

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ РСФСР

КУЙБЫШЕВСКИЙ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
АВИАЦИОННЫЙ ИНСТИТУТ ИМЕНИ АКАДЕМИКА С. П. КОРОЛЕВА

МОСКОВСКИЙ ОРДЕНА ЛЕНИНА
И ОРДЕНА ОКТЯБРЬСКОЙ РЕВОЛЮЦИИ
АВИАЦИОННЫЙ ИНСТИТУТ ИМЕНИ СЕРГО ОРДЖОНИКИДЗЕ

*А. С. Голубков, В. А. Виттих, Н. А. Максимов,
В. А. Цыбатов, А. В. Шароватов*

АНАЛИЗ ХАРАКТЕРИСТИК
И ВЫБОР ПАРАМЕТРОВ СЛОЖНЫХ
ТЕРРИТОРИАЛЬНО
РАСПРЕДЕЛЕННЫХ
СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ

Утверждено
редакционно-издательским
советом института
в качестве учебного пособия

УДК 681.32(075)

Анализ характеристик и выбор параметров сложных территориально распределенных систем обработки информации / А. С. Голубков, В. А. Виттих, Н. А. Максимов, В. А. Цыбатов, А. В. Шароватов. — Куйбышев: Куйбышевский авиационный институт, Московский авиационный институт. 1984. — 168 с.

В пособии рассматривается комплекс вопросов, связанных с анализом характеристик и выбором параметров сложных территориально распределенных систем сбора, передачи и обработки информации, особенности функционирования которых на ранних этапах проектирования могут быть описаны в рамках теории массового обслуживания и стохастических дискретных систем.

В качестве типового примера рассматривается система, предназначенная для сбора и обработки данных летных испытаний. Анализ системы проведен на уровне системотехнических характеристик, связанных с временными задержками вычислительных процессов и загрузкой оборудования.

Обсуждаются вопросы построения оптимальной топологической схемы системы.

Материалы пособия могут быть использованы при курсовом и дипломном проектировании студентами, обучающимися по специальностям «Автоматические информационные устройства», «Автоматизированные системы управления».

Ил. 43, табл. 23, библиогр. — 48 назв.

Рецензенты: д-р техн. наук проф. В. М. Агеев,
канд. техн. наук С. В. Касаткин

В соответствии с решениями партии [1] в последнее десятилетие развернулась большая работа по внедрению средств электронно-вычислительной техники во все сферы общественного производства. Создаются информационно-вычислительные центры на предприятиях, в министерствах, научных учреждениях и учебных заведениях. В стране ведутся работы по созданию единой общегосударственной сети вычислительных центров. Широко внедрилась вычислительная техника и в инженерную деятельность. Сегодня ни один проект, ни одна разработка практически не обходятся без расчетов на ЭВМ, все шире осуществляется переход к автоматизации проектирования.

Все это стало возможным благодаря разработке и серийному производству ЭВМ третьего поколения с развитыми периферийными средствами и системами передачи данных. Совместное использование вычислительных машин и средств передачи данных открывает новые возможности в решении как вопросов сбора, хранения, распределения информации, так и вопросов коллективного пользования растущими вычислительными ресурсами.

Наличие системы передачи данных позволяет перейти от локальных вычислительных центров, в которых сам процессор и все периферийные устройства находятся в одном месте, к распределенным на значительной территории многомашинным комплексам. Такие распределенные комплексы получили название *сети ЭВМ*. Разработке и внедрению подобных сетей предшествовали многочисленные научные исследования, когда разрабатывались методы анализа, оценки и управления отдельными режимами работы их весьма дорогостоящих элементов. Методы базировались, как правило, на использовании традиционных функциональных представлений процессов и их преобразований, которые не позволяют исследовать характеристики комплексов в смысле их пропускной способности, времени реакции

и прочих показателей, связанных с задачами исследования операций.

Параллельно с созданием методов исследования конкретных вычислительных комплексов и сетей ЭВМ шло развитие специальных математических дисциплин, таких как теория массового обслуживания, линейное и нелинейное программирование и т. п. Осмысление и использование инженерами идей и методов, накопленных в ходе развития этих математических дисциплин, способствовало улучшению качества проектных решений и сокращению сроков проектирования.

Разработка сложных вычислительных систем и сетей ЭВМ порождает не только технические, но и научные проблемы, связанные с анализом и синтезом математических моделей функционирования, а также разработкой на их базе аналитических подходов и статистического моделирования систем. Нередко в результате совместной работы инженеров и математиков возникают новые математические идеи и методы. Такое взаимодействие привело к стремительному росту числа публикаций в области вычислительной техники и примыкающих к ней математических дисциплин.

Несмотря на значительные успехи в области создания математических методов описания и анализа информационно-вычислительных систем и сетей, до сих пор не разработаны полностью формализованные инженерные методики проектирования таких систем. Это объясняется большим количеством факторов, которые необходимо учитывать, а также сложностью взаимосвязей между ними. Как показывает практика, создание адекватных моделей сложных систем по временным и материальным затратам соизмеримо с разработкой и созданием самой системы.

В этих условиях общепринятой является многоэтапная процедура исследования и проектирования сложных систем, основанная на декомпозиции сложной системы на ряд более простых подсистем и использовании аналитических и имитационных моделей, описывающих функционирование отдельных подсистем и принципы их взаимосвязи. В рамках такого подхода можно говорить о возможности разработки стандартных математических конструкций, пакетов прикладных программ и т. п., что, в свою очередь, обеспечило бы необходимую автоматизацию процесса моделирования и принятия проектных решений. Указанная концепция, сформулированная в виде многоэтапной процедуры анализа и синтеза операционных характеристик сложной территориально распределенной системы сбора и обработки информации, легла в основу настоящей книги.

В качестве типового примера будем рассматривать систему, предназначенную для сбора и обработки данных, получаемых

при испытаниях и штатной эксплуатации летательных аппаратов (ЛА). Такие системы состоят из наземных пунктов приема и передачи информации, центров коммутации сообщений, центров обработки информации и абонентской сети пользователей. Таким образом, мы имеем сложную сеть специализированных и универсальных вычислительных средств, предназначенных для решения задач сбора и обработки телеметрической, внешнетраекторной, командной информации и обслуживания ее потребителей. Причем все эти задачи решаются преимущественно в реальном масштабе времени. С учетом данной специфики построения системы сбора и обработки информации указанная выше процедура проектирования может быть представлена в виде схемы (рис. В1). На рисунке показана типовая последовательность этапов проектирования системы с учетом их взаимосвязи и распределения функций между ними.

На *первом* этапе решается задача формирования исходных данных в виде структуры и параметров информационных потоков, поступающих от внешних источников информации (комплекс ЛА). Здесь же решаются задачи выработки оптимальной стратегии взаимодействия ЛА и наземных средств приема. Взаимодействие указанных средств является сложным информационным процессом, оценка качества которого должна проводиться на основе заданного множества критериев. Задача поиска оптимальных стратегий управления информационным процессом, которая формулируется как задача планирования, не может быть решена непосредственно в рамках теории расписаний методами параметрической оптимизации. Данную задачу предлагается решать на базе человеко-машинного подхода к поиску наилучших стратегий управления ресурсами комплекса технических средств приема информации. Роль человека при этом состоит в нахождении целей управления, ранжировании критериев эффективности и выборе пороговых значений управляющих параметров при заданных ограничениях. Деятельность человека, принимающего решение, протекает в диалоговом режиме с ЭВМ, реализующей программу эвристического поиска оптимальных стратегий. Конечным итогом процедуры поиска оптимальных стратегий является локально-оптимальный вариант распределения ресурсов технических средств наземного комплекса во времени и пространстве.

Вторым этапом в процедуре синтеза системы является генерация предварительных альтернативных вариантов структуры сети передачи данных. Эти варианты отражают представления разработчика и строятся на базе имеющихся прототипов и возможностей современных технических средств. При этом учитывается как наличие «жестких», заранее запланированных свл-

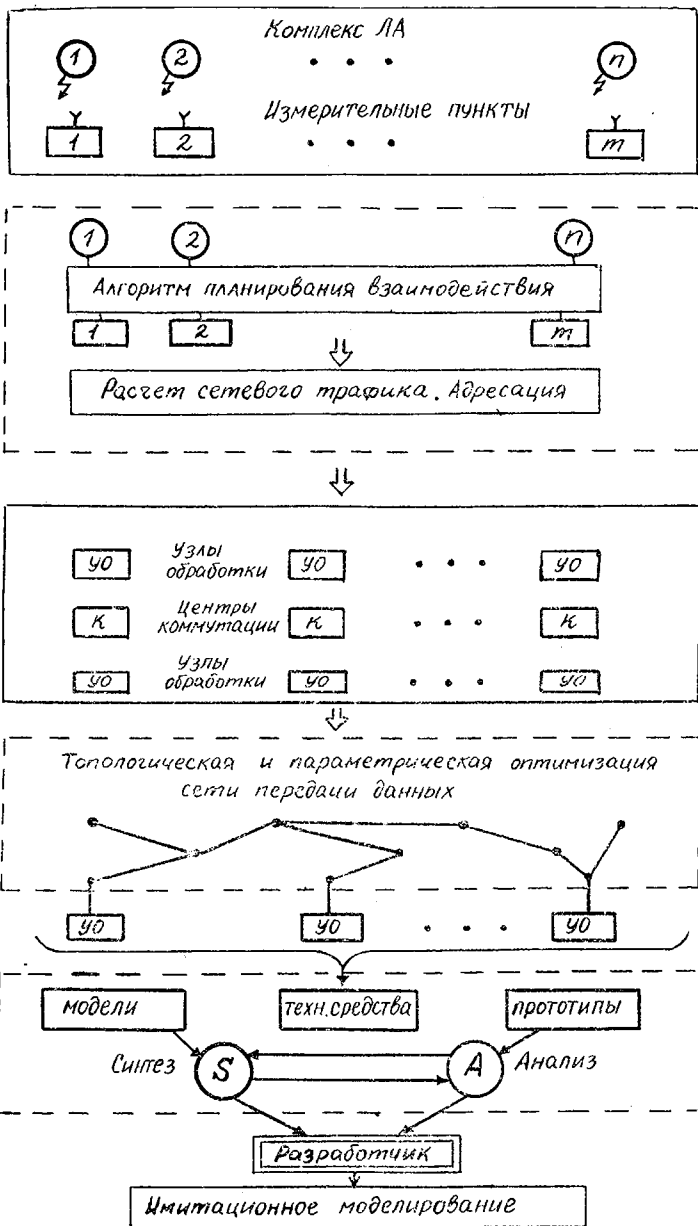


Рис. В1. Схема проектирования

зей, так и возможность распределения информационных потоков.

На *третьем* этапе для каждого из альтернативных вариантов, полученных на втором этапе, решается задача оптимизации характеристик сети передачи данных, включающая оптимизацию топологической схемы системы, выбор пропускных способностей каналов связи и оптимальное распределение сетевого графика. Итогом является нахождение информационной нагрузки на каждый узел обработки. Таким образом осуществляется декомпозиция системы на отдельные подсистемы обработки информации.

На *четвертом* и *пятом* этапах осуществляется выбор технических средств и синтез управляющих алгоритмов автоматизированных систем обработки информации (АСОИ). Эти две задачи тесно взаимосвязаны, так как выбор технических средств часто предопределяет и состав специального математического обеспечения, включая и управляющие алгоритмы. Они могут быть решены на основе итерационного процесса синтеза и анализа моделей функционирования узлов обработки — АСОИ. Результаты, полученные на последних этапах, дают возможность выбрать конкретный состав технической базы узлов обработки, структуру и параметры управляющих алгоритмов.

Следующий, *шестой* этап предполагает проведение имитационного моделирования синтезированных вариантов узлов обработки, что позволяет:

подтвердить обоснованность принятых аналитических моделей и принятых ограничений;

дополнить результаты аналитических исследований машинным экспериментом в достаточно узком диапазоне варьируемых параметров при модифицированном представлении о структуре и параметрах управляющих алгоритмов комплексов обработки;

выявить «узкие» места комплексов обработки и дать рекомендации по их ликвидации.

Структура настоящего пособия в основных чертах воспроизводит указанную последовательность этапов процесса проектирования. Ввиду того, что книга носит учебный характер, она снабжена достаточным количеством примеров, большинство из которых доведено до числовых расчетов характеристик исследуемых систем.

Ограниченный объем книги не позволяет дать исчерпывающее описание пакетов прикладных программ аналитического и имитационного моделирования, однако, авторы надеются, что предложенный материал в определенной мере будет служить ориентиром в исследовании сложных территориально распределенных систем сбора и обработки информации.

Авторы выражают искреннюю благодарность аспирант И. М. Соколову, студентам А. Г. Коновалову, Ю. Р. Львову, М. Н. Обуховой, О. Б. Павликовой, Л. Д. Поляку, А. В. Тучкину, В. В. Лещенко, М. А. Банникову, принимавшим участие в создании, отладке и пробной эксплуатации отдельных методических пакетов прикладных программ.

Глава 1

ПЛАНИРОВАНИЕ ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ ЛЕТАТЕЛЬНЫХ АППАРАТОВ (ЛА) С ИЗМЕРИТЕЛЬНЫМИ ПУНКТАМИ

1.1. ЗАДАЧА РАСПРЕДЕЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ ИЗМЕРИТЕЛЬНЫХ ПУНКТОВ НА МНОЖЕСТВЕ ЛА

Проведение испытаний ЛА сопряжено с использованием большого количества аппаратурных средств, распределенных на большой территории. Организация согласованной и высокоэффективной работы всех средств при одновременном проведении испытаний и штатной эксплуатации параллельно большого количества объектов (ЛА) требует разработки теоретических основ планирования и управления работой всего комплекса технических средств [2, 3].

В силу большой размерности, многокритериальности и сложного характера информационного взаимодействия ЛА и средств наземного комплекса, задача планирования их оптимального взаимодействия является весьма сложной и не поддается решению известными методами. Поэтому ниже предлагается упрощенная методика, основанная на декомпозиции общей задачи на две частные, достаточно слабо связанные между собой, которые можно решать последовательно. Результатом решения первой задачи является расписание взаимодействия ЛА с измерительными пунктами или пунктами приема информации с учетом временных и пространственных ограничений, налагаемых траекториями ЛА, географическими положениями пунктов приема информации и особенностями проводимых экспериментов. Вторая задача заключается в оптимизации загрузки аппаратуры каждого измерительного пункта и оптимизации его взаимодействия с группой ЛА при условии, что распределение сеансов связи уже проведено.

Задача назначения сеансов связи (СС) с объектами на определенных измерительных пунктах (ИП) сводится к составлению временных графиков работы наземных средств в соответствии с заданной программой проведения экспериментов с ЛА и оперативными потребностями пользователей. Под пользователями в данном случае будем понимать специалистов, проводящих испытания ЛА или их штатную эксплуатацию.

Составленное расписание должно в наибольшей степени от-
вечать всем запросам пользователей, по возможности полно и
равномерно загружать технические средства системы и обеспе-
чивать своевременное выполнение технологических операций,
проводимых при испытаниях ЛА.

Попытаемся формализовать эту задачу. Пусть задано мно-
жество измерительных пунктов $\{M\}$ со своими географическими
координатами и множество объектов, которые двигаются по из-
вестным траекториям O_i ($i = \overline{1, M}$). Имеется множество заявок
 $\{Z\}$, которые несут требования на организацию проведения опре-
деленных актов взаимодействия между объектом (объектами)
и измерительным пунктом (пунктами) в определенном интер-
вале времени $t_j - t_{j+n}$. Требуется составить расписание работы
измерительных пунктов с объектами с учетом нахождения объ-
ектов в период работы в зонах видимости ИП таким образом,
чтобы некоторый функционал, оценивающий качество составлен-
ного расписания, принял экстремальное значение на заданном
множестве критериев $\{K\}$.

Оценка расписания может проводиться, например, по таким
показателям, как:

удовлетворение максимального количества заявок на выпол-
нение работ (с учетом приоритета заявок);

назначение минимального количества ИП по каждому
виду работ;

максимальная компактность сеансов связи;

максимальная загрузка средств ИП;

равномерность загрузки ИП.

Конечно, этот перечень не исчерпывает всех возможных по-
казателей оценки качества составленного расписания. Нетрудно
заметить, что в большинстве показателей, оценивающих эффек-
тивность планирования, в прямой или косвенной форме вклю-
чается показатель степени загруженности системы, поэтому рас-
смотрим его несколько подробнее.

Целесообразно оценивать степень загрузки ИП, системы в це-
лом, а также равномерность загрузки с помощью безразмерных
параметров, учитывающих весовые коэффициенты по каждому
из средств ИП или всей системы в целом. В качестве таких
весовых коэффициентов можно принять *относительную стои-
мость* i -го компонента (средства ИП). Степень загрузки систе-
мы можно в этом случае охарактеризовать параметром
 F — *средневзвешенной загрузкой системы*, который определя-
ется как

$$F = \sum_{i=1}^n p_i U_i,$$

где p_i — относительная стоимость i -го компонента;

n — число средств ИП;

U_i — коэффициент использования во времени i -го компонента.

Таким образом, коэффициент F характеризует степень использования (загрузки) системы (ИП) в целом. Он изменяется от 0 (в неиспользуемой) до 1 в полностью загруженной системе.

В качестве критерия равномерности загрузки удобно использовать коэффициент *дебаланса*, который определяется как

$$B = 2 \left[\sum_{i=1}^n (F - U_i)^2 p_i \right]^{1/2}.$$

Выражение в круглых скобках есть мера отклонения загрузки i -го компонента U_i от средневзвешенной загрузки системы F . Как и в случае подсчета коэффициента F , эти индивидуальные отклонения суммируются с весами p_i . Нормирующий множитель обеспечивает изменение B от 0 (для полностью сбалансированной системы) до 1 в случае максимального дисбаланса. Для полностью сбалансированной системы соблюдается $U_i = F$ для всех i , то есть загрузка каждого компонента равна средневзвешенной загрузке системы. Максимально разбалансирована такая система, в которой

$$U_i = 1 \quad (i = \overline{1, m}); \quad \sum_{i=1}^m p_i = 0,5$$

$$U_i = 0 \quad (i = \overline{m+1, n}); \quad \sum_{i=m+1}^n p_i = 0,5,$$

так что

$$F = \sum_{i=1}^m U_i p_i + \sum_{i=m+1}^n U_i p_i = 0,5 + 0 = 0,5;$$

$$B = 2 \left[(0,5 - 1)^2 \sum_{i=1}^m p_i + (0,5 - 0)^2 \sum_{i=m+1}^n p_i \right]^{1/2} = \\ = 2 \left[(0,25 \cdot 0,5) + (0,25 \cdot 0,5) \right]^{1/2} = 1.$$

Физически случай максимального дебаланса отражает ситуацию, когда одна часть компонентов системы, стоимость которой составляет 50% стоимости всей системы, загружена на 100%, а остальная часть не загружена вовсе. Этот случай можно проиллюстрировать примером вычислительной системы, состоящей из двух идентичных ЭВМ, когда на рассматриваемом временном интервале для решения поставленной задачи достаточно использовать лишь одну ЭВМ из двух, имеющихся в наличии.

Введенные выше критерии отражают основные требования, предъявляемые к качеству планирования взаимодействия объектов с наземными средствами командно-измерительного комп-

лекса. Однако на пути синтеза самого алгоритма планирования возникают существенные трудности в связи с многокритериальностью задачи и сильным влиянием субъективных факторов при окончательном принятии решения. В этих условиях попытка использовать классические методы упорядочивания, распределения, теории расписаний и т. п. наталкивается на серьезные трудности.

В качестве альтернативы ниже предложен адаптивный подход с использованием человеко-машинного взаимодействия, показавший во многих случаях свою высокую эффективность. Роль человека заключается в выборе и назначении величин управляющих параметров, которые используются при работе алгоритма, оценке вариантов полученного расписания и ряде других операций.

Описание алгоритма планирования

Исходными данными при планировании является следующая информация:

1. Программа полета и взаимодействия каждого объекта со средствами наземного комплекса.

2. Состав и расположение средств наземного комплекса.

3. Заявки на включение в план определенных работ, проводимых объектом во взаимодействии со средствами наземного комплекса.

4. Информация о ранее разработанных планах взаимодействия объектов и средств наземного комплекса и сведения о ходе выполнения этих планов к текущему моменту времени.

Предлагаемый алгоритм планирования по своей сути является переборным. Вся имеющаяся информация упорядочена в нескольких таблицах. Для работы алгоритма требуется три типа таблиц:

таблица возможных сеансов связи (ТВС) ;

таблица-план ;

таблица заявок на обслуживание (ТЗО) .

Привязка ко времени осуществляется путем разбиения интервала планирования на дискретные промежутки времени и выделения необходимого числа таких дискретных интервалов на выполнение той или иной задачи.

Таблица ТВС (табл. 1-1) содержит информацию о времени видимости каждого объекта последовательно со всех ИП и составляется на основе программы полета каждого объекта и решения навигационных задач .

Таблица 1.1

Возможные сеансы связи

Номер суток	Номер объекта	Номер ИП	Временной интервал					
			1	2	3	

В том случае, если в данный интервал времени определенный объект находится в зоне видимости какого-нибудь ИП, то в соответствующей клетке таблицы ставится 1, в противном случае — 0.

В табл. 1.2 содержится информация о состоянии средств наземного комплекса в определенные моменты времени.

Таблица 1.2

Таблица - план

Номер суток	Номер ИП	Номер средства	Временной интервал				
			1	2	3	4	...
1	1	1 2 ⋮					
	2	1 2 ⋮					
	⋮						
2	⋮						
⋮							

В соответствующие клетки таблицы проставляется 1, если какое-либо средство ИП не может находиться в рабочем состоя-

ний в данный интервал времени (отсутствует, не работает, находится на профилактике и т. п.), 0 — если средство ожидает поступления задания, и, наконец, №_{об} — номер объекта, если средство занято обслуживанием объекта с соответствующим номером.

Наконец, последняя таблица (табл. 1.3), которая является наиболее информативной.

Таблица 1.3

Заявки на обслуживание

Приоритет заявки	Номер объекта	Номер средства	Время обработки заявки	Временной интервал	Допуск на изменение временного интервала	Признак обслуживания	№ ИП, обслуживающего объект
1	2	3	4	5	6	7	8
...		

Порядок заполнения этой таблицы следующий. Поступившая заявка содержит сведения о номере объекта, для которого будет производиться планирование его взаимодействия со средствами командно-измерительного комплекса, приоритет объекта, а также интервал времени, в течение которого желательно организовать его обслуживание. Кроме этой информации, в заявке содержится время обслуживания, т. е. время, в течение которого необходимо производить передачу информации с борта или на борт ЛА, перечень средств наземного комплекса, с которыми будет взаимодействовать ЛА в течение сеанса связи. Графы 7 и 8 заполняют в процессе работы алгоритма.

Описание планирующего алгоритма

После заполнения таблиц, описание которых приведено выше, алгоритм планирования производит упорядочивание всех заявок по приоритетам. Таким образом, планирование начинается с заявок высшего приоритета. Из таблицы заявок на обслуживание выбирается первая заявка и определяется, какие ресурсы и в какое время она запрашивала. Используя эти данные, алгоритм планирования по ТВС определяет ИП, в зоне которого находится объект в запрашиваемый промежуток времени. Если объект находится сразу в зоне радиовидимости двух

ИП, то в соответствии с текущей загрузкой этих ИП для взаимодействия с объектом выделяется наименее загруженный. При этом в соответствующей графе ИП загрузку, которую средство имело до прихода данной заявки, слагают с ресурсом; потребным для ее обработки, после чего производят проверку достаточности ресурсов средств ИП для обслуживания пришедшей заявки.

Если загрузка средства не превышает единицы, то принимается решение об обслуживании заявки данным средством, и в столбцах 7 и 8 ТЗО проставляются соответственно «1» и номер ИП, производящего обслуживание. В противном случае, а также, если в заданный момент времени объект не может установить связь ни с одним ИП, алгоритм начинает поиск возможности проведения сеанса связи для этого объекта с любым ИП в соседних с запрошенным интервалах времени, и процедура повторяется. Конечно, подобное «расширение» запрошенного интервала возможно лишь до определенных пределов, так как пользователь системы вряд ли будет удовлетворен, если заявка, обслужить которую требовалось в начале суток, получит возможность обслуживаться в конце суток.

Для устранения возникновения подобных ситуаций в алгоритм вводится дополнительный параметр — *допуск на расширение временного интервала*, так что заявка может быть обслужена только в пределах этого временного интервала. Величина этого допуска является одним из управляющих параметров и может назначаться лицом, ответственным за планирование по согласованию с пользователями. Можно использовать два варианта алгоритма планирования: в первом из них величина допуска задается пользователем и записывается в ТЗО (графа 6). Например, если там указано $TI = 2$, а $DT = 2$, то желательно, чтобы объект был обслужен во временной интервал (TI) под номером 2, а если такой возможности нет, то либо в первый, либо в третий временной интервал (так как допуск на время обслуживания $DT = 2$).

Второй вариант алгоритма обслуживания не предусматривает такого априорного задания допуска на обслуживание, а производит «расширение» интервала планирования, исходя из оптимальности плана в целом.

Для достижения большей равномерности загрузки в алгоритме планирования предусматривается введение еще одного управляющего параметра, который тоже задается человеком и может меняться в ходе проведения планирования. Это так называемая *критическая загрузка*. Ее значение принимается равным некоторой положительной величине $FK < 1$, причем значения FK могут назначаться независимо для каждого ИП и да-

же для каждого отдельного средства, входящего в состав ИП.

Алгоритм планирования при введении этого параметра реализуется следующим образом. После ввода исходных данных проверяется условие достаточности ресурсов для обслуживания объекта. Если оно выполняется, проверяется более строгое условие $FK < 1$, т. е. проверяется, превосходит ли фактическая загрузка критическую. Если загрузка ниже критической, то заявка принимается на обслуживание данным ИП или средством. В противном случае происходит расширение интервала в пределах допуска, осуществляется поиск подходящих ИП и повторяется описанная процедура.

По окончании работы алгоритма в таблице заявок будет содержаться информация о месте и времени обслуживания каждого объекта. Если сложится такая ситуация, что какое-либо средство ИП в один и тот же момент времени обслуживает несколько объектов, то вступает в действие алгоритм оптимизации работы планирующего алгоритма, описанный ниже. В случае, если составить удовлетворяющий пользователей план все равно не удастся, то может быть сделана попытка расчленив сеанс связи для некоторых объектов на отдельные части и попытаться найти удовлетворительный план для отдельного ИП.

Оптимизация плана проводится с использованием описанных выше критериев: загрузки системы FK и дебаланса системы B . Указанные критерии отражают эффективность использования технических средств системы, но практически не учитывают интересов пользователей, ради которых и проводится планирование. Для устранения указанного несоответствия введем еще два критерия. Первым из них будет критерий, учитывающий количество обслуженных заявок за период планирования

$$Y = \frac{Z_0}{Z},$$

где Z_0 и Z — соответственно число обслуженных и принятых к планированию заявок.

В качестве второго критерия будем использовать величину среднего отклонения выделенного интервала обслуживания к запрошенному по множеству принятых к обслуживанию заявок:

$$DT = \sum_{i=1}^z \frac{|T_i' - T_i|}{kZ},$$

где k — общее число интервалов;

T_i' — заказанный интервал;

T_i — назначенный интервал;

Z — общее число заявок, принятых к планированию.

Физический смысл этого критерия ясен, если положить $k=1$.

Тогда DT равняется числу интервалов, на которое было в среднем сдвинуто обслуживание объекта по отношению к запрошенному интервалу при составлении плана.

Свертку указанных критериев будем осуществлять в виде аддитивной формы:

$$E_c = \sum_{j=1}^S \lambda_j E_j,$$

где E_j — частный критерий;

λ_j — коэффициент относительной важности.

Для уменьшения субъективности оценок в определении λ_j может быть использован метод, заключающийся в ранжировании частных критериев, а затем присвоении им оценок в баллах по определенной шкале.

Поиск оптимального плана производят на множестве планов, сформированных в ходе работы алгоритма планирования при различных значениях критической загрузки и допуска на временной интервал. Найдя оптимальный план, мы получаем описание работы средств наземного комплекса с летательными аппаратами, что позволяет определить информационную нагрузку на каждый ИП. Далее, имея значения информационной нагрузки для всех ИП, мы можем определить временные параметры прохождения заявок через ИП, решить задачу оптимального выбора технических средств для ИП, а также ряд других задач.

1.2. ОПТИМАЛЬНОЕ ПЛАНИРОВАНИЕ ВЗАИМОДЕЙСТВИЯ ГРУППЫ ЛА С ИЗМЕРИТЕЛЬНЫМ ПУНКТОМ

Рассмотрим задачу оптимального планирования взаимодействия объектов (ЛА) с одним измерительным пунктом. Под взаимодействием будем понимать передачу информации с борта объекта на измерительный пункт по радиоканалу.

Пусть заданы моменты входа объектов в зону радиовидимости t_{x_i} и моменты выхода t_{y_i} ($i = \overline{1, m}$). Тогда в течение времени $T_i = t_{y_i} - t_{x_i}$ объект может взаимодействовать с ИП. Интервал $[t_{x_i}, t_{y_i}]$ — зона радиовидимости для i -го объекта. В общем случае эти интервалы для некоторой части объектов могут пересекаться. Измерительный пункт может осуществлять одновременное взаимодействие с k объектами. Если число пересекающихся зон радиовидимости $S > k$, то возникают проблемы реализации взаимодействия ИП с группой из S объектов. В этом случае необходимо решать задачу оптимального планирования работы ИП с объектами. Рассмотрим ее в следующей конкретной постановке.

Предположим, что взаимодействие объекта с ИП может осуществляться на интервале $[t_{x_i}, t_{y_i}]$ в течение промежутков времени τ_i^l произвольной длительности (рис. 1.1), т. е. работа ИП

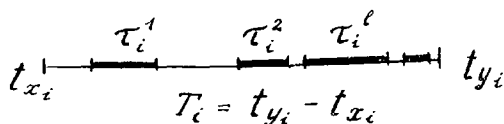


Рис. 1.1. Интервал взаимодействия

с объектом на $[t_{x_i}, t_{y_i}]$ может прерываться. Обозначим суммарную длину интервала τ_i^l через x_i :

$$x_i = \sum_l \tau_i^l. \quad (1.1)$$

Таким образом, x_i — суммарное время взаимодействия i -го объекта с ИП в зоне радиовидимости $[t_{x_i}, t_{y_i}]$.

Предположим, что на x_i наложено ограничение $x_i \leq \tau_i$, т. е. x_i может принимать значения от 0 до τ_i , причем $\tau_i \leq T_i$. Сформулируем теперь задачу оптимального планирования взаимодействия объектов и ИП как задачу выбора значений x_i ($i = \overline{1, m}$), доставляющих максимум функции

$$E = \sum_{i=1}^m \alpha_i e_i(x_i) \quad (1.2)$$

при ограничениях вида

$$0 \leq x_i \leq \tau_i; \quad (1.3)$$

$$\tau_i \leq T_i; \quad (1.4)$$

$$i = \overline{1, m},$$

где $e_i(x_i)$ — функция, выражающая положительный эффект от взаимодействия i -го объекта с ИП в зависимости от времени x_i , отводимого на данное взаимодействие. В частности, $e_i(x_i)$ может выражаться через объем информации, передаваемой с борта объекта на ИП за время x_i (например, $e_i(x_i) = V_i x_i$, где V_i — скорость передачи информации); α_i — коэффициент относительной важности i -го объекта.

В указанные выше ограничения (1.3), (1.4) в явном виде не входит ограничение на количество одновременно взаимодействующих с ИП объектов, что является наиболее существенным фактором при формализации и решении поставленной задачи планирования. Кроме того, нами пока не определен интервал, на котором производится планирование. Определим интервал, рассмотрев диаграмму, иллюстрирующую расположение зон радиовидимости объектов относительно ИП (рис. 1.2).

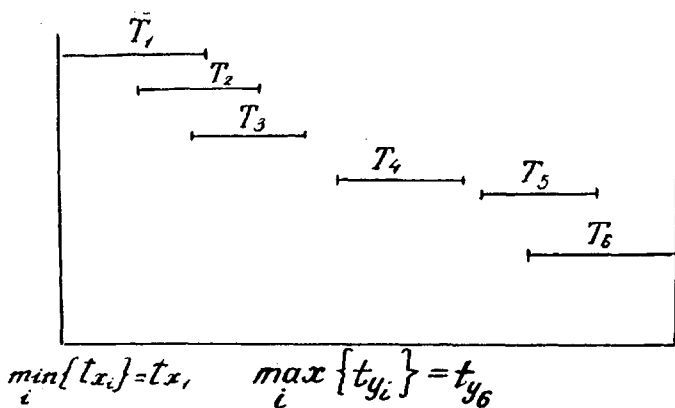


Рис. 1.2. Диаграмма зон радиовидимости

В представленном примере мы имеем шесть объектов со своими зонами радиовидимости $T_1, T_2, T_3, \dots, T_6$. В данном случае весь рассматриваемый интервал возможного взаимодействия m объектов ($m = 6$) с ИП $[t_{x_1}, t_{y_6}]$ можно разбить на три независимых интервала:

$$[t_{x_1}, t_{y_3}];$$

$$[t_{x_4}, t_{y_4}];$$

$$[t_{x_5}, t_{y_6}].$$

Их можно рассматривать как самостоятельные интервалы планирования. Причем, если $k = 2$, то сформулированная задача имеет смысл только для первого из указанных интервалов, так как в этом случае $s > k$.

Таким образом, для формирования интервалов планирования необходимо выполнить следующие операции:

найти интервал возможного взаимодействия m объектов с ИП:

$$[\min_i \{t_{x_i}\}, \max_i \{t_{y_i}\}];$$

разбить найденный интервал на независимые (непересекающиеся) интервалы, на которых, в свою очередь, наблюдается пересечение до $s > k$ зон радиовидимости;

каждый из найденных таким образом интервалов будет являться интервалом, на котором необходимо решить задачу оптимального планирования (1.2) — (1.4).

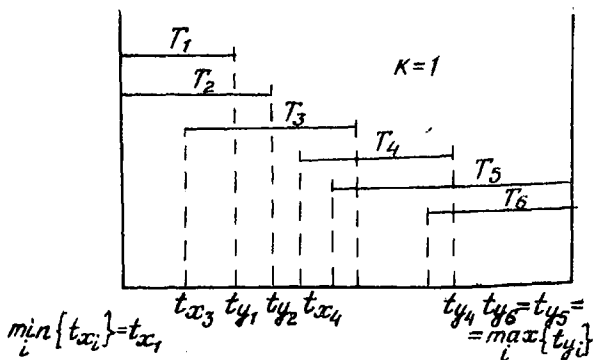


Рис. 1.3. Интервал планирования

Для решения задачи (1.2) — (1.4) можно предложить квази-оптимальный алгоритм следующего вида:

1. Разбить интервал $[\min_i \{t_{x_i}\}, \max_i \{t_{y_i}\}]$ на последовательность смежных интервалов, границы которых определяются моментами t_{x_i} , t_{y_i} . В частности, для рассматриваемого примера (рис. 1.3) будем иметь интервалы $[t_{x_1}, t_{x_3}]$, $[t_{x_3}, t_{y_1}]$, $[t_{y_1}, t_{y_2}]$, ..., $[t_{y_4}, t_{y_6}]$.

2. Задачу планирования (1.2) — (1.4) будем решать последовательно для полученных интервалов, начиная с первого. Результаты планирования для каждого из интервалов являются исходными данными для последующего интервала планирования.

Рассмотрим произвольный j -й интервал. Пусть его длина r_j . Обозначим через S_j число пересекающихся зон радиовидимости на данном интервале, а через x_i^j — аналог переменной x_i для j -го интервала. Введем в рассмотрение функцию

$$E_j = \sum_{i=1}^{S_j} e_i(x_i^j).$$

Ограничения для x_i^j имеют вид

$$x_i^j \leq \tau_i^j, \quad x_i^j \leq r_j, \quad (1.5)$$

$$\sum_{i=1}^{S_j} x_i^j \leq k r_j, \quad (1.6)$$

где

$$\tau_i^j = \tau_i - \sum_{l=1}^{j-1} x_i^l.$$

Ограничения (1.5), (1.6) равносильны условию работы ИП одновременно с k объектами на интервале r_j . В самом деле, если $k=1$, то распределяемый ресурс для x_i^j ($i=1, S_j$) равен r_j , т. е. величине

интервала. Если $k = s_j$, то фактически для каждого объекта отводится интервал планирования величиной r_j , т. е. каждый параметр x_i^j может принимать значение от 0 до r_j . Если $1 < k < s_j$, то kr_j — суммарный временной ресурс планирования для s_j объектов, гарантирующий работу ИП одновременно с k объектами.

Задачу нахождения $\max E_j$ можно решить, воспользовавшись процедурой динамического программирования, описанной в 3-й главе, введя дискретизацию переменных x_i^j .

3. В результате последовательного решения задачи нахождения $\max E_j$ для всех интервалов r_j получим искомые значения

$$x_i = \sum_j x_i^j, \quad i = \overline{1, m}. \quad (1.7)$$

Квазиоптимальность предложенного алгоритма обусловлена тем, что $\sum_j \max E_j$ не обязательно является глобальным максимумом E .

Решение задачи нахождения $\max E$ прямым способом при нелинейных $e_i(x_i)$ требует существенных затрат вычислительного характера.

В том случае, если $e_i(x_i)$ — линейные функции вида $V_i x_i$, задача поиска $\max E$ может рассматриваться как задача линейного программирования с целевой функцией

$$E = \sum_{j=1}^d \sum_{i \in j}^{s_j} V_i x_i^j,$$

где на переменные x_i^j ($i = \overline{1, m}; j = \overline{1, d}$), наложены следующие ограничения:

$$\sum_{j=1}^d x_i^j \leq v_i;$$

$$\sum_{i \in j}^{s_j} x_i^j \leq kr_j,$$

где $i \in j$ обозначает, что i -й объект расположен в j -м интервале времени длиной r_j ; d — количество таких интервалов.

Для упрощения записи можно представить указанную выше задачу в одноиндексном виде, введя для всех x_i^j сквозную нумерацию.

ТОПОЛОГИЧЕСКАЯ ОПТИМИЗАЦИЯ СИСТЕМ СБОРА, ПЕРЕДАЧИ И ОБРАБОТКИ ИНФОРМАЦИИ

2.1. ЗАДАЧИ ТОПОЛОГИЧЕСКОЙ ОПТИМИЗАЦИИ

Непрерывное усложнение систем сбора, передачи и обработки информации (ССПОИ) неизбежно приводит к увеличению удельного веса межэлементной сети связи в общем балансе материальных затрат на создание систем. Поэтому проблема пространственной организации ССПОИ (рис. 2.1) и связанные с ней вопросы оптимального размещения ее компонентов становятся все более актуальными.

Под *топологической оптимизацией* ССПОИ понимают построение топологической схемы системы, обладающей сетью связи минимальной стоимости. Топологическая схема ССПОИ

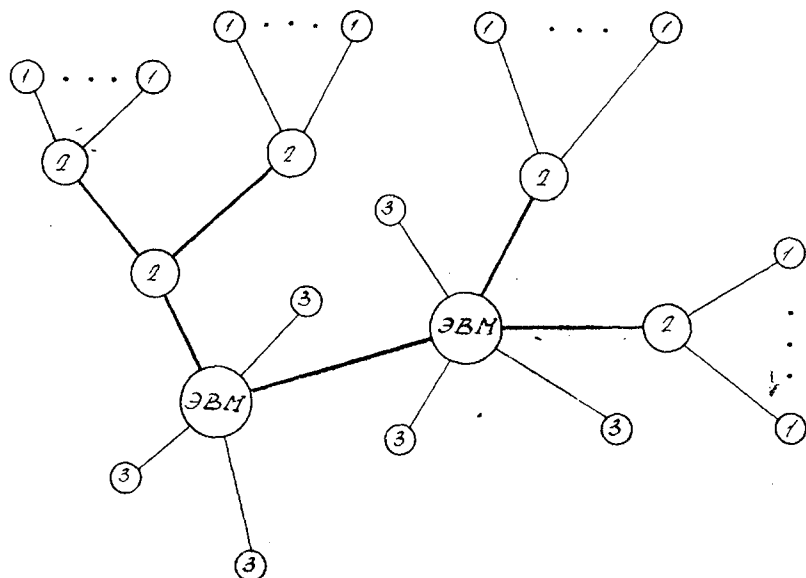


Рис. 2.1. Структурная схема ССПОИ: 1 — измерительные пункты (ИП); 2 — узлы коммутации (УК); 3 — абонентские пункты (АП)

характеризует пространственную организацию системы, а именно: топологию сети связи, размещение компонентов системы в пространстве, а также трассировку линий связи. Под *топологией сети связи* понимают структуру соединений объектов системы, т. е. совокупность отношений, показывающих, какой объект системы с каким объектом связан и какова сложность этой связи. Топология сети связи описывает структуру соединений, но не указывает положение объектов системы в пространстве. Это положение описывает план размещения, содержащий координаты объектов системы в пространстве. Описание пространственной организации системы будет не полным, если мы не зададим трассировку линий связи. *Трассировка линий связи* — это геометрически определенный способ соединения объектов системы. Трассировка имеет *метрический и топологический* аспекты. Первый аспект связан с выбором метрики для оценки расстояния между соединяемыми объектами, второй обусловлен ограничениями на прокладку линий связи в монтажном пространстве.

В процессе топологической оптимизации ССПОИ необходимо решить следующие задачи:

синтезировать топологию сети связи ССПОИ;

выбрать модель трассировки линий связи;

разместить объекты системы в пространстве по критерию минимума стоимости сети связи.

Ясно, что указанные задачи взаимосвязаны. Их решение начнем с построения модели топологической схемы ССПОИ.

2.2. МОДЕЛЬ ТОПОЛОГИЧЕСКОЙ СХЕМЫ ССПОИ

Для обеспечения электрической связи между объектами системы достаточно задать конфигурацию межобъектных соединений в виде дерева. Поэтому модель сети связи ССПОИ можно представить в виде древовидной иерархической системы (рис. 2.2), нулевой уровень которой образуют неподвижные объекты системы (измерительные и абонентские пункты), а остальные R уровней иерархии составляют объекты, положением которых можно варьировать (в данном случае это ЭВМ и узлы коммутации). Объекты нулевого уровня будем называть *неперемещаемыми* объектами, а объекты остальных уровней — *перемещаемыми*.

Если уровни иерархии, имеющие неполный набор модулей, доукомплектовать так называемыми псевдообъектами (ПО) и пронумеровать все объекты системы на каждом r уровне ($r = 0, 1, 2, \dots, R$) от 1 до M_r , включая ПО, то сеть связи ССПОИ можно описать следующими матрицами: U_0, U, E, S [42].

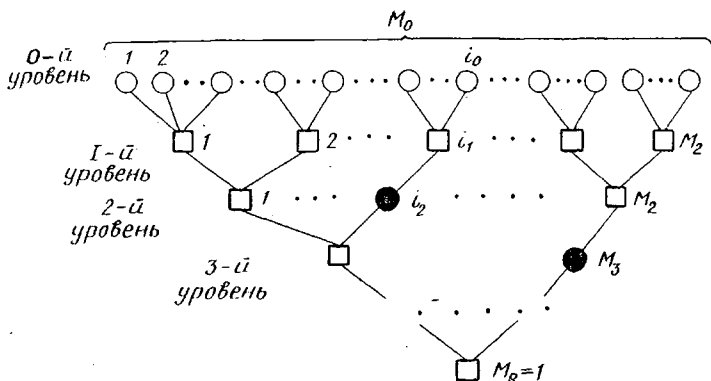


Рис. 2.2. Модель сети связи неархивической системы; ○ — перемещаемые объекты; □ — перемещаемые объекты; ● — псевдообъекты

где $U = [u_{oi_0}]$ — матрица координат неподвижных объектов системы; здесь $u_{oi_0} = [x_{oi_0}, y_{oi_0}, z_{oi_0}]$ — координаты объекта с номером i_0 в системе координат монтажного пространства; $\vec{U} = [u_{ri_r}]$ — матрица координат перемещаемых объектов системы; здесь $u_{ri_r} = [x_{ri_r}, y_{ri_r}, z_{ri_r}]$ — координаты объекта r уровня ($r = \overline{1, R}$) с номером i_r ($i_r = \overline{1, M_r}$);

$E = [e_{ri_r}]$ — матрица связанности; здесь e_{ri_r} ($r = \overline{1, R}$, $i_r = \overline{1, M_r}$) — количество объектов $(r-1)$ -го уровня, связанных с i_r объектом r уровня. Указанные объекты $(r-1)$ -го уровня имеют номера, принадлежащие следующему индексному множеству:

$$J(r-1, i_r) = \{k+1, k+2, \dots, k+e_{ri_r}\}, \quad k = \sum_{j=1}^{i_r-1} e_{rj}, \quad (2.1)$$

$S = [s_{ri_r}]$ — матрица удельных стоимостей линий связи; здесь s_{ri_r} — стоимость единицы длины линии связи [ед/м], исходящей от i_r -го объекта r уровня ($i_r = \overline{1, M_r}$, $r = \overline{0, R}$). Часто нулевую строку этой матрицы задают отдельно матрицей $S_0 = [s_{0i_0}]$.

Матрицы E и S описывают топологию сети связи ССПОИ, матрицы U_0 и U задают план размещения объектов системы в монтажном пространстве.

Однако для оценки стоимости сети связи ССПОИ помимо топологии и плана размещения объектов необходимо знать модель трассировки межобъектных линий связи. Как уже отмечалось выше, трассировка линий связи имеет топологический и метрический аспекты. Метрический аспект обусловлен метрикой

(или функцией), согласно которой оценивается расстояние между соединяемыми объектами. Наиболее популярны и наиболее изучены метрики вида l_p , $p \geq 1$, определяемые формулой

$$l_{ij} = (|x_i - x_j|^p + |y_i - y_j|^p + |z_i - z_j|^p)^{1/p}, \quad (2.2)$$

а также чебышевская метрика, определяемая формулой

$$l_{ij} = \max(|x_i - x_j|, |y_i - y_j|, |z_i - z_j|).$$

Здесь l_{ij} — расстояние между объектами i и j в трехмерном пространстве.

Заметим, что при $p = 2$ это обычная евклидова метрика, а при $p = 1$ — ортогональная метрика, называемая также прямоугольной метрикой или метрикой городских кварталов. В реальной ситуации неизбежны отклонения от идеализированной модели, какой является та или другая метрика, как по соображениям экономического характера, так и в связи с ограничениями на трассировку (например в случае, когда можно «срезать» угол или требуется обойти препятствие). Для учета таких отклонений иногда предлагают использовать в качестве метрики линейную комбинацию ортогональной и чебышевской метрик. Однако использование взвешенной метрики для расчета топологических схем сложных ССПОИ приводит к непреодолимым трудностям вычислительного характера.

В настоящей главе будут рассматриваться вопросы размещения объектов ССПОИ в двухмерном пространстве. Для оценки расстояний между объектами системы будут использоваться ортогональная и евклидова метрики

$$l_{ij} = |x_i - x_j| + |y_i - y_j| = \|\vec{u}_i - \vec{u}_j\|_1, \quad (2.3)$$

$$l_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2.4)$$

в зависимости от конкретной ситуации.

Матрицы E , S , U_0 , U совместно с метриками (2.3), (2.4) задают модель топологической схемы ССПОИ.

2.3. ОБЩАЯ ПОСТАНОВКА ЗАДАЧИ

В общем виде задача синтеза топологической схемы ССПОИ (см. рис. 2.1) формулируется следующим образом:

определить количество, размещение и структуру соединений ЭВМ и узлов коммутации (УК) при минимуме стоимости сети связи системы, чтобы: 1 — все объекты системы лежали в разрешенных областях; 2 — все ИП были связаны с ЭВМ либо напрямую, либо через УК; 3 — выполнялись ограничения на коммутационную способность каждого УК и пропускные способности линий связи.

В символьном виде эта задача может быть записана так:

$$\min_{E, S, U} Q_{\text{сети}}(E, S, U) \quad (2.5)$$

$$\{E, S\} \in D_{ES} \quad (2.6)$$

$$U \in D_u. \quad (2.7)$$

Здесь $Q_{\text{сети}}$ — стоимость сети связи как функция от топологии сети связи системы и плана размещения ее объектов;

D_{ES} — множество возможных топологий сети связи ССПОИ, обусловленное реальными возможностями интерфейса, коммутационной способностью УК и пропускными способностями линий связи;

D_u — область метрического пространства, доступная для размещения модулей системы (монтажное пространство).

Таким образом, задача синтеза топологической схемы заключается в определении топологии сети связи системы (E, S) и плана размещения ее объектов (U). Условия реального проектирования исключают возможность совместного решения указанных задач и предопределяют последовательно-итерационную процедуру проведения метрико-топологических преобразований сети связи системы.

Один из возможных подходов к решению задачи (2.4) — (2.6) заключается в выполнении следующих этапов:

Этап 1. Определение начальной топологии ССПОИ.

Этап 2. Оптимальное размещение объектов ССПОИ в монтажном пространстве.

Этап 3. Коррекция топологии сети связи ССПОИ с возвратом на этап 2.

Этапы 2 и 3 повторяем циклически до тех пор, пока не получим установившееся решение.

В настоящей главе будем придерживаться указанной схемы решения. Изложение материала построим в последовательности по мере возрастания трудности решения: сначала рассмотрим задачу размещения системы с заданной топологией сети связи и без пространственных ограничений, далее рассмотрим задачу размещения с пространственными ограничениями (2.7), после чего перейдем к собственно задаче синтеза топологической схемы ССПОИ. Такое построение материала целесообразно с двух точек зрения, так как, во-первых, позволяет излагать материал, наращивая его сложность, во-вторых, обеспечивает преемственность методов и алгоритмов.

2.4. БЕЗУСЛОВНОЕ ОПТИМАЛЬНОЕ РАЗМЕЩЕНИЕ

Постановка задачи

При оптимизации ССПОИ варьируемым параметром является только матрица U физических координат объектов системы, а состав аппаратуры и структура электрических соединений объектов системы (топология) считаются известными. Задача оптимального размещения непосредственно вытекает из общей задачи синтеза топологической схемы (2.5) — (2.7) при условии, что матрицы E и S известны, и формулируется следующим образом.

Найти такой \overline{U} план размещения объектов системы $U = [u_{ri_r}]$, $r = \overline{1, R}$, $i_r = \overline{1, M_r}$, для которого обобщенная стоимость сети связи

$$Q_{\text{сети}}(U) = \sum_{i_{R-1} \in J(R-1, i_R)} \{ \dots \sum_{i_1 \in J(1, i_2)} \{ \dots \sum_{i_0 \in J(0, i_1)} s_0 i_0 \times \\ \times \| \vec{u}_{i_0} - \vec{u}_{i_1} \|^{p_1} s_1 \| \vec{u}_{i_1} - \vec{u}_{i_2} \| \} + \dots \} + s_{R-1, i_{R-1}} \times \\ \times \| \vec{u}_{R-1, i_{R-1}} - \vec{u}_{R, 1} \| \quad (2.8)$$

достигает минимального значения и ограничения на размещение объектов выполняются:

$$V_p(\vec{u}_{ri_r}) \leq 0, \quad \forall (r = \overline{1, R}, i_r = \overline{1, M_r}), \quad \exists p = \overline{1, P}. \quad (2.9)$$

Задача (2.8) — (2.9) является задачей нелинейного программирования с ограничениями, образующими невыпуклую область допустимых решений. Функция $Q_{\text{сети}}(U)$ является многоэкстремальной, кроме того, решение задачи зависит от вида используемой метрики (2.2).

Предлагаемый в настоящей главе метод решения задачи (2.8) — (2.9) заключается в поиске безусловного оптимального плана U^{opt} , минимизирующего (2.8) для метрики (2.3) при снятых пространственных ограничениях (2.9), и коррекции этого решения специальными методами на область допустимых решений (2.9).

Метод безусловного оптимального размещения

Обозначим $Q_{ri_r}(\vec{u}_{r+1, i_{r+1}})$ стоимость фрагмента сети связи, связанного с i_r -м объектом r уровня иерархии сети связи, как функцию положения i_{r+1} -го объекта $(r+1)$ -го уровня иерархии. Здесь $i_r \in J(r, i_{r+1})$, $i_{r+1} = 1, 2, \dots, M_{r+1}$. Объект с номером i_{r+1} является вершиной указанного фрагмента. Показан

пример разбиения сети связи на фрагменты (рис. 2.3). Тогда суммарную стоимость сети связи можно представить следующим образом:

$$Q_{\text{сети}}(\vec{u}_{R,1}) = \sum_{l_{R-1}=1}^{M_{R-1}} Q_{R-1, l_{R-1}}(\vec{u}_{R,1}).$$

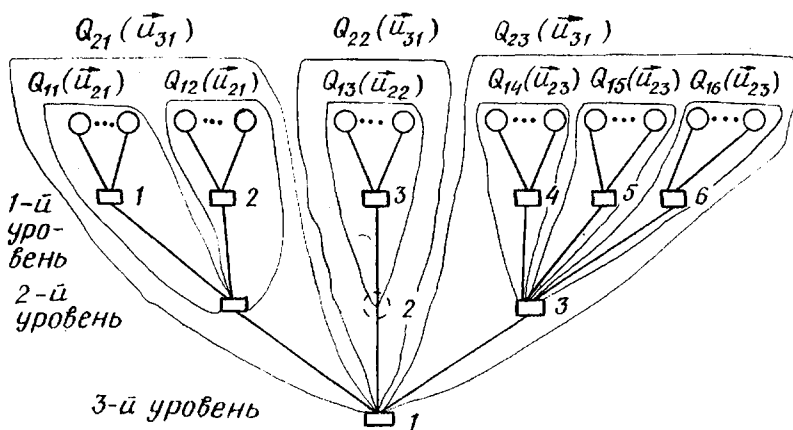


Рис. 2.3. Пример разбиения сети связи на фрагменты

Суть метода оптимального размещения заключается в сведении многопараметрической функции сети связи (2.8) к однопараметрической $Q_{\text{сети}}^{\min}(\vec{u}_{R,1})$, которая для каждого значения $\vec{u}_{R,1}$ имеет минимально возможное значение стоимости сети связи. Этот переход производится последовательно за R шагов в предположении, что на каждом r -м шаге ($r = \overline{1, R}$) все фрагменты имеют минимальную стоимость для данного положения вершины фрагмента. Оптимальные координаты объектов системы определяют после минимизации функции $Q_{\text{сети}}^{\min}(\vec{u}_{R,1})$.

Если метрика, по которой подсчитывают длины межобъектных линий связи, является ортогональной (2.3), то указанный переход сделать наиболее просто. В этом случае, в силу специфики метрики,

$$\min_U Q_{\text{сети}}(U) = \min_X Q_{\text{сети}}(X) + \min_Y Q_{\text{сети}}(Y),$$

и общая процедура оптимизации разбивается на 2 независимые процедуры по осям x и y соответственно (здесь X, Y — матрица абсцисс и ординат объектов системы соответственно).

Ниже приведена запись метода оптимального размещения для оси x в виде рекуррентной процедуры оптимизации [43]. Для простоты записи производные $dQ_{ri_l}^{\min}(x_{r+1}, i_{r+1})/dx_{r+1}, i_{r+1}$ обозначены как $\dot{Q}_{ri_r}^{\min}(x_{r+1}, i_{r+1})$.

Начальные условия: $r = 1$,

$$\dot{Q}_{0i_0}^{\min}(x_{1i_1}) = \begin{cases} -s_0 i_0 & \text{при } x_{1i_1} < x_{0i_0} \\ +s_0 i_0 & \text{при } x_{1i_1} \geq x_{0i_0} \end{cases} \quad (2.10)$$

Далее, для $r = 1, 2, \dots, R$ последовательно находим:

а) производные от стоимости фрагментов:

$$\dot{Q}_{ri_r}(x_{ri_r}) = \sum_{i_{r-1} \in J(r-1, i_r)} \dot{Q}_{r-1, i_{r-1}}^{\min}(x_{ri_r}), \quad (2.11)$$

$$i_r = 1, 2, \dots, M_r;$$

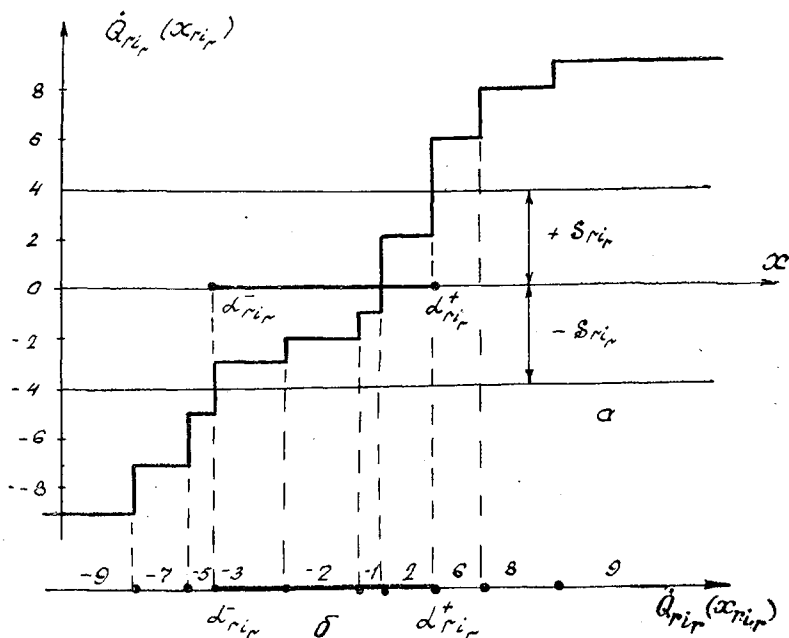


Рис. 2.4. Производная от стоимости фрагмента сети связи: а — в системе координат; б — на отрезке прямой

б) $\alpha_{ri_r}^+$, $\alpha_{ri_r}^-$ — корни уравнений (рис. 2.4,а)

$$\dot{Q}_{ri_r}(x_{ri_r}) = (\pm) s_{ri_r}, \quad i_r = 1, 2, \dots, M_r; \quad (2.12)$$

в) координаты объектов, минимизирующие стоимости соот-

ветствующих фрагментов сети связи как функции от положения вершин этих фрагментов:

$$x_{ri_r}^{\text{opt}}(x_{r+1}, i_{r+1}) = \begin{cases} \alpha_{ri_r}^-, & \text{при } x_{r+1}, i_{r+1} < \alpha_{ri_r}^-; \\ x_{r+1}, i_{r+1}, & \text{при } x_{r+1}, i_{r+1} \in [\alpha_{ri_r}^-, \alpha_{ri_r}^+]; \\ \alpha_{ri_r}^+, & \text{при } x_{r+1}, i_{r+1} > \alpha_{ri_r}^+; \end{cases} \quad (2.13)$$

$$i_r \in J(r, i_{r+1}), \quad i_{r+1} = \overline{1, M_{r+1}},$$

г) производные от стоимостей фрагментов сети связи как функции координат положения вершин этих фрагментов (см. рис. 2.4,а): $Q_{ri_r}^{\min}(x_{r+1}, i_{r+1}) =$

$$= \begin{cases} -s_{ri_r}, & \text{при } x_{r+1}, i_{r+1} < \alpha_{ri_r}^-; \\ Q_{ri_r}(x_{ri_r}) \mid x_{ri_r} = x_{r+1}, i_{r+1}, & \text{при } x_{r+1}, i_{r+1} \in [\alpha_{ri_r}^-, \alpha_{ri_r}^+]; \\ +s_{ri_r}, & \text{при } x_{r+1}, i_{r+1} > \alpha_{ri_r}^+; \end{cases} \quad (2.14)$$

$$i_r \in J(r, i_{r+1}), \quad i_{r+1} = \overline{1, M_{r+1}}.$$

После указанных преобразований на R шаге процедуры оптимизации в пункте «а» получим производную от стоимости сети связи всей системы в виде суммы производных от стоимостей фрагментов сети, вершинами которых является объект R уровня:

$$Q_{\text{сети}}^{\min}(x_{R,1}) = \sum_{i_{R-1}=1}^{M_{R-1}} Q_{R-1, i_{R-1}}^{\min}(x_{R,1}).$$

Поскольку R -й уровень иерархии сети связи является последним, то $s_{R,1} = 0$.

Решая в пункте «б» уравнение

$$Q_{\text{сети}}^{\min}(x_{R,1}) = 0, \quad (2.15)$$

находим оптимальное значение абсциссы объекта R уровня: $x_{R,1}^{\text{opt}}$. (Следует отметить, что условию (2.15) может соответствовать целый интервал оси x . Обозначим границы этого интервала α_{R1}^- , α_{R1}^+ , тогда в качестве x_{R1}^{opt} можно брать любое значение из интервала $[\alpha_{R1}^-, \alpha_{R1}^+]$). Подставляя значение x_{R1}^{opt} в рекуррентное соотношение (2.13), получим оптимальные абсциссы объектов $(R-1)$ -го уровня:

$$x_{R-1, i_{R-1}}^{\text{opt}} = x_{R-1, i_{R-1}}^{\text{opt}}(x_{R,1}^{\text{opt}}), \quad i_{R-1} = \overline{1, M_{R-1}}.$$

Затем для каждого значения $r = R-2, R-3, \dots, 1$, путем последовательных подстановок каждого нового значения в выражение (2.13), находим оптимальные абсциссы объектов системы на всех уровнях иерархии:

$$x_{ri_r}^{\text{opt}} = x_{ri_r}^{\text{opt}}(x_{r+1}, i_{r+1}),$$

$$i_{r+1} = \overline{1, M_{r+1}}, i_r \in J(r, i_{r+1}),$$

Процедура оптимизации по оси y полностью повторяет рассмотренную.

В результате будем иметь план размещения объектов системы

$$U^{\text{opt}} = [\vec{u}_{ri_r}^{\text{opt}}], \quad (2.16)$$

имеющий минимальную стоимость сети связи для ортогональной метрики (2.3) [43]. Если рассматривать полученный «ортогональный» план (2.16) при евклидовом способе оценки межобъектных расстояний (2.4), то он, в общем случае, будет иметь большую стоимость сети связи, чем чисто «евклидовый» план, полученный при оптимизации функции (2.8) с евклидовыми расстояниями. Однако с практической точки зрения в большинстве случаев «ортогональный» план с евклидовыми расстояниями оказывается предпочтительнее чисто «евклидового» плана. Это вызвано тем, что, во-первых, линии связи никогда не прокладываются внатяг, т. е. необходима некоторая технологическая «слабина» линий связи, которая есть у «ортогонального» плана и полностью отсутствует у «евклидового»; во-вторых, «ортогональный» план более геометрически правильный, чем «евклидовый», что является существенным аргументом в его пользу при трассировке коммуникаций.

Рассмотренную процедуру поиска «ортогонального» плана (2.10) — (2.14) целесообразно выполнять графически. При этом производные от фрагментов сети связи удобно изображать отрезками прямой с указанием на каждом из них значений производной (рис. 2.4,б) — это упрощает и делает наглядным процесс суммирования и решения уравнений (2.12).

Пример [45]. На рис. 2.5 приведена схема электрических соединений ССПОИ. Координаты измерительных пунктов (ИП),

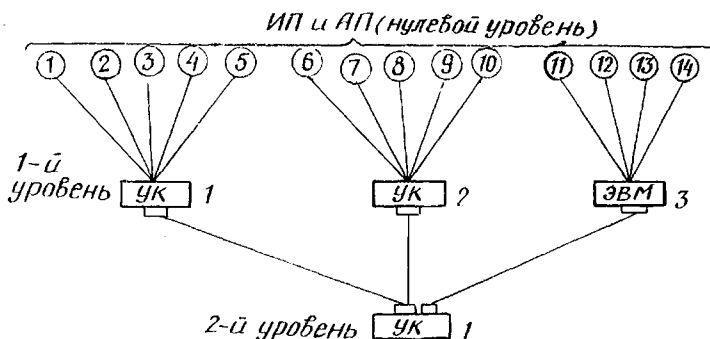


Рис. 2.5. Структура сети связи ССПОИ (к примеру)

Таблица 2.1

Метрические характеристики ИП и АП

Номер устройства	Наименование	Координаты, м		Удельная стоимость линий связи, ед/м
		x_{0i_0}	y_{0i_0}	
1	ИП	3	8	1
2	»	3	10	1
3	»	7	14	1
4	»	5	5	1
5	»	9	11	1
6	»	8	16	1
7	»	10	18	1
8	»	20	13	1
9	»	21	18	1
10	»	22	12	1
11	АП	11	3	1
12	»	12	6	1
13	»	16	2	1
14	»	18	7	1

абонентских пунктов (АП) и стоимостные характеристики линий связи приведены в табл. 2.1. Удельная стоимость магистрали равна 3 ед/м.

Требуется найти такое положение узлов коммутации (УК) и ЭВМ, при котором обеспечивается минимум стоимости всей сети связи.

Решение. Матрицы связанности и удельной сложности для данной системы имеют следующий вид:

$$E = \begin{bmatrix} 5 & 5 & 4 \\ 3 \\ 1 \end{bmatrix} \quad S = \begin{bmatrix} 3 & 3 & 3 \end{bmatrix}$$

Индексные множества находим по формуле (2.1):

$$J(0,1) = \{1, 2, 3, 4, 5\}, \quad J(0,2) = \{6, 7, 8, 9, 10\},$$

$$J(0,3) = \{11, 12, 13, 14\}, \quad J(1,1) = \{1, 2, 3\}.$$

Оптимизацию начнем с поиска оптимальных абсцисс УК и ЭВМ. Для каждого i_1 -го объекта 1-го уровня ($i_1 = 1, 2, 3$), со-

Согласно (2.11) находим производные от стоимости соответствующих фрагментов

$$Q_{1i_1}(x_{1i_1}) = \sum_{i_0 \in J(0, i_1)} Q_{0i_0}^{\min}(x_{1i_1}).$$

Эти производные будем обозначать отрезками оси с указанием значений производной на каждом из них (см. рис. 2.4). Решая уравнение $Q_{1i_1}(x_{1i_1}) = (\pm) S_{1i_1} = \pm 3$ для $i_1 = 1, 2, 3$ последовательно находим корни:

$$\begin{array}{ll} \alpha_{11}^- = 3, & \alpha_{11}^+ = 7; \\ \alpha_{12}^- = 10, & \alpha_{12}^+ = 21; \\ \alpha_{13}^- = 11, & \alpha_{13}^+ = 18 \text{ (рис. 2.6, а, б, в)}. \end{array}$$

Согласно выражению (2.13) оптимальные абсциссы $x_{1i_1}^{\text{opt}}(x_{21})$, $i_1 = 1, 2, 3$ равны

$$\begin{aligned} x_{11}^{\text{opt}}(x_{21}) &= \begin{cases} 3, & x_2 < 3; \\ x_2, & x_2 \in [3, 7]; \\ 7, & x_2 > 7; \end{cases} \\ x_{12}^{\text{opt}}(x_{21}) &= \begin{cases} 10, & x_2 < 10; \\ x_2, & x_2 \in [10, 21]; \\ 21, & x_2 > 21; \end{cases} \\ x_{13}^{\text{opt}}(x_{21}) &= \begin{cases} 11, & x_2 < 11; \\ x_2, & x_2 \in [11, 18]; \\ 18, & x_2 > 18. \end{cases} \end{aligned}$$

Производные от стоимостей фрагментов равны (рис. 2.6, г, д, е):

$$\begin{aligned} Q_{11}^{\min}(x_{21}) &= \begin{cases} -3, & x_{21} < 3; \\ Q_{11}(x_{11})|_{x_{11}=x_{21}}, & x_{21} \in [3, 7]; \\ +3, & x_{21} > 7; \end{cases} \\ Q_{12}^{\min}(x_{21}) &= \begin{cases} -3, & x_{21} < 10; \\ Q_{12}(x_{12})|_{x_{12}=x_{21}}, & x_{21} \in [10, 21]; \\ +3, & x_{21} > 21; \end{cases} \\ Q_{13}^{\min}(x_{21}) &= \begin{cases} -3, & x_{21} < 11; \\ Q_{13}(x_{13})|_{x_{13}=x_{21}}, & x_{21} \in [11, 18]; \\ +3, & x_{21} > 18. \end{cases} \end{aligned}$$

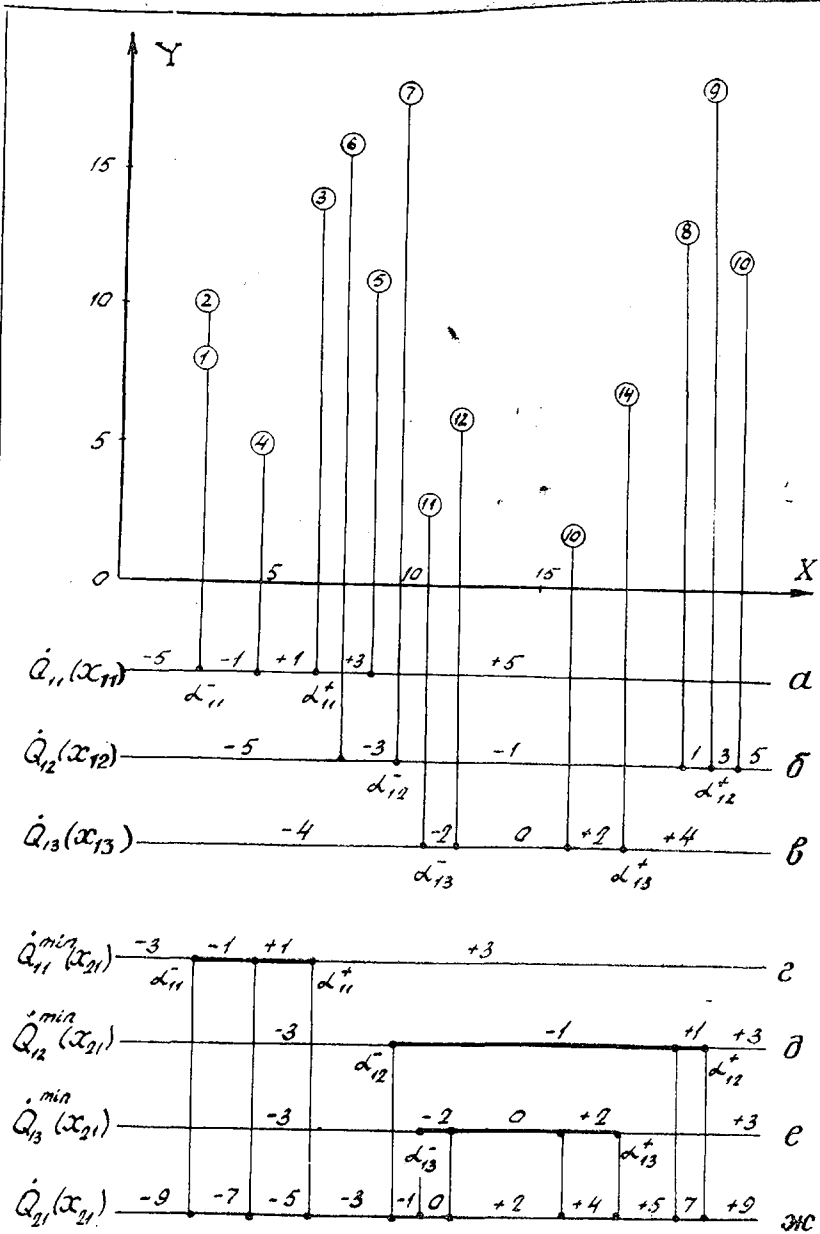


Рис. 2.6. Пример расчета оптимальных координат объектов системы по оси x

2.5. РАЗМЕЩЕНИЕ ПРИ ПРОСТРАНСТВЕННЫХ ОГРАНИЧЕНИЯХ

В отличие от задачи безусловного оптимального размещения, решение которой в силу метрики (2.3) распалось на две независимые процедуры по соответствующим осям координат, задача (2.8) — (2.9) является задачей векторной оптимизации в том смысле, что компоненты векторов $\vec{U}_{ri_r} = [x_{ri_r}, y_{ri_r}]$, $r = \overline{1, R}$, $i_r = \overline{1, M_r}$ связаны пространственными ограничениями (2.9). Последнее не позволяет получить простые соотношения, подобные (2.10) — (2.14).

Будем искать решение задачи (2.8) — (2.9) как решение следующей рекуррентной задачи оптимизации:

$$\min_{\substack{\vec{u}_{ri_r} \in D_u \\ i_{r-1} \in J(r-1, i_r)}} \left\{ \sum_{i_{r-1} \in J(r-1, i_r)} s_{r-1, i_{r-1}} \|\vec{u}_{r-1, i_{r-1}} - \vec{u}_{ri_r}\| + s_{ri_r} \|\vec{u}_{ri_r} - \vec{u}_{r+1, i_{r+1}}\| \right\}, \quad r = \overline{1, R}, \quad i_r \in J(r, i_{r+1}), \quad i_{r+1} = \overline{1, M_{r+1}}. \quad (2.17)$$

Здесь \vec{u}_{ri_r} — вектор координат объектов размещаемого уровня, $\vec{u}_{r-1, i_{r-1}}$, и $\vec{u}_{r+1, i_{r+1}}$ — оценки векторов координат объектов смежных уровней.

Выбор оценок $\vec{u}_{r-1, i_{r-1}}$ и $\vec{u}_{r+1, i_{r+1}}$ зависит от того, каким образом при коррекции осуществляется движение по уровням системы — от ИС к объекту верхнего уровня или наоборот. Если движение начинается от объектов первого уровня, т. е. $r = 1, 2, \dots, R$, то на каждом шаге r решения рекуррентной задачи (2.17) в качестве оценок векторов координат объектов смежных уровней целесообразно выбрать следующие:

$$\begin{aligned} \vec{u}_{r-1, i_{r-1}} &= \vec{u}_{r-1, i_{r-1}}^*, \\ \vec{u}_{r+1, i_{r+1}} &= \vec{u}_{r+1, i_{r+1}}^{\text{opt}}, \end{aligned}$$

где $\vec{u}_{r-1, i_{r-1}}^*$ — решение задачи (2.17) на предыдущем шаге $r-1$;

$\vec{u}_{r+1, i_{r+1}}^{\text{opt}}$ — элементы $(r+1)$ -й строки безусловной оптимальной матрицы (2.16).

Если коррекция начинается с определения допустимых координат объекта уровня R , т. е. $r = R, R-1, \dots, 1$, то на каждом r -м шаге решения задачи (2.17) в качестве оценок

\vec{u}_{r-1}, i_{r-1} и \vec{u}_{r+1}, i_{r+1} следует брать векторы

$$\begin{cases} \widehat{\vec{u}}_{r-1}, i_{r-1} = \widehat{\vec{u}}_{r-1}^{\text{opt}}, i_{r-1}^{\text{opt}}, \\ \widehat{\vec{u}}_{r+1}, i_{r+1} = \vec{u}_{r+1}^{\text{opt}}, i_{r+1}^{\text{opt}}, \end{cases}$$

где $\vec{u}_{r-1}^{\text{opt}}, i_{r-1}^{\text{opt}}$ — элемент $(r-1)$ -й строки матрицы (2.16);

$\vec{u}_{r+1}^*, i_{r+1}^*$ — решение задачи (2.17) на предыдущем шаге.

Граничные условия для обоих алгоритмов

$$\begin{cases} \widehat{u}_0, i_0 = \vec{u}_0, i_0 \\ \widehat{u}_{R+1}, i_{R+1} = u_{R+1}^{\text{opt}}, i_{R+1}^{\text{opt}}. \end{cases}$$

Первый алгоритм коррекции будем обозначать $(1 \rightarrow R)$, а второй — $(R \rightarrow 1)$. Выражение в круглых скобках означает направление движения коррекции согласно иерархии сети связи.

Следует отметить, что в общем случае $u^{*(1 \rightarrow R)} \neq u^{*(R \rightarrow 1)}$, где $u^{*(1 \rightarrow R)}$ и $u^{*(R \rightarrow 1)}$ — допустимые решения задачи (2.8) — (2.9), полученные с помощью алгоритмов $(1 \rightarrow R)$ и $(R \rightarrow 1)$ соответственно. Эти решения могут быть существенно неоптимальны. Действительно, после алгоритмов коррекции $(1 \rightarrow R)$ и $(R \rightarrow 1)$ объекты ССПОИ в общем случае будут находиться в неравновесном состоянии из-за того, что решение задачи (2.17) на каждом r -м шаге ищут при условии неизменности оценок \vec{u}_{r-1}, i_{r-1} и \vec{u}_{r+1}, i_{r+1} . В самом деле, одна из этих оценок (в зависимости от направления коррекции) на следующем шаге задачи (2.17) сама является оптимизируемым параметром и может измеряться. Поэтому целесообразно еще раз проделать процедуру коррекции, но в направлении, противоположном предыдущему и т. д. до тех пор, пока не получим установившееся решение. Образно говоря, происходит «встряхивание» иерархической системы выведением из равновесия объектов исходного уровня ($r=R$ или $r=1$ — согласно направлению коррекции), в результате чего образуется «волна» перемещений объектов, которая доходит

до граничного уровня, «отражается» и движется в обратном направлении. Если модули исходного уровня для «отраженной» волны размещены неоптимальным образом, то движение возобновляется и так до тех пор, пока не образуется установившееся решение — «стоячая» волна. Так как при каждой элементарной коррекции ($1 \rightarrow R$) или ($R \rightarrow 1$) решается задача минимизации (2.17), то величина $Q_{\text{сети}}$ не может возрастать, т. е.

$$\begin{aligned} Q_{\text{сети}}(u^{*(R \rightarrow 1 \rightarrow \dots \rightarrow R)}) &\geq Q_{\text{сети}}(u^{*(R \rightarrow 1 \rightarrow \dots \rightarrow R \rightarrow 1)}), \\ Q_{\text{сети}}(u^{*(1 \rightarrow R \rightarrow \dots \rightarrow 1)}) &\geq Q_{\text{сети}}(u^{*(1 \rightarrow R \rightarrow \dots \rightarrow 1 \rightarrow R)}). \end{aligned}$$

Следует отметить, что установившееся решение $U^{*(\infty)}$ зависит от того, в каком направлении — ($R \rightarrow 1$) или ($1 \rightarrow R$) начата коррекция; алгоритмы ($R \rightarrow 1 \rightarrow R \rightarrow \dots$, $1 \rightarrow R \rightarrow 1 \rightarrow \dots$) в общем случае дадут различные результаты, причем, априори трудно отдать предпочтение тому или иному алгоритму коррекции. По этой причине при решении задачи (2.8)—(2.9) целесообразно использовать оба алгоритма. Окончательное решение определим при сравнении результатов.

Модели разрешенных областей

Нетрудно заметить, что на каждом этапе решения задачи (2.17) в монтажном пространстве D_u минимизируется функция вида

$$Q(\vec{u}) = \sum_{i \in I} s_i \|\vec{U}_i - \vec{u}\|, \quad (2.18)$$

т. е. определяется вектор $\vec{u}^* = [x^*, y^*] \in D_u$, при котором $Q(\vec{u})$ минимальна. Если область D_u многосвязна и представлена P разрешенными областями, то для поиска \vec{u}^* необходимо перебрать P «подозрительных» на экстремум точек, доставляющих минимум функции $Q(\vec{u})$ на соответствующей разрешенной области.

Если разрешенные области имеют вид прямоугольников, ориентированных параллельно координатным осям, то для метрики (2.3) «подозрительные» точки будут иметь следующие координаты [43]:

$$x_p^* = \begin{cases} x_{p1}, & \text{при } x^{\text{opt}} < x_{p1}, \\ x^{\text{opt}}, & \text{при } x^{\text{opt}} \in [x_{p1}, x_{p2}], \\ x_{p2}, & \text{при } x^{\text{opt}} > x_{p2}, \end{cases} \quad (2.19)$$

$$y_p^* = \begin{cases} y_{p1}, & \text{при } y^{\text{opt}} < y_{p1}, \\ y^{\text{opt}}, & \text{при } y^{\text{opt}} \in [y_{p1}, y_{p2}], \\ y_{p2}, & \text{при } y^{\text{opt}} > y_{p2}. \end{cases}$$

Здесь $u^{\text{opt}} = [x^{\text{opt}}, y^{\text{opt}}]$ — безусловный оптимум функции (2.18); $x_{p1}, x_{p2}, y_{p1}, y_{p2}$ — концы отрезков проекции прямоугольника p ($p = \bar{1}, P$) на соответствующие координатные оси. Точкой глобального оптимума $Q(\vec{u})$ на области D_u будет одно из значений $\vec{u}_1^*, \vec{u}_2^*, \vec{u}_3^*, \dots, \vec{u}_p^*$.

Простота вычисления особых точек (2.19) для ориентированных прямоугольников делает целесообразной аппроксимацию области D_u набором прямоугольников, ориентированных параллельно осям координат (рис. 2.8). Следует отметить, что ориентированные прямоугольники позволяют проводить аппроксимацию D_u с любой степенью точности. Правда, при этом следует учитывать, что увеличение числа прямоугольников ведет к расширению множества «подозрительных» точек, в результате чего увеличивается и время поиска \vec{u}^* .

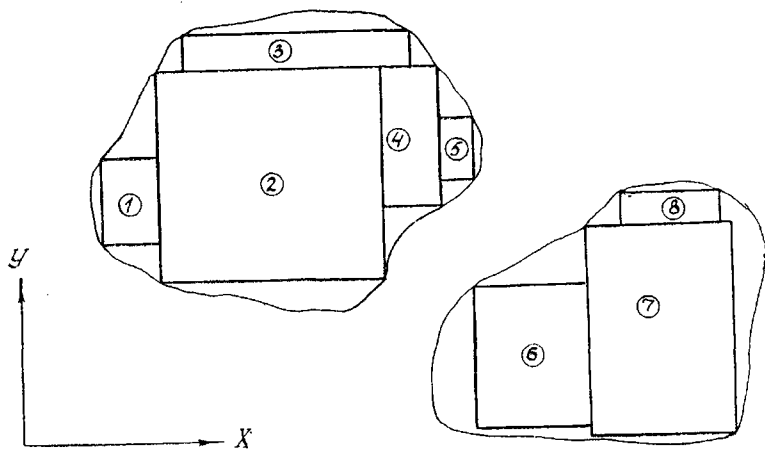


Рис. 2.8. Аппроксимация монтажного пространства ориентированными прямоугольниками

В некоторых случаях для каждого объекта ССПОИ выделяется своя собственная разрешенная область, которая может пересекаться или не пересекаться с разрешенными областями других объектов. Если каждая собственная область представляет собой прямоугольник, то задача (2.8) — (2.9) с помощью

метода (2.10)—(2.14) может быть решена с абсолютной точностью. Достаточно ввести в рассмотрение совокупности так называемых «вешек», расположенных в вершинах прямоугольников, и соединить их с соответствующими объектами линиями связи, имеющими удельную стоимость S_{\max} (рис. 2.9) :

$$S_{\max} \geq \sum_{r=0}^{R-1} \sum_{t_r=1}^{M_r} S_{r t_r}$$

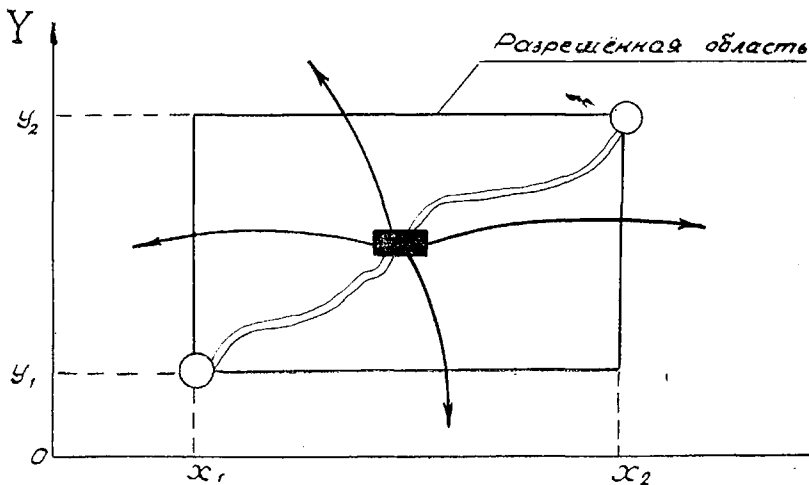


Рис. 2.9. К введению «вешек»: ■ — перемещаемый объект, ○ — «вешка», — линия связи от вешки

При этом «вешки» для ССПОИ рассматривают как объекты неперемещаемого (нулевого) уровня наравне с ИП и внешними устройствами ЭВМ (АП). В работе [43] доказано, что в этом случае задача с пространственными ограничениями (2.8)—(2.9) формально сводится к метрически эквивалентной задаче безусловной оптимизации (2.8) и может быть решена с помощью метода (2.10)—(2.14).

Алгоритм оптимального размещения при пространственных ограничениях

На основании результатов, полученных в предыдущем пункте, разработан простой алгоритм оптимального размещения объектов иерархических систем при наличии пространственных ограничений на их размещение. Алгоритм заключается в коррекции на область D_u «ортогонального» плана размещения

объектов системы (2.16), полученного с помощью метода (2.10) — (2.14). Коррекция состоит в установлении соответствия между объектами системы и разрешенными областями (прямоугольниками), после чего пространственные ограничения замещаются связями с соответствующими «вешками» (см. рис. 2.9), и задача (2.8) — (2.9) формально сводится к метрически эквивалентной задаче безусловного оптимального размещения (2.8). Алгоритм состоит из подготовительного этапа и 3-х этапов решения [43].

На подготовительном этапе монтажное пространство D_u аппроксимируется набором ориентированных прямоугольников (см. рис. 2.8). На 1-м этапе строится безусловный оптимальный «ортогональный» план размещения объектов системы (2.16). На 2-м этапе устанавливается соответствие между перемещаемыми объектами системы (в безусловном оптимальном размещении) и разрешенными областями. При этом для каждого перемещаемого объекта системы решается задача (2.17), и рассматриваемому объекту ставится в соответствие та область (прямоугольник) монтажного пространства D_u , в которой лежит решение указанной задачи. На 3-м этапе каждый выбранный прямоугольник заменяется парой «вешек» (см. рис. 2.9), расположенных в его вершинах и соединенных с соответствующим объектом линиями связи с удельной сложностью S_{\max} . После этого задача (2.8) — (2.9) формально сводится к задаче безусловного оптимального размещения, для решения которой применим метод (2.10) — (2.14).

2.6. АЛГОРИТМЫ ОПТИМАЛЬНОГО РАЗМЕЩЕНИЯ НЕДРЕВОВИДНЫХ СИСТЕМ

До сих пор рассматривались ССПОИ с древовидной структурой соединений. Это является оправданным, поскольку древовидная конфигурация соединений достаточна для обеспечения электрической связи между объектами системы.

Однако на практике монтаж оборудования зачастую ведут конструктивными блоками (стойками), объединяющими в себе несколько объектов системы. Последнее может привести к образованию замкнутых контуров в структуре соединений «стойечного» варианта системы. Появлению контуров способствует также подключение объектов системы к одному и тому же источнику питания. Однако наличие контуров не представляет принципиальных трудностей для алгоритмов оптимального размещения, описанных в п. п. 2.4, 2.5, поскольку недревовидной структуре соединений можно поставить в соответствие метрически эквивалентную древовидную структуру.

Для сети с ортогональными расстояниями (2.3) рассмотрим некоторые способы замены недревовидных структур эквивалентными древовидными. Сначала рассмотрим структуру, содержащую трехзвенные контуры (ортогональные «треугольники»). Выделим один из «треугольников» с параметрами s_{12} , s_{23} , s_{31} , Q_{Δ} , где s_{ij} — удельная стоимость линии связи между соответствующими вершинами; Q_{Δ} — суммарная стоимость линий, образующих «треугольник» (рис. 2.10,а). Этот «треугольник» метрически эквивалентен «звезде» с центральной точкой (точкой Штейнера), соединенной с каждой вершиной «треугольника» лучом, имеющим удельную стоимость, равную сумме удельных стоимостей сторон «треугольника», связанных с этой вершиной (рис. 2.10,б,в):

$$s_1 = s_{12} + s_{31}; \quad s_2 = s_{12} + s_{23}; \quad s_3 = s_{31} + s_{23}. \quad (2.20)$$

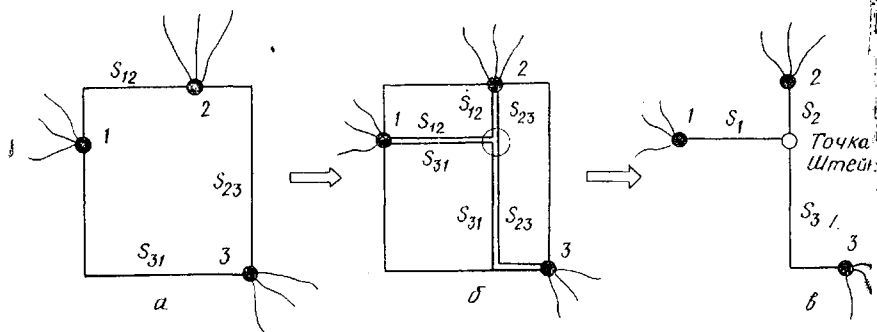


Рис. 2.10. Замена ортогонального «треугольника» «звездой»: а—«ортогональный» треугольник; б— введение точки Штейнера; в—«ортогональная» звезда

Метрическая эквивалентность означает, что $Q_{\Delta} = Q_*$, где Q_* — суммарная стоимость линий, образующих «звезду». Очевидность этого результата для ортогональной метрики (2.3) не требует доказательств [43].

На основании сказанного можно сформулировать следующий алгоритм размещения ССПОИ с трехзвенными контурами:

1. Каждый «треугольник» в структуре сети связи системы заменить эквивалентной «звездой» с параметрами (2.20).

2. Для полученной древовидной сети связи системы применить метод (2.10)—(2.14).

3. В полученном решении провести обратные замены «звезд» на соответствующие «треугольники».

Поскольку метод (2.10)—(2.14) оптимален, а «звезды» для метрики (2.3) метрически эквивалентны «треугольникам», то полученный «ортогональный» план размещения системы с трехзвенными контурами будет оптимальным (для ортогональной метрики).

Соединение объектов системы по евклидовой метрике приведет к уменьшению стоимости сети связи. Правда, она будет несколько больше стоимости наилучшего «евклидового» плана, полученного для системы с трехзвенными контурами (конечно, если такой план удалось бы получить).

Переменная фиксация

При наличии многозвенных и вложенных контуров решение несколько усложняется.

Рассмотрим сеть связи, содержащую L контуров и M перемещаемых объектов. Пусть перемещаемые объекты имеют номера, образующие множество $J = \{1, 2, \dots, M\}$. Обозначим l_j — степень j -го объекта (количество объектов из множества J , смежных с j -м объектом). Зафиксируем несколько объектов системы с номерами из множества $J^* \subset J$. В этом случае каждый объект $j \in J$ можно интерпретировать как l_j объектов нулевого уровня (неперемещаемых) с одними и теми же координатами $u_j = [x_j, y_j]$. При этом все контуры, в состав которых входит j -й объект (рис. 2.11), разомкнутся.

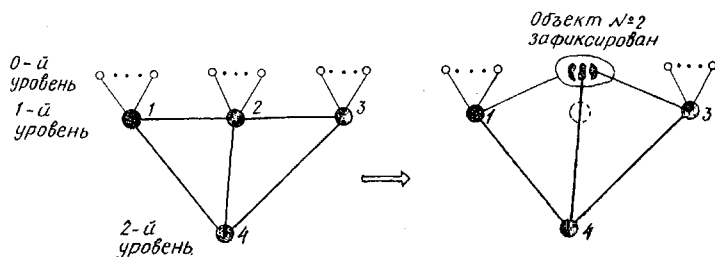


Рис. 2.11. Переход от многоконтурной сети связи к древовидной

Выберем множество J^* таким образом, чтобы фиксация объектов с номерами из J^* превратила структуру сети связи системы в древовидную. Аналогичным образом выберем множество J^{**} , но уже из оставшегося множества объектов. При этом $J^* \cap J^{**} = \emptyset$. Для преобразования структуры сети связи системы в древовидную достаточно зафиксировать объект с номерами из множества J^* либо из J^{**} . Но для этого необходимо знать координаты этих объектов.

Пусть каким-либо образом получена начальная оценка координат \hat{u}_j^* , $j \in J^*$ положения объектов с номерами из множества J^* . Выполним следующие действия:

1. Объявим объекты из множества J^* перемещаемыми объектами нулевого уровня с координатами \hat{u}_j^* , $j \in J^*$ (т. е. зафиксируем эти объекты) и перейдем от исходной многоконтурной сети связи к сети с древовидной иерархической структурой соединений (см. рис. 2.11).

2. С помощью метода (2.10)—(2.14) вычислим координаты перемещаемых объектов системы для полученной структуры соединений:

$$\hat{u}_j, j \in J \setminus J^*,$$

при этом будут вычислены координаты объектов, имеющие номера из множества J^{**} : \hat{u}_k^{**} , $k \in J^{**}$, поскольку $J^{**} \subset J \setminus J^*$.

3. Объявим объекты с номерами из множества J^{**} перемещаемыми с координатами \hat{u}_k^{**} , $k \in J^{**}$, а объекты с номерами из множества J^* освободим.

4. Для вновь полученной древовидной иерархической структуры сети связи вычислим координаты перемещаемых объектов

с помощью метода (2.10)—(2.14): \hat{u}_j , $j \in J \setminus J^{**}$, при этом будут оценены координаты объектов с номерами из множества J^* : \hat{u}_j^* , $j \in J^*$, поскольку $J^* \subset J \setminus J^{**}$.

В результате проведенной итерации будем иметь оценку положения всех объектов системы: $\hat{U} = [\hat{u}_j]$, $j \in J$. Для уточнения этой оценки вновь обратимся к п. 1 и т. д. до тех пор, пока в некоторой k -й итерации не получим установившееся решение

$$\hat{U}^{(k)} = \hat{U}^{(k-1)}.$$

Количество итераций зависит от начальной оценки \hat{u}_j^* , $j \in J^*$. В качестве оценки начального положения объектов системы целесообразно взять «ортогональный» план (2.16), полученный для рассматриваемой системы с L разомкнутыми контурами в структуре межобъектных соединений.

Обозначим погрешность оценки стоимости сети связи системы для метрики (2.3):

$$\delta Q_S^{(k)} = (Q_S^{(k)} - Q_S^{\min}) / Q_S^{\min},$$

здесь $Q_S^{\text{min}} = Q_S(U^{\text{opt}})$ — минимальная стоимость сети связи системы;

$Q_S^{(k)} = Q_S(\hat{U}^{(k)})$ — стоимость сети связи системы на k -й итерации.

Поскольку метод (2.10)—(2.14), используемый для получения оценки $\hat{U}^{(k)}$, оптимален, то $\delta Q_S^{(k)} \leq \delta Q_S^{(k-1)}$, т. е. алгоритм пп. 1—4 сходится.

Сходимость предложенного алгоритма до конца не исследована, но удалось показать, что для систем с трехзвенными контурами для метрики (2.3) он дает точное решение: $\exists k, \delta Q_S^k = 0$. Этот факт в работе [43] установлен путем сопоставления результатов, полученных с помощью алгоритма пп. 1—4 и алгоритма, основанного на замене «треугольников» «звездами». Точное решение достигалось уже после двух—трех итераций.

*Пример оптимизации системы
с недревовидной топологией сети связи [43]*

Оптимизация размещения системы (табл. 2.2) со структурой соединений, содержащей трехзвенный цикл (рис. 2.12,а), проводилась двумя способами.

Т а б л и ц а 2.2

Метрические характеристики системы

Номер ЭВМ	Параметры коммутируемых измерительных пунктов			
	Номер, i	Координаты, м		Удельная стоимость линии связи, ед/м
		x_i	y_i	
1	2	3	4	5
1	1	5	5	2
	2	15	35	3
	3	35	30	1
	4	20	25	2
	5	35	10	3
	6	45	0	2
2	7	50	65	1
	8	60	75	2
	9	70	85	3
	10	85	90	2

Окончание таблицы 2.2

1	2	3	4	5
2	11	105	95	1
	12	90	80	3
	13	75	70	2
3	14	125	55	1
	15	115	40	2
	16	130	20	2
	17	150	15	3
	18	160	35	2
	19	160	50	1
	20	170	50	3
	21	155	70	3
	22	130	80	2

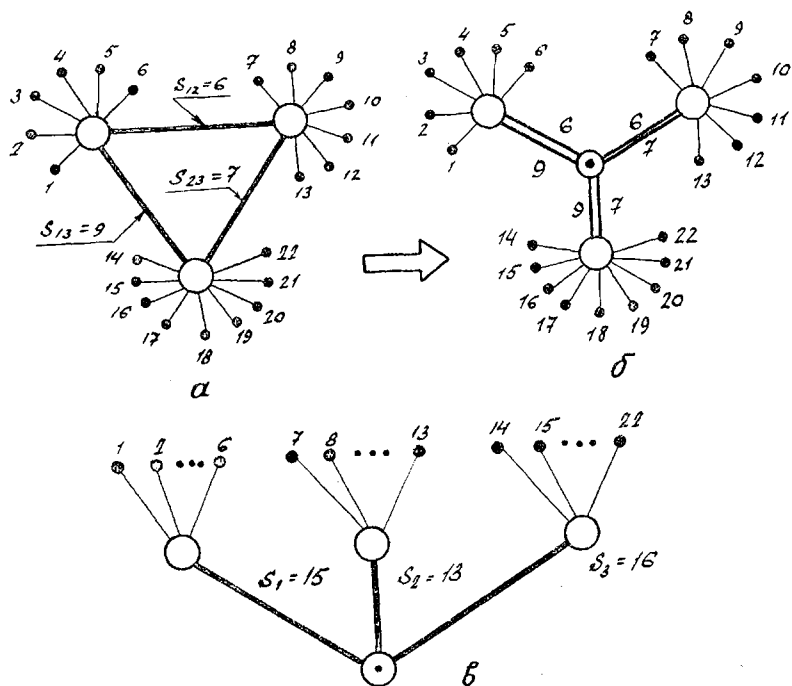


Рис. 2.12. Пример замены «треугольника» «звездой»: а — структура соединений до замены; б — после замены; в — иерархия сети связи после замены

1 способ заключался в замене «треугольника» эквивалентной «звездой» (рис. 2.12), после чего задача формально сводилась к задаче безусловного оптимального размещения.

2 способ состоял в попеременном фиксировании объектов с номерами 1 и 2 ($J^* = \{1\}$, $J^{**} = \{2\}$). На каждой итерации фиксировался один из объектов (№ 1 или № 2), в результате чего структура системы приобретала древовидный характер (рис. 2.13) и задача размещения такой системы решалась с помощью метода (2.10) — (2.14). При этом на каждой итерации соответствующий объект (№ 1 или № 2) фиксировался с координатами, полученными на предыдущей итерации. На первой итерации начальное положение объекта № 1 выбрано с координатами: $x = 0$, $y = 0$. Окончательное (установившееся) решение было получено на третьей итерации и полностью совпало с решением, полученным первым способом (рис. 2.14).

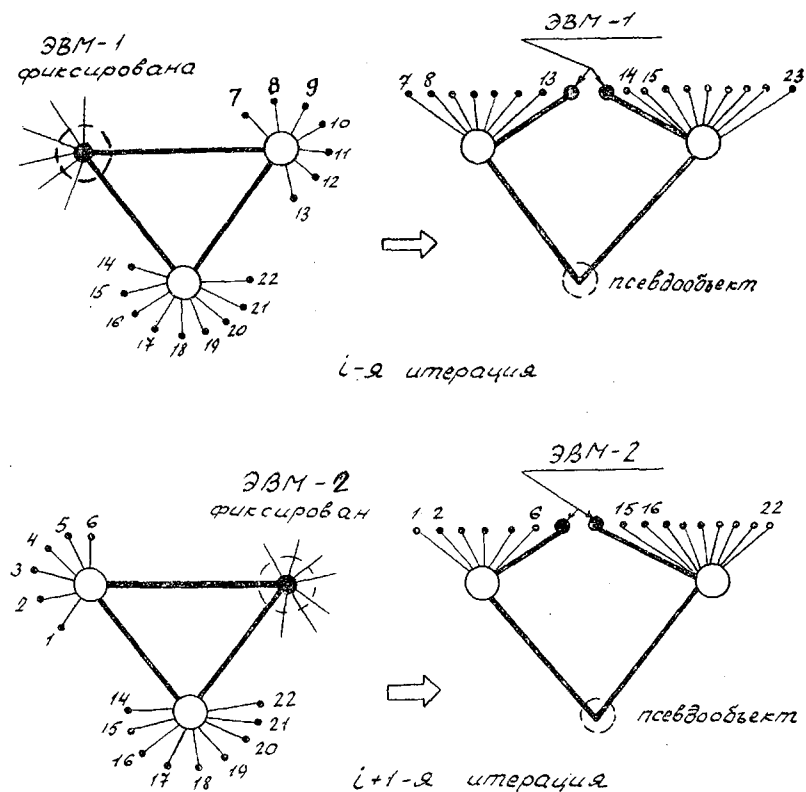


Рис. 2.13. Попеременное фиксирование

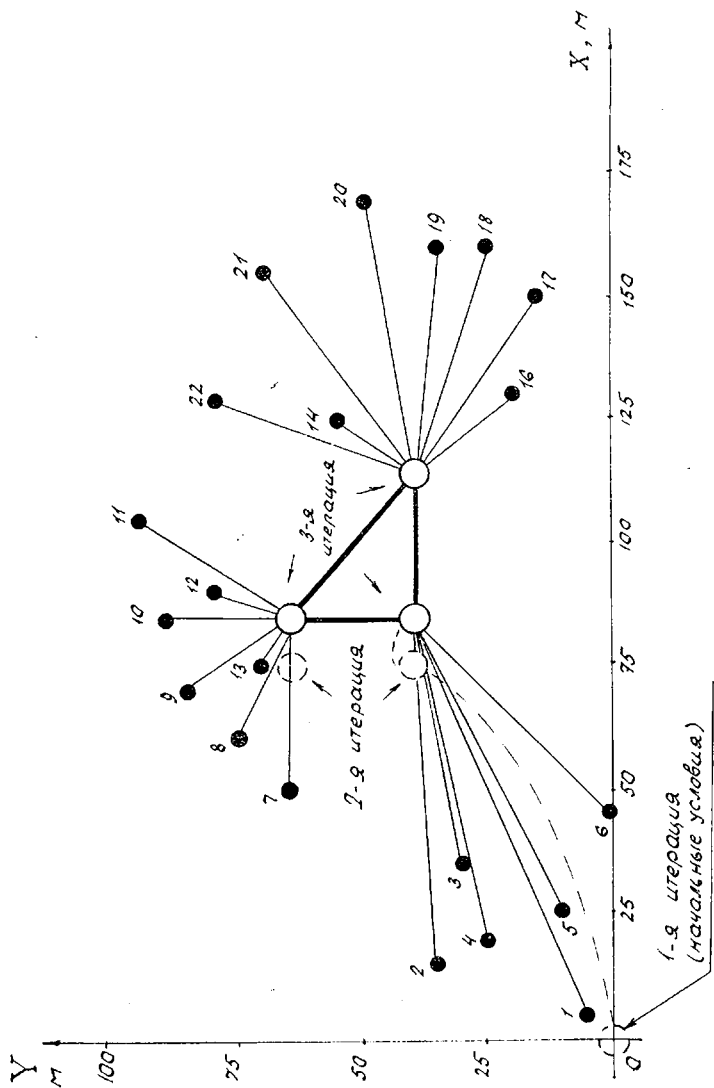


Рис. 2.14. Установившееся решение: ● — ИП; ○ — ЭВМ; (---) — движение объектов по итерациям (координаты 15-го ИП совпали с координатами ЭВМ-3)

2.7. ПОСТРОЕНИЕ НАЧАЛЬНОЙ ТОПОЛОГИИ СЕТИ СВЯЗИ

В пп. 2.4—2.6 рассматривались вопросы размеюения ССПОИ при заданной топологии сети связи, т. е. матрицы E , S считались известными. На знание топологии сети связи были также ориентированы все изложенные алгоритмы оптимального размещения. Однако проектировщику, разрабатывающему ССПОИ с самого начала, топология сети связи системы, как правило, неизвестна. Возникает задача ее определения. При этом появляются определенные трудности, поскольку для построения оптимальной топологии сети связи необходимо знать план размещения объектов системы, а для построения плана последнего необходимо знать топологию сети связи.

Для разрыва этого круга взаимосвязанных задач ниже предлагается алгоритм построения начальной топологии сети связи, основанный на интерпретации кратчайшего дерева, построенного на множестве ИП, в виде иерархической системы с перемещаемыми и перемещаемыми объектами (см. рис. 2.2).

Выбор начальной топологии ССПОИ предполагает установление соответствия между ИП и УК и построение топологии матрицали, объединяющей ЭВМ с УК (см. рис. 2.1). При этом будем считать, что положение ИП и АП известно, т. е. эти объекты являются перемещаемыми. Обозначим множество ИП как X_0 . Для построения кратчайшего дерева на множестве X_0 необходимо найти такую систему дуг, чтобы: 1 — любые два элемента из X_0 были бы связаны; 2 — общая длина всей сети связи была минимальна. Этим условиям удовлетворяет алгоритм Прима [46], который предписывает следующие действия: на множестве исходных объектов (в данном случае на множестве X_0) строится полный граф, и на первом шаге алгоритма выделяется ребро графа, имеющее наименьшую длину. Далее, на каждом последующем шаге из оставшихся ребер полного графа выбирается ребро с наименьшей длиной и проверяется, образует ли это ребро цикл с фрагментом графа, состоящим из ранее выделенных ребер. Если цикл не образуется, то рассматриваемое ребро выделяется, в противном случае — отбрасывается. Алгоритм Прима завершает свою работу после перебора всех ребер полного графа. Множество выделенных ребер полного графа будет соответствовать искомому кратчайшему дереву, которое будем обозначать $G^{(0)}$ (рис. 2.15,а).

Далее для построения начальной топологии сети связи необходимо провести следующие преобразования над полученным деревом $G^{(0)}$:

каждому элементу $\alpha \in X_0$, являющемуся внутренним узлом дерева $G^{(0)}$, поставим в соответствие дополнительный узел (так

называемую «точку Штейнера») таким образом, чтобы в итоге все элементы множества X_0 оказались внешними узлами вновь полученного дерева $G_{Ш}^{(0)}$ (рис. 2.15,б);

считая элементы множества X_0 нулевым уровнем иерархии, расположим точки Штейнера по уровням, сохраняя неизменной структуру соединений дерева $G_{Ш}^{(0)}$ (рис. 2.15,в); при этом количество уровней иерархии будет

$$R = [r_{\max} / 2],$$

где r_{\max} — количество узлов в самой длинной цепи дерева; $[\cdot]$ — целая часть «·»;

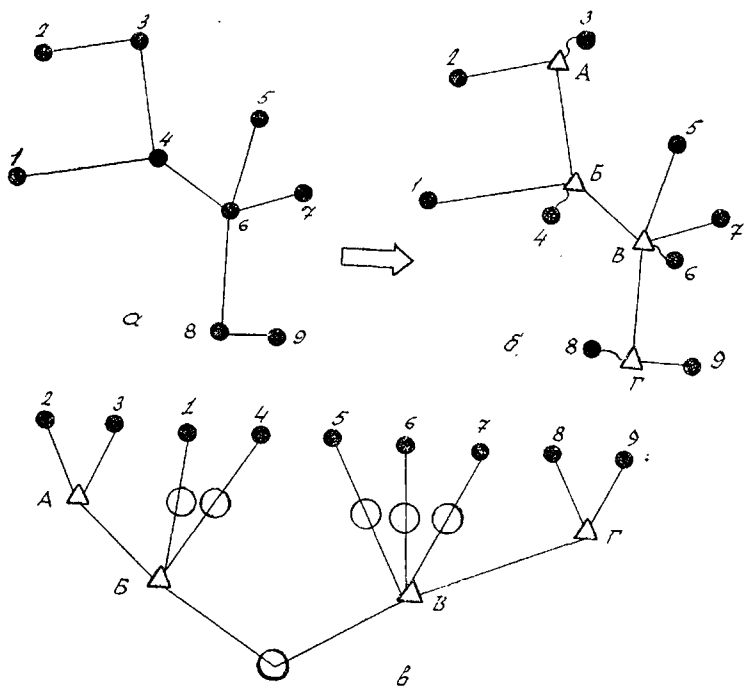


Рис. 2.15. Задание начальной топологии сети связи: а — кратчайшее дерево, построенное на множестве ИП (● — ИП); б — введение точек Штейнера (△ — точки Штейнера); в — распределение точек Штейнера по уровням иерархии (○ — псевдоточки)

недостающие объекты на уровнях иерархии представим псевдоточками и пронумеруем все точки Штейнера на каждом уровне ($r = \overline{1, R}$) от 1 до M_r . Полученную структуру нетрудно описать матрицами E и S . Например, для структуры, приведенной

ной на рис. 2.15,в, матрица E будет иметь следующий вид:

$$E = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 & 2 \\ 3 & 4 & & & & & \\ 2 & & & & & & \end{bmatrix}$$

Для определения матрицы удельных стоимостей линий связи S необходимо установить соответствие между точками Штейнера реальными объектами ССПОИ (см. рис. 2.1). В рассматриваемом случае физическими аналогами точек Штейнера являются либо УК, либо разъемы на магистрали. При этом сеть связи, объединяющая точки Штейнера, эквивалентна системной магистрали, к которой в любом месте могут подключаться ЭВМ. После идентификации точек Штейнера не составляет труда определить удельные стоимости всех линий связи ССПОИ с учетом информационной производительности ИП, коммутационной способности УК и технологических характеристик проводных линий связи.

На этом построение начальной топологии сети связи ССПОИ можно считать законченным. Полученный результат: $E^{(0)}$, $S^{(0)}$ — является опорным решением задачи построения топологической схемы ССПОИ.

2.8. СИНТЕЗ ТОПОЛОГИЧЕСКОЙ СХЕМЫ ССПОИ

Синтез топологической схемы ССПОИ подразумевает решение следующих взаимосвязанных задач:

выбор необходимого количества УК и распределение ИП между ними (составление коммутационной таблицы);
выбор структуры магистрали, объединяющей все УК с ЭВМ;
размещение объектов ССПОИ в монтажном пространстве по критерию минимума стоимости сети связи.

В общих чертах алгоритм синтеза топологической схемы ССПОИ был рассмотрен в п.2.3. Алгоритм состоит из 3-х этапов:

- 1 этап — определение начальной топологии ССПОИ;
- 2 этап — оптимальное размещение объектов ССПОИ в монтажном пространстве;
- 3 этап — коррекция топологии сети связи ССПОИ.

Два последних этапа циклически повторяются.

Первый этап алгоритма подробно рассмотрен в предыдущем параграфе. Результатом его выполнения будет ССПОИ с параметрами U_0 , $E^{(0)}$, $S^{(0)}$, где U_0 — матрица физических координат Т и АП; $\{E^{(0)}$, $S^{(0)}\}$ — начальная топология сети связи ССПОИ, соответствующая кратчайшему дереву $G_{ш}^{(0)}$, построенному на

множестве ИП. Оптимизация размещения такой системы на втором этапе алгоритма приведет к построению допустимого плана размещения объектов системы $U^{(1)*}$, обращающего в минимум стоимость сети связи системы. При этом стоимость сети связи складывается из следующих величин:

$$Q_{\text{сети}}^{(1)\min} = Q_{\text{ЛС}}^{(1)} + Q_{\text{Маг}}^{(1)}, \quad (2.2)$$

где $Q_{\text{ЛС}}$ — суммарная стоимость линий связи между ИП и УАП и ЭВМ (периферийная сеть);

$Q_{\text{Маг}}$ — стоимость системной магистрали, объединяющей УК и ЭВМ.

Алгоритмы оптимального размещения подробно рассмотрены в пп. 2.4—2.6.

Следует отметить, что при оптимизации размещения системы с топологией кратчайшего дерева $G_{\text{ш}}^{(0)}$, построенного на множестве ИП, точки Штейнера (УК и разъемы на магистрали) будут иметь тенденцию к слиянию, поскольку, как правило, удельная стоимость линий связи периферийной сети меньше удельной стоимости системной магистрали. Стягивание магистрали приведет к концентрации УК в некоторых точках пространства, которые будем называть *центрами коммутации сообщений* (ЦКС). По окончании второго этапа магистраль примет конфигурацию дерева, построенного на множестве ЦКС.

В силу оптимальности алгоритмов размещения (см. пп. 2.4—2.6) полученный план $U^{(1)}$ будет наилучшим для системы с топологией сети связи $S^{(0)}$, $E^{(0)}$, т. е. для этой топологии невозможно построить план размещения объектов, обладающий стоимостью сети связи меньшей, чем (2.21). Однако, если изменить топологию сети связи системы, то стоимость сети можно уменьшить. В самом деле, на втором этапе алгоритма при оптимизации размещения изменяются расстояния между объектами системы, в результате чего начальная топология сети связи $E^{(0)}$, $S^{(0)}$ может оказаться неоптимальной для нового плана размещения объектов $U^{(1)}$. Улучшения топологии сети можно достигнуть, во-первых, за счет перераспределения ИП между ЦКС, во-вторых, за счет построения кратчайшего дерева на множестве ЦКС (оптимизация магистрали). Указанные задачи решаются на третьем этапе алгоритма.

Для перераспределения ИП между ЦКС достаточно найти x_{ij} , $i = 1, 2, \dots, M_0$; $j = 1, 2, \dots, N_k$, минимизирующие

$$Q_{\text{ЛС}} = \sum_{i=1}^{M_0} \sum_{j=1}^{N_k} x_{ij} q_{ij} \quad (2.22)$$

* Здесь и в дальнейшем индекс в круглых скобках означает номер итерации рассматриваемого алгоритма.

ри условиях

$$\sum_{j=1}^{M_0} x_{ij} \lambda_j \leq k_j C_{УК}, \quad j = \overline{1, N_k}, \quad (2.23)$$

де M_0 — количество ИП, N_k — количество ЦКС;

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-й ИП подключен к } j\text{-му ЦКС,} \\ 0, & \text{в противном случае;} \end{cases}$$

q_{ij} — стоимость линии связи между i -м ИП и j -м ЦКС;

λ_j — информационная производительность j -го ИП;

k_j — количество стандартных УК в j -м ЦКС;

$C_{УК}$ — пропускная способность одного УК.

Задача (2.22), (2.23) является задачей линейного программирования транспортного типа. Разработаны методы, дающие точное решение $X^{opt} = [x_{ij}^{opt}]$ [8].

Обозначим $Q_{ЛС}^{(2)} = Q_{ЛС}(X^{opt}, U^{(1)})$. При этом справедливо неравенство $Q_{ЛС}^{(2)} \leq Q_{ЛС}^{(1)}$, поскольку решение x^{opt} является наилучшим.

Построение кратчайшего дерева на множестве ЦКС приведет к магистрали стоимостью $Q_{маг}^{(2)} \leq Q_{маг}^{(1)}$.

Перераспределение ИП и оптимизации магистрали даст в итоге новую топологическую схему ССПОИ с параметрами $U_0, U^{(1)}, E^{(2)}, S^{(2)}$, которая будет иметь стоимость сети связи $Q_{сети}^{(2)} \leq Q_{сети}^{(1)min}$. Причем, уменьшение стоимости сети достигнуто исключительно за счет изменения топологии сети связи (размещение объектов осталось неизменным). Для новой топологии сети связи прежний план размещения объектов $U^{(1)}$ будет, в общем случае, неоптимальным, поскольку построен для другой топологии. Поэтому целесообразно вернуться ко второму этапу алгоритма и провести размещение системы с параметрами $U_0, E^{(2)}, S^{(2)}$. Полученный план размещения объектов системы $U^{(2)}$ будет иметь стоимость сети связи $Q_{сети}^{(2)min} < Q_{сети}^{(2)}$ поскольку используемые алгоритмы размещения оптимальны.

Дальнейшее уменьшение стоимости сети связи ССПОИ возможно за счет нового перераспределения ИП между вновь образовавшимися ЦКС и построения на них кратчайшего дерева магистрали, после чего вновь воспользоваться алгоритмами оптимального размещения и т. д.

Таким образом, на каждой итерации алгоритма синтеза топологической схемы ССПОИ выполняются следующие действия:

сначала по критерию минимума стоимости сети выбирается топология сети связи (2-й этап), затем проводится оптимальное размещение ССПОИ с выбранной топологией (3-й этап). Выполнение условия $Q_{\text{сети}}^{(k)\text{min}} < Q_{\text{сети}}^{(k-1)\text{min}}$ обеспечивает сходимость алгоритма за конечное число шагов. Итерация K , на которой не происходит дальнейшего уменьшения стоимости сети связи является последней. Полученный в итоге план размещения объектов системы $U^{(k)}$ будет обладать наименьшей стоимостью сети связи $Q_{\text{сети}}^{\text{min}}$.

3.1. СЕТИ ПЕРЕДАЧИ ДАННЫХ. ОСНОВНЫЕ ПОНЯТИЯ

Рассмотренные в 1-й главе методики планирования позволяют оценить информационную нагрузку на автоматизированную систему обработки информации (АСОИ) каждого из ИП и соответствующую составляющую суммарного внешнего трафика сети передачи данных.

Основа данной сети — АСОИ ИП, АСОИ, последующих уровней обработки измерительной информации, а также коммуникационные ЭВМ (К). Указанные технические средства являются узлами сети и связаны между собой с помощью каналов передачи сообщений (КП) и связанных процессоров (СП). В целом указанный комплекс технических средств можно отнести к сетям ЭВМ (рис. 3.1). В такой сети по каналам передаются сообщения в виде данных измерений, команд, запросов и т. п.

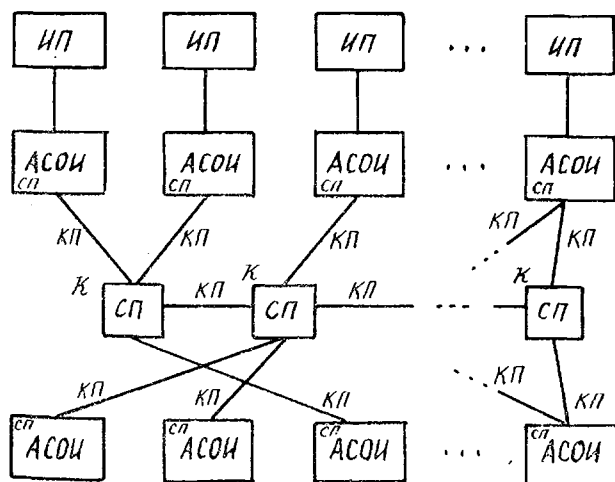


Рис. 3.1. Структура системы сбора, передачи и обработки информации

Коммутационные ЭВМ решают задачи маршрутизации сообщений, а связанные процессоры обеспечивают ввод и вывод сообщений, которые возникают на входе или выходе ЭВМ каждого узла. С точки зрения анализа и выбора параметров технических средств, входящих в данную сеть ЭВМ, ее удобнее разбить на сеть передачи данных и на совокупность АСОИ, которые составляют подсеть ресурсов обработки.

Сети передачи данных можно разбить на три типа: сети с коммутацией каналов (цепей), сети с коммутацией сообщений и сети с коммутацией пакетов.

Сеть с коммутацией каналов вначале устанавливает путь из соединенных цепей от источника к адресату. После этого начинается передача данных и все каналы этого пути используются одновременно. Весь путь используется только данным источником до тех пор, пока тот его не освободит. Тогда все каналы, входящие в этот путь, вновь доступны для использования их другими источниками. Коммутация каналов широко используется в телефонных системах.

При *коммутации сообщений* в определенный момент времени используется только один канал сети для передачи сообщения от источника к адресату. Сообщение на каждом шаге выбирает канал и передается по нему от узла к узлу. Если выбранный канал занят, то сообщение ожидает в очереди, и, наконец, когда канал освобождается, передача возобновляется.

Наиболее универсальным и гибким способом передачи данных в сети ЭВМ является *коммутация пакетов*. Он напоминает в основном, коммутацию сообщений, но сообщения разбиваются на меньшие части, называемые *пакетами*. Эти пакеты нумеруются и снабжаются адресом (как и при коммутации сообщений) и прокладывают себе путь по сети, которая их коммутирует. Как и в случае коммутации сообщений пакеты могут находиться в очереди, ожидая освобождения выбранного канала.

Таким образом, множество пакетов одного и того же сообщения может передаваться одновременно, что является одним из главных преимуществ данного типа коммутации. Для цифровой передачи данных, что типично для сетей ЭВМ, более предпочтительно использование коммутации сообщений и пакетов. Вместе с тем, если поток данных имеет группирующую структуру (что характерно для данных, выходящих из ЭВМ), то тогда разумным является выбор коммутации пакетов.

В общем случае сеть передачи данных составляют: физическая сеть, состоящая из коммутационных ЭВМ и каналов связи;

потоки сообщений, которые описываются местом их возник-

ования, местом назначения, временем возникновения, длиной и приоритетным классом;

множество операционных правил, согласно которым обрабатывается поток этих сообщений.

К последнему множеству относятся процедуры выбора маршрута, процедуры управления потоками, дисциплины обслуживания очередей и т. п.

Процедура выбора маршрута сообщений представляет собой правила выбора (согласно некоторому алгоритму) следующего узла, в который сообщение придет на своем пути по сети. *Фиксированной* процедурой выбора маршрутов будем называть такую, при которой для данной пары источник-адресат на сети имеется только один путь.

Если в процедуре разрешено использовать более одного пути, то ее называют *процедурой выбора маршрутов допускающей альтернативы*. Выбор альтернативных путей среди действующих цепочек каналов можно осуществлять либо детерминированно, либо случайно. Если алгоритм выбора маршрутов основан на решениях относительно некоторых параметров потоков данных в сети и (или) выхода из строя узлов или каналов, то соответствующую процедуру выбора маршрутов можно назвать *динамической*, или *адаптивной*.

Процедура управления потоками предсказывает и предотвращает перегрузки путем регулирования значений интенсивности входных потоков сети.

3.2. МОДЕЛЬ СЕТИ ПЕРЕДАЧИ ДАННЫХ. ЗАДАЧИ ПРОЕКТИРОВАНИЯ

Эффективное проектирование сетей передачи данных с коммутацией пакетов и сообщений является весьма сложной задачей в связи с большим количеством параметров, сложной структурой и стохастическим характером потоков данных. В данной главе рассматриваются некоторые из наиболее типичных и важных задач, которые возникают в процессе проектирования [9], такие как:

выбор пропускных способностей (ВПС) каналов передачи данных;

распределение потоков данных (РП) в сети.

Более тщательные формулировки задач мы дадим после рассмотрения принятой математической модели сети передачи данных.

Выберем в качестве единственного критерия качества функционирования сети передачи данных среднюю задержку сообщения в сети T . Для получения аналитической формы зависимо-

сти T рассмотрим следующую модель сети, имеющую M каналов и N узлов [9]. Каналы являются бесшумными и абсолютно надежными, а пропускная способность i -го канала равна C_i (бит/с). Все N узлов, соответствующие центрам коммутации сообщений, предполагают абсолютно надежными. Считается, что время обработки сообщений в узлах является постоянной величиной и обычно предполагается пренебрежимо малой. Кроме того, в модели имеются очереди к каналам и задержки при передаче. Поток сообщений, поступающих в сеть из внешнего источника, принимают стационарным пуассоновским со средним значением γ_{jk} (сообщений в секунду) для тех сообщений, которые возникают в узле j и предназначаются для узла k . Полный внешний трафик γ (суммарная интенсивность потоков, поступающих в сеть) можно определить по формуле

$$\gamma = \sum_{j=1}^N \sum_{k=1}^N \gamma_{jk}. \quad (3.1)$$

Длины всех сообщений одинаковы и распределены по экспоненциальному закону со средним значением $S = 1/\mu$ бит. Для размещения сообщений, ожидающих освобождения канала, используется память неограниченной емкости. Предположим также, что процедура выбора маршрутов фиксированная, т. е. для данной пары источник-адресат на сети имеется только один путь.

Обозначим через π_{jk} путь, по которому идут сообщения, возникающие в узле j и направляющиеся в узел k (трафик $j-k$). Считаем, что i -й канал с пропускной способностью C_i включается в путь π_{jk} , если сообщения, идущие по этому пути, проходят по указанному каналу; при этом используется обозначение $C_i \in \pi_{jk}$. Поэтому ясно, что средняя интенсивность потока сообщений в i -м канале должна равняться сумме средних интенсивностей потоков сообщений, проходящих через этот канал, т. е.

$$\lambda_i = \sum_j \sum_{k: C_i \in \pi_{jk}} \gamma_{jk}, \quad (3.2)$$

Определим задержку сообщения как полное время, которое сообщение проводит в сети. Наибольший интерес для нас представляет средняя задержка сообщения T , которая принята нами за главную характеристику сети. Обозначим через z_{jk} среднюю задержку сообщения, возникающего в узле j и имеющего узел назначения k .

Величины T и z_{jk} связаны равенством

$$T = \sum_{j=1}^N \sum_{k=1}^N \frac{\gamma_{jk}}{\gamma} z_{jk}, \quad (3.3)$$

к как доля γ_{jk}/γ полного входящего трафика сообщений имеет среднюю задержку, равную z_{jk} и представляющую собой сумму средних задержек, испытываемых сообщением при передаче по различным каналам пути π_{jk} .

Обозначим через T_i время, затраченное сообщением на ожидание и процесс передачи по i -му каналу. Тогда можно записать, что $z_{jk} = \sum_{i: C_i \in \pi_{jk}} T_i$.

Следовательно, из формулы (3.3) получим

$$T = \sum_{j=1}^N \sum_{k=1}^N \frac{\gamma_{jk}}{\gamma} \sum_{i: C_i \in \pi_{jk}} T_i.$$

Изменим порядок суммирования, в результате чего будем иметь

$$T = \sum_{i=1}^M \frac{T_i}{\gamma} \sum_j \sum_k \gamma_{jk}, \quad (j, k: C_i \in \pi_{jk}).$$

В последнюю формулу подставим выражение (3.2), тогда

$$T = \sum_{i=1}^M \frac{\lambda_i}{\gamma} T_i. \quad (3.4)$$

Теперь средняя задержка T разложена на компоненты, относящиеся к отдельным каналам. Для вычисления средней задержки T_i , относящейся к одиночному каналу, «погруженному» в сеть, можно воспользоваться результатом анализа экспоненциальной сети Джексона [11], суть которого в том, что погруженный канал может рассматриваться как канал, действующий независимо от сети, но с пуассоновским потоком на входе. Интенсивность этого потока равна интенсивности, задаваемой сетью. В данных сетях время обслуживания является независимой случайной величиной.

В нашем случае время обслуживания данного сообщения в различных каналах прямо связано с длиной сообщения и фиксированными значениями пропускной способности каналов. В связи с этим говорить о независимости времени обслуживания сообщения в различных каналах здесь нельзя. Кроме того, интервал времени между моментами поступления двух последовательных сообщений в данный канал не может быть меньше, чем время обслуживания первого из них в указанном канале. Так как время обслуживания этого сообщения в его следующем канале непосредственно связано с предыдущим временем его обслуживания, процесс поступления сообщений в узел сети не является независимым от времени обслуживания сообщения в данном узле.

Однако, если сообщения, уходящие из узла по данному каналу, пришли к узлу из разных каналов или, если сообщения, прошедшие по одному и тому же каналу, отправляются в разные каналы, то для сетей со средней связанностью (т. е. узлы сети должны иметь более чем один входной канал и более чем один выходной) справедливо следующее предположение о независимости. Всякий раз, когда сообщение принимается в узле внутри сети, выбирается его новая длина b согласно плотностям вероятностей $f(b) = \mu e^{-\mu b}$, $b \geq 0$.

Таким образом, утверждается, что всякий раз, когда сообщение принимается в узле, новая длина сообщений выбирается (генерируется) в соответствии с указанным распределением. Хотя это утверждение и не соответствует действительности, так как сообщения сохраняют длину при их прохождении по сети, вычисление величины T производится при этом с достаточной степенью точности.

Пользуясь указанным утверждением о независимости, можно рассматривать нашу сеть как сеть Джексона, и вычисление T производить, представив одиночный погруженный канал как систему массового обслуживания $M/M/1$. Тогда

$$T_i = \frac{1}{\mu C_i - \lambda_i}. \quad (3.5)$$

Окончательное выражение для задержки сообщения в сети получим, используя формулы (3.4) и (3.5),

$$T = \sum_{i=1}^M \frac{\lambda_i}{\gamma} \left(\frac{1}{\mu C_i - \lambda_i} \right). \quad (3.6)$$

3.3. ВЫБОР ПРОПУСКНЫХ СПОСОБНОСТЕЙ КАНАЛОВ СВЯЗИ

Решение задачи в условиях полной априорной информации

Используя математическую модель сети, представленную выше, сформулируем задачу выбора пропускных способностей (ВПС) следующим образом.

Дано: потоки $\{\lambda_i\}$ и структура сети.

Минимизировать: T .

Варьируются: $\{C_i\}$.

Ограничения: $D = \sum_{i=1}^M d_i(C_i)$, $C_i > \frac{\lambda_i}{\mu}$,

где D — заданная стоимость создания сети передачи данных, $d_i(C_i)$ — функция стоимости создания i -го канала с выбранной пропускной способностью.

Будем в дальнейшем рассматривать случай линейных стоимостных функций, а именно

$$d_i(C_i) = d_i C_i, \quad (3.8)$$

где d_i — стоимость создания канала в расчете на единицу пропускной способности для i -го канала.

В реальных условиях, как правило, решение сформулированной выше задачи ВПС нужно искать на ограниченном множестве дискретных значений пропускных способностей $\{C_i^*\}$, где C_i^* — значение, обусловленное типовой аппаратурой и стандартами на организацию передачи данных в сети ЭВМ.

Для простоты рассмотрим вначале задачу ВПС, когда выбираемые пропускные способности могут иметь любые неотрицательные значения.

Средняя длина сообщения, как было сказано ранее, равна $S = 1/\mu$, λ_i — интенсивность потока сообщений, проходящего по i -му каналу. Тогда λ_i/μ — среднее число битов в секунду, проходящих через i -й канал, и, следовательно, любое реализуемое решение задачи ВПС должно быть таким, чтобы i -й канал имел пропускную способность не меньшую, чем λ_i/μ , т. е. необходимо выполнение условия $C_i > \lambda_i/\mu$.

Итак, мы имеем выражение для задержки сети T :

$$T = \sum_{i=1}^M \frac{\lambda_i}{\gamma} \left(\frac{1}{\mu C_i - \lambda_i} \right);$$

ограничение по стоимости $D = \sum_{i=1}^M d_i C_i$.

Известны топология сети и интенсивность потоков сообщений в каналах $\{\lambda_i\}$. Необходимо выбрать оптимальные пропускные способности каналов $\{C_i\}$, ($i = \overline{1, M}$), которые минимизировали бы задержку передачи информации сети T .

Для решения задачи используется метод неопределенных множителей Лагранжа. Составим функцию Лагранжа

$$G = \sum_{i=1}^M \frac{\lambda_i}{\gamma} \frac{1}{\mu C_i - \lambda_i} - \beta \left(\sum_{i=1}^M d_i C_i - D \right),$$

где β — неопределенный множитель Лагранжа.

Продифференцируем функцию G по каждой из переменных C_i ($i = \overline{1, M}$) и приравняем производные нулю. Получим систему M уравнений:

$$\left(-\frac{\lambda_i}{\gamma} \right) \frac{\mu}{(\mu C_i - \lambda_i)^2} - \beta d_i = 0, \quad i = \overline{1, M}.$$

Решая систему относительно \bar{C}_i , получаем

$$C_i = \frac{\lambda_i}{\mu} + \frac{1}{\sqrt{\beta \gamma \mu}} \sqrt{\frac{\lambda_i}{d_i}}, \quad i = \overline{1, M}. \quad (3.9)$$

Теперь, если найти коэффициент β , то выражение (3.9) станет решением поставленной задачи. Для этого умножим обе части выражения (3.9) на d_i и просуммируем по i от 1 до M :

$$\sum_{i=1}^M d_i C_i = \sum_{i=1}^M \frac{\lambda_i d_i}{\mu} + \frac{1}{\sqrt{\beta \gamma \mu}} \sum_{i=1}^M \sqrt{\lambda_i d_i}$$

Как следует из формулы (3.8), левая часть последнего равенства равна D , поэтому:

$$\frac{1}{\sqrt{\beta \gamma \mu}} = \frac{D - \sum_{i=1}^M (\lambda_i d_i / \mu)}{\sum_{i=1}^M \sqrt{\lambda_i d_i}}. \quad (3.10)$$

Обозначим $D - \sum_{i=1}^M \frac{\lambda_i d_i}{\mu}$ через D_e и назовем величину D_e добавочной стоимостью.

Представим выражение (3.10) в выражение (3.9) и получим оптимальное решение задачи ВПС:

$$C_i = \frac{\lambda_i}{\mu} + \left(\frac{D_e}{d_i} \right) \frac{\sqrt{\lambda_i d_i}}{\sum_{j=1}^M \sqrt{\lambda_j d_j}}, \quad i = \overline{1, M}. \quad (3.11)$$

Проанализируем равенство (3.11). При таком наборе пропускных способностей каждый канал будет иметь по крайней мере пропускную способность λ_i/μ . Отметим, что стоимость минимальной пропускной способности i -го канала равна $\lambda_i d_i / \mu$. Взяв сумму по всем каналам, можно заметить, что полная стоимость сети должна быть больше этой суммы, чтобы средняя задержка сети была величиной конечной. Разность между полной и минимальной допустимой стоимостями равна D_e и задается соотношением

$$D_e = D - \sum_{i=1}^M \left(\frac{\lambda_i d_i}{\mu} \right). \quad (3.12)$$

Подставив выражение (3.11) в формулу для задержки сети (3.6), получим величину задержки T при выбранных значениях пропускных способностей:

$$T = \frac{1}{\mu \gamma D_e} \left(\sum_{i=1}^M \sqrt{\lambda_i d_i} \right)^2. \quad (3.13)$$

Выражение (3.13) дает минимальную среднюю задержку сети, пропускные способности которой выбраны оптимально. Величина D_e играет здесь важную роль: при $D_e \rightarrow 0$ средняя задержка сообщения T неограниченно возрастает, если $D_e > 0$, задача ВПС имеет реализуемое решение (т. е. $T < \infty$). Если $D_e \leq 0$, то задача ВПС не имеет реализуемого решения. Условие $D_e = 0$ называется *условием устойчивости* (стабильности) системы.

Вышеизложенное решение задачи ВПС является идеальным, так как в общем случае значения пропускных способностей нельзя считать непрерывными. Их необходимо выбирать из некоторого заданного ограниченного дискретного множества ввиду того, что система строится из стандартных узлов и блоков. Поэтому практически для решения задачи ВПС приходится привлекать более сложный математический аппарат. Для этих целей возможно использование метода динамического программирования [12].

Рассмотрим следующую процедуру динамического программирования [13].

Найти x_1, x_2, \dots, x_N , удовлетворяющие условиям

$$\sum_{i=1}^N d_i x_i \leq b, \quad (3.14)$$

$$x_i \geq 0, \quad i = \overline{1, N} \quad (\text{все } x_i \text{ — целые}) \quad (3.15)$$

и минимизирующие

$$T = \sum_{i=1}^N T_i(x_i). \quad (3.16)$$

Будем считать все d_i и b целыми. Вектор оптимальных решений $X^* = (x_1^*, x_2^*, \dots, x_N^*)$ можно найти с помощью следующей вычислительной процедуры.

Введем в рассмотрение последовательность функций

$$F_k(\xi) = \min \left(\sum_{j=1}^k T_j(x_j) \right), \quad (3.17)$$

где минимум берется по неотрицательным целым значениям x_1, x_2, \dots, x_k , удовлетворяющим условию

$$\sum_{j=1}^k d_j x_j = \xi, \quad (3.18)$$

при этом $0 \leq \xi \leq b$. (3.19)

Функции (3.17) определим последовательно с помощью рекуррентных соотношений, которые получаем в ходе следующих рассуждений: $F_k(\xi)$ есть минимум функции $V = \sum_{j=1}^k T_j(x_j)$ от k

переменных x_1, x_2, \dots, x_k , удовлетворяющих условию (3.18). Выберем значение x_k ; зафиксировав его, будем минимизировать V по всем остальным переменным x_1, \dots, x_{k-1} . При этом в силу условия (3.18) величины x_1, \dots, x_{k-1} будут зависеть от выбранного значения x_k . Представим себе, что минимизация выполнена для всех возможных значений x_k . Тогда наименьшее из всех полученных значений V будет равно $F_k(\xi)$, и мы найдем совокупность значений x_1, x_2, \dots, x_k , минимизирующую V .

Таким образом, выбирается и вычисляется величина x_k :

$$\min_{x_1, \dots, x_{k-1}} \left(\sum_{j=1}^k T_j(x_j) \right) = T_k(x_k) + \min_{x_1, \dots, x_{k-1}} \left(\sum_{j=1}^{k-1} T_j(x_j) \right), \quad (3.20)$$

где слагаемое $T_k(x_k)$ вынесено из-под знака минимума, поскольку не зависит от x_1, x_2, \dots, x_{k-1} . Когда x_k выбрано, значения x_1, x_2, \dots, x_{k-1} , кроме условий целочисленности и неотрицательности, должны удовлетворять неравенству

$$\sum_{j=1}^{k-1} d_j x_j \leq \xi - d_k x_k, \quad (3.21)$$

а это означает, что в выражении (3.20) второе слагаемое справа есть $F_{k-1}(\xi - d_k x_k)$. Учитывая, что

$$\min_{x_1, \dots, x_k} \sum_{j=1}^k T_j(x_j) = \min_{x_k} \left(T_k(x_k) + \min_{x_1, \dots, x_{k-1}} \sum_{j=1}^{k-1} T_j(x_j) \right), \quad (3.22)$$

приходим к рекуррентному соотношению

$$F_k(\xi) = \min_{x_k} \left(T_k(x_k) + F_{k-1}(\xi - d_k x_k) \right), \quad (3.23)$$

где x_k может принимать целочисленные значения от 0 вплоть до $[\xi/d_k]$, а $[\xi/d_k]$ означает целую часть числа ξ/d_k . Причем

$$0 \leq \frac{\xi}{d_k} \leq \left[\frac{\xi}{d_k} \right]. \quad (3.24)$$

Функцию $F_1(\xi)$ определяют непосредственно:

$$F_1(\xi) = \min T_1(x_1), \quad (3.25)$$

$$0 \leq x_1 \leq \left[\frac{\xi}{d_1} \right],$$

учитывая, что должно выполняться условие $d_1 x_1 \leq \xi$ для всех $\xi = 0, 1, \dots, b$. Зная $F_1(\xi)$, определяем функцию $F_2(\xi)$ согласно формуле (3.23):

$$F_2(\xi) = \min_{0 \leq x_2 \leq \left[\frac{\xi}{d_2} \right]} \left(T_2(x_2) + F_1(\xi - d_2 x_2) \right), \quad (3.26)$$

так же для всех $\xi = 0, 1, \dots, b$.

Затем определяем функцию $F_3(\xi)$ и т. д., наконец,

$$F_N(\xi) = \min_{0 \leq x_N \leq \left\lfloor \frac{\xi}{d_N} \right\rfloor} (T_N(x_N) + F_{N-1}(\xi - d_N x_N)). \quad (3.27)$$

Для нас важно, что $\min_{x_i} \sum_{i=1}^N T_i(x_i) = F_N(b)$.

Оптимальное решение $x_1^*, x_2^*, \dots, x_N^*$ находим следующим образом.

Значение (или значения x_N), при котором достигается $F_N(b)$, есть $x_N^* = \bar{x}_N(b)$. Так как x_N^* известно, то остальные переменные должны удовлетворять условию

$$\sum_{j=1}^N d_j x_j \leq b - d_N x_N^*, \quad (3.28)$$

и потому выражение $\sum_{j=1}^{N-1} T_j(x_j)$ должно достигать минимума на множестве неотрицательных целых чисел, удовлетворяющих этому условию, а согласно выражениям (3.17) — (3.18), этот минимум есть $F_{N-1}(b - d_N x_N^*)$. Тогда значение x_{N-1} , при котором минимум достигается, есть $\bar{x}_{N-1}(b - d_N x_N^*)$. И так, получаем

$$x_{N-1}^* = \bar{x}_{N-1}(b - d_N x_N^*); \quad (3.29)$$

иными словами, чтобы найти x_{N-1}^* , мы должны в таблице для \bar{x}_{N-1} , взять число, соответствующее $\xi = b - d_N x_N^*$.

Аналогично получим $x_{N-2}^* = \bar{x}_{N-2}(b - d_N x_N^* - d_{N-1} x_{N-1}^*)$. Для $i = 1, 2, \dots, N-1$ найдем

$$x_{N-i}^* = \bar{x}_{N-i}(b - \sum_{\alpha=0}^{i-1} d_{N-\alpha} x_{N-\alpha}^*). \quad (3.30)$$

Таким образом, получаем оптимальное решение $x_1^*, x_2^*, \dots, x_N^*$ задачи (3.14) — (3.16).

Описанная процедура динамического программирования применима к решению задачи ВПС, если переменные x_i рассматривать как выбираемые значения пропускных способностей канала связи C_i . Однако в процедуре предполагается, что переменные x_i могут принимать все целые значения от 0 до ∞ . В нашем случае мы имеем, как правило, весьма ограниченное множество возможных дискретных значений пропускных способностей. Кроме того, элементы множества $\{C_i^*\}$ кратны, как правило, некоторому минимальному значению из этого множества.

В этих условиях изложенная процедура несколько видоизменяется, в частности за счет того, что на каждом шаге вычисления функции $F_k(\xi)$ появляются ограничения на возможные значения ξ . С учетом сказанного рассмотрим пример, иллюст-

рирующий соответствующую модификацию представленной выше процедуры динамического программирования.

Пусть заданы функции $T_1(x_1)$, $T_2(x_2)$, $T_3(x_3)$, $T_4(x_4)$. Аргументы x_1, x_2, x_3, x_4 могут принимать значения из заданного множества целых чисел $X_0 = (10, 20, 30)$; $b = 150$; $d_1 = 1$, $d_2 = d_3 = 4$, $d_4 = 2$.

Требуется найти значения x_1, x_2, x_3, x_4 , минимизирующие сумму вида $T = \sum_{i=1}^4 T_i(x_i)$ при $\sum_{i=1}^4 d_i x_i \leq b$.

На основании известных $T_i(x_i)$ ($i = \overline{1,4}$) и X_0 можно составить таблицу следующего вида (табл. 3.1)

Таблица 3.1

Исходные данные

x	$T_i(x)$			
	$T_1(x)$	$T_2(x)$	$T_3(x)$	$T_4(x)$
10	1	10	2	5
20	2	3	4	10
30	10	1	6	3

Второй столбец таблицы содержит значения функции $F_1(\xi)$, соответствующие значения аргумента x есть $\bar{x}_1(\xi)$.

Вычислим далее значения $F_2(\xi)$ и $\bar{x}_2(\xi)$ для чего составим табл. 3.2. При фиксированном значении ищем величину $F_2(\xi)$ являющуюся минимумом

мом функции (3.26). Значения ξ помечены в таблице звездочкой

Таблица 3.2

Поиск $F_2(\xi)$ и значения $F_2(\xi)$, $\bar{x}_2(\xi)$

$\xi - d_2 x_2$	$T_2(x_2) + F_1(\xi - d_2 x_2)$		
	$x_2 = 10$	$x_2 = 20$	$x_2 = 30$
10	11; 40*	4; 70*	2; 100*
20	12; 50*	5; 80*	3; 110*
30	20; 60*	13; 90*	11; 120*

Значения параметра ξ	$F_2(\xi)$	$\bar{x}_2(\xi)$
40	11	10
50	12	10
60	20	10
70	4	20
80	5	20
90	13	20
100	2	30
110	3	30
120	11	30

Следует отметить, что ξ может изменяться в пределах от 0 до 150, но функции $T_i(x_i)$ определены только в точках 10, 20, 30, поэтому на первом шаге диапазон изменения $\xi = [40 \dots 120]$. Аналогично строим табл. 3.3, откуда находим значения $F_3(\xi)$ и $\bar{x}_3(\xi)$.

Таблица 3.3

Поиск $F_3(\xi)$ и значения $F_3(\xi)$, $\bar{x}_3(\xi)$

$\xi - d_3 x_3$	$T_3(x_3) + F_2(\xi - d_3 x_3)$		
	$x_3=10$	$x_3=20$	$x_3=30$
40	13; 80*	15; 120*	—
50	14; 90*	16; 130*	—
60	22; 100*	24; 140*	—
70	6; 110*	8; 150*	—
80	7; 120*	—	—
90	15; 130*	—	—
100	4; 140*	—	—
110	5; 150*	—	—

Значения параметра ξ	$F_3(\xi)$	$\bar{x}_3(\xi)$
80	13	10
90	14	10
100	22	10
110	6	10
120	7	10
130	15	10
140	4	10
150	5	10

Наконец, чтобы определить $F_4(\xi)$ для единственного значения аргумента $\xi = 150$, составляем табл. 3.4.

Из табл. 3.4 следует, что $f_4(150) = 16$. Соответствующее значение $x_4^* = \bar{x}_4(150) = 20$.

Теперь можно найти искомые значения x_1^* , x_2^* , x_3^* :

1. Из табл. 3.3 найдем x_3^* :

$$x_3^* = \bar{x}_3(b - x_4^* d_4) = \bar{x}_3(150 - 40) = \bar{x}_3(110) = 10.$$

2. Из табл. 3.2 находим x_2^* :

$$x_2^* = \bar{x}_2(b - x_4^* d_4 - x_3^* d_3) = \bar{x}_2(150 - 40 - 40) = \bar{x}_2(70) = 20.$$

3. Наконец, находим последнее значение x_1^* :

$$x_1^* = (b - x_4^* d_4 - x_3^* d_3 - x_2^* d_2) / d_1 = (150 - 140) / 1 = 10.$$

Итак: $x_1^* = 10$, $x_2^* = 20$, $x_3^* = 10$, $x_4^* = 20$.

$$\sum_{i=1}^4 d_i x_i = 150; \quad T = \sum_{i=1}^4 T_i(x_i) = 16.$$

Полученное минимальное значение T совпадает со значением функции Беллмана (см. табл. 3.4).

Рассмотренную процедуру динамического программирования за счет некоторого усложнения вычислительного алгоритма можно модифицировать для случая, когда дискретное множество

возможных значений переменных $x_i (i=1, N)$ (для задачи ВПС-дискретное множество значений пропускных способностей) имеет производную структуру (за исключением требований целочисленности).

Решение задачи в условиях риска

Задачу выбора пропускных способностей (ВПС), рассмотренную выше, решали в предположении, что внешний трафик $\{\gamma_{jk}\}$ известен с точностью до своих постоянных значений. На практике же может возникнуть ситуация, когда $\{\gamma_{jk}\}$ известен с точностью до распределения вероятностей на некоторых интервалах $[\{\min \gamma_{jk}\}, \{\max \gamma_{jk}\}]$. В этом случае с учетом фиксированного маршрута распределения внешнего трафика можно осуществить выбор $\{C_i\}$, рассмотрев максимально возможные значения интенсивностей $\{\lambda_i\}$ в каналах связи (предполагаются наихудшие условия эксплуатации сети связи). С другой стороны, было бы разумным попытаться найти, хотя бы приближенно, оптимальные $\{C_i\}$ с учетом априорной информации о вероятностном характере $\{\gamma_{jk}\}$. Задача ВПС в этих условиях может иметь различные постановки. Рассмотрим одну из них.

При заданной структуре сети и фиксированном алгоритме маршрутизации необходимо выбрать такие значения пропускных способностей каналов связи $\{C_i\}$, для которых стоимость сети $D = \sum_{i=1}^M d_i C_i$ была бы минимальной, а вероятность того, что средняя задержка в сети T превысит некоторое заданное максимальное значение T_{\max} , была бы не больше заданной P_d . Данную постановку задачи можно представить в виде

$$\min_{\{C_i\}} D = \sum_{i=1}^M d_i C_i, \quad (3.31)$$

$$P\{T > T_{\max}\} \leq P_d, \quad (3.32)$$

$$\frac{\max \lambda_i}{\mu} < C_i, \quad (3.33)$$

$$C_i \in \{C_j^*\} \quad i = 1, 2, \dots, M.$$

Из условия задачи видно, что в связи с конечностью множества $\{C_j^*\}$ она не всегда может иметь физически реализуемое решение. Решение задачи ВПС в данной постановке предполагает, в частности, необходимость определения $P\{T > T_{\max}\}$.

Напомним, что средняя задержка в сети T определяется из выражения (3.6):

$$T = \frac{1}{\gamma} \sum_{i=1}^M \frac{\lambda_i}{\mu C_i - \lambda_i},$$

где

$$\gamma = \sum_{j=1}^N \sum_{k=1}^N \gamma_{jk};$$
$$\lambda_i = \sum_j \sum_k \gamma_{jk}; \quad j, k: C_i \in \pi_{jk}.$$

Предположим, что γ_{jk} независимые случайные величины, дисперсии и математические ожидания которых соизмеримы. Число каналов и пар источник—адресат достаточно велико. Тогда согласно центральной предельной теореме теории вероятностей [15] случайные величины γ , $\{\lambda_i\}$ подчиняются нормальному закону распределения. Параметры этих законов распределения — математические ожидания и дисперсии — находятся соответствующим суммированием математических ожиданий и дисперсий случайных величин γ_{jk} . Вектор $\{\max \lambda_i\}$ можно найти, исходя из полученных законов распределения для $\{\lambda_i\}$, например, по правилу «3 σ ».

В общем случае величины γ , $\{\lambda_i\}$ зависимы, так как каждый трафик $j-k$ может прокладывать себе путь в сети по нескольким каналам.

Для оценки $P\{T > T_{\max}\}$ необходимо определить вид распределения случайной величины T , например, как функцию плотности распределения вероятностей $f_T(T)$. Если пренебречь зависимостью величин γ , $\{\lambda_i\}$, то закон распределения для T можно найти на основании следующего. Известно [16], что, если имеются независимые непрерывные случайные величины x и y с соответствующими плотностями вероятностей $f(x)$ и $g(y)$, то плотность вероятности случайной величины $\omega = x/y$ определяется из выражения

$$\varphi(\omega) = \int \left| \frac{z}{\omega^2} \right| f(z) g\left(\frac{z}{\omega}\right) dz. \quad (3.34)$$

Воспользовавшись им, можно найти плотность вероятности для величины $\lambda_i/(\mu C_i - \lambda_i)$. Тогда, согласно центральной предельной теореме, сумму вида $R = \sum_i \frac{\lambda_i}{\mu C_i - \lambda_i}$ можно считать нормально распределенной величиной. В итоге с помощью выражения (3.34) можно найти закон распределения для $T = R/\gamma$.

В общем случае, с учетом зависимости случайных величин, входящих в выражение для T , задача отыскания закона распределения $f_T(T)$ может быть решена с помощью методики, изложенной в работе [16]. Методика предусматривает, в частности, нахождение моментов распределения случайной величины — результата нелинейного преобразования системы случайных величин и аппроксимацию ее распределения с помощью специальных семейств кривых Пирсона или Джонсона.

Другим реальным, хотя и весьма трудоемким способом определения $f_T(T)$, является метод статистических испытаний. В этом случае находится гистограмма, являющаяся оценкой плотности вероятности $f_T(T)$ при различных допустимых $\{C_i\}$ и T_{max} при заданных $\{f_{jk}(\gamma_{jk})\}$. Полученная гистограмма может быть аппроксимирована, например, на основе уже упомянутых семейств кривых или с помощью известных стандартных законов распределения.

Предположим, что в результате применения одного из выше описанных подходов получено выражение для плотности вероятности $f_T(T)$ и соответствующий интегральный закон распределения $F_T(T)$. Тогда $P\{T > T_{max}\} = 1 - F_T(T_{max})$.

Решение задачи (3.31) — (3.33) можно найти, воспользовавшись, например, модифицированным методом Вейсмана [14]. При этом задача нелинейного программирования с ограничениями сводится к задаче без ограничений. Это достигается за счет введения функции штрафа Φ , для которой проводится минимизация:

$$\Phi(C, r) = D(C) + \delta r G(C), \quad (3.35)$$

где

$$D(C) = \sum_{i=1}^M d_i C_i,$$

$$G(C) = F_T(T_{max}) - 1 + P_d.$$

Причем $\delta = 0$, если $G(C) \geq 0$, $\delta = 1$, если $G(C) < 0$. Переменные C_i ($i = 1, M$) могут принимать значения из заданного дискретного множества $\{C_i^*\}$, причем $C_i > \frac{\max \lambda_i}{\mu}$, $\{r\}$ — поэтапно возрастающая последовательность, $r^0 = \frac{1}{G(C^0) D(C^0)}$ — начальное значение.

Поиск минимума Φ ведут с помощью метода Хука и Дживса из некоторой начальной точки $\{C_1^0, C_2^0, \dots, C_M^0\}$. После того, как штрафная функция перестает улучшаться, используют алгоритм ПОИСК. С его помощью находят новую начальную точку, на которой вновь проводят поиск методом Хука и Дживса. Процедура поиска минимума $\Phi(C, r)$ заканчивается, если оба алгоритма дают один и тот же результат. Если результатом поиска методом Хука и Дживса является такой набор пропускных способностей $\{C_i\}$, для которых $G(C) < 0$, то прежние значения корректируют путем умножения на некоторый масштабный множитель, значение которого больше 1. Новое значение r используют на следующем шаге.

Предположим, что множество допустимых значений пропускных способностей $\{C_i^*\}$ упорядочено по возрастанию элементов. Общее их число k, m .

Поиск минимума функции штрафа Φ с помощью метода Хука и Дживса проводят следующим образом. Вычисляют Φ в начальной точке $\{C_i^0\}$, и это значение принимают за временный минимум. Затем делается шаг (в области $\{C_i^*\}$) в сторону увеличения аргумента (переменной), например C_1 . Величины других переменных при этом остаются неизменными. Вычисляют Φ в этой точке: если полученное значение меньше, чем временный минимум, то координаты этой новой точки запоминают и значение Φ в ней принимают за новый временный минимум, если больше, то делается шаг в сторону уменьшения переменной и значение Φ в этой точке сравнивают со значением в начальной.

Вновь либо выбирают (фиксируют) новый временный минимум, либо сохраняют старый. Затем те же операции проделывают для следующей переменной и т. д. до $i = M$. В результате мы получаем новую точку, значение функции Φ в которой меньше, чем в начальной. Из этой точки по переменным, которые изменили свое значение в ходе поиска, делаем шаг в найденном направлении. Те переменные, которые не изменили своего значения в ходе поиска, так и остаются без изменения. Так мы получаем новую точку и вычисляем значение функции Φ . Если оно меньше, чем последний временный минимум, то его принимают за временный минимум. В противном случае сохраняется прежний минимум. Далее увеличиваем весовой коэффициент r и вновь производим поиск временного минимума. И так до тех пор, пока значения переменных в конце этапа поиска отличаются от значений переменных в начале этапа. Если же точки в начале и в конце этапа совпали, то переходим к алгоритму ПОИСК.

В качестве минимизируемой функции берем только $(G(C))$. За исходную принимаем точку, полученную на последнем этапе поиска по методу Хука и Дживса. Текущую переменную при реализации ПОИСК — C_i выбираем последовательно из условия $\min_i \{d_i\}$ с тем, чтобы обеспечить выполнение ограничений (3.33) при наименьшей скорости возрастания $D(C)$. Для каждой из переменных шаги в области $\{C_i^*\}$ делаются вначале в два раза больше, чем в методе Хука и Дживса. После того, как проделаны шаги по всем M переменным, из полученной точки проводим поиск с шагом в три раза большим и т. д. до тех пор, пока значение шага не станет равным $k_M - 1$. При этом, если при совершении шага мы вынуждены выйти за пределы допустимой области, то переменная принимает одно из своих предельно допустимых значений — $C_{i^*}, C_{k_M^*}$. В результате применения алгоритма ПОИСК получаем точку, в которой значение функции штрафа меньше, чем на последнем этапе поиска Хука и Дживса. Полученная точка служит исходной для нового этапа поиска ме-

тодом Хука и Дживса. При переходе к новому этапу поиска Хук и Дживса сохраняем значение r прежнего этапа.

Решение задачи заканчивается либо при равенстве значений функции Φ для двух смежных этапов поиска методом Хука и Дживса, либо по заданному числу этапов поиска.

3.4. ОПТИМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ ПОТОКОВ В СЕТИ ПЕРЕДАЧИ ДАННЫХ

Рассмотрим задачу, обратную задаче выбора пропускных способностей (см. п. 3.3).

Необходимо решить задачу оптимального распределения потоков (РП), когда задана топология сети, внешний трафик $\{\gamma_{ij}\}$ и набор пропускных способностей каналов $\{C_i\}$. Потоки необходимо распределить таким образом, чтобы минимизировать среднюю задержку T .

Выбирая пропускные способности, мы считали, что маршруты в сети фиксированы, т. е. для каждого трафика $j-k$ предполагался только один маршрут.

При решении задачи РП для каждого внешнего трафика $j-k$ может быть более одного пути.

В качестве исходного выражения для T примем, как и прежде, выражение (3.6). Обобщенный алгоритм решения задачи РП представлен в работе [9].

Рассмотрим выражение для T : оно представляет собой сумму слагаемых, каждое из которых зависит лишь от потока в

одном канале. Кроме того, $\frac{\partial T}{\partial (\lambda_i/\mu)} = \frac{C_i}{\gamma (C_i - \lambda_i/\mu)^2}$, $i = 1, 2, \dots, M$

Из выражения видно, что частные производные для любого канала неотрицательны, следовательно, T — выпуклая функция потоков. Если задача минимизации T имеет реализуемое решение, то любой локальный минимум является глобальным минимумом для T .

Алгоритм РП основывается на так называемом *методе отклонения потоков*. Его суть заключается в следующем. Каждому каналу сети соответствует «длина» l_i ,

$$l_i = \frac{\partial T}{\partial (\lambda_i/\mu)} \quad (3.36)$$

С помощью формулы (3.36) можно находить кратчайшие пути связывающие две любые вершины j и k . Кратчайший путь является самым дешевым и лучшим в смысле снижения T , по нему может отклоняться часть потока. Необходимо найти ее и отклонить, «пустить» по кратчайшему пути. Найдя новые «длины» l_i

основании новых, отклоненных потоков, процедуру следует повторить, определяя новую отклоненную часть потока и т. п.

Алгоритм распределения потоков (РП)

Введем вектор потока на n -й итерации алгоритма следующего вида:

$$f^{(n)} = \left(\frac{\lambda_1^{(n)}}{\mu}, \frac{\lambda_2^{(n)}}{\mu}, \dots, \frac{\lambda_M^{(n)}}{\mu} \right), \quad (3.37)$$

где i -я компонента $\lambda_i^{(n)}/\mu$ — суммарная интенсивность потока в i -м канале сети на n -й итерации.

Начальный вектор (поток) $f^{(0)}$ будем считать реализуемым. Это значит, что для каждой компоненты вектора $f^{(0)}$ справедливо соотношение

$$\frac{\lambda_i^{(0)}}{\mu} < C_i. \quad (3.38)$$

Тогда РП можно представить в виде следующей последовательности шагов:

1. Положить $n = 0$.
2. Для каждого $i = 1, 2, \dots, M$ найти

$$l_i = \frac{C_i}{\gamma [C_i - (\lambda_i^{(n)}/\mu)]^2}.$$

3. Определить β_n :

$$\beta_n = \sum_{i=1}^M l_i \frac{\lambda_i^{(n)}}{\mu}.$$

4. Реализовать алгоритм поиска кратчайшего маршрута в сети, используя длины l_i .

Пусть φ_i — результирующий поток, проходящий по i -му каналу, который получается, если весь поток направляется по этим кратчайшим путям. Обозначим такой вектор потоков через

$$\varphi = (\varphi_1, \varphi_2, \dots, \varphi_M).$$

5. Найти b_n :

$$b_n = \sum_{i=1}^M l_i \varphi_i.$$

6. Правило остановки. Если $\beta_n - b_n < \epsilon$, где $\epsilon > 0$ — выбранный допуск, то алгоритм заканчивает работу. В противном случае перейти к шагу 7.

7. Найти такое значение α из интервала $0 \leq \alpha \leq 1$, для которого поток $(1 - \alpha)f^{(n)} + \alpha\varphi$ минимизирует T . Пусть это оптимальное значение обозначено через α^* . Его можно найти с помо-

цию любого метода поиска (например с помощью метода чисел Фибоначчи [48]).

8. Отклонение потоков: положить $f^{(n+1)} = (1 - \alpha^*)f^{(n)} + \alpha^* \varphi$

9. Положить $n = n + 1$. Перейти к шагу 2.

Как уже было указано, данный алгоритм дает решение только в случае реализуемого начального потока $f^{(0)}$.

Алгоритм отыскания реализуемого начального потока заключается в следующем. Введем некоторый масштабный коэффициент $h \leq 1$. С его помощью можно последовательно изменять поток в сети, добиваясь, чтобы он постепенно «вписался» в систему ограничений вида $\lambda_i/\mu < C_i$, ($i = \overline{1, M}$). Конкретная реализация данного принципа изложена ниже.

Алгоритм нахождения реализуемого потока

1. Положим $n = 0$, $h_0 = 1$. Предположим, что мы имеем некоторый начальный (не обязательно реализуемый) поток

$f^{(0)} = \left(\frac{\lambda_1^0}{\mu}, \dots, \frac{\lambda_M^0}{\mu} \right)$. Так как $h_0 = 1$, то поток, который направляется в сеть, равен $f^{(0)}$.

2. Пусть

$$\sigma_n = \max_i \left(\frac{\lambda_i^{(n)}}{\mu C_i} \right).$$

Если $\sigma_n/h_n < 1$, то положить $f^{(0)} = f^{(n)}/h_n$ (если это произошло на шаге $n = 0$, то это значит, что $\lambda_i^{(0)}/\mu < C_i$, $i = \overline{1, M}$, то есть $f^{(0)}$ — поток реализуем и равен $f^{(0)} = f^{(0)}/h_0 = f^{(0)}$). СТОП — это реализуемый поток. Если $\sigma_n/h_n \geq 1$, то необходимо уменьшить h_n , а именно положить $h_{n+1} = h_n [1 - \epsilon_1 (1 - \sigma_n)]/\sigma_n$, где ϵ_1 — заданный допуск, такой, что $0 < \epsilon_1 < 1$.

3. Положить $g^{(n+1)} = \left(\frac{h_{n+1}}{h_n} \right) f^{(n)}$. Суммарный трафик этого потока γh_n . Следует помнить, что это реализуемый поток, но он меньше реального.

4. Взяв в качестве исходного поток $g^{(n+1)}$, провести операцию по его отклонению, выполнив шаги 2, 4, 7, 8 алгоритма РП. Найти соответствующий поток φ и оптимальное значение α^* так, чтобы поток $f^{(n+1)} = (1 - \alpha^*)g^{(n+1)} + \alpha^*\varphi$ минимизировал T .

Если $n = 0$, то перейти к шагу 6; в остальных случаях перейти к шагу 5.

5. Если

$$\left| \sum_{i=1}^M l_i (\varphi_i - g_i^{(n+1)}) \right| < \Theta \text{ и } |h_{n+1} - h_n| < \delta,$$

где Θ, δ — выбранные допуски, то СТОП — задача не имеет

решения (реализуемого потока) при допусках Θ и δ . В остальных случаях перейти к шагу 6.

6. Положить $n = n + 1$ и перейти к шагу 2.

Следует отметить, что условие проверки окончания работы алгоритма на шаге 5 означает, что, во-первых, суммарное приращение задержки меньше заданного Θ , т. е. потоки φ и $g^{(n+1)}$ мало отличаются друг от друга и, во-вторых, изменение масштабного коэффициента h слишком несущественно для того, чтобы пытаться провести еще одну итерацию.

Пример. Отыскание реализуемого начального потока.

Пусть дана сеть передачи данных, которую можно представить следующим неориентированным графом (рис. 3.2). Круг-

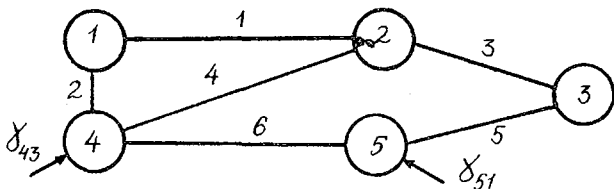


Рис. 3.2. Граф сети передачи данных

ками обозначены узлы сети. Ребра графа — каналы передачи данных. Нумерация узлов и каналов ясна из рисунка.

Задан входной трафик сети: $\gamma_{43} = 40$, $\gamma_{51} = 60$, $\mu = 0,5$, суммарный трафик $\gamma = 100$.

Заданы пропускные способности каналов: $C_1 = 80$, $C_2 = 150$; $C_3 = 95$; $C_4 = 60$; $C_5 = 110$; $C_6 = 120$.

Предположим, что входной трафик заранее распределен по сети так, что интенсивности в каждом канале равны соответственно: $\lambda_1 = 30$, $\lambda_2 = 30$, $\lambda_3 = 50$, $\lambda_4 = 20$, $\lambda_5 = 50$, $\lambda_6 = 50$. Тогда мы имеем следующий начальный поток в сети

$$f^{(0)} = \left(\frac{\lambda_1}{\mu}; \frac{\lambda_2}{\mu}; \frac{\lambda_3}{\mu}; \frac{\lambda_4}{\mu}; \frac{\lambda_5}{\mu}; \frac{\lambda_6}{\mu} \right) = (60, 60; 100; 40; 100; 100).$$

Попытаемся скорректировать начальные условия, отыскав реализуемый начальный поток.

Зададим $\varepsilon_1 = 0,3$.

Тогда

$$1. n = 0; h_0 = 1; f^{(0)} = (60; 60; 100; 40; 100; 100).$$

$$2. \sigma_0 = \max_i \left(\frac{\lambda_i^{(0)}}{\mu C_i} \right) = \max \left(\frac{60}{80}; \frac{60}{150}; \frac{100}{95}; \frac{40}{60}; \frac{100}{110}; \frac{100}{120} \right) =$$

$$= 1,05 \approx 1,1; \quad \sigma_0/h_0 > 1, \text{ тогда}$$

$$h_1 = h_0 \frac{[1 - \varepsilon(1 - \sigma_0)]}{\sigma_0} = \frac{1 + 0,3 \cdot 0,1}{1,1} = 0,94.$$

$$3. g^{(1)} = (h_1/h_0) f^{(0)} = 0,94 (60; 60; 100; 40; 100; 100) = (56,4; 56,4; 94; 37,6; 94; 94).$$

$$4. l_1 = \frac{C_1}{\gamma (C_1 - g_1^{(1)})^2} = \frac{80}{100 (80 - 56,4)^2} = 0,0012;$$

$$l_2 = \frac{C_2}{\gamma (C_2 - g_2^{(1)})^2} = \frac{150}{100 (150 - 56,4)^2} = 0,001;$$

$$l_3 = \frac{95}{100 (95 - 94)^2} = 0,95;$$

$$l_4 = \frac{60}{100 (60 - 37,6)^2} = 0,0012;$$

$$l_5 = \frac{110}{100 (110 - 94)^2} = 0,004;$$

$$l_6 = \frac{120}{100 (120 - 94)^2} = 0,0017.$$

Чтобы найти кратчайший путь в сети для распределения трафика γ_{43} и γ_{51} в нашем примере достаточно рассмотреть все возможные пути движения сообщений от 4-го узла к 3-му и от 5-го к 1-му.

Рассмотрим вновь граф сети с учетом найденных расстояний. Кратчайшее расстояние (4→3) — это путь по каналам 6, 5. Соответственно кратчайшее расстояние (5→1) = 6, 2.

С учетом величины масштабного коэффициента h_1 наш трафик в сети уменьшился и стал равным $\gamma_{43}' = h_1 \gamma_{43}$, $\gamma_{51}' = h_1 \gamma_{51}$. Теперь необходимо его пустить по найденным кратчайшим маршрутам. Если мы это сделаем, то найдем вектор потока $\varphi = (0; 112,8; 0; 0; 75,2; 188)$. Найдем теперь такое $\alpha = \alpha^*$, чтобы поток $f^{(1)} = (1 - \alpha^*) g^{(1)} + \alpha^* \varphi$ минимизировал T .

Учитывая иллюстративность данного примера, поиск α^* будем вести методом перебора среди небольшого числа значений, показав лишь тенденцию к уменьшению T .

Итак, положим $\alpha = 0$.

$$\text{Тогда } f^{(1)} = g^{(1)}, T = \frac{1}{\gamma'} \sum_{i=1}^M \frac{\lambda_i}{\mu C_i - \lambda_i} = \frac{1}{94} \left(\frac{56,4}{80 - 56,4} + \frac{56,4}{150 - 56,4} + \frac{94}{95 - 94} + \frac{37,6}{60 - 37,6} + \frac{94}{110 - 94} + \frac{94}{120 - 94} \right) = 1,08.$$

При $\alpha = 0,1$

$$f^{(1)} = 0,9 g^{(1)} + 0,1 \varphi = (54; 65,28; 90; 36; 97,52; 108,8);$$

$$T = 0,398.$$

При $\alpha = 0,15$

$$f^{(1)} = 0,85 g^{(1)} + 0,15 \varphi = (51; 67,92; 85; 34; 96,28; 113,2);$$

$$T = 0,444.$$

При $\alpha = 0,12$

$$f^{(1)} = 0,88 g^{(1)} + 0,12 \varphi = (52,8; 66,3; 8,8; 35,2; 97; 110; 56);$$

$$T = 0,357.$$

Предположим, мы остановились на $\alpha = 0,12$, считая, что поток $f^{(1)}$ при $\alpha = 0,12$ дает минимальное значение для T . Теперь положим $n = n + 1 = 1$ и перейдем к шагу 2.

Имеем $f^{(1)} = (52,8; 66,3; 88; 35; 97; 110; 56)$;

$$\sigma_1 = \max\left(\frac{52,8}{80}; \frac{66,3}{150}; \frac{88}{95}; \frac{35,2}{60}; \frac{97}{110}; \frac{110,56}{120}\right) = 0,926.$$

Непосредственно из того, что $\sigma_1 = 0,926$, реализуемости потока не следует, так как мы имеем дело с потоком, уменьшенным в соответствии с величиной масштабного коэффициента h_1 . Поэтому вычисляем $\frac{\sigma_1}{h_1} = 0,985 < 1$! Значит поток реализуем.

Величину реализуемого потока получим, разделив каждую компоненту вектора $f^{(1)}$ на h_1 . Итак, $f^{(0)} = \frac{f^{(1)}}{h_1} = (52,8; 66,3; 88; 35; 2; 97; 110; 56)$ — реализуемый поток.

ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ
МНОГОКАНАЛЬНОЙ МНОГОФАЗНОЙ СИСТЕМЫ
ОБРАБОТКИ ИНФОРМАЦИИ
В РЕЖИМЕ РАЗДЕЛЕНИЯ ВРЕМЕНИ

4.1. МОДЕЛЬ СИСТЕМЫ

В главе изложена одна из упрощенных методик анализа и синтеза характеристик АСОИ, математическая модель которой рассматривается как многофазная, многоканальная система массового обслуживания (СМО) с простейшими потоками и экспоненциальным распределением времени обслуживания. Эти допущения обусловлены тем, что, с одной стороны, позволяют упростить математическую модель СМО, а с другой — приводят к получению гарантированных, т. е. верхних оценок, таких показателей качества АСОИ, как время ожидания, длина очереди и т. п. Кроме того, введение указанных допущений вполне целесообразно, так как различные характеристики системы обслуживания мало зависят от закона распределения времени обслуживания, а зависят, главным образом, от среднего значения [15]. И, наконец, погрешность решения при замене пуассоновских потоков пуассоновскими с теми же интенсивностями, как правило, находится в пределах точности исходных данных [17].

Структура процессов обслуживания рассматриваемого варианта АСОИ представлена на рис. 4.1. Данная система состоит из набора процессоров (Π), реализующих фазу обработки, каналов обмена и ввода—вывода (K), осуществляющих связь пользователей и внешней памяти с оперативной памятью системы. Модель взаимодействия пользователей с системой представляет собой суммарный поток заданий с интенсивностью $\lambda_{вх}$.

Техническая реализация АСОИ такого типа может быть осуществлена на базе многомашинного комплекса, использующего ЭВМ третьего поколения, например, типа ЕС.

Характер процесса функционирования данной системы следующий. На вход АСОИ поступает поток заданий от пользователей на обработку (интенсивность суммарного потока заданий $\lambda_{вх}$). Каждое задание (заявка на решение определенной задачи) первоначально проходит фазу ввода (I) и поступает

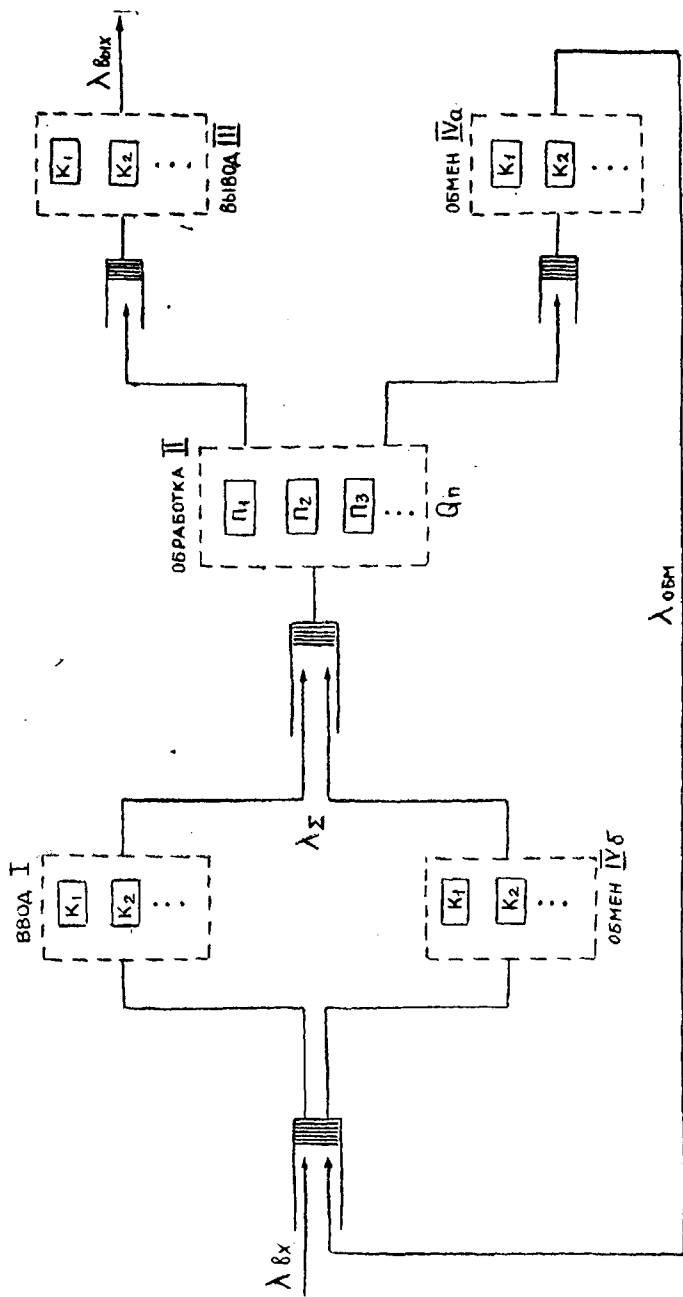


Рис. 4.1. Структура процессов обслуживания в АСОИ

в оперативное запоминающее устройство (ОЗУ). После этого задание поступает на фазу обработки (II). В фазе обработки задание поступает в любой из свободных процессоров (П). Здесь реализуется режим разделения времени, т. е. задание обрабатывается не более чем Q_n единиц времени: Q_n — квант времени обработки. Если выделенный заданию квант времени Q_n оказался достаточным для его обработки, то задание покидает фазу обработки и поступает в фазу вывода (III).

Если по окончании кванта времени задание полностью не обработано, то оно поступает в фазу обмена (IV,а). Здесь реализуется перепись промежуточных результатов во внешнюю память (ВП). Из ВП задания поступают на дообработку в один из процессоров через каналы обмена (IV,б).

В данной модели принято, что дообрабатываемые задания встают в общую очередь с впервые поступающими заданиями, их выбирают оттуда в процессоры в порядке поступления. На всех фазах обслуживания в данной системе принята дисциплина «первый пришел — первый обслужен».

В дальнейшем при расчетах будем использовать следующие основные обозначения: Q_n — величина кванта обработки в процессоре (с); $\lambda_{вх}$ — интенсивность суммарного входного потока заданий от пользователей ($1/с$); ρ_n — коэффициент загрузки фазы обработки; $r_{выв}$ — среднее время (с), затрачиваемое каналом на вывод единицы объема информации (байт); $N_{выв}$ — среднее количество байтов, выводимых из системы по одному заданию; $N_{обм}$ — среднее количество байтов, характеризующее процесс обмена между оперативной и внешней памятью по одному заданию; T_0 — требуемое среднее время (с) обработки задания в процессоре (без учета квантования); $r_{вв}$ — среднее время (с), затрачиваемое каналом на ввод единицы объема информации (байт); $r_{обм}$ — среднее время для фазы обмена; $N_{вв}$ — средний объем информации (байт), вводимой по каждому из заданий; $\rho_{выв}$ — коэффициент загрузки фазы вывода; $\rho_{обм}$ — коэффициент загрузки фазы обмена; $\rho_{вв}$ — коэффициент загрузки фазы ввода; λ_a — интенсивность потока заданий, поступающих в очередь на обработку в первый раз; $\lambda_{обм1}$ — интенсивность потока заданий, выводимых во внешнюю память; $\lambda_{обм2}$ — интенсивность потока заданий, поступающих на дообработку; M_n — количество процессоров в фазе обработки; $M_{вв}$ — количество каналов фазы ввода; $M_{выв}$ — количество каналов фазы вывода; $M_{обм1}$, $M_{обм2}$ — количество каналов фазы обмена; $1/\mu$ — среднее время обслуживания задания в соответствующей фазе.

4.2. МЕТОДИКА ОЦЕНКИ ХАРАКТЕРИСТИК И ВЫБОРА ПАРАМЕТРОВ СИСТЕМЫ

Задачи анализа и выбора параметров

Необходимо на основании заданных технических требований к системе и принятой модели функционирования оценить: количественный состав технических средств системы (количество процессоров, каналов ввода-вывода, обмена); временные параметры, характеризующие задержки, возникающие при выполнении заданий на различных фазах; интегральные показатели задержки в системе.

Исходные данные

Будем предполагать, что задана номенклатура технических средств для построения АСОИ, т. е. выбраны типы процессора и каналов ввода-вывода, обмена.

Таким образом, заданы:

B — быстродействие процессора, опер/с;

$r_{вв} = r_{выв} = r_{обм}$ (с) предполагается, что для операций ввода-вывода и обмена используют однотипные каналы).

Относительно характеристик взаимодействия пользователей с системой заданы:

$\lambda_{вх}$ — интенсивность входного потока заданий, 1/с;

Θ — трудоемкость алгоритмов обработки заданий в процессоре, опер;

$N_{вв} = N_{выв}$, байт;

$N_{обм}$, байт;

ρ_n — допустимый коэффициент загрузки процессора;

Q_n — величина кванта обработки, с.

Поскольку заданы величины B и Θ , можем определить

$$T_0 = \Theta/B.$$

Прежде чем перейти к непосредственному рассмотрению методики расчета характеристик и выбору параметров системы, сделаем ряд замечаний.

Так как система обслуживания рассматривается в стационарном режиме и предполагается отсутствие потерь заданий, то справедливы следующие соотношения (см. рис. 4.1):

$$\lambda_{вх} = \lambda_{вых}, \quad (4.1)$$

$$\lambda_{обм} = \lambda_{\Sigma} - \lambda_{вых}. \quad (4.2)$$

Кроме того учтем, что для режима разделения времени при экспоненциальном распределении времени обработки вероят-

ность того, что задание покинет систему после окончания квадрата обработки Q_n , составляет

$$P_{\text{вых}} = 1 - e^{-\mu_0 Q_n}, \quad (4.3)$$

где $\mu_0 = 1/T_0$.

Тогда $\lambda_{\text{вых}} = \lambda_{\Sigma} P_{\text{вых}}$; с учетом соотношений (4.1) и (4.2)

$$\lambda_{\text{обм}} = \lambda_{\text{вх}} (\bar{k} - 1), \text{ где } \bar{k} = 1/P_{\text{вых}}.$$

Определение количественного состава технических средств системы

Оценку необходимого количества процессоров и каналов будем производить, исходя из вполне естественного и обоснованного требования сбалансированности по загрузке различных фаз системы. Поскольку в исходных данных мы имеем значение допустимого коэффициента загрузки фазы обработки ρ_n , то вышеуказанное требование может быть выражено системой неравенств вида

$$\left. \begin{aligned} \rho_{\text{вв}} &\leq \rho_n \\ \rho_{\text{выв}} &\leq \rho_n \\ \rho_{\text{обм}_1} &\leq \rho_n \\ \rho_{\text{обм}_2} &\leq \rho_n \\ \rho_{\text{вв}} &\leq \rho_n \end{aligned} \right\} \quad (4.4)$$

1. Оценим потребное число процессоров M_n для фазы обработки. Для этого учтем, что

$$\rho_n = \frac{\lambda_{\Sigma}}{M_n \mu_{\text{обр}}},$$

где $\mu_{\text{обр}} = 1/\bar{T}_{\text{обр}}$;

$\mu_{\text{обр}}$ — результирующая интенсивность обработки в процессоре с учетом квантования.

Величину M_n найдем из выражения

$$\frac{\lambda \cdot \bar{k}}{M_n \mu_0 \bar{k}} = \frac{\lambda}{M_n \mu_0} \leq \rho_n. \quad (4.5)$$

В выражении (4.5) использовано то обстоятельство, что $\mu_{\text{обр}} = \mu_0 \bar{k}$, так как распределение времени обработки в главном процессоре с учетом квантования имеет вид

$$F_{\text{обр}}(t) = \begin{cases} 1 - e^{-\mu_0 t} & 0 < t \leq Q_n \\ 1, & t > Q_n, \end{cases} \quad (4.6)$$

а среднее значение распределения (первый начальный момент)

$$\bar{T}_{\text{обр}} = 1 / \mu_{\text{обр}} = 1 / \mu_0 \bar{k}. \quad (4.7)$$

Кроме того, $\sigma^2 = \bar{T}_{\text{обр}}^2$ в силу экспоненциальности распределения. Найденное значение $M_{\text{п}}$ округлим до ближайшего целого числа.

После этого сделаем пересчет коэффициента загрузки для фазы обработки. Найдем фактическое значение для $\rho_{\text{п}}$ из выражения (4.5), подставив в него округленное значение $M_{\text{п}}$. Найденное значение обозначим через $\rho_{\text{п}}^*$.

2. Число каналов ввода и вывода ($M_{\text{вв}}$, $M_{\text{выв}}$) определяем из соотношений

$$\rho_{\text{выв}} = \frac{\lambda_{\text{вых}} N_{\text{выв}} r_{\text{выв}}}{M_{\text{выв}}} \leq \rho_{\text{п}}^*, \quad (4.8)$$

$$\rho_{\text{вв}} = \frac{\lambda_{\text{вх}} N_{\text{вв}} r_{\text{вв}}}{M_{\text{вв}}} \leq \rho_{\text{п}}^*, \quad (4.9)$$

3. Для определения числа каналов обмена рассмотрим среднее время, необходимое для реализации обмена ОЗУ-ВП: $\bar{t}_{\text{обм}} = t_{\text{дост}} + \bar{t}_{\text{пер}}$, где $\bar{t}_{\text{пер}} = N_{\text{обм}} r_{\text{обм}}$, $t_{\text{дост}}$ — среднее время доступа к ВП — время, необходимое для поиска соответствующей зоны во ВП при записи или считывании. Оно зависит от типа внешних накопителей и дисциплины доступа [18].

Если не рассматривать время доступа к ВП, то

$$\rho_{\text{обм1}} = \frac{\lambda_{\text{обм}}}{M_{\text{обм}} \mu_{\text{обм1}}} = \frac{\lambda (\bar{k} - 1) \bar{N}_{\text{обм}} r_{\text{обм}}}{M_{\text{обм1}}} \leq \rho_{\text{п}}^*. \quad (4.10)$$

Тогда из условия (4.10) определяем $M_{\text{обм1}}$. Аналогично находим число каналов для второй фазы обмена $M_{\text{обм2}}$.

Оценка характеристик функционирования подсистем

Проводим анализ характеристик функционирования каждой из подсистем (фаз) АСОИ.

Необходимо оценить:

средние задержки в обслуживании заданий — \bar{t}_w, \bar{t}_q ;

средние квадратические отклонения (с.к.о.) задержек — $\sigma_{t_w}, \sigma_{t_q}$;

средние длины очередей — \bar{w}, \bar{q} ;

коэффициенты использования процессоров и каналов — $k_{\text{исп}}$.

Приведем необходимые расчетные соотношения для многоканальных СМО при обслуживании в порядке поступления [19—21].

Вероятность того, что все обслуживающие устройства данной фазы заняты,

$$B_p = P\{q \geq M\} = \sum_{n=M}^{\infty} P\{q = n\} \quad (4.11)$$

где M — число обслуживающих устройств данной фазы (процессоров или каналов);

q — количество заданий, находящихся на обслуживании и ожидающих в очереди.

При малом M величина B_p может быть подсчитана по формуле

$$B_p = \frac{(M\rho)^M}{M!(1-\rho)} P_0 = P_M \frac{1}{1-\rho}, \quad (4.12)$$

где

$$P_M = \frac{(M\rho)^M P_0}{M!};$$

$$P_0 = \frac{1}{\sum_{n=0}^{M-1} \frac{(M\rho)^n}{n!} + (M\rho)^M \frac{1}{M!(1-\rho)}}; \quad (4.13)$$

ρ — коэффициент загрузки соответствующей фазы: $\rho = \frac{\lambda}{M\mu}$

Вероятность того, что занято ровно n обслуживающих устройств,

$$P_n = \begin{cases} \frac{1}{n!} (\rho M)^n P_0 & 1 \leq n \leq M \\ \frac{(\rho M)^n}{M! M^{n-M}} & n > M. \end{cases} \quad (4.14)$$

В общем случае B_p находим из следующего выражения

$$B_p = \left[1 - \left(\frac{\sum_{n=0}^{M-1} \frac{(M\rho)^n}{n!}}{\sum_{n=0}^M \frac{(M\rho)^n}{n!}} \right) \right] \left[1 - \rho \left(\frac{\sum_{n=0}^{M-1} \frac{(M\rho)^n}{n!}}{\sum_{n=0}^M \frac{(M\rho)^n}{n!}} \right) \right]. \quad (4.15)$$

Обозначим: \bar{w} — число заданий, ждущих обслуживания; t_w — время ожидания обслуживания; q — число ждущих и обслуживаемых заданий; t_q — суммарное время ожидания и обслуживания. Тогда

$$\bar{w} = B_p \frac{\rho}{1-\rho}; \quad (4.16)$$

$$\bar{q} = B_p \frac{\rho}{1-\rho} + M\rho; \quad (4.17)$$

$$\sigma_w = \frac{1}{1-\rho} \sqrt{B_p \rho (1 + \rho - B_p \rho)} \quad (4.18)$$

$$\bar{t}_w = \frac{B_p}{M\mu (1-\rho)}; \quad (4.19)$$

$$\bar{t}_q = \bar{t}_w + \frac{1}{\mu}; \quad (4.20)$$

$$\bar{l}_w = \frac{\bar{w}}{\lambda}, \quad \bar{l}_q = \frac{\bar{q}}{\lambda}; \quad (4.21)$$

$$\sigma_{t_w} = \frac{1}{M \mu (1 - \rho)} \sqrt{B_p (2 - B_p)}; \quad (4.22)$$

$$\sigma_{t_q} = \frac{1}{M \mu (1 - \rho)} \sqrt{B_p (2 - B_p) + M^2 (1 - \rho)^2}. \quad (4.23)$$

Среднее число занятых устройств (каналов) обслуживания

$$\bar{M} = \sum_{n=1}^{M-1} P_n n + M \sum_{n=M}^{\infty} P_n. \quad (4.24)$$

Коэффициент использования устройства обслуживания

$$k_{исп} = \frac{\bar{M}}{M}. \quad (4.25)$$

В соотношениях (4.12)–(4.25) λ — интенсивность потока заданий, поступающего на данное устройство обслуживания; для устройства ввода $\lambda = \lambda_{вх}$; в фазе обработки $\lambda = \lambda_{вх} \bar{k}$; в канале обмена $\lambda = \lambda_{вх} (\bar{k} - 1)$ и т. д.

Оценка интегральных характеристик функционирования системы

На предыдущем этапе анализа получены оценки качества функционирования системы для каждой из фаз обслуживания.

Для интегральной оценки качества АСОИ необходимо определить характеристики времени пребывания задания от момента входа в систему до момента выхода из нее. Это время пребывания можно назвать *временем реакции АСОИ на задание пользователя*.

Для определения среднего значения и дисперсии времени необходимо определить среднее и дисперсию числа циклов, которые совершает задание в системе, прежде чем покинет ее.

Распределение числа циклов в силу отсутствия «последствия» при экспоненциальном распределении времени полной обработки является геометрическим, т. е. вероятность совершения m циклов обработки в процессоре определяется согласно выражению

$$P_m = (1 - P_{вых})^{m-1} P_{вых}.$$

Среднее и дисперсию числа циклов находят из выражений [19]

$$N_n = \sum_{m=1}^{\infty} m P_m = \frac{1}{P_{вых}} = \bar{k},$$

$$\sigma_n^2 = \sum_{m=1}^{\infty} m^2 P_m - (\bar{N}_n)^2 = \bar{k} (\bar{k} - 1).$$

Тогда среднее и дисперсия суммарного времени обработки m_0 будут найдены из соотношений [20]

$$\begin{aligned}\bar{T}_{\Sigma} &= \bar{T} \bar{N}_{\Pi} = \bar{T}_{\Pi} \bar{k}, \\ \sigma_{T_{\Sigma}}^2 &= \bar{k} (\bar{k} - 1) T_{\Pi}^2 + \sigma_{T_{\Pi}}^2 \bar{k},\end{aligned}$$

где $\bar{T}_{\Pi} = \bar{t}_w + \bar{T}_{\text{обр}}$.

Беличину \bar{t}_w определим из соотношения (4.19), а $\bar{T}_{\text{обр}}$ — из выражения (4.7). Число процессоров было определено ранее из выражения (4.5)

$$\sigma_{T_{\Pi}}^2 = \sigma_{t_w}^2 + \sigma_{T_{\text{обр}}}^2,$$

где $\sigma_{t_w}^2$ определяется из формулы (4.22), а $\sigma_{T_{\text{обр}}}^2 = \bar{T}_{\text{обр}}^2$. Совершенно аналогично для фазы обмена среднее число циклов обмена

$$\bar{N}_{\text{обм}} = \sum_{i=1}^{\infty} (i-1) P_{\text{вых}} (1 - P_{\text{вых}})^{i-1} = (\bar{k} - 1),$$

$$\sigma_{N_{\text{обм}}}^2 = \sum_{i=1}^{\infty} (i-1)^2 P_{\text{вых}} (1 - P_{\text{вых}})^{i-1} - (\bar{N}_{\text{обм}})^2 = \bar{k} (\bar{k} - 1).$$

Далее, если допустить, что число каналов вывода из внешней памяти равно числу каналов ввода, то $\bar{T}_{\Sigma \text{ обм}} = 2 \bar{T}_{\text{обм}} (\bar{k} - 1)$,

$$\sigma_{T_{\Sigma \text{ обм}}}^2 = 2 (\bar{k} - 1) (\bar{k} \bar{T}_{\text{обм}}^2 + \sigma_{T_{\text{обм}}}^2),$$

где значения $\bar{T}_{\text{обм}}$, $\sigma_{T_{\text{обм}}}^2$ подсчитывают из соотношений (4.20) и (4.23).

Тогда суммарное время реакции АСОИ

$$\begin{aligned}\bar{T}_{\Sigma} &= \bar{T}_{\text{вв}} + \bar{T}_{\Pi} \bar{k} + 2 \bar{T}_{\text{обм}} (\bar{k} - 1) + \bar{T}_{\text{выб}} \\ \sigma_{T_{\Sigma}}^2 &= \sigma_{\text{вв}}^2 + \bar{k} [(\bar{k} - 1) \bar{T}_{\Pi}^2 + \sigma_{T_{\Pi}}^2] + \sigma_{\text{выб}}^2 + \\ &+ 2 (\bar{k} - 1) (\bar{k} \bar{T}_{\text{обм}}^2 + \sigma_{T_{\text{обм}}}^2),\end{aligned}$$

где $\bar{T}_{\text{вв}}$, $\bar{T}_{\text{выб}}$, $\sigma_{\text{вв}}^2$, $\sigma_{\text{выб}}^2$ определяют при оценке качества функционирования соответствующих фаз из соотношений (4.20), (4.23).

ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ (АСОИ)

5.1. ПОСТАНОВКА ЗАДАЧИ ОПТИМИЗАЦИИ ПАРАМЕТРОВ

Оптимизацию параметров конкретного варианта построения АСОИ проводят на основе математической модели этой системы, которая связывает основные технические параметры оптимизируемой системы с множеством ограничений, налагаемых при проектировании, с внешними параметрами, а также с показателями качества проектируемой системы. Задача оптимизации является максимализация или минимизация принятого критерия эффективности АСОИ. В общем виде задача оптимизации АСОИ может быть сформулирована в следующем виде.

Задан некоторый критерий эффективности АСОИ

$$E = E(x_1, x_2, \dots, x_k), \quad (5.1)$$

где x_1, x_2, \dots, x_k — параметры системы.

Для нахождения оптимальной системы требуется найти вектор технических параметров $\bar{x} = x_1^*, x_2^*, \dots, x_k^*$, при котором заданный показатель эффективности принимает экстремальное значение на множестве допустимых векторов

$$E = \text{extr}(E(x)); \quad (5.2)$$

$$\xi_i(\bar{x}) < b_i, \quad (5.3)$$

где ξ_i — налагаемые ограничения;

b_i — граничные значения,

i — номер ограничений.

Существенно важным при рассмотрении задачи оптимизации является тот факт, что часть параметров АСОИ может принимать только целочисленные или дискретные значения. При оптимизации такого рода систем возникают специфические трудности, заключающиеся в том, что возможность применения классических, основанных на дифференцировании, методов оптимизации практически исключается.

Наиболее целесообразно решать описанный класс задач методом динамического программирования или случайного поиска [7, 12].

Рассматриваемые варианты построения АСОИ могут отличаться числом и производительностью ЭВМ, типом и характером

ристикami запоминающих устройств, операционными системами и т. д. Все эти величины должны входить в вектор технических параметров системы. Однако в этом случае в силу большой размерности и сложной, а порой и неизвестной на стадии проектирования взаимосвязи между показателями эффективности и техническими параметрами АСОИ, задача анализа системы очень трудна. Поэтому мы рассмотрим более простой случай, когда АСОИ является однопроцессорной, а в качестве основных технических параметров выступают быстродействие процессора, дисциплина обслуживания и порядок назначения приоритетов потокам заявок. Эти технические параметры системы в наибольшей степени влияют на временные характеристики АСОИ, которые являются определяющими в системах оперативной обработки информации. Организуя и анализируя обслуживание при сделанных допущениях, сопоставим величину выигрыша, получаемого от использования той или иной дисциплины, с затратами ресурсов на ее реализацию.

Ограниченность реальных характеристик и структуры конкретной системы, а также используемая дисциплина обслуживания снижают идеальную эффективность на некоторую величину, которую рассматривают как *функционал потерь*, зависящий от потока заявок и параметров системы. Потери можно представить суммой следующих составляющих: потери от ожидания заявками начала обслуживания, от простоя обслуживающего прибора, от прерывания обслуживания, вызванного прибытием заявок высшего приоритета. Неравноценность заявок по допустимому времени ожидания позволяет изменять суммарные потери в обслуживании заявок, даже при фиксированных структурных характеристиках системы, меняя дисциплину обслуживания или последовательность выбора заявок при заданной дисциплине.

Проектирование АСОИ обычно ведут в условиях противоборства двух факторов. С одной стороны, чем более быстродействующий процессор мы выберем, тем меньше будут величины среднего суммарного штрафа из-за ожидания заявок в очередях, а с другой стороны, чем выше быстродействие процессора, тем больше вероятность его простоев, что естественно вытекает из стохастического характера поступления заявок в систему. Поэтому вполне логично предположить существование некоторого оптимума, позволяющего определить компромиссное значение быстродействия процессора с учетом вышеназванных противоречий. В качестве критериальной функции при таком подходе может быть использован функционал вида [5].

$$C_B^{(s)} = \beta_0 \eta(B) + \sum_{i=1}^M \beta_i \lambda_i \omega_i^{(s)}(B), \quad (5.4)$$

$$\text{где } \eta(B) = 1 - \sum_{i=1}^M \lambda_i T_i = 1 - \left(\frac{1}{B}\right) \sum_{i=1}^M \lambda_i \Theta_i; \quad (5.5)$$

T_i, Θ_i — среднее время обслуживания и трудоемкость выполнения заявок i -го типа;

β_i, ω_i — штраф за единицу времени ожидания и средняя длительность ожидания заявок i -го типа;

M — число типов заявок;

λ_i — интенсивность поступления заявок i -го типа;

s — индекс используемой дисциплины обслуживания.

Таким образом, показатель эффективности $C_B^{(s)}$ является функционалом, зависящим от трех величин: дисциплины обслуживания, быстродействия, порядка назначения приоритетов.

Как было отмечено выше, решение задачи оптимизации в общем виде достаточно сложно в силу ее комбинаторного характера и большой размерности.

Характер ограничений, общий вид которых записан в форме (5.3), может быть различным. В АСОИ, как правило, задаются ограничения по временные характеристики. Можно выделить два типа АСОИ: с *абсолютными* ограничениями на время пребывания заявок, когда требования по времени ожидания должны выполняться для всех заявок, и с *относительными* ограничениями, когда ограничения налагаются на средние времена пребывания заявок. Возможен и третий случай, когда ограничения вообще не накладываются. Учитывая перечисленные факторы целесообразно для решения задачи применить упрощенную методику [5].

Идея состоит в разбиении общей задачи на ряд частных, показатели эффективности которых зависят лишь от одного параметра. Так, на первом этапе находят нижнюю границу быстродействия системы из условия наличия стационарного режима или выполнения ограничений на время ожидания, если они существуют, и закона сохранения времени ожидания. На втором этапе ищут дисциплину обслуживания при фиксированном быстродействии. Причем с целью облегчения поиска достаточно простой дисциплины найденное на предыдущем этапе значение минимального быстродействия несколько завышается (на 10—20%). И, наконец, на последнем этапе для выбранной дисциплины находят так называемое оптимальное быстродействие, т. е. «завышенное» на предыдущем шаге быстродействие мы снова пытаемся несколько уменьшить, имея в виду ограничение на время ожидания, нижнюю границу быстродействия и руководствуясь экономическими соображениями о наиболее полной загрузке процессора.

5.2. ОПРЕДЕЛЕНИЕ МИНИМАЛЬНО ДОПУСТИМОГО БЫСТРОДЕЙСТВИЯ ПРОЦЕССОРА

Для определения нижней оценки быстродействия рассмотрим АСОИ с многомерным потоком заявок интенсивности λ_i ($i = \overline{1, M}$) и параметрами трудоемкости алгоритмов обработки заявок типа i — Θ_i и $\Theta_i^{(2)}$, где $\Theta_i^{(2)}$ — второй начальный момент.

Минимально возможное быстродействие для случая отсутствия ограничений на время ожидания определяем почти точно из условия наличия стационарного режима в АСОИ, т. е.

$$\sum_{i=1}^M \lambda_i T_i < 1,$$

учитывая, что $T_i = \Theta_i / B_i$ ($i = \overline{1, M}$) окончательно получим

$$B_{\min} > \sum_{i=1}^M \lambda_i \Theta_i. \quad (5.6)$$

Если на среднее время ожидания (или пребывания) заявок наложены ограничения, например: $\omega_1 \leq \omega_1^*$; $\omega_2 \leq \omega_2^*$; ...; $\omega_M \leq \omega_M^*$, то постановка задачи несколько сложнее. В этом случае можно воспользоваться законом сохранения времени ожидания

$$\sum_{i=1}^M \rho_i \omega_i = R \omega_0 = \text{const}, \quad (5.7)$$

где $\rho_i = \lambda_i T_i$;

$$R = \rho_1 + \rho_2 + \dots + \rho_M;$$

ω_0 — среднее время ожидания заявок при бесприоритетном обслуживании, определяемое как

$$\omega_0 = \sum_{i=1}^M \frac{\lambda_i T_i^{(2)}}{2(1-R)}, \quad R < 1. \quad (5.8)$$

С учетом ограничений на среднее время ожидания заявок $\omega_i \leq \omega_i^*$ равенство (5.8) можно переписать в виде

$$\sum_{i=1}^M \rho_i \omega_i^* > R \omega_0,$$

т. е.
$$\sum_{i=1}^M \rho_i (\omega_i^* - \omega_0) > 0 \quad (5.9)$$

Таким образом, если неравенство (5.9) выполнимо, то среднее множества дисциплин обслуживания существует такая, для которой выполнимы все ограничения вида $\omega_i \leq \omega_i^*$.

Из выражения (5.9) нетрудно определить и значения B_{\min}

Для чего в него следует подставить выражение (5.8) и произвести необходимые алгебраические преобразования [5]:

$$B_{\min} > \frac{1}{2} \sum_{i=1}^M \lambda_i \Theta_i + \sqrt{\frac{1}{4} \sum_{i=1}^M \lambda_i Q_i \left(\sum_{i=1}^M \lambda_i Q_i + \frac{2 \sum_{i=1}^M \lambda_i \Theta_i (\Theta_i^2)}{M \sum_{i=1}^M \lambda_i \Theta_i \omega_i^*} \right)}. \quad (5.10)$$

На практике значение B_{\min} обычно выбирают с запасом 10—20% по отношению к вычисленному по формуле (5.10) с тем, чтобы расширить класс дисциплин, удовлетворяющих поставленным условиям.

5.3. ВЫБОР ДИСЦИПЛИНЫ ОБСЛУЖИВАНИЯ

Задачи выбора дисциплины обслуживания и распределения приоритетов между потоками заявок (если выбрана приоритетная дисциплина) решают обычно после выбора минимального быстродействия процессора. Обе эти задачи носят комбинаторный характер. Отметим, что в силу конечности числа перестановок на s (число возможных дисциплин) или из M (число потоков информации) оптимальная дисциплина или оптимальная последовательность приоритетов всегда существует.

Задача выбора дисциплины обслуживания решена только для случая отсутствия ограничений на время пребывания (ожидания) заявок в АСОИ для класса абсолютных и относительных приоритетов. Для сравнения дисциплин обслуживания можно использовать критерий вида (5.4). Более того, поскольку первое слагаемое в этом критерии не зависит от дисциплины обслуживания, то для оценки можно использовать лишь второе слагаемое указанного критерия.

Сравнивая последовательно беспriorитетную дисциплину и дисциплину с относительными приоритетами, а затем дисциплину с относительными и абсолютными приоритетами, мы приходим к одним и тем же достаточным условиям выигрыша от введения относительных приоритетов по отношению к беспriorитетной дисциплине и абсолютных приоритетов по отношению к относительным, а именно:

$$\alpha_i / T_i > \alpha_{i+1} / T_{i+1}, \quad i = 1, 2, \dots, M-1. \quad (5.11)$$

Приведенное неравенство не зависит от интенсивностей входных потоков заявок и высших моментов распределения длительностей обслуживания, в нем α_i, α_{i+1} — штрафы за задержку заявок соответствующего типа.

Теперь перейдем к выбору оптимальной последовательности приоритетов. До сих пор предполагалось, что последовательность приоритетов совпадает с последовательностью 1, 2, ..., M . Потери, связанные с образованием очереди, зависят только от среднего времени ожидания, т. е. мы снова можем в качестве критериальной функции использовать второе слагаемое функции (5.4). Задача состоит в выборе такой последовательности приоритетов, которая минимизирует средние потери. Для вывода условий оптимального назначения приоритетов поменяем порядок приоритетов заявок $(k-1)$ -го и k -го типов. Это приведет к новым средним потерям C' . Знак разности $C-C'$ покажет, как изменятся потери от перемены порядка приоритетов. Проведя необходимые подстановки и преобразования, можно показать, что $C > C'$ тогда и только тогда, когда

$$a_k / T_k > a_{k+1} / T_{k+1}, \quad (5.12)$$

т. е. средние потери от ожидания заявок в очереди можно уменьшить, изменив порядок приоритетов таким образом, чтобы большему значению отношения a_k / T_k соответствовал и больший приоритет. Однако это простое правило справедливо лишь для относительных приоритетов и при экспоненциальном законе распределения времени обслуживания.

Для дисциплины с абсолютными приоритетами, дообслуживанием и при произвольном законе распределения времени обслуживания условие оптимизации последовательности приоритетов принимает более сложный вид:

$$2T_i a_i / T_i^{(2)} > a_j / T_j \quad \text{при } i < j, \quad (5.13)$$

$$2T_i a_i / T_i^{(2)} < a_j / T_j \quad \text{при } i > j, \quad (5.14)$$

где $T_i^{(2)}$ — второй момент распределения времени обслуживания заявок, при одновременном выполнении ранее введенных ограничений

$$a_i / T_i > a_j / T_j \quad \text{при } i < j; \quad a_i / T_i < a_j / T_j \quad \text{при } i > j.$$

Запись $i < j$ означает, что поток с номером i имеет абсолютный приоритет более высокий, чем поток с номером j .

Для более сложных дисциплин обслуживания, например для смешанной, когда для обслуживания одной части потоков заявок используется дисциплина обслуживания с абсолютными, а для другой части — с относительными приоритетами, такие простые рекомендации найти не удастся. Приходится решать задачу поиска оптимальной дисциплины обслуживания и оптимального порядка назначения приоритетов методом перебора.

Наиболее часто встречающиеся дисциплины обслуживания сведены в табл. 5.1.

Из приведенной классификации (рис. 5.1) видно, что приоритетные дисциплины подразделяются на внесистемные и внутри-

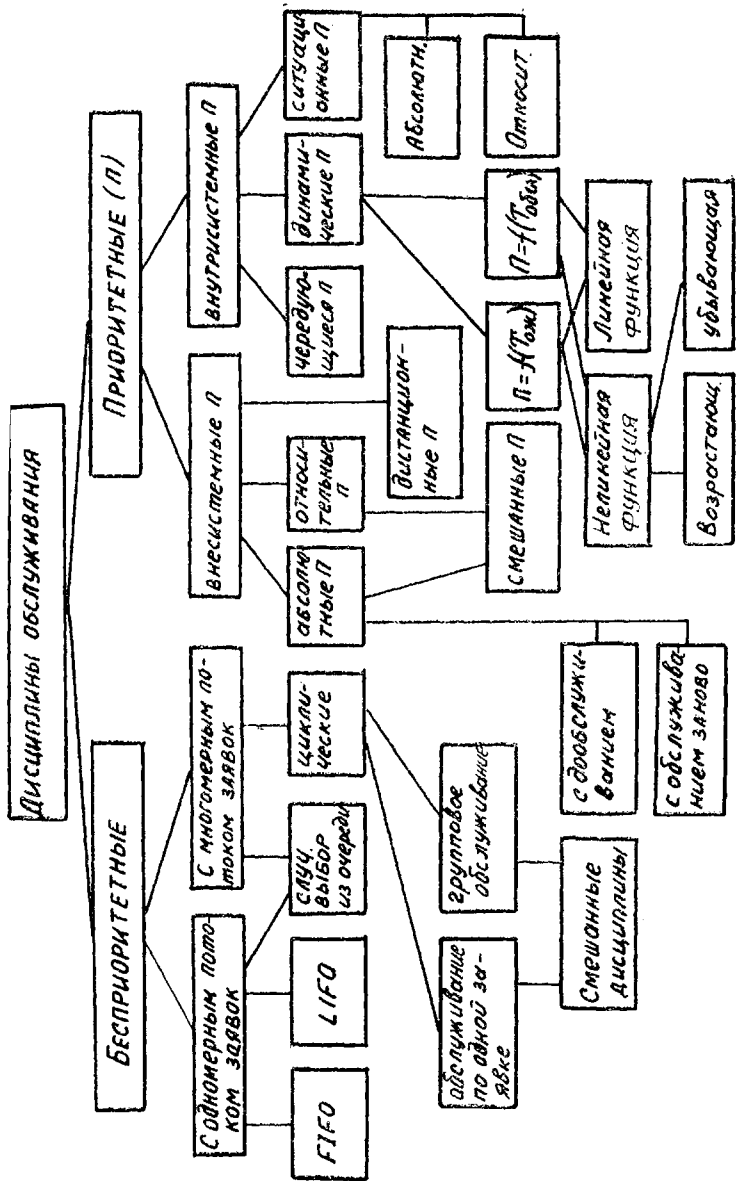


Рис. 5.1. Классификация дисциплин обслуживания

системные. Для *внесистемных* дисциплин задание приоритета тому или иному потоку заявок полностью определяется характеристиками данного класса требований. В случае использования *внутрисистемных* приоритетов приоритет каждого класса заявок зависит от текущего состояния процесса обработки заявок, например, от оставшегося до конца времени обработки заявок или от длительности пребывания заявок в очереди и т. п.

Сравнивая приведенные внутрисистемные и внесистемные приоритеты дисциплины (см. табл. 5.1) отметим, что последние обладают одним существенным преимуществом: в случае их использования для определенных классов заявок (с более высоким приоритетом) может установиться равновесие даже в случае, если в системе в целом установившийся режим отсутствует. К недостаткам же внесистемных дисциплин относят обычно отсутствие у них каких-либо управляющих параметров, с помощью которых можно было бы регулировать характеристиками качества обслуживания для каждого класса требований. Исключением являются смешанные дисциплины, где путем изменения распределения абсолютных и относительных приоритетов можно в некоторой степени осуществлять регулирование.

5.4. ВЫБОР ОПТИМАЛЬНЫХ ПАРАМЕТРОВ

Выше мы определили минимальное быстродействие АСОИ, при котором существует хотя бы одна дисциплина обслуживания, удовлетворяющая всем требованиям оперативности, предъявляемым к данной системе.

Также было отмечено, что действительное быстродействие следует выбирать из условия $B > B_{\min}$ — это позволит расширить класс применяемых дисциплин и облегчить поиск оптимальной. При таком подходе мы создаем себе некоторый «запас» быстродействия, а в обмен расширяем множество допустимых дисциплин. Однако встает вопрос: какую же из этого множества подходящих дисциплин нам выбрать? Очевидно, надо выбрать ту дисциплину, которая наиболее проста в реализации, т. е. может быть реализована с наименьшими материальными затратами. Таким образом, мы переходим к последнему этапу синтеза АСОИ, связанному с определением быстродействия процессора, которое, обеспечивая заданное качество обслуживания заявок, позволяет создать систему с наименьшими материальными затратами.

Одним из показателей уровня материальных затрат в АСОИ является ее коэффициент загрузки R (или коэффициент простоя $\eta = 1 - R$). Критерий эффективности (5.4), построенный с учетом качества обслуживания заявок и коэффициента загрузки системы, позволяет определить допустимое быстродействие про-

№ п/п	Название дисциплины	Литературный источник	Выражение для определения времени ожидания заявки в очереди	Примечание
1.	Бесприоритетные дисциплины обслуживания, одномерный поток заявок; обслуживание в порядке поступления; обслуживание в обратном порядке; случайный и равновероятный выбор на обслуживание	[6] [22] [22]	$\omega = \frac{\omega_0}{1 - \rho} = \frac{\lambda T^2 (1 - v^2)}{2(1 - \rho)},$ <p>где $v = \sigma_T / m_T$ — коэффициент вариации времени обслуживания заявок</p>	Показано [5], что время ожидания в очереди одинаково
2.	Бесприоритетные дисциплины, многомерный поток; обслуживание в порядке поступления;	[22]	$\omega_i = \omega = \frac{\frac{1}{2} \sum_{l=1}^M \rho_l T_l (1 + v_l^2)}{1 - R},$ <p>где $R = \sum_{l=1}^M \rho_l$.</p>	
3.	Многомерное обслуживание с весовыми приоритетами;	[23]	$\omega_1 = \frac{\lambda_1 v_1^{(2)}}{2(1 - \rho_1)} + \frac{\lambda_2 v_2^{(2)} (1 - \rho_1)^2 + \lambda_1 v_1^{(2)} \rho_2^2}{2(1 - \rho_1)(1 - R)(1 - R + 2\rho_1 \rho_2)};$ $\omega_2 = \frac{\lambda_2 v_2^{(2)}}{2(1 - \rho^2)} + \frac{\lambda_1 v_1^{(2)} (1 - \rho_2)^2 + \lambda_2 v_2^{(2)} \rho_2^2}{2(1 - \rho_2)(1 - R)(1 - R + 2\rho_1 \rho_2)}$ $\omega_k = \frac{\sum_{k=1}^M \rho_k T_k (1 + v_k^2)}{(1 - R_{k-1})(1 - R_k)},$	Аналитическое исследование данной дисциплины для случая и потоков довольно сложно и осуществляется обычно методами имитационного моделирования

№ п/п	Название дисциплины	Литературный источник	Выражение для определения времени ожидания заявки в очереди	Примечание
	обслуживание с от-носительными при-оритетами;	[22]	<p>где $R_{k-1} = \rho_1 + \rho_2 + \dots + \rho_{k-1}$;</p> $R_k = \rho_1 + \rho_2 + \dots + \rho_k; \rho_k = \frac{\lambda_k}{\mu_k} = \lambda_k T_k$	
	обслуживание с от-носительными при-оритетами в режиме пакетной обработ-ки информации;	[22]	$\omega_n = \frac{\lambda_n v_n^{(2)}}{2(1-\rho_n)} + \frac{(1-\rho_n)^2 M_{n-1} + \lambda_n v_n^{(2)} R_{n-1}^2}{2(1-\rho_n)(1-R_n)(1-R_n+2\rho_n R_{n-1})}$ <p>где $M_{n-1} = \sum_{l=1}^{n-1} \lambda_l v_l^{(2)}$; $R_n = \sum_{l=1}^n \lambda_l T_l$;</p> $R_{n-1} = \sum_{l=1}^{n-1} \lambda_l T_l.$	В режиме пакетной обработки с относительными приоритетами заявки каждого приоритета обслуживаются до тех пор, пока очередь заявок данного приоритета не будет исчерпана
			<p>Для заявок с приоритетом $k < n$ выражение в яв-ном виде не получено, поэтому для вычисления ω_k используют рекуррентное соотношение вида</p> $\omega_k = \frac{\lambda_k v_k^{(2)}}{2(1-\rho_k)} + \frac{\Lambda \cdot \tau_k}{\lambda_k^{(2)} \cdot 2(1-\rho_k)},$	
			<p>где τ_k находят по следующей рекуррентной фор-муле:</p> $\tau_k = \frac{1}{b_{kk}} \left(C_k - \sum_{i=k+1}^n b_{ki} \tau_i \right),$	
			<p>где</p> $b_{ki} = \frac{\lambda_i^2}{(1-R_{i-1})^2} \frac{T_i^2}{(1-\rho_i)^2} \left(i = \overline{k+1, n}; k = \overline{1, n} \right);$	

$$b_{kk} = 1 - \left[\frac{\rho_k R_{k-1}}{1 - R_k - \rho_k R_{k-1}} \right]^2;$$

$$C_k = \frac{\lambda_k^2 M_{k-1}}{\Lambda (1 - R_{k-1})^2} +$$

$$+ \frac{\lambda_k^2 v_k^{(2)} R_{k-1}^2}{\Lambda (1 - R_k + \rho_k R_{k-1})^2} + \frac{\lambda_k^2}{\Lambda (1 - R_{k-1})^2} \times$$

$$\times \sum_{i=k-1}^n \frac{\lambda_i v_i^{(2)}}{(1 - \rho_i)^2}; \quad (k = \overline{1, m})$$

[22]

обслуживание с абсолютными приоритетами и дообслуживанием

$$\omega_k = \frac{R_{k-1} T_k}{1 - R_{k-1}} + \frac{\sum_{i=1}^k \lambda_i T_i (1 + v_i^2)}{2 (1 - R_{k-1}) (1 - R_k)}$$

4. Внутрисистемные приоритетные дисциплины обслуживания:

$$\omega_i = \frac{\frac{\omega_0}{1 - R} \sum_{j=i+1}^n \rho_j \left(1 - \frac{b_j}{b_i} \right)}{1 - \sum_{j=1}^i \rho_j \left(1 - \frac{b_i}{b_j} \right)},$$

[9]

относительные динамические приоритеты с линейной возрастающей функцией потерь;

где $b_i \geq 0$ — коэффициент изменения приоритета заявки

$$\omega_0 = \frac{1}{2} \sum_{i=1}^n \lambda_i T_i^2 (1 + v_i^2)$$

Формула выведена в предположении пуассоновости поступающих потоков информации и экспоненциальности распределения времени обслуживания заявок. Необслуженные заявки образуют очередь, выбор из которой осуществляют в соответствии с текущим значением функции приоритета $\beta_i(t)$, где

$$\beta_i(t) = b_i(t - t_{0i}),$$

$$\tau = t - t_{0i},$$

t_{0i} — момент поступления заявки.

№ л/п	Название дисциплины	Литературный источник	Выражение для определения времени ожидания заявки в очереди	Примечание
	абсолютные динамические приоритеты с линейной возрастающей функцией потерь;	[25]	Выражение для ω_k не приводим из-за его громоздкости	Состояние системы характеризуется вектором $x = (x_1, x_2, \dots, x_n),$ где x_i — число заявок i -го потока. Множество таких векторов с целочисленными неотрицательными компонентами образует пространство состояний системы $x = \{x = (x_1, x_2, \dots, x_n)\}$. Пусть это пространство разбито на непересекающиеся множества $x_k (k = \overline{0, n})$, $x = \bigcup_{k=0}^n x_k$, $x_k \cap x_s = \emptyset$. Для каждого разбиения $x = \{x_k, k = \overline{0, n}\}$ определена стратегия управления. Если система в момент принятия решения находится в состоянии x_k , то принимается управление $y = V(x_k) = k$
	ситуационные относительные и абсолютные приоритеты	[26]		

процессора, при котором обеспечиваются минимальные простои. Такое быстродействие мы будем называть *оптимальным*.

Как и ранее, сначала рассмотрим случай, когда ограничения на время выполнения заявок не наложены. Тогда при использовании дисциплины ПП—ПО (первым пришел — первым обслужен) выражение для длительности обслуживания заявок имеет вид

$$\omega = \frac{\sum_{i=1}^M \lambda_i \Theta_i^{(2)}}{2B(B - \sum_{i=1}^M \lambda_i \Theta_i)}, \quad (5.15)$$

где Θ_i и $\Theta_i^{(2)}$ — среднее значение и второй начальный момент трудоемкости при обслуживании заявок типа $i = 1, \dots, M$.

Характер зависимостей $\eta(B)$, $\omega(B)$ и C_B приведен на рис. 5.2. Как явствует из рисунка, функция C_B имеет минимум, которому соответствует оптимальное значение быстродействия процессора B_{opt} .

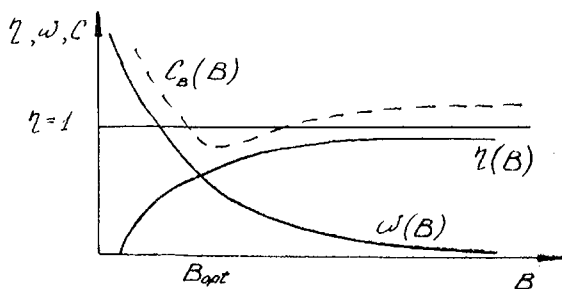


Рис. 5.2. Характеристики АСОИ с неограниченным временем пребывания заявок

Оптимальное быстродействие B_{opt} может быть найдено путем дифференцирования критериальной функции по B с предварительной подстановкой развернутых выражений для $\eta(B)$, т. е.

$$C_B' = \beta_0 \frac{1}{B^2} \sum_{i=1}^M \lambda_i \Theta_i + \sum_{i=1}^M \beta_i \lambda_i \omega_i'(B). \quad (5.16)$$

Подставляя в формулу (5.16) значение $\omega_i'(B)$ для выбранной дисциплины обслуживания, из условия $C_B' = 0$ находим

$$B_{opt} = F(\beta_0, \dots, \beta_M, \lambda_1, \dots, \lambda_M, \Theta_1, \dots, \Theta_M).$$

В качестве примера рассмотрим наиболее простой случай с использованием в АСОИ беспriorитетной дисциплины. Положим для простоты, что $\beta_1 = \beta_2 = \dots = \beta_M = \beta = k \beta_0$, где k — не-

который коэффициент пропорциональности. Тогда выражение (5.16) примет вид

$$C_B' = \beta_0 \frac{1}{B^2} \sum_{i=1}^M \lambda_i \Theta_i + k \beta_0 \Lambda \omega' (B),$$

где $\Lambda = \sum_{i=1}^M \lambda_i$.

Подставляя значение $\omega' (B)$, полученное в результате дифференцирования выражения (5.15), имеем [5]

$$B_{\text{opt}} = \sum_{i=1}^M \lambda_i \Theta_i + \frac{1}{2} \frac{1}{\sum_{i=1}^M \lambda_i \Theta_i} (k \Lambda \sum_{i=1}^M \lambda_i \Theta_i^{(2)} \sqrt{\kappa}), \quad (5.17)$$

где

$$\kappa = [2 (\sum_{i=1}^M \lambda_i \Theta_i)^2 + k \Lambda \sum_{i=1}^M \lambda_i \Theta_i^{(2)}] k \Lambda \sum_{i=1}^M \lambda_i \Theta_i^{(2)}.$$

Если нет ограничений на время обслуживания заявок и $k=0$, то величина B_{opt} совпадает со значением нижней оценки быстродействия, т. е. быстродействие АСОИ в этом случае следует выбирать только из условия максимальной загрузки процессора.

В случае, если при проектировании наложены ограничения на среднее время ожидания заявок ω_i^* , задачу выбора B_{opt} следует решать с учетом этих ограничений, т. е. необходимо решить систему $M + 1$ неравенств

$$\begin{cases} \eta (B) \leq \eta^* \\ \omega_i (B) \leq \omega_i^*, \quad i = \overline{1, M}. \end{cases} \quad (5.18)$$

Решение этой системы неравенств позволит определить область допустимых значений B , если она существует. Возможны два случая:

1. $B_k \leq B \leq B_0$, где B_0, B_k — значения быстродействия процессора АСОИ при η^* и ω_k^* соответственно, причем

$$B_k = \max B_i, \quad (i = \overline{1, M}).$$

В случае, если C_B имеет на $[B_k, B_0]$ минимум, то в качестве оптимального выбирают значение B , соответствующее этому минимуму. Если C_B монотонно убывает на $[B_k, B_0]$, то $B_{\text{opt}} = B_0$ (рис. 5.3,а).

2. $B_0 < B_k$ — при таком соотношении B_0 и B_k задача нахождения B_{opt} неразрешима, и мы вынуждены выбирать значение $B = B_k$, ведущее к значительным простоям процессора АСОИ (рис. 5.3,б).

Для некоторых простейших случаев легко получить искомые соотношения в замкнутой форме. Так при использовании дис-

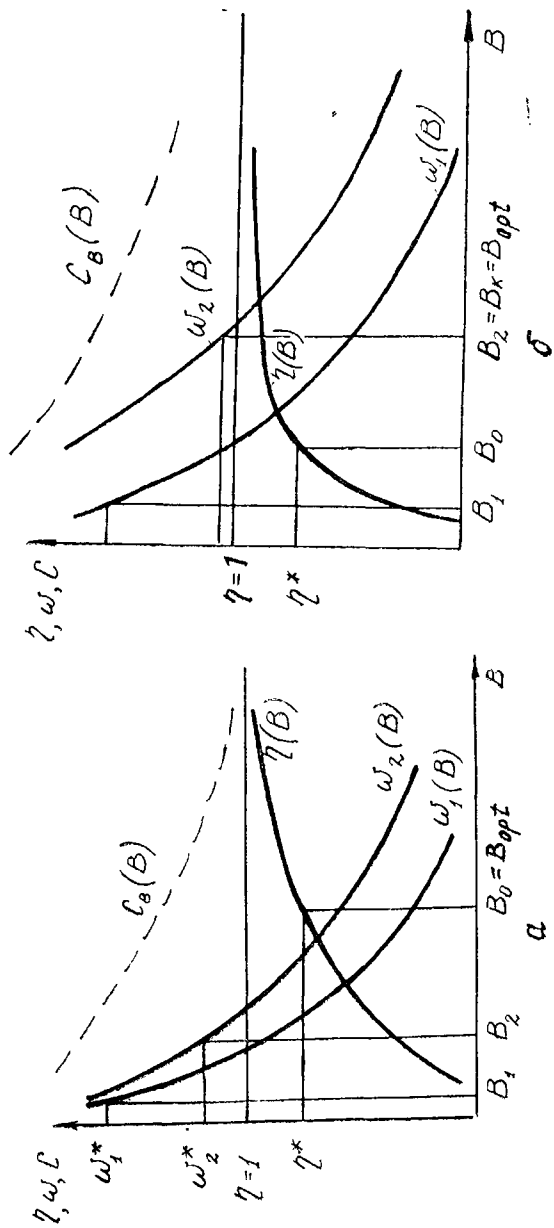


Рис. 5.3. Характеристики системы при: а — $B_k \ll B \ll B_0$; б — $B_0 < B_k$

циплины «первый пришел — первый обслужен» система неравенств примет вид [5]

$$\begin{cases} 1 - \frac{\sum_{i=1}^M \lambda_i \Theta_i}{B} \leq \eta^* \\ \sum_{i=1}^M \lambda_i \Theta_i^{(2)} / [2B (B - \sum_{i=1}^M \lambda_i \Theta_i)] \leq \omega_i^*, \quad i = \overline{1, M}. \end{cases}$$

Нетрудно определить область допустимых значений B [5]:

$$\frac{L}{2} \left[\frac{L^{(2)}}{4} + \frac{L^{(2)}}{2\omega^*} \right]^{1/2} \leq B \leq \frac{1}{1 - \eta^*},$$

где $L = \sum_{i=1}^M \lambda_i \Theta_i$; $L^{(2)} = \sum_{i=1}^M \lambda_i \Theta_i^{(2)}$.

Верхней границе B соответствует зависимость $B = B(\eta^*)$, нижней — $B = B(\omega^*)$.

5.5. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ ВЫБОРА ПАРАМЕТРОВ СИСТЕМЫ

Исходные данные

На входе АСОИ четыре пуассоновских потока заявок с интенсивностями $\lambda_1 = 2000$ 1/с; $\lambda_2 = 200$ 1/с; $\lambda_3 = 15$ 1/с; $\lambda_4 = 30$ 1/с. Заданы следующие ограничения: $\omega_1^* = 10^{-2}$ с; $\omega_2^* = 10^{-3}$ с; $\omega_3^* = 0,5$ с; $\omega_4^* = 1$ с.

Предельное ограничение на коэффициент простоя процессора $\eta^* = 0,18$.

Средние трудоемкости реализации алгоритмов по обслуживанию соответствующих типов заявок: $\Theta_1 = 100$ опер., $\Theta_2 = 200$ опер., $\Theta_3 = 100$ опер., $\Theta_4 = 150$ опер.

Трудоемкости алгоритмов подчинены экспоненциальному закону распределения. Следовательно, параметр Θ полностью описывает вероятностные характеристики времени реализации алгоритмов.

Решение задачи синтеза АСОИ

1. Определение нижней оценки быстродействия процессора: воспользуемся выражением

$$B > 0,5 \sum_{i=1}^M \lambda_i \Theta_i + \left[0,25 \sum_{i=1}^M \lambda_i \Theta_i \left(\sum_{i=1}^M \lambda_i \Theta_i + \right. \right. \\ \left. \left. + 2 \sum_{i=1}^M \lambda_i \Theta_i^{(2)} / \sum_{i=1}^M \lambda_i \Theta_i \omega_i^* \right) \right]^{1,2}.$$

Учтем, что для экспоненциального распределения справедливо соотношение $\Theta^{(2)} = 2\Theta^2$. Подставив исходные данные, получим: $B > 250000$ опер/с, т. е. $B_{\min} = 250000$ опер/с.

2. Выбор дисциплины обслуживания для АСОИ с относительными ограничениями на время пребывания заявок в значительной степени зависит от ограничений, налагаемых на время ожидания заявок. Выполнение этих ограничений в общем случае может быть достигнуто за счет использования смешанных дисциплин обслуживания — рациональным назначением приоритетов потокам заявок.

Вид (абсолютный или относительный) и степень приоритетности каждого потока заявок по отношению к другим потокам можно выбирать любыми способами. Один из них — целенаправленный перебор некоторых вариантов распределения приоритетов с учетом параметров потоков заявок, ограничений на время ожидания заявок и степени изменения качества обслуживания при переходе от одного варианта к другому.

Показателем, определяющим необходимость изменения приоритета некоторого потока служит относительное отклонение $\delta\omega_i$ времени ожидания ω_i , полученного при данном распределении приоритетов, от допустимого времени ожидания ω_j . Если отклонения $\delta\omega$ для одних потоков значительно отличаются от отклонений для других потоков, то необходимо изменить приоритет этих потоков. Распределение приоритетов удовлетворительно, если для всех потоков заявок значения $\delta\omega$ приблизительно одинаковы [5].

Определим характеристики потоков заявок произвольного типа для минимального быстродействия процессора АСОИ B_{\min} .

Среднее значение длительности обслуживания заявок разных типов определяем как

$$T_i = \frac{\Theta_i}{B_{\min}}, \quad i = \overline{1, M}.$$

Подставляя в эту формулу значения заданных трудоемкостей для заявок каждого из четырех типов, получаем средние значения их длительностей обслуживания:

$$T_1 = \frac{100}{25 \cdot 10^4} = 4 \cdot 10^{-4} \text{ с}; \quad T_2 = \frac{200}{25 \cdot 10^4} = 8 \cdot 10^{-4} \text{ с};$$

$$T_3 = \frac{100}{25 \cdot 10^4} = 4 \cdot 10^{-4} \text{ с}; \quad T_4 = \frac{150}{25 \cdot 10^4} = 6 \cdot 10^{-4} \text{ с}.$$

Вторые начальные моменты длительностей обслуживания определяем из выражения $T_i^{(2)} = 2T_i^2$, которое справедливо для случая экспоненциального распределения длительности обслуживания:

$$T_1^{(2)} = 32 \cdot 10^{-8} \text{ с}^2; \quad T_2^{(2)} = 128 \cdot 10^{-8} \text{ с}^2;$$

$$T_3^{(2)} = 32 \cdot 10^{-8} \text{ с}^2; \quad T_4^{(2)} = 72 \cdot 10^{-8} \text{ с}^2.$$

Значения коэффициентов загрузки со стороны заявок i -го типа ($i = 1, 4$) составляют $\rho_1 = 0,8$; $\rho_2 = 0,16$; $\rho_3 = 0,006$; $\rho_4 = 0,018$. Суммируя эти величины, получим общий коэффициент загрузки процессора АСОИ, который будет

$$R = \sum_{i=1}^4 \rho_i = 0,8 + 0,16 + 0,006 + 0,018 = 0,984.$$

Рассчитаем среднее время ожидания заявок в очереди. Для бесприоритетной дисциплины воспользуемся выражением

$$\omega = \frac{\sum_{i=1}^M \lambda_i T_i^{(2)}}{2(1-R)} =$$

$$= \frac{2 \cdot 10^3 \cdot 32 \cdot 10^{-8} + 2 \cdot 10^2 \cdot 128 \cdot 10^{-8} + 15 \cdot 4 \cdot 10^{-4} + 30 \cdot 72 \cdot 10^{-8}}{2(1-0,984)} =$$

$$= 28,8 \cdot 10^{-3} \text{ с}.$$

Полученное время ожидания не удовлетворяет заданным ограничениям, поэтому перейдем к рассмотрению приоритетного обслуживания, воспользуемся следующими приемами [5].

Для уменьшения числа переборов упорядочим потоки, располагая их в порядке возрастания допустимого времени ожидания ω_i^* . В результате этой операции рассматриваемые в примере потоки расположатся следующим образом: 2, 1, 3, 4. В этом случае, если два или более потоков имеют одинаковые ограничения по ω_i^* , то рекомендуется располагать их в порядке возрастания времени обслуживания (T_i).

Далее назначаем приоритет потокам заявок следующим образом: каждому потоку приоритет не ниже, чем потокам, расположенным правее, и не выше, чем потокам, расположенным левее, в указанной последовательности. Получаем $P_1 = 2$; $P_2 = 1$; $P_3 = 3$; $P_4 = 4$, где P_i — приоритет. Считая назначенные приоритеты относительными, рассчитываем среднее время ω_k ожидания заявок в каждом из потоков и относительные отклонения $\delta \omega$:

$$\omega_k = \frac{\sum_{k=1}^M \lambda_k T_k^{(2)}}{2(1-R_{k-1})(1-R_k)},$$

где $\sum_{k=1}^M \lambda_k T_k^{(2)} = 10^{-5} (64 + 25,6 + 0,48 + 2,16) \text{ с} = 92,24 \cdot 10^{-5} \text{ с}$,

$$R_{k-1} = \rho_1 + \dots + \rho_{k-1}$$

и $R_k = \rho_1 + \rho_2 + \dots + \rho_k$ — загрузки, создаваемые потоком заявок 1, ..., $k-1$ и 1, ..., k ; k — степень приоритета заявок.

$$\text{При } k = 1 \quad \omega_2 = \frac{92,24 \cdot 10^{-5}}{2 \cdot 0,84} \text{ с} = 0,54 \cdot 10^{-3} \text{ с};$$

$$\delta \omega_1 = \frac{10^{-3} - 0,49 \cdot 10^{-3}}{0,549 \cdot 10^{-3}} = 0,82;$$

$$\text{при } k = 2 \quad \omega_1 = \frac{92,24 \cdot 10^{-5}}{2 \cdot 0,84 \cdot 0,04} \text{ с} = 1,37 \cdot 10^{-2} \text{ с};$$

$$\delta \omega_2 = \frac{10^{-2} - 1,37 \cdot 10^{-2}}{1,07 \cdot 10^{-2}} = 0,065;$$

$$\text{при } k = 3 \quad \omega_3 = \frac{92,24 \cdot 10^{-5}}{2 \cdot 0,034 \cdot 0,04} \text{ с} = 0,339 \text{ с};$$

$$\delta \omega_3 = \frac{0,5 - 0,339}{0,339} = 0,47;$$

$$\text{при } k = 4 \quad \omega_4 = \frac{92,24 \cdot 10^{-5}}{2 \cdot 0,034 \cdot 0,016} \text{ с} = 0,848 \text{ с};$$

$$\delta \omega_4 = \frac{1 - 0,848}{0,848} = 0,12.$$

Таким образом видно, что только время ожидания заявок первого потока не удовлетворяет ограничениям по ω^* . $\omega_1 = 1,37 \cdot 10^{-2}$ с, а $\omega_1 \leq 10^{-2}$ с ($k = 1$), следовательно, нужно изменить приоритет этого потока. Для этого был произведен перебор большого количества возможных вариантов распределения приоритетов, в том числе и с введением приоритетов абсолютного класса. Но все они дали худшие результаты. Таким образом, приведенный первый вариант считаем наиболее приемлемым вариантом распределения приоритетов.

Выполнение ограничения $\omega_1 \leq 10^{-2}$ с можем достигнуть за счет увеличения быстродействия процессора, оставив найденную дисциплину обслуживания неизменной.

Таким образом, считаем оптимальной дисциплину обслуживания со следующим распределением относительных приоритетов: $P_1 = 2$; $P_2 = 1$; $P_3 = 3$; $P_4 = 4$.

3. Определение оптимального быстродействия процессора: решаем систему неравенств

$$\begin{cases} \eta(B) \leq \eta^* \\ \omega_i(B) \leq \omega_i^* \end{cases}$$

подставляя $\eta = 1 - \frac{1 \cdot \sum_{i=1}^M \lambda_i \theta_i}{B}$ и ω_i , выраженные через B , с за

менной $T_i = \Theta_i / B$ для принятой дисциплины обслуживания с относительно приоритетами. Получим следующую систему неравенств:

$$1 - \frac{246000}{B} \leq \eta^* = 0,18;$$

$$B \geq 26,6 \cdot 10^4 \text{ опер/с } (i = 1);$$

$$B \geq 26 \cdot 10^4 \text{ опер/с } (i = 2);$$

$$B \geq 25,1 \cdot 10^4 \text{ опер/с } (i = 4).$$

В результате имеем область допустимых значений для B :
 $266\,000 \text{ опер/с} \leq B \leq 300\,000 \text{ опер/с}.$

Оптимальным будет то значение B , которому соответствует наименьшее значение функции C_B . На интервале $26,6 \cdot 10^4 \text{ опер/с} \leq B \leq 30 \cdot 10^4 \text{ опер/с}$ функция C_B монотонно возрастает. Следовательно, ее наименьшее значение соответствует значению $B = 266\,000 \text{ опер/с}$. Итак, $B_{\text{opt}} = 266\,000 \text{ опер/с}$. Найдем время ожидания заявок и коэффициент простоя процессора при

$$k = 2: \quad \omega_2 = 0,81 \cdot 10^{-3} \text{ с} < 10^{-3} \text{ с} = \omega_2^*;$$

$$k = 1: \quad \omega_1 < 0,82 \cdot 10^{-2} \text{ с} < 10^{-2} \text{ с} = \omega_1^*;$$

$$k = 3: \quad \omega_3 = 0,075 \text{ с} < 0,5 \text{ с};$$

$$k = 4: \quad \omega_4 = 0,096 \text{ с} < 1 \text{ с};$$

$$\eta = 0,075 < 0,18 = \eta^*.$$

Итак, в результате решения задачи синтеза АСОИ:

выбрана дисциплина обслуживания с относительными приоритетами и следующим распределением их по потокам: $P_1 = 2$, $P_2 = 1$; $P_3 = 3$; $P_4 = 4$;

найден оптимальное быстродействие процессора $266\,000 \text{ опер/с}$, удовлетворяющее ограничениям на время ожидания заявок различных типов (потоков), и коэффициент простоя процессора.

Глава 6

АНАЛИТИЧЕСКОЕ МОДЕЛИРОВАНИЕ АСОИ С ИСПОЛЬЗОВАНИЕМ АППАРАТА СТОХАСТИЧЕСКИХ СЕТЕЙ МАССОВОГО ОБСЛУЖИВАНИЯ

6.1. ПРЕДСТАВЛЕНИЕ МОДЕЛИ АСОИ В ВИДЕ СТОХАСТИЧЕСКОЙ СЕТИ

Описанные задачи анализа и синтеза систем оперативной обработки информации и разделения времени могут быть рассмотрены с единых позиций в рамках так называемого *сетевого* подхода. В этом случае модель соответствующей АСОИ представляют в виде стохастической сети (замкнутой или разомкнутой), после чего в соответствии с известными из теории стохастических сетей алгоритмами и формулами определяют искомые характеристики и соотношения, а затем проводят их оптимизацию.

Учитывая большую универсальность сетевого подхода при моделировании АСОИ, целесообразнее бы иметь готовый пакет программ, пригодный для моделирования широкого класса систем обработки информации. В данной главе приведено описание методических основ, принципов построения и порядка работы с такого рода пакетом программ «Сеть», позволяющим моделировать замкнутые и разомкнутые стохастические сети.

Для понимания излагаемого в дальнейшем материала необходимо привести краткие сведения и сводку основных формул из теории стохастических сетей [5].

Под *стохастической сетью* обычно понимают совокупность одно- или многоканальных СМО, в которых требования перемещаются из одной СМО в другую случайным образом до тех пор, пока не покинут систему в какой-либо точке. Марковские стохастические сети образуют класс стохастических сетей, для которых принята гипотеза о том, что входящие в сеть потоки пуассоновские, а длительности обслуживания запросов в различных узла (подсистемах) сети распределены по экспоненциальному закону.

Рассмотрим произвольную сеть с n узлами, в i -м узле которой находится СМО S_i с одной очередью и k_i приборами. Время обслуживания в каждом из приборов распределено по показательному закону со средним значением $T_i = 1/\mu_i$. В i -й узел поступает извне (по отношению к сети) последовательность требований из независимого источника с интенсивностью λ_{0i} тре-

бований в секунду. После обслуживания в i -м узле требование переходит в следующий узел j с вероятностью P_{ij} . Таким образом, это требование поступает в j -й узел изнутри (по отношению к сети). Вместе с тем, уходя из j -го узла, требование может покинуть сеть с вероятностью $1 - \sum_{j=1}^n P_{ij}$. Определим полную интенсивность потока требований, поступающих в i -й узел как λ_i требований в секунду; в среднем она включает в себя как внутренние, так и внешние требования. Система уравнений для определения интенсивностей имеет вид

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^n \lambda_j P_{ji}, \quad i = 1, n. \quad (6.1)$$

Узлы рассматриваемой сети в значительной мере независимы, что видно из полученного Джексоном [11] выражения для совместного распределения вероятностей нахождения M_1 требований в первом узле, M_2 требований во втором и т. д.:

$$P(M_1, M_2, \dots, M_n) = P(M_1) \cdot P(M_2) \dots P(M_n).$$

В этом равенстве о независимости свидетельствует произведение распределений в правой части. Воспользуемся этой независимостью, в частности, каждый множитель $P(M)$ в приведенном равенстве является решением для изолированной СМО типа $M/M/k$, работающей самостоятельно при интенсивности поступления требований λ , а $P(M_i)$ определяется равенством

$$P(M_i) = \begin{cases} P_0 \frac{(k\rho)^{M_i}}{M_i!}; & M_i \leq k, \\ P_0 \frac{\rho^{M_i} k^k}{k!}; & M_i \geq k, \end{cases}$$

где

$$P_0 = \left[\sum_{M=0}^{k-1} \frac{(k\rho)^M}{M!} + \frac{(k\rho)^k}{k!(1-\rho)} \right]^{-1}.$$

Такие стохастические сети называются *разомкнутыми*.

Еще один класс марковских стохастических сетей включает сети, узлы которых требования не могут ни покидать, ни поступать в них извне. В частности, предположим, что в сети, подобной описанной выше, находится M требований и что они перемещаются из узла в узел, но уйти из сети не могут; иначе говоря, для всех j справедливо равенство $\sum_{j=1}^n P_{ij} = 1$. Совместное распределение вероятностей наличия определенного числа требований в различных узлах таких *замкнутых сетей* выражается следующим образом:

$$P(M_1, M_2, \dots, M_n) = \frac{1}{G(M)} \prod_{i=1}^n \rho_i^{M_i} ; \quad \rho_i = \frac{\lambda_i}{\mu_i} ;$$

где множество чисел λ_i должно удовлетворять линейным уравнениям, аналогичным уравнениям (6.1) при $\lambda_{0i} = 0$. Здесь $G(M)$ — константа, обеспечивающая условие нормировки и определяемая как

$$G(M) = \sum_{k \in A} \prod_{i=1}^n \rho_i^{M_i},$$

причем $M = \{M_1, M_2, \dots, M_n\}$; A — множество векторов M , для которых $M_i \geq 0$, а $\sum_{i=1}^n M_i = M$.

Параметры марковских стохастических сетей и определение их характеристик

Стохастическая сеть определяется совокупностью следующих параметров:

числом n систем массового обслуживания S_1, S_2, \dots, S_n , образующих сеть;

числом каналов (обслуживающих приборов) k_1, k_2, \dots, k_n , входящих в состав систем S_1, S_2, \dots, S_n ;

матрицей вероятностей передач $\|P\| = \{P_{ij}\}$, где P_{ij} — вероятность того, что заявка, покидающая систему S_i , поступит в систему S_j ($i, j = \overline{0, n}$);

числом M заявок, циркулирующих в замкнутой системе, или интенсивностью источника заявок λ_0 в разомкнутой сети;

средними длительностями обслуживания заявок T_1, T_2, \dots, T_n . Без дальнейших пояснений, которые легко можно отыскать в литературе [5], приведем характеристики разомкнутых и замкнутых стохастических сетей в виде сводки формул (табл. 6.1), используемых для расчета основных характеристик сетей.

Используя понятие разомкнутых и замкнутых стохастических сетей, можно легко строить модели отдельных подсистем АСОИ, а затем «собирать» модель АСОИ в целом, как из кубиков.

В качестве отдельных подсистем АСОИ можно выделить следующие компоненты: «процессор — оперативная память», «мультиплексорный канал — периферийные устройства», «селекторный канал — ВЗУ (или любая другая периферия)». Эти комбинации практически исчерпывают основные структурные блоки, из которых можно строить сетевую модель любой АСОИ (табл. 6.2).

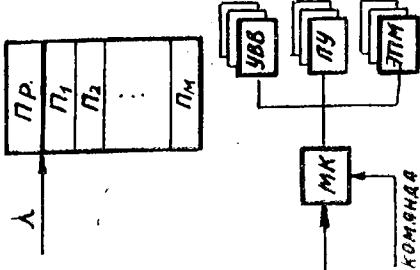
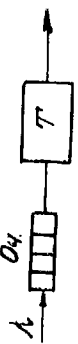
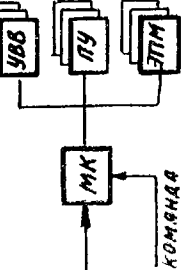
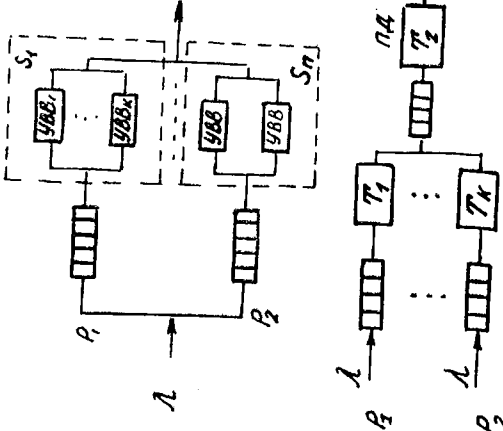
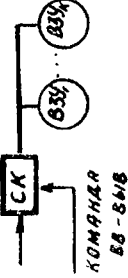
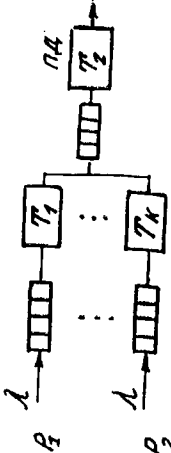
Таблица 6.1

Основные характеристики замкнутых и разомкнутых стохастических сетей

Характеристика	Разомкнутые сети	Замкнутые сети
Вероятности состояний	$P(M_1, M_2, \dots, M_n) = \frac{R_j(M_j) \beta_j^{M_j}}{\prod_{j=1}^n k_j - 1} + \frac{\beta_j^{k_j}}{k_j! (1 - \beta_j/k_j)}$ <p style="text-align: center;">где</p> $\beta_j = \lambda_j T_j = \begin{cases} \rho_j & \text{при } k_j = 1 \\ k_j & \text{при } k_j > 1, \end{cases}$ $R_j(M_j) = \begin{cases} 1/M_j! & \text{при } M_j \leq k_j \\ 1/k_j! k_j^{M_j - k_j} & \text{при } M_j > k_j \end{cases}$	$P(M_1, M_2, \dots, M_n) = \frac{\prod_{j=1}^n R_j(M_j) (\alpha_j T_j)^{M_j}}{A(M, n) \prod_{j=1}^n R_j(M_j) (\alpha_j T_j)^{M_j}}$
Условие существования стационарного режима	$\lambda_0 < \min \left\{ \frac{k_j}{\alpha_j T_j} \right\}; \quad j = \overline{1, n}$	
Характеристики систем в сети: средняя длина очереди в системе	$l_j = \frac{\beta_j^{k_j+1}}{k_j! k_j (1 - \beta_j/k_j)^2} \cdot T_{0j}$	$l_j = \sum_{r=k_j+1}^M (r - k_j) P(M_j = r)$
среднее число заявок в системе	$m_j = l_j + \beta_j$	$m_j = \sum_{r=0}^m r P(M_j = r)$
среднее время ожидания заявки в очереди	$\omega_j = \frac{l_j}{\lambda_j}$	$\omega_j = \frac{l_j}{\lambda_j}$

Окончание табл. 6.1

Характеристика	Разомкнутые сети	Замкнутые сети
среднее время пребывания заявки в системе	$u_j = \omega_j + T_j$	$u_j = \frac{m_j}{\lambda_j}$
Характеристики сети: среднее число заявок, ожидающих обслуживания	$l = \sum_{j=1}^n l_j$	
среднее число заявок, пребывающих в сети	$m = \sum_{j=1}^n m_j$	
среднее время ожидания	$\omega = \sum_{j=1}^n \alpha_j \omega_j$	
среднее время пребывания заявок в сети	$u = \sum_{j=1}^n \alpha_j u_j$	

Структурный блок	Функциональная модель	Стохастическая модель
Процессор — ОЗУ		
Мультиплексный канал — периферийные устройства		
Селекторный канал — ВЗУ		

6.2. ПРИНЦИПЫ ПОСТРОЕНИЯ ПАКЕТА ПРИКЛАДНЫХ ПРОГРАММ «СЕТЬ» И ПОРЯДОК РАБОТЫ С НИМ

Назначение и состав пакета

Пакет программ «Сеть» [10] позволяет проводить аналитическое моделирование АСОИ и их элементов на основе аппарата стохастических сетей массового обслуживания. В результате моделирования могут быть получены все основные характеристики сетей (см. табл. 6.1).

Пакет программ построен по модульному принципу и состоит из следующих модулей или блоков:

- модуль ввода исходных данных;
- модуль определения коэффициентов передачи систем сети;
- блок проверки наличия в системе стационарного режима и блок проверки возможности проведения моделирования (для замкнутых сетей);

- модуль определения вероятностей состояний;
- модуль определения характеристик систем в сети (для замкнутых и разомкнутых сетей) и характеристик сети (для разомкнутых сетей);

- модуль толерантного преобразования;
- модуль определения характеристик толерантной разомкнутой сети;

- блок вспомогательных алгоритмов «А» и «Б»;

- модуль интерполяции результатов;

- модуль вывода на печать результатов моделирования.

Подробная блок-схема пакета программ «Сеть» приведена на рис. 6.1. Остановимся подробнее на порядке функционирования пакета программ.

После того, как исходные данные введены, в модуле ввода происходит анализ требований пользователя (т.е. определяется тип моделируемой сети и ее структура). При моделировании разомкнутых сетей после ввода исходных данных и определения коэффициентов передачи управление передается на блок расчета характеристик, если в сети существует стационарный режим.

При моделировании замкнутых сетей и при условии, что структура сети отвечает ограничительному требованию на количество каналов в системе*, производят соответствующие расчеты и определяют характеристики замкнутой сети. Если замкнутая сеть не удовлетворяет вышеуказанному ограничительному требованию, то управление передается на модуль толерантных пре-

* В рассматриваемой версии пакета моделирование замкнутых сетей возможно лишь в случае, когда все системы одноканальные.

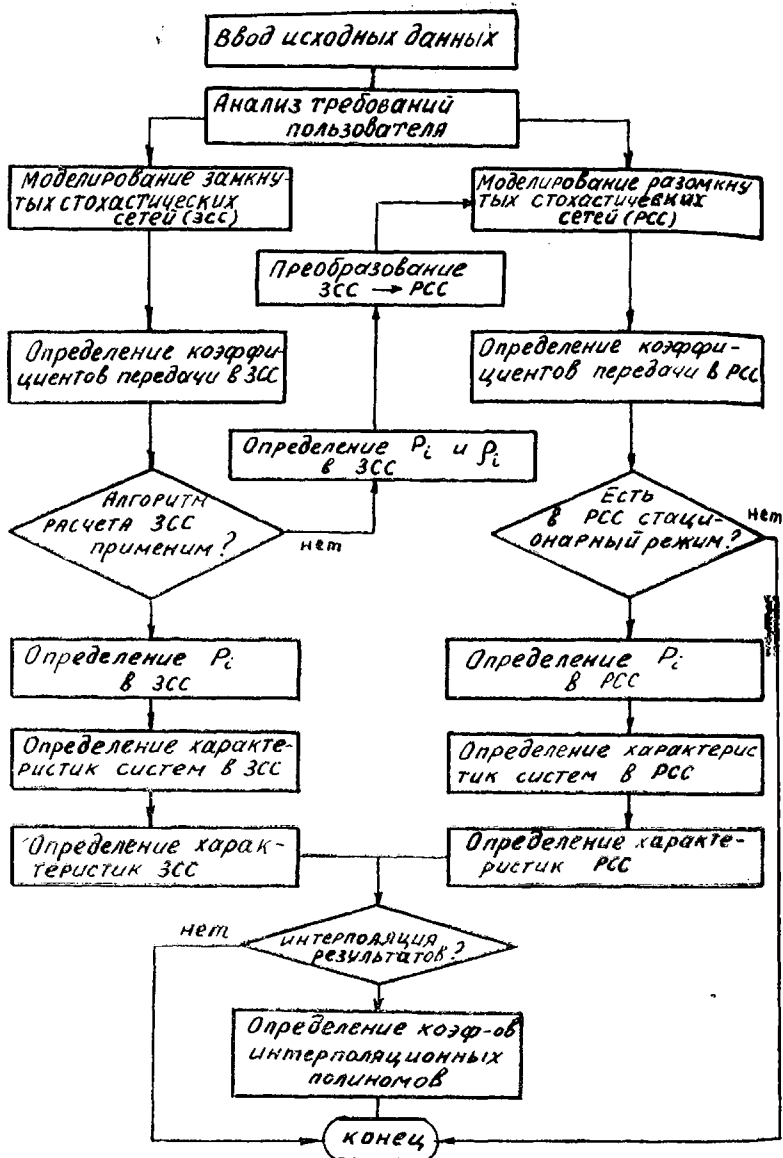


Рис. 6.1. Функциональная схема пакета программ «Сеть»

образований, и замкнутая сеть преобразуется в эквивалентную ей разомкнутую сеть, для которой вычисляют все характеристики.

После моделирования сети, если это необходимо, проводят интерполяцию результатов, в результате чего получают аналитические выражения зависимости одного параметра от другого.

Таким образом, пакет программ «Сеть» позволяет моделировать разомкнутые и замкнутые стохастические сети любой конфигурации и определять все необходимые характеристики.

Порядок работы с пакетом

Для использования пакета программ «Сеть» необходимо ввести числовой материал для соответствующего варианта расчета. Числовой материал должен быть представлен в стандартизованном виде (табл. 6.3).

Таблица 6.3

Вводимый числовой материал

Вводимая величина	Идентификатор		Формат	
	РСС	ЗСС	РСС	ЗСС
Тип моделируемой сети	I7		I2	
Количество систем в сети	NS	NSZ	I2	
Число состояний, вероятность которых надо подсчитать	NV	NVZ	I2	
Количество шагов интерполяции	B1		I2	
Стохастическая матрица вероятностей передачи	SMR	SMZ	9 F 4.2	
Число каналов, входящих в системы	KS	KSZ	8 I 2	
Матрица, задающая состояния, которые необходимо подсчитать	VSR	VSZ	8 I 2	
Интенсивность обслуживания в каждом канале	MU	MUZ	8 F 7.3	
Матрица, задающая коэффициент изменения параметра для интерполяции	ADR		8 F 7.3	
Матрица, задающая величину шага и диапазон изменения параметра при интерполяции:	ANT		3 F 10.5	
а) интенсивность входного источника заявок (для РСС)	V		F 7.3	
б) количество заявок в сети (для ЗСС)		MZ	I3	

Ввести в программу операторы, описывающие необходимые границы массивов, используемых в программе.

Ниже дадим некоторые пояснения к табл. 6.3.

1. Тип моделируемой сети задают вводом либо цифры «1», которая соответствует разомкнутой, либо цифры «2», которая соответствует замкнутой стохастической сети.

2. Стохастическую матрицу вероятности передач (идентификатор SMR (SMZ)) строят следующим образом: элемент матрицы $SMR(I, J)$ равен вероятности перехода заявки из I -го устройства в J -е.

3. Масштаб интерполяции вводят для того, чтобы обеспечить различную скорость изменения параметров для каждой из систем сети при интерполяции (идентификатор ADR).

В данном пакете программ использована линейная зависимость

$$PZ = B3 + (A4 - 1) B2 ADR(I, J),$$

где PZ — последующее значение параметров;

$A4$ — текущий шаг интерполяции;

$B2$ и $B3$ описаны ниже.

Если пользователю необходимо реализовать другой вид зависимости, то следует заменить в программе перфокарты операторами 302 и 311 на перфокарты с операторами, реализующими требуемую зависимость.

4. Диапазон и шаг интерполяции задают введением специальной матрицы, обозначаемой идентификатором ANT . В общем случае можно задать 12 интерполяционных полиномов.

При моделировании замкнутых стохастических сетей в случае, когда хотя бы одно значение массива KSZ не равно 1, после расчета загрузки систем сети и вероятности состояний сети происходит автоматическое толерантное преобразование замкнутой сети в разомкнутую. После такого преобразования моделируют уже разомкнутую сеть и для нее получают все необходимые характеристики. В этом случае интерполяцию не проводят.

6.3. ПРИМЕР ИСПОЛЬЗОВАНИЯ ПАКЕТА ПРОГРАММ «СЕТЬ» ДЛЯ АНАЛИЗА РАБОТЫ ИЗМЕРИТЕЛЬНОГО ПУНКТА

Рассмотрим пример использования описанного пакета программ для определения и анализа характеристик измерительного пункта (рис. 6.2), где происходит прием и первичная обработка телеметрической информации при испытаниях ЛА.

Информация, поступающая с борта ЛА, после приема приходит в систему сбора телеметрической информации, которая

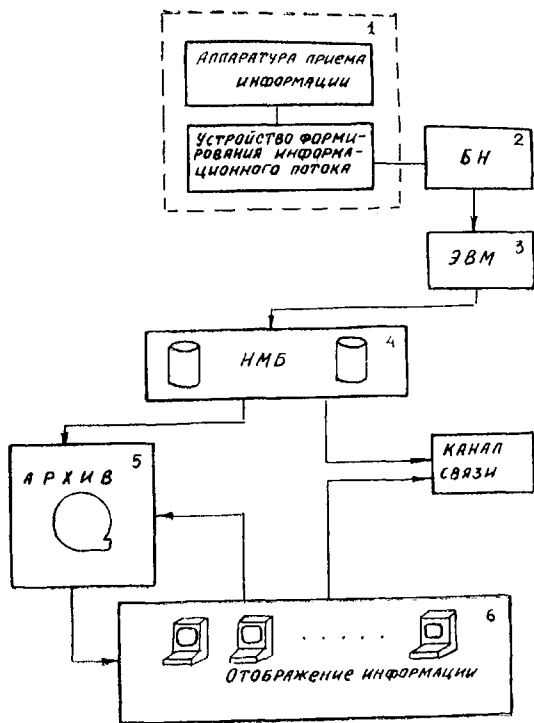


Рис. 6.2. Структурная схема измерительного пункта

формирует поток информации для записи в буферный накопитель (БН). Запись в БН производится со скоростью 1 блок за 1,5 мс, причем один блок содержит 15 измерений. Отметим, что здесь и далее все приводимые технические характеристики являются вымышленными и приводятся лишь для иллюстрации. С БН информация поступает на оперативную обработку в ЭВМ, способную обработать один блок измерений за 250 мс. После обработки информация записывается на магнитный барабан со скоростью 200 измерений за 10 мс. В дальнейшем в зависимости от результатов обработки она отправляется по каналу связи для последующей обработки в другую ЭВМ или в архив, откуда поступает на экраны операторов. Операторы проводят анализ этой информации. В зависимости от результатов оператор может либо затребовать дополнительную информацию, либо отослать поступившую к нему порцию информации по каналу связи для более полной обработки. Будем считать, что в распоряжении операторов имеется 16 экранов (дисплеев) и на

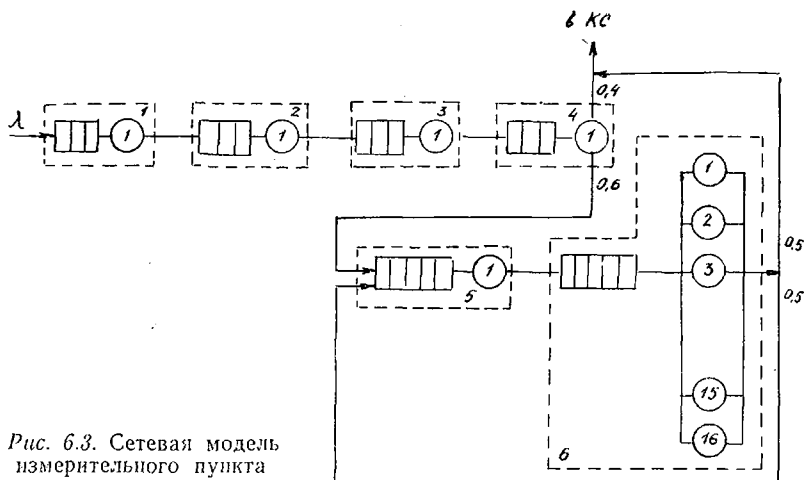


Рис. 6.3. Сетевая модель измерительного пункта

каждый дисплей может быть представлено 16 измерений. Частота смены информации на каждом дисплее составляет один кадр за 7 с. В ходе анализа функционирования описанной системы необходимо получить характеристики оперативности обработки информации и качества работы отдельных подсистем измерительного пункта. Для этого необходимо, во-первых, представить описанную систему в виде стохастической сети, используя таблицу типовых модулей структурных блоков (см. табл. 6.2) и, во-вторых, заполнить табл. 6.3. На приведенной сетевой модели информационно-измерительного пункта (рис. 6.3) нумерация элементов соответствует нумерации подсистем, представленных на рис. 6.2. Интенсивности обслуживания, определенные из условий функционирования системы, сведены в табл. 6.4.

Таблица 6.4

Интенсивности обслуживания систем

Номер системы	Значения интенсивности, 1/с
1	$\approx 10^4$
2	10^4
3	60
4	$2 \cdot 10^3$
5	$\approx 10^4$
6	2,28

потока информации с шагом 6.

Переходные вероятности определяются опытным путем. Интенсивность входного потока информации изменялась в пределах 18..36 заявок/с. В результате эксперимента были получены значения параметров сети при заданных исходных данных. В приведенных ниже табл. 6.5—6.9 все значения характеристик даны в порядке нумерации отдельных систем (см. рис. 6.3).

Интенсивность входного потока информации изменялась в пределах 18..36 заявок/с.

В результате эксперимента были получены значения параметров сети при заданных исходных данных. В приведенных ниже табл. 6.5—6.9 все значения характеристик даны в порядке нумерации отдельных систем (см. рис. 6.3).

Таблица 6.5

Средняя длина очереди в системе

λ , заяв/с	Номер системы					
	1	2	3	4	5	6
18	0,0005	0,2580	0,0000	0,0000	0,0000	56,7431
24	0,0006	0,2667	0,0000	0,0001	0,0000	58,9381
30	0,0006	0,2802	0,0001	0,0001	0,0001	59,8760
36	0,0007	0,2896	0,0003	0,0002	0,0001	61,9810

Таблица 6.6

Количество заявок в системе

λ , заяв/с	Номер системы					
	1	2	3	4	5	6
18	0,0194	0,6331	0,0015	0,0014	0,0012	70,9851
24	0,0246	0,6667	0,0024	0,0121	0,0029	71,5697
30	0,0293	0,6974	0,0031	0,0129	0,0036	72,0031
36	0,0315	0,7081	0,0038	0,0137	0,0041	72,9810

Таблица 6.7

Время ожидания в очереди

λ , заяв/с	Номер системы					
	1	2	3	4	5	6
18	0,0000	0,0105	0,0000	0,0000	0,0000	1,9504
24	0,0000	0,0111	0,0000	0,0000	0,0000	2,0465
30	0,0000	0,0116	0,0000	0,0000	0,0000	2,0713
36	0,0000	0,0118	0,0000	0,0000	0,0000	2,0898

Т а б л и ц а 6.8

Время пребывания заявок в системе

λ , заяв/с	Номер системы					
	1	2	3	4	5	6
18	0,0010	0,0278	0,0010	0,0005	0,0001	2,4003
24	0,0010	0,0278	0,0010	0,0005	0,0001	2,4851
30	0,0010	0,0279	0,0010	0,0005	0,0001	2,5164
36	0,0010	0,0279	0,0010	0,0005	0,0001	2,5498

В табл. 6.9 приведены характеристики сети в целом.

Т а б л и ц а 6.9

Искомые характеристики измерительного пункта

Характеристика	Интенсивность входного потока, заяв/с			
	18	24	30	36
Число заявок в очередях	57,0016	59,2055	60,1565	62,2719
Число заявок в сети	71,6417	72,2784	72,7494	73,7422
Время ожидания, с	2,4114	2,4669	2,5003	2,5287
Время пребывания заявки в сети, с	2,9431	3,0016	3,0215	3,0670

Ограниченность проведенного эксперимента не позволяет сделать достаточно полных выводов о качестве функционирования измерительного пункта. Однако даже из приведенных данных видно, что «узким» местом исследуемой системы являются экранные пульта операторов, вносящие наиболее существенную задержку в процесс обработки информации. Кроме того, можно оценить необходимые объемы буферных накопителей, которые должны быть у экранных пультов и, наконец, можно отметить, что параметры обслуживания мало меняются при изменении входного потока в заданном диапазоне.

7.1. ВВЕДЕНИЕ В ПРОБЛЕМУ

После того как определены основные параметры функционирования АСОИ, обычно приступают к выбору конкретных технических средств, на которых она будет реализована. Уже сегодня отечественная электронная промышленность предлагает достаточно широкий спектр средств вычислительной техники, например, семейства универсальных ЭВМ—ЕС ЭВМ, СМ ЭВМ. Большинство моделей этих семейств имеют общие периферийные устройства и могут комплексоваться в многомашинные комплексы. Поэтому весьма актуальной является задача выбора оптимального состава технических средств для АСОИ. Ниже будут рассмотрены два варианта решения этой задачи. В первом случае мы продемонстрируем решение задачи выбора комплектации АСОИ, обеспечивающее минимальное (или заданное) время обслуживания, при выполнении ряда ограничений. Второй вариант предусматривает одновременное решение двух проблем: оптимальной комплектации и распределения информационного потока по устройствам проектируемой АСОИ.

Ранее было показано, что для широкого класса АСОИ, функционирующих в режиме реального времени, в качестве критериальной функции выбирают некоторую величину, зависящую от среднего времени полной обработки поступающего в АСОИ запроса (заявки). В терминах теории массового обслуживания в такой роли может выступать величина, называемая *средним временем ответа*, которую мы и выберем критериальной функцией при решении поставленной нами задачи.

7.2. РЕШЕНИЕ ЗАДАЧИ ВЫБОРА
ПО КРИТЕРИЮ МИНИМУМА СРЕДНЕГО ВРЕМЕНИ ОТВЕТА

Представим модель рассматриваемой АСОИ в виде стохастической сети массового обслуживания. Для случая марковской стохастической сети и дисциплины «первым пришел — первым обслужен» среднее время ответа (пребывания заявки в сети) определяют следующим выражением:

$$U_i = \sum_{i=1}^n \frac{\alpha_i}{\mu_i} \left[\frac{k_i^{k_i} \rho_i^{k_i}}{k_i! (1 - \rho_i)^2} P_{0i} + k_i \right];$$

$$P_{0i} = \left[\sum_j^{k_i-1} \frac{(k_i \rho_i)^j}{j!} + \frac{(k_i \rho_i)^{k_i}}{k_i! (1 - \rho_i)} \right]^{-1},$$

- где P_{0i} — вероятность отсутствия заявок в S_i -й СМО;
 n — общее количество СМО в рассматриваемой сети;
 α — коэффициент передачи, равный $\alpha_i = -\lambda_i/\lambda_0$;
 $\mu_i = \mu_i(x_j^{(i)})$ — интенсивность обслуживания заявок S_i -й СМО при использовании в ее составе устройства x_j -го типа;
 k_i — количество каналов обслуживания в S_i -й СМО;
 ρ_i — коэффициент загрузки S_i -й СМО.

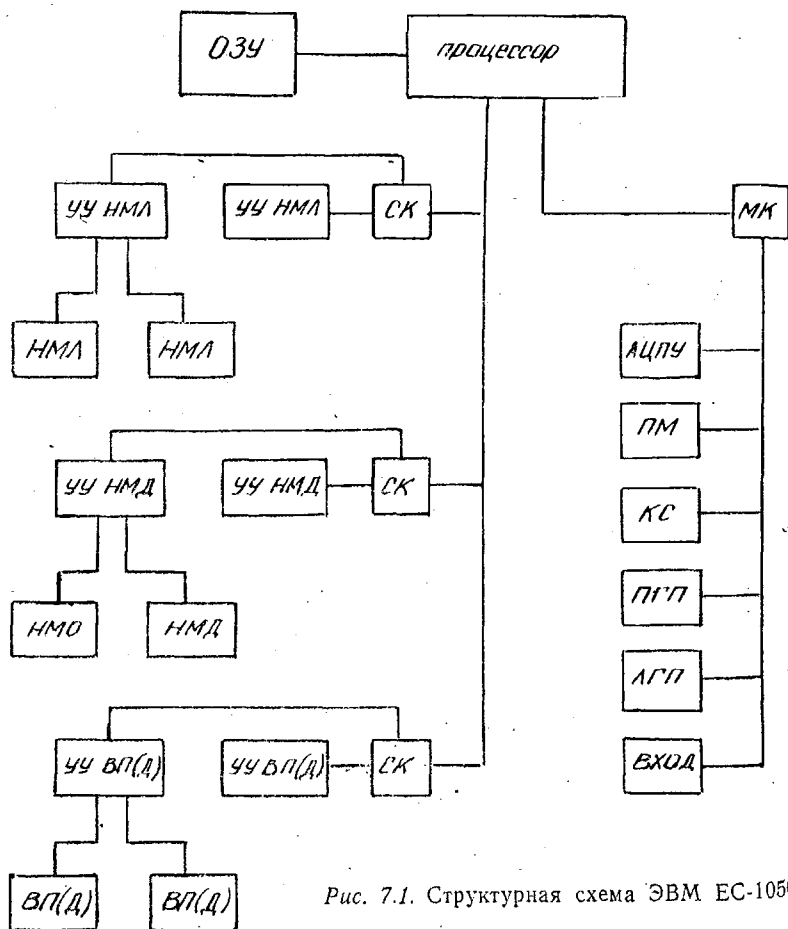


Рис. 7.1. Структурная схема ЭВМ ЕС-1050

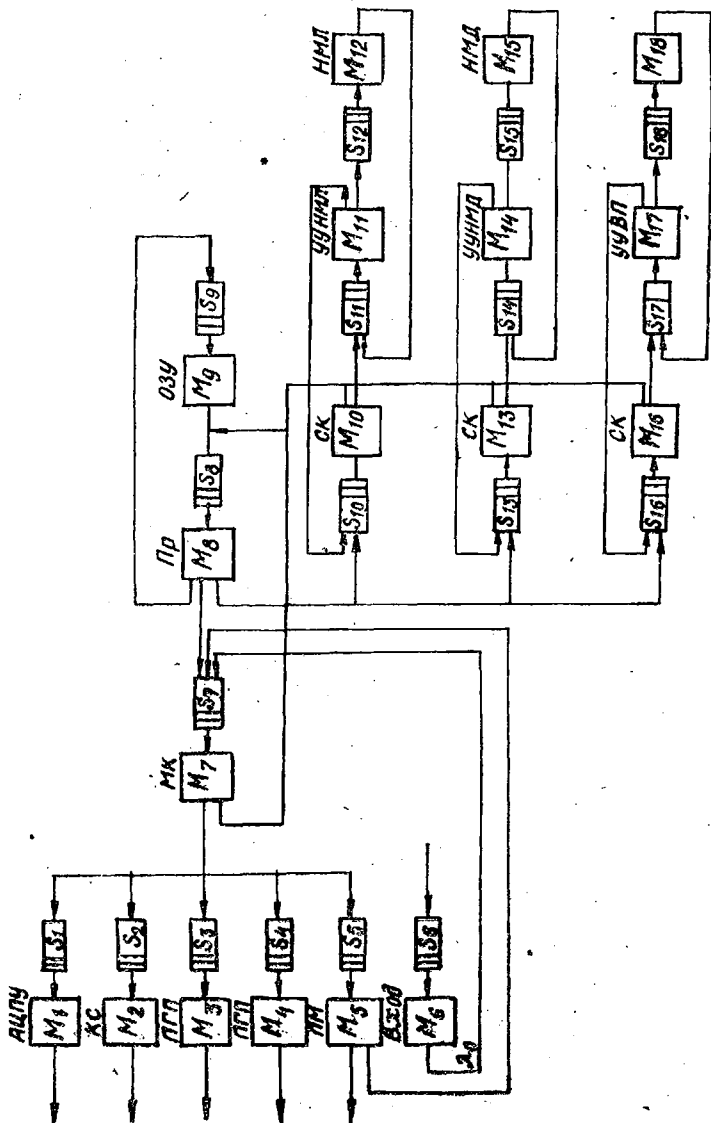


Рис. 7.2. Стохастическая сетевая модель АСОИ

Расчет величины времени пребывания U легко может быть произведен для любой структуры сети с помощью описанного в настоящей работе пакета программ «Сеть».

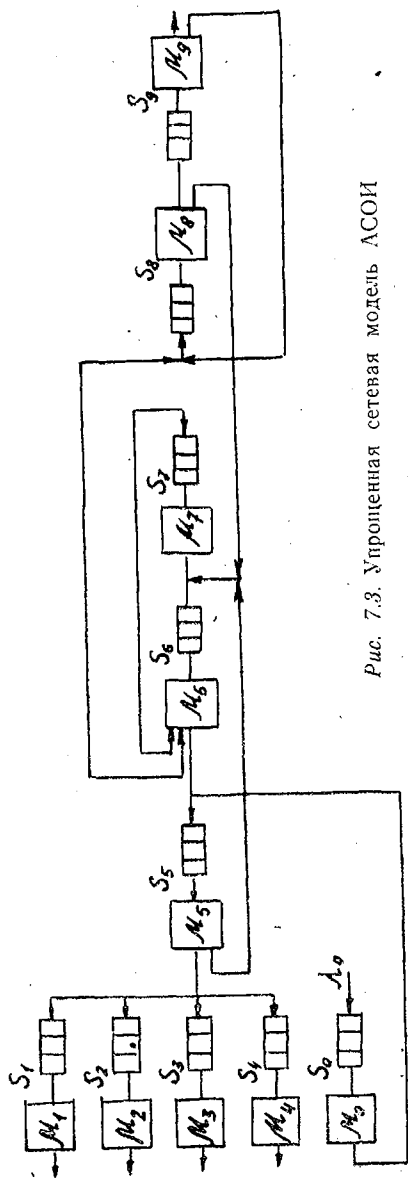


Рис. 7.3. Упрощенная сетевая модель АСОИ

Рассмотрим, как будет выглядеть стохастическая сетевая модель АСОИ, построенной на базе ЭВМ ЕС-1050 (рис. 7.1). Как показано выше, все физические устройства, используемые в АСОИ, могут моделироваться с помощью СМО определенных типов. Объединяя эти модели согласно структурной схеме (см. рис. 7.1), получим модель АСОИ в виде стохастической сети (рис. 7.2). Данная сеть характеризуется, с одной стороны, структурой, т. е. конкретным составом аппаратуры с конкретными связями внутри него, а с другой — свойствами самих СМО.

В дальнейшем, не нарушая общности предлагаемого метода, несколько упростим полученную сеть. Во-первых, будем считать, что при обработке результатов испытаний в режиме реального времени все необходимые программы хранятся в ОЗУ, и обмена с внешними ЗУ не происходит. Во-вторых, объединим устройство управления внешней памятью и саму внешнюю память в одну СМО. В-третьих, исключим из рассмотрения пишущую машинку (ЭПМ), так как она используется оператором ЭВМ и при обработке информации не выполняет непосредственных технологических функций. В результате получим сеть, представленную на рис. 7.3, где

S_1 — АЦПУ, S_2 — канал связи (КС), S_3 — планшетный графопостроитель (ПГП), S_4 — ленточный графопостроитель (ЛПГ), S_0 — вход, S_5 — мультиплексорный канал (МК), S_6 — процессор (Пр), S_7 — ОЗУ, S_8 — селекторный канал (СК), S_9 — дисплей (ВП).

Каждая из девяти СМО, входящих в состав сетевой модели рассматриваемой АСОИ, может быть реализована на базе некоторого стандартного устройства вычислительной техники или набора таких устройств. В связи с этим введем множества ε_l ($l = 1, 9$), соответствующие множествам типов технических средств, используя следующую символику: $\varepsilon_1 = \{e_i^{(1)}\}$ — множество типов АЦПУ, $i = \overline{1, n_1}$, где n_1 показывает максимальное число типов АЦПУ, из которых можно производить выбор; $\varepsilon_2 = \{e_i^{(2)}\}$ — множество типов каналов связи $i = \overline{1, n_2}$ и т. д. Таким образом, будем иметь девять множеств $\varepsilon_1, \varepsilon_2, \varepsilon_3, \dots, \varepsilon_9$. Условия сопрягаемости элементов этих множеств зададим с помощью бинарных отношений вида

$$Q_1 \subseteq \varepsilon_2 \times \varepsilon_5; \quad Q_1 = [q_{ij}^{(1)}]_{|\varepsilon_1| \times |\varepsilon_5|}, \quad i = \overline{1, n_1}; \quad j = \overline{1, n_5}$$

$$Q_2 \subseteq \varepsilon_2 \times \varepsilon_5; \quad Q_3 \subseteq \varepsilon_3 \times \varepsilon_5; \quad Q_4 \subseteq \varepsilon_4 \times \varepsilon_5; \quad Q_5 \subseteq \varepsilon_5 \times \varepsilon_6;$$

$$Q_7 \subseteq \varepsilon_7 \times \varepsilon_6; \quad Q_8 \subseteq \varepsilon_8 \times \varepsilon_6; \quad Q_9 \subseteq \varepsilon_9 \times \varepsilon_6,$$

которые описываются матрицами с элементами 0 и 1. Единица на пересечении i -й строки и j -го столбца свидетельствует о физической реализуемости (допустимости) соединения соответствующих элементов рассматриваемых множеств.

Введем булевы переменные $x_i^{(l)}$ ($l = 1, 9$), которые принимают следующие значения:

$$x_i^{(l)} = \begin{cases} 1, & \text{если в состав АСОИ входит } e_i^{(l)} \in \varepsilon_l; \\ 0 & \text{— в противном случае.} \end{cases}$$

Будем считать известными величины средних интенсивностей обслуживания на каждом из устройств, входящих в множество ε_l . Обозначим их через $\mu_i^{(l)}$. В этом случае интенсивность обслуживания формально можно записать в следующем виде:

$$\mu_j(x_i^{(l)}) = \sum_{e^{(l)} \in \varepsilon_l} \mu_i^{(l)} x_i^{(l)}, \quad i = \overline{1, n_1}; \quad j = \overline{1, 9}.$$

Такая форма записи правомерна, если считать, что в состав каждой СМО включены только однотипные устройства в количестве k_i . Последнее условие можно записать в виде следующей системы равенств:

$$\sum_{e^{(1)} \in \varepsilon_1} x_i^{(1)} = 1;$$

$$\sum_{e^{(2)} \in \varepsilon_2} x_i^{(2)} = 1;$$

$$\sum_{e^{(9)} \in \varepsilon_9} x_i^{(9)} = 1;$$

Ранее неоднократно отмечалось, что анализ стохастических сетей обычно проводят в стационарном (установившемся) режиме, следовательно, это необходимо ввести в качестве ограничения.

Для того, чтобы существовал установившийся режим, каждая система сети должна удовлетворять условию ненасыщения, т. е.

$$\lambda_1 < \mu_1 (\bar{x}^{(1)}) \text{ или } \rho_1 < 1;$$

$$\lambda_2 < \mu_2 (\bar{x}^{(2)}) \text{ или } \rho_2 < 1;$$

$$\lambda_9 < \mu_9 (\bar{x}^{(9)}) \text{ или } \rho_9 < 1.$$

Следовательно, значение интенсивности внешнего источника должно удовлетворять неравенству (при условии, что интенсивность обслуживания не зависит от интенсивности поступления)

$$\lambda_0 < \min_{i=1,9} \frac{\mu_i (\bar{x}^{(e)})}{\alpha_i}.$$

Выполнение этого неравенства является необходимым условием существования установившегося режима в сети, состоящей из устройств, определяемых набором переменных $x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(9)}$. Таким образом, структура сети влияет только на расчет интенсивностей потоков. Если последние вычислены, то параметры установившегося режима зависят только от свойств составляющих сеть систем.

Выбор типов устройств, относящихся ко всем множествам $\varepsilon_1, \dots, \varepsilon_9$, должен производиться при выполнении физической стыковки их между собой. Ранее были введены бипарные матрицы $Q_1 - Q_9$, которые определяли возможность сопряжения устройств друг с другом, однако они не затрагивали количественной стороны вопроса. Например, к МК типа ЕС-4012, как это следует из его технических характеристик, можно подключать не более 192 периферийных устройств. Такого рода информация тоже должна найти отражение в ограничениях на оптимизируемую функцию.

Для связей типа процессор — мультиплексор (Пр—МК), процессор — селекторный канал (Пр—СК), процессор — ОЗУ (Пр—ОЗУ) можно записать соответственно три следующих выражения:

$$\eta^{(5)} = \sum_{e^{(5)} \in \varepsilon_5} \sum_{e^{(6)} \in \varepsilon_6} q_{ij}^{(5)} \cdot \eta_{ij}^{(5)} \cdot x_i^{(5)} \cdot x_j^{(6)};$$

$$\eta^{(8)} = \sum_{e^{(8)} \in \varepsilon_8} \sum_{e^{(6)} \in \varepsilon_6} q_{ij}^{(8)} \cdot \eta_{ij}^{(8)} \cdot x_i^{(8)} \cdot x_j^{(6)};$$

$$\eta^{(7)} = \sum_{e^{(7)} \in \varepsilon_7} \sum_{e^{(6)} \in \varepsilon_6} q_{ij}^{(7)} \cdot \eta_{ij}^{(7)} \cdot x_i^{(7)} \cdot x_j^{(6)},$$

где $\eta^{(5)}$ — максимально допустимое число МК, одновременно работающих с процессором; $\eta^{(8)}$ и $\eta^{(7)}$ — максимально допустимое число СК и ОЗУ.

Значения $\eta_{ij}^{(5)}$, $\eta_{ij}^{(8)}$, $\eta_{ij}^{(7)}$ имеют аналогичный $\eta^{(5)}$, $\eta^{(7)}$, $\eta^{(8)}$ смысл, но относятся к конкретным типам устройств, определяемых значениями булевых переменных.

Для связей типа МК — периферийное устройство (ПУ) и СК — ПУ, где ПУ — самостоятельные k_i -канальные СМО, можно записать следующие выражения:

$$\eta_{\text{МК}} = k_0 + k_1 \sum_{e^{(5)} \in \varepsilon_5} \sum_{e^{(1)} \in \varepsilon_1} q_{ij}^{(1)} \cdot x_i^{(5)} \cdot x_j^{(1)} + k_2 \sum_{e^{(5)} \in \varepsilon_5} \sum_{e^{(2)} \in \varepsilon_2} \times$$

$$\times q_{ij}^{(2)} \cdot x_i^{(2)} \cdot x_j^{(5)} + k_3 \sum_{e^{(5)} \in \varepsilon_5} \sum_{e^{(3)} \in \varepsilon_3} q_{ij}^{(3)} \cdot x_i^{(5)} \cdot x_j^{(3)} +$$

$$+ k_4 \sum_{e^{(5)} \in \varepsilon_5} \sum_{e^{(4)} \in \varepsilon_4} q_{ij}^{(4)} \cdot x_i^{(5)} \cdot x_j^{(4)};$$

$$\eta_{\text{СК}} = k_9 + \sum_{e^{(8)} \in \varepsilon_8} \sum_{e^{(9)} \in \varepsilon_9} q_{ij}^{(9)} \cdot x_i^{(8)} \cdot x_j^{(9)}.$$

Здесь $\eta_{\text{МК}}$ и $\eta_{\text{СК}}$ имеют смысл уже не максимальных количеств устройств, подключаемых к МК и СК, а фактического количества устройств, т. е. они показывают, сколько ПУ подключено к каждому из каналов в конкретной системе. Максимальное же значение ПУ, подключаемых к каналам, можно формально представить как

$$\eta_{\text{МК}_i}^{\text{max}} = \sum_{e^{(5)} \in \varepsilon_5} \eta_{\text{МК}_i}^{\text{max}} \cdot x_i^{(5)};$$

$$\eta_{\text{СК}_i}^{\text{max}} = \sum_{e^{(8)} \in \varepsilon_8} \eta_{\text{СК}_i}^{\text{max}} \cdot x_i^{(8)},$$

где $\eta_{\text{МК}_i}^{\text{max}}$, $\eta_{\text{СК}_i}^{\text{max}}$ — максимальное число ПУ, подключенных к каналам конкретных типов.

Представляя все найденные выражения в виде соответствующих ограничений, получим совокупность физических ограничений по стыкуемости отдельных устройств в систему:

$$k_5 - \eta^{(5)}(x_i^{(5)}, x_j^{(6)}) k_6 \leq 0;$$

$$k_8 - \eta^{(8)}(x_i^{(8)}, x_j^{(6)}) k_6 \leq 0;$$

$$k_7 - \eta^{(7)}(x_i^{(7)}, x_j^{(6)}) k_6 \leq 0;$$

$$\eta_{\text{МК}} - \eta_{\text{МК}}^{\text{max}} \leq 0;$$

$$\eta_{\text{СК}} - \eta_{\text{СК}}^{\text{max}} \leq 0.$$

Естественно потребовать, чтобы стоимость АСОИ не превосходила некоторой заданной величины. Это ограничение с учетом переменных $k_1, \dots, k_9; x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(9)}$ может быть представлено в виде следующего неравенства:

$$\sum_{e^{(1)} \in \varepsilon_1} C_i x_i^{(1)} k_i + \sum_{e^{(2)} \in \varepsilon_2} C_i x_i^{(2)} k_i + \dots + \\ + \sum_{e^{(9)} \in \varepsilon_9} C_i x_i^{(9)} k_i \leq C_{\text{АСОИ}}.$$

Таким образом, задача выбора оптимального состава АСОИ может быть сформулирована как задача определения переменных $x_i^{(1)}, x_i^{(2)} \dots x_i^{(9)}; k_1, k_2, \dots, k_9$, доставляющих минимум функционалу,

$$U = \sum_{i=1}^9 \frac{\alpha_i}{\mu_i} \left[\frac{k_i^{k_i} \rho_i^{k_i}}{k_i! (1 - \rho_i)^2} P_{0i} + k_i \right],$$

где

$$P_{0i} = \left[\frac{k_i^{k_i} \rho_i^{k_i}}{k_i! (1 - \rho_i)} + \sum_{j=0}^{k_i-1} \frac{k_i^j \rho_i^j}{j!} \right]^{-1}$$

при удовлетворении следующих условий:

$$\lambda_0 < \min \frac{\mu_i (\bar{x}^{(i)})}{\alpha_i};$$

$$\sum_{e^{(1)} \in \varepsilon_1} x_i^{(1)} = 1;$$

$$\sum_{e^{(2)} \in \varepsilon_2} x_i^{(2)} = 1;$$

• • • • •

$$\sum_{e^{(9)} \in \varepsilon_9} x_i^{(9)} = 1;$$

$$\begin{aligned}
& k_5 - \eta^{(5)}(x_i^{(5)}, x_j^{(6)}) k_6 \leq 0; \\
& k_8 - \eta^{(8)}(x_i^{(8)}, x_j^{(6)}) k_6 \leq 0; \\
& k_7 - \eta^{(7)}(x_i^{(7)}, x_j^{(6)}) k_6 \leq 0; \\
& \eta_{\text{МК}} - \eta_{\text{МК}}^{\text{max}} \leq 0; \\
& \eta_{\text{СК}} - \eta_{\text{СК}}^{\text{max}} \leq 0; \\
& e^{(1)} \sum_{\varepsilon_1} C_i x_i^{(1)} k_i + \sum_{e^{(2)} \in \varepsilon_2} C_i x_i^{(2)} k_i + \dots + \\
& + \sum_{e^{(9)} \in \varepsilon_9} C_i x_i^{(9)} k_i \leq C_{\text{АСОИ}}.
\end{aligned}$$

Сформулированная задача относится к классу задач целочисленного нелинейного программирования. Для решения задач нелинейного целочисленного программирования можно использовать, например, следующие методы [14]:

- линеаризации исходной целевой функции;
- ветвей и границ;
- случайного поиска;
- перебора и некоторые другие.

Линеаризация целевой функции сопряжена с существенными трудностями, вызванными прежде всего тем фактом, что каждое из слагаемых, составляющих целевую функцию, является нелинейной функцией двух переменных. Поэтому линеаризация такой функции методом плоскостей приведет к появлению большого числа дополнительных ограничений, потребует оценки точности аппроксимации, что в конце концов создаст значительные вычислительные трудности.

Метод ветвей и границ достаточно хорошо разработан и его можно было бы рекомендовать для решения стоящей перед нами задачи. Однако негативной стороной метода является его критичность к изменению начальных условий (например, множества альтернативных вариантов оборудования).

Наиболее приемлемыми, с нашей точки зрения, являются две последние группы методов — *случайного поиска и перебора*. Эти методы различаются способами перехода от одного найденного локального экстремума к другому. Самое главное, что они инвариантны по отношению к расширению множества $\varepsilon_1 \dots \varepsilon_9$, так как увеличение их размерности приводит лишь к дополнительным вычислительным итерациям.

Остановимся несколько подробнее на процедуре перебора, предложенной Балашем. Для сокращения числа вычислительных операций Балаш предложил осуществлять перебор допустимых решений с использованием фильтрующих ограничений. Поясним суть этого метода.

Пусть найдено какое-либо решение, удовлетворяющее всем имеющимся ограничениям. Тогда исследуемая функция U при-

нимает вполне определенное конкретное значение U'_{\min} . Введем дополнительное ограничение $U \leq U'_{\min}$. Оно названо *фильтрующим* и при дальнейшем рассмотрении новых векторов $x = \{x_{i1}^{(1)}, x_{i2}^{(2)}, \dots, x_{i9}^{(9)}\}$ и $k = \{k_{i1}, k_{i2}, \dots, k_{i9}\}$ анализируется прежде всего. Если ограничение выполняется, то проверяют, принадлежит ли новая точка $(x; k)$ области допустимых решений, определяемой основными ограничениями задачи. Если это условие принадлежности не выполняется, то рассматривают новую точку (x', k') . Если она принадлежит области допустимых решений, а значение функции U в этой точке меньше значения U'_{\min} , то вводят новое фильтрующее ограничение $U \leq U''_{\min}$, которое заменяет первое и т. д.

При построении конкретного алгоритма можно руководствоваться дополнительно следующим соображением. Очевидно, что влияние различных СМО на общее время реакции системы неравнозначно. Характеристикой этого влияния могут служить найденные ранее коэффициенты α_i , которые определяют частоту использования каждой СМО. Например, если считать, что в данной АСОИ преобладают процессы обработки информации и выдачи результатов обработки на следующий уровень испытательного комплекса, а операции вывода на ПУ составляют лишь незначительную часть времени работы АСОИ, то, очевидно, будет выполняться двойное неравенство $\alpha_6 > \alpha_1$; $\alpha_7 > \alpha_2$. Имея их в виду, можно показать целесообразность осуществления сначала перебора элементов множества $\varepsilon_6(\varepsilon_7)$, а лишь затем $\varepsilon_1(\varepsilon_2)$. В этом случае быстрее удастся получить наиболее сильное фильтрующее ограничение, вследствие чего резко сокращаются вычислительные затраты. Для наиболее полного осуществления указанного приема целесообразно провести упорядочение элементов $e^{(i)} \in \varepsilon_i$ по возрастанию частоты их использования.

7.3. ЭВРИСТИЧЕСКИЙ МЕТОД СИНТЕЗА ОПТИМАЛЬНОЙ СТРУКТУРЫ АСОИ И РАСПРЕДЕЛЕНИЯ ПО ЕЕ ЭЛЕМЕНТАМ ИНФОРМАЦИОННОГО ПОТОКА*

Один из существенных недостатков метода математического программирования — это то, что получаемое решение оперирует с непрерывными величинами, а на практике выбор характерис-

* В разработке данного раздела принимал участие А. В. Тучкин.

тик устройств АСОИ возможен лишь из дискретного множества значений. Кроме того, обычно предполагается, что проектировщик располагает априорной информацией о количестве устройств в АСОИ и способе их комплексации в единую систему, который однозначно определяет величины информационных потоков на входе отдельных подсистем АСОИ. Наконец, в большинстве работ, дающих возможность при синтезе АСОИ учесть и экономические показатели, использованы трудно определяемые на ранних этапах проектирования коэффициенты, отражающие связь основной характеристики (обычно это быстродействие) с выделенными на ее создание стоимостными или какими-либо другими ресурсами.

Предлагаемый метод аналитического синтеза оптимальных структур АСОИ свободен от перечисленных выше недостатков. В качестве математической модели АСОИ выбрана матричная или многофазная модель сети [27], что правомерно, поскольку процесс обработки информации в АСОИ обычно состоит из нескольких этапов или фаз. Под фазой будем понимать такие операции, как ввод, процессирование, запись и хранение информации в накопителях, отображение и передача данных, причем на каждой фазе могут быть использованы несколько устройств, возможно с различными операционными характеристиками. При проектировании АСОИ, как правило, априорно известно среднее значение информационного потока, поступающего на вход системы, а также в каждую из ее фаз обработки (ФО). На общую же конфигурацию АСОИ и количество устройств на каждой фазе обработки не наложено никаких ограничений.

Поскольку на начальном этапе проектирования еще не известно, сколько устройств и какого типа будет на каждой из фаз обработки, а тем более каково распределение потока информации между устройствами, то совместное решение этих задач затруднено.

Отличие описываемого метода от известных ранее существенно и состоит в том, что при оптимальном синтезе используют лишь характеристики реальных технических устройств, имеющих в распоряжении проектировщика. Синтез АСОИ проводят в два этапа. На первом этапе выбирают отдельные фазы обработки, причем каждая фаза обработки может быть как однородной, так и модульной. На втором — решают задачу синтеза АСОИ в целом. Теперь опишем подробно процедуры, используемые на каждом из этапов.

Задача выбора фазы обработки заключается в определении количества устройств каждого типа и распределении информационного потока по этим устройствам таким образом, чтобы выбранный критерий эффективности принимал экстремальное

значение. Математически эту задачу можно записать в следующем виде:

$$\text{найти } \underset{k_i \lambda_i}{\text{extr}} \left[\sum_{i=1}^n k_i f_i(\bar{\pi}_i \lambda_i) \right] \quad (7.1)$$

$$\text{при условии, что } \sum_{i=1}^n k_i \lambda_i = \Lambda, \quad k_i = \overline{0,1}, \quad (7.2)$$

где n — число типов устройств, которые могут войти в состав фазы обработки;

k_i — количество устройств i -го типа, вошедших в состав фазы обработки;

$f_i(\bar{\pi}_i, \lambda_i)$ — некоторая заданная критериальная функция (заметьте, что в состав вектора $\bar{\pi}_i$ могут входить такие параметры, как стоимость, оперативность, вес, потребляемая энергия и т. п.);

Λ — общий поток информации, поступающей на фазу обработки;

λ_i — поток информации, поступающей на i -е устройство.

Сформулированную задачу относят к классу задач «о рюкзаке», однако она имеет несколько отличную от классической постановку, что усложняет ее решение. Отличие состоит в том, что наряду со значениями k_i требуется определить оптимальные значения неизвестных величин потоков λ_i , поступающих на каждое из устройств фазы обработки (ФО).

Решение можно проводить с использованием «вложенных» алгоритмов динамического программирования, однако тогда сложно добиться сколько-нибудь приемлемой точности решения. В связи с этим был разработан новый алгоритм решения, основанный на последовательном формировании промежуточных оптимальных структур при пошаговом изменении входного потока информации.

Алгоритм синтеза фазы обработки

Предварительно произведем дискретизацию входного потока Λ . Выберем число уровней дискретизации равным m , тогда $m \Delta \lambda = \Lambda$, где $\Delta \lambda$ — шаг дискретизации. Кроме того, для определенности предположим, что необходимо произвести минимизацию целевой функции (7.1) при условии (7.2).

ШАГ А. Формируем элементы параметрической матрицы H размерностью $m \times n$, каждый элемент которой h_{ij} ($i = \overline{1, m}, j = \overline{1, n}$) равен:

$$h_{ij} = f_i(\bar{\pi}_j, i \Delta \lambda).$$

В случае, когда система j по своим характеристикам не способ-

на обработать поток $i \Delta \lambda$, элементу присваивается значение ∞ .

ШАГ В. Формируем элементы вектор-столбцов:

$$X = \{ |x_i| \} \quad \text{и} \quad Y = \{ |y_i| \}, \quad (i = \overline{1, m});$$
$$y_i = \min_{j=1, n} \{ h_{ij} \}; \quad x_i = j,$$

где j — номер устройства, для которого в i -й строке значение минимально.

ШАГ С. Конструктивно шаг С оформлен в виде подпрограммы, алгоритм которой представлен на рис. 7.4.

Используемые идентификаторы: NS — массив размерности m , i -й элемент которого равен числу систем, в которые поступает поток величиной $i \Delta \lambda$ (массив промежуточных структур); $NSPR$ — массив размерности $m \times m$, в котором хранятся промежуточные структуры для всех значений распределяемого потока; MXP — массив размерности m , содержащий корректирующий список, его элементы равны порядковым номерам итераций алгоритма, на которых происходила замена промежуточной структуры ΦO одним устройством. Напомним, что номер шага итерации однозначно определяет величину потока информации, поэтому добавление очередного элемента к корректирующему списку означает, что данный поток выгоднее направить в одну систему ΦO , чем распределять его по нескольким устройствам; IMX — число элементов в корректирующем списке;

E — массив размерности m , i -й элемент которого равен величине критериальной функции для оптимальной промежуточной структуры на i -м шаге;

Y — массив размерности m , сформированный на шаге В.

Разберем подробнее алгоритм работы программы на шаге С. Она состоит из отдельных подблоков, обозначенных как С1-С10. Подблок С1 осуществляет обнуление всех элементов массивов NS , $NSPR$, MXP и переменной TMX . Организуется цикл последовательного синтеза оптимальных промежуточных структур $II = \overline{1, m}$. В С2 производится формирование очередной промежуточной структуры для потока ($II \Delta \lambda$). Подблок С3 определяет величину критерия эффективности для промежуточной структуры $E(II)$. В С4 проверяется условие необходимости корректировки промежуточной структуры по алгоритму «А». Алгоритм корректировки «А» проводит целенаправленное изменение текущей промежуточной структуры с целью уменьшения значения критерия эффективности. Работа алгоритма «А» будет продемонстрирована на примере, приводимом ниже. Возможность замены ранее сформированной промежуточной структуры одной системой проверяется в подблоке С5. Если такая замена не выгодна, то алгоритм переходит к С10. В С6 осуществляется изменение величины критерия эффективности промежуточной струк-

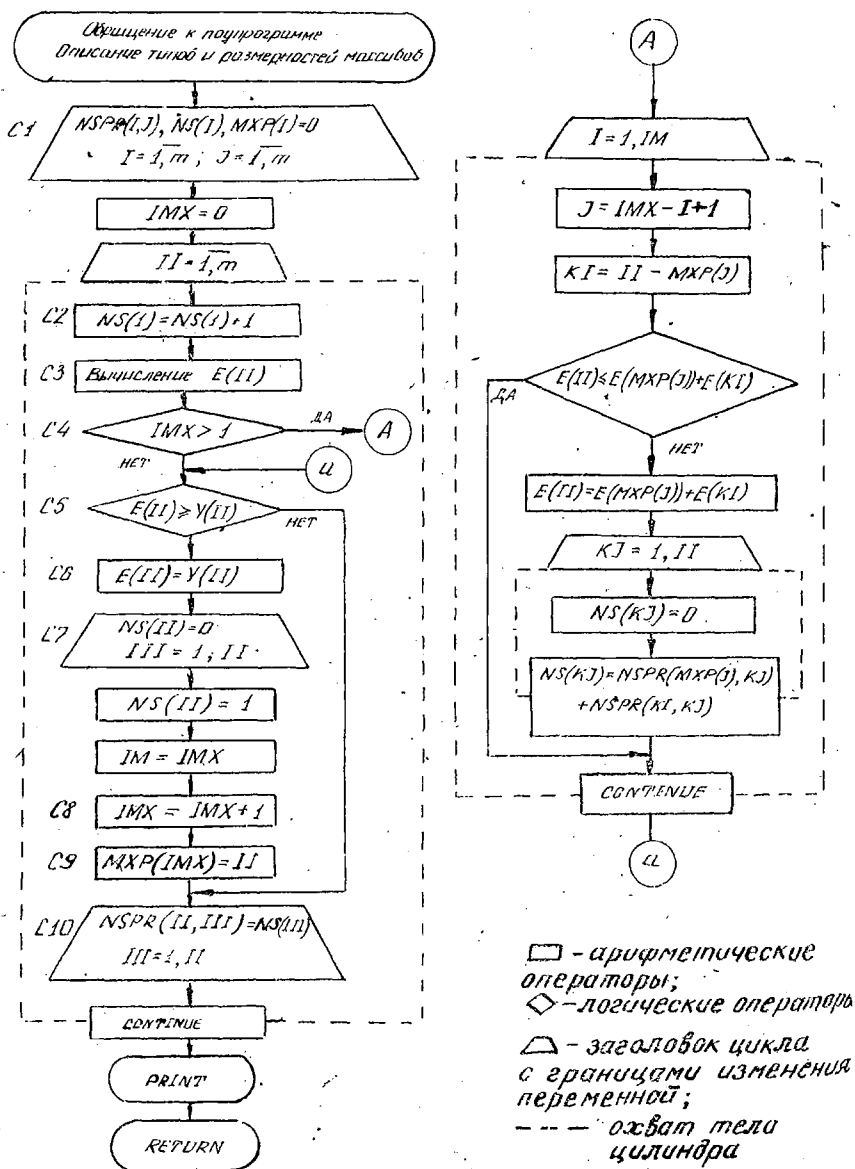


Рис. 7.4. Алгоритм формирования и анализа структур фазы обработки

туры. С7 формирует новую промежуточную структуру, а в С8 производится изменение величины IMX . Включение номера текущего шага в корректирующий список осуществляется в С9, а хранение полученной на шаге II оптимальной промежуточной структуры — в С10. Если по системам распределен не весь поток информации Λ , переходим к С2. Сформированная конечная оптимальная структура выводится на печать.

Для пояснения принципа работы алгоритма синтеза фазы обработки рассмотрим числовой пример.

Пример. Пусть в состав фазы обработки входят устройства шести различных типов, отличающиеся своими техническими характеристиками. В ФО поступает поток информации в 22 единицы. Для примера примем шаг дискретизации равным двум.

ШАГ А. Подставив заданные операционные характеристики устройств в выражение выбранного нами критерия эффективности, получим матрицу следующего вида (табл. 7.1).

Таблица 7.1

Параметрическая матрица H

Поток	Номер системы					
	1	2	3	4	5	6
2	3,2	4,4	3,0	4,8	3,3	4,3
4	3,3	5,1	3,1	5,1	3,8	4,8
6	3,4	5,8	4,6	6,3	4,2	5,6
8	7,5	6,3	15,4	8,5	6,6	7,8
10	9,3	6,4	∞	12,3	7,1	8,2
12	12,6	8,6	∞	18,4	7,5	10,8
14	15,7	∞	∞	∞	10,3	12,6
16	∞	∞	∞	∞	∞	13,8
18	∞	∞	∞	∞	∞	∞
20	∞	∞	∞	∞	∞	∞
22	∞	∞	∞	∞	∞	∞

ШАГ В. Векторы X и Y имеют вид

$$X = \{3, 3, 1, 2, 2, 5, 5, 6, \infty, \infty, \infty\}.$$

$$Y = \{3,0; 3,1; 3,4; 6,3; 6,4; 7,5; 10,3; 13,8; \infty; \infty; \infty\}$$

ШАГ С

E

3.0 $II=1$; $NS=\{1,0,\dots,0\}$; $E(1)=3.0$; $IMX=1$; $MXP=\{1,0,\dots\}$; $NS \rightarrow NSPR$

3.1 $II=2$; $NS=\{2,0,\dots,0\}$; $E(2)=6.0$; $E(2) > Y(2)$; $E(2)=3.1$

$NS=\{0,1,\dots,0\}$; $IMX=2$; $MXP=\{1,2,0,\dots,0\}$; $NS \rightarrow NSPR$

3.4 $II=3$ $NS=\{1,1,0,\dots,0\}$; $E(3)=6.1$; $E(3) > Y(3)$; $E(3)=3.4$

$NS=\{0,0,1,\dots,0\}$; $IMX=3$; $MXP=\{1,2,3,0,\dots\}$; $NS \rightarrow NSPR$

6.2 $II=4$; $NS=\{1,0,1,0,\dots\}$; $E(4)=6.4$ $\begin{bmatrix} E(3) + E(4) \\ E(2) + E(2) \\ E(1) + E(3) \end{bmatrix} \begin{bmatrix} 6.4 \\ 6.2 \\ 6.4 \end{bmatrix} \rightarrow NS=\{0,2,0,\dots\}$

$E(4)=6.2$; $E(4) < Y(4)$

$NS \rightarrow NSPR$

6.5 $II=5$; $NS=\{1,2,0,\dots\}$; $E(5)=9.2$; $\begin{bmatrix} E(3) + E(2) \\ E(2) + E(3) \\ E(1) + E(4) \end{bmatrix} \begin{bmatrix} 6.5 \\ 6.5 \\ 9.2 \end{bmatrix} \rightarrow NS=\{0,1,1,0,\dots\}$

$E(5)=6.5$; $E(5) > Y(5)$; $E(5)=6.4$

$NS=\{0,0,0,0,1,\dots\}$; $IMX=4$; $MXP=1,2,3,5,0,\dots$

6.8 $II=6$; $NS=\{1,0,0,0,1,0,\dots\}$; $\begin{bmatrix} E(5) + E(1) \\ E(3) + E(3) \\ E(2) + E(4) \\ E(1) + E(5) \end{bmatrix} \begin{bmatrix} 9.4 \\ 6.8 \\ 9.3 \\ 9.4 \end{bmatrix} \rightarrow NS=\{0,0,2,0,\dots\}$

$E(6)=9.4$

$E(6)=6.8$; $E(6) < Y(6)$;

$NS \rightarrow NSPR$

$II=7$; $NS=\{1,0,2,\dots\}$; $E(7)=9.8$

9.5 $E(7)=9.5$; $E(7) < Y(7)$ $\begin{bmatrix} E(5) + E(2) \\ E(3) + E(4) \\ E(2) + E(5) \\ E(1) + E(6) \end{bmatrix} \begin{bmatrix} 9.5 \\ 9.6 \\ 9.5 \\ 9.8 \end{bmatrix} \rightarrow NS=\{0,1,0,0,1,\dots\}$

$NS \rightarrow NSPR$

9.8 $II=8$; $NS=\{1,1,0,0,1,\dots\}$; $E(8)=12.5$

$E(8)=9.8$; $E(8) < Y(8)$; $\begin{bmatrix} E(5) + E(3) \\ E(3) + E(5) \\ E(2) + E(6) \\ E(6) + E(7) \end{bmatrix} \begin{bmatrix} 9.8 \\ 9.8 \\ 9.9 \\ 12.5 \end{bmatrix} \rightarrow NS=\{0,0,1,0,1,\dots\}$

$NS \rightarrow NSPR$

$$10.2 \quad II=9; NS=\{1,0,1,0,1,\dots\}; E(9)=12.4$$

$$E(9)=10.2; E(9) < Y(9); \begin{bmatrix} E(5)+E(4) \\ E(3)+E(6) \\ E(2)+E(7) \\ E(1)+E(9) \end{bmatrix} \begin{bmatrix} 12.6 \\ 10.2 \\ 12.6 \\ 12.4 \end{bmatrix} \rightarrow NS=\{0,0,3,0,\dots\}$$

$NS \rightarrow NSPR$

$$12.8 \quad II=10; NS=\{1,0,3,0,\dots\}$$

$$E(10)=13.2$$

$$E(10)=13.8; E(10) < Y(10)$$

$NS \rightarrow NSPR$

$$\begin{bmatrix} E(5)+E(5) \\ E(3)+E(7) \\ E(2)+E(8) \\ E(1)+E(9) \end{bmatrix} \begin{bmatrix} 12.8 \\ 12.9 \\ 12.9 \\ 13.2 \end{bmatrix} \rightarrow NS=\{0,0,0,0,2,\dots\}$$

13.2

$$II=11; NS=\{1,0,0,0,2,\dots\}; E(11)=15.8$$

$$E(11)=13.2$$

$$E(11) < Y(11); NS \rightarrow NSPR$$

$$\begin{bmatrix} E(5)+E(6) \\ E(3)+E(8) \\ E(2)+E(9) \\ E(1)+E(10) \end{bmatrix} \begin{bmatrix} 13.2 \\ 13.2 \\ 13.4 \\ 15.8 \end{bmatrix} \rightarrow NS=\{0,0,2,0,1,\dots\}$$

$$13.6 \quad II=12 \quad NS=\{1,0,2,0,1,\dots\}; E(12)=16.2$$

$$E(12)=13.6; E(12) < Y(12)$$

$NS \rightarrow NSPR$

$$\begin{bmatrix} E(5)+E(7) \\ E(3)+E(9) \\ E(2)+E(10) \\ E(1)+E(11) \end{bmatrix} \begin{bmatrix} 15.9 \\ 13.6 \\ 15.9 \\ 16.2 \end{bmatrix} \rightarrow NS=\{0,0,4,0,\dots\}$$

В результате работы алгоритма на печать выводится оптимальная структура и распределение потока информации по системам. В данном случае оптимальная структура фазы обработки должна содержать два устройства первого типа, на которые поступает поток информации в шесть условных единиц и одно устройство второго типа, на которое поступает поток в десять единиц.

Особенностью приведенного алгоритма является возможность получения оптимальных структур на всех промежуточных шагах, т. е. для всех потоков $\Delta \lambda_k$ ($k=1, m$), для которых производился расчет. Программная реализация алгоритма требует небольших временных затрат и размеров памяти ЭВМ, которые почти линейно зависят от числа уровней дискретизации m и мало зависят от числа типов устройств, используемых в процессе синтеза АСОИ.

Синтез оптимальной структуры АСОИ

Переходя ко второму, заключительному этапу синтеза АСОИ, отметим, что наиболее общий случай, когда необходимо синтезировать систему в соответствии с многомерной целевой функ-

щей, весьма сложен. Более того, построение многомерной целевой функции само по себе является трудной задачей.

Рассмотрим простой вариант, часто встречающийся в практике проектирования, когда требуется минимизировать стоимость проектирования АСОИ при заданной оперативности. При чем под стоимостью будем понимать любой аддитивный показатель, с помощью которого оценивается АСОИ: деньги, вес, потребляемая энергия и т. п.

Опять, как и ранее, проводим последовательное проектирование фаз обработки АСОИ. Считаем, что ФО может быть либо однородной, т. е. состоять из однотипных устройств, либо модульной, когда в ее состав входят устройства различных типов. Для решения задачи синтеза запишем выражение (7.1) в виде:

$$\text{найти} \quad \min \left[\sum_{i=1}^n k_i (a_i U_i + \kappa S_i) \right], \quad (7.3)$$

где $a = \lambda_i/\Lambda$ — коэффициент передачи, показывающий, какая часть потока информации поступает в i -е устройство ФО;

U_i — время пребывания единицы информации (заявки) в i -м устройстве;

S_i — стоимость i -го устройства;

κ — положительный коэффициент, смысл которого будет рассмотрен ниже.

Представив все устройства ФО как системы массового обслуживания типа $M/M/1$ и учитывая, что $U_i = (\mu_i - \lambda_i)^{-1}$, выражение (7.3) перепишем окончательно в виде

$$\sum_{i=1}^n k_i \frac{\lambda_i}{\Lambda (\mu_i - \lambda_i)} + \kappa \sum_{i=1}^n k_i S_i \rightarrow \min, \quad (7.4)$$

Для случая синтеза однородной фазы обработки $n=1$ и $\Lambda = k\lambda$ поиск безусловного минимума выражения (7.4) по k (индексы i здесь можно опустить) приводит к следующему результату:

$$k = \min_T \left[\frac{\Lambda + \sqrt{\frac{\Lambda}{\kappa S}}}{\mu} \right]^{\pm} \quad (7.5)$$

где μ — интенсивность обслуживания;

T — множество типов устройств, из которых можно создать однородную фазу обработки;

$[\cdot]^{\pm}$ — символ округления выражения $[\cdot]$ до ближайшего целого значения сверху или снизу.

Поскольку все параметры в выражении (7.4) положительные, то, варьируя численные значения коэффициента κ , можно получить табличные значения функции $T = \varphi(S)$. Иными сло-

вами, для каждой фазы обработки, зная объем выделенных средств S на ее создание, можно не только определить, какой оперативностью будет обладать оптимальная структура, но и получить саму структуру.

Для однородных структур определим функцию $\varphi(S)$ прямой подстановкой различных коэффициентов κ в формулу (7.5), а для модульной структуры — последовательным многократным применением описанного выше алгоритма с подстановкой различных значений κ в выражение (7.4) и использованием уравнения связи (7.2). Действительно, при $\kappa \rightarrow 0$ получается самая быстродействующая структура ΦO , при этом стоимость не имеет решающего значения. При $\kappa \rightarrow \infty$, наоборот, определяющее значение имеет стоимость, поэтому синтезируется структура минимальной стоимости, и количество устройств определяется лишь условием существования стационарного режима в ΦO (см. выражение (7.5)). Между граничными значениями κ лежит весь спектр оптимальных по быстродействию структур ΦO , различных по стоимости создания.

Функциональные зависимости $T_i = \varphi_i(S)$ для каждой из ΦO АСОИ с помощью программной реализации алгоритма можно построить достаточно быстро, полученные данные будут являться исходными для дальнейшего проектирования.

В качестве примера рассмотрим синтез одной из ΦO АСОИ, в состав которой входят устройства, характеристики которых сведены в табл. 7.2 (значения характеристик даны в условных единицах).

Таблица 7.2

Характеристики устройств

Операционная характеристика	Тип устройства							
	A	B	C	D	E	F	G	H
Быстродействие	3	21	9	18	16	7	12	14
Стоимость	10	486	57	296	207	33	131	162

Результаты моделирования в виде графических зависимостей представлены на рис. 7.5. Из графиков следует, что если в ΦO АСОИ, на создание которой выделен ресурс 375 ед., поступает поток $\lambda = 56$ ед., то из заданной номенклатуры устройств (см. табл. 7.2) нельзя построить ΦO со временем реакции меньше, чем 0,5 ед. Оптимальная ΦO для рассматриваемого случая состоит из 11 устройств типа F и одного устройства типа A, в которые поступает поток 5 и 1 ед. соответственно.

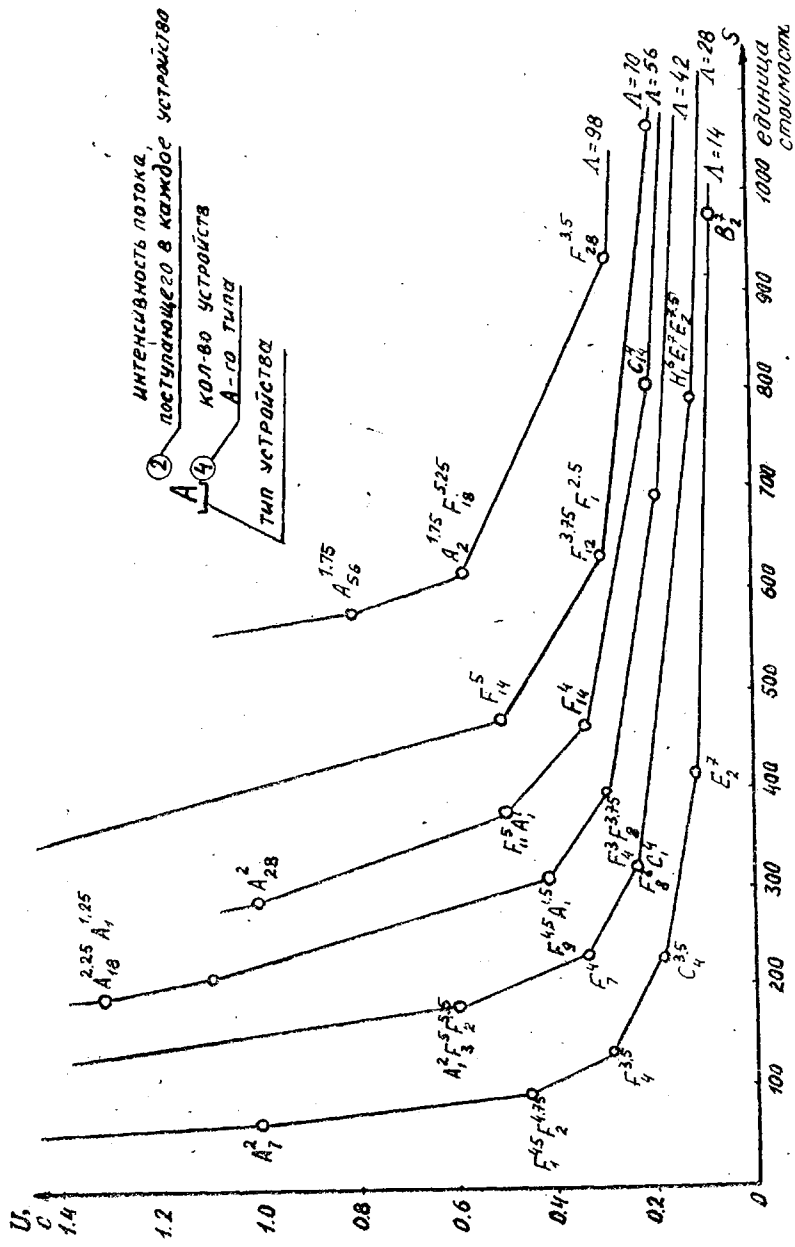


Рис. 7.5. Результаты моделирования

От создания отдельных ФО можно переходить к проектированию АСОИ в целом. Время пребывания заявок в АСОИ будет определяться по формуле

$$U = \sum_{j=1}^N \alpha_j T_j,$$

где α_j — коэффициент передачи;

T_j — время пребывания заявки в j -й ФО;

N — число ФО в АСОИ.

Таким образом, задача синтеза АСОИ минимальной стоимости с ограничениями на среднее время ответа математически будет сформулирована следующим образом: определить

$$S = \sum_{j=1}^N \varphi_j^{-1}(T_j) \rightarrow \min \quad (7.6)$$

при
$$\sum_{j=1}^N \alpha_j T_j \leq U^*, \quad (7.7)$$

где $\varphi_j^{-1}(T_j)$ — функция, обратная ранее полученной $\varphi_j(S_j)$;

U^* — максимально допустимое время ответа АСОИ;

S — суммарная стоимость АСОИ.

Поиск минимума аддитивной функции (7.6) при ограничениях (7.7) оказывается типичной однопараметрической задачей динамического программирования. Решение задачи (7.6) — (7.7) можно получить с помощью алгоритма, основанного на следующем функциональном уравнении Беллмана:

$$S_{k+1} = \min_{0 < \alpha_{k+1} T_{k+1} < U^*} \{ \varphi_{k+1}^{-1}(T_{k+1}) + S_k (U^* - \alpha_{k+1} T_{k+1}) \}.$$

После выполнения указанной вычислительной процедуры можно протабулировать значения функции $S = \varphi(U^*)$, что даст возможность определять оптимальные величины стоимостных затрат на создание АСОИ с различными временами ответа.

Изложенный метод синтеза оптимальных структур АСОИ дополняет известные и описанные ранее в гл. 4, 5 настоящего пособия. Его применение может быть полезно в случаях синтеза многофазной АСОИ из отдельных модульных устройств. Перспективность подобного подхода может быть объяснена не только тенденцией создания АСОИ из унифицированных и стандартизированных блоков, но и возможностью использования разработанных методов в системах автоматизированного проектирования АСОИ.

7.4. ВЫБОР ЧИСЛА ОДНОРОДНЫХ ВЫЧИСЛИТЕЛЬНЫХ МОДУЛЕЙ И КОМПЛЕКСИРОВАНИЕ ЗАДАЧ ОБРАБОТКИ ИНФОРМАЦИИ

В определенных условиях задачу выбора количества однородных вычислительных модулей АСОИ и распределения по ним реализуемых алгоритмов обработки информации можно сформулировать следующим образом:

требуется единственным способом разбить множество задач обработки информации $S = (S_1, S_2, \dots, S_n)$ на непересекающиеся подмножества. Причем количество образуемых подмножеств m заранее не задано. Каждое подмножество задач решается на своем вычислительном модуле.

Задано ограничение вида

$$\sum_{S_i \in M_j} \rho_i \leq \rho^*, \quad j = \overline{1, m}, \quad (7.8)$$

где ρ_i — коэффициент загрузки вычислительного модуля i -й задачей;

ρ^* — заданный предельный уровень загрузки j -го модуля;
 M_j — подмножество задач, решаемых на j -м модуле.

Подход к решению данной задачи зависит от уровня априорной информации относительно характеристик решаемых задач, степени их взаимосвязи и принятого критерия разбиения задач на подмножества (комплексирования) E . Так, например, в постановке задачи, описанной ниже, учтена достаточно доступная на ранних стадиях проектирования АСОИ априорная информация относительно попарной связности задач обработки информации.

Степень парной связности задач — степень положительного (отрицательного) эффекта от назначения любой пары задач из заданного множества S на один вычислительный модуль задается квадратной симметричной матрицей A . Физическая интерпретация элементов матрицы A может быть различной, в зависимости от принятого уровня описания характеристик задач и вычислительных модулей, а также в зависимости от принятого критерия эффективности результата комплексирования E . Так, если элементы матрицы A выражают собой объемы информации попарного обмена между задачами, а E — суммарный объем межмодульного обмена, то задача комплексирования может трактоваться как задача минимизации объема межмодульного обмена информацией.

С учетом вышесказанного задачу комплексирования можно сформулировать следующим образом. Необходимо распределить множество задач S по некоторому числу m (заранее неизвестному) однородных вычислительных модулей таким образом, чтобы обеспечить экстремальное (максимальное или минималь-

ное) значение критерия E -суммарного объема межмодульных связей, если заданы матрица A и вектор $R = (r_1, r_2, \dots, r_n)$. При этом необходимо учесть ограничение (7.8).

Если A и E единственные, то эта задача может быть решена с помощью эвристического алгоритма, являющегося развитием метода корреляционных плеяд [31]. В случае, если задан единственный вектор R и несколько матриц связности, имеющих различный физический смысл, задача отыскания единственного плана распределения задач по модулям может быть решена с помощью подхода, дающего квазиоптимальное решение.

Пусть мы имеем матрицы связности A_1, A_2, \dots, A_l . E_1, E_2, \dots, E_l — соответствующие показатели качества комплексирования (распределения задач по модулям). Пусть P_1, P_2, \dots, P_l — планы распределения, обеспечивающие экстремальные значения критериям E_1, E_2, \dots, E_l . Обозначим эти значения как $E_1^0, E_2^0, \dots, E_l^0$; $E_i(P_j)$ — значения критерия E_i , если выбран план распределения P_j . Если $i=j$, то $E_i(P_i) = E_i^0$. Если $i \neq j$, то $E_i(P_j) \geq < E_i^0$,

Введем показатель

$$\varepsilon_i(P_j) = \frac{|E_i(P_j) - E_i(P_i)|}{E_i(P_i)} = \frac{|E_i(P_j) - E_i^0|}{E_i^0}, \quad (7.9)$$

где $\varepsilon_i(P_j)$ при $i \neq j$ — относительное ухудшение эффекта комплексирования по i -му критерию при замене оптимального плана P_i на некоторый план P_j . Ясно, что при $i = j$, $\varepsilon_i(P_i) = 0$,

Тогда задачу отыскания единственного плана распределения P^* можно сформулировать, например, как задачу отыскания такого плана $P_j \in (P_1, P_2, \dots, P_l)$, который обеспечивал бы минимум средней суммы относительного ухудшения эффекта комплексирования, т. е. необходимо найти

$$\delta = \min_{P_j \in (P_1, P_2, \dots, P_l)} \frac{1}{l} \sum_{i=1}^l \varepsilon_i(P_j). \quad (7.10)$$

Развитие алгоритма комплексирования заключается в следующем:

1. Для каждой пары A_i, E_i найти оптимальный план распределения P_i . Получим P_1, P_2, \dots, P_l .

2. Для каждого P_i ($i = \overline{1, l}$) подсчитать эффект комплексирования согласно A_j и $j \neq i$.

3. Используя показатель (7.9), составить матрицу вида

$$\varepsilon = \begin{vmatrix} 0 & \varepsilon_2(P_1) & \varepsilon_3(P_1) & \dots & \varepsilon_l(P_1) \\ \varepsilon_1(P_2) & 0 & \varepsilon_3(P_2) & \dots & \varepsilon_l(P_2) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \varepsilon_1(P_l) & \varepsilon_2(P_l) & \dots & \dots & 0 \end{vmatrix}$$

4. Найти строку матрицы ϵ с минимальной средней суммой элементов. Номер данной строки j определит единственный квазиоптимальный план распределения $P_j = P^*$. При необходимости в выражение (7.10) можно включить весовые коэффициенты, учитывающие важность соответствующих критериев.

8.1. СРЕДСТВА МОДЕЛИРОВАНИЯ И МОДЕЛИ

Наиболее универсальным подходом к созданию и исследованию математических моделей функционирования АСОИ является имитационное моделирование [32, 34]. Имитационная модель позволяет программно описать и «проиграть» на инструментальной ЭВМ процесс функционирования таких систем, адекватные аналитические модели которых получить невозможно. Однако разработка (программирование) имитационной модели — моделирующего алгоритма — весьма трудоемкий процесс. Поэтому эффективность решения задач анализа АСОИ с помощью имитационного моделирования существенно зависит от имеющихся в распоряжении исследователя средств взаимодействия с инструментальной ЭВМ. В качестве таких средств применимы:

универсальные языки программирования, такие как ФОРТ-РАН, АЛГОЛ, ПЛ/1 и т. п.;

общецелевые языки моделирования дискретных систем, такие как GPSS, НЕДИС, СИМУЛА и др. [36, 47];

специализированные языки моделирования вычислительных систем: GSS-II, ECSS, SONET и др. [35, 39].

По уровню отображения структурных и функциональных компонент АСОИ можно выделить следующие типы моделей:

1. Обобщенные модели, отображающие АСОИ как многофазную систему массового обслуживания, в которой все компоненты сети представлены как обслуживаемые приборы типа устройство или буфер, а отдельные сообщения — заявками. Эти модели достаточно эффективно можно описать на общецелевых языках моделирования типа GPSS.

2. Модели АСОИ, в которых в явной форме отображены такие компоненты, как топология — узлы, линии связи, алгоритмы маршрутизации и диспетчеризации. Для построения этих моделей часто применяются специализированные системы моделирования, оформленные как пакеты процедур на универсальных языках программирования или проблемно-ориентированные языки моделирования [37, 40]. К подобным средствам ана-

лиза АСОИ, и в частности сетей передачи и обработки информации, относят язык SONET, позволяющий в 2—10 раз снизить объем описания моделей по сравнению с GPSS и увеличить скорость моделирования; пакет МОДЕЛЬ, ориентированный на имитацию сетей с коммутацией сообщений [38].

3. Модели, отображающие структуру операционных систем, параметры отдельных устройств, обрабатывающих ЭВМ, например: имитационная система GSS-II, ориентированная на анализ вычислительных систем на базе ЭВМ IBM 360, 370.

Естественно, что специализация языков увеличивает компактность и наглядность описаний моделей АСОИ за счет определенной потери гибкости языка и, в ряде случаев, эффективности использования ресурсов инструментальной ЭВМ. Однако следует учесть, что структура и параметры АСОИ часто могут быть описаны в рамках некоторой комбинации элементов, взятых из стандартного ограниченного набора модулей аппаратного и программного обеспечения (процессор, канал ввода-вывода, буферное запоминающее устройство и т. п.). Тогда альтернативные варианты АСОИ, среди которых отыскивают наилучший, будут отличаться структурой связей элементов, характеристиками алгоритмов преобразования информации, а также значениями параметров технических устройств. В этих условиях целесообразно использовать специализированные системы моделирования, ориентированные на построение макромодульных моделей АСОИ. Это позволяет повысить наглядность моделей. В главе изложены принципы построения и использования системы макромодульной имитации функционирования информационных систем (СММИФ) [44], как возможного альтернативного варианта построения специализированного пакета прикладных программ, предназначенного для исследования имитационных моделей АСОИ.

8.2. ПРИНЦИПЫ ПОСТРОЕНИЯ И СТРУКТУРА СИСТЕМЫ МАКРОМОДУЛЬНОЙ ИМИТАЦИИ ФУНКЦИОНИРОВАНИЯ АСОИ

СММИФ — удобный инструмент для исследования и анализа таких существенных операционных характеристик АСОИ, как среднее, дисперсия, максимальное значение времени задержки сообщений в системе, включая интервал пребывания сообщения в прерванном состоянии; вероятность потери сообщений, статистический ряд (гистограмма) времени задержки, коэффициенты использования процессоров, накопителей, памяти, максимальная и средняя длины очередей, статистический ряд для интервала между сообщениями в потоке. Средства СММИФ позво-

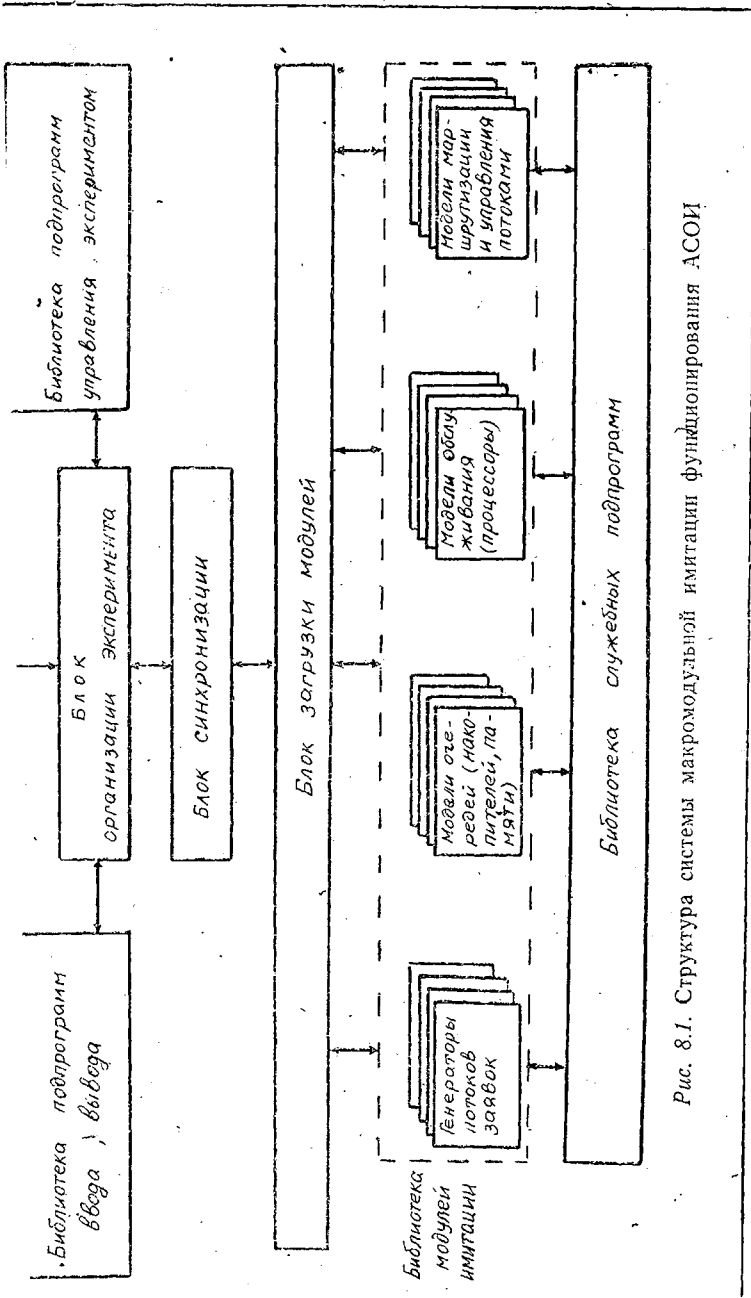


Рис. 8.1. Структура системы макромодульной имитации функционирования АСОИ

ляют освободить пользователя от непосредственного программирования имитационной модели. Его задача — грамотно сформулировать задачу исследований, провести планирование модельного эксперимента, задать набор исходных данных для активизации модели, а после окончания экспериментов с моделью — провести интерпретацию результатов согласно поставленной задаче.

Описание модели системы и задание исходных данных производят с помощью стандартных табличных конструкций и схем формирования последовательности управляющих параметров. Это достигается за счет введения стандартной формы сопряжения отдельных элементов моделируемой системы и разработки библиотеки программ имитации на макромодульной основе. Библиотека пакета содержит ограниченное (для отдельной стадии его развития) число типовых имитационных модулей, с помощью которых можно «набирать» модель сложной системы с количеством подсистем, превышающим число модулей. Последнее осуществляется путем специальной настройки модуля на место структурного элемента модели.

Структура СММИФ показана на рис. 8.1.

Пакет реализован на языке ФОРТРАН в ОС ЕС ЭВМ. *Блок организации эксперимента* осуществляет функции настройки пакета на заданную программу модельного эксперимента и управляет процессом генерации вариантов моделируемой системы. Он осуществляет ввод исходных данных, сбор статистики, ее обработку и вывод на печать в виде таблиц результатов моделирования. *Блок синхронизации* осуществляет контроль за протеканием параллельных процессов в моделируемой системе, выдает сигналы на реализацию событий в порядке очередности, согласно «списку будущих времен» их наступления. *Блок загрузки* производит настройку (загрузку) имитационного модуля на заданное место в структурной схеме модели, т. е. ставит в соответствие элементу моделируемой системы физический модуль имитации и осуществляет обращение к нему с указанием координат событий, требующих реализации. *Библиотека имитационных модулей* представляет собой набор подпрограмм имитации отдельных физических объектов — типовых подсистем АСОИ. *Библиотека служебных подпрограмм* включает набор процедур генерации потоков событий, имитации отдельных дисциплин обслуживания и т. п.

8.3. СОСТАВ ПАКЕТА И ПРИНЦИПЫ ПОСТРОЕНИЯ МОДУЛЕЙ ИМИТАЦИИ

Модули имитации, входящие в состав СММИФ, могут быть разделены условно на четыре группы. В первую входят модули, имитирующие генерацию потоков заявок различной структуры:

источники независимых ординарных потоков заявок, синхронизируемые источники (конечные источники заявок), источники, имитирующие групповое поступление заявок и т. п. Ко второй группе относят модули, имитирующие накопители заявок (очереди) с различными дисциплинами заполнения. Третья группа содержит модули, имитирующие различные дисциплины обслуживания заявок. К четвертой принадлежат модули, имитирующие процессы управления движением заявок в моделируемой системе. Существуют еще и некоторые модули специального назначения, не входящие в указанную классификацию. К таким относится, например, модуль, позволяющий изменять по заданному закону некоторые параметры модели в течение прогона, т. е. организовать нестационарный процесс обслуживания с заданными параметрами и структурой.

Все операции, реализуемые в стандартном модуле имитации, входящем в библиотеку пакета, можно разделить на три основные группы:

операции, осуществляющие анализ входных параметров настройки модуля на заданное место (элемент) в структуре модели и установку параметрической связи с остальными элементами (смежными). Последнее реализуется через фактические параметры обращения к подпрограмме и через блоки COMMON;

операции по заданию условий синхронизации на активизацию данного элемента по внешним входным управляющим сигналам и времени наступления очередного события;

операции, имитирующие непосредственно физические процессы, события, характерные для данного элемента моделируемой системы в данный момент времени.

Синхронизация параллельных дискретных процессов, протекающих в модели; осуществляется в соответствии с состояниями массивов $DEL(IL, IN)$, $WA(IL, IN)$, $EX(IL, IN)$, где IL — номер элемента модели, IN — номер процесса, дискретные состояния которого имитируют соответствующие физические события в IL -м элементе модели, $DEL(IL, IN)$ — момент наступления события в IL -м элементе, для IN -го процесса.

Так, например, для элемента модели-копии модуля $FLOW$ элемент $DEL(IL, IN)$ характеризует момент времени появления заявки потока, генерируемого с выходного контакта с номером IN .

Массивы WA , EX — логические. Если $WA(IL, IN) \equiv true$, то это означает, что в элементе IL процесс IN ждет внешнего управляющего сигнала для реализации своего очередного события. Если $EX(IL, IN) \equiv true$, то соответствующий сигнал послан и событие будет реализовано. Например, для элемента с номером IL копии модуля $BUFFER$ состояние $true$ для $WA(IL, 1)$ означает, что буферный накопитель ожидает приема заявки на

SUBROUTINE <имя модуля> (*, IL, IN)

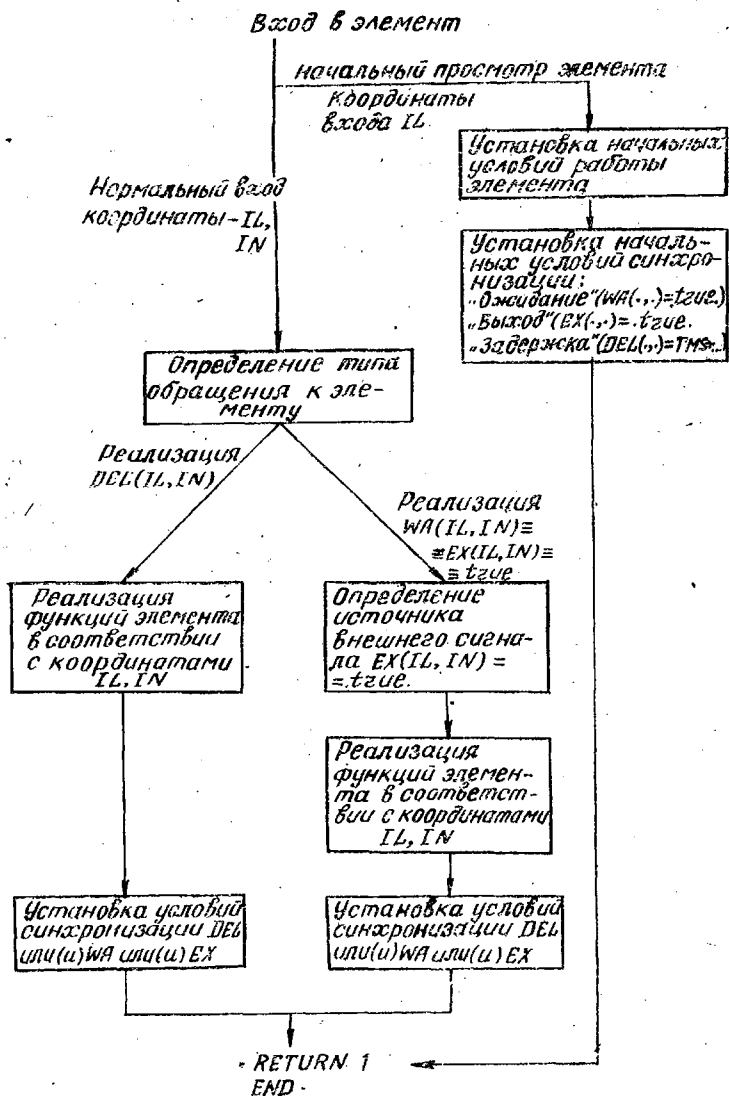


Рис. 8.2. Блок-схема типового модуля имитации

SUBROUTINE FLOW (*, IL, IN)

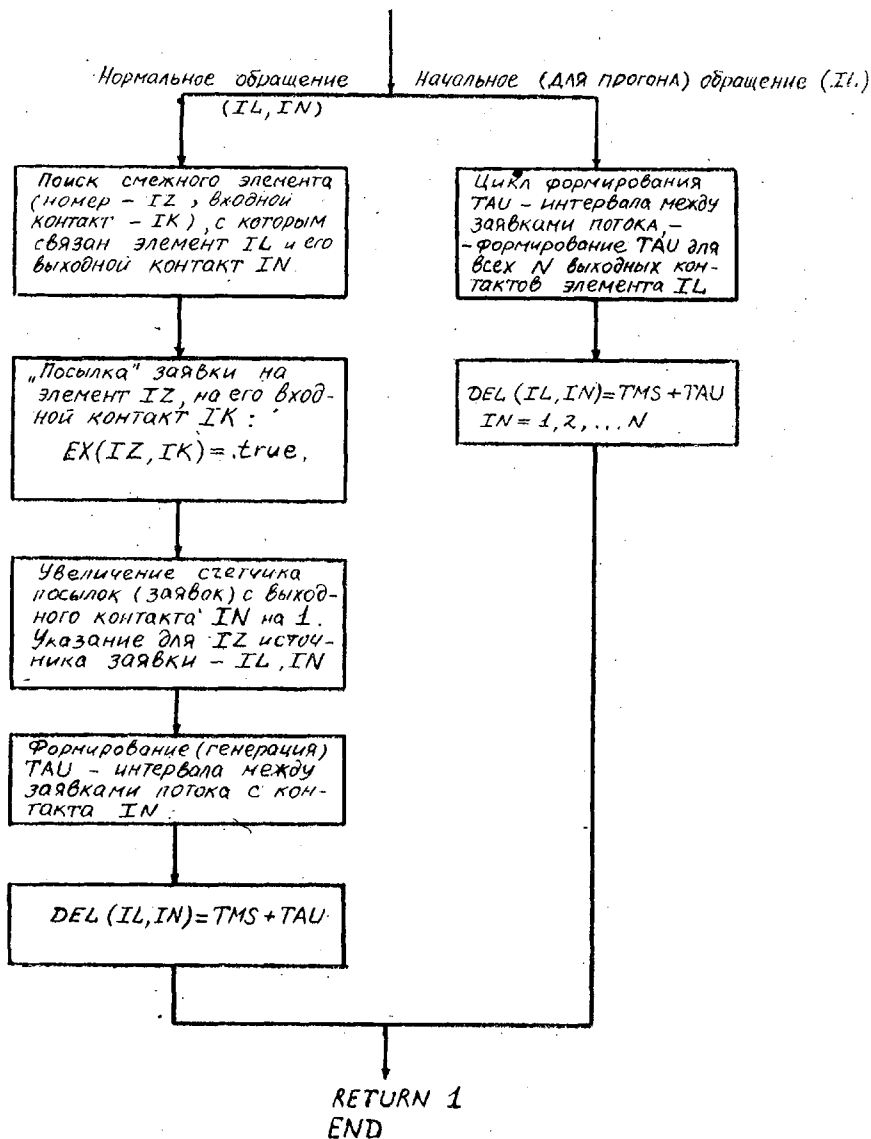


Рис 8.3. Блок-схема модуля FLOW

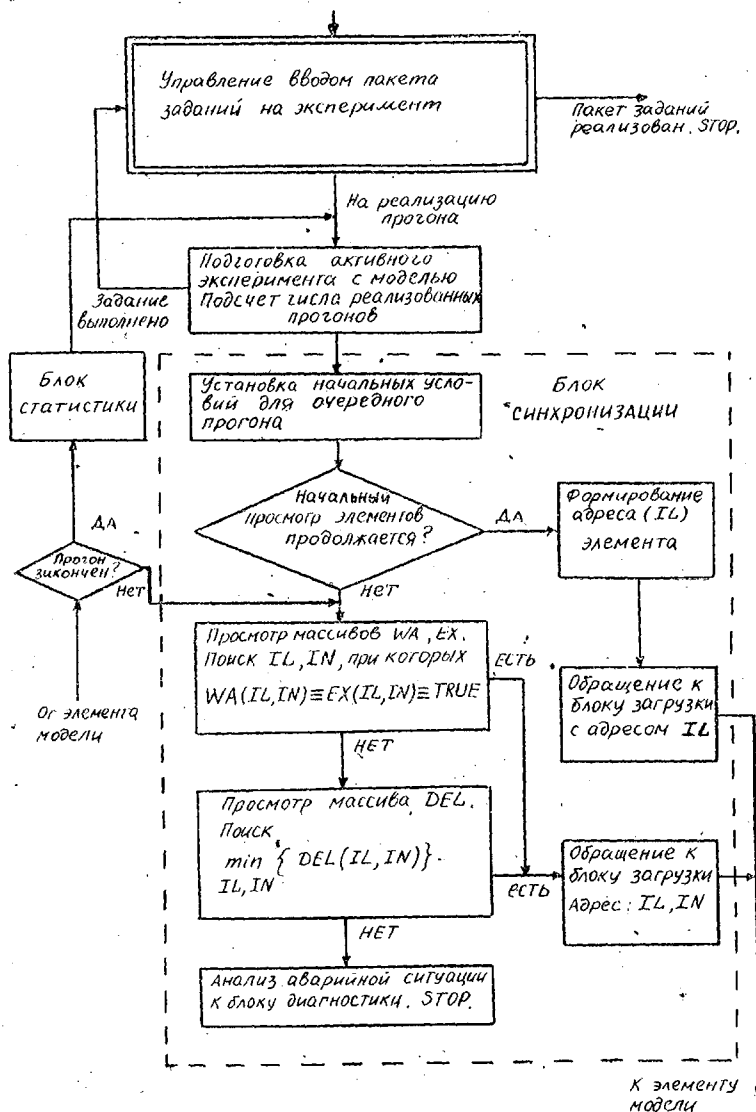


Рис. 8.4. Схема блока организации и синхронизации

свой первый входной контакт. Элемент-копия модуля *FLOW* может послать заявку с соответствующего выходного контакта в элемент *IL* путем операции присваивания $EX(IL,1) = true$. При $EX(IL,1) \equiv WA(IL,1) \equiv true$ заявка будет помещена в очередь, которая имитируется в элементе *BUFER* специальными массивами, где фиксируется время поступления заявки и индекс потока, которому она принадлежит.

Состояние массивов *DEL*, *WA*, *EX* анализируется в блоке синхронизации СММИФ. В результате находятся координаты ближайшего по времени события. Если событие найдено по состоянию массива *DEL*, то модельное время *TMS* становится равным значению соответствующего элемента массива *DEL*. После нахождения координат события *IL; IN* следует передача управления в точку *IN* элемента с индексом *IL*. Фактически управление передается модулю имитации (подпрограмме), код (адрес) которой находится по значению согласно списку *NABOR*. Возвращение из элемента модели в блок организации производится через блок управления прогоном, который следит за моментом его окончания.

Обобщенная блок-схема типового модуля имитаций приведена на рис. 8.2. На рис. 8.3 в качестве примера представлена блок-схема модуля имитации *FLOW* — генератора независимых, стационарных ординарных потоков, описываемых распределением Эрланга. Упрощенная схема блоков организации и синхронизации изображена на рис. 8.4.

Краткое описание наиболее типичных модулей имитации, входящих в состав СММИФ, дано в табл. 8.1.

Таблица 8.1

Описание модулей имитации

Имя модуля	Основные функции	Модификации
<i>FLOW</i>	Генерация независимых потоков Эрланга. Количество потоков определяется числом задействованных выходных контактов	
<i>BUFER</i>	Имитатор очереди заявок, буферного накопителя. Общая зона для хранения заявок произвольного числа потоков. Объем ограничен. Дисциплина выборки задается обслуживающим устройством.	

Имя модуля	Основные функции	Модификации
<i>PROCES</i>	Устройство обслуживания многоканальное. На входе произвольное число очередей. Каналы имеют равный доступ ко всем очередям. Дисциплина выборки заявок из очередей задается модификациями	1. Дисциплина «первый пришел — первый обслужен» (<i>FIFO</i>) среди всех очередей и заявок. 2. Относительные приоритеты
<i>PROCQU</i>	Одноканальное устройство обслуживания, реализующее режим разделения времени. На входе произвольное число очередей. Дисциплина выборки задается модификациями	См. модуль <i>PROCES</i>
<i>PROCAB</i>	Устройство обслуживания одноканальное с иерархической смешанной дисциплиной обслуживания, включая прерывания с дообслуживанием. На входе произвольное число очередей	
<i>DISPLY</i>	Генерация независимых синхронизируемых (конечных источников заявок). Время «обдумывания» распределено по закону Эрланга	
<i>SING</i>	Синхронизируемый групповой источник заявок	
<i>FILTER</i>	Фильтрация потока (прореживание)	1. Регулярное прореживание. 2. Случайное прореживание
<i>CORE</i>	Имитация временно занимаемого ресурса (оперативной памяти). На входе произвольное число очередей. Выборка заявок при свободной части ресурса задается модификациями	См. модуль <i>PROCES</i>

Имя модуля	Основные функции	Модификации
<i>SWITCH</i>	Узел коммутации (переключения), изменяющий направление движения заявок входных потоков по дисциплине, задаваемой модификацией	Имеется три дисциплины регулярного циклического и случайного выбора направления
<i>MULTI</i>	«Размножитель» заявок входного потока. Создание копии заявок и посылка их по направлениям, заданным коммутацией выходных контактов элемента	

8.4. ФОРМИРОВАНИЕ МОДЕЛИ И ОРГАНИЗАЦИЯ ЭКСПЕРИМЕНТА

Приведем основные практические приемы (без детализации), позволяющие осуществить в рамках СММИФ конструирование имитационной модели АСОИ и формирование программы модельного эксперимента. Последовательность действий разработчика имитационной модели такова:

- формирование структуры модели;
- параметрическое описание структуры модели;
- компоновка пакета заданий на эксперимент;
- реализация на ЭВМ.

Формирование структуры модели

На данном этапе производят конструирование структуры модели на основании анализа моделируемой системы и возможностей СММИФ. При графическом представлении структуры модели удобно использовать стандартное изображение элемента модели (рис. 8.5).

Элемент i модели содержит в общем случае k_i входных и N_i выходных контактов, $\{x_j\}_1^{k_i}$, $\{y_j\}_1^{N_i}$ — соответственное поле входных и выходных контактов. При параметрическом описании модели задается параметр $MW = \max\{k_i, N_i\}$.

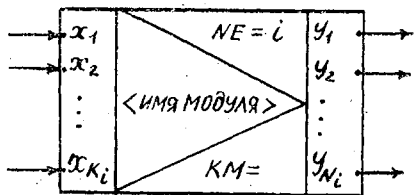


Рис. 8.5. Стандартное изображение элемента модели

Для каждого из модулей имитации, входящих в состав СММИФ, оговорена специализация входных и выходных контактов. Контакты обеспечивают связь элементов по потоку (передача измерений, сообщений, заявок и т. п.) и по управляющим воздействиям. В соответствии с логикой функционирования и структурой построения моделируемой системы контакты элементов модели соединены линиями — каналами сопряжения, по которым передача сообщений, управляющих сигналов осуществляется без потерь, искажений и задержек. Реальные каналы связи должны представляться самостоятельными элементами.

Ниже представлен обобщенный алгоритм формирования структуры модели:

1. Ознакомиться с каталогом и описанием модулей имитации, входящих в личную библиотеку пакета.

2. В соответствии с особенностями исследуемой системы, целями анализа ее функционирования произвести декомпозицию системы до уровня подсистем, для которых можно найти аналоги в виде модулей имитации.

3. Разработать структурную схему модели, для чего каждую подсистему графически представить в виде элемента, воспользовавшись стандартным способом его изображения.

4. Произвести графически соединения входных и выходных контактов элементов каналами сопряжения.

Параметрическое описание структуры модели

На данном этапе необходимо определить ряд параметров, которые описывают наиболее общие закономерности процессов, протекающих в имитационной модели. В частности, нужно провести нумерацию (индексацию) элементарных потоков в модели. *Элементарный поток* — это канал сопряжения, связывающий входной и выходной контакты двух элементов, по нему будет осуществляться передача сообщений (заявок) без потерь и задержек. Каждому элементарному потоку должен быть присвоен номер IP , который служит для выборки в ходе модельного эксперимента параметров генерации потока, параметров времени обслуживания; приоритета и т. п.

Маршрут движения заявок в модели задается массивом $MPR(\cdot)$. Элемент массива $MPR(IP)$ есть номер элементарного потока, в который «превращается» исходный поток с номером IP . Соответствие элементов модели модулям имитации, входящим в состав пакета, задается списком $NABOR(\cdot)$. Элемент массива $NABOR(NE)$ представляет собой код модуля имитации — КМ для элемента с номером NE .

Структура связи между элементами модели задается с помощью массивов $IR(I, J)$, $IM(I, J)$. Элемент массива $IR(I, J)$ со-

держит номер элемента, который связан по потоку или по управляющему сигналу с I -м элементом и его J -м выходным контактом. Элемент массива $IM(I, J)$ является номером входного контакта элемента $\langle IR(I, J) \rangle$, с которым связан элемент и его выходной контакт J .

«Привязка» элементарных потоков к выходным контактам элементов модели осуществляется путем формирования массива $NPT(\dots)$. Элемент массива $NPT(I, J)$ является номером элементарного потока, выходящего из выходного контакта J элемента с номером I .

Заполнение элементов массивов IM, IR, MPR, NPT происходит при вводе исходных данных по каждому из элементов, входящих в модель. Включение в состав модели таких элементов, как $MULTI, SWITCH, DYNP$ позволяет изменять содержимое указанных массивов в течение времени прогона, т. е. делает возможной динамическую корректировку алгоритма маршрутизации и структуры модели.

Ввод исходных данных и управление модельным экспериментом

При исследовании имитационной модели с помощью СММИФ общение пользователя с ЭВМ происходит в пакетном режиме. Для загрузки СММИФ в ОЗУ ЭВМ и активизации необходимо ввести исходные данные в виде набора управляющих операторов ОС ЕС ЭВМ и набора данных, описывающих модель и условия модельного эксперимента. Структура входного потока данных на перфокартах при работе со СММИФ представлена на рис. 8.6.

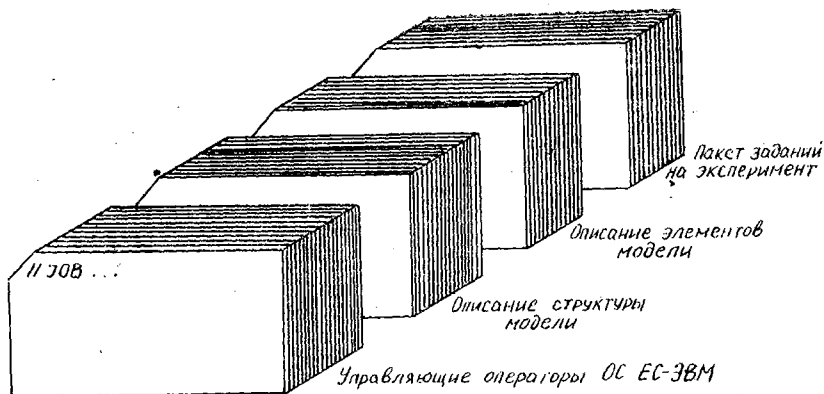


Рис. 8.6. Структура входного потока исходных данных для моделирования.

Входной поток исходных данных можно разделить на три основные части:

- описание структуры модели (ОС);
- описание элементов модели (ОЭ);
- пакет заданий на эксперимент (ПЭ).

Первая часть содержит данные, полученные на этапе формирования структуры модели и ее параметрического описания. Вторая часть включает данные по описанию внешних связей и внутренних параметров элементов модели. Конкретный перечень набора данных по каждому из элементов зависит от стандартного модуля имитации, копией которого является элемент. (Этот перечень особо оговаривается в инструкции для пользователя СММИФ).

Настоящая версия СММИФ допускает разработку моделей следующей максимальной размерности. Число элементов — 50, число потоков — 100, параметр модели MW — 20. При этом, однако, практически не ограничивается число одновременно циркулирующих в модели заявок или число заявок, которые проходят через модель за время прогона. Следует учесть также, что равенство $MW=20$ означает, что, например, один элемент-копия модуля *FLOW* может одновременно генерировать до 20 независимых внешних потоков, а элемент-копия модуля *DISPLY* — имитировать работу одновременно до 20 терминальных устройств в режиме «запрос—ответ» с обдумыванием запроса и т. п.

Формирование пакета заданий на эксперимент

Следующей группой числового материала, располагаемого на перфокартах, является пакет заданий на работу (эксперименты) с моделью. Различают три типа заданий: по критерию окончания прогона; на моделирование; по выводу на печать результатов моделирования.

Пакет заданий имеет в начале и конце признак — соответствующее значение идентификатора *KNS*: начинается пакет признаком $KNS = 11$; заканчивается признаком $KNS = 12$. Пакет может содержать произвольное количество заданий. Если в пакете несколько заданий, то каждое из них начинается признаком $KNS = 11$, а в конце последнего ставится признак $KNS = 12$, что определяет окончание работы с моделью.

Помимо признака начала задания *KNS*, каждое задание имеет в начале определяющий его тип признак *KTS*. Причем $KTS = 10$ соответствует заданию по критерию окончания прогона, $KTS = 11$ определяет задание на моделирование, $KTS = 12$ определяет задание по выводу результатов моделирования на печать.

Задание по критерию окончания прогона. Прогон, или реализация, — это развернутый во времени процесс поступления, накопления, обслуживания заявок в системе. В течение прогона происходит сбор и обработка статистики относительно характеристик функционирования моделируемой системы при фиксированных значениях ее параметров. Предполагается эргодичность процессов обслуживания, в связи с чем исследование по множеству заменяют исследованием случайных процессов по одной достаточно длинной временной реализации. От длительности реализации зависит точность статистических оценок, полученных в результате моделирования. Критерием окончания прогона может являться либо число испытаний N , под которым мы будем далее понимать число заявок определенного генерируемого потока, прошедших через моделируемую систему, либо некоторое значение интервала времени, прошедшего с момента начала моделирования на данном прогоне от $TMS = 0,0$ до момента времени $TMS = TFIN$, где TMS — момент времени по часам модели. Масштаб модельного времени зависит от размерности задаваемых параметров модели: Размерность времени может быть произвольной, но одинаковой для всех задаваемых параметров. Так, если мы задаем интенсивность некоторого потока как $\lambda [1/c]$, то параметры обслуживания заявок потока должны быть выражены в секундах. Тогда и TMS будет иметь такую же размерность. TMS имеет тип *REAL*.

Из вышесказанного следует, что сигнал на окончание прогона может вырабатываться по двум критериям: или по истечении определенного времени, или по достижении количества заявок одного либо нескольких потоков некоторого заданного числа.

Задание данного типа оформляют в виде следующего набора перфокарт:

1. В начале набора находится перфокарта с числом KTK , которое определяет вид критерия окончания прогона. Если необходимо заканчивать прогон, сравнивая текущее модельное время с заданным, то $KTK = 10$; если же прогон заканчивается, сообразуясь с количеством заявок, то $KTK = 11$.

2. Если $KTK = 10$, то далее следует п/к с выбранным значением $TFIN$.

3. Если $KTK = 11$, то вслед за п/к с KTK следует п/к с числами $NFIN$ и NG , где NG — количество элементарных потоков, заявки которых подсчитывают для определения момента окончания прогона; $NFIN$ — заданное число заявок (испытаний). В течение прогона подсчитывают число заявок потоков. Когда минимальное из них достигает значения $NFIN$, прогон заканчивается.

4. Вслед за перфокартой с числами $NFIN$ и NG вводят массивом номера потоков, используемых для определения конца прогона. Количество номеров потоков равно NG . Номера потоков следует располагать обязательно в порядке возрастания. Число таких потоков NG может колебаться в пределах 1...10.

Задание на моделирование. В случае, если идентификатор признака типа задания $KTS = 11$, это означает, что далее должны располагаться перфокарты, определяющие данные для эксперимента с моделью — собственно *моделирования*. Задание на моделирование — это, по сути дела, корректировка исходных данных одних параметров модели и задание серии прогонов (реализаций) при изменении какого-то одного параметра модели-аргумента. Результат моделирования на одном прогоне — это некоторая точка на графике, отражающем зависимость некоторого показателя (характеристики) системы от выбранного аргумента, варьируемого параметра.

Таким образом, после введения карты признака типа задания вводится карта конца пунктов задания на моделирование: $KPSM = 11$ — свидетельствует о том, что в задании есть еще пункты (не выполненные ранее);

$KPSM = 12$ — свидетельствует о том, что в задании уже нет пунктов, т. е. все пункты выполнены.

Если был введен признак $KPSM = 12$, то управление переходит на начало работы СММИФ по полученному заданию на моделирование. Если был введен признак $KPSM = 11$, то необходимо ввести карту признака кода (номера) параметра KTI модели, для которого задаются значения.

Затем нужно ввести карту с признаком активности пункта задания KAW . Дело в том, что пункты задания могут быть либо активными, либо пассивными. *Активным пунктом* задания считается такой, в котором параметр-аргумент будет принимать несколько значений; *пассивным пунктом* задания — такой, в котором параметр системы будет иметь только одно значение (т. е. параметр будет «заморожен»).

Если пункт задания активный, то $KAW = 1$, если пункт задания пассивный, то $KAW = 0$. Но необходимо учитывать, что при формировании задания в нем (и каждом задании на моделирование) должен быть только один активный пункт.

Задание по выводу на печать результатов моделирования. В рамках СММИФ предусмотрено два типа вывода на печать результатов моделирования. Первый определяется включением в состав модели элемента — копии модуля SS , который принудительно после каждого прогона дает распечатку, содержащую распределение (гистограмму) вероятностей значения времени задержки заявок или (и) интервалов

между заявками исследуемых потоков, длины очередей в накопителях после прогона, число заявок, прошедших через модель каждого из потоков. Второй тип — табличный выборочный метод вывода результатов, позволяющий формировать таблицы с заданной номенклатурой показателей качества функционирования АСОИ.

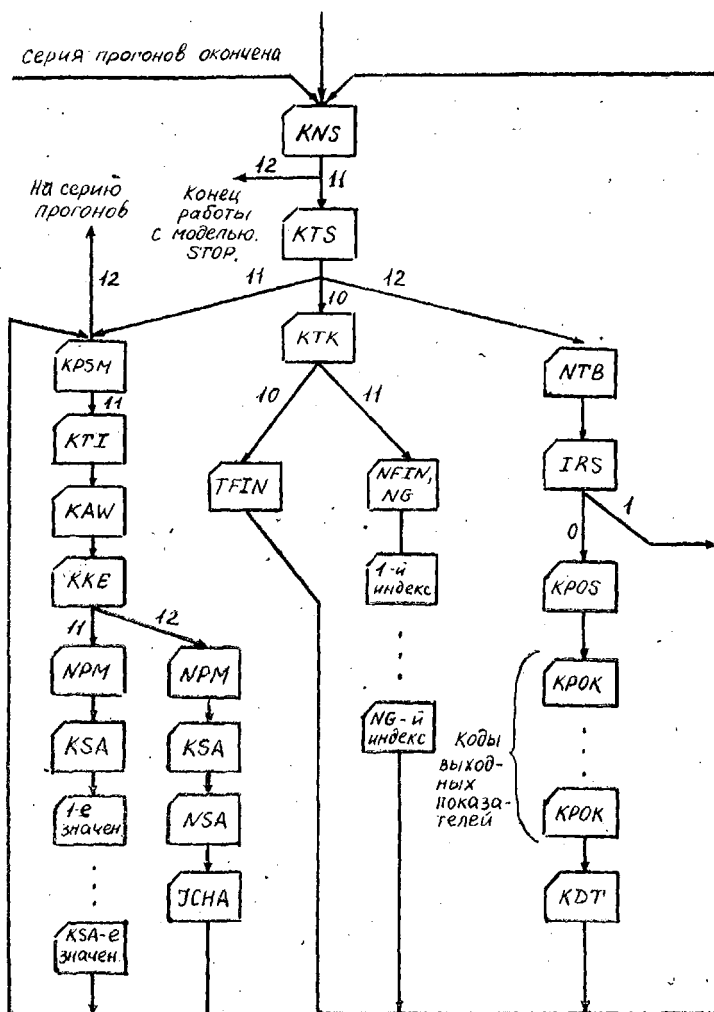


Рис. 8.7. Схема формирования пакета заданий на эксперимент

Для табличного выборочного вывода на печать значений показателей, характеризующих качество обслуживания потоков и функционирования элементов (среднее время реакции, вероятность потерь и т. п.), в зависимости от заданного аргумента необходимо сформировать следующую последовательность данных:

1. $KTS = 12$.
2. NTB — номер таблицы.
3. IRS — признак повторения формата таблицы.
4. $KPOS$ — количество показателей, выводимых в данном формате таблицы.
5. Последовательность п/к (общее количество — $KPOS$) с кодом задаваемого показателя ($KPOK$).
6. Количество копий данной таблицы (KDT).

Если задание на печать первое, то $IRS = 0$ и вводятся все перечисленные выше параметры. При повторном выводе того же набора показателей можно сократить объем вводимых данных, задав $IRS = 1$. Тогда остальные позиции не вводятся. На печать будет выведена таблица с номером NTB по формату, т. е. с набором показателей, определенных в первом предшествующем задании, где $IRS = 0$.

Общая блок-схема алгоритма формирования пакета заданий представлена на рис. 8.7. Учитывая специфику описанных выше типов заданий, следует отметить, что первым заданием в пакете должно быть задание на моделирование, далее — задание по выводу результатов моделирования на печать (хотя это и необязательно). В дальнейшем порядок следования заданий в пакете произвольный, обусловленный лишь целями исследований. Так, например, задание по окончанию прогона может быть единственным. Но одно такое задание в пакете должно быть обязательно.

8.5. ПРИМЕР ПОСТРОЕНИЯ ИМИТАЦИОННОЙ МОДЕЛИ

Рассмотрим систему оперативной обработки информации (рис. 8.8). На вход процессора через буферное запоминающее устройство (буфер) поступают заявки трех потоков — z_1, z_2, z_3 . Потоки z_1, z_2 — внешние, являются ординарными, независимыми и стационарными. Известно, что z_1 — регулярный поток, интенсивность которого может принимать значения в диапазоне от 60,5 до 120,2 1/с. Поток z_2 — случайный, закон распределения интервалов между заявками у которого аппроксимируется распределением Эрланга с коэффициентом, равным 3. Интенсивность потока $\lambda_{z2} = 65,8$ 1/с.

После обработки в процессоре заявки потоков z_1 и z_2 сум-

мируются, образуя новый поток z_3 . Заявки этого потока вновь через буфер поступают в процессор. Обработанные заявки потока образуют выходной поток z_4 .

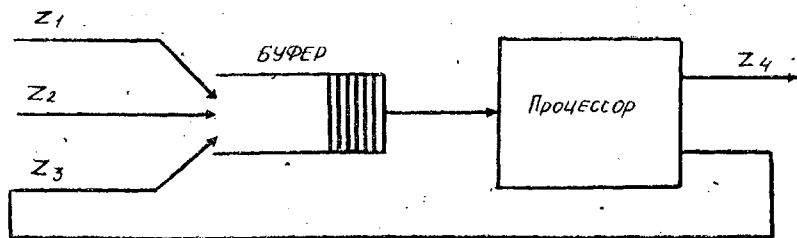


Рис. 8.8. Структура АСОИ

Предположим, что объем буфера равен 50. Заполнение буфера беспriorитетное на свободное место. В случае занятости буфера происходит потеря заявки. Процессор осуществляет приоритетное обслуживание заявок, причем наивысший приоритет имеют заявки потока z_1 , которые прерывают обработку заявок потоков z_2, z_3 . Обслуживание заявок потоков z_2, z_3 осуществляется с учетом относительных приоритетов, причем заявки потока z_3 имеют более высокий приоритет.

Пусть установлено, что трудоемкости алгоритмов A_1, A_2 , с помощью которых производится обработка заявок поток z_1, z_2 , являются независимыми случайными величинами. Причем, трудоемкость алгоритма A_1 хорошо аппроксимируется экспоненциальным распределением, а для A_2 — распределением Эрланга с параметром, равным 5. Известны соответствующие средние трудоемкости Θ_1 и Θ_2 . Время обработки заявок потока z_3 — величина постоянная. Известна трудоемкость соответствующего алгоритма — Θ_3 .

Пусть $\Theta_1 = 100$ опер., $\Theta_2 = 150$ опер., $\Theta_3 = 75$ опер., а быстродействие процессора $B = 60$ тыс. опер./с. Тогда соответствующие интенсивности обработки потоков будут следующими: $\mu_1 = 600$ 1/с, $\mu_2 = 400$ 1/с, $\mu_3 = 800$ 1/с.

Предположим, что требуется провести анализ функционирования системы в зависимости от значений интенсивности входного потока z_1 . В частности, поставлена задача оценки среднего и дисперсии времени задержки в системе заявок каждого потока, а также максимальной длины очереди в течение прогона. Несмотря на небольшую размерность исследуемой системы, параметры потоков, характеристики алгоритмов обработки и дисциплина обслуживания затрудняют применение для этих целей аналитических методов.

Имитационную модель, достаточно точно отражающую характер функционирования системы, можно построить, используя средства СММИФ (рис. 8.9). Для этого достаточно воспользоваться тремя модулями имитации — *FLOW*, *BUFER*, *PROCAB*.

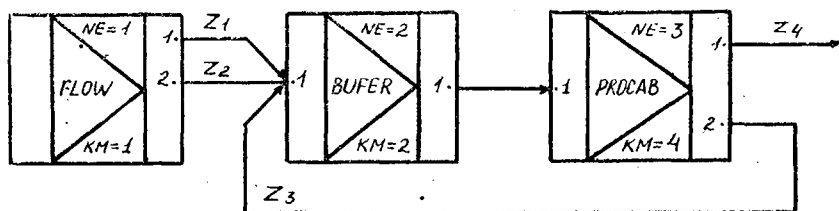


Рис. 8.9. Структура имитационной модели АСОИ

Элемент модели — копия модуля *FLOW* — будет имитировать поступление заявок потоков z_1 , z_2 ; элемент — копия модуля *BUFER* — имитатор буферного запоминающего устройства; имитацию работы процессора можно осуществить с помощью элемента — копии модуля *PROCAB*.

Значения выходных показателей функционирования модулируемой системы для 3-х прогонов указаны в табл. 8.2. Каждый прогон характеризуется интенсивностью входного потока z_1 — аргумента модельного эксперимента. Длительность прогона распределялась предельным числом сформированных заявок потока z_1 (1000). Время реализации данного модельного эксперимента на ЭВМ ЕС-1033 около 3-х минут.

Таблица 8.2

Результаты имитационного эксперимента

Интенсивность входного потока	Среднее время задержки заявок i -го потока			Дисперсия времени задержки заявок i -го потока			Вероятность потери заявок P_3	Максимальный размер очереди Q_2
	λz_1	τ_1	τ_2	τ_3	D_1	D_2		
60,5	$0,16 \cdot 10^{-2}$	$0,38 \cdot 10^{-2}$	$0,19 \cdot 10^{-2}$	$0,29 \cdot 10^{-5}$	$0,47 \cdot 10^{-5}$	$0,12 \cdot 10^{-5}$	0	4
80,5	$0,17 \cdot 10^{-2}$	$0,40 \cdot 10^{-2}$	$0,20 \cdot 10^{-2}$	$0,31 \cdot 10^{-5}$	$0,51 \cdot 10^{-5}$	$0,17 \cdot 10^{-5}$	0	3
120,2	$0,16 \cdot 10^{-2}$	$0,48 \cdot 10^{-2}$	$0,22 \cdot 10^{-2}$	$0,28 \cdot 10^{-5}$	$0,10 \cdot 10^{-5}$	$0,21 \cdot 10^{-5}$	0	3

ЛИТЕРАТУРА

1. Материалы XXVI съезда КПСС. — М.: Политиздат, 1980.
2. Максимов Н. А., Тучкин А. В., Ушакова З. Н. Оценка характеристик и выбор вычислительных средств для АСУП. — В сб.: Прогрессивная технология приборостроения: Межвуз. сб. научных трудов. — ВЗМИ. М., 1980.
3. Кравец В. Г., Любимский В. Е. Основы управления космическими полетами. — М.: Машиностроение, 1983. — 224 с.
4. Karp R. M. A note on the application of graph theory to digital computer programming. — Information and Control, 1960, v. 13, № 2, pp. 179—190.
5. Основы теории вычислительных сетей/Под ред. С. А. Майорова. — М.: Высшая школа, 1978.
6. Липаев В. В., Яшков С. Ф. Эффективность организации вычислительного процесса в АСУ. — М.: Статистика, 1975.
7. Скурихин В. И. Принципы организации и исследования некоторых классов автоматизированных систем управления и обработки данных: Автореферат диссертации на соискание степени доктора технических наук. — Киев, 1970.
8. Вентцель Е. С. Исследование операций. — М.: Сов. радио, 1972.
9. Клейнрок Л. Вычислительные системы с очередями. — М.: Мир, 1979.
10. Максимов Н. А. Об одном алгоритме оптимизации загрузки средств сети информационно-телеметрического комплекса. — В сб.: Информационное обеспечение испытаний летательных аппаратов: Тематич. сб. трудов МАИ. — М., 1980.
11. Jackson J. R. Networks of Waiting Lines. — Operation Research, 1975, v. 5, pp. 518—521.
12. Беллман Р. Динамическое программирование. Пер. с англ. — М.: Иностранная литература, 1960.
13. Снапелев Ю. М., Старосельский В. А. Моделирование и управление в сложных системах. — М.: Сов. радио, 1974.
14. Химмельблау Д. Прикладное нелинейное программирование. Пер. с англ. — М.: Мир, 1975.
15. Вентцель Е. С. Теория вероятностей. — М.: Наука, 1969.
16. Хан Г., Шапиро С. Статистические модели в инженерных задачах. — М.: Мир, 1969.
17. Овчаров Л. А. Прикладные задачи теории массового обслуживания. — М.: Машиностроение, 1969.
18. Липаев В. В. Распределение ресурсов в вычислительных системах. — М.: Статистика, 1979.
19. Abd-alla A. M., Lichtenberg J. On a balanced multiprocessing computer utility. — Computer-communications networks and teletraffic, 1972.
20. Саати Т. Л. Элементы теории массового обслуживания и ее приложения. Пер. с англ. — М.: Сов. радио, 1971.
21. Мартин Дж. Системный анализ передачи данных. Пер. с англ. — М.: Мир, 1975, т. 2.
22. Липаев В. В., Колин К. К., Серебровский Л. А. Математическое обеспечение управляющих ЦВМ. — М.: Сов. радио, 1972.
23. Avi-Itzhak B., Maxwell W. L., Miller L. W., Queueing with alternating priorities. — Operation Research, 1965, v. 13, № 2.
24. Колин К. К., Липаев В. В. Проектирование алгоритмов управляющих ЦВМ. — М.: Сов. радио, 1970.
25. Клейнрок Л. Коммуникационные сети. — М.: Наука, 1970.
26. Мова В. В., Пономаренко Л. А., Калиновский А. М. Организация приоритетного обслуживания в АСУ. — Киев: Техника, 1977.

27. *Соболевский М. И.* Анализ и оптимизация структур матричных вычислительных систем. — М.: Энергия, 1979.
28. *Ковалев М. И.* Дискретная оптимизация (целочисленное программирование). — Минск: БГУ, 1977.
29. *Раков Г. К.* Методы оптимизации структур вычислительных систем. — М.: Энергия, 1974.
30. *Голубков А. С.* Аналитические модели и оценка характеристик систем сбора и обработки информации. — М.: МАИ, 1980.
31. *Аркадьев А. Г., Браверман Э. М.* Обучение машины классификации объектов. — М.: Наука, 1971.
32. *Бусленко Н. П.* Моделирование сложных систем. — М.: Наука, 1968.
33. *Гилберт Е. Н., Поллак Г. О.* Минимальные деревья Штейнера. — В кн.: Кибернетический сб., Новая серия. — М.: Мир, 1974, вып. 8, с. 19—51.
34. *Шеннон Р.* Имитационное моделирование систем — искусство и наука. — М.: Мир, 1978.
35. *Seaman P. H., Soucy R. C.* Simulating operating systems. — IBM System Journal, 1969, № 4.
36. *Шрайбер Т. Дж.* Моделирование на GPSS. Пер. с англ. — М.: Машиностроение, 1980.
37. *Бусленко В. Н.* Автоматизация имитационного моделирования сложных систем. — М.: Наука, 1977.
38. *Шабанов Л. В., Шахин В. Н.* Комплекс МОДЕЛЬ-2 для моделирования сетей передачи данных с коммутацией сообщений. — М.: МЭИС, 1979.
39. *Tsutomu Shino.* A new Traffic Simulator for network systems-SONET. 1974. Winter simulation congress. N. Y. 1974.
40. *Богомолов Л. П., Гайстеров С. Ф., Ермилов В. А., Свирский И. П.* Имитационное моделирование вычислительных систем. — В кн.: Моделирование сложных систем. Вып. 4. — Рига: Зинатне, 1975.
41. Летные испытания ракет и космических ЛА./Под ред. *Е. И. Кринецкого.* — М.: Машиностроение, 1980.
42. *Виттих В. А., Куклин Г. Н., Томников Г. Н., Цыбагов В. А.* Оптимизация структур систем сбора экспериментальных данных. — Автоматика и вычислительная техника, 1978, № 2, с. 64—71.
43. *Виттих В. А., Цыбагов В. А.* Оптимизация бортовых систем сбора и обработки экспериментальных данных. — М.: Наука, 1985.
44. *Шароватов А. В.* Принципы построения пакета прикладных программ моделирования информационных систем. — В сб.: Информационное обеспечение испытаний летательных аппаратов. — М.: МАИ, 1980.
45. *Виттих В. А., Куклин Г. Н., Томников Г. Н., Цыбагов В. А.* Топологическая оптимизация систем сбора информации. — Проблемы управления и теории информации. — Будапешт: АН Венгрии, 1978, т. 7, № 1.
46. *Прим Р. К.* Кратчайшие связывающие сети и некоторые обобщения. — В кн.: Кибернетический сб. — М.: Иностранная литература, 1961, № 2, с. 95—107.
47. *Дал У., Мюрхауг Б., Ньюгорд К.* СИМУЛА-67. — М.: Мир, 1969.
48. *Десярев Ю. И.* Методы оптимизации. — М.: Сов. радио, 1980.

ОГЛАВЛЕНИЕ

Введение	3
<i>Глава 1.</i> Планирование информационного взаимодействия летательных аппаратов (ЛА) с измерительными пунктами	9
1.1. Задача распределения вычислительных ресурсов измерительных пунктов на множестве ЛА	9
1.2. Оптимальное планирование взаимодействия группы ЛА с измерительным пунктом	17
<i>Глава 2.</i> Топологическая оптимизация систем сбора, передачи и обработки информации	22
2.1. Задачи топологической оптимизации	22
2.2. Модель топологической схемы ССПОИ	23
2.3. Общая постановка задачи	25
2.4. Безусловное оптимальное размещение	27
2.5. Размещение при пространственных ограничениях	36
2.6. Алгоритмы оптимального размещения недревовидных систем	41
2.7. Построение начальной топологии сети связи	49
2.8. Синтез топологической схемы ССПОИ	51
<i>Глава 3.</i> Выбор и оптимизация параметров сети передачи данных	55
3.1. Сети передачи данных. Основные понятия	55
3.2. Модель сети передачи данных. Задачи проектирования	57
3.3. Выбор пропускных способностей каналов связи	60
3.4. Оптимальное распределение потоков в сети передачи данных	72
<i>Глава 4.</i> Определение параметров многоканальной многофазной системы обработки информации в режиме разделения времени	78
4.1. Модель системы	78
4.2. Методика оценки характеристик и выбор параметров системы	81
<i>Глава 5.</i> Определение параметров автоматизированной системы обработки информации (АСОИ)	87
5.1. Постановка задачи оптимизации параметров	87
5.2. Определение минимально допустимого быстродействия процессора	90
5.3. Выбор дисциплины обслуживания	91
5.4. Выбор оптимальных параметров	94
5.5. Пример решения задачи выбора параметров системы	102
<i>Глава 6.</i> Аналитическое моделирование АСОИ с использованием аппарата стохастических сетей массового обслуживания	107
6.1. Представление модели АСОИ в виде стохастической сети	107
6.2. Принципы построения пакета прикладных программ «Сеть» и порядок работы с ними	113
6.3. Пример использования пакета программ «Сеть» для анализа работы измерительного пункта	116
<i>Глава 7.</i> Выбор комплекса технических средств АСОИ	121
7.1. Введение в проблему	121

7.2. Решение задачи выбора по критерию минимума среднего времени ответа	121
7.3. Эвристический метод синтеза оптимальной структуры АСОИ и распределения по ее элементам информационного потока	130
7.4. Выбор числа однородных вычислительных модулей и комплексирование задач обработки информации	142
<i>Глава 8. Имитационное моделирование АСОИ</i>	145
8.1. Средства моделирования и модели	145
8.2. Принципы построения и структура системы макромодульной имитации функционирования АСОИ	146
8.3. Состав пакета и принципы построения модулей имитации	148
8.4. Формирование модели и организация эксперимента	155
8.5. Пример построения имитационной модели	162

Доп. план, 1984, поз. 45

*Аркадий Семенович Голубков,
Владимир Андреевич Витих,
Николай Анатольевич Максимов,
Владимир Андреевич Цыбагов,
Анатолий Васильевич Шароватов*

**АНАЛИЗ ХАРАКТЕРИСТИК
И ВЫБОР ПАРАМЕТРОВ СЛОЖНЫХ
ТЕРРИТОРИАЛЬНО РАСПРЕДЕЛЕННЫХ
СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ**

Редактор Е. Д. Антонова
Техн. редактор Н. М. Каленюк
Корректор В. П. Петрова

Сдано в набор 11.12.84 г. Подписано в печать
ЕО 00357. Формат 60x84 1/16. Бумага оберточная.
Высокая печать. Литературная гарнитура.
Усл. п. л. 9,8. Уч.-изд. л. 9,98. Т. 1000 экз.
Заказ 1047. Цена 35 к.

Куйбышевский ордена Трудового Красного Знамени
авиационный институт имени академика С. П. Королева,
г. Куйбышев, ул. Молодогвардейская, 151.

Типография УЭЗ КуАИ, г. Куйбышев, ул. Ульяновская, 18.