоторых ресурсов определить априори затруднительно, процесс обработки заявок этими ресурсами рассматривается более подробно, в частности, для каждого этапа обработки заявок таким ресурсом определяют множество ресурсов, используемых совместно с данным ресурсом (этот ресурс называют *составным*). Такое рассмотрение приводит к моделям многоуровневой структуры.

Анализ многоуровневых моделей, рассматриваемый в 5.3, основывается на концепции вложенности процессов и ресурсов.

Вложенным (в ресурс W_i уровня q) *ресурсом* называется ресурс, который занимается и освобождается заявкой на интервале ее обслуживания ресурсом W_i.

По аналогии с процессом уровня q теперь можно рассматривать вложенные процессы – процессы уровня $q+1$, инициируемые при выполнении этапа процесса уровня q (порождающего процесса). В общем случае, когда в системе имеются заявки разных типов, заявки каждого типа порождают в составных ресурсах вложенные процессы, отличные от остальных. При переходе с верхнего уровня на уровень вложенных процессов определяется новое множество типов заявок.

Если все процессы обслуживания заявок построенного следующего уровня являются определенными, т.е. для них имеются числовые значения параметров функций распределения времени обслуживания заявок всеми ресурсами этого уровня, то процеду-

ра построения формализованной схемы заканчивается. В противном случае производится дальнейшая детализация отдельных этапов процесса и построение вложенных процессов, определение нового множества типов заявок и т.д.

Удобная форма представления процессов обработки заявок – направленный граф, вершины которого соответствуют этапам обработки заявок ресурсами, а дуги – возможным переходам заявок от ресурса к ресурсу.

Для случайных треков дугам графа сопоставлены значения вероятностей переходов заявок от ресурса к ресурсу.

Для определенных процессов вершинам графа сопоставлены временные характеристики обслуживания заявок.

> Таким образом, построение формализованной схемы взаимодействия ресурсов, основанное на раздельном представлении процессов обработки заявок на отдельных уровнях, обеспечивает обозримость модели при сохранении необходимой степени ее разрешающей способности (степени детализации), а анализ, основанный декомпозиции, резко снижает затраты времени на его проведение.

5.2. Анализ стохастических сетей

Методы анализа характеристик производительности многоуровневых моделей КИС основаны на представлении процессов обслуживания на отдельных уровнях разомкнутыми, замкнутыми или смешанными стохастическим сетями.

Стохастическая сеть (СС) – совокупность систем массового обслуживания (СМО), в которой циркулируют заявки, переходящие из одной СМО в другую.

Сеть называют стохастической (случайной) потому, что маршруты заявок в общем случае носят вероятностный характер (сети с детерминированными маршрутами заявок представляют собой частный случай СС).

Структуру СС представляют в виде графа, вершины которого (узлы сети) соответствуют СМО, а дуги – пути перехода заявок из одной СМО в другую.

Разомкнутые стохастические сети

Рассмотрим стохастическую сеть, состоящую из M узлов.

Пусть i-й узел ($i \in \{\overline{1,M}\}$) содержит m_i одинаковых обслуживающих аппаратов (ОА) и общую очередь заявок. Время обслуживания каждого ОА имеет экспоненциальное распределение со средним значением $T_i = 1/\mu_i$. В дальнейшем можно будет отказаться от некоторых из введенных допущений и рассмотреть более общие ***случаи обслуживания***:

- С переменной интенсивностью, зависящей от длины очереди;
- С различными дисциплинами организации очередей к ОА;

и т.д.

Имеются M внешних пуассоновских источников заявок. Интенсивность внешнего потока заявок в i-й узел обозначим γ_i, $i = \overline{1,M}$. В общем случае интенсивность поступления заявок в сеть от i-го источника может зависеть от общего числа заявок в сети. После обслуживания в i-м узле заявка поступает в узел j с вероятностью r_{ij} или покидает сеть с вероятностью $\left(1 - \sum_{j=1}^{M} r_{ij}\right)$.

Для описания такой сети надо задать следующие исходные данные:

M – число узлов;

$m = (m_1, m_2, \ldots, m_M)$ – вектор числа ОА в узлах;

$\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_M)$ – вектор интенсивностей обслуживания в узлах;

$\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_M)$ – вектор интенсивностей внешних источников;

$\mathbf{R} = [r_{ij}]_{M \times M}$ – матрица вероятностей переходов.

Вместо вектора γ иногда задают общую интенсивность входного потока заявок в сеть $\boldsymbol{\Lambda}$ и вектор $\boldsymbol{q}^0 = (q_1^0, q_2^0, \ldots, q_M^0)$ – вектор вероятностей поступления входных заявок в каждый узел сети. Нетрудно видеть, что

$$\gamma_i = \mathbf{\Lambda} q_i^0,\ i = \overline{1,M}\ . \tag{5.1}$$

Состояние сети в момент времени *t* представляется вектором

$$\xi(t) = \left(\xi_1(t), \xi_2(t), \ldots, \xi_M(t)\right),$$

где $\xi_i(t)$ – число заявок в *i*-й СМО в момент времени *t*.

При сделанных допущениях $\xi(t)$ представляет собой марковский случайный процесс, являющийся многомерным аналогом процесса размножения/гибели. Для него можно представить граф переходов и стандартным образом записать систему линейных алгебраических уравнений относительно вероятностей, являющихся компонентами стационарного распределения.

В частном случае, когда интенсивность поступления в сеть постоянна (не зависит от числа заявок в сети), решение этой системы уравнений имеет вид произведения, в котором каждый сомножитель определяется характеристиками *i*-го узла сети:

$$p(n_1, n_2, \ldots, n_M) = \prod_{i=1}^{M} p_i(n_i)\,. \tag{5.2}$$

Этот результат известен под названием *теоремы Джексона*, утверждающей, что рассматриваемая сеть ведет себя как совокупность независимых СМО с пуассоновскими входными потоками интенсивности $\boldsymbol{\lambda_i}$ и интенсивностями обслуживания $\boldsymbol{\mu_i(n_i)}$.

В частности, когда *i*-й узел сети содержит m_i одинаковых ОА с общей очередью заявок, то

$$p_i(n_i) = \begin{cases} p_i(0)\dfrac{(m_i\rho_i)^{n_i}}{n_i!},\ n_i \le m_i; \\ p_i(0)\dfrac{\rho_i^{n_i} m_i^{m_i}}{m_i!},\ n_i > m_i, \end{cases} \tag{5.3}$$

где

$$\rho_i = \frac{\lambda_i}{m_i \mu_i}, \tag{5.4}$$

$$p_i(0)=\left[\sum_{l=0}^{m_i-1}\frac{(m_i\rho_i)^l}{l!}+\frac{(m_i\rho_i)^{m_i}}{m_i!(1-\rho_i)}\right]^{-1},\ i=\overline{1,M}\ . \tag{5.5}$$

Чтобы воспользоваться формулами (5.2)–(5.5), предварительно вычисляют значения λ_i. Для этого необходимо решить систему уравнений баланса интенсивностей:

$$\lambda_i=\gamma_i+\sum_{j=1}^{M}\lambda_j r_{ji},\ i=\overline{1,M}\ . \tag{5.6}$$

Зная стационарное распределение $P[n]$ и используя свойства производящих функций (см. 2.4), можно вычислить основные показатели сети – среднее число заявок и среднее время ответа в каждом узле сети.

Замкнутые стохастические сети

При замкнутых стохастических сетях в их описании отсутствуют внешние источники заявок. В сети циркулируют N заявок, переходя от узла к узлу. При этом $\sum_{j=1}^{N} r_{ij}=1,\ i=\overline{1,M},$ т.е. заявки не покидают сеть.

Интенсивность входных потоков в каждый узел

$$\lambda_i=\sum_{j=1}^{M}\lambda_j r_{ji},\ i=\overline{1,M}\ . \tag{5.7}$$

Эта система уравнений определяет λ_i не единственным образом (с точностью до множителя).

> Модель произвольной замкнутой марковской сети впервые была исследована Гордоном и Ньюэллом. Основной результат, полученный ими, – соотношение, аналогичное (5.2), также представляющее собой декомпозицию сети на отдельные узлы.

Для СС, в которой $m_i=1$ для $i=\overline{1,M}$, стационарная вероятность

$$p_{n_1,\, n_2,\, \ldots,\, n_M} = \frac{1}{G(N)} \prod_{i=1}^{M} x_i^{n_i}, \tag{5.8}$$

где $G(N)$ – постоянная, обеспечивающая равенство суммы вероятностей единице:

$$G(N) = \sum_{n \in A_N} \prod_{i=1}^{M} x_i^{n_i}, \tag{5.9}$$

где

$$\mathbf{n} = (n_1,\, n_2, \ldots,\, n_M); \quad A_N = \left\{ \mathbf{n} : n_i \geq 0,\, i = \overline{1, M},\ \sum_{i=1}^{N} n_i = N \right\};$$

$x_i = \lambda_i / \mu_i,\ i = \overline{1, M}$.

Для получения расчетных соотношений для среднего числа заявок и коэффициента использования (нагрузки) *i*-й СМО используется аппарат производящих функций, а для среднего времени ответа – формула Литтла.

> Формула Литтла [10] устанавливает связь между средним числом требований в системе обслуживания (или в очереди) $N_{\text{ср}}$ и средним временем пребывания в системе (или ожидания в очереди) W в виде $N_{\text{ср}} = \lambda W$. При этом на структуру системы обслуживания не накладываются никакие ограничения. Интуитивное доказательство формулы Литтла основано на том, что требование, входящее в систему, находит в ней то же среднее число требований, которое остается в системе, когда это требование покидает ее. Это число равно интенсивности поступления требований в систему, умноженной на время пребывания требования в системе.

Рассмотренный метод анализа, базирующийся на расчете компонентов стационарного распределения, является трудоемким и не может служить основой для диалоговых процедур взаимодействия исследователя с ЭВМ.

Для расчета средних характеристик замкнутой сети можно применять гораздо более простой метод, предложенный в 1979 г. Райзером и обобщенный в ряде последующих работ [13]. Этот метод известен под названием *прямого метода расчета средних*.

Идеи, лежащие в основе метода, рассмотрим на примере замкнутой сети из последовательно соединенных ОА, имеющих экспоненциальные распределения длительности обслуживания с постоянными интенсивностями обслуживания, равными $\mu_i = 1/T_i$, $i = \overline{1,M}$.

Пусть N – число заявок в сети. Тогда

$$t_{pi}(N) = \frac{1}{\mu_i} + \frac{1}{\mu_i}\overline{\nu}_i(N),\ i = \overline{1,M}\ ; \tag{5.10}$$

$$\mu^*(N) = \frac{N}{\sum_{i=1}^{M} t_{pi}(N)}; \tag{5.11}$$

$$\overline{N}_i(N) = \mu^*(N) t_{pi}(N),\ i = \overline{1,M}, \tag{5.12}$$

где $t_{pi}(N)$ – среднее время ответа (время реакции) i-й СМО сети, имеющей N заявок; $\nu_{\text{ср}i}(N)$ – среднее число заявок в i-й СМО в момент поступления в нее новой заявки в сети с N заявками; $\mu^*(N)$ – пропускная способность сети с N заявками; $\overline{N}_i(N)$ – среднее число заявок в i-й СМО в произвольный момент времени в сети с N заявками.

Соотношение (5.10) выражает тот факт, что при экспоненциальной длительности обслуживания заявки среднее время дообслуживания равно среднему времени обслуживания заявки, поэтому среднее время ответа складывается из среднего времени обслуживания вновь поступившей заявки и средней длительности обслуживания всех заявок, находившихся в i-й СМО в момент поступления в нее новой заявки.

Соотношения (5.11) и (5.12) вытекают из формулы Литтла, применяемой, соответственно, ко всей сети и к i-й СМО в отдельности.

Основой для расчета характеристик производительности служит **теорема Райзера**: *стационарные вероятности состояний замкнутой стохастической сети с N заявками в момент поступления заявки в i-й узел сети совпадают со стационарными вероятностями состояний этой же сети с* $(N-1)$ *заявкой для произвольного момента времени* [13].

Из теоремы следует, что

$$\nu_{\text{ср}i}(N) = \bar{N}_i(N-1),\ i=\overline{1,M}\ ; \tag{5.13}$$

$$t_{p\,i}(N) = \frac{1}{\mu_i} + \frac{1}{\mu_i}\overline{N_i}(N-1),\ i=\overline{1,M}\ . \tag{5.14}$$

Система уравнений (5.10)–(5.14) легко решается рекуррентно, начиная с числа заявок в сети, равного 0: $\bar{N}_i(0)=0$, $i=\overline{1,M}$ и последовательного увеличения его на 1 вплоть до N.

Результат обобщается на замкнутые сети произвольной структуры, задаваемой матрицей **R** вероятностей переходов.

Зафиксируем в сети узел с номером 1, обычно в качестве которого при формализации выбирают источник заявок.

Обозначим через e_i – среднее число посещений каждой заявкой *i*-й СМО (*i*-го узла сети) между двумя последовательными посещениями ею СМО с номером 1. Тогда

$$e_1 = 1,\ e_i = \sum_{j=1}^{M} e_j r_{ji},\ i=\overline{2,M}\ .$$

Так как среднее число посещений пропорционально пропускной способности, то e_i измеряет также пропускную способность *i*-й СМО μ_i^* в единицах пропускной способности μ_1^* СМО с номером 1, т.е.

$$\mu_i^* = e_i\mu_1^*,\ i=\overline{1,M}\ .$$

Если $t_{\text{p}i}$ – время реакции *i*-й СМО, то среднее время между двумя последовательными уходами из СМО с номером 1:

$$T_{\text{ц}} = \sum_{i=1}^{M} e_i t_{\text{p}i}\ .$$

Тогда аналогично уравнениям (5.10)–(5.14) можно записать

$$t_{\text{p}i}(N) = \frac{1}{\mu_i} + \frac{1}{\mu_i}\overline{N_i}(N-1); \tag{5.15}$$

$$\mu_1^*(N) = \frac{N}{\sum_{i=1}^{M} e_i t_{\text{p}i}(N)}; \tag{5.16}$$

$$\overline{N}_i(N) = \mu_1^*(N) e_i t_{\mathrm{p}i}(N),\ i = \overline{1, M}; \tag{5.17}$$

Полагая $\tilde{t}_{\mathrm{p}i} = e_i t_{\mathrm{p}i}$ и вводя переменные $\beta_i = e_i / \mu_i$, получим

$$\tilde{t}_{\mathrm{p}i}(N) = \beta_i \left[1 + \overline{N_i}(N-1)\right]; \tag{5.18}$$

$$\mu_1^*(N) = \frac{N}{\sum_{i=1}^{M} \tilde{t}_{\mathrm{p}i}(N)}; \tag{5.19}$$

$$\overline{N}_i(N) = \mu_1^*(N) \tilde{t}_{\mathrm{p}i}(N),\ i = \overline{1, M}\ . \tag{5.20}$$

Выражения (5.15)–(5.20) позволяют рекуррентно для числа заявок в сети последовательно от 0 до N, начиная с $\overline{N}_i(0) = 0$ вычислять характеристики сети.

Дальнейшее обобщение метода расчета связано с *разными типами узлов сети*. При этом для разных типов получаются разные соотношения, связывающие время реакции узла с числом заявок в нем. Эти соотношения имеют вид:

тип 0 – источник заявок:

$$t_{\mathrm{p}i}(N) = \frac{1}{\mu_i} = T_i$$

тип 1 – многоканальная СМО

$$t_{\mathrm{p}i}(N) = \begin{cases} T_i, & \overline{N}_i(N-1) \le k_i - 1; \\ \dfrac{T_i}{k_i}\left[k_i + \overline{N}_i(N-1)\right], & \overline{N}_i(N-1) > k_i - 1; \end{cases}$$

тип 2 – группа параллельных одноканальных СМО с равномерным распределением потока заявок по устройствам

$$t_{\mathrm{p}i}(N) = \frac{T_i}{k_i}\left[k_i + \overline{N}_i(N-1)\right];$$

тип 3 – группа последовательных одноканальных СМО

$$t_{\mathrm{p}i}(N) = T_i\left[k_i + \overline{N}_i(N-1)\right],$$

где k_i – емкость узла, т.е. число устройств в узле.

В результате расчета получаются следующие *характеристики сети*:

среднее время цикла обработки заявки

$$T_{\text{ц}} = \sum_{i=1}^{M} e_i t_{\text{р}\,i}(N);$$

время реакции системы

$$t_{\text{р}}(N) = T_{\text{ц}}(N) - T_{\text{и}},$$

где $T_{\text{и}}$ – время пребывания заявки в источнике;

пропускная способность

$$\mu^*(N) = \frac{N}{T_{\text{ц}}(N)}.$$

Для каждого *узла* вычисляются следующие *характеристики*:

загрузка каждого устройства

$$\rho_i = \frac{\mu^*(N) e_i T_i}{k_i};$$

среднее число заявок в узле

$$\bar{N}_i = \mu^*(N) e_i t_{\text{р}\,i}(N);$$

средняя длина очереди заявок к узлу

$$L_i(N) = \begin{cases} \bar{N}_i - k_i \rho_i & \text{для типа 1;} \\ \dfrac{\bar{N}_i}{k_i} - \rho_i & \text{для типов 2,3;} \end{cases}$$

время реакции узла $t_{\text{р}\,i}(N)$;

среднее время ожидания в узле $W_i(N) = t_{\text{р}\,i}(N) - T_i.$

Смешанные стохастические сети

Предыдущие модели представляют собой *однородные стохастические сети*, в которых все заявки имеют одинаковые вероятностные свойства: матрица перехода **R** одинакова для всех заявок, а распределение времени обслуживания в каждом узле

зависит лишь от номера этого узла и не зависит от поступающей заявки. Кроме того, в этих моделях используется допущение об экспоненциальном распределении длительности обслуживания в ОА каждого узла.

Более общая модель СС исследована в [14]. В сети имеются M узлов и допускается конечное число L различных классов заявок. Структура сети для каждого l-го класса заявок задается матрицей вероятностей перехода:

$\mathbf{R}(l) = r_{ij}(l)_{M\times M}$, $l = \overline{1, L}$.

Сеть может быть замкнута для заявок класса l_1 ($\sum_{j=1}^{M} r_{ij}(l_1) = 1$; $\gamma_i(l_1) = 0$, $i = \overline{1, M}$ и в сети имеются $N(l_1)$ заявок класса l_1) и разомкнута для заявок класса l_2. Допускаются четыре типа узлов.

Узел типа 1 имеет один обслуживающий аппарат с общей очередью заявок независимо от их класса; заявки обслуживаются в порядке поступления. Время обслуживания заявок – экспоненциальное распределение, одинаковое для всех классов заявок, но с параметром (интенсивностью обслуживания), который может быть функцией числа заявок в данном узле.

В остальных типах узлов распределение длительности обслуживания может зависеть от класса заявки.

Узел типа 2 содержит один ОА с произвольным распределением длительности обслуживания, имеющим рациональное преобразование Лапласа–Стилтьеса (ПЛС). Обслуживание заявок осуществляется в соответствии с алгоритмом разделения процессора или алгоритмом кругового опроса с квантом времени, стремящимся к нулю (рис. 5.1).

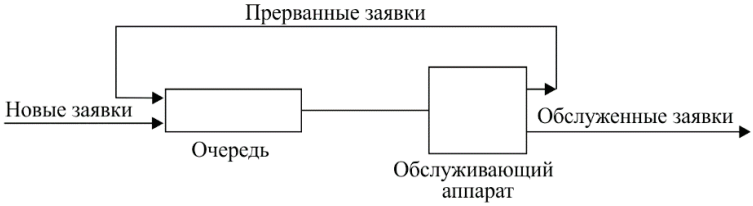


Рис. 5.1. Схема алгоритма кругового опроса

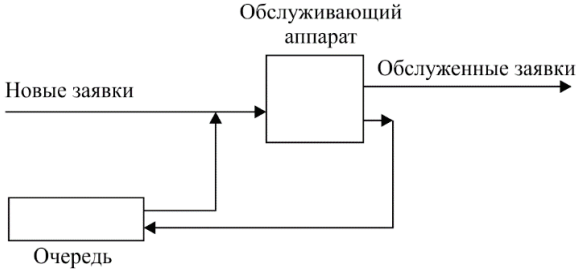


Рис. 5.2. Схема обслуживания в порядке, обратном поступлению заявок

Узел типа 3 имеет количество ОА, равное наибольшему числу заявок, которые могут одновременно поступать на обслуживание в данном узле. Распределения длительности обслуживания произвольны, но должны иметь рациональные ПЛС.

Узел типа 4 имеет единственный ОА с абсолютными приоритетами и дообслуживанием заявок и выбором заявок из очереди в порядке, обратном их поступлению (рис. 5.2). Распределение времени обслуживания произвольно, но имеет рациональное ПЛС.

Для определения интенсивности поступления заявок каждого класса в каждый узел решается совокупность систем уравнений:

$$e_i(l) = \gamma_i(l) + \sum_{j=1}^{M} e_j(l) r_{ji}(l),\ i = \overline{1,M},\ l = \overline{1,L}. \qquad (5.21)$$

Для классов требований, соответствующих замкнутой структуре сети, $\gamma_i(l) = 0,\ i = \overline{1,M}$, и в таком случае значение $e_i(l)$ равно относительной частоте посещений узла i заявками класса l (значение $e_i(l)/e_1(l)$ равно частоте посещения заявкой из класса l узла i между двумя последовательными посещениями ею узла 1). Для классов l заявок, соответствующих разомкнутой структуре, некоторые значения $\gamma_i(l)$ будут ненулевыми, и $e_i(l)$ в этом случае равно суммарной интенсивности поступления заявок класса l в узел i от внешних источников и других узлов.

Описание состояния системы носит сложный характер, указывающий на количество заявок каждого класса в каждом узле и этап достигнутого обслуживания.

Основной результат исследования сводится к тому, что стационарные вероятности состояний сети могут быть записаны в виде

$$P(\alpha_1, \alpha_2, \ \ldots, \alpha_M) = C\prod_{i=1}^{M} g_i(\alpha_i), \qquad (5.22)$$

где α_i – описание состояния узла i, зависящее от типа узла; C – нормирующая константа; g_i – функция, зависящая от типа узла i и его состояния.

Для достаточно общих сетей также справедлив принцип декомпозиции, позволяющий анализировать отдельно каждый узел сети. Использование метода укрупнения состояний марковской модели (см. 3.4) приводит к системе соотношений, показывающей, что стационарное распределение вероятностей состояния модели также имеет вид произведения.
Несмотря на произвольный характер функций распределения длительностей обслуживания в узлах, выходные параметры зависят только от значений ρ_i, определяемых лишь первыми моментами (средними значениями) длительностей обслуживания.

Итеративные методы анализа стохастической сети

Существование решения для стационарного распределения вероятностей состояния СС в мультипликативной форме значительно облегчает задачу определения характеристик сети.

Для записи решения в мультипликативной форме, достаточно, чтобы сеть удовлетворяла уравнению локального баланса [13].

Однако существует м***ножество различных видов СС с широко распространенными дисциплинами обслуживания заявок, не имеющие решения в мультипликативной форме***:

- Сети с приоритетными дисциплинами обслуживания в узлах;
- Сети с разного рода блокировками;
- Сети с не экспоненциальными узлами и обслуживанием заявок в порядке поступления;
- Замкнутые сети с одновременным занятием нескольких ресурсов одной заявкой.

В подобных случаях прибегают к приближенным методам анализа СС. Основой многих приближенных методов анализа СС общего вида является теорема Нортона [13], названная так по аналогии с известной теоремой в теории электрических цепей. Для каждого узла *i* исходной сети оставшуюся часть сети (дополнение D_i узла *i* по сети) заменяют экспоненциальным источником заявок конечной емкости, получая таким образом совокупность *M* двухузловых СМО. В каждой такой СМО первый узел совпадает с *i*-м узлом исходной сети ($i = \overline{1, M}$), а второй (агрегированный), являющийся эквивалентом оставшейся части сети, имеет экспоненциально распределенное время обслуживания с параметром $\mu_{D_i}(n)$, зависящим от числа заявок в нем.

Теорема Нортона: *стационарное распределение числа заявок в i-м узле исходной сети совпадает с соответствующим распределением эквивалентной сети, если параметр* $\boldsymbol{\mu_{D_i}(n)}$ *агрегированного узла равен интенсивности* $\lambda_i(n)$ *поступления заявок в i-й узел исходной сети, в которой* $1/\boldsymbol{\mu_i} = 0$.

Таким образом $\overline{\mu}_{D_i}(n) = \lambda_i(n)$ – интенсивность потока заявок через «закороченный» узел *i* (т.е. узел *i* с нулевым временем обслуживания).

Для СС, не удовлетворяющих уравнению локального баланса, невозможна точная замена дополнения любого узла эквивалентным экспоненциальным источником. Тогда при каждой итерации производится приближенная замена дополнения узла экспоненциальным источником из условия, что среднее время пребывания заявки в экспоненциальном источнике равно среднему времени пребывания заявки в дополнении узла *i*.

Такой подход применяется для анализа замкнутых неоднородных СС с приоритетами заявок и различными типами узлов и реализован, например, в виде комплекса прикладных программ «Перевал» [15]. Этот комплекс является методо-ориентированным и предназначен для оценки производительности многомашинных и многопроцессорных ВС с помощью аналитических моделей в виде замкнутых неоднородных СС с приоритетами. Исследованию производительности ВС с помощью комплекса «Перевал» предшествует формализация работы ВС, в результа-

те которой определяются: модель ВС в виде замкнутой СС; соответствие между множеством контролируемых параметров ВС и множеством входных параметров СС, позволяющее каждому значению контролируемых параметров ВС поставить в соответствие единственное значение вектора входных параметров СС (для установления этого соответствия в отдельных случаях приходится составлять специальные интерфейсные модели); соответствие между множеством выходных параметров СС и множеством параметров производительности ВС.

Сетевая модель ВС может включать до 20 узлов, 40 классов заявок и 160 заявок каждого класса.

Обслуживание очередей заявок в каждом узле

- В порядке поступления.
- С относительными приоритетами.
- С абсолютными приоритетами с дообслуживанием (для узлов типа 2 и 3 номер класса заявки является номером приоритета).
- Без ожидания (число 0А равно максимальному числу заявок в узле).
- С разделением процессора (круговой опрос с нулевым квантом времени).
- С квантованием времени (ненулевой квант времени).

Число ОА в узле в случае дисциплины с абсолютными приоритетами от 1 до 99, в остальных случаях –1.

Различие заявок

- Время обслуживания в узлах сети, которое имеет гамма-распределение, задаваемое двумя параметрами – среднее значение и квадрат коэффициента вариации.
- Матрицы вероятностей перехода.
- Количество заявок в сети.

В случае дисциплин разделения процессора и квантования времени заявки всех классов должны иметь одинаковые параметры гамма-распределений, в остальных случаях этого ограничения нет.

Структура стохастической сети

- Число и тип узлов.
- Матрица вероятностей переходов для заявок каждого класса.

Характеристики стохастической сети

- Коэффициенты использования сети заявками каждого класса и суммарные коэффициенты использования узлов (соответствуют загрузкам устройств ВС).
- Средние значения и дисперсии времени пребывания заявок каждого класса в узлах сети, времени отсутствия заявок в узлах (время отсутствия заявки в узле, представляющем терминал, соответствует времени реакции ВС на запросы с этого терминала), длина очереди.
- Коэффициенты корреляции времени пребывания заявок каждого класса в узлах сети (используются для оценки дисперсии пропускной способности ВС в пакетном режиме).

5.3. Основы построения многоуровневых моделей КИС

Модели, имеющие структуру стохастических сетей (типа СС), описывают достаточно общие случаи взаимодействия ресурсов: разные классы заявок, разнообразные дисциплины обслуживания и т.д.

Общий недостаток моделей типа СС – с их помощью затруднительно описывать ситуации, в которых для выполнения какого-либо действия требуется одновременное участие нескольких ресурсов системы. Если в модели представлена некоторая совокупность ресурсов, одновременно участвующих в обработке заявок, то их взаимодействие обычно отражается введением блокировок и логических условий, запрещающих поступление на обслуживание новой заявки, пока предыдущая не закончит обслуживание на данной совокупности ресурсов. Кроме того, что это затрудняет процедуру расчета, теряется обозримость и наглядность модели, следовательно, в значительной мере пропадает ее смысл для исследователя, состоящий том, что модель нужна, в общем, не столько для получения конкретной оценки, а для выяснения, что повлияло на значение этой оценки, т.е. в моделях прежде всего необходимо выявлять взаимосвязи параметров.

Единственное средство согласования сложности описания исследуемых объектов с возможностями восприятия и анализа их человеком, как и в проектировании любых сложных систем [5] – *блочно-иерархический подход*, предусматривающий структурирование описаний и расчленение представлений об объекте на иерархические уровни. В его основе лежит разделение описаний по степени детализации отображаемых свойств и характеристик объекта. Принцип иерархичности при этом означает структурирование представлений об исследуемых объектах по степени детальности описаний, а принцип декомпозиции – возможность раздельного анализа (расчета) каждого уровня и раздельного анализа внутри уровней.

Формализованное представление взаимодействия ресурсов ВС в виде многоуровневых моделей, с одной стороны, соответствует логике работы системы и имеет наглядную физическую интерпретацию, с одной стороны, отражая последовательность предоставлении ресурсов заданию, а с другой стороны, обеспечивая существенное сокращение времени разработки, отладки и расчета характеристик по сравнению с одноуровневым представлением, имеющим такую же разрешающую способность. Это свойство не зависит от того, какой аппарат используется для получения числовых оценок выходных параметров многоуровневых моделей (имитация или аналитические соотношения).

Классификация многоуровневых моделей

Признаки классификации многоуровневых моделей оценки характеристик производительности

- Детерминированность или стохастичность треков заявок.
- Однородность или неоднородность (с точки зрения характеристик обработки или требований к срочности) заявок.
- Разомкнутость, замкнутость или смешанность треков заявок.

Такая схема классификации показана на рис. 5.3. Любой путь в этой схеме представляет собой некоторый класс многоуровневых моделей. При этом *детерминированные модели* представляют собой подмножество стохастических, модели с однородными заявками – подмножество моделей с неоднородными заявками, а *смешанные* (разомкнуто-замкнутые) *модели* включают в себя разомкнутые и замкнутые модели как частные случаи.

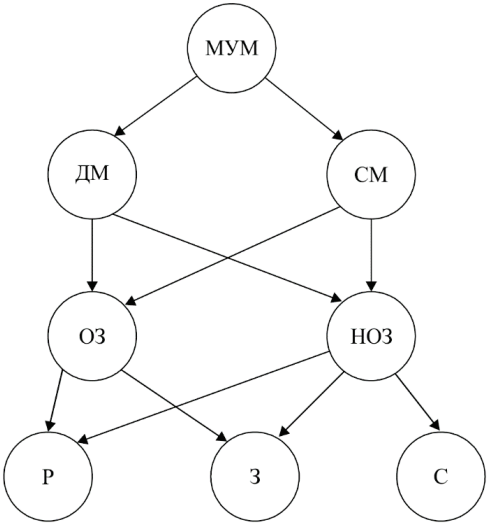


Рис. 5.3. Схема классификации многоуровневых моделей:
МУМ – многоуровневые, ДМ – детерминированные; СМ – стохастические;
ОЗ – с однородными заявками; НОЗ – с неоднородными заявками;
Р – разомкнутые; З – замкнутые; С – смешанные

Таким образом, *наиболее общий класс многоуровневых моделей – стохастические разомкнуто-замкнутые модели с неоднородными заявками*. Однако выделение отдельных классов из этого общего класса бывает полезно при проведении конкретного исследования, так как при этом можно существенно сократить описание и перечень задаваемых исходных данных по сравнению с общим способом описания, а также использовать более эффективную в вычислительном отношении схему расчета.

Формализованное описание многоуровневых моделей

Формализованное описание наиболее общего класса многоуровневых моделей – стохастических разомкнуто-замкнутых моделей с неоднородными заявками – довольно громоздко [16]. Выделим в нем основные компоненты, попутно отмечая упрощения, которые можно внести в описание моделей более узких классов. Общая ***схема*** такого ***описания*** состоит из трех этапов.

Этап 1. Описание верхнего (первого) уровня включает следующее: задание множества узлов первого уровня и множества классов заявок и их разбиение на два подмножества по признаку замкнутости или разомкнутости сети; задание интенсивностей внешних потоков заявок, для которых сеть разомкнута; приоритетов заявок на входах узлов первого уровня, структуры модели верхнего уровня (матриц переходов), характеристик ресурсов каждого уровня (в процессе этого обнаруживается наличие или отсутствие составных ресурсов); выделение множества составных ресурсов.

Этап 2. Если множество составных ресурсов не пустое, строятся вложенные процессы обработки, составляющие основу описания второго уровня. Второй уровень описывается так же, как первый.

Этап 3. Если множество составных ресурсов q-го уровня пустое, описание на этом заканчивается. Имеем q-уровневую модель.

Рассмотрим подробнее отдельные этапы формализованного описания модели. На верхнем уровне задаются множество классов заявок $L^{(1)} : l^{(1)} \in L^{(1)}$ и множество узлов (ресурсов) первого уровня, соответствующих отдельным этапам обслуживания заявок $R^{(1)} : i^{(1)} \in R^{(1)}$. Верхний индекс относится к номеру уровня модели.

Для смешанных моделей задается разбиение множества $L^{(1)}$ на подмножества $L_1^{(1)}$ и $L_2^{(1)}$ заявок, для которых модель соответственно замкнута или разомкнута: $L^{(1)} = L_1^{(1)} \cup L_2^{(1)}$, $L_1^{(1)} \cap L_2^{(1)} = \varnothing$. Для заявок $l \in L_1^{(1)}$ указывается их количество в модели для каждого класса.

Замкнутость модели относительно заявок некоторого класса либо означает, что источник этих заявок содержится внутри модели (в этом случае количество заявок определяется числом источников единичного объема), либо соответствует такому представлению работы системы (такому режиму), при котором заявки этого класса всегда имеются на входе системы, и в момент выхода заявки из системы в систему поступает новая заявка.

В формализованной схеме это отражается «зацикливанием» маршрута заявки и постоянной ее циркуляцией в системе. Таким способом можно задать некоторый, «наиболее тяжелый» режим работы системы или ее части.

Для заявок $l \in L_2^{(1)}$, поступающих в модель верхнего уровня извне, задаются интенсивности внешних потоков заявок каждого класса на входе каждого ресурса $\gamma_1(l)$, $i = \in R^{(1)}$. Как и в случае одноуровневых СС, общая интенсивность $\lambda_i(l)$ каждого потока заявок класса l на входе ресурса i складывается из внешнего потока заявок $\gamma_i(l)$ и потоков таких же заявок от остальных узлов (вторичных потоков).

Вместо явного задания интенсивностей внешнего потока на входе каждого ресурса может использоваться задание общей интенсивности источника заявок данного класса $\boldsymbol{\gamma}(l)$, $l \in L_2^{(1)}$ и вектора $\boldsymbol{q}(l)$ вероятностей поступления входных заявок класса l в каждый узел сети. В случае, когда заявки данного класса поступают в сеть через один входной узел, задание входного потока упрощается: указываются интенсивность внешнего потока $\boldsymbol{\gamma}(l)$ заявок класса l и номер входного узла.

Для заявок каждого класса указывается их приоритет при обработке ресурсами первого уровня модели. Заявка класса l может сохранять этот приоритет при обслуживании всеми ресурсами первого уровня, тогда приоритет задается либо в виде вектора $\boldsymbol{P} = \left(P(l)\right)$, $l \in L^{(1)}$, либо для каждого класса l заявок указывается их приоритет при обработке в каждом узле i (ресурсе), т.е. $\boldsymbol{P} = \left(P_i(l)\right)$, $i \in R^{(1)}$.

Структура модели верхнего уровня описывается совокупностью матриц перехода:

$$\boldsymbol{R}(l) = \left(r_{ij}(l)\right), l \in L^{(1)}, i, j \in \boldsymbol{R}^{(1)},$$

где $r_{ij}(l)$ – вероятность того, что заявка класса l после обработки ресурсом i попадает на обработку ресурсом j.

Задание матриц $\mathbf{R}(l)$ вместе с интенсивностями внешних потоков $\gamma_i(l)$ позволяет рассчитать потоки заявок каждого класса, для которых верхний уровень разомкнут, на входе любого ресурса верхнего уровня в соответствии с уравнением баланса интенсивностей:

$$\lambda_i(l) = \gamma_i(l) + \sum_{j \in R^{(1)}} \lambda_i(l) r_{ij}(l),\ i \in \mathbf{R}^{(1)}. \qquad (5.23)$$

Для *заявок классов* $l \in L^{(1)}_1$, *для которых верхний уровень замкнут*, уравнения (5.23), в которых $\gamma_i(l) = 0$, определяют $\lambda_i(l)$ не единственным образом (с точностью до постоянного множителя). Зафиксировав некоторый узел i^* и положив $\lambda_i(l) = 1$, можно решить систему (5.23) относительно $\lambda_i(l), i \in R^{(1)}, i \neq i^*$. Это решение определяет среднее число посещений каждой заявкой класса l каждого узла *i* между двумя последовательными посещениями ею узла i^* (см. §5.2).

Для *детерминированных моделей верхнего уровня, в которых маршруты (треки) заявок фиксированы*, матрицы вероятностей перехода содержат только нули и единицы. В этом случае вместо задания матриц перехода используется явное задание треков и для каждого класса заявок задается количество посещений каждого узла либо до момента выхода из системы, либо до момента возврата в узел-источник.

Задается характеристика каждого ресурса верхнего уровня, которая включает в себя параметры функции распределения длительности обслуживания заявки ресурса без учета конкурирующих заявок (обычно среднее значение и дисперсию, или коэффициент вариации) и число однотипных ресурсов в узле (объем ресурса). Кроме того, в узлах, где возникают очереди заявок, должны задаваться *дисциплина обслуживания*, т.е. общая очередь или раздельные очереди, порядок их просмотра, тип приоритетов и т.п.

Если параметры функции распределения длительности обслуживания заявки в узле неизвестны, необходимо попытаться

построить вспомогательную (интерфейсную) модель, позволяющую вычислить их на основе параметров, которые можно получить в результате измерений, из справочных данных, опыта эксплуатации систем-аналогов.

Если в ходе построения вспомогательной модели обнаруживается, что время обслуживания в узле связано с ожиданиями на входах каких-то других ресурсов, используемых совместно с данным (на интервале занятия и освобождения данного ресурса), то рассматриваемый ресурс относится к разряду составных ресурсов, и более детальное описание порождаемых им процессов обработки заявки производится на следующем уровне.

Таким образом, описание верхнего уровня завершается выделением из множества ресурсов подмножества составных ресурсов $RS^{(1)} \subseteq R^{(1)}$. Для этого подмножества строятся вложенные процессы обработки. Вложенный процесс в общем случае строится для каждого класса заявок l и каждого составного ресурса $i \in RS^{(1)}$. Это отражает то обстоятельство, что *вложенный процесс, инициированный составным ресурсом i, может иметь разные характеристики обработки ресурсами следующего уровня для разных классов заявок* (последовательность занятия ресурсов, временные характеристики).

Для построения вложенного процесса, порожденного (инициированного) этапом обслуживания заявки класса l ресурса $i \in RS^{(1)}$, выявляется, какие ресурсы следующего уровня и в какой последовательности используются на интервале занятия и освобождения ресурса i. Пара (l,i) представляет теперь заявку для модели следующего (второго уровня) и для нее строится трек обработки ресурсами следующего уровня. В силу того, что для разных пар (l,i) треки и их характеристики могут оказаться одинаковыми, заявки (l,i) объединяются в классы $l^{(2)} \in L^{(2)}$, где класс $l^{(2)}$ заявок второго уровня представляет собой совокупность пар (l,i), для которых вложенные процессы одинаковы.

Объединение треков вложенных процессов определит структуру модели второго уровня, которая описывается так же, как модель первого уровня, либо с помощь детерминированных тре-

ков, либо матрицами вероятностей перехода заявок каждого класса второго уровня от ресурса к ресурсу второго уровня.

Если среди ресурсов второго уровня имеются составные $RS^{(2)} \neq \varnothing$, то для этого подмножества ресурсов второго уровня (точнее, множества пар классов заявок составных ресурсов второго уровня) по такой же схеме строятся вложенные процессы третьего уровня.

Процедура построения многоуровневой модели завершается, когда среди ресурсов некоторого уровня Q нет составных ($RS^{(Q)} = \varnothing$), т.е. для всех ресурсов параметры функций распределения времени обслуживания заявок каждого класса определены.

5.4. Декомпозиция многоуровневых моделей КИС

Анализ многоуровневых моделей осуществляют на основе декомпозиции. *Элементами декомпозиции* (субмоделями в многоуровневом представлении) выступают уровни описания процессов обслуживания, для которых выполняется условие вложенности. Совокупность описаний процессов обслуживания одного уровня называют *уровнем вложенности* [16]. Связь между уровнями вложенности осуществляется через совокупность интерфейсных переменных. Состав этих переменных и количество информации, связывающей субмодели разных уровней, в каждой декомпозиционной схеме расчета может быть разным.

Один из наиболее простых подходов, обеспечивающий достаточную для инженерных расчетов точность и высокую вычислительную эффективность, состоит в том, что для каждой пары взаимодействующих уровней с верхнего уровня на нижний передаются пересчитанные значения интенсивностей потоков заявок, а с нижнего на верхний – значения моментов функций распределения времени пребывания на этом уровне, выступающие для верхнего уровня в качестве моментов функции распределения времени обслуживания заявки составным ресурсом.

Основой для такой декомпозиции является то, что как показали результаты экспериментальных исследований на имитаци-

онных и эталонных аналитических моделях, а также результаты натурных экспериментов, временные характеристики вложенных процессов слабо зависят от вторых и более высоких моментов функции распределения интервалов времени их инициализации, следовательно, – от организации процессов на более высоких уровнях. Поэтому *последовательность расчета интерфейсных переменных* такова:

1) от верхних уровней к нижним пересчитываются интенсивности потоков заявок;

2) от нижних уровней к верхним – временные характеристики, причем в силу свойства вложенности времена выполнения процессов на уровне $q+1$, являются временем обслуживания заявок ресурсами уровня q.

В случае, когда треки заявок верхнего уровня являются замкнутыми (режим диалога), интенсивность входных потоков от терминалов, через которые рассчитывается интенсивность входных потоков к ресурсам следующего уровня, зависит от временных характеристик обработки заявок:

$$\lambda_i = \frac{1}{M\left[t_{\text{обд}i}\right] + t_{\text{р}i}}.$$

В этом случае приходится решать систему уравнений, в которую в качестве неизвестных входят как время реакции, так и интенсивность. Если есть основания предполагать, что $t_{\text{р}i} \ll M\left[t_{\text{обд}i}\right]$, то для решения можно использовать метод простой итерации, когда на первом шаге итерации полагают $\lambda_i = \frac{1}{M\left[t_{\text{обд}i}\right]}$. Далее осуществляется проход по уровням сверху вниз и снизу вверх, а затем корректируются значения λ_i с учетом полученных значений времени реакции, и расчет повторяется.

Для решения также можно применять схему, построенную на основе *метода половинного деления.*

Таким образом, построение формализованной схемы взаимодействия ресурсов, основанное на раздельном представлении процессов обработки заявок на отдельных уровнях, обеспечивает обозримость модели при сохранении необходимой степени ее

разрешающей способности (степени детализации), а анализ, основанный на декомпозиции, резко снижает затраты времени на его проведение.

Последовательность анализа многоуровневой модели

Анализ многоуровневой модели осуществляется на основе декомпозиции, предусматривающей раздельный расчет характеристик каждого уровня. Такой расчет может происходить с использованием как аналитических, так имитационных средств, при этом количество информации, связывающей между собой модели разных уровней, в каждой конкретной декомпозиционной схеме может быт разным.

В частности, связь двух соседних уровней q и $q+1$, $q=\overline{1,Q-1}$ может осуществляться через параметр интенсивностей потоков заявок вложенных процессов уровня $q+1$, порождаемых в составных узлах уровня q (расчет сверху вниз), и параметры времени обслуживания (расчет снизу вверх).

Расчет интенсивности потока заявок, поступающих от составных ресурсов верхнего уровня в модель (сеть) следующего уровня, производится с учетом переопределения типов (классов) заявок при переходе с уровня на уровень.

Обозначим $L^{(q+1)}_{l}$ – множество пар $(l^{(q)},i)$ таких, что $l^{(q)} \in L^{(q)}$, $i \in RS^{(Q)}$ ($l^{(q)}$ – класс заявки уровня q; i – составной ресурс уровня q), и соответствующих одному фиксированному классу заявок $l^{(q+1)}$, $l^{(q+1)} \in L^{(q+1)}$, т.е. пар уровня q, образующих одинаковые треки на уровне $q+1$. Тогда для каждого класса заявок уровня $q+1$ интенсивность выходного потока заявок с уровня q

$$\gamma(l^{(q+1)}) = \sum_{(l^{(q)},i)\in Pl_{q+1}} \lambda_i\left(l^{(q)}\right), \tag{5.24}$$

где $\lambda_i\left(l^{(q)}\right)$ – интенсивность потока заявок класса $l^{(q)}$ на входе ресурса i уровня q.

Если указать узел (ресурс) в схеме ресурсов уровня $q+1$, в который попадает каждый внешний поток заявок с уровня q, то при известной интенсивности потоков заявок для верхнего уровня по (5.24) определяется интенсивность каждого суммарного внешнего потока заявок на следующий уровень.

Далее, на основании уравнений баланса интенсивностей, аналогичных (5.23), можно определить интенсивность потоков заявок каждого класса на входе каждого ресурса этого уровня.

При наличии замкнутых треков заявок на верхнем уровне расчет интенсивности производится итерационно.

Основной принцип, на котором строится процедура расчета параметров времени обслуживания заявок, выражается соотношением

$$B^{(q)}(t) = V^{(q+1)}(t), \tag{5.25}$$

где $B^{(q)}(t)$ – функция распределения длительности обслуживания заявок составным ресурсом уровня q; $V^{(q+1)}(t)$ – функция распределения времени выполнения соответствующего вложенного процесса на уровне $q+1$, т.е. функция распределения времени пребывания заявки, порожденной уровнем q на уровне $q+1$ (индексы, определяющие класс заявки и номер составного узла, порождающего вложенный процесс, опущены).

Алгоритм расчета разомкнутой модели верхнего уровня

Этап 1. Рассчитывается последовательно для $q = \overline{1,Q}$ интенсивность потоков заявок каждого класса на входе каждого ресурса на основе соотношений (5.23) и (5.24).

Этап 2. Определяются параметры функции распределения времени пребывания заявок каждого класса на уровне, начиная с $q = Q$. Для этого может использоваться либо имитационная модель уровня q, либо один из методов расчета СС (см. §5.2), либо приближенные методы, основанные на частичном укрупнении моделей, принципе эквивалентности потоков [см. (3.1–3.3)], либо декомпозиция одноуровневой модели на отдельные фазы и рас-

чет параметров времени пребывания заявки каждого класса на каждой фазе с помощью соотношений, приведенных в §5.2.

Этап 3. Подставляются рассчитанные параметры времени пребывания заявок на данном уровне вместо параметров времени обслуживания соответствующими ресурсами предыдущего, более высокого уровня, в соответствии с соотношением (5.25).

Этап 4. Этапы 2 и 3 повторяются до тех пор, пока не будут получены параметры времени пребывания заявок каждого класса на первом уровне.

ГЛАВА 6

ЗАДАЧИ И АЛГОРИТМЫ ИССЛЕДОВАНИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

6.1. Задача анализа чувствительности

При использовании математических моделей в исследовании КИС необходимо иметь в виду, что такие исследования всегда связаны:

1) с *методическими погрешностями* – упрощение моделей с целью получения алгоритмов, эффективных в вычислительном отношении;

2) с *погрешностями исходных данных* – невозможность точного определения статистических характеристик входных потоков, временных параметров решаемых в анализируемой системе задач, параметров траекторий заявок в стохастических сетях;

3) и т.п.

Поэтому важно определить, насколько сильно каждый вид погрешности влияет на точность оценки выходных параметров математической модели. Такая задача и называется *задачей анализа чувствительности выходных параметров математической модели к изменению входных параметров*. Формально она записывается так.

Пусть математическая модель представляет собой реализованную алгоритмически зависимость вектора V выходных пара-

метров модели от вектора X входных параметров (для простоты изложения в число компонентов вектора X включаем и *контролируемые факторы* – параметры, которые необходимо выбрать в результате анализа модели, и *неконтролируемые факторы* – параметры, которые исследователь должен учитывать, но не может изменять или выбирать), т.е. имеется алгоритм расчета вектор-функции $V(X)$.

Пусть $V=(V_1,V_2,\ldots,V_n)$; $X=(X_1,X_2,\ldots,X_m)$. Если все функции $V_i(X)$ $(i=1,2,\ldots,n)$ гладкие, т.е. имеют частные производные по X_j $(j=1,2,\ldots,m)$, то задача анализа чувствительности вектор-функции $V(X)$ в точке $X=X_{\text{ном}}$ состоит в расчете матрицы частных производных:

$$\mathbf{A}=\left[A_{ij}\right],$$

где $A_{ij}=\left.\dfrac{\partial V_i(X)}{\partial X_j}\right|_{X=X_{\text{ном}}}$, $i=1,2,\ldots,n$, $j=1,2,\ldots,m$.

Коэффициент A_{ij} называется *абсолютным коэффициентом влияния параметра X_j на параметр V_i*. Он показывает, примерно на сколько единиц, в которых рассчитывается параметр V_i, изменяется этот параметр при единичном изменении параметра X_j. При этом параметр X_j измеряется в своих единицах.

> Слово «примерно» связано с тем, что реально рассчитывается не частная производная A_{ij}, а лишь ее оценка, как разность приращения функции $V_i(X)$ в точке $X_{\text{ном}}$ при небольшом изменении аргумента X_j в этой точке, деленная на это изменение аргумента X_j.

В связи с разнобоями в единицах параметров наряду с расчетом *матрицы абсолютных коэффициентов влияния* обычно рассчитывают и *матрицу относительных* (*безразмерных*) *коэффициентов влияния:*

$$\mathbf{B} = \left[B_{ij} \right], \; B_{ij} = A_{ij} \frac{X_{j\text{ном}}}{V_i\left(X_{\text{ном}}\right)}.$$

При этом относительный коэффициент влияния B_{ij} показывает, на сколько процентов изменится выходной параметр V_i по отношению к $V_i(X)_{\text{ном}}$ при изменении параметра X_j по отношению к $X_{j\text{ном}}$ на 1%.

В случае, если некоторые входные параметры являются дискретными (например, число процессоров, каналов) или отсутствует аналитическая связь некоторых выходных параметров модели с некоторыми входными параметрами (хотя алгоритмы расчета вектор функции $V\left(X\right)$ программно реализованы), то производится такая же численная оценка абсолютных и относительных коэффициентов влияния, где вместо расчета частных производных в качестве абсолютных коэффициентов влияния A_{ij} берут отношения приращений выходных параметров V_i в точке $X_{\text{ном}}$ при последовательном малом приращении каждого входного параметра X_j к величине этих приращений.

Иными словами, рассчитывают значения всех *n* выходных параметров V_i в точке $X_{\text{ном}}$, затем дают приращение δX_j входному параметру X_j в точке номинального значения $X_{\text{ном}}$ и рассчитывают для каждого $i = \overline{1,n}$ разность

$$\delta V_i = V_i\left(X_{1\text{ном}}, X_{2\text{ном}}, \ldots, X_{j-1\text{ном}}, X_{j\text{ном}} + \right.$$
$$\left. + \delta X_j, X_{j+1\text{ном}}, \ldots, X_{m\text{ном}}\right) - V_i(X)_{\text{ном}}$$

и отношение

$$A_{ij} = \delta V_i / \delta X_j .$$

Процесс повторяют для каждого $j = \overline{1,m}$.

На самом деле процедура анализа чувствительности имеет более широкий спектр применения, чем оценка возможных погрешностей расчета выходных параметров модели от погрешностей входных параметров.

6.2. Алгоритмы комплексного анализа методических погрешностей математических моделей КИС

Рассмотрим примеры, в которых проводится исследование методических погрешностей математических моделей КИС, а также оценка коэффициентов влияния (анализ чувствительности).

Пример 6.1. Разработать математическую модель анализа ВС, содержащей один процессор и несколько каналов, с фиксированным числом задач, позволяющую оценивать влияние второго момента длительности решения задачи в процессоре на среднюю производительность системы при условии, что коэффициент вариации длительности решения задачи в процессоре $\nu<1$. Анализ проводить в предположении, что время обмена данными посредством канала – случайная величина, распределенная экспоненциально.

Структура модели показана на рис. 6.1.

Пусть T и m_2 – соответственно математическое ожидание и второй момент длительности решения задачи в процессоре, математическое ожидание времени обмена, осуществляемого посредством канала $M\left[t_{\text{обм}}\right]=1/\alpha$, N – число каналов, M – число задач в системе ($N<M$). Поскольку коэффициент вариации длительности решения задачи $\nu<1$, то распределение длительности решения задачи в процессоре можем аппроксимировать распределением Эрланга k-го порядка с параметрами $\mu=1/T$ и $k=T^2/\left(m_2-T^2\right)$.

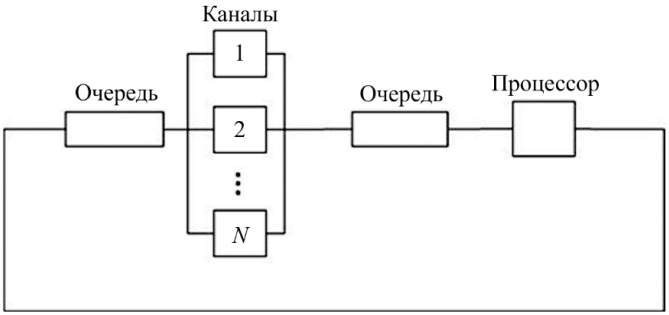


Рис. 6.1. Структура модели

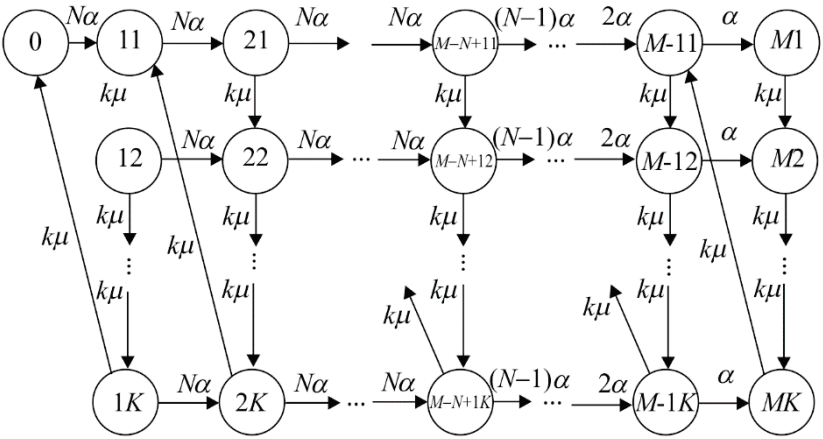


Рис 6.2. Граф переходов системы

Для анализа воспользуемся методом Эрланга, в соответствии с которым решение задачи в процессоре представим состоящим из K псевдофаз. Длительность каждой псевдофазы имеет экспоненциальное распределение со средним значением T / K. В качестве состояния системы в момент времени t возьмем вектор $\xi(t) = (\xi_1(t), \xi_2(t))$, где $\xi_1(t)$ – число задач в процессоре и очереди к процессору, $\xi_2(t)$ – номер псевдофазы, на которой находится задача в процессоре. Число задач на фазе каналов в состоянии $\xi(t)$ равно $M - \xi_1(t)$. Граф переходов системы изображен на рис. 6.2.

Для решения системы уравнений относительно стационарных вероятностей P_{ij}, $i = \overline{0, M}$, $j = \overline{1, K}$ необходимо использовать ЭВМ. Для этого надо написать программу формирования коэффициентов системы уравнений, отвечающих изображенному на рис. 6.2. графу переходов, и использовать одну из стандартных подпрограмм решения системы линейных алгебраических уравнений. Далее следует просчитать среднюю производительность системы $\bar{\mu}$ по формуле

$$\bar{\mu} = k\mu \sum_{i=1}^{M} P_{ik} \ . \qquad (6.1)$$

Изменяя значения m_2 от $T^2(\nu=0)$ до $2T^2(\nu=1)$ с некоторым шагом, решая каждый раз систему уравнений относительно P_{ij} и вычисляя $\bar{\mu}$ по формуле (6.1), можно построить искомую зависимость $\bar{\mu}=\delta(m_2)$.

Пример 6.2. Разработать алгоритм анализа чувствительности погрешности оценки средней производительности многопроцессорной ВС с фиксированным числом задач и ненадежными процессорами и каналами, вызванной использованием метода укрупнения состояний системы. Чувствительность погрешности оценивать от числа каналов, числа процессоров, числа задач, коэффициента готовности канала, коэффициента готовности процессора.

Разработку алгоритма будем проводить методом «сверху вниз».

В структуре алгоритма можно выделить три уровня. На верхнем уровне находится алгоритм анализа чувствительности, осуществляющий расчет коэффициентов влияния параметров, указанных в условиях примера, на погрешность оценки средней производительности. Алгоритм верхнего уровня осуществляет обращение к алгоритму второго уровня – расчету погрешности. В свою очередь, алгоритм второго уровня – расчет погрешности – обращается к алгоритмам третьего уровня – подпрограммам расчета средней производительности ВС точным и приближенным методом.

Задача анализа чувствительности вектор-функции $\mathbf{V}=\mathbf{V}(\mathbf{X})$, где $\mathbf{V}=(V_1,\ldots,V_m)$, $\mathbf{X}=(x_1,\ldots,x_n)$ в точке $\mathbf{x}^{\text{ном}}$ состоит в расчете матрицы $\mathbf{A}(a_{ij})$, $i=\overline{1,m}$, $j=\overline{1,n}$, где a_{ij} представляет собой коэффициент влияния параметра x_j на параметр V_j в точке $\mathbf{x}^{\text{ном}}$. Если функция V_j имеет частную производную по x_j, то

$$a_{ij}=\frac{\partial V_j\left(\mathbf{x}^{\text{ном}}\right)}{\partial x_j}. \tag{6.2}$$

Если же параметр x_j дискретный, то в ряде случаев, например, когда x_j принимает последовательные целочисленные значения, в качестве коэффициента влияния x_j на V_j можно брать

$$a_{ij}=\frac{\Delta_j V_j\left(x^{\text{ном}}\right)}{\Delta x_j}, \tag{6.3}$$

где $\Delta_j V_j\left(x^{\text{ном}}\right)=V_j\left(x_1^{\text{ном}},\ldots,x_{j-1}^{\text{ном}},x_j^{\text{ном}}+\Delta x_j,x_{j+1}^{\text{ном}},\ldots,x_n^{\text{ном}}\right)-V_j\left(x^{\text{ном}}\right)$.

При этом часто берут $\Delta x_j=1$. В других же случаях дискретности параметров x_j понятие коэффициента влияния теряет смысл. Для единообразия записи далее будем использовать обозначение (6.2), имея в виду, что если параметр x_j дискретный, то расчет ведется в соответствии с (6.3).

В данном случае в результате анализа чувствительности должны быть получены значения $\partial\delta/\partial N_{\text{к}}$, $\partial\delta/\partial N_{\text{п}}$, $\partial\delta/\partial z$, $\partial\delta/\partial K_{\text{г}_{\text{п}}}$, $\partial\delta/\partial K_{\text{г}_{\text{к}}}$, вычисленные в некоторой точке ($N_{\text{к}}$, $N_{\text{п}}$, z, $\alpha_{\text{к}}$, $\beta_{\text{к}}$, $\alpha_{\text{п}}$, $\beta_{\text{п}}$, $\mu_{\text{к}}$, $\mu_{\text{п}}$) пространства параметров. Здесь δ – погрешность оценки производительности; $N_{\text{к}}$ – число каналов; $N_{\text{п}}$ – число процессоров; z – число задач в системе; $\alpha_{\text{к}}$ – интенсивность отказов канала; $\beta_{\text{к}}$ – интенсивность восстановления канала; $\alpha_{\text{п}}$ – интенсивность отказов процессоров; $\beta_{\text{п}}$ – интенсивность восстановления процессора; $\mu_{\text{к}}$ – интенсивность обмена; осуществляемого посредством канала; $\mu_{\text{п}}$ – интенсивность решения задачи в процессоре.

Объединим перечисленные коэффициенты влияния в массив **A**, а параметры, от которых они зависят, в два массива, **XV** и **XP**, содержащие соответственно варьируемые и постоянные параметры.

Пусть массив **H** содержит значения приращений варьируемых параметров. Тогда алгоритм верхнего уровня может иметь вид, представленный на рис. 6.3.

Алгоритм второго уровня реализует формулу

$$\delta=\frac{\bar{\mu}^*-\bar{\mu}}{\bar{\mu}}, \tag{6.4}$$

где $\bar{\mu}$ – средняя производительность ВС, рассчитанная точным методом; $\bar{\mu}^*$ – средняя производительность, полученная способом укрупнения состояний.

Алгоритм второго уровня осуществляет обращение к двум подпрограммам, реализующим алгоритмы третьего уровня и вычисляющим значения $\bar{\mu}$ и $\bar{\mu}^*$.

В основе алгоритмов третьего уровня лежит модель системы, структура которой показана на рис. 6.4.

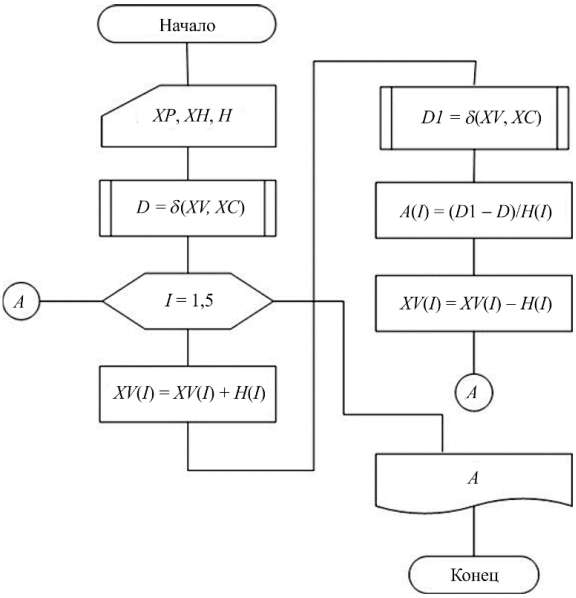


Рис. 6.3. Схема алгоритма анализа чувствительности верхнего уровня

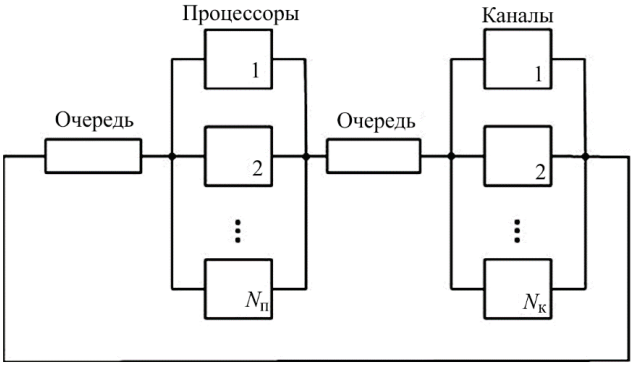


Рис. 6.4. Схема модели системы, на основе которой осуществляется расчет алгоритма третьего уровня

Состояние системы в момент времени t можно описать вектором $\xi(t)=(\xi_1(t),\xi_2(t),\xi_3(t))$, где $\xi_1(t)$ – число задач на процессорной фазе; $\xi_2(t)$ – число исправных процессоров; $\xi_3(t)$ – число исправных каналов в момент времени t.

Допустим, восстановление каналов и процессоров осуществляется двумя отдельными бригадами, одна из которых восстанавливает каналы, а другая – процессоры. Тогда интенсивность перехода из состояния ijk в соседние выражается формулами:

$$\begin{aligned}
&\lambda_{(ijk)(i+1jk)}=\mu_{\text{к}}\min\{z-i,k\};\\
&\lambda_{(ijk)(i-1jk)}=\mu_{\text{п}}\min\{i,j\};\\
&\lambda_{(ijk)(ijk-1)}=\alpha_{\text{к}}k;\\
&\lambda_{(ijk)(ijk+1)}=\beta_{\text{к}};\\
&\lambda_{(ijk)(ij-1k)}=\alpha_{\text{пj}};\\
&\lambda_{(ijk)(ij+1k)}=\beta_{\text{п}}.
\end{aligned} \qquad (6.5)$$

С учетом формул (6.5) нетрудно написать систему уравнений относительно компонент стационарного распределения P_{ijk}, $i=\overline{0,z}$, $j=\overline{0,N_{\text{п}}}$, $k=\overline{0,N_{\text{к}}}$.

Алгоритм третьего уровня, получающий $\overline{\mu}$, включает в себя блок формирования матрицы коэффициентов системы линейных алгебраических уравнений (СЛАУ) относительно P_{ijk}, обращение к стандартной подпрограмме решения СЛАУ и расчет по формуле

$$\overline{\mu}=\sum_{i=0}^{z}\sum_{j=0}^{N_{\text{п}}}\sum_{k=0}^{N_{\text{к}}}\mu_{\text{п}}\min\{i,j\}P_{ijk}.$$

Приближенный способ расчета средней производительности ВС, основанный на укрупнении состояний системы, состоит в следующем.

Объединяя в одно макросостояние j все состояния, у которых совпадают значения второй компоненты вектора $\xi(t)$, получим граф макросостояний (рис. 6.5).

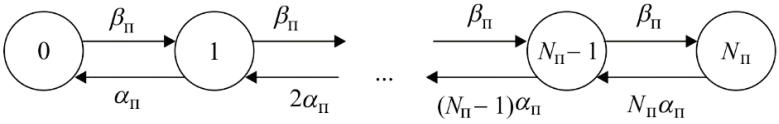


Рис. 6.5. Граф макросостояний
(объединение по второй компоненте)

Обозначим вероятность этих макросостояний через π_j, $j=\overline{0,N_{\text{п}}}$. Тогда

$$\pi_0=\left(1+\frac{\beta_{\text{п}}}{\alpha_{\text{п}}}+\frac{\beta_{\text{п}}{}^2}{2\alpha_{\text{п}}{}^2}+\ldots+\frac{1}{N_{\text{п}}!}\left(\frac{\beta_{\text{п}}}{\alpha_{\text{п}}}\right)^{N_{\text{п}}}\right)^{-1};$$

$$\pi_1=\frac{\beta_{\text{п}}}{\alpha_{\text{п}}}\pi_0; \qquad (6.6)$$

… … … … … … … … … … … … … … … …

$$\pi_{N_{\text{п}}}=\frac{1}{N_{\text{п}}!}\left(\frac{\beta_{\text{п}}}{\alpha_{\text{п}}}\right)^{N_{\text{п}}}\pi_0.$$

Объединяя в одно макросостояние все состояния графа, у которых совпадают значения третьей компоненты вектора $\xi(t)$, получим граф аналогичной структуры, у которого стационарная вероятность макросостояний r_k, $k=\overline{0,N_{\text{к}}}$ выражается соотношениями

$$r_0=\left(1+\frac{\beta_{\text{к}}}{\alpha_{\text{к}}}+\frac{\beta_{\text{к}}{}^2}{2\alpha_{\text{к}}{}^2}+\ldots+\frac{1}{N_{\text{к}}!}\left(\frac{\beta_{\text{к}}}{\alpha_{\text{к}}}\right)^{N_{\text{к}}}\right)^{-1};$$

$$r_1=\frac{\beta_{\text{к}}}{\alpha_{\text{к}}}r_0; \qquad (6.7)$$

… … … … … … … … … … … … … … … …

$$r_{N_{\text{к}}}=\frac{1}{N_{\text{к}}!}\left(\frac{\beta_{\text{к}}}{\alpha_{\text{к}}}\right)^{N_{\text{к}}}r_0.$$

Объединяя теперь в одно макросостояние все состояния, у которых совпадают значения первой компоненты вектора $\xi(t)$, получим граф макросостояний, изображенный на рис. 6.6.

Обозначим вероятности этих макросостояний через p_i, $i=\overline{1,z}$.

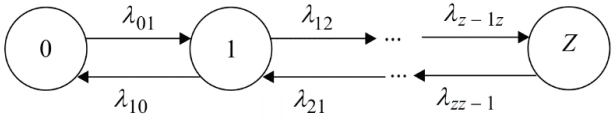


Рис. 6.6. Граф макросостояний
(объединение по первой компоненте)

Интенсивность переходов на графе (рис. 6.6) выражается соотношениями:

$$\lambda_{01}=\begin{cases}\mu_{\text{к}}\left(r_1+2r_2+\ldots+N_{\text{к}}r_{N_{\text{кК}}}\right), & \text{если} \quad z\geq N_{\text{к}},\\ \mu_{\text{к}}\left(r_1+2r_2+\ldots+z\left(r_z+r_{z+1}+\ldots+r_{N_{\text{к}}}\right)\right), & \text{если} \quad z<N_{\text{к}};\end{cases}$$

$$\lambda_{12}=\begin{cases}\mu_{\text{к}}\left(r_1+2r_2+\ldots+N_{\text{к}}r_{N_{\text{к}}}\right), & \text{если} \quad z-1\geq N_{\text{к}},\\ \mu_{\text{к}}\left(r_1+2r_2+\ldots+(z-1)\left(r_{z-1}+r_z+\ldots+r_{N_{\text{к}}}\right)\right), & \text{если} \quad z-1<N_{\text{к}};\end{cases}$$

… …

$$\lambda_{z-1\,z}=\mu_{\text{к}}\left(1-r_0\right);$$

$$\lambda_{10}=\mu_{\text{п}}\left(1-\pi_0\right); \qquad (6.8)$$

$$\lambda_{21}=\mu_{\text{п}}\left(\pi_1+2\left(\pi_2+\pi_3+\ldots+\pi_{N_{\text{п}}}\right)\right);$$

… …

$$\lambda_{z\,z-1}=\begin{cases}\mu_{\text{п}}\left(\pi_1+2\pi_2+\ldots+N_{\text{п}}\pi_{N_{\text{п}}}\right), & z\geq N_{\text{п}},\\ \mu_{\text{п}}\left(\pi_1+2\pi_2+\ldots+z\left(\pi_z+\pi_{z+1}+\ldots+\pi_{N_{\text{п}}}\right)\right), & z<N_{\text{п}}.\end{cases}$$

Соотношения для стационарных вероятностей макросостояний имеют вид:

$$p_0=\left(1+\frac{\lambda_{01}}{\lambda_{10}}+\frac{\lambda_{01}\lambda_{12}}{\lambda_{10}\lambda_{21}}+\ldots+\prod_{i=0}^{z-1}\frac{\lambda_{i\,i+1}}{\lambda_{i+1\,i}}\right)^{-1};$$

$$p_1=\frac{\lambda_{01}}{\lambda_{10}}p_0;$$

… (6.9)

$$p_z=\prod_{i=0}^{z-1}\frac{\lambda_{i\,i+1}}{\lambda_{i+1\,i}}p_0.$$

Средняя производительность системы вычисляется по формуле

$$\overline{\mu}^{*} = \sum_{i=1}^{z} p_i \lambda_{i\,i-1}\,. \qquad (6.10)$$

Алгоритм третьего уровня, осуществляющий приближенный расчет средней производительности, включает в себя расчет по формулам (6.6)–(6.10).

Контрольные задачи

Задача 6.1. Разработать математическую модель анализа многопультовой ВС, содержащей одну ЭВМ, позволяющую оценивать влияние второго момента длительности решения задачи на среднее время реакции системы (при условии, что коэффициент вариации $\nu < 1$).

Задача 6.2. Разработать алгоритм исследования погрешности системы частично укрупненных моделей при анализе времени реакции на запрос в многопультовой системе, содержащей пульты, процессоры и каналы.

Задача 6.3. Разработать алгоритм анализа чувствительности погрешности оценки времени реакции системы на запрос.

Погрешность вызвана использованием частично укрупненных моделей от числа пультов, числа процессоров, числа каналов. Анализ чувствительности проводится в точке, где фиксировано число и производительность каналов и процессоров, число пультов и среднее время обдумывания задачи пользователем.

Задача 6.4. Разработать алгоритм исследования погрешности алгоритма анализа чувствительности, вызванной использованием частично укрупненных моделей.

Система содержит пульты, процессоры и каналы. Анализ чувствительности состоит в оценке коэффициентов влияния числа пультов, числа процессоров, числа каналов на время реакции системы на запрос пользователя.

Задача 6.5. Разработать алгоритм исследования влияния числа пользователей на время реакции на запрос в многопроцессорной ВС коллективного пользования (состав ВС: терминалы, процессоры, селекторные каналы, ОП, НМД). Для анализа использовать частично укрупненные модели.

Задача 6.6. Разработать алгоритм исследования влияния ограниченной надежности селекторных каналов на время реакции

системы на запрос в ВС коллективного пользования, содержащей пульты, процессоры и каналы. Использовать методы укрупнения моделей.

Задача 6.7. Разработать алгоритм исследования совместного влияния ограниченной надежности каналов и процессоров на среднюю производительность и время реакции системы на запрос в ВС коллективного пользования, содержащей пульты, процессоры и каналы. Использовать методы укрупнения моделей.

ПРИЛОЖЕНИЕ

ПРИМЕРЫ РЕАЛИЗАЦИИ АЛГОРИТМОВ АНАЛИЗА СТОХАСТИЧЕСКИХ МОДЕЛЕЙ НА ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРАХ

Общие рекомендации

Предлагаются возможные варианты реализации некоторых описанных выше методов анализа стохастических моделей на персональных компьютерах с использованием программных средств общего назначения (Microsoft office 2010). Расчеты осуществляются на основе приложений Excel.

Непосредственная реализация в виде файла Excel, содержащего ряд примеров, имеется на электронном носителе (флэшкарте), позволяющая читателю открыть его с помощью Excel и провести в нем собственное редактирование, ввести изменения и дополнения в каждом представленном примере. Книга Excel имеет имя «Приложение 13.xl». На ее отдельных листах представлены алгоритмы реализации примеров, рассмотренных в предыдущих главах.

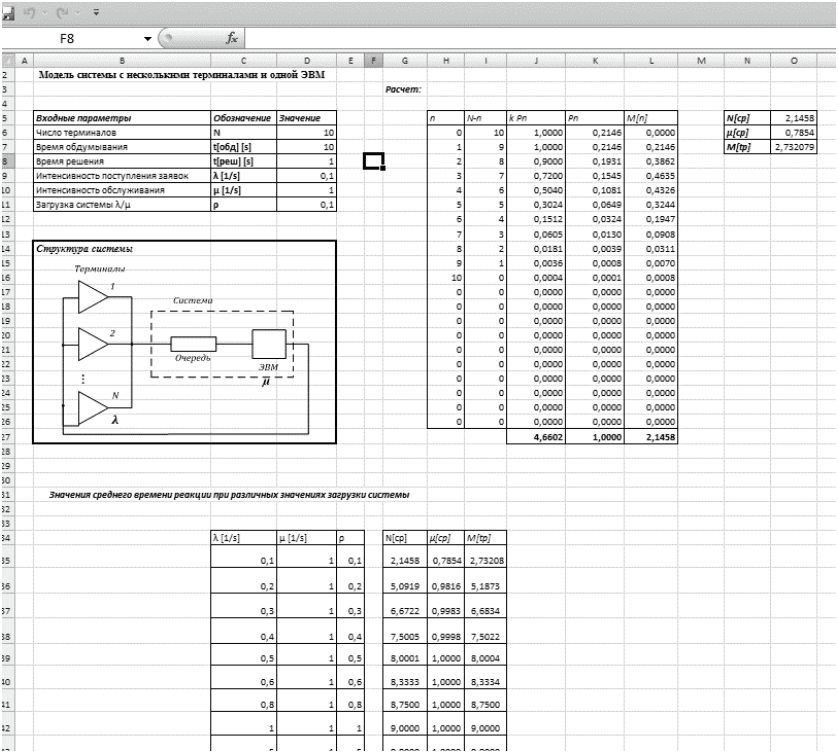
Приведенные примеры следует рассматривать лишь как простейшие образцы, не претендующие на оптимальность реализации. Предполагается, что знакомство с ними позволит дополнить проведенный анализ, проведя проверку «правдоподобия поведения модели» при изменении ее входных параметров. Такая проверка одновременно является средством тестирования, обнаружения, локализации и устранения возможных ошибок выполненной реализации алгоритмов.

После выполнения этой процедуры далее проводится многовариантный анализ с формированием таблиц и построением на их

основе графиков зависимостей выходных параметров моделей от входных параметров.

Эта работа также стимулирует читателя к разработке собственных вариантов реализации алгоритмов, возможно, с использованием других программных продуктов.

Пример 2.8. Простейшая реализация многовариантного анализа модели, описанной в §2.5 (лист «Пример 2.8»), средствами Excel представлена на рис. П1.



Модель системы с несколькими терминалами и одной ЭВМ

Расчет:

Входные параметры	Обозначение	Значение
Число терминалов	N	10
Время обдумывания	t[обд] [s]	10
Время решения	t[реш] [s]	1
Интенсивность поступления заявок	λ [1/s]	0,1
Интенсивность обслуживания	μ [1/s]	1
Загрузка системы λ/μ	ρ	0,1

Структура системы

n	N-n	k·Pn	Pn	M[n]
0	10	1,0000	0,2146	0,0000
1	9	1,0000	0,2146	0,2146
2	8	0,9000	0,1931	0,3862
3	7	0,7200	0,1545	0,4635
4	6	0,5040	0,1081	0,4326
5	5	0,3024	0,0649	0,3244
6	4	0,1512	0,0324	0,1947
7	3	0,0605	0,0130	0,0908
8	2	0,0181	0,0039	0,0311
9	1	0,0036	0,0008	0,0070
10	0	0,0004	0,0001	0,0008
0	0	0,0000	0,0000	0,0000
0	0	0,0000	0,0000	0,0000
0	0	0,0000	0,0000	0,0000
0	0	0,0000	0,0000	0,0000
0	0	0,0000	0,0000	0,0000
0	0	0,0000	0,0000	0,0000
0	0	0,0000	0,0000	0,0000
0	0	0,0000	0,0000	0,0000
0	0	0,0000	0,0000	0,0000
0	0	0,0000	0,0000	0,0000
		4,6602	1,0000	2,1458

N[ср]	2,1458
μ[ср]	0,7854
M[tp]	2,732079

Значения среднего времени реакции при различных значениях загрузки системы

λ [1/s]	μ [1/s]	ρ	N[ср]	μ[ср]	M[tp]
0,1	1	0,1	2,1458	0,7854	2,73208
0,2	1	0,2	5,0919	0,9816	5,1873
0,3	1	0,3	6,6722	0,9983	6,6834
0,4	1	0,4	7,5005	0,9998	7,5022
0,5	1	0,5	8,0001	1,0000	8,0004
0,6	1	0,6	8,3333	1,0000	8,3334
0,8	1	0,8	8,7500	1,0000	8,7500
1	1	1	9,0000	1,0000	9,0000

Рис. П1

Проанализировав ее, можно легко провести проверку на правдоподобие правильности расчета математического ожидания времени реакции (задавая граничные значения входных парамет-

ров, например, среднего времени решения от 0,01 с до 100 с, при сохранении остальных значений), а также построить графики зависимости математического ожидания числа требований в системе, времени реакции и эффективной производительности от загрузки системы.

Основные расчеты параметров модели по (2.42) произведены в центральной таблице (ячейки H5–L27). Первый и второй столбцы таблицы содержат числа заявок в системе; столбец *kPn* – рассчитанные коэффициенты перед $P_i, i=\overline{1,N}$ в (2.42); столбец *Pn* – рассчитанная вероятность, $M[n]$ – слагаемые математического ожидания СВ *n* (среднего числа запросов в системе). Выходные параметры рассчитаны по (2.43), (2.44) и (2.47) и располагаются в ячейках (O5–O7).

Ниже (ячейки C34–I45) представлена таблица, содержащая параметры $N_{\text{ср}}, \mu_{\text{ср}}, M\left[t_{\text{р}}\right]$ при разной интенсивности поступления заявок в систему.

Пример 3.3. В этом примере, алгоритм расчета которого изложен в §3.2, представлена реализация одновариантного анализа трехфазной замкнутой модели обслуживания на основе принципа эквивалентности потоков.

Реализация алгоритма в Приложении выполнена в двух вариантах (Лист Пример 3.3, вариант 1 и Пример 3.3, вариант 2). Результаты двух вариантов отличаются. Поэтому, возможно, в одном или в каждом варианте допущена ошибка в реализации алгоритма, описанного в §3.2.

Предлагается провести, аналогично примеру 2.8, проверку правдоподобия реализации расчета выходных параметров в каждом варианте, на основании этого найти возможные ошибки и, исправив их, дополнить соответствующий лист таблицы Excel, сформировав таблицы значений и построив на их основе графики зависимостей выходных параметров модели от входных параметров.

Вариант 1. Вариант реализован для расчета таких выходных параметров как математическое ожидание количества заявок на фазе обслуживания в системе Каналы–процессоры ($N_{\text{ср}}$) и математическое ожидание времени реакции фазы обслуживания (M_{tp}). Допускается изменение исходных данных в диапазонах:

число терминалов *N* ..... 1…20
число каналов *K* ..... 1…5
число процессоров *M* ..... 1…3

Кроме того, произвольно можно изменять математическое ожидание времени обдумывания, времени передачи по каналу и времени обработки в процессоре.

Таблица «Расчет АЭМ2» содержит результаты вычислений значений π_m по формулам (3.22). Поскольку при расчете компонент вектора $\mu_e(n)$ $(n=1,2,\ldots,N)$ число слагаемых последовательно увеличивается от $\overline{1,N}$, рассчитанные значения $\pi(i)$ для каждого $(n=1,2,\ldots,N)$ для наглядности расположены в одном столбце таблицы $\mu e(m)$ (ячейки O5-AL27). Ниже аналогичная таблица (ячейки O31–AL52) содержит рассчитанные по (3.23) компоненты суммы $\mu_e(n)$ и компоненты вектора, которые выделены жирным шрифтом (ячейки P32, Q34, R35, рис. П1).

Таблица «Расчет АЭМ1» (ячейки H55–N77) содержит значения P_n, рассчитанные по (3.16), (3.17). Значения исследуемых выходных параметров, полученных по формулам (3.18), (3.19), получены в ячейках R55–R56.

Вариант 2. Рассчитываются те же выходные параметры, что в варианте 1, дополнительно считается средняя пропускная способность фазы обслуживания ($\mu e_{\text{ср}}$), но при этом допускается изменение числа терминалов N от 1 до 15. Остальные входные параметры можно варьировать в тех же диапазонах, как в варианте 1.

Рекомендуется, разобравшись в реализации алгоритма расчета в этом варианте, самостоятельно расширить диапазон изменения числа терминалов N от 1 до 20 (как в варианте 1).

Таблица «Расчет АЭМ2» (ячейки I6–M27) содержит результаты вычислений значений π_m по (3.22) для N от 1 до 20, но расчет компонентов вектора $\{\mu_e(m)\}$ для m от 1 до 15 вынесен отдельно и представлен в столбцах P–AL.

Таблица «Расчет АЭМ1» (ячейки H49–N70) содержит значения P_n, рассчитанные по (3.16), а в ячейках O50–O70 рассчитывается средняя пропускная способность фазы обслуживания ($\mu e_{\text{ср}}$).

Пример 3.4. В этом примере, алгоритм расчета которого был изложен в §3.3, представлена реализация одновариантного анализа более сложной модели (замкнутой модели ИВС коллективного пользования с несколькими процессорами и раздельными очередями к каналам в подсистеме обмена), для которой

структура взаимодействия агрегированных субмоделей изображена на рис. 3.8.

Проверку правильности реализации алгоритма, коррекцию в случае необходимости и многовариантное исследование настоящей модели предлагается провести самостоятельно.

Таблица «Расчет АЭМ2» содержит значения, полученные с помощью (3.31) (ячейки I6–O28). Таблицы расчетов моделей АЭМ 2.1 (ячейки J33–AJ80) и АЭМ 1 (ячейки J85–O107) имеют структуру, аналогичную структуре таблиц «Расчет АЭМ1», «Расчет АЭМ2» предыдущего примера и содержат значения, полученные с помощью (3.28)–(3.30). Исследуемые выходные параметры рассчитаны в ячейках S85–S86 с помощью (3.26) и (3.27).

Пример 3.6 (расширенный). В этом примере (листы «Пример 3.6_расш», «tbl», «wrk») представлена реализация метода квазиэквивалентного укрупнения состояний марковского процесса для модели системы, описанной в §3.4 (пример 3.6), но несколько расширенной. На процессорной фазе допускается наличие не двух, а M процессоров, где M может принимать значения от 1 до 10, а на фазе терминалов число пользователей N может принимать значения от 1 до 50. Остальные условия работы системы соответствуют описанию Примера 3.6.

Прежде, чем рассматривать реализацию метода в таблице Excel, рекомендуется, проработав описание алгоритма в тексте примера 3.6, самостоятельно нарисовать графы укрупненных состояний процесса, аналогичные графам переходов марковского процесса, изображенных на рис. 3.15, графы б и в, и на основании этих графов и допущения (3.38), лежащего в основе метода квазиэквивалентного укрупнения состояний марковского процесса, написать расчетные соотношения для стационарных вероятностей укрупненных состояний (макросостояний) графа.

Для этого сначала надо написать выражения для интенсивностей переходов $\mu(i)$ графа макросостояний, изображенного на рис. 3.15,*в*.

В графе i – число заявок в системе обслуживания. Допустим, что в системе 4 процессора. Пусть $\pi(m)$ – стационарная вероятность, что в системе m процессоров получили отказ. При этом (4–m) процессоров могут обрабатывать заявки.

Тогда $\mu(1)=\mu*\left[\pi(0)+\pi(1)+\pi(2)+\pi(3)\right]$ – одна заявка: достаточно иметь хотя бы один работающий процессор; $\mu(2)=\mu*\left[2\pi(0)+2\pi(1)+2\pi(2)+\pi(3)\right]$ – две заявки, с вероятно-

стью $\pi(0)+\pi(1)+\pi(2)$ они обе обрабатываются, а с вероятностью $\pi(3)$ обрабатывается только одна из них;

$$\mu(3)=\mu*\left[3\pi(0)+3\pi(1)+2\pi(2)+\pi(3)\right];$$

$$\mu(4)=\mu*\left[4\pi(0)+3\pi(1)+2\pi(2)+\pi(3)\right];$$

$$\mu(5)=\mu(6)=\mu(7)=\ldots=\mu(N)=\mu(4).$$

Для другого числа процессоров формулы для $\mu(i)$ имеют несколько другой вид, но их можно написать на основании таких же соображений.

Особенностями реализации метода автоматического формирования подобных формул для $\mu(i)$ в таблице Excel является использование управляющего элемента «Поле со списком» и функций СМЕЩЕНИЕ и ИНДЕКС.

Выбор числа процессоров в системе производится в Поле со списком, а эти функции в результате позволяют в основном листе «Пример 3.6_расш» автоматически получить рабочую таблицу для расчета коэффициентов $\mu(i)$, где *i* принимает значения от 1 до *N*, соответствующую выбранному числу процессоров в системе. Пояснения к использованию каждого элемента приведены в соответствующих листах описываемого Приложения.

Задание к примеру – надо проверить правдоподобие модели, задавая разные значения исходных данных, а также провести многовариантный анализ, включающий графики зависимости выходных параметров от входных. При продумывании необходимых зависимостей надо иметь в виду, что они должны помочь ответить на вопросы типа «Что будет, если …» и «Как сделать, чтобы …», которые всегда стоят в практике построения и исследования сложных систем обработки информации при использовании математических моделей.

В отдельном файле Excel с именем «Quasy2a.xlsm» на листах «Qua_tbl» и «Results» представлена программная реализация алгоритма анализа методической погрешности метода квазиэквивалентного укрупнения состояний марковской модели рассмотренного примера 3.6 расширенного.

Алгоритм реализован на языке VBA (Visual Basic for Applicatioms) в указанной книге Excel. Основная программа – макрос Super (), использующая ряд вспомогательных подпрограмм. Вызов макроса производится из листа «Qua_tbl» нажатием комбинации клавиш «Ctrl+z» после того, как на указанном листе Вы ввели исходные данные модели – в ячейки A9, B16, C9–F9 значения пере-

менных N – число терминалов, M – число процессоров, а также $T_{\text{отк}}$, $T_{\text{восст}}$, $T_{\text{обд}}$, $T_{\text{реш}}$.

Результаты появляются на листе «Results» в ячейках H11–R11 и одновременно добавляются к списку результатов протокола проведенных экспериментов – строки 18–19, при этом все предыдущие результаты автоматически смещаются вниз.

В алгоритме сначала рассчитываются выходные параметры модели на основе метода квазиэквивалентного укрупнения состояний (через вероятности стационарного распределения макросостояний), затем решается СЛАУ для вероятностей стационарного распределения состояний исходного графа переходов, снова рассчитываются выходные параметры модели и вычисляется относительная погрешность расчета каждого выходного параметра.

С целью как можно более простой реализации для расчета вероятности стационарного распределения состояний исходного графа при решении СЛАУ вместо метода Гаусса использован метод итераций, где уравнения Колмогорова для каждого состояния записываются непосредственно на основании структуры графа переходов. При этом левые части уравнений (вероятности стационарного состояния) – это последующее приближение в итеративном процессе, а вероятности в правых частях уравнений – предыдущее. В качестве начального приближения берется стационарная вероятность, рассчитанная из допущения (3.38), лежащего в основе метода квазиэквивалентного укрупнения состояний марковского процесса.

В алгоритме для любой комбинации исходных данных автоматически проверяется сходимость итерационного процесса, и в списке результатов выводится оценка максимального расхождения вероятностей стационарного состояния последнего и предпоследнего приближения. Эта оценка дает погрешность решения СЛАУ относительно вероятности стационарного состояния марковского процесса (точности инструмента анализа методической погрешности). Далее сравниваются результаты расчета выходных параметров.

Многочисленные расчеты, проведенные с помощью реализованного алгоритма, подтверждают правомерность использования метода квазиэквивалентного укрупнения состояний при оценке выходных параметров моделей большой размерности.

ЛИТЕРАТУРА

1. *Клейнрок Л.* Вычислительные системы с очередями. М.: Мир. 1979.
2. *Норенков И.П.* Основы автоматизированного проектирования. Изд. 4-е, перераб. и доп. М.: Изд-во МГТУ им. Н.Э. Баумана. 2009.
3. *Пашин В.М.* Критерии для согласованной оптимизации подсистем судна. Л.: Судостроение. 1976.
4. *Верба В.С., Михеев В.А.* Системный анализ методов проектирования многофункциональной информационной системы // Известия ЮФУ. Сер. «Технические науки». 2008. С. 109–116.
5. *Моисеев Н.Н.* Математические задачи системного анализа. Изд. 2-е. М.: ЛИБРОКОМ. 2012.
6. *Норенков И.П.* Системы автоматизированного проектирования: Учеб. пособие для втузов: в 9-ти кн. Кн. 1. Принципы построения и структуры. М.: Высшая школа. 1986.
7. *Михеев В.А.* Основы проектирования и построения многофункциональных информационных систем интегрированных структур оборонно-промышленного комплекса. Теория и практика. М.: Высшая школа экономики. 2014.
8. *Михеев В.А.* Задачи анализа и синтеза многофункциональной информационной системы интегрированной структуры промышленного комплекса // Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение». Спецвыпуск № 5. Информатика и системы управления. 2012. С. 62–66.
9. *Шкатов П.Н.* Методы построения математических моделей оценки характеристик производительности ИВС АСУ. М.: Изд-во МВТУ. 1984.
10. *Клейнрок Л.* Теория массового обслуживания. М.: 1979.
11. *Полуян Л.Я.* Методика построения формальных моделей вычислительных комплексов АСУ // Вычислительные системы обработки измерительной информации. Рязань. 1981.
12. *Балыбердин В.А.* Оценка и оптимизация характеристик систем обработки данных. М. 1987.
13. *Жожикашвили В.А., Вишневский В.М.* Сети массового обслуживания. Теория и применение к сетям ЭВМ. М. 1988.
14. *Baskett F., Chandy K.M., Muntz R.R. Palacios-Gomez F.* Open, closed, mixed networks of queues with different classes of customers // Journal of the Assoc. Comp.March 1975.
15. *Нестеров Ю.Г., Галстян А.Г.* Декомпозиционный подход к анализу сетевых моделей вычислительных систем // В кн. «Алгоритмы и структуры специальных вычислительных систем». Тула. 1986.

16. *Олифер Н.А., Байкенов А.С.* Исследование сетей массового обслуживания методами вложенных процессов // В кн. «Алгоритмы и структуры специальных вычислительных систем». Тула. 1985.
17. *Кузовлев В.И., Шкатов П.Н.* Разработка САПР: В 10-ти кн. Кн. 8 «Математические методы анализа производительности и надежности САПР». Практич. пособие / Под ред. *А.В. Петрова*. М.: Высшая школа. 1990.
18. *Михеев В.А.* Методология разработки и аттестации автоматизированных систем в защищенном исполнении // Материалы IX Междунар. науч.-практич. конф. «Информационная безопасность». Таганрог: Изд-во ТТИ ЮФУ. 2007. С. 86–89.
19. *Михеев В.А.* Научно-технологические проблемы информатизации интегрированных структур оборонно-промышленного комплекса и пути их решения // Технологии ЭМС. 2012. № 4(43). М.: Технологии. С. 36–43.
20. *Постников В.М., Черненький В.М.* Методы принятия решений в системах организационного управления. М.: Изд-во МГТУ им. Баумана. 2014. 205 с.
21. *Черненький В.М.* Алгоритмический метод описания дискретных процессов функционирования систем // Информационно-измерительные и управляющие системы. 2016. Т. 14. № 12. С. 11–21.
22. Патент № 2574281 (РФ), МПК, HO4W12/00, G06F17/40, G06F 15/16. Многофункциональные информационные системы интегрированных структур оборонно-промышленного комплекса. / *В.А. Михеев*.
23. *Давыдов Е.Г.* Интегрированная система Scientific Workplace 4.0: Технология работы и практика решения задач. М.: Финансы и статистика. 2003. 208 с.
24. *Дьяконов В.П.* Mathematica 5.1/5.2/6. Программирование и математические вычисления. М.: ДМК Пресс. 2008. 576 с.
25. *Подиновский В.В., Ногин В.Д.* Парето-оптимальные решения многокритериальных задач. М.: Наука. 1982. 254 с.
26. *Hendrickson B.A., Wright M.H.* Mathematical Research Challenges in Optimization of Complex Systems. DOE Workshop Report. December 2006.
27. *Dunlavy D.M., Hendrickson B.A., Kolda T.G.* Mathematical Challenges in Cybersecurity. Sandia Report SAND 2009-0805. February 2009.
28. *Newman M.E.J.* The structure and function of complex networks // SIAM Review. 2003. № 45. P. 167–256.
29. *Corogovtsev A.N., Goltsev A.V., Mendes J.F.F.* Critical phenomena in complex networks // Reviews of Modern Physics. 2008. № 80. P. 1275–1335.
30. *Amaral L.A.N., Ottino J.M.* Complex networks // Eur. Phys. J. 2004. B38. P. 147–162.
31. *Perumalla K. and Sundaragopalan S.* High-Fidelity Modeling of Computer Network Worms // Proc. 20th Annual Computer Secuity Applications Conference. 2005. 126 p.
32. *Arionos S., Bonpard E., Carbone A., and Xue F.* Power Grid Vulnerability: A Complex Network Approach // Chaos. March 2009. V 19. № 1.
33. *Oberkampf W.L. and Roy C.J.* Verification and validation in scientific computing. Cambridge Univ. Pr. 2010.

34. *Borggaard Hay J.T. and Pelletier D.* Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition // Journal of Fluid Mechanics. 2009. V. 629. № 1. P. 41–72.
35. *Singler J.R and Batten B.A.* Balanced proper orthogonal decomposition for model reduction of infinite dimensional linear systems // In Proceedings of the Int. Conf. on Computational and Mathematical Methods in Science and Engineering. CMMSE. 2007
36. *Gorguis A.* A comparison between cole-hopf transformation and the decomposition method for solving burgers' equations // Applied Mathematics and Computation. 2006. V. 173. № 1. P. 126–136.
37. *Kaya D.* An explicit solution of coupled viscous burgers' equation by the decomposition method // International Journal of Mathematics and Mathematical Sciences. 2001. V. 27. № 11. P. 675–680.

Учебное издание

А в т о р ы :

Вячеслав Алексеевич Михеев,
Валерий Михайлович Черненький,
Петр Николаевич Шкатов

ПРОЕКТИРОВАНИЕ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

МЕТОДЫ И АЛГОРИТМЫ РАСЧЕТА

Учебное пособие

Под редакцией д.т.н., проф. В.М. Черненького

Изд. № 20. Сдано в набор 11.07.2017.
Подписано в печать 24.10.2017.
Формат 60×90 1/16. Бумага офсетная.
Гарнитура Таймс. Печать офсетная.
Печ. л. 11. Тираж 500 экз. Зак. №

Издательство «Радиотехника»
107031, Москва, К-31, Кузнецкий мост, д. 20/6
Тел./факс: (495)621-48-37; 625-78-72, 625-92-41
e-mail: info@radiotec.ru
www.radiotec.ru

Отпечатано в типографии
ФГУП «Издательство «Известия»