

Министерство образования Республики Беларусь

Учреждение образования
«Полоцкий государственный университет»

О. Е. Шестопалова

СИСТЕМОТЕХНИКА

Учебно-методический комплекс
для студентов специальности
1-40 02 01 «Вычислительные машины, системы и сети»

Новополоцк
ПГУ
2011

УДК 004.9(075.8)
ББК 32.97я73
Ш52

Рекомендовано к изданию методической комиссией
факультета информационных технологий
в качестве учебно-методического комплекса
(протокол № 6 от 03.06.2011)

РЕЦЕНЗЕНТЫ:

зам. начальника отдела АСУ ОАО «Нафтан» В. В. ДЮЖЕВ;
зав. каф. технологий программирования С. В. КУХТА

Шестопалова, О. Е.

Ш52 Системотехника: учеб.-метод. комплекс для студентов специальности 1-40 02 01 «Вычислительные машины, системы и сети» / О. Е. Шестопалова. – Новополоцк: ПГУ, 2011. – 296 с.
ISBN 978-985-531-281-0.

Приведены введения в дисциплину «Системотехника» для студентов дневной и заочной форм обучения, регламентирующие состав и содержание всех видов учебных активностей, методы и формы контроля. Представлен теоретический материал дисциплины, изучаемый с целью освоения методологии общей теории систем и системного анализа в рамках квалификационных требований, предъявляемых к инженеру-системотехнику по специальности 1-40 02 01 «Вычислительные машины, системы и сети». Для приобретения практических навыков проектирования и исследования технических систем приведены индивидуальные задания, методические рекомендации к выполнению и примеры выполнения лабораторных работ. Рассмотрены примеры аттестационных заданий, вопросы к экзамену и типы экзаменационных задач. Дан перечень и характеристика дополнительных электронных материалов, выдаваемых в комплекте с УМК.

Предназначен для студентов специальности 1-40 02 01 «Вычислительные машины, системы и сети».

УДК 004.9(075.8)
ББК 32.97я73

ISBN 978-985-531-281-0

© Шестопалова О. Е., 2011
© УО «Полоцкий государственный университет», 2011

Содержание

Введение в дисциплину «Системотехника»	6
Дневная форма обучения	6
Заочная форма обучения.....	9
ЛЕКЦИОННЫЙ КУРС	12
1. Основы системотехники	12
1.1. Основные понятия системотехники.....	14
1.2. Структура и иерархия систем.....	19
1.3. Принципы системного управления	23
1.4. Системный подход.....	30
2. Методология системного анализа	35
2.1. Моделирование систем	36
2.1.1. Элементы теории множеств.....	37
2.1.2. Типовые математические схемы моделирования систем	41
2.2. Поиск и принятие решений.....	43
2.2.1. Общая характеристика методов	51
2.2.2. Системотехнические задачи исследования операций.....	53
2.3. Элементы теории графов	62
2.3.1. Основные понятия и способы задания графов	63
2.3.2. Создание и анализ топологических моделей систем	69
2.3.3. Использование теории графов в задачах исследования операций	86
2.4. Дискретно-детерминированное моделирование систем абстрактными конечными автоматами	98
2.4.1. Задание конечных автоматов в терминах теории множеств	98
2.4.2. Задание конечных автоматов в терминах теории графов	99
2.4.3. Моделирование дискретных систем конечными автоматами	102
2.5. Непрерывно-дискретное моделирование систем сетями Петри	106
ЛАБОРАТОРНЫЙ ПРАКТИКУМ	115
Лабораторная работа № 1	
Агрегирование и эквивалентные преобразования непрерывно-детерминированных моделей систем	115
Теория	115
Ход выполнения работы	115
Практические рекомендации по выполнению работы и использованию программного обеспечения	117
Основные принципы работы с пакетом VisSim	117
Пример 1. Моделирование источников внешних воздействий кусочно-линейного вида	119
Пример 2. Решение задач агрегирования линейных систем	123
Пример 3. Моделирование источников внешних воздействий нелинейного вида	127
Пример 4. Решение задач эквивалентных преобразований разомкнутых линейных систем	131
Варианты индивидуальных заданий	138
Лабораторная работа № 2	140
Управление системными ресурсами.....	150
Теория.....	150

Ход выполнения работы	151
Практические рекомендации по выполнению работы и использованию программного обеспечения.....	152
Пример 1. Решение систем линейных уравнений в MS Excel и сохранение модели (сценария) поиска решения	152
Пример 2. Решение задачи распределение ресурсов системы при заданном приоритете выполнения определенных видов работ	156
Пример 3. Решение задачи распределение ресурсов системы при наличии дополнительных условий и ограничений распределения	161
Пример 4. Использование формул массива для выполнения операций над матрицами коэффициентов в MS Excel.....	166
Варианты индивидуальных заданий	169
Лабораторная работа № 3	
Декомпозиция информационных систем с использованием теории графов.....	181
Теория.....	181
Ход выполнения работы	182
Практические рекомендации по выполнению работы и использованию программного обеспечения.....	182
Пример 1. Построение матриц графа структуры ИС, необходимых для решения задачи формальной декомпозиции	183
Пример 2. Решение задачи формальной декомпозиции для структуры ИС	187
Пример 3. Решение задачи формальной декомпозиции для произвольного графа	189
Пример 4. Расчет и анализ структурных свойств систем	191
Варианты индивидуальных заданий	196
Лабораторная работа № 4	
Анализ структурно-топологических характеристик топологий вычислительных сетей	199
Теория.....	199
Ход выполнения работы	202
Практические рекомендации по выполнению работы и использованию программного обеспечения.....	202
Пример 1. Расчет и анализ СТХ топологий ВС.....	202
Пример 2. Решение задачи оптимизации топологии	204
Варианты индивидуальных заданий	206
Лабораторная работа № 5	
Моделирование системной динамики сетями Петри	211
Теория	211
Ход выполнения работы	211
Практические рекомендации по выполнению работы и использованию программного обеспечения.....	212
Основные принципы работы с пакетом PetriNets InProject	212
Пример 1. Моделирование динамики функционирования системы с параллельным осуществлением процессов.....	219
Пример 2. Моделирование динамики функционирования системы с ограничением ресурса длины очереди	220
Пример 3. Моделирование динамики функционирования	223
Варианты индивидуальных заданий	225

Лабораторная работа № 6	
Моделирование систем массового обслуживания сетями Петри	232
Теория	232
Ход выполнения работы	234
Практические рекомендации по выполнению работы и использованию программного обеспечения	235
Пример 1. Моделирование устройства СМО с ограничением времени ожидания	235
Пример 2. Моделирование узла СМО с неоднородным входным потоком и абсолютным приоритетом обслуживания	238
Пример 3. Моделирование узла СМО с отказом основного устройства и остановкой по условию	243
Варианты индивидуальных заданий	248
Лабораторная работа № 7	
Моделирование систем массового обслуживания с визуализацией характеристик системной динамики	257
Теория	257
Ход выполнения работы	258
Практические рекомендации по выполнению работы и использованию программного обеспечения	259
Пример 1. Моделирование узла СМО с визуализацией динамики функционирования	261
Пример 2. Моделирование узла СМО с общей очередью	270
Пример 3. Моделирование СМО с ограниченной очередью	272
Варианты индивидуальных заданий	273
Контроль успеваемости	283
Первая аттестация	283
Вторая аттестация	284
Экзамен	285
Вопросы к экзамену	286
Типы экзаменационных задач	288
Электронные материалы по дисциплине «Системотехника»	291
Литература	293

ВВЕДЕНИЕ В ДИСЦИПЛИНУ «СИСТЕМОТЕХНИКА»

Учиться надо весело! Чтобы переваривать знания, надо поглощать их с аппетитом.

А. Франс,
французский писатель

Целью изучения дисциплины «Системотехника» является освоение методологии общей теории систем и системного анализа в рамках квалификационных требований, предъявляемых к инженеру-системотехнику по специальности 1-40 02 01 «Вычислительные машины, системы и сети» и приобретение практических навыков проектирования и исследования технических систем.

Дневная форма обучения

Структура дисциплины. Дисциплина «Системотехника», согласно учебному плану специальности 1-40 02 01 «Вычислительные машины, системы и сети», изучается на 3-м курсе обучения, в осеннем семестре и включает в себя 34 часа лекционных занятий и 34 часа лабораторного практикума. Итоговой формой контроля по дисциплине является экзамен.

Рейтинговая система контроля успеваемости. Для оценивания аудиторной и самостоятельной работы в процессе прохождения дисциплины используется рейтинговая система контроля успеваемости. Она является накопительной и предполагает суммирование балльных оценок и бонусов, выставляемых за все виды учебных активностей в течение семестра. Итоговая сумма, набранная за время прохождения дисциплины, является индивидуальным рейтингом студента (ИРС).

Лекционная часть дисциплины. Лекции по дисциплине проводятся в традиционной форме. Основная часть теоретического материала приведена в *лекционном курсе*. Помимо этого на лекциях будет рассматриваться дополнительный материал в виде пояснений и практических примеров. Ссылки на дополнительный материал в тексте *Лекционного курса* оформлены специальным значком, например:

ПРИМЕР: сравнение свойств системы и комплекса



Некоторые практические примеры подробно рассматриваются в лабораторном практикуме, и на них также имеются специальные ссылки в тексте *Лекционного курса*, например:

ПРИМЕРЫ моделирования системной динамики и систем массового обслуживания – см. Лабораторные работы № 5 и № 6



Промежуточный контроль освоения лекционной части дисциплины проводится в виде теста, дважды в течение семестра, на аттестационных неделях. Тест состоит из 10 – 15 вопросов по пройденному материалу с ответами выборочного типа в каждом вопросе. Правильный ответ оценивается в 10 баллов рейтинга. Результат тестирования, составляющий *80 и более процентов от максимального результата тестирования* на потоке, поощряется бонусом в 50 баллов рейтинга. Кроме тестовых вопросов в аттестацию входят практические задания по пройденным на лабораторном практикуме методам проектирования и исследования систем.

На лекциях, посвященных конкретным методам проектирования и исследования систем, для индивидуального или группового выполнения могут быть предложены практические задания, правильное выполнение которых оценивается определенным числом баллов рейтинга. Оригинальные и первоочередные решения поощряются бонусами.

Лабораторный практикум. Целью практикума является освоение практических навыков проектирования, исследования систем и решения задач системного анализа. Работы выполняются в соответствии с индивидуальным вариантом задания. Номер индивидуального варианта выдается на первом занятии и сохраняется до конца семестра.

Отчет по работе представляется в электронном виде. В отчет включается: название работы, вариант задания, анализ задания, основные и промежуточные результаты по каждому пункту хода выполнения работы, выводы по работе. Защита работ проводится индивидуально и оценивается в соответствии со следующими правилами:

- работа, защищенная непосредственно на занятии, отведенном на ее выполнение, оценивается в 150 баллов рейтинга (в этом случае оформление отчета не обязательно, достаточно представление основных и промежуточных результатов);
- работа, защищенная на занятии, следующем за занятием, отведенным на выполнение работы, оценивается в 100 баллов рейтинга;
- работа, защищенная не в срок, оценивается в 75 баллов рейтинга.

Набор 750 баллов рейтинга по лабораторному практикуму поощряется дополнительным бонусом в 100 баллов рейтинга. Суммарный рейтинг по лабораторному практикуму, равный 375 баллам, позволяет претендовать только на итоговую экзаменационную оценку «5 баллов» и является нижней границей допуска к экзамену.

Защита лабораторных работ проводится исключительно на занятиях по лабораторному практикуму в соответствии с расписанием. Защиты лабораторных работ принимаются только в течение семестра, до календарной даты начала сессии (за исключением случаев недобора 375 баллов рейтинга по лабораторному практикуму). В случае, если на лабораторном занятии возникает очередь из желающих защищать лабораторные работы, студенты, имеющие больший суммарный рейтинг по лабораторному практикуму, пользуются преимущественным правом первоочередной защиты.

Аттестации (промежуточный контроль успеваемости). Для положительной аттестации ИРС по всем учебным активностям на момент аттестации должен составлять не менее 2/3 от среднего ИРС на потоке. Аттестационная ведомость (список потока с суммарным рейтингом каждого студента) выдается старосте группы.

Экзамен (итоговый контроль успеваемости). Экзамен представляет собой письменную работу, на выполнение которой отводится 2 часа. Билет включает в себя:

- один теоретический вопрос, оцениваемый из 100 баллов;
- тестовую часть, состоящую из одного обязательного и трех дополнительных вопросов выборочного типа, каждый из которых оценивается из 10 баллов;
- задачу, оцениваемую из 60 баллов.

На дополнительные тестовые вопросы студент отвечает по желанию, если хочет повысить экзаменационную оценку. Неверные ответы на тестовые вопросы не уменьшают суммы баллов, набранной за теоретический вопрос и задачу. Стандартные десятибалльные оценки определяются по результатам экзаменационной работы, как показано в табл. 1.

Таблица 1

Сумма баллов за экзаменационную работу	0...25	26...45	46...74	75...85	86...105	106...125	126...145	146...165	166...189	190...200
Оценка	1	2	3	4	5	6	7	8	9	10

Экзаменационная оценка выставляется на основании письменной работы, дополнительные вопросы и обсуждения не практикуются. Работа выполняется на специальном бланке. Бланк и черновики выдаются преподавателем. При себе на экзамене следует иметь ручку, карандаш, ластик, а также зачетную книжку. Пользование конспектом и консультации друг с другом запрещены. Экзамен начинается в 9.00. Время опоздания вычитается из времени выполнения работы. Выходить во время работы запрещается. К экзамену разрешается подготовить страницу справочного материала, на которую вынести все формулы и схемы, запоминание которых представляет трудности, но без комментариев, обозначений и подписей. Допустимый объем материала – одна сторона формата А4, шрифт – не менее 10 pt. Материал утверждается и разрешается к использованию преподавателем на консультации к экзамену.

Студенты, имеющие суммарный рейтинг по результатам семестра, *равный или больший 80 % от максимального* на потоке, имеют право на два дополнительных балла к экзаменационной оценке (за исключением неудовлетворительных) или на автоматическое засчитывание максимально возможного результата за любую часть экзаменационного билета (теоретическую, тестовую или задачу). Выбор экзаменационного бонуса осуществляется студентом на экзамене, после выдачи билета.

Студенты, имеющие суммарный рейтинг *на 80 % или более, чем средний* на потоке, получают 10 баллов за экзамен автоматически. Студенты, занимающие в конце семестра *три верхние строчки итогового рейтинга*, получают соответственно 9, 8 и 7 баллов за экзамен автоматически, если они не претендуют на более высокую оценку – в этом случае они могут сдавать экзамен на общих основаниях.

Заочная форма обучения

Структура дисциплины. Дисциплина «Системотехника», согласно учебному плану специальности 1-40 02 01 «Вычислительные машины, системы и сети», изучается на 4-м курсе заочной формы и включает в себя:

- установочную лекцию;
- 4 часа лекционных занятий;
- 6 часов лабораторного практикума в каждой подгруппе;
- контрольную работу.

Итоговой формой контроля по курсу является экзамен.

Лекционная часть курса. Основная обязательная часть теоретического материала по дисциплине приведена в *лекционном курсе*. Целью лекци-

онных занятий является реферативное представление теоретического материала по курсу с рекомендациями по самостоятельной работе с конспектом лекций и дополнительными материалами.

Контрольная работа. Целью выполнения контрольной работы по дисциплине «Системотехника» студентами заочной формы обучения является практическое освоение методов системного анализа и моделирования систем, рассматриваемых в теоретической части курса.

Контрольная работа состоит в самостоятельном выполнении по индивидуальному варианту 4-х работ лабораторного практикума: № 3, № 5, № 6 и № 7. Варианты заданий по номерам приведены в конце каждой лабораторной работы. При выполнении работ следует пользоваться методическими указаниями к лабораторному практикуму. В отчет по каждой работе включается: название работы, цель и ход выполнения работы, индивидуальный вариант задания, анализ задания, основные и промежуточные результаты по каждому пункту хода выполнения работы, выводы по работе. Отчеты по всем четырем работам подшиваются в общую папку контрольной работы. Контрольная работа сдается на проверку в распечатанном виде *не позднее* календарной даты начала сессии. Первой страницей после титульного листа подшивается распечатанный вариант индивидуального задания, оформленного в соответствии с номером варианта. Второй страницей подшивается распечатанный бланк проверки и защиты контрольной работы. Пример оформления титульного листа, задания, бланка проверки и защиты выдается в электронном виде. Решение о времени и месте защиты принимаются преподавателем после проверки. Пожалуйста, не забудьте указать на обложке контрольной работы Ф.И.О. полностью и контактный телефон или адрес электронной почты для сообщения о результатах проверки работы и назначения срока защиты. До начала сессии контрольные на проверку можно присылать по электронной почте.

Защита контрольных работ проводится на занятиях по дисциплине в соответствии с расписанием дней заочника и на лабораторном практикуме во время сессии. *Выполнение и успешная защита контрольной работы является обязательным условием допуска к экзамену по дисциплине.* На защите проверяется самостоятельность выполнения работы путем проверочных заданий на демонстрацию основных навыков использования программного обеспечения; способность анализировать полученные результаты; способность вносить изменения в модельные реализации при изменении исходных данных по заданию преподавателя. В процессе защиты разрешается пользоваться распечатками выданных УМК и прочих электронных материалов по курсу, а также любой справочной литературой.

Лабораторный практикум. На занятиях по лабораторному практикуму во время сессии рассматриваются теория и примеры работ лабораторного практикума, вызвавшие затруднения при самостоятельном выполнении работ; производится защита контрольных работ; при необходимости даются консультации по примерам тестовых заданий, условиям и решению задач, включенных в экзаменационные билеты.

Экзамен (итоговый контроль успеваемости). Экзамен представляет собой самостоятельную письменную работу. Билет разделен на две части по категориям сложности и включает в себя:

- 1) часть I – теоретический вопрос, оцениваемый из 100 баллов;
- 2) часть II – тестовая часть, состоящая из одного обязательного и трех дополнительных вопросов выборочного типа, каждый из которых оценивается из 10 баллов, и задачи, оцениваемой в 60 баллов.

Если Вы не претендуете на оценку выше 5 баллов, то на экзамене можете выполнять только часть I экзаменационной работы. Теоретические вопросы, включенные в часть I, относятся к основной обязательной части теоретического материала по курсу и в перечне экзаменационных вопросов курса отмечены *звездочкой*.

Полная категория сложности позволяет претендовать на любую оценку выше 5 баллов и предполагает выполнение частей I и II билета. На дополнительные тестовые вопросы студент отвечает по желанию, если хочет повысить экзаменационную оценку. Неверные ответы на тестовые вопросы не уменьшают суммы баллов, набранной за теоретический вопрос и задачу.

Правила организации и проведения экзамена, а также таблицу перевода экзаменационных баллов в оценки десятибалльной системы – см. в разделе *Дневная форма обучения*.

ЛЕКЦИОННЫЙ КУРС

Последнее, что понимаешь, когда пишешь книгу, – это с чего ее следует начать.

Блез Паскаль, математик

1. Основы системотехники

Общая системотехника носит междисциплинарный характер: изучаемые ею законы не зависят от конкретного типа системы. Теоретической базой системотехники является общая теория систем. Объектами исследования общей теории систем являются классы систем, объединенных не только по традиционным признакам (биологические, технические, социальные и т.д.), но и по видам отношений элементов в системе. Под термином «отношение» понимается структура, информация, ограничение, организация, управление и т.п.

Предметом дисциплины «Системотехника» для специальности 1-40 02 01 «Вычислительные машины, системы и сети» является *техническая системотехника*, изучающая:

- 1) методы проектирования и исследования технических систем;
- 2) организацию процесса их создания, использования и развития.

Основой методологии технической системотехники является системный анализ: совокупность методов и технологий, использующих в себе системный подход. В свою очередь, системный подход основывается на представлении объекта в виде системы, изучении ее структуры, анализе на микроуровне (элементы и связи между ними) и макроуровне (интерфейсы системы), анализе системных функций с точки зрения достижения общих системных целей.

Структура и составляющие междисциплинарных связей предметной области представлены на рис. 1.

Квалификация «инженер-системотехник», присваиваемая специалистам в области проектирования технических систем, была введена в 1977 году. Согласно квалификационному справочнику, инженер-системотехник должен осуществлять исследовательскую, методическую и методологическую деятельность.

Исследование существующих аналогов является исходным пунктом разработки любой новой системы. Комплексное исследование в области систем включает:



Рис. 1

1) планирование, координацию и проведение исследований системы в целом и на уровне подсистем;

2) выявление и описание общих системных характеристик и закономерностей, не зависящих от конкретного типа системы, классификацию систем для привязки к типовым проектным решениям;

3) оценку эффективности функционирующей системы.

Методическая деятельность инженера-системотехника включает:

1) разработку технического задания на систему в целом и на отдельные подсистемы;

- 2) формирование состава методик проектирования и других документов, регламентирующих проектировочную деятельность;
- 3) описание типового состава проектов подсистем;
- 4) определение этапности проектирования, внедрения и функционирования системы;
- 5) выпуск документов, регламентирующих внедрение и эксплуатацию системы.

Внедрение методологии в практику проектирования осуществляются с помощью координации всех работ по созданию системы, направленной на решение следующих задач:

- 1) организация стыковки проектирования отдельных подсистем: выдача исходных данных проектировщиками подсистем, увязка частных проектов и их интеграция в единый проект;
- 2) обеспечение реализации принципов системного подхода в процессе проектирования;
- 3) разработка координационного плана организации внедрения;
- 4) текущий контроль за ходом разработки и его оценка.

Объектами разработки инженера-системотехника по специальности 1-40 02 01 «Вычислительные машины, системы и сети» являются информационные системы: собственно вычислительные системы и сети общего и специального назначения; автоматизированные системы управления технологическими процессами, производствами и предприятиями; системы автоматизированного проектирования и поддержки полного жизненного цикла и т.п. [1 – 3].

1.1. Основные понятия системотехники

– *Ненавижу определения!*

Бенджамин Дизраэли,
англ. государственный деятель

Элемент – объект любой природы (материальный, энергетический, информационный), неделимый в контексте текущего рассмотрения. Т.е. элемент системы обладает рядом свойств, но его внутренне содержание (строение) безотносительно к цели данного рассмотрения – это «черный ящик» (рис. 2). Например, если элемент входит как составная часть в автомобиль, то это двигатель, кузов, колеса и т.д. На более низком, более детализированном уровне рассмотрения элемент верхнего уровня уже, в свою очередь, рассматривается как система – элементами кузова являются двери, стекла, крылья, бампер и т.д.

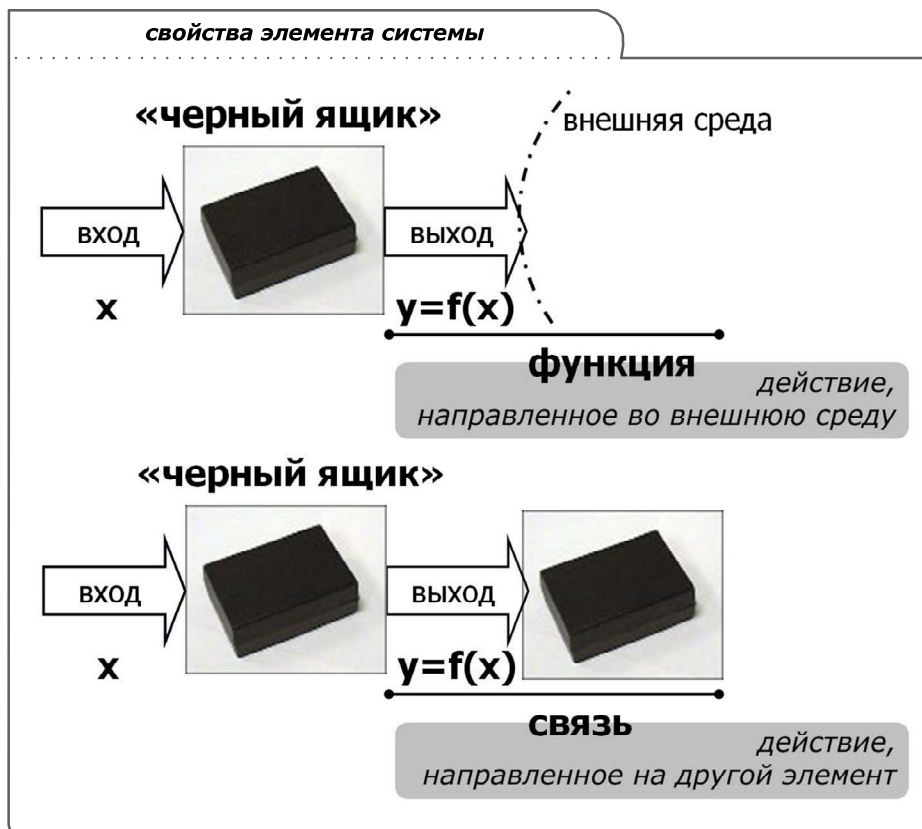


Рис. 2

Функция – действие того или иного объекта, направленное во внешнюю среду. *Связь* – это отношение объектов друг к другу (см. рис. 2). Связи могут иметь различную природу, от организационной до виртуальной. Они описывают важный для целей рассмотрения обмен между элементами веществом, энергией, информацией.

Рассмотрим в качестве примера системного элемента *RS-триггер*. Как самостоятельный объект триггер характеризуется внутренней структурой, которая может быть представлена логической или электрической принципиальной схемой. Но как системный элемент *RS-триггер* представляет собой «черный ящик», изображаемый условным обозначением (рис. 3) и характеризующийся только функцией: он обеспечивает ввод в систему цифровых сигналов с заданными свойствами (сохраняет текущее значение выхода T при «нулевых» входах, формирует на выходе T значение 0 при подаче единичного сигнала на вход R , формирует на выходе T значение 1 при подаче единичного сигнала на вход S).

Внешняя среда есть совокупность окружающих систему элементов внешнего мира, не входящих в ее состав, но оказывающих на нее влияние

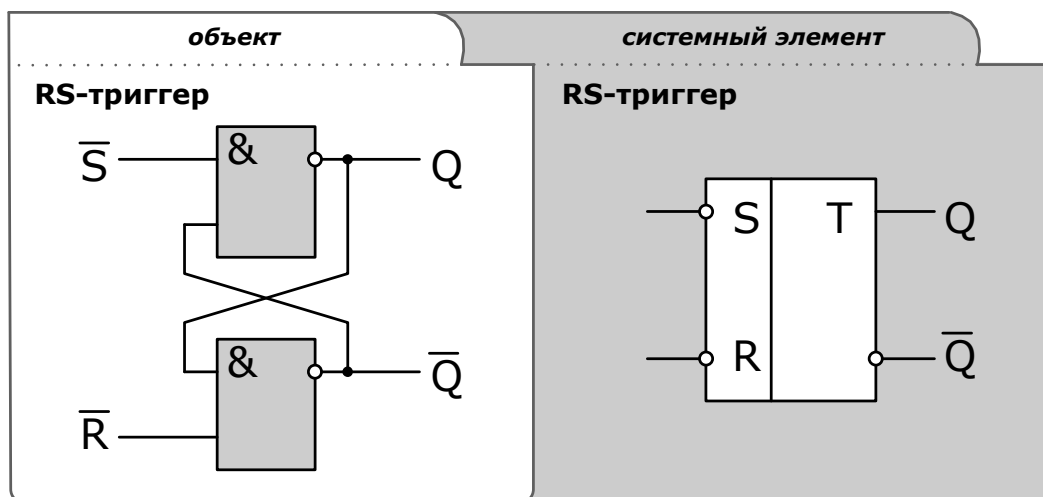


Рис. 3

ПРИМЕРЫ других элементов систем



или подверженных влиянию с ее стороны. Внешняя среда в свою очередь также является системой. Для характеристики взаимодействия системы со средой существенны направленность этого взаимодействия (среда-система или система-среда), а также его природа – информационное, вещественное или энергетическое. Если такое взаимодействие отсутствует или с некоторым приближением им можно пренебречь, система называется *закрытой* (изолированной). При наличии такого взаимодействия система называется *открытой*.

Комплекс – это совокупность элементов, объединенных по какому-либо принципу, некоторый набор, группа. Чаще всего в комплекс объединяются объекты (элементы) одинаковой природы. Комплексы элементов в структуре системы обычно выделяются по принципу простых или относительно более слабых связей между элементами разных комплексов.

Система также является совокупностью элементов, но, в отличие от комплекса, система обладает рядом свойств, присущих только ей.

Связность. Элементы системы связаны друг с другом существенными связями, которые значительно сильнее в контексте рассмотрения, чем связи между этими элементами и внешними по отношению к системе объектами. Например, с точки зрения автомобиля как системы колеса связаны

с другими элементами автомобиля существенно сильнее, чем с дорогой, по которой автомобиль едет.

Целостность. Ни один из элементов системы не может быть изъят из нее. Система предназначена для выполнения конкретных функций, и все без исключения элементы участвуют в выполнении этих функций. Если элемент выделить из системы, он становится независимым объектом, и система перестает выполнять весь набор функций, для которых она предназначена.

Делимость. Каждый из элементов системы является самостоятельным объектом с четкими границами и связями с другими элементами.

Организация. Элементы системы ориентированы относительно друг друга в пространстве, во времени и в связях друг с другом в отличие от комплекса, в котором элементы могут быть представлены как простой, неупорядоченный набор.

Таким образом, *система* – это совокупность элементов, обладающая свойствами организации, связности, целостности и делимости. Прежде чем приступить к описанию системы, исследователь должен четко определить следующее: границы системы, совокупность элементов и связей между ними, интерфейсы системы (внешние связи со сторонними объектами), совокупность функций, которые система выполняет.

ПРИМЕР: сравнение свойств системы и комплекса



Система называется *сложной*, если образующим ее элементам присущи все свойства систем: в этом случае они называются *подсистемами*.

Параметр – величина, выражающая свойство системы в целом, ее отдельной части (элемента, подсистемы) или влияющей на систему внешней среды. Соответственно различают *выходные, внутренние и внешние параметры* (рис. 4). Параметры могут быть постоянными и переменными.

На рис. 5 представлена параметризация компьютера как системы. Компьютер является сложной системой в силу наличия у него большого числа элементов, разнообразных связей между элементами и подсистемами, свойств целостности, организации и пр. К подсистемам компьютера относятся процессор (процессоры), оперативная память, кэш-память, шины, устройства ввода-вывода.

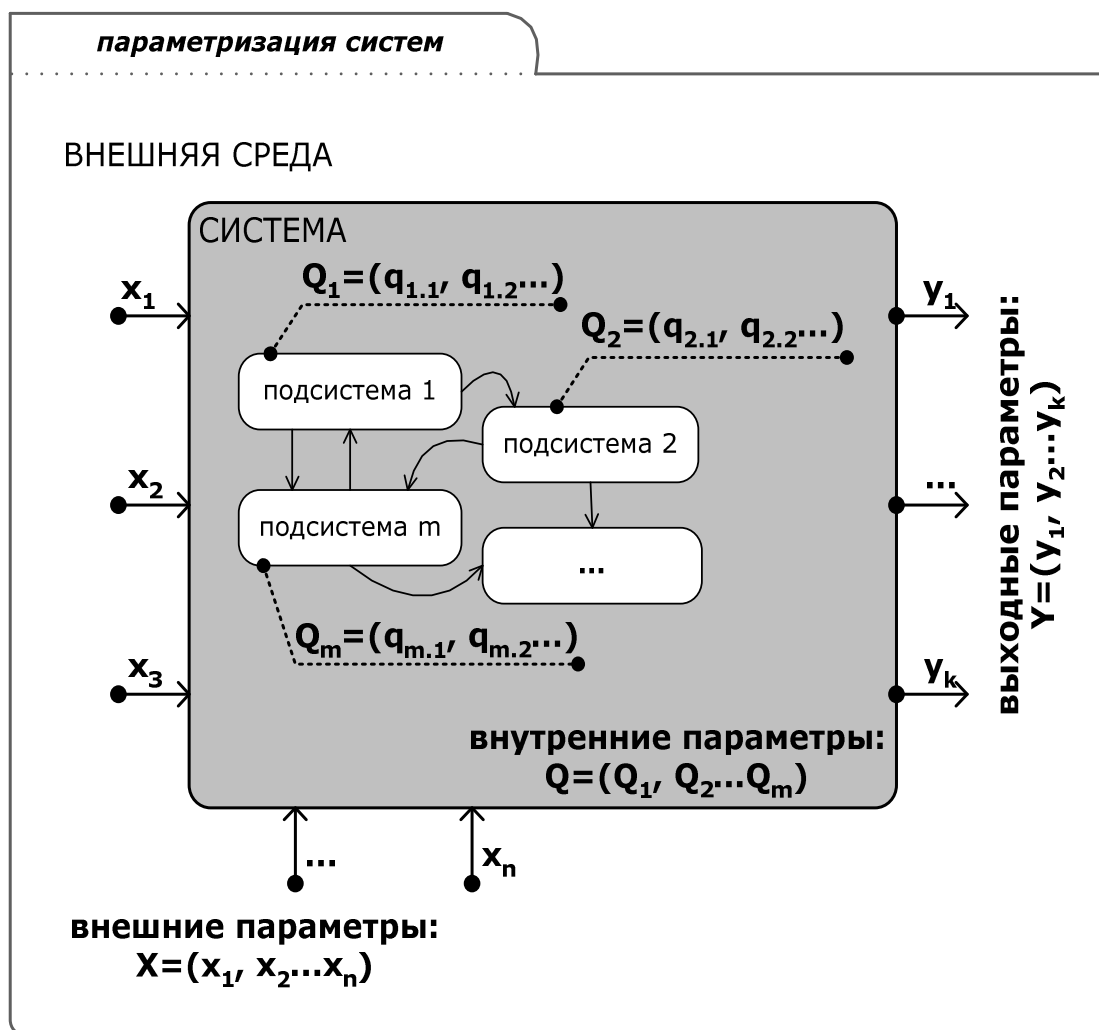


Рис. 4

В качестве надсистемы могут выступать вычислительная сеть, автоматизированная и (или) организационная система, к которым принадлежит компьютер. Внутренние параметры – время выполнения арифметических операций, чтения (записи) в накопителях, пропускная способность шин и др. Выходные параметры – производительность компьютера, емкость оперативной и внешней памяти, себестоимость, время наработки на отказ и др. Внешние параметры – напряжение питания сети и его стабильность, температура окружающей среды и др.

Состояние – совокупность значений переменных параметров, зафиксированных в конкретный момент времени процесса функционирования.

Поведение (динамика) системы – изменение состояния системы в процессе функционирования [1 – 4, 7].

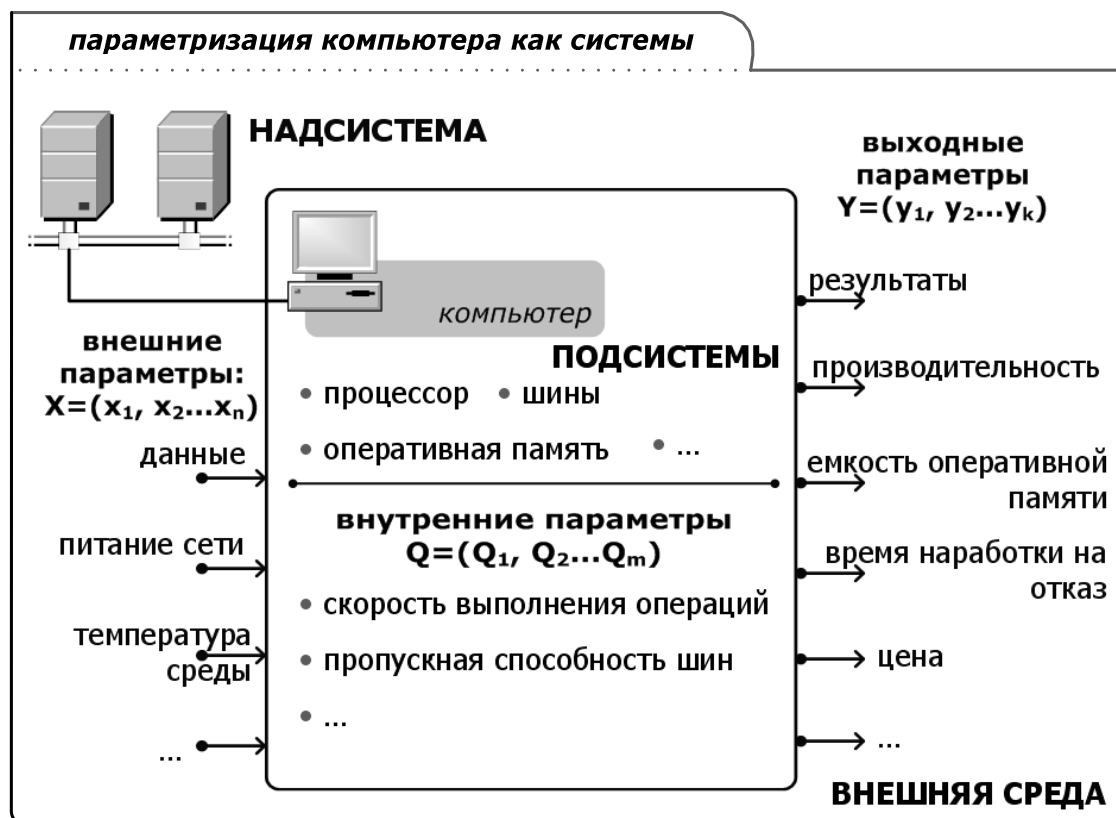


Рис. 5

ПРИМЕРЫ параметризации других систем



1.2. Структура и иерархия систем

Каждая часть хочет быть в своем целом, в коем лучше себя сохраняет.

Леонардо да Винчи

Процесс последовательного деления системы на образующие ее подсистемы (элементы) называется *декомпозицией*. Декомпозиция является одной из важнейших процедур системного анализа. Обратный процесс объединения отдельных подсистем (элементов) в систему называется *агрегированием* (композицией). *Структура системы* – это представление результата декомпозиции с указанием связей между элементами. Декомпозиция системы может иметь различную основу: материальную; функциональную (рис. 6); алгоритмическую (алгоритм программы, инструкция); временную (календарь событий, происходящих в системе) и т.д.

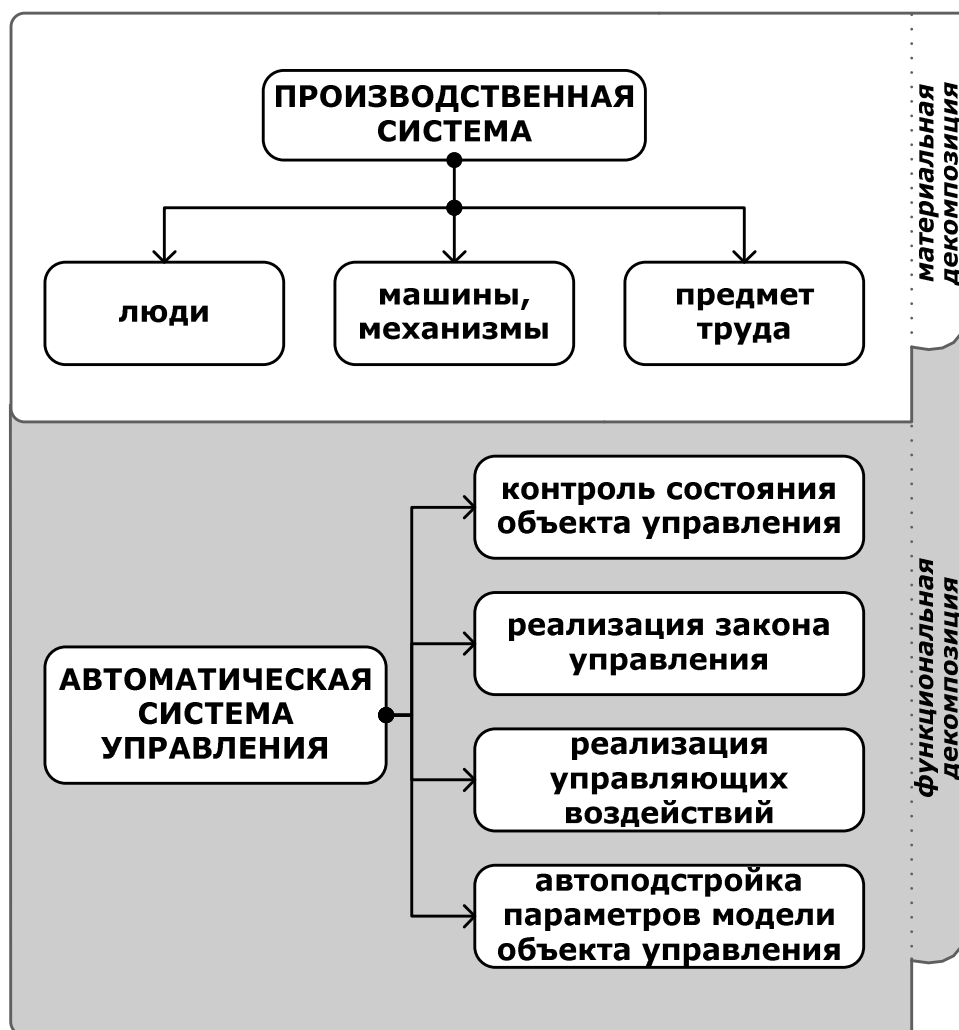


Рис. 6

Структура системы может быть охарактеризована по имеющимся в ней или преобладающим типам соединений (рис. 7).

Последовательным называется такое соединение элементов в системе, когда реакция первого элемента на внешнее воздействие является входным воздействием для второго элемента и т.д. Структура системы, в которой имеются только последовательные связи, называется *линейной*.

При *параллельном* соединении внешнее воздействие является входным для всех элементов структуры, а реакции элементов суммируются.

Обратная связь изменяет входное воздействие по результатам функционирования элемента, т.е. внешнее воздействие модифицируется с учетом текущего состояния элемента системы. Крайне редко встречаются системы без обратных связей.

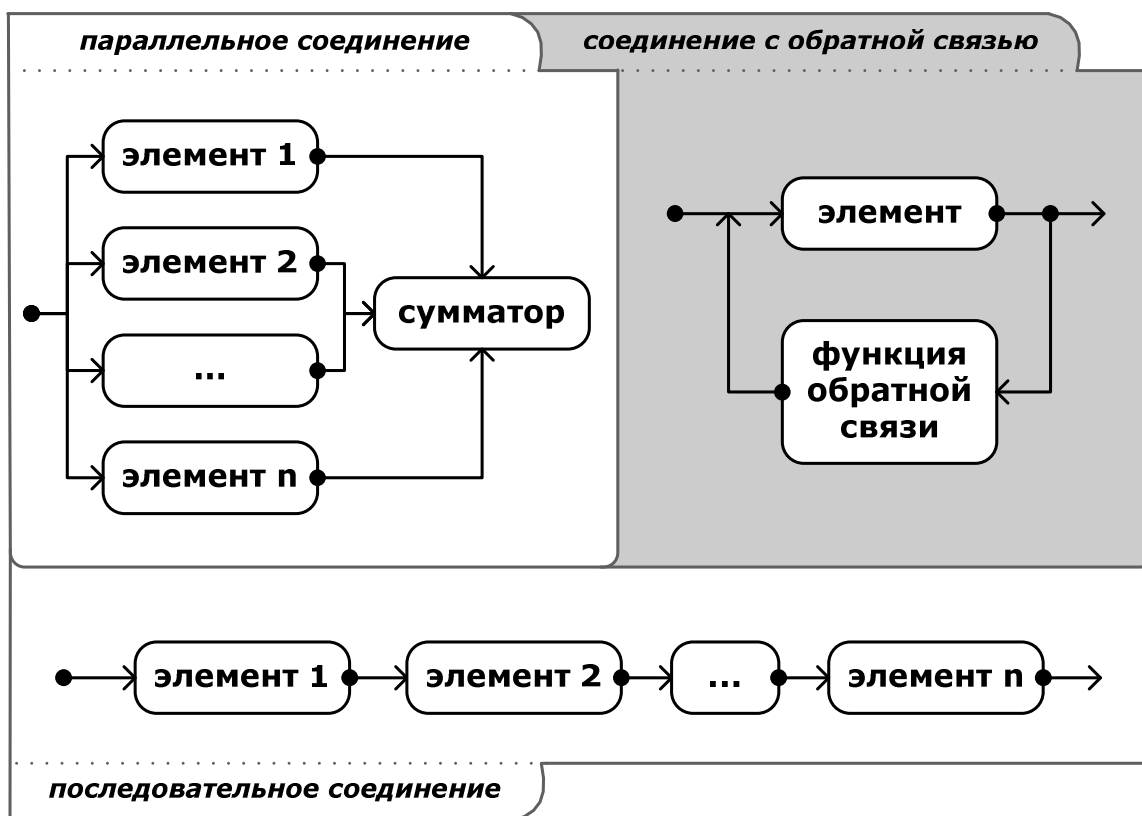


Рис. 7

ПРИМЕРЫ: 1) функции элементов обратной связи;
2) решение задач идентификации типов связей
и агрегирования



Структура с наличием подчиненности, т. е. неравноправных связей между элементами, называется *иерархической*. Фрагмент иерархической структуры вуза представлен на рис. 8.

Иерархическое представление структуры предполагает одновременное применение двух схем декомпозиции – вертикальной и горизонтальной. Вертикальная декомпозиция позволяет представить систему как иерархию взаимозависимых уровней. Горизонтальная декомпозиция означает выделение на одном и том же уровне иерархии равнозначных в каком-либо смысле подсистем и выявление существенных с точки зрения решаемой задачи взаимосвязей между ними (рис. 9).

Иерархическое представление структуры является основополагающим в системотехнике. Также в технических системах встречаются *сетевые* и, реже, *матричные (решетчатые)* структуры (рис. 10).

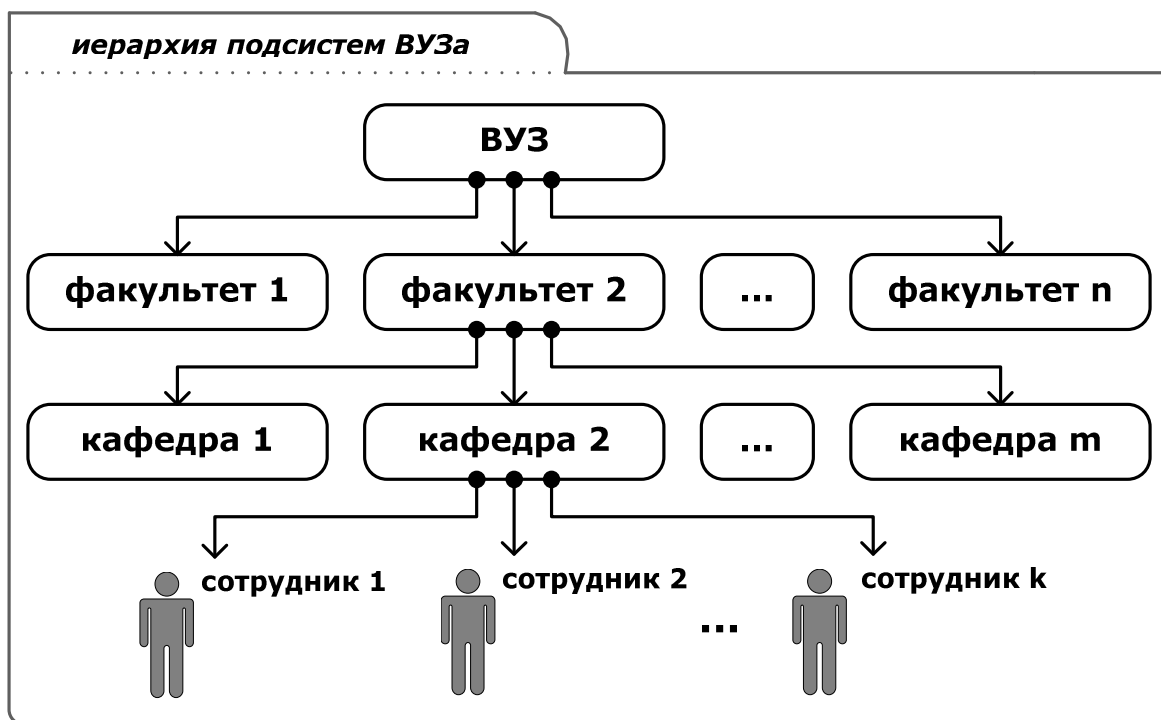


Рис. 8

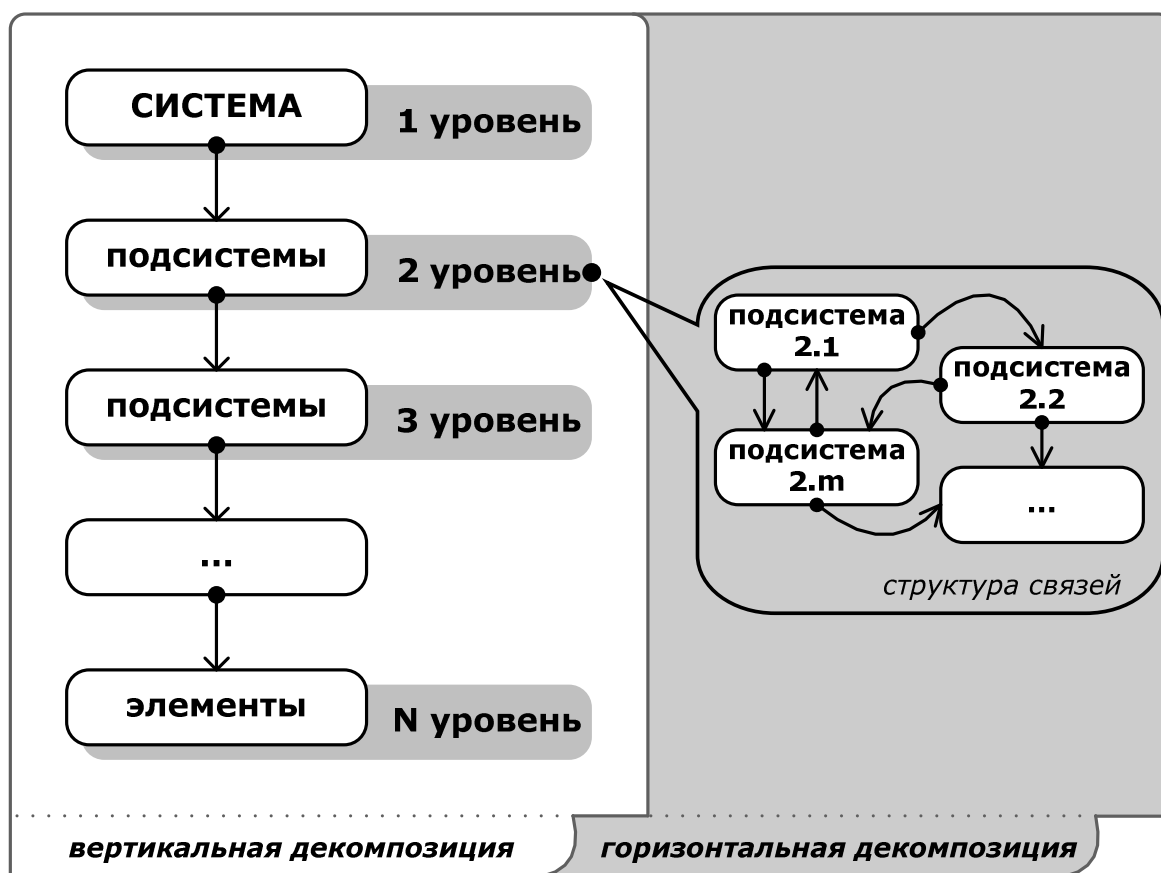


Рис. 9

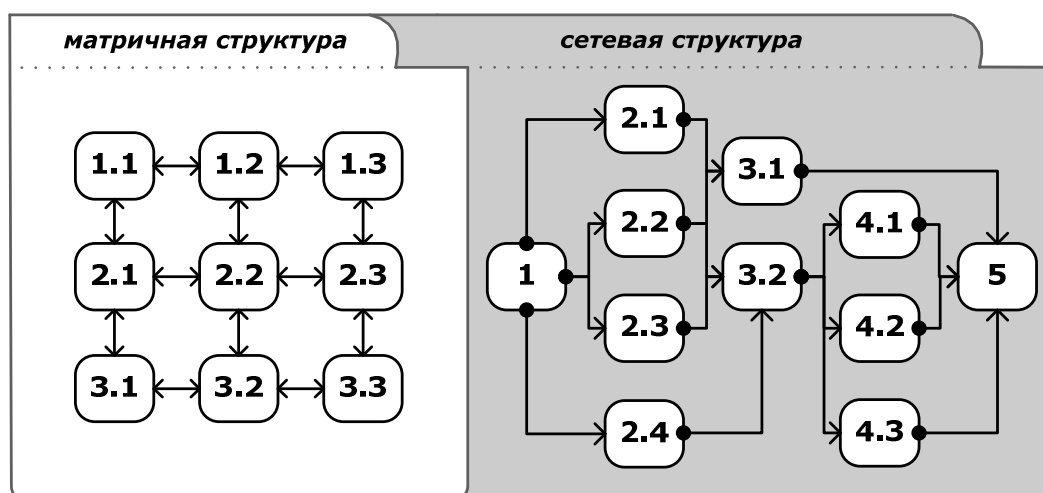


Рис. 10

Универсальным средством описания структуры является *граф*. Граф состоит из обозначений элементов, называемых вершинами, и обозначений связей между ними, называемых ребрами (если направление связи не учитывается) или дугами (если направление учитывается). Обратные связи систем задаются на графах кольцевыми путями.

Если в графе требуется кроме структуры отразить параметрические характеристики системы, то связям приписывают различные веса (*взвешенные графы*). Для задания различий между элементами или связями также используют раскрашивание вершин или связей (*раскрашенные графы*) [1 – 3, 6, 7].

Для графов построена содержательная теория, имеющая многочисленные приложения. Задачи этой теории связаны с различными преобразованиями графов, а также с возможностью рассмотрения различных отношений на графах: весов, рангов и т.д. Системотехнические приложения теории графов рассматриваются в разделе 2.3.

1.3. Принципы системного управления

– Неужели вы скажете, что он сам собою управил так? Не правильнее ли думать, что управился с ним кто-то совсем другой?

М.А. Булгаков
«Мастер и Маргарита»

Как уже отмечалось, взаимодействия системы с внешней средой и внутренние взаимодействия могут быть:

- *целенаправленными* (целесообразными), т.е. санкционированными, предусмотренными алгоритмом функционирования системы, изменяющими состояние системы желаемым образом;

- *возмущающими*, т.е. оказывающими отрицательное влияние на выполнение системой ее функций вплоть до полной или частичной потери работоспособности.

Целенаправленное внешнее воздействие на систему, формируемое с целью изменения состояния и выходной ситуации желаемым образом, называется *управлением*. Поддержание работоспособного состояния системы в условиях действия внешних и внутренних возмущений и помех является основной целью системного управления. Достижение этой цели осложняется случайным характером возмущений.

Для снижения отрицательного влияния возмущений в технических системах используются кибернетические принципы:

- управление по отклонению (*принцип обратной связи*);
- управление по возмущению (*принцип компенсации*).

При реализации *управления по отклонению* управляющее воздействие на систему вырабатывается регулятором как функция отклонения внутреннего или выходного параметра системы от его требуемого значения. Схема управления содержит обратную связь, то есть является *замкнутой*. На рис. 11 показана структурная схема управления системой по отклонению для случая действия внешних возмущений (для внутренних реализаций аналогично, при помощи локальных обратных связей для подсистем и элементов системы).

Если под действием внешних возмущений состояние системы изменится несанкционированным образом, на выходе элемента сравнения (рис. 11) формируется воздействие $\epsilon(t)$, равное разнице между требуемым значением выходного параметра системы $y_{i_{demand}}(t)$ и его текущим реальным значением $y_i(t)$: $\epsilon(t) = y_{i_{demand}}(t) - y_i(t)$. Величина $\epsilon(t)$ называется *рассогласованием*, а обратная связь такого типа – *отрицательной*. Регулятор, функцию которого может выполнять программа, устройство, внешняя система или внутренняя подсистема, обеспечивает модификацию входных санкционированных воздействий таким образом, чтобы свести рассогласование к нулю.

Универсальность и эффективность принципа управления по отклонению состоит в том, что он позволяет осуществить заданный закон изменения выходных Y или внутренних Q параметров системы независимо от того, изменение какого типа входных воздействий – санкционированных X

или возмущающих G – вызвало возникновение рассогласования. Система с управлением по отклонению реагирует на интегрированное внешнее воздействие, проявляющееся в изменении контролируемой выходной или внутренней величины.

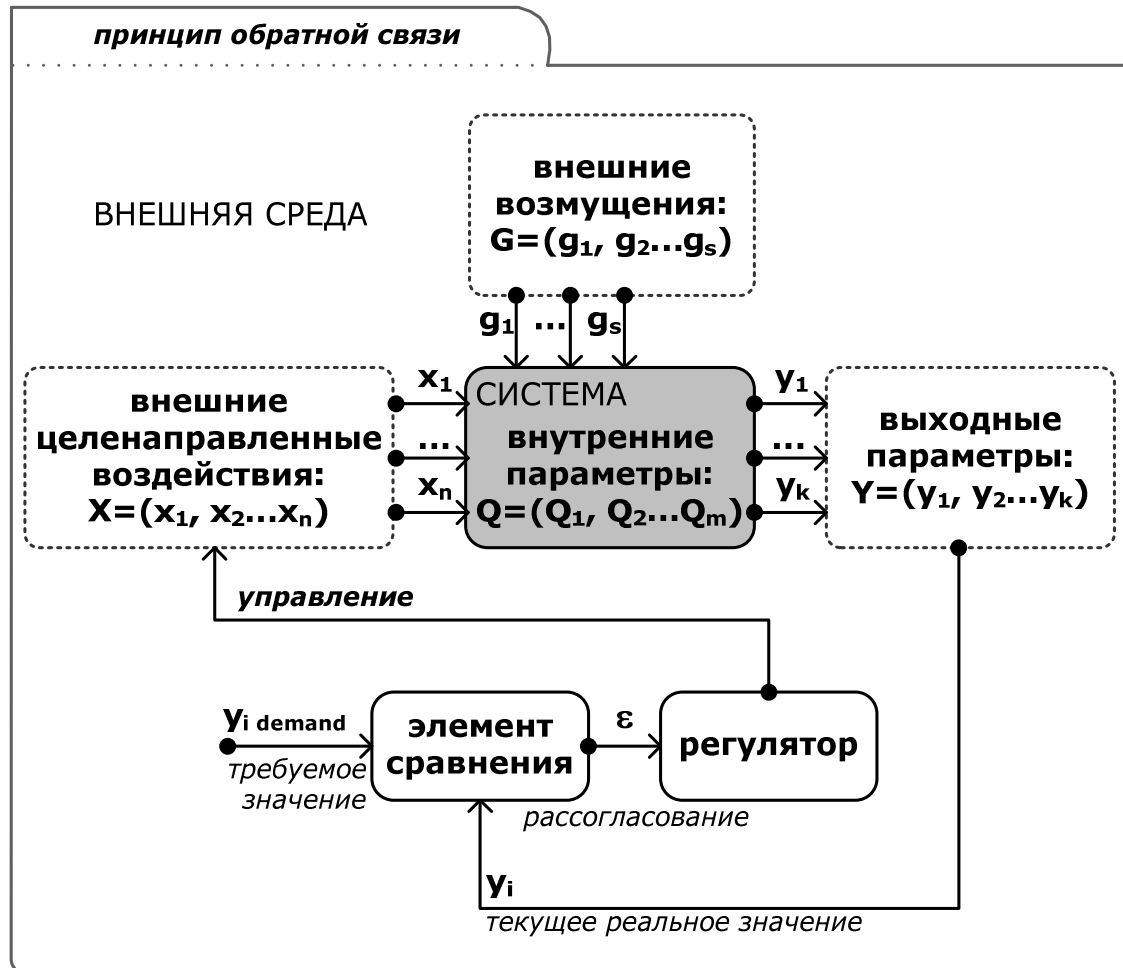


Рис. 11

ПРИМЕРЫ системного управления по отклонению



К недостаткам управления по отклонению следует отнести недостаточную оперативность, обусловленную тем, что действие управления направлено на ликвидацию рассогласования. То есть сначала допускается изменение контролируемой величины под воздействием внешних или внутренних возмущений, а потом управление его ликвидирует. При управ-

лении по отклонению влияние возмущающих воздействий на систему в значительной мере ослабляется, но не устраняется.

В случае, когда действие одного или нескольких определенных возмущений на систему в принципе недопустимо, используют *управление по возмущению* (рис. 12). Сущность такого управления состоит в том, что на основании результата измерения конкретного возмущения регулятор (компенсатор возмущения) обеспечивает модификацию входных санкционированных воздействий таким образом, чтобы компенсировать действие возмущения.

ПРИМЕРЫ системного управления по возмущению

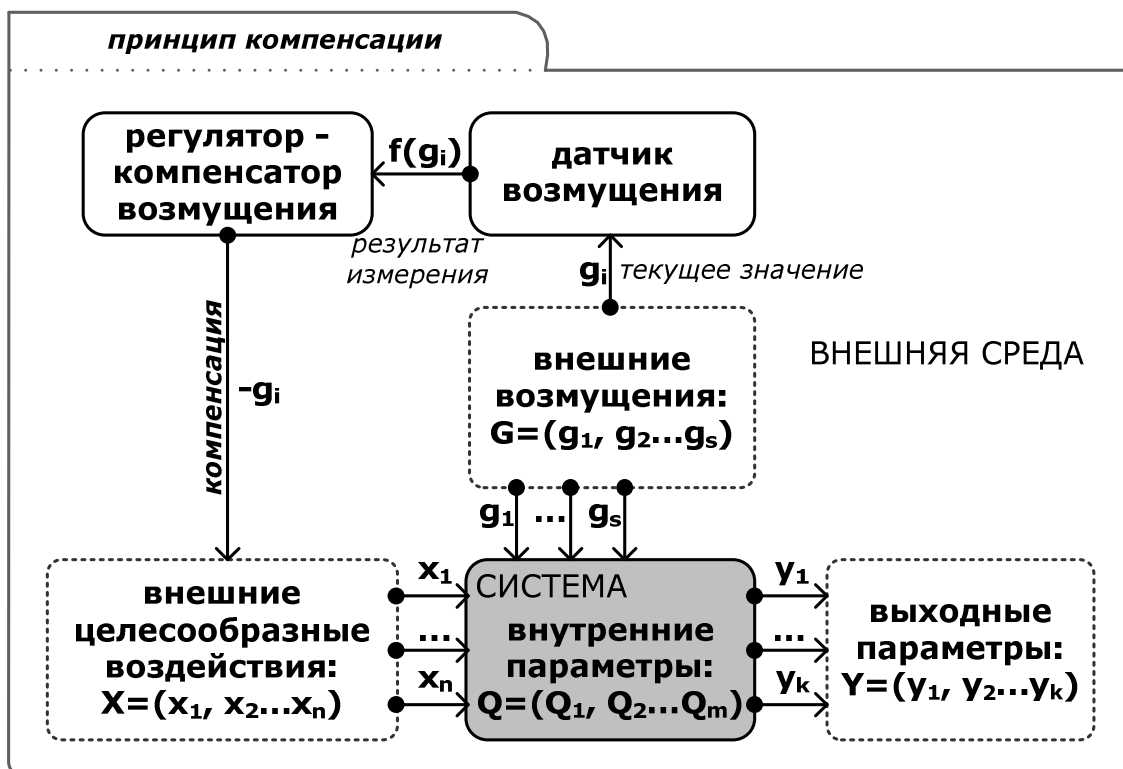


Рис. 12

ПРИМЕРЫ комбинированного управления



Принцип управления по возмущению ориентирован не на следствие, как принцип обратной связи, а на причину, нарушающее состояние системы, т.е. собственно возмущающее воздействие, и преобразование его в управляющее (компенсирующее) воздействие. К достоинствам управления по возмущению относится бóльшая оперативность по сравнению с управлением по отклонению.

Недостатком управления по возмущению является то, что при такой реализации обеспечивается компенсация одного или нескольких заранее определенных возмущений и никак не предотвращается влияние на систему других возмущающих воздействий. При этом ошибка управления имеет место даже при учете всех внешних возмущений, т.к. система не может противостоять изменению внутренних параметров.

Улучшение качества управления в условиях действия возмущений может быть достигнуто с использованием *комбинированного управления*, при реализации которого датчиками в контуре управления по возмущению измеряются только основные, наиболее критические возмущения, а влияние остальных возмущений учитывается в контуре обратной связи.

Все рассмотренные принципы управления могут быть реализованы двумя способами: *сигнальной* или *параметрической* настройкой. На рис. 11 и 12 показана сигнальная настройка, состоящая в реализации управления путем модификации внешних целесообразных воздействий системы. Параметрическая настройка состоит в реализации управления путем изменения одного или нескольких внутренних параметров системы. На рис. 13 показаны пример параметрической настройки внутреннего параметра системы – коэффициента пропорционального преобразования по условию $|k| \leq k_0$ с использованием управления z .

Кибернетические принципы управления универсальны, однако их использование для информационных систем должно учитывать особенности последних.

Первая особенность информационных систем с точки зрения управления состоит в природе возмущений: они могут иметь не только физическую, но и информационную природу. Отсутствие информации в определенное время или ее ошибочность также можно рассматривать как возмущения. Для устранения таких возмущений предусматривают соответствующие программы контроля достоверности поступающей информации, а также введение в систему информационной избыточности (использование альтернативных каналов получения информации, методов краткосрочного прогнозирования и т.п.)

Вторая особенность информационных систем с точки зрения управления состоит в том, что эти системы являются, как правило, *системами с последствием*.

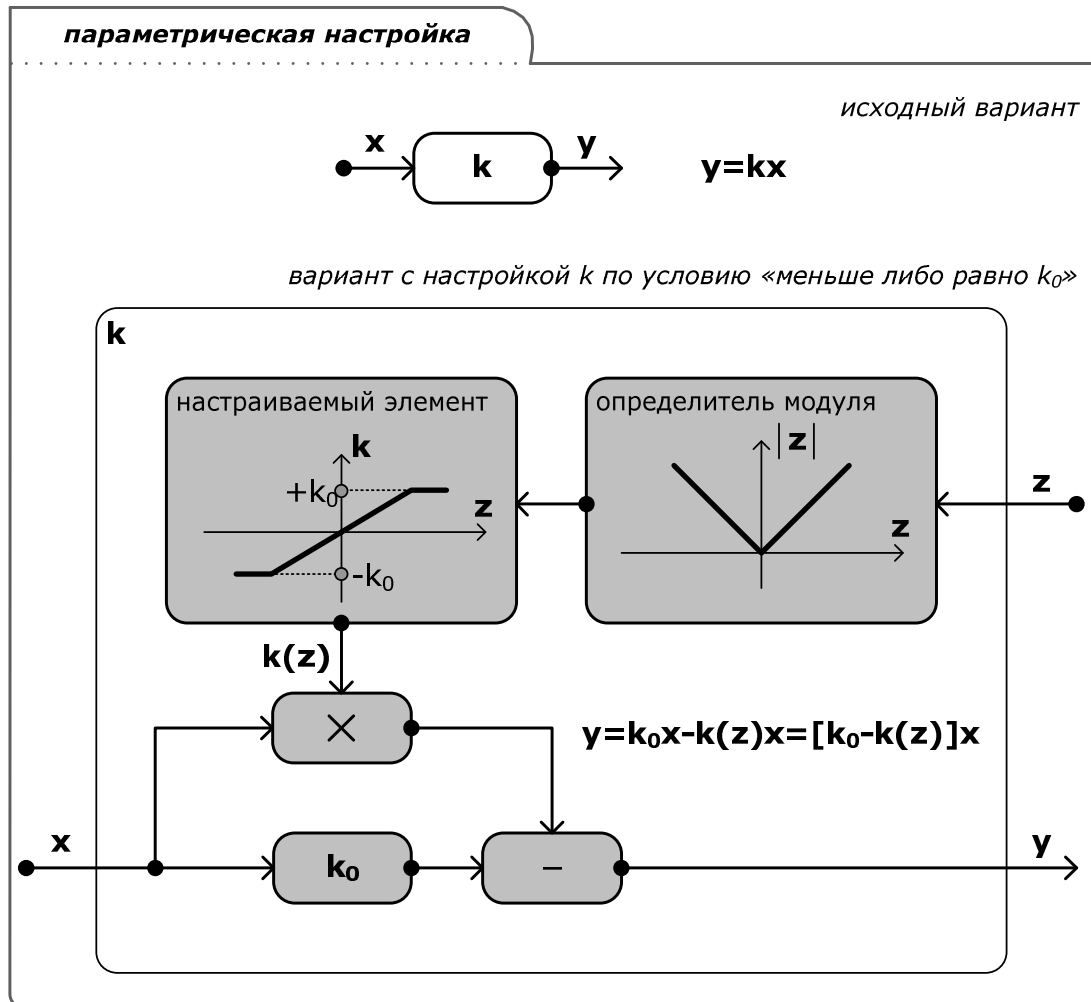


Рис. 13

Общим для кибернетических принципов управления является то, что выработка управляющих воздействия осуществляется только на основании информации о текущем состоянии системы. Так, при управлении по отклонению управляющее воздействие является функцией текущего значения рассогласования $\varepsilon(t)$, при управлении по возмущению – текущего значения возмущения $g_i(t)$, т.е. реализуется *простой* закон управления $z(t)$:

$$z(t) = f[\varepsilon(t)] \text{ или } z(t) = f[g_i(t)].$$

Это оправдано для большинства технических систем – *систем без последствия*, для которых поведение при $t \geq t_0$ зависит только от текуще-

го состояния и внешних воздействий и не зависит от того, каким образом система пришла в это состояние.

ПРИМЕР системы без последствия



Для управления системами с последствием необходимо знать предысторию поведения, т.е. состояния системы в моменты, предшествующие t_0 . Так, например, пусть информационная система обслуживает поток внешних запросов, и по условиям функционирования время ожидания обслуживания запросов во входной очереди ограничено величиной τ (при превышении ограничения запросы получают отказ в обслуживании). Состояние такой системы в моменты времени $t_0 + h$ для любого $h > 0$ зависит не только от длины очереди на момент t_0 и будущих событий: завершения обслуживаний, начатых ранее t_0 и поступления заявок после t_0 , но и от того, как долго ожидали обслуживания заявки, поступившие ранее t_0 . Для управления в подобном случае требуется знать предысторию поведения, т.е. состояния системы в моменты, предшествующие t_0 . Помимо этого для больших, сложных и территориально распределенных информационных систем актуальна задача учета в управлении динамических свойств системы: с момента принятия управленческого решения и до момента его реализации состояние системы может измениться. Простейший способ учета прошлого состояния – накопление данных об изменении параметра управления, т.е. введение в закон управления интеграла от него. Прогнозирование осуществляют на основании производной от параметра. Таким образом, эффективность использования кибернетических принципов управления для информационных систем должна обеспечиваться сложным законом управления, например, для управления по отклонению законом вида

$$z(t) = f_1[\varepsilon(t)] + f_2 \left[\int_0^{t_0} \varepsilon(t) dt \right] + f_3 \left[\frac{d\varepsilon(t)}{dt} \right],$$

а для управления по возмущению законом вида

$$z(t) = f_1[g_i(t)] + f_2 \left[\int_0^{t_0} g_i(t) dt \right] + f_3 \left[\frac{dg_i(t)}{dt} \right].$$



Третья особенность информационных систем с точки зрения управления состоит в неоднозначности интерпретации внешних воздействий персоналом, который является подсистемой и обеспечивает многие ресурсы информационной системы: знания, навыки и проч. Так, например, внедрение новых информационных систем обязательно сопровождается последствиями, которые неоднозначно воспринимаются людьми внутри организации и вне ее: они приветствуются, отвергаются или игнорируются. В результате реакция системы на целенаправленное, санкционированное внешнее воздействие может не оказаться ожидаемой даже в отсутствии возмущений.

Техническая система в идеале должна сохранять возможность целенаправленно функционировать в условиях действия внешних возмущений. Способность системы возвращаться в состояние равновесия, которое является наиболее благоприятным для выполнения системой функций после воздействия на систему каких-либо внешних факторов называется *устойчивостью системы* [4, 5, 8, 9, 26].

1.4. Системный подход

*... И если снова проиграет голытьба,
Тут не система виновата, а судьба!*

Песня из к/ф
«Остров погибших кораблей»

Основы методологии проектирования сложных систем выражены в системном подходе. Для специалиста с инженерным образованием положения системного подхода являются очевидными и естественными. Как большинство взрослых образованных людей правильно используют родной язык без привлечения правил грамматики, так инженеры в повседневной практике используют системный подход без привлечений пособий по системному анализу. Однако интуитивный подход может оказаться недостаточным для сложных задач инженерной деятельности.

Системный подход это общенаучная методология исследования любых объектов при рассмотрении их в виде сложных систем, которая не содержит конкретных средств исследования систем, а обосновывает и разрабатывает принципы этих исследований. Сущность системного подхода заключается:

- во-первых, в понимании объекта исследования как системы;
- во-вторых, в понимании процесса исследования объекта как системного по своей логике и применяемым средствам.

Как любая методология, системный подход подразумевает наличие определенных принципов и способов организации деятельности, в данном случае деятельности, связанной с анализом и синтезом систем.

Рассмотрим фундаментальные принципы системного подхода.

Принцип системности утверждает первичность системы по отношению к ее элементам, однако учитывает взаимозависимость системы и элементов. Системный подход основан на предположении, что целое – *система* – обладает качествами, не свойственными ее элементам: при объединении элементов в систему благодаря системообразующим связям возникают новые качественные характеристики. Несводимость свойств системы к сумме свойств ее элементов, возникновение новых свойств при объединении элементов в систему называется *эмерджентностью*, или *системным эффектом*.

Принцип цели определяет генеральным назначением любой системы целенаправленное функционирование во взаимодействии с внешней средой, устойчивое при любых возмущениях.

Принцип иерархичности отражает подчиненный характер взаимозависимости элементов и системы в целом. Методы декомпозиции и агрегирования являются средством реализации данного принципа в процессе проектирования систем: при проектировании идут от общего к частному; общей задаче, сформулированной для системы в целом, подчиняют задачи, решаемые ее составляющими.

Принцип многомодельности (множественности описаний) предполагает, что полное описание системы может быть достигнуто только путем привлечения различных типов моделей, каждая из которых отражает какой-то отдельный аспект системы, т.е. сложная система представима конечным множеством моделей, каждая из которых отражает определенную грань ее сущности.

ПРИМЕР реализации принципов иерархичности и многомодельности



Процедуры системного подхода делятся на процедуры анализа и синтеза. *Процедуры анализа* заключаются в исследовании системы или ее описания, направленном на получение определенной информации о свой-

ствах системы. Эти процедуры применяются в том случае, когда система или ее модель реально существуют и задача сводится к их изучению. Чаще приходится сталкиваться с противоположной задачей: системы нет, и ее предстоит создать, спроектировать – это задача синтеза.

Методология системного подхода при решении *задач анализа* систем сводится к тому, что исследования объекта ориентируются на раскрытие его интегративных качеств, на выявлении многообразных связей и механизмов, обеспечивающих эти качества. Анализ, имеющий своей целью установление соответствия двух различных описаний (моделей) одной и той же системы, например, соответствующих различным уровням иерархии описаний, или различным аспектам, называется *верификацией*.

Процедуры синтеза (рис. 14) заключаются в создании описаний объекта в соответствии с принципом многомодельности.

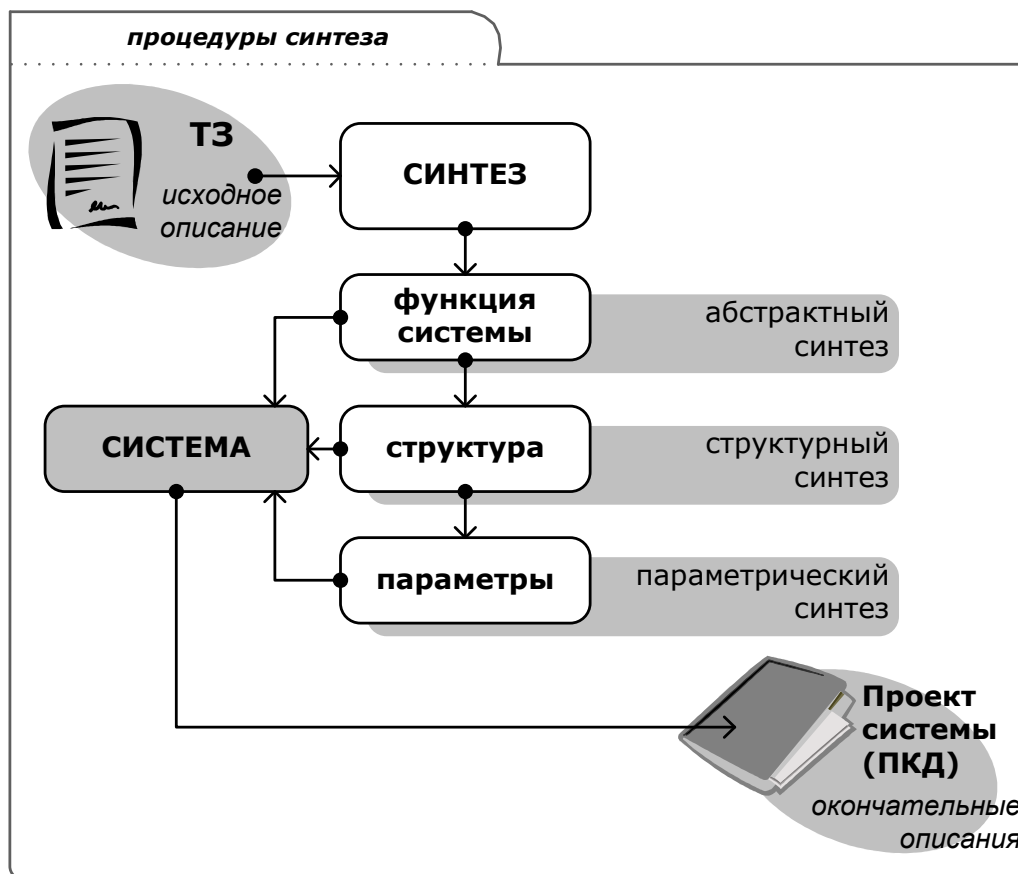


Рис. 14

Методология системного подхода при решении *задач синтеза* систем состоит в следующем: задача проектирования системы расчленяется на подзадачи проектирования ее элементов, причем каждый из элементов

рассматривается не изолированно, а во взаимодействии с другими элементами. Решение подзадач должно происходить при условии обеспечения интегративных качеств функционирования всей системы. Для выполнения этого требования необходим единый план проектирования, связывающий все этапы, начиная от исследовательской проработки до изготовления и эксплуатации.

Для решения задачи синтеза системы необходимо выполнить три шага (см. рис. 14): выявить функцию системы, разработать структуру системы и определить параметры системы так, чтобы получить желаемое качество проекта.

При выявлении глобальной функции системы она рассматривается на абстрактном уровне в виде черного ящика, о котором известно только то, что он будет делать. Затем осуществляется преобразование функционального описания в структурное. Как правило, этот процесс осуществляется декомпозицией глобальной функции системы на подфункции с повторением, при необходимости, этого деления до тех пор, пока не будут получены элементарные функции, структуры которых очевидны или уже реализованы. Этот этап порождения структуры, реализующий необходимые функции, называют *структурным синтезом*. На третьем этапе остается подобрать параметры системы так, чтобы достичь желаемого качества ее работы.

Несмотря на внешнюю простоту и логичность на практике процедуры формального синтеза оказываются исключительно сложными. Специалисты в теории систем даже склоняются к мысли, что в общей постановке задача строгого синтеза неразрешима. Только для узкого класса технических систем разработаны и используются на практике методы формального синтеза. К ним можно отнести пассивные электрические цепи (в частности, фильтры), типовые системы автоматического управления, цифровые автоматы, комбинационные и несложные регистровые схемы.

В остальных случаях у инженера-системотехника нет альтернативы методу проб и ошибок, реализованному в алгоритме итерационного проектирования: сначала создается базовый вариант проекта, а затем он улучшается, пока не будет достигнуто желаемое качество работы (рис. 15).

Если глобальную функцию системы записать в виде $Y = F(X, Q, t)$, то понятно, что улучшить качество системы можно тремя способами:

- воздействием на вектор Q – параметрический метод;
- изменением функции F – схемотехнический метод;
- воздействием на внешние параметры X .

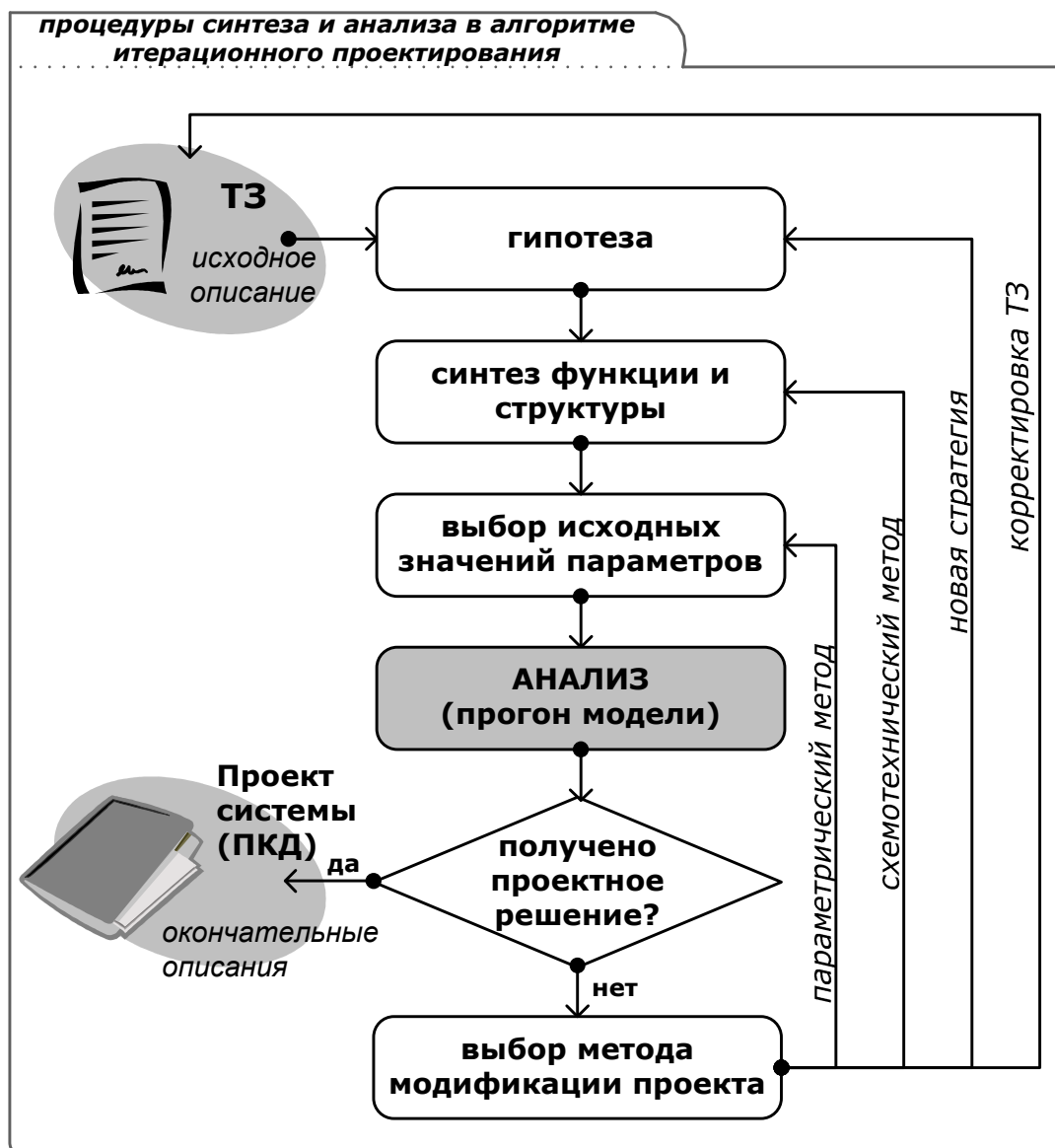


Рис. 15

Параметрический метод есть не что иное, как улучшение качества входящих в систему элементов. Например, если в разработке есть резисторы с большим технологическим разбросом и они ухудшают работу, то их можно заменить высокоточными (прецизионными) резисторами, для критичной по быстродействию рабочей станции в сети – увеличить оперативную память, для повышения точности управления выбрать средства контроля параметров с меньшей погрешностью измерений и т.п. Этот метод самый простой и потому рекомендуется начинать «борьбу за качество» именно с него. Он не требует никаких изменений модели, достаточно выполнить ее прогон на новом наборе Q -параметров.

Схемотехнический метод требует введения в разработку структурной избыточности. Арсенал этих средств – это различного рода автоподстройки, резервирование, введение отрицательной обратной связи и т.п. Понятно, что изменения, внесенные в структуру системы, потребуют и аналогичной коррекции ее модели. Схемотехнические приемы весьма эффективны, но более дорогостоящи. К ним обычно обращаются после того, как исчерпаны возможности параметрического метода.

Если модификация параметров и структуры направлены на поиск наилучшего значения некоторого показателя качества, то процедура синтеза называется процедурой *оптимизации*.

У третьего метода нет общего названия. Некоторые авторы называют его методом уменьшения отрицательного влияния внешней среды. Сюда входят такие трудоемкие и дорогие мероприятия, как термостатирование, экранирование, замена коаксиального кабеля оптоволоконном, развязки по нагрузке, повышение требований к источникам питания и генераторам входных воздействий и т.п.

В случае безуспешности всех трех методов улучшения проекта приходится прибегать к отказу от первоначального варианта технического задания и корректировать его в сторону упрощения и обеспечения реализуемости проекта.

Резюмируя вышесказанное можно утверждать, что только системный подход к решению задач анализа, синтеза и улучшение проекта может гарантировать хорошее качество разрабатываемой системы [2, 5, 9 – 11].

2. Методология системного анализа

Как уже отмечалось, основой методологии технической системотехники является системный анализ: совокупность методов и технологий, использующих в себе системный подход (см. рис. 1). Реализация системного подхода в проектировании систем требует:

- выбора способов описаний системы и формализации таких описаний (моделирование систем);
- оптимизации характеристик системы с учетом ограничений в модельных вычислительных экспериментах (поиск и принятие решений с использованием методов исследования операций и математической статистики).

В современном понимании системный анализ представляет собой синтез идей и методов общей теории систем с возможностями вычислительной техники [15].

2.1. Моделирование систем

Моделирование систем включает в себя последовательное осуществление этапов:

- постановка задачи и разработка концептуальной (содержательной) модели системы;
- разработка математической модели;
- проверка адекватности и корректировка модели.

Постановка задачи и разработка концептуальной модели является чрезвычайно важным этапом проектирования систем. Для детализации целей функционирования изучаются информационные и материальные потоки внутри системы; ее связи с внешней средой; возможные способы организация управления данной системой (объектом), а также закономерности функционирования системы: показатели качества или критерии эффективности, и внешние факторы, которые влияют на эти характеристики.

После сбора результатов обследования проводится их анализ, в результате которого обнаруживаются существенные факторы и переменные, обосновывается выбор тех или иных показателей качества функционирования системы, а также существенных внешних факторов, структура самой системы, состав ее элементов, их взаимосвязи, внутренние переменные.

Результатом выполнения этого этапа есть концептуальная модель исследуемой системы, в которой в содержательной форме, с использованием простейшей логики, описывается состав системы и структура, перечень основных показателей качества, переменных, контролируемых и неконтролируемых внешних факторов, а также их взаимосвязей с показателями качества системы, перечень стратегий управления или решений, которые нужно получить.

Именно на концептуальном уровне моделирования решаются задачи декомпозиции (разбиения) на подсистемы и агрегирования (объединения) подсистем в систему.

Разработка математической модели. После получения концептуальной модели системы (содержательной постановки задачи) нужно построить ее математическую модель, которую можно использовать в анализе и при принятии решений. Этот процесс называется *формализацией*.

Проверка адекватности и корректировка модели. Любая модель только частично отражает свойства реальной системы, поэтому необходимо проводить проверку степени соответствия (адекватности) модели и объекта моделирования путем сравнения выходных характеристик модели $Y_{\text{мод}}$ с фактическими характеристиками объекта $Y_{\text{реал}}$ при изменении значений внешних и внутренних параметров системы.

В качестве меры адекватности модели используют, например:

- абсолютное отклонение $\varepsilon_a = |y_{\text{мод}_i} - y_{\text{реал}_i}| \leq \varepsilon_{a_{\text{дон}}}$;
- относительное отклонение $\varepsilon_b = \left| \frac{y_{\text{мод}_i} - y_{\text{реал}_i}}{y_{\text{реал}_i}} \right| \cdot 100\% \leq \varepsilon_{b_{\text{дон}}}$;
- вероятностную оценку $P(\varepsilon_a \leq \varepsilon_{a_{\text{дон}}}) \geq P_{\text{зад}}$ или $P(\varepsilon_b \leq \varepsilon_{b_{\text{дон}}}) \geq P_{\text{зад}}$,

где $\varepsilon_{a_{\text{дон}}}$ и $\varepsilon_{b_{\text{дон}}}$ – допустимое отклонение, которое задается разработчиком и определяет степень адекватности модели; $P_{\text{зад}}$ – заданная величина вероятности выполнения неравенства $\varepsilon_a \leq \varepsilon_{a_{\text{дон}}}$.

Если выбранная мера адекватности модели не удовлетворяет заданным ограничениям, то это свидетельствует о том, что при разработке модели были учтены не все значимые факторы или исходная концептуальная модель построена на неверной гипотезе. Корректировка может потребовать проведения дополнительных исследований на расширенном диапазоне варьируемых факторов, уточнения набора переменных и структуры модели [16, 18].

2.1.1. Элементы теории множеств

– Ты когда-нибудь видела, как рисуют множество?
– Множество чего? – спросила Алиса.
– Ничего. – отвечала Соня. – Просто множество!

Льюис Кэрролл
«Алиса в стране чудес»

Наиболее общие концептуальные описания систем создаются с использованием языка теории множеств. Рассмотрим основные элементы этой теории.

Множество это совокупность элементов, объединенных по какому-либо признаку. Элементами множества могут быть любые объективные и субъективные понятия, объединяемые в соответствии с некоторым законом, правилом, признаком и т.д. Множество, элементами которого являются множества, обычно называют *классом*. Множество, не содержащее элементов, называется *пустым*: $B = \emptyset$.

Через « $\in$ » обозначается *отношение принадлежности*, т.е. « $x \in A$ » означает, что элемент x принадлежит множеству A . Если x не является элементом A , то пишется « $x \notin A$ ».

Чтобы задать некоторое множество A , нужно указать, какие элементы ему принадлежат. Это можно сделать различными способами:

- перечислением $A: = \{a_1, a_2, \dots, a_n\}$, например, $A_5: = \{1, 2, 3, 4, 5\}$;
- характеристическим *предикатом* (правилом) $A: = \{x \mid P(x)\}$, например, $A_5: = \{n \mid n \in \mathbb{N} \ \& \ n < 6\}$;
- порождающей процедурой $A: = \{x \mid x: = f\}$, например, $A_5: = \{n \mid \text{for } n \text{ from } 1 \text{ to } 5 \text{ yield } n\}$.

Перечислением можно задавать только конечные множества. Бесконечные множества задаются характеристическим предикатом или порождающей процедурой. Характеристический предикат это некоторое условие, выраженное в форме логического утверждения или процедуры, возвращающей логическое значение. Если для данного элемента условие выполнено, то он принадлежит определяемому множеству, в противном случае – не принадлежит. Порождающая процедура это процедура, которая, будучи запущенной, порождает некоторые объекты, являющиеся элементами определяемого множества.

Множества A и B считаются равными, если они состоят из одних и тех же элементов: $A = B$. В противном случае $A \neq B$.

Множества могут содержать *подмножества*. Если каждый элемент множества B является также и элементом множества A , то говорят, что множество B является подмножеством множества A . Обозначение этого отношения записывается через *отношение включения*: « $B \subset A$ », означающее, что каждый элемент B является элементом A , но не наоборот (примеры 1 и 2 на рис. 16). *Семейство* всех подмножеств множества A обозначается как $P(A)$.

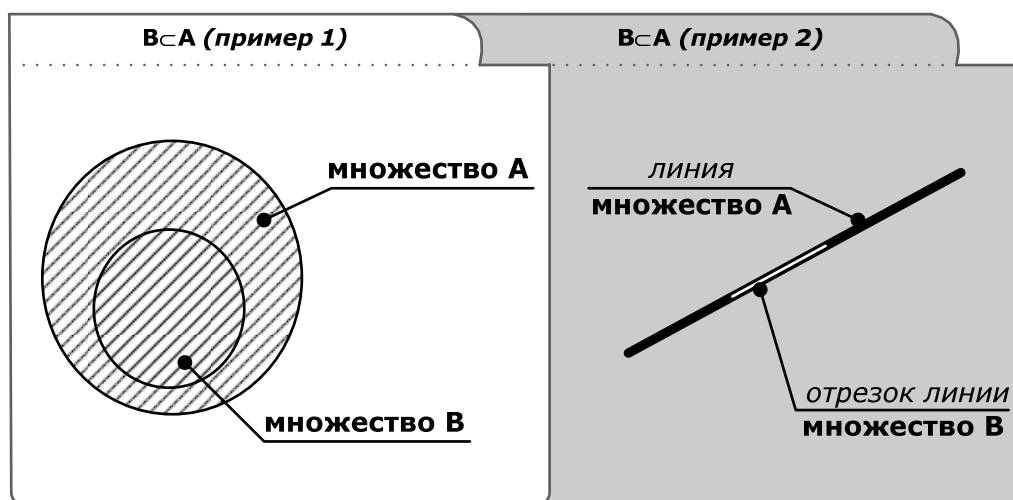


Рис. 16

К основным операциям над множествами относятся:

- 1) объединение (сумма, дизъюнкция): $A \cup B = X := \{x \mid x \in A \text{ или } x \in B\}$;
- 2) пересечение (произведение, конъюнкция): $A \cap B = X := \{x \mid x \in A \text{ и } x \in B\}$ (рис. 17);
- 3) разность: $A \setminus B = X := \{x \mid x \in A \text{ и } x \notin B\}$;
- 4) симметрическая разность: $(A \setminus B) \cup (B \setminus A) = A \Delta B = X := \{x \mid x \in A \text{ и } x \notin B, x \in B \text{ и } x \notin A\}$ (рис. 18);
- 5) дополнение: $\bar{A} := \{x \mid x \notin A\}$;
- 6) прямое, или декартово, произведение имеет результатом множество упорядоченных пар элементов множеств: $A \times B = X := \{(a, b) \mid a \in A, b \in B\}$. Например, если $A = \{a_1, a_2\}$ и $B = \{b_1, b_2, b_3\}$, то $A \times B = \{(a_1, b_1), (a_1, b_2), (a_1, b_3), (a_2, b_1), (a_2, b_2), (a_2, b_3)\}$.

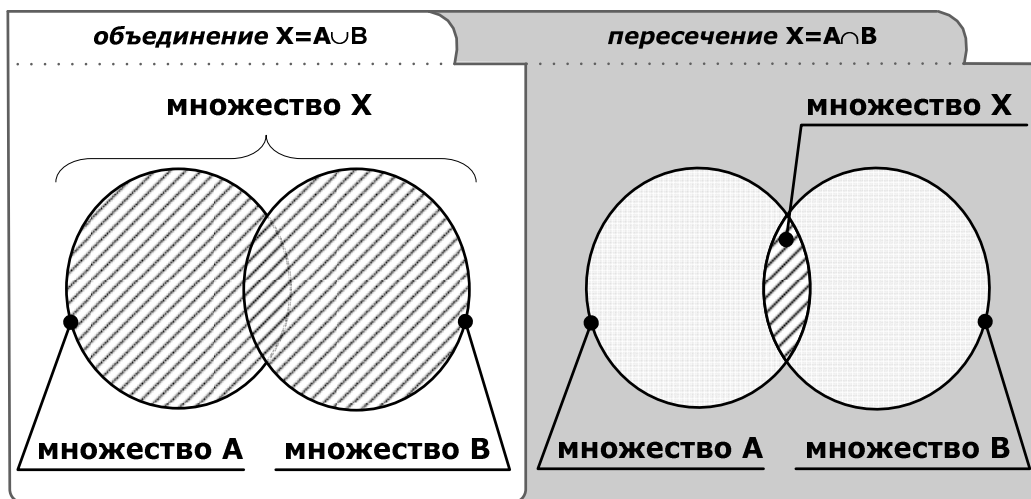


Рис. 17

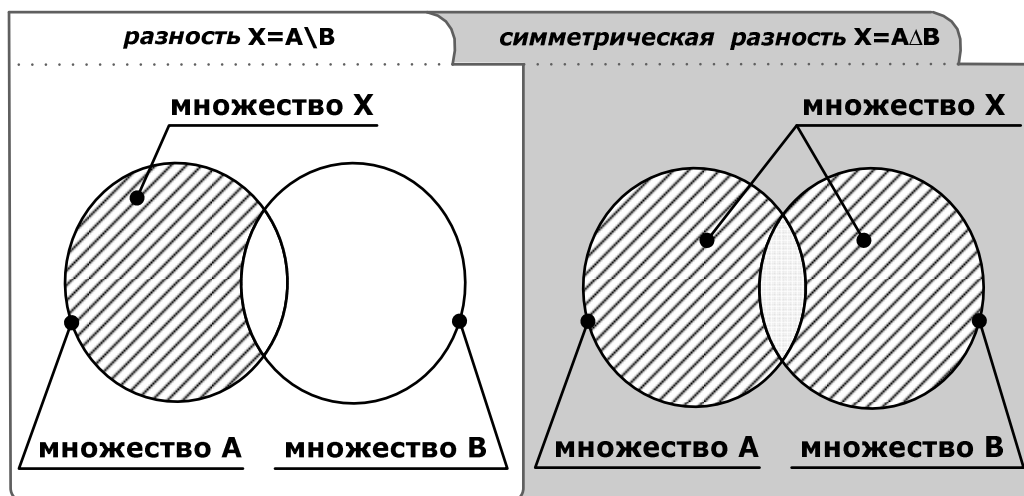


Рис. 18

Если A и B непустые множества, то если каждому элементу $x \in A$ ставится в соответствие единственный элемент $y \in B$, то говорят, что задано *отображение* F множества A в множество B . Форма записи отображения множеств имеет вид: $F: A \rightarrow B$. Иначе говорят, что задана функция F с областью определения A и областью значения B . При этом элемент y называется *образом* элемента x (обозначается $y = F(x)$), а элемент x – *прообразом* элемента y . Отображение может быть задано таблицей, списком и т.п.

Параметризация системы (см. раздел 1.1, рис. 4) позволяет сформировать *базисные множества*, из которых непосредственно образуется конструкция математической модели. К этим множествам относятся:

- 1) $X = \{x_1, x_2, \dots, x_n\}$ – множество внешних целенаправленных (управляющих) воздействий;
- 2) $G = \{g_1, g_2, \dots, g_s\}$ – множество внешних возмущающих воздействий;
- 3) $Y = \{y_1, y_2, \dots, y_k\}$ – множество выходных параметров (реакций);
- 4) $A = \{a_1, a_2, \dots, a_h\}$ – множество внутренних состояний системы, каждое из которых характеризуется собственным множеством значений внутренних параметров системы $Q_{a_i} = \{Q_{1_i}, Q_{2_i}, \dots, Q_{m_i}\}$;
- 5) вспомогательное множество моментов времени функционирования системы $T = \{t_i\}$.

Концептуальную основу модели системы образуют операторы перехода и выхода. Для их представления используется теоретико-множественные конструкции отображений базисных множеств.

Оператор перехода имеет вид:

$$\Psi: T \times X \times G \times A \rightarrow A, \quad (1)$$

где левое множество (прообраз отображения) представляет собой декартово произведение базисных множеств.

Декартовым произведением множеств T, X, G, A называется множество $T \times X \times G \times A = \{(t, x, g, a) | t \in T, x \in X, g \in G, a \in A\}$, т.е. множество, состоящее из элементов, каждый из которых является сочетанием элементов базисных множеств. Отображением из множества $T \times X \times G \times A$ в множество A называется такое отношение Ψ , что

$$\forall (t, x, g, a) \in T \times X \times G \times A \exists a \in A: (t, x, g, a) \Psi a.$$

Каждому элементу $(t, x, g, a) \in T \times X \times G \times A$ при отображении Ψ в соответствие ставится элемент $a \in A$ для момента времени $t_i > t$. Отображе-

нием Ψ задается закон преобразования множеств. В явном виде это совокупность детерминированных или стохастических зависимостей, полученных с использованием типовых математических схем.

Таким образом, каждому возможному сочетанию управляющего, возмущающего воздействий и состояния системы в заданный момент времени оператор перехода ставит в соответствие состояние системы в следующий момент времени: $\psi : (t, x, g, a) \rightarrow a(t_i), t_i > t$.

Оператор выхода имеет вид:

$$\Phi : T \times X \times G \times A \rightarrow Y. \quad (2)$$

Каждое возможное сочетание управляющего, возмущающего воздействий и состояния системы в заданный момент времени однозначно определяет реакцию системы в этот момент времени: $\phi : (t, x, g, a) \rightarrow y(t)$. В явном виде оператор выхода это также совокупность зависимостей, полученных с использованием конкретного вида моделирования [3].

2.1.2. Типовые математические схемы моделирования систем

Математической схемой называется конкретная математическая форма, используемая при разработке математической модели системы. Математическая схема является промежуточным звеном при переходе от концептуальной модели к формальной математической модели, т.е. является средством формализации.

Математические схемы моделирования систем делят [12, 27]:

- по признаку учета случайной природы воздействий и взаимодействий в системе на *детерминированные* (не учитывают) и *стохастические* (учитывают);
- по типу переменных модели на *дискретные* и *непрерывные*.

Классификация типовых математических схем моделирования систем представлена на рис. 19.

Схемы *непрерывно-детерминированного* вида или *D-схемы* (англ. dynamic) позволяют получить математическую модель системы в виде дифференциальных уравнений. Математические модели такого вида отражают динамику функционирования системы, т.е. описывают изменение ее состояния в непрерывном времени. Наиболее широкое применение D-схемы находят в проектировании и анализе систем автоматического управления. Применение моделей данного типа для решения практических задач анализа поведения систем рассматривается в лабораторной работе «Агрегирование и эквивалентные преобразования непрерывно-детерминированных

типовые математические схемы моделирования систем

детерминированные → F-схемы (конечные автоматы) (C)
 дискретные → F-схемы (конечные автоматы) (C)
 дискретные → P-схемы (вероятностные автоматы) (C)
 непрерывные → D-схемы (диф. уравнения) (CM)
 непрерывные → N-схемы (Сети Петри) (CM)
 непрерывные → Q-схемы (модели массового обслуживания) (M)
 стохастические → P-схемы (вероятностные автоматы) (C)
 стохастические → Q-схемы (модели массового обслуживания) (M)

(M) - изучаются в дисциплине «Моделирование»
 (C) - изучаются в дисциплине «Системотехника»
 (CM) - изучаются в обеих дисциплинах

Схемы *дискретно-детерминированного* вида или *F-схемы* (англ. finite automaton) позволяют получить модель в виде абстрактного конечного автомата – дискретного потактного преобразователя информации с памятью, реакция (выходной параметр) которого в каждый момент дискретного времени определяется текущим значением внешнего воздействия x_i и внутренним состоянием (памятью) a_i .

42

кретных систем, проявляющих статистически закономерное случайное поведение, для выяснения алгоритмических возможностей таких систем в обоснования границ целесообразности их использования, а также для решения задач синтеза по выбранному критерию дискретных стохастических систем, удовлетворяющих заданным ограничениям.

Моделирование систем абстрактными конечными и вероятностными автоматами рассматривается в разделе 2.4.

Схемы *непрерывно-стохастического* вида или *Q-схемы* (англ. queueing system) позволяют получить модель в терминах теории массового обслуживания. В теории массового обслуживания изучаются системы, на вход которых в случайные моменты времени поступают заявки на обслуживание. Заявка обслуживается в системе путем предоставления ей ресурсов и покидает систему. Помимо случайного появления заявок на обслуживание и случайной длительности обслуживания каждой заявки для систем массового обслуживания характерным является наличие очередей, в которых заявки ждут момента освобождения ресурсов, занятых обслуживанием других заявок. Разработка и применение вероятностных моделей теории массового обслуживания является предметом дисциплины «Моделирование».

Для описания систем, в которых имеет место протекание нескольких взаимодействующих параллельных процессов, а также сочетается непрерывность изменения параметров с дискретной сменой состояний, используются *сетевые схемы* или *N-схемы* (англ. network). Для моделирования информационных и вычислительных систем основной разновидностью моделей, использующих данную схему, являются сети Петри [27].

Элементы формализации модельных описаний и моделирование систем сетями Петри рассматривается в разделе 2.5.

2.2. Поиск и принятие решений

Это такая штука, которую вы спокойно можете объяснить два раза, не опасаясь, что кто-нибудь поймет, о чем вы говорите.

А. Милн
«Вини-Пух и все-все-все»

При постановке задачи *поиска и принятия решений* рассматриваются:

1) множества внешних целенаправленных воздействий системы и подмножество внутренних параметров, значения которых можно целенаправленно изменять – *управляющие переменные*: $X = \{x_n\}$, $Q_{cont} \in Q = \{Q_m\}$;

2) множество управляемых параметров системы, которые зависят от управляющих переменных – *выходные переменные*, или *решения*: $Y = \{y_k\}$;

3) параметры, значения которых не регулируются: подмножество неуправляемых внутренних параметров Q_{uncont} ($Q_{uncont} \cup Q_{cont} = Q = \{Q_m\}$) и множество внешних возмущений $G = \{G_s\}$ системы;

4) *ограничения* на управляющие и выходные переменные, а также ресурсы системы, которые задаются в виде ресурсных функций от управляющих переменных и выходных переменных;

5) *целевая функция*: критерий эффективности f , который зависит от принятых стратегий, параметров системы и возмущений.

Как следует из приведенного описания, в задачах поиска и принятия решений в качестве исходных данных используются базисные множества концептуальной модели системы 1) – 3), а также, дополнительно, ограничения 4) и целевая функция 5).

Задача принятия решений может быть одно- и многокритериальной. В случае однокритериальной задачи постановка имеет следующий вид:

$$\begin{aligned} F = f(X, Y, Q, G) \rightarrow \max(\min) \\ X, Q_{cont} \\ \text{при ограничениях} \\ h_i(X, Q, G) \leq b_i, \end{aligned} \quad (3)$$

где $f(X, Y, Q, G)$ – целевая функция (критерий эффективности); $h_i(X, Q, G)$ – функция ограничений i -той управляющей, выходной переменной или i -того ресурса; b_i – ограничение i -той управляющей, выходной переменной или запас i -того ресурса в системе. Искомое решение может принимать разные формы: аналитическую, численную или алгоритмическую (в виде набора процедур, правил, и т.п.).

Примером однокритериальной задачи поиска решения может служить задача оптимизации характеристик функционирования устройства обслуживания с разделяемой памятью (устройства параллельного обслуживания – рис. 20). Пусть в очередь устройства поступает поток заявок основного обслуживания с фиксированной интенсивностью $\lambda_{оч}$ и несколько дополнительных потоков с собственными интенсивностями ($\lambda_1 \dots \lambda_n$). Цель поиска решения: определить максимально возможную интенсивность дополнительных потоков λ_i , равную для всех ($\lambda_1 = \lambda_2 = \dots = \lambda_n = \lambda_i$), с учетом заданных ограничений по коэффициенту использования устройства $KИ_{дон}$ (доля суммарного времени работы устройства от общего времени функционирования системы), длины очереди устройства $L_{дон}$ и объему памяти, выделяемой для обслуживания одной заявки m_i .

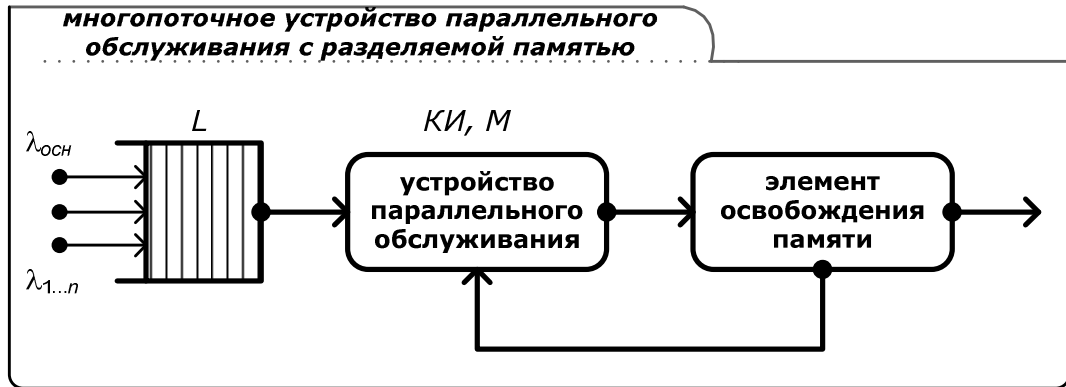


Рис. 20

Постановка задачи поиска решений для заданного примера имеет следующий вид:

- 1) управляющие переменные: собственные интенсивности поступления заявок дополнительных потоков заявок $X = \{x_n\} = \{\lambda_n\}$ и объемы памяти, выделяемые для обслуживания заявок $Q_{cont} \in Q = \{m_i\}$;
- 2) неуправляемые параметры: интенсивность потока основного обслуживания $G = \{g_s\} = \lambda_{осн}$ и объем памяти устройства $Q_{uncont} = M$;
- 3) целевая функция: максимально возможная равная для всех дополнительных потоков $F = f(X, Y, Q, G) \rightarrow \max(\lambda_i)$.

Ограничениями по условию данной задачи выступают предельно допустимый минимальный объем памяти, выделяемый для обслуживания одной заявки $m_i \geq m_{дон}$; общее ограничение по разделяемому объему памяти $\sum m_i \leq M$; ограничения по допустимым значениям коэффициента использования устройства и длине очереди: $L \leq L_{дон}$ и $КИ \leq КИ_{дон}$. Реальные значения L и $КИ$ являются, кроме того, выходными параметрами рассматриваемой системы: $Y = \{y_k\} = \{КИ, L\}$.

Для многоцелевых систем степень достижения каждой частной цели характеризуется собственным показателем эффективности, т. е. задача становится многокритериальной. Пусть система имеет k целей, и эффективность ее функционирования, соответственно, характеризуется целевыми функциями $F_1, F_2, \dots, F_k$. Теоретически можно представить случай, когда на множествах X и Q_{cont} найдется единственное решение, обеспечивающее наилучшие значения всех критериев (максимальные или минимальные значения всех целевых функций). Однако на практике такие случаи практически не встречаются, и возникает задача выбора альтернатив. Оптимизация общего решения путем выбора альтернатив в условиях многокритериальности может быть получена несколькими способами.

*ПРИМЕР неоднозначности выбора решения в двухкритериальной
однофакторной задаче*



1. Сведение многокритериальной задачи к однокритериальной. Используется в случае, когда частные критерии эффективности неравнозначны. В этом способе вводится обобщенный показатель в виде аддитивной или мультипликативной функции:

$$F^* = \sum_{i=1}^k \alpha_i F_i \text{ или } F^* = \prod_{i=1}^k \beta_i F_i,$$

где α_i и β_i – весовые коэффициенты, отражающие значимость каждой целевой функции.

Условия нормировки коэффициентов имеют вид:

$$\sum_{i=1}^k \alpha_i = 1 \text{ и } \prod_{i=1}^k \beta_i = 1$$

для аддитивной и мультипликативной функции соответственно. Для обобщенного критерия эффективности F^* задача решается как однокритериальная в постановке вида (3).

*ПРИМЕР сведения двухкритериальной однофакторной задачи
к однокритериальной*



Примером многокритериальной задачи поиска решения, решаемой методом сведения к однокритериальной с использованием аддитивной целевой функции, может служить модификация рассмотренного выше примера оптимизации характеристик функционирования устройства обслуживания с разделяемой памятью, с дополнительным условием: рассматриваемые потоки имеют различные приоритеты. В этой постановке целью поиска решения является определение максимально возможной интенсивности каждого из дополнительных потоков λ_i с учетом их приоритетов. Целевая функция для этого случая будет иметь вид:

$$F^* = f(X, Y, Q, G) \rightarrow \max \left(\frac{\sum_{i=1}^n \alpha_i \cdot \lambda_i}{n} \right),$$

где α_i – весовой коэффициент, зависящий от приоритета i -того дополнительного потока.

*ПРИМЕРЫ сведения многокритериальных задач к
однокритериальным*



2. *Поиск условного экстремума.* Данный способ используется в случае, когда частные критерии эффективности существенно неравнозначны (имеется критерий с максимальным приоритетом) и/или взаимно противоречивы (при возрастании одних другие убывают и наоборот). В этом случае ставится задача нахождения условного экстремума основного критерия при заданных ограничениях на другие. Если из k целевых функций $F_1, F_2, \dots, F_k$ основным критерием является F_1 , то постановка задачи имеет вид:

$$F_1 = f(X, Y, Q, G) \rightarrow \max(\min)_{X, Q_{cont}} \quad (4)$$

при ограничениях

$$h_i(X, Q, G) \leq b_i, F_j \leq c_j, j = 2, 3, \dots, k.$$

*ПРИМЕРЫ решения двухкритериальной однофакторной задачи
методами поиска условного экстремума*



Примером задачи многокритериальной оптимизации, решаемой методом поиска условного экстремума, может являться задача оптимизации функционирования многопоточного устройства обслуживания с ограничением времени ожидания неприоритетных запросов. Пусть заявки трех входных потоков поступают в общую очередь устройства, причем заявки первого потока имеют максимальный относительный приоритет, а для неприоритетных заявок второго и третьего потоков время ожидания обслуживания в очереди ограничено, и при нарушении ограничения заявки теряются. Время обслуживания заявок в устройстве является случайной величиной, распределенной по заданному закону. Постановка задачи оптимизации по форме (4) требует в данном случае задания основной целевой функции и ограничений по прочим целевым функциям, например, следующего вида:

$$F_1 = f(X, Y, Q, G) \rightarrow \max(\lambda_1);$$

$$F_2 = f(X, Y, Q, G) = \left(\frac{\sum_{i=1}^k t_{ож_i}}{k} \right) \leq t_{дон2};$$

$$F_3 = f(X, Y, Q, G) = \left(\frac{\sum_{i=1}^m t_{ож_i}}{m} \right) \leq t_{дон3},$$

где λ_1 – интенсивность поступления заявок с первым, приоритетным, потоком; k и m – количество заявок, поступивших в очередь обслуживания со вторым и третьим потоками; $t_{ож_i}$ – время ожидания в очереди i -той заявки; $t_{дон2}$ и $t_{дон3}$ – ограничения времени ожидания заявок второго и третьего потоков. В качестве второго и третьего критерия выступают ограничения среднего значения времени ожидания в очереди заявок соответствующих потоков, т.е. закладывается допуск на некоторое количество потерь. Общими ограничениями вида $h_i(X, Q, G) \leq b_i$ в данной задаче могут выступать ограничения по допустимым значениям коэффициента использования устройства и длине очереди: $L \leq L_{дон}$ и $KI \leq KI_{дон}$, а также ограничения по допустимому числу суммарных потерь заявок потоков 2 и 3: $V_{lost} \leq V_{lost\ дон}$.

3. *Нахождение множества Парето.* Используется в случае, когда все критерии равнозначны. Как следует из названия, данный способ, в отличие от рассмотренных выше, предполагает поиск не единственного решения, а множества предпочтительных альтернатив. Предпочтительной альтернативой называется сочетание значений частных целевых функций $F_1, F_2, \dots, F_k$, при котором имеет место *оптимальность Парето* – такое состояние системы, при котором значение каждого частного критерия, описывающего состояние системы, не может быть улучшено без ухудшения других критериев.

Алгоритм способа состоит в попарном сравнении альтернатив. Предпочтение одной альтернативе перед другой отдается, только если первая по всем k критериям лучше второй. Если же предпочтение хотя бы по одному критерию расходится с предпочтением по другому, то эти альтернативы признаются несравнимыми. В результате попарного сравнения альтернатив все худшие по всем критериям альтернативы отбрасываются, а все оставшиеся, несравнимые между собой, принимаются и образуют множество Парето.

*ПРИМЕР решения двухкритериальной однофакторной задачи
методом поиска множества Парето.*



Рассмотрим формальную постановку задачи поиска по методу Парето для двухкритериальной однофакторной системы. Пусть требуется максимизация двух целевых функций системы F_1 и F_2 , зависящих от единственного входного воздействия системы x , т.е. необходимо, чтобы $F_1 = f_1(x) \rightarrow \max$ и $F_2 = f_2(x) \rightarrow \max$. Тогда постановка задачи требует нахождения множества пар предпочтительных альтернатив $\{P(x)\}$ вида:

$$\left\{ (f_1(x_i), f_2(x_i)) \mid \text{не существует } x \in X \text{ } f_1(x) \geq f_1(x_i) \text{ и } f_2(x) \geq f_2(x_i) \right\} \quad (5)$$

при ограничениях $h_i(X, Q, G) \leq b_i$.

Рассмотрим пример снижения размерности множества альтернативных вариантов в задаче поиска решения путем построения множества Парето. Пусть в процессе проектирования ВС требуется осуществить выбор устройства обслуживания (УО) из 10-ти альтернативных вариантов по двум критериям: интенсивность обслуживания (F_1) и надежность функционирования (F_2). Приведенные значения частных критериев, рассчитанные как отношения значений критериев для рассматриваемых вариантов УО к максимальному значению критерия, даны в табл. 2.

Таблица 2

Частный критерий	Приведенные значения для УО									
	УО ₁	УО ₂	УО ₃	УО ₄	УО ₅	УО ₆	УО ₇	УО ₈	УО ₉	УО ₁₀
F_1	0,6	0,4	1	0,3	1	0,1	0,2	0,4	0,6	0,7
F_2	0,6	0,2	0,1	0,7	0,4	0,4	1	0,4	0,8	0,2

Решение задачи можно представить графически в плоскости двух рассматриваемых критериев. Для поиска решения по методу Парето из каждой точки множества альтернатив строятся два взаимно перпендикулярных вектора с направлением, совпадающим с направлением оптимизации значения каждого из критериев. В нашем случае оба критерия подлежат максимизации, т.е. вектора должны быть направлены по возрастанию критериев. Если в угол, ограничиваемый векторами, попадают точки альтернативных вариантов, значит, одно или оба значения для рассматриваемого

варианта не являются оптимальными по Парето и могут быть улучшены, следовательно, этот вариант из числа рассматриваемых следует исключить. Исключение вариантов осуществляют до тех пор, пока в числе рассматриваемых решений не останутся точки с отсутствием альтернативных решений в углах, ограничиваемых векторами. Эти решения и составят множество Парето. Построение множества Парето по данным табл. 2 показано на рис. 21.

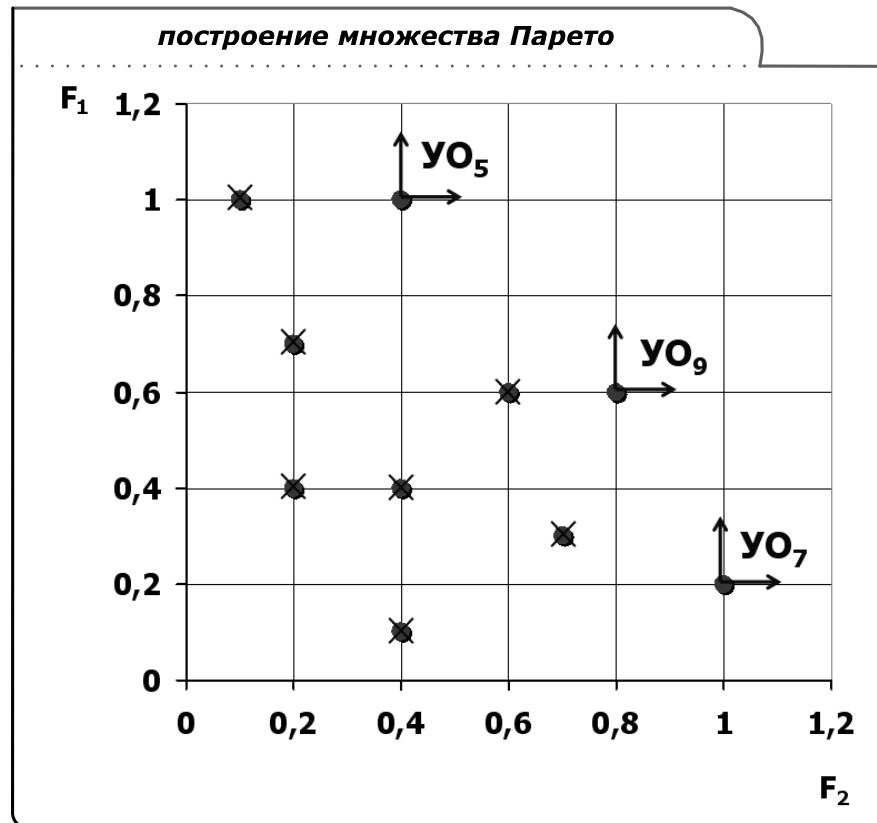


Рис. 21

Как следует из результата построения, размерность множества альтернативных решений сократилось с 10-ти до 3-х, т.е. полученное множество Парето имеет вид:

$$\{P\} = \{(F_1; F_2)\} = \{(1; 0,4), (0,6; 0,8), (0,2; 1)\}.$$

Окончательный выбор единственного варианта УО может быть осуществлен по методу сведения двухкритериальной задачи к однокритериальной с введением целевой аддитивной функции вида:

$$F^* = \alpha_1 F_1 + \alpha_2 F_2 \rightarrow \max,$$

где F^* – обобщенный критерий; α_1 и α_2 – весовые коэффициенты значимости критериев – интенсивности обслуживания и надежности УО соответственно.

Зададим, с учетом условия нормировки, $\alpha_1 = 0,55$ и $\alpha_2 = 0,45$. Тогда значения F^* для УО из множества Парето будут равны:

$$F_{yO_5}^* = 0,73; F_{yO_9}^* = 0,69; F_{yO_7}^* = 0,56,$$

т.е. оптимальным следует считать выбор пятого устройства.

Для многокритериальной задачи произвольной размерности при построении множества Парето рассматриваются сочетания альтернативных значений критериев размерностью k [14, 17, 28].

Частные содержательные случаи постановки задач поиска и принятия решений называются задачами *исследования операций* и будут рассмотрены в пункте 2.2.2.

2.2.1. Общая характеристика методов

Для нахождения оптимального решения задач (3) – (5) используют методы дифференциального исчисления и методы теории принятия решений, также называемые методами математического программирования. *Математическое программирование* это раздел математики, занимающийся разработкой методов отыскания экстремальных значений функции, на аргументы которой наложены ограничения. Для решения задач исследования операций используются:

1) *линейное программирование*, если функция $f(X, Y, Q, G)$ линейна относительно управляющих переменных X ;

2) *целевое программирование* – представляет собой методы решения задач линейного программирования с несколькими целевыми функциями, которые могут конфликтовать друг с другом;

3) *нелинейное программирование*, если функция $f(X, Y, Q, G)$ нелинейна относительно управляющих переменных X ;

4) *дискретное программирование*, если на управляющие переменные X наложено условие дискретности, например, целочисленности: x_j – целое, $j = 1, \dots, n$;

5) *динамическое программирование*, если функция $f(X)$ является аддитивной $f(X) = f(x_1, x_2, \dots, x_n) = \sum_{j=1}^n f_j(x_j)$ или мультипликативной

$f(x_1, x_2, \dots, x_n) = \prod_{j=1}^n f_j(x_j)$, и при этом возможно разбиение решения задачи

на несколько этапов, каждый из которых представляет собой решение подзадачи относительно одной переменной отдельно от других подзадач;

6) *стохастическое программирование*, если вектора внешних переменных X , возмущений G и/или параметров системы Q имеют случайную (стохастическую) природу;

7) *нечеткое математическое программирование*, если решение принимается в условиях неопределенности, когда вектора внешних переменных X , возмущений G и/или параметров системы Q точно неизвестны, и лицо, принимающее решение, может лишь указать интервал их значений и свои субъективные оценки относительно шансов появления тех или других значений;

8) *эвристическое программирование*, если точный оптимум найти невозможно из-за комбинаторного характера задачи и связанного с ним огромного количества вариантов. В таком случае отказываются от поиска оптимальных решений и ограничиваются поиском удовлетворительного решения с использованием специальных приемов (*эвристик*), позволяющих существенно сократить число рассматриваемых вариантов.

В ряде случаев при решении задач исследования операций альтернативной методом математического программирования могут выступать:

1) *теория графов*, например, при решении задач нахождения кратчайшего пути, критического пути, максимального потока, минимизации стоимости потока в сети с ограниченной пропускной способностью и др. Эти задачи могут быть решены и методами линейного программирования, но алгоритмы теории графов позволяют решать их более эффективно;

2) *теория вероятностей*, например, для прогнозирования (регрессионный и корреляционный анализ), вероятностного управления запасами, моделирования систем массового обслуживания, имитационного моделирования и др.;

3) *теория игр* – рассматривает процессы выбора наилучшей из нескольких альтернатив в ситуациях определенности (данные известны точно), в условиях риска (данные можно описать с помощью вероятностных распределений), и в условиях неопределенности (вероятностное распределение либо неизвестно, либо не может быть определено);

4) *теория нечетких множеств* позволяет в математической форме представить и использовать для принятия решений субъективную экспертную информацию: предпочтения, правила, оценки значений количественных и качественных показателей.

Реализация найденного решения на практике – один из важнейших этапов, завершающий операционное исследование. Внедрение в практику найденного на модели решения можно рассмотреть как самостоятельную

задачу, требующую применения системного подхода и анализа. Полученную на модели оптимальную стратегию необходимо привести в соответствующую содержательную форму в виде инструкций, правил, алгоритма, то есть формализовать для обеспечения однозначности и простоты выполнения персоналом [12, 13].

2.2.2. Системотехнические задачи исследования операций

Исследование операций представляет собой искусство давать плохие ответы на те практические вопросы, на которые даются еще худшие ответы другими способами.

Т.Л. Саати,
редактор журнала «Исследование операций»

Как уже отмечалось, предпосылками к решению задач исследования операция являются предположения о том, что для системы известны:

- цель, которая должна быть достигнута;
- ограничения, которые должны быть удовлетворены;
- параметры, управляя которыми, можно достичь цели.

Системный подход в исследовании операций заключается в том, что любая задача решается с точки зрения влияния на критерии функционирования системы в целом. При проектировании сложных систем и управлении ими приходится решать несколько классов задач, традиционно относимых к исследованию операций:

- задачи управления ресурсами;
- задачи транспортного типа;
- задачи календарного планирования;
- задачи массового обслуживания,

а также различные комбинации перечисленных задач.

Рассмотрим содержательную постановку частных задач исследования операций.

Задачи управления ресурсами. К основным задачам этого класса относятся задачи: а) организации поставок и управления запасами ресурсов; б) распределения ресурсов.

Задача организации поставок и управления запасами призвана разрешить следующее противоречие: с увеличением объема запасов увеличиваются затраты на их хранение, но уменьшаются потери вследствие возможного дефицита.

Функция затрат представляет собой критерий эффективности избранной стратегии управления запасами и включает такие составляющие, как расходы на хранение запасов, стоимость поставки, потери вследствие дефицита.

Наиболее употребляемыми в задачах управления запасами являются следующие ограничения: на максимальный объем (уровень) запасов на складах, на максимальную стоимость запасов, на число поставок, на стоимость поставки, на вероятность дефицита и т.д.

Задачи распределения ресурсов возникают, если есть некий набор работ, которые нужно выполнить, а наличных ресурсов для выполнения каждой работы наилучшим образом не достаточно. В классической постановке задачи распределения ресурсов предполагается, что заданы и работы, и ресурсы, а требуется оптимизировать некоторый критерий эффективности распределения (максимизировать прибыль, минимизировать издержки и т.п.) Например, пусть для осуществления n видов работ необходимо m типов ресурсов, причем известно:

a_{ij} – количество i -того ресурса, необходимого для выполнения единичного объема j -того вида работ: $a_{ij} \geq 0$; $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$;

b_i – доступный объем (запас) i -того ресурса: $b_i > 0$;

c_j – цена выполнения единичного объема j -того вида работ: $c_j > 0$.

Пусть x_j – планируемый объем выполнения j -того вида работ. Тогда допустимым является такое планирование объемов выполняемых работ $x = (x_1, x_2, \dots, x_n)$, при котором суммарные затраты каждого i -того ресурса не превосходят его доступного объема (запаса):

$$\sum_{j=1}^n a_{ij} x_j \leq b_i; i = 1, \dots, m. \quad (6)$$

Кроме того, имеем следующее естественное ограничение:

$$x_j \geq 0; j = 1, \dots, n. \quad (7)$$

Стоимость общего объема запланированных работ x выражается величиной

$$\sum_{j=1}^n c_j x_j. \quad (8)$$

Задача планирования (распределения) ресурсов ставится следующим образом: среди всех векторов x , удовлетворяющих ограничениям (6) и (7), найти такой, при котором величина (8) принимает наибольшее значение:

$$\sum_{j=1}^n c_j x_j \rightarrow \max.$$

Следует отметить, что в практических задачах системотехники в качестве ресурсов, подлежащих распределению, могут рассматриваться не только материальные, но информационные потоки, а также характеристики функционирования – например, производительность или коэффициенты использования оборудования.

Задача распределения ресурсов может иметь, кроме рассмотренной выше, две альтернативные постановки. Первый вариант – заданы наличные ресурсы и требуется определить, какой состав работ можно выполнить с этими ресурсами, чтобы обеспечить максимум некоторой меры эффективности. Частным случаем такой задачи является, например, задача загрузки уникального оборудования: известна производительность оборудования и предельно допустимая загрузка, а также известен перечень работ с заданной трудоемкостью и значимостью выполнения. Требуется определить набор выполняемых работ, обеспечивающий максимальную суммарную значимость выполнения.

Второй вариант – заданы работы, подлежащие выполнению, а требуется подобрать такие ресурсы, которые дают возможность выполнить их с минимальными производственными затратами. Примером практической задачи такого типа является распределение объемов работ в неоднородной кластерной системе.

Задачи транспортного типа решаются при исследовании процессов на транспорте и в системах связи. Типичной задачей этого типа является задача нахождения некоторого маршрута транспортировки при наличии нескольких альтернативных маршрутов через разные промежуточные пункты. На допустимые маршруты может быть наложен ряд ограничений. Так, например, вводят запрет на возврат к уже пройденному пункту или требование обхода всех пунктов транспортной сети с условием, что в каждом пункте можно побывать лишь один раз (задача коммивояжера). Часто вводят ограничения на пропускные способности коммуникаций.

Пусть для осуществления работ в n пунктах необходим ресурс, поставляемый из m источников. Необходимо организовать доставку ресурса из источников в пункты потребления так, чтобы при полном удовлетворении потребностей минимизировать суммарные потери транспортировки, причем известно:

a_i – доступный объем (запас) ресурса в i -том источнике: $a_i > 0$;

b_j – потребность в ресурсе на j -том пункте: $b_j > 0$;

c_{ij} – цена доставки единичного объема ресурса из i -того источника в j -тый пункт выполнения работ: $c_{ij} > 0$; $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$.

При ограничивающем условии – предположении о том, что суммарные запасы равны суммарным потребностям ресурса – транспортная задача сводится к задаче линейного программирования. При условии:

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j, \quad (9)$$

постановка задачи имеет следующий вид:

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \text{ при условиях:} \\ \sum_{i=1}^m \sum_{j=1}^n x_{ij} = \sum_{j=1}^n b_j = \sum_{i=1}^m a_i, \quad x_{ij} > 0, \end{aligned} \quad (10)$$

где x_{ij} – количество ресурса, доставляемого из i -того источника в j -тый пункт.

Условие (9) является необходимым и достаточным для существования по крайней мере одной матрицы перевозок $[x_{ij}]$, удовлетворяющей ограничениям задачи (10).

Если условие (9) не выполняется, т.е. ресурсов заведомо не хватает для обеспечения полной потребности, то транспортная задача решается в системе с задачей оптимизации распределения ресурсов. К задачам прикладной системотехники такого типа относится задача распределения ограниченных мощностей каналов передачи данных между различными узлами компьютерной сети.

Содержательная постановка задачи имеет целью поиск максимально эффективного распределения пропускной способности каналов, если известны:

- 1) потребности абонентов сети в получении объемов информации (ресурса);
- 2) возможности сети в предоставлении каналов той или иной мощности между различными узлами связи;
- 3) критерии эффективности распределения каналов относительно их пропускной способности.

Рассмотрим постановку задачи транспортного типа на примере распределения ограниченных мощностей каналов передачи данных между узлами в сети одно провайдера. Пусть известны потребности абонентов сети в получении того или иного количества информации и возможности провайдеров в предоставлении каналов той или иной мощности между различными узлами связи. С учетом этих возможностей заданы предпочтения абонентов и провайдеров относительно возможности передачи того или иного

количества информации тому или иному абоненту или узлу и определены условия признания эффективности распределения каналов относительно их пропускной способности. Пусть также заданы следующие ограничения:

- 1) информация распределяется от центра к абонентам через коммутационные узлы по каналам связи;
- 2) каждый узел или абонент сети обслуживается одним или несколькими коммутационными узлами;
- 3) количество распределяемой информации для коммутационных узлов и абонентов может быть ограничено как сверху (принципиальные ограничения возможностей провайдера), так и снизу (минимальная потребность абонентов в получаемой информации).

Требуется распределить пропускную способность каналов максимально эффективно, учитывая как потребности и предпочтения абонентов, так и возможности провайдеров.

Пусть P – множество провайдеров сети, R – множество коммуникационных узлов, U – множество абонентов. Обозначим через A_i^- и A_i^+ верхнее и нижнее ограничение суммарной мощности канала передачи данных, которую способен предоставить провайдер i ; B_j^- , B_j^+ – верхнее и нижнее ограничение суммарной мощности канала передачи данных, которую способен обработать коммуникационный узел j ; C_k^- , C_k^+ – верхнее и нижнее ограничение суммарной мощности канала передачи данных, которую необходимо предоставить абоненту k ; D_{ijk}^- , D_{ijk}^+ – верхнее и нижнее ограничение мощности канала передачи данных, ведущего от провайдера i к абоненту k через коммуникационный узел j ; h_i – затраты, связанные с предоставлением провайдера i связи единичной мощности; g_j – затраты, связанные с обработкой передающей станции j связи единичной мощности; q_k – доход, связанный с получением абонента k связи единичной мощности; $i \in P$, $j \in R$, $k \in U$. Тогда, предполагая что распределение мощностей каналов связи удовлетворяет условиям аддитивности и пропорциональности, можно рассматривать задачу максимизации суммарной прибыли, которая заключается в определении таких величин x_{ijk} – мощность канала связи, предоставляемая абоненту k через коммуникационный узел j провайдером i , $i \in P$, $j \in R$, $k \in U$, для которых выполняются ограничения:

$$A_i^- \leq \sum_{j \in R} \sum_{k \in U} x_{ijk} \leq A_i^+, \quad i \in P;$$

$$B_j^- \leq \sum_{i \in P} \sum_{k \in U} x_{ijk} \leq B_j^+, \quad j \in R;$$

$$C_k^- \leq \sum_{i \in P} \sum_{j \in R} x_{ijk} \leq C_k^+, \quad k \in U;$$

$$D_{ijk}^- \leq x_{ijk} \leq D_{ijk}^+, \quad i \in P, j \in R, k \in U,$$

и принимает максимальное значение критерий $\sum_{i \in P} \sum_{j \in R} \sum_{k \in U} (q_k - h_i - g_j) x_{ijk}$, характеризующий суммарную прибыль, которую получит система.

Задачи календарного планирования. К этому классу относятся задачи: а) составления расписаний; б) сетевого планирования.

Задача составления расписаний возникает при планировании работ по проектированию сложных систем, обслуживанию потока заявок в многопроцессорной системе, загрузке технологического оборудования и т.п. В классической постановке требуется найти такое назначение очередности работ, при которых полная совокупность работ будет выполнена с оптимальной эффективностью (за минимальное время, с минимальными простоями, с минимальными издержками, с максимальной загрузкой оборудования, с минимальным запаздыванием относительно директивных сроков, с максимальным опережением директивных сроков и проч.).

В общей постановке задачи составления расписания обычно дано:

- множество *работ* (технологических операций, задач и т.п.) $J = \{j_k\}$;
- множество *машин* (станков, процессоров и т.п.) $M = \{m_i\}$.

Цель задачи – составить расписание работ, т. е. определить каждой машине перечень и время выполнения работ, с учетом того, что в любой момент времени каждая машина выполняет не более одной работы, а каждая работы выполняется на одной машине или не выполняется.

Постановка задачи может быть дополнена ограничениями двух типов. Ограничения первого типа описывают взаимную зависимость работ, т.е. определяют порядок их выполнения. Это ограничения логического характера: например, выполнению работы j_k должна предшествовать некоторая совокупность работ, без завершения которой она начаться не может. Ограничения этого типа могут иметь и сложную форму (работы могут быть взаимозаменяемы, вестись параллельно и т.п.).

Ограничения второго типа связаны с объемом ресурса, который может быть выделен на реализацию проекта. Пусть известны:

$a_j(t)$ – доступный объем (запас) j -того ресурса, выделенный на реализацию работ по проекту в срок времени t ;

$u_k(t)$ – доля работы j_k , которую планируется выполнить в срок t ($0 \leq u_k(t) \leq 1$);

$b_j[u_k(t)]$ – требуемый объем j -того ресурса для выполнения $u_k(t)$.

Тогда ограничения второго типа записываются в виде:

$$\sum b_j[u_k(t)] \leq a_j(t) \quad \forall t.$$

Оптимизация расписания, например, по условию минимума суммарного времени осуществления проекта в целом или его отдельного этапа в срок времени t , является задачей дискретного программирования (задачей комбинаторики). Общее количество возможных вариантов расписаний в задаче для i машин и k выполняемых видов работ составляет $(k!)^i$. Ее решение методом полного перебора требует существенных затрат времени даже для небольших систем. Например, если требуется спланировать изготовление четырех изделий, каждое из которых проходит обработку на пяти станках, то существует $(4!)^5 \approx 8$ млн различных вариантов последовательностей выполнения работ. Поэтому для решения таких задач применяются приближенные эвристические методы, например, выполняют ранжирование работ с использованием методов принятия решений.

ПРИМЕРЫ задач составления расписания



В задачах сетевого планирования рассматривают соотношение между сроком окончания определенного комплекса операций и моментами начала выполнения всех операций комплекса. Эти задачи актуальны при разработке сложных проектов.

Для строгой постановки задачи сетевого планирования необходимы следующие условия:

- 1) наличие точно определяемого множества операций, которые надо выполнить для завершения всего комплекса работ, включающего эти операции как свои составляющие;
- 2) множество операций комплекса упорядочено так, что для каждой из них известно, какие операции непосредственно ей предшествуют, а которые непосредственно следуют за ней;
- 3) в пределах заданного отношения упорядочения операции можно начинать и заканчивать независимо одну от другой;
- 4) известна взаимосвязь между величиной потребляемого ресурса и длительностью каждой операции.

Использование сетевых моделей позволяет:

- 1) построить сетевой график, который представляет взаимосвязи операций, что позволяет детально анализировать все работы и вносить улучшения в структуру проекта еще до начала его реализации;

2) построить календарный график, который определяет моменты начала и окончания каждой операции, минимально возможное время выполнения проекта, критические операции; позволяет оптимизировать параметры проекта: выявить и устранить проблемы в обеспечении работ исполнителями, снизить количество одновременно занятых исполнителей, сократить длительность отдельных работ и проекта в целом;

3) контролировать и корректировать ход выполнения проекта.

Задачи сетевого планирования в зависимости от начальных условий и постановки делятся на два типа:

1) задача по критерию «время». Задан сетевой график выполнения проекта, доступные для выполнения полного комплекса запасы ресурсов и их изменение во времени $t \in [0, T]$. Нужно распределить эти ресурсы по операциям (i, j) и определить такие моменты начала и окончания всех операций комплекса, при которых минимизируется общая продолжительность всего комплекса операций T ;

2) задача по критерию «стоимость». Задана общая продолжительность всего комплекса работ T . Определить сроки начала каждой операции и распределение ресурсов по операциям, при которых минимизируется один из следующих критериев:

- общие затраты на выполнение всего комплекса работ;
- показатель неравномерности расходования ресурсов во время осуществления комплекса;
- вероятность невыполнения комплекса в срок.

Как и задачи составления расписаний, задачи сетевого планирования носят комбинаторный характер, и для их решения используются преимущественно эвристические алгоритмы, а также теория графов.

Задачи массового обслуживания исследуют процессы в системах, на вход которых поступает случайный поток заявок (запросов, требований и т.п.), приходящихся в общем случае на случайные моменты времени. Поступившая заявка обслуживается в системе путем предоставления ей некоторых ресурсов на какое-то время и, будучи в той или иной мере обслуженной, покидает систему.

Помимо случайного появления заявок на обслуживание и случайной длительности обслуживания каждой заявки для систем массового обслуживания характерным является наличие очередей, в которых заявки ждут момента освобождения ресурсов, занятых обслуживанием других заявок.

Поскольку события, происходящие в вычислителях сетях и системах, носят случайный характер, то для их изучения наиболее подходящими яв-

ляются вероятностные математические модели теории массового обслуживания. Так, используемые в локальных сетях протоколы канального уровня используют методы доступа к среде, основанные на ее совместном использовании несколькими узлами за счет разделения во времени. В этом случае, как и во всех случаях разделения ресурсов со случайным потоком заявок, могут возникать очереди.

Очереди возникают вследствие того, что поток заявок не управляем и случаен, а количество устройств обслуживания n ограничено. Если количество устройств обслуживания взять довольно большим, то очереди будут образоваться редко и среднее время ожидания в очереди $t_{ож}$ будет небольшим, но неизбежны продолжительные простои устройств обслуживания. Если, наоборот, количество устройств обслуживания маленькое, то возникают большие очереди и имеют место большие потери из-за ожидания в очереди: $F_1(t_{ож}) = t_{ож} \cdot c_1$, где c_1 – затраты на единицу времени ожидания. Поэтому одна из возможных задач массового обслуживания следующая: определить такое число устройств обслуживания n_0 , при котором минимизируется сумма ожидаемых потерь от ожидания в очереди $F_1(n)$ и простоев оборудования $F_2(n)$ (рис. 22):

$$F_{\Sigma}(n) = F_1(n) + F_2(n) \rightarrow \min_n.$$

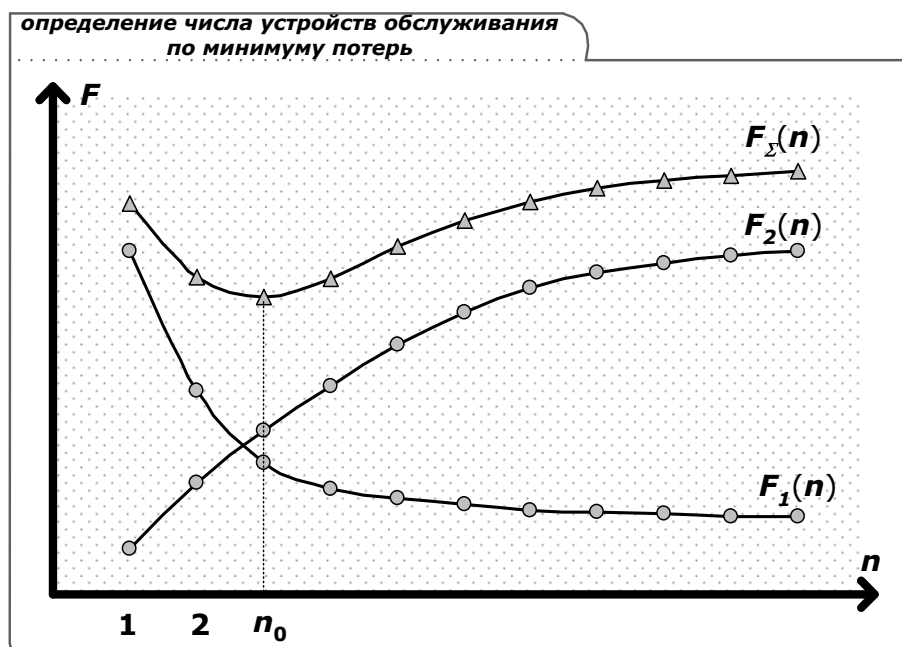


Рис. 22

Кроме рассмотренного критерия оптимизации используются и другие, например: вероятность задержки обслуживания заявки; вероятность

потери заявки без обслуживания; математическое ожидание числа обслуженных заявок за фиксированное время или заявок, обслуживание которых было задержано; математическое ожидание числа занятых каналов обслуживания; математическое ожидание длины очереди.

Параметры оптимизации систем массового обслуживания можно разделить на два класса: характеристики структуры системы (выбор числа каналов обслуживания, их последовательности, пропускной способности) и характеристики функционирования (вид и параметры входящего потока заявок, выбор наилучшей дисциплины обслуживания и т.п.).

Методы решения задач массового обслуживания делятся на аналитические и имитационные. Аналитическим путем решаются простые частные задачи с существенными ограничениями. Большинство практических задач решаются имитационными методами статистического моделирования, в основном методом Монте-Карло. Методы решения задач массового обслуживания для специальности 1-40 02 01 «Вычислительные машины, системы и сети» являются основным предметом теоретического курса, лабораторного практикума и курсового проектирования по дисциплине «Моделирование».

Часто практические задачи исследования операций представляют собой совокупность нескольких рассмотренных выше типовых задач одновременно (например, совокупное решение задач планирования и управления ресурсами). Такие задачи называются *комбинированными*. Их нельзя решать независимо друг от друга. Последовательная оптимизация частных подзадач не всегда приводит к оптимальному решению в целом. Поэтому для решения комбинированных задач используют метод последовательных приближений, который дает возможность приблизиться к искомому решению комбинированной задачи как можно ближе [12 – 17, 23, 24].

Предложенная классификация задач исследования операций обычно не является исчерпывающей и окончательной. Рассмотрены только задачи, которые представляют интерес для инженера-системотехника по специальности 1-40 02 01.

2.3. Элементы теории графов

– Знаете, граф, ваши аргументы ужасны...

А. Дюма
«Граф Монте-Кристо».

Наряду с теоретико-множественным представление концептуальных моделей систем в виде графов является одним из наиболее распространенных. В системном анализе теория графов применяется [3, 15, 17]:

- 1) для создания топологических моделей систем;
- 2) для решения ряда задач исследования операций (задачи нахождения кратчайшего пути, максимального пути, максимального потока, минимизации стоимости потока в сети с ограниченной пропускной способностью и др.).

2.3.1. Основные понятия и способы задания графов

Графом называется G геометрическая фигура, построенная на множестве вершин $V=\{v_1, v_2, \dots, v_m\}$ и ребер $R=\{r_1, r_2, \dots, r_n\}$: $G=(V, R)$. Если ребра ориентированы (рис. 23), то их называют *дугами*, а граф – ориентированным (*орграфом*). Вершины ориентированного графа называют *узлами*.

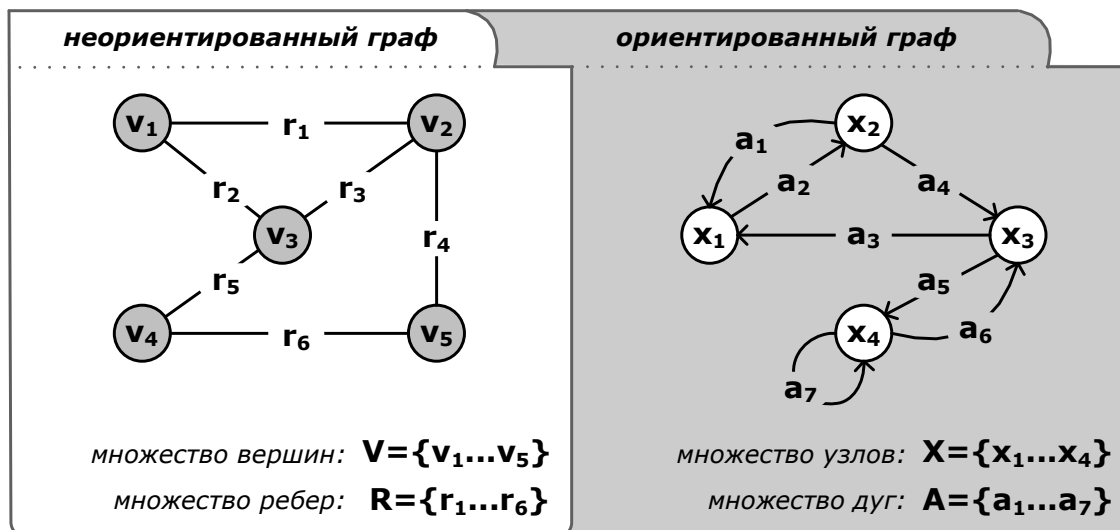


Рис. 23

Графы благодаря наглядности и высокой общности служат в системотехнике основным средством для описания структур и функционирования систем. При описании структур вершинами являются элементы системы, а ребрами – связи между ними. Примерами таких структурных описаний являются вычислительные сети, логические схемы, иерархические структуры. При описании функционирования чаще используются ориентированные графы, узлам которых соответствуют состояния системы, а дугам – переходы между состояниями. Иногда описание системы в виде графа содержит оба аспекта. Например, узлы графа (блок-схемы) алгоритма представляют его компоненты, а прохождение по дугам изображает передачу активности от одной компоненты к другой, т.е. процесс выполнения алгоритма.

Дуга a_i , связанная с узлом x_j , называется *инцидентной* этому узлу, причем, если она входит в этот узел – положительно инцидентная, а если выходит – отрицательно инцидентная. Два узла x_k и x_i являются *смежными*, если им инцидентна одна дуга. Аналогично, две дуги смежны, если они инцидентны одному узлу, причем, если одна входит, а другая выходит из узла – последовательно смежны, в противном случае – параллельно смежны (рис. 24).

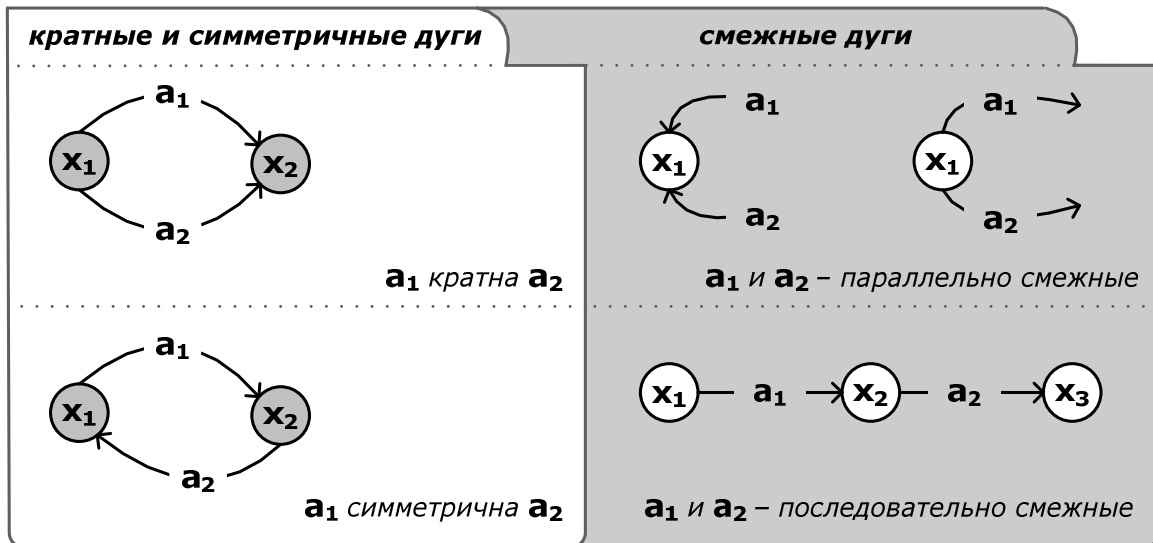


Рис. 24

Дуга, выходящая из узла и в него же входящая, называется *петлей* (например, дуга a_7 на рис. 23). Узел, из которого дуги только выходят, называется *истоком*, а в который только входят – *стоком*.

Каждая вершина неориентированного графа v_i может характеризоваться *степенью вершины* $d(v_i)$, которая равна количеству ребер, инцидентных этой вершине. Например, для неориентированного графа на рис. 23 степени вершин имеют следующие значения: $d(v_1) = 2$, $d(v_2) = 3$, $d(v_3) = 2$, $d(v_4) = 2$, $d(v_5) = 2$.

Каждый узел орграфа x_i может характеризоваться *полустепенью исхода* $d_{out}(x_i)$ и *полустепенью захода* $d_{in}(x_i)$. Полустепенью исхода $d_o(x_i)$ называется количество дуг, исходящих из узла x_i . Полустепенью захода $d_i(x_i)$ называется количество дуг, входящих в этот узел. Например, для ориентированного графа на рис. 23 значения полустепеней исхода для узлов следующие: $d_{out}(x_1) = 1$, $d_{out}(x_2) = 2$, $d_{out}(x_3) = 2$, $d_{out}(x_4) = 2$, а значения полустепеней исхода $d_{in}(x_1) = 2$, $d_{in}(x_2) = 1$, $d_{in}(x_3) = 2$, $d_{in}(x_4) = 2$.



Маршрут в графе это последовательность смежных узлов или смежных дуг. В маршруте могут повторяться узлы, но не дуги. Маршрут называется *циклом*, если в нем первый узел совпадает с последним. *Путь* в графе это маршрут без повторения узлов и дуг. Для ориентированных графов дуги в маршруте или пути должны быть ориентированны в одном направлении. *Контур* это цикл без повторения узлов, за исключением первого узла, совпадающего с последним (рис. 25).

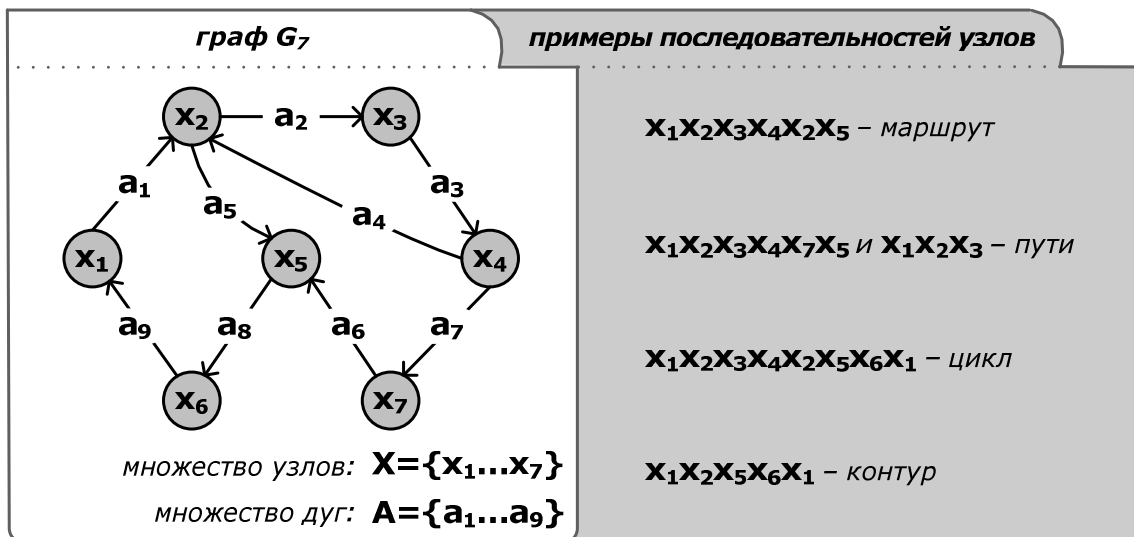


Рис. 25

Граф называется *взвешенным*, если каждой дуге (или каждому ребру) поставлено в соответствие некое значение – вес. Весом пути для такого графа называется сумма весов дуг пути.

Граф может быть задан различными способами. При использовании *теоретико-множественного представления* граф описывается перечислением множества узлов и дуг. Пример такого описания для орграфов приведен на рис. 26.

Задание графов соответствием. Такое описание графов состоит в задании множества узлов X и *соответствия* C , которое показывает, как вершины связаны между собой. Граф в этом случае обозначается парой $G = (X, C)$. Соответствием узла $C(x_i)$ является множество узлов, в которые существуют дуги из узла x_i , т.е. $C(x_i) = \{x_j: \exists \text{ дуга } (x_i, x_j) \in A\}$. Так, для орг-

графа на рис. 26, описание заданием множества узлов и соответствия выглядит следующим образом: $G_4 = (X, C)$, где $X = \{x_1...x_4\}$, $C(x_1) = \{x_2\}$, $C(x_2) = \{x_3, x_4\}$, $C(x_3) = \{x_3\}$, $C(x_4) = \{x_1, x_2\}$ – соответствия.

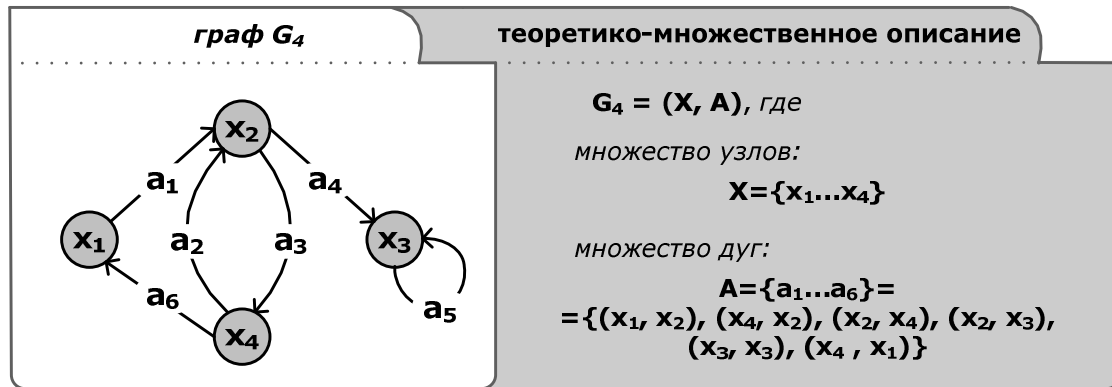


Рис. 26

Для неориентированного графа соответствие C задает эквивалентный ориентированный граф, который получается из исходного графа заменой каждого неориентированного ребра двумя противоположно направленными дугами, соединяющими те же самые вершины. Например, для неориентированного графа на рис. 23 $C(v_2) = \{v_1, v_3, v_5\}$, $C(v_4) = \{v_3, v_5\}$ и т.д.

Для решения задач компьютерного моделирования и анализа удобно использовать матричное или списочное представление графов. Структуры данных различаются в т.ч. и с точки зрения эффективности алгоритмов их анализа. Существует два основных вида матриц графов: матрица инцидентности (или инциденций) и матрица смежности.

Матрица инцидентности ориентированного графа это в общем случае прямоугольная матрица $M_I = [m_{ij}]$, $i = \overline{1, m}$, $j = \overline{1, n}$, где m – число узлов, n – число дуг, т.е. матрица размером $m \times n$, строки которой соответствуют узлам, а столбцы – дугам графа. Элементы матрицы M_I определяются следующим образом:

- $m_{ij} = 1$, если x_i является начальным узлом дуги a_j ;
- $m_{ij} = -1$, если x_i является конечным узлом дуги a_j ;
- $m_{ij} = 0$, если дуга a_j не связана с узлом x_i (не инцидентна узлу x_i).

Для неориентированного графа элементы матрицы инцидентности определяются следующим образом:

- $m_{ij} = 1$, если с вершиной x_i связано ребро a_j (ребро инцидентно вершине);
- $m_{ij} = 0$, если связи нет.

Матрицы инцидентности в описанном виде применимы только к графам без петель. В случае наличия в графе петель эту матрицу можно расчленить на две полуматрицы: положительную и отрицательную, или представлять петлю в матрице другим значением, например, 2, в строке, соответствующей вершине.

Пример матрицы инцидентности для ориентированного графа представлен на рис. 27.

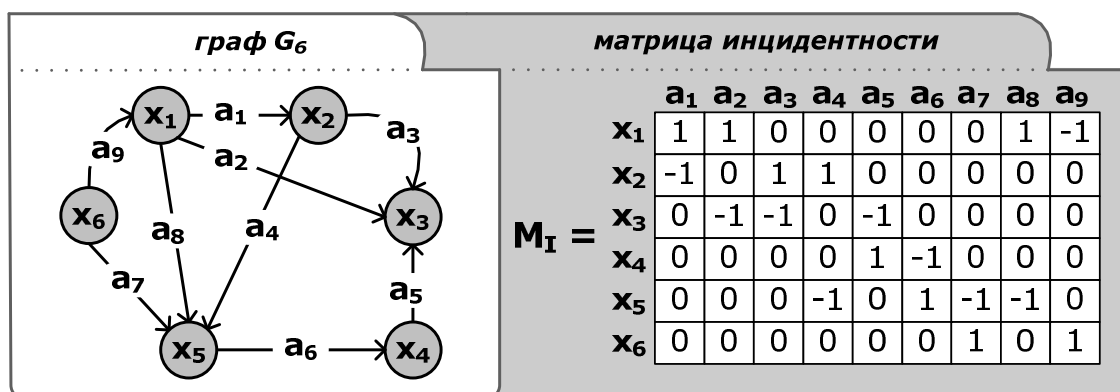


Рис. 27

С алгоритмической точки зрения матрица инцидентности является одним из самых худших способов представления графа, т.к. :

1) для задания графа требуется $n \times t$ ячеек памяти (где t – число узлов, n – число дуг графа), причем матрица сильно разрежена – большинство ячеек занято нулями;

2) анализ матрицы, например, определение, существует ли дуга (x_i, x_j) или к каким вершинам ведут дуги из x_k , требует в худшем случае перебора всех столбцов, т.е. t шагов.

Произведение матрицы инцидентности на транспонированную матрицу инцидентности позволяет получить *матрицу структуры связей*:

$$M_{LS} = M_I^T \cdot M_I.$$

Матрица структуры связей имеет размерность $n \times n$, где n – число дуг графа, а значения ее элементов означают следующее:

- $m_{ij} = 2$, если $i=j$ или дуга a_i кратна дуге a_j ;
- $m_{ij} = -2$, если дуга a_i симметрична дуге a_j ;
- $m_{ij} = 1$, если дуга a_i параллельно смежна дуге a_j ;
- $m_{ij} = -1$, если дуга a_i последовательно смежна дуге a_j ;
- $m_{ij} = 0$, если дуга a_i не смежна дуге a_j .

Таким образом, матрица структуры связей содержит информацию о взаимной ориентации пар дуг графа.

Матрицей смежности называется квадратная матрица $M_C = [m_{ij}]$, $i, j = \overline{1, m}$, где m – число узлов, т.е. $M_{m \times m}$, для которой:

- $m_{ij} = 1$, если существует $(x_i, x_j) \in A$;
- $m_{ij} = 0$, если не существует $(x_i, x_j) \in A$.

Число единиц в матрице M_C равно числу дуг n . Основное достоинство матрицы смежности состоит в том, что анализ существования дуги (x_i, x_j) осуществляется за один шаг. Недостатком является тот факт, что независимо от числа дуг объем занимаемой памяти составляет m^2 . На практике это неудобство иногда можно исправить, храня целую строку или столбец в одном машинном слове (для малых m). На рис. 28 приведен пример графа и его матрицы смежности, по которой можно определить характеристики вершин. Так сумма элементов i -ой строки матрицы дает полустепень исхода вершины x_i , а сумма элементов i -го столбца дает полустепень захода вершины x_i . По матрице смежности можно найти прямые и обратные соответствия. Если элемент $m_{ij}=1$, то элемент графа x_j входит в соответствие $C(x_i)$. Например, в 2-ой строке матрицы на рис. 28 единицы стоят в 2-м и 5-м столбцах, следовательно, $C(x_2) = \{x_2, x_5\}$.

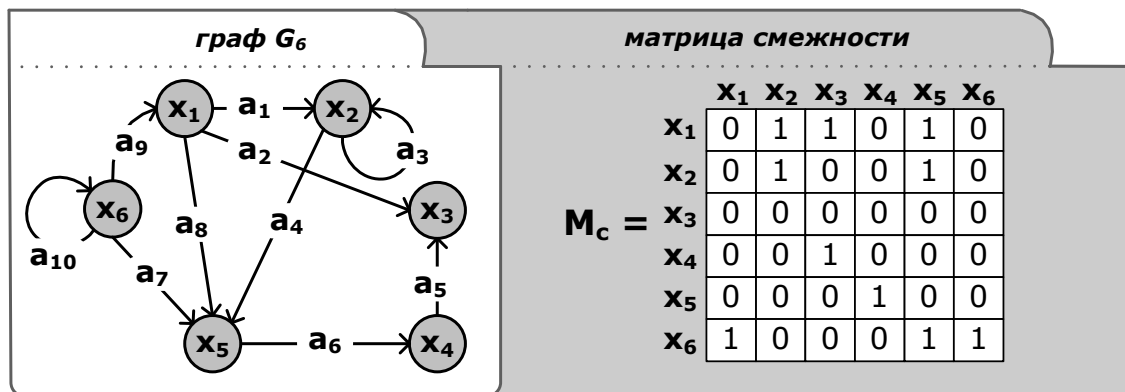


Рис. 28

Матрица смежности обладает еще одним полезным свойством: если возвести матрицу M_C в k -тую степень, то каждый элемент матрицы M_C^k будет равен числу путей из узла x_i в узел x_j длиной в k дуг.

Результат возведения в квадрат матрицы смежности графа, приведенного на рис. 28, имеет вид:

$$M_C^2 = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}^2 = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 1 \end{bmatrix}.$$

Как видно из этого результата, через две ($k = 2$) смежные дуги можно попасть, например, из узла x_1 в узел x_2 (см. рис. 28 – путь a_1a_3), x_4 (путь a_8a_6) и x_5 (путь a_1a_4). Значение, равное 2 на пересечении строки x_6 и столбца x_5 , показывает, что из вершины x_6 в вершину x_5 существует два альтернативных пути длиной в две смежных дуги ($a_{10}a_7$ и a_9a_8 соответственно).

Таким образом, можно алгоритмически определить, существует ли путь из x_i в x_j длиной в k дуг. Очевидно, что если число узлов m , максимальная степень k , в которую нужно возвести M_C для определения самого длинного пути, равна $(m - 1)$.

Для списочного задания графов используют списки пар и списки связи. Для задания *списка пар*, соответствующих дугам графа, требуется двумерных массив размерностью $2m$, где m – число узлов графа. Список соответствует множеству дуг. По сравнению с матрицей смежности этот способ представления с точки зрения занимаемой памяти более экономичен, особенно для так называемых неплотных графов, когда m гораздо меньше n^2 , где n – число дуг графа. В ориентированном графе пара (x_i, x_j) соответствует дуге a_k , если эта дуга выходит из вершины x_i и входит в вершину x_j . Недостатком этого способа задания является большое число шагов (порядка m в худшем случае) для нахождения множества узлов, к которым ведут дуги из данной вершины.

Задание графа *списками связи* осуществляется с помощью одномерного массива размерности n для хранения указателей. Элемент массива – указатель на начало списка, в котором содержится информация о вершинах графа, смежных с рассматриваемой. Задание графа списками связи эквивалентно заданию графа соответствием [3, 19-21].

2.3.2. Создание и анализ топологических моделей систем

Модели систем в виде графов называются *топологическими*. Топологические модели *первого типа* являются описаниями структуры связей внешних, внутренних и выходных параметров системы. Они создаются на уровне концептуального моделирования и далее последовательно детализи-

зируются путем определения операторов связей в математическом моделировании.

Топологические модели *второго типа* являются описаниями функциональной структуры систем. Они используются для решения задач декомпозиции (выделения подсистем в системе) и определения структурных свойств системы.

Примером топологической модели первого типа может служить модель схемы электрической принципиальной, используемая для решения задач анализа динамических и статических характеристик в ПО САПР функционального анализа электронной и вычислительной аппаратуры. Уравнения связи между элементами в анализируемой схеме могут быть заданы уравнениями Кирхгофа. Рассмотрим алгоритмы получения топологических моделей принципиальных схем.

Алгоритм топологического моделирования табличного метода. На рис. 29 приведена схема электрическая принципиальная и соответствующий ей граф. Граф состоит из тех же дуг и узлов, что и исходная схема, и отличается от схемы только отсутствием условных обозначений элементов. Направление дуг совпадает с направлениями токов в схеме (на схеме на рис. 29 показаны стрелками).

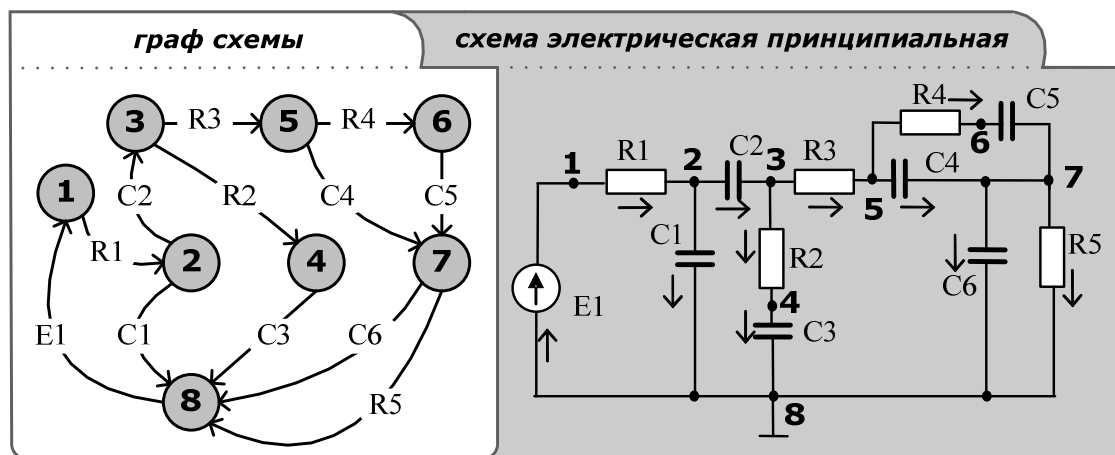


Рис. 29

Для получения топологической модели схемы полное множество дуг графа разделяют на подмножества *ветвей* и *хорд* путем построения фундаментального дерева в графе схемы. *Фундаментальным деревом графа* называется подграф, состоящий из всех вершин исходного графа, максимально возможного числа дуг и не содержащий замкнутых циклов. В дереве любые две вершины связаны единственной цепью. Для любого графа,

если он сам не является деревом, можно построить несколько фундаментальных деревьев. Дуги исходного графа, вошедшие в фундаментальное дерево, называются *ветвями дерева*, остальные – *хордами*. *Контуром i -той хорды* называют совокупность дуг цикла, образуемого при подключении i -той хорды к фундаментальному дереву. Алгоритм построения фундаментального дерева гарантирует уникальность контура каждой хорды, т.е. каждой хорде в фундаментальном дереве соответствует единственный контур.

По результатам построения фундаментального дерева для графа схемы строят особую разновидность структурной матрицы графа: *матрицу контуров и сечений*. Каждой хорде в матрице M контуров и сечений соответствует строка, а каждой ветви – столбец. В строке i -той хорды записываются единицы в тех столбцах, которым соответствуют ветви дерева, входящие в контур i -той хорды. Единица имеет знак «плюс», если направление дуги данной ветви дерева и i -той хорды совпадают, иначе – знак «минус». В остальных клетках i -той строки записываются нули.

Для графа схемы, приведенного на рис. 29, одно из возможных фундаментальных деревьев (выделено жирными линиями) и соответствующая ему матрица контуров и сечений показаны на рис. 30.

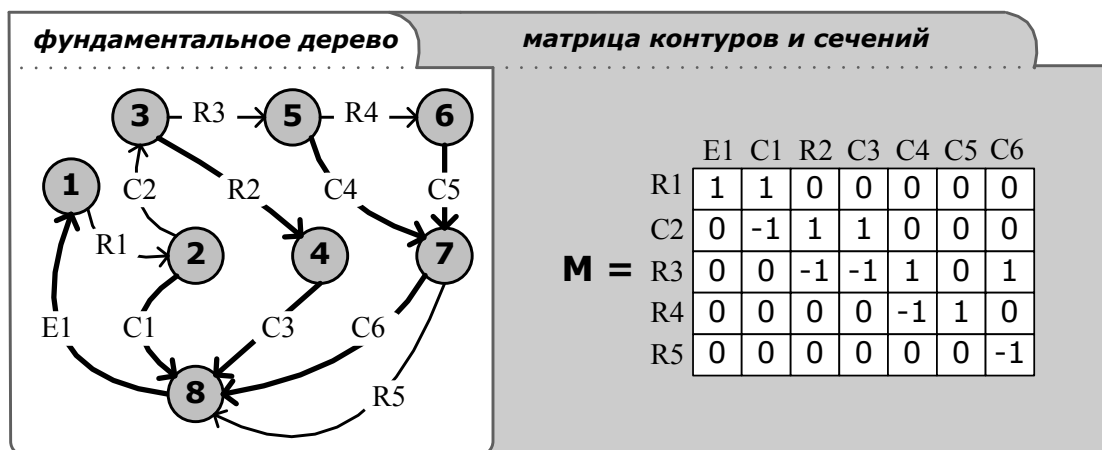


Рис. 30

Как следует из рис. 30, множество ветвей фундаментального дерева для заданного графа схемы, имеет вид: $\{E1, C1, R2, C3, C4, C5, C6\}$, а множество хорд: $\{R1, C2, R3, R4, R5\}$.

Топологические уравнения, соответствующие законам напряжений и токов Кирхгофа, в табличном методе имеют вид:

$$U_x = -M \cdot U_{\partial}, I_{\partial} = M^T \cdot I_x,$$

где U_x и I_x – напряжения и токи ветвей, являющихся хордами; U_{∂} и I_{∂} – то же, для ветвей дерева; M – матрица контуров и сечений; M^T – транспонированная матрица контуров и сечений. В развернутом виде для данного примера результат топологического моделирования имеет вид:

$$\begin{bmatrix} U_{R1} \\ U_{C2} \\ U_{R3} \\ U_{R4} \\ U_{R5} \end{bmatrix} = - \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} U_{E1} \\ U_{C1} \\ U_{R2} \\ U_{C3} \\ U_{C4} \\ U_{C5} \\ U_{C6} \end{bmatrix} \Rightarrow \begin{cases} U_{R1} = -U_{E1} - U_{C1}; \\ U_{C2} = U_{C1} - U_{R2} - U_{C3}; \\ U_{R3} = U_{R2} + U_{C3} - U_{C4} - U_{C6}; \\ U_{R4} = U_{C4} - U_{C5}; \\ U_{R5} = U_{C6}, \end{cases}$$

$$\begin{bmatrix} I_{E1} \\ I_{C1} \\ I_{R2} \\ I_{C3} \\ I_{C4} \\ I_{C5} \\ I_{C6} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} I_{R1} \\ I_{C2} \\ I_{R3} \\ I_{R4} \\ I_{R5} \end{bmatrix} \Rightarrow \begin{cases} I_{E1} = I_{R1}; \\ I_{C1} = I_{R1} - I_{C2}; \\ I_{R2} = I_{C2} - I_{R3}; \\ I_{C3} = I_{C2} - I_{R3}; \\ I_{C4} = I_{R3} - I_{R4}; \\ I_{C5} = I_{R4}; \\ I_{C6} = I_{R3} - I_{R5}. \end{cases}$$

Нетрудно убедиться, что результат топологического моделирования, полученный с использованием табличного метода, идентичен уравнениям, которые можно составить для заданной схемы, пользуясь законами токов и напряжений Кирхгофа.

ПРИМЕР альтернативного фундаментального дерева и матрицы контуров и сечений



Алгоритм топологического моделирования метода узловых потенциалов. В этом методе для построения топологической модели используется матрица инцидентности графа схемы. Топологическими уравнениями являются уравнения закона токов Кирхгофа, записанные для всех узлов схемы, за исключением одного – опорного, и уравнения связи напряжений ветвей с узловыми потенциалами. В матричной форме топологическая модель метода узловых потенциалов имеет вид:

$$-M_I \cdot I = 0, \quad -M_I^T \cdot \Phi + U = 0,$$

где U и I – напряжения и токи дуг графа (ветвей схемы); $-M_I$ – матрица, противоположная матрице инцидентности, иначе называемая матрицей «узел-ветвь»; $-M_I^T$ – транспонированная матрица «узел-ветвь». Матрицей, противоположной матрице M , называется матрица $-M = (-1) \times M$. При умножении матрицы на постоянное число каждый элемент матрицы умножается на это число.

Для графа схемы электрической принципиальной из предыдущего примера (см. рис. 29) при выборе узла 8 в качестве опорного матрица «узел-ветвь» имеет вид, приведенный на рис. 31.

матрица «узел-ветвь» (противоположная матрице инцидентности)

	C_1	C_2	C_3	C_4	C_5	C_6	R_1	R_2	R_3	R_4	R_5	E_1
1	0	0	0	0	0	0	-1	0	0	0	0	1
2	-1	-1	0	0	0	0	1	0	0	0	0	0
3	0	1	0	0	0	0	0	-1	-1	0	0	0
4	0	0	-1	0	0	0	0	1	0	0	0	0
5	0	0	0	-1	0	0	0	0	1	-1	0	0
6	0	0	0	0	-1	0	0	0	0	1	0	0
7	0	0	0	1	1	-1	0	0	0	0	-1	0

Рис. 31

В развернутом виде результат топологического моделирования по методу узловых потенциалов для рассматриваемого примера имеет вид:

$$\begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\
 -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0
 \end{bmatrix} \times \begin{bmatrix} I_{C1} \\ I_{C2} \\ I_{C3} \\ I_{C4} \\ I_{C5} \\ I_{C6} \\ I_{R1} \\ I_{R2} \\ I_{R3} \\ I_{R4} \\ I_{R5} \\ I_{E1} \end{bmatrix} = 0 \Rightarrow$$

$$\Rightarrow \begin{cases} -I_{R1} + I_{E1} = 0; \\ -I_{C1} - I_{C2} + I_{R1} = 0; \\ +I_{C2} - I_{R2} - I_{R3} = 0; \\ -I_{C3} + I_{R2} = 0; \\ -I_{C4} + I_{R3} - I_{R4} = 0; \\ -I_{C5} + I_{R4} = 0; \\ +I_{C4} + I_{C5} - I_{C6} - I_{R5} = 0, \end{cases}$$

$$\begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \\ \varphi_5 \\ \varphi_6 \\ \varphi_7 \end{bmatrix} + \begin{bmatrix} U_{C1} \\ U_{C2} \\ U_{C3} \\ U_{C4} \\ U_{C5} \\ U_{C6} \\ U_{R1} \\ U_{R2} \\ U_{R3} \\ U_{R4} \\ U_{R5} \\ U_{E1} \end{bmatrix} = 0 \Rightarrow \begin{cases} U_{C1} = \varphi_2; \\ U_{C2} = \varphi_2 - \varphi_3; \\ U_{C3} = \varphi_4; \\ U_{C4} = \varphi_5 - \varphi_7; \\ U_{C5} = \varphi_6 - \varphi_7; \\ U_{C6} = \varphi_7; \\ U_{R1} = \varphi_2 - \varphi_1; \\ U_{R2} = \varphi_3 - \varphi_4; \\ U_{R3} = \varphi_3 - \varphi_5; \\ U_{R4} = \varphi_5 - \varphi_6; \\ U_{R5} = \varphi_7; \\ U_{E1} = \varphi_1. \end{cases}$$

Видно, что полученная топологическая система уравнений токов с учетом промежуточных подстановок эквивалентна аналогичной системе уравнений, полученной с применением алгоритма табличного метода.

Примерами топологических моделей второго типа служат:

1) неориентированный граф компьютерной сети, в котором вершины соответствуют активному оборудованию, а ребрам присвоены весовые коэффициенты, характеризующие функциональные характеристики линий связи (длина, скорость передачи, пропускная способность и т.п.);

2) ориентированный граф информационной системы предприятия, в которой источники информации (персонал, измерительные устройства), приемники (персонал, исполнительное оборудование) и устройства обработки информации (активное оборудование компьютерной сети) представлены узлами, а дуги интерпретируют каналы связи с заданным направлением передачи информации.

Задача выделения подсистем в системе относится к задачам декомпозиции и может решаться:

- концептуально, на основании априорных знаний о системе, ее функциональных характеристиках и проч.;
- формальными методами, базирующимися на анализе топологической модели системы, представленной ориентированным графом.

Рассмотрим примеры решения задачи декомпозиции и оптимизации структурных свойств системы.

Пример концептуальной декомпозиции и оптимизации структуры связей компьютерной сети. Пусть топологии компьютерной сети соответствует неориентированный граф, ребрам которого присвоены весовые коэффициенты, характеризующие удельные приведенные затраты при передаче информации по каналам связи (рис. 32).

Задачу декомпозиции такой системы можно решить концептуально: анализ топологии сети, приведенного на рис. 32, показывает, что в структуре системы можно выделить две слабо связанные между собой подсистемы – подсети первого и второго сервера. Выделим в подсеть первого сервера компьютеры 1, 3, 4, 6 и 7, а также выполним оптимизацию структуры связей между ними по критерию минимальной суммы приведенных затрат при передаче информации по каналам связи.

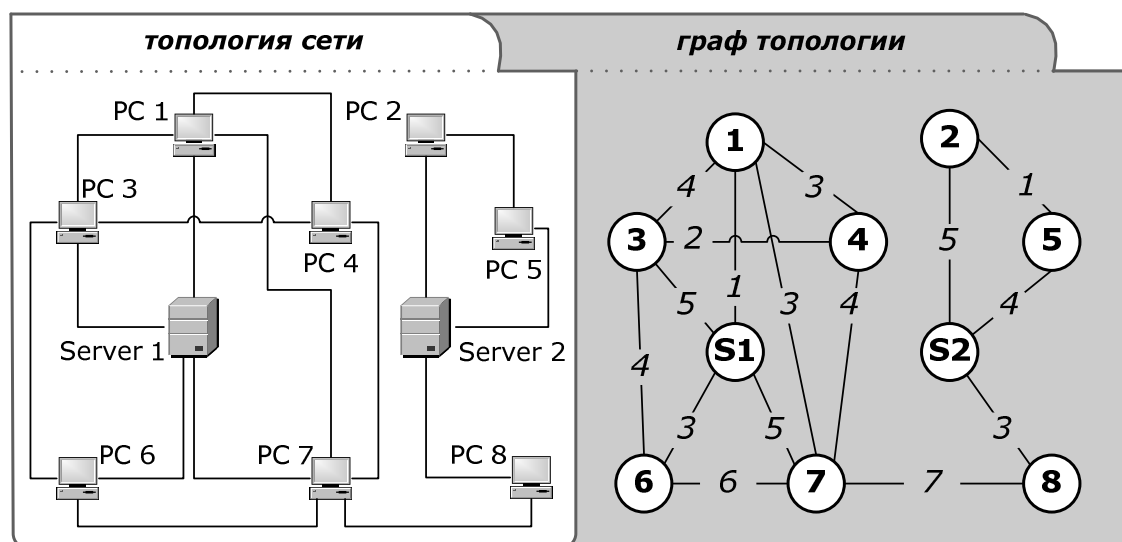


Рис. 32

Решение поставленной задачи требует построения так называемого минимального связывающего дерева в подграфе, соответствующем рассматриваемой подсети. *Минимальным связывающим деревом* графа назы-

вается дерево, связывающее между собой все вершины графа и имеющее минимальный суммарный вес ребер. Для построения таких деревьев используют алгоритмы *Прима* и *Краскала*.

В алгоритме *Прима* на первом шаге выбирается произвольная вершина графа (так называемый корень дерева) и из множества связанных с ней ребер выбирается ребро с минимальным весом. В дерево включается корень дерева, выбранное ребро и смежная вершина. На всех последующих шагах алгоритма рассматриваются множества вершин, имеющие ребра, инцидентные уже построенному дереву, из которых выбирается ребро с минимальным весом для включения в дерево очередной вершины. Построение завершается, когда дерево включает все вершины графа.

На рис. 33 приведен граф рассматриваемой подсети и результат выполнения первого шага алгоритма Прима, для которого в качестве корня дерева была выбрана вершина **6**, и из множества весов инцидентных ребер $\{3, 4, 6\}$ – ребро с минимальным весом, равным 3. Построенный фрагмент дерева выделен цветом, а множество ребер, инцидентных построенному фрагменту дерева – пунктиром.

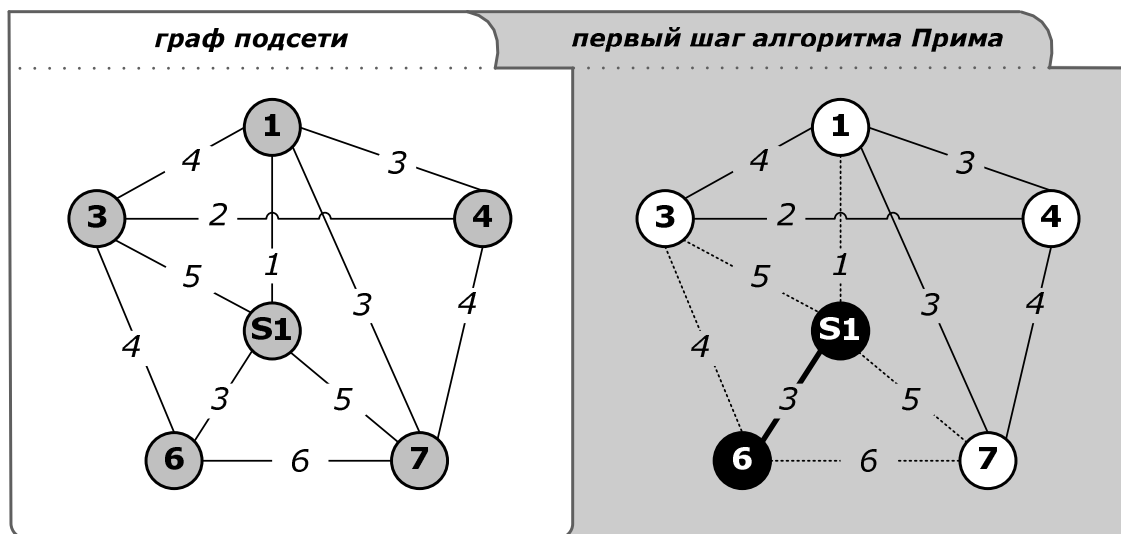


Рис. 33

На втором шаге алгоритма из множества весов инцидентных ребер $\{4, 6, 5, 5, 1\}$ выбирается ребро с весом 1, в дерево добавляется вершина **1**, а множество рассматриваемых весов ребер расширяется до $\{4, 6, 5, 5, 4, 3, 3\}$. На третьем шаге из этого множества выбирается ребро с весом 3, в дерево добавляется вершина **4**, а в множество, рассматриваемое на следующих шагах, включаются веса всех оставшихся ребер графа (рис. 34).

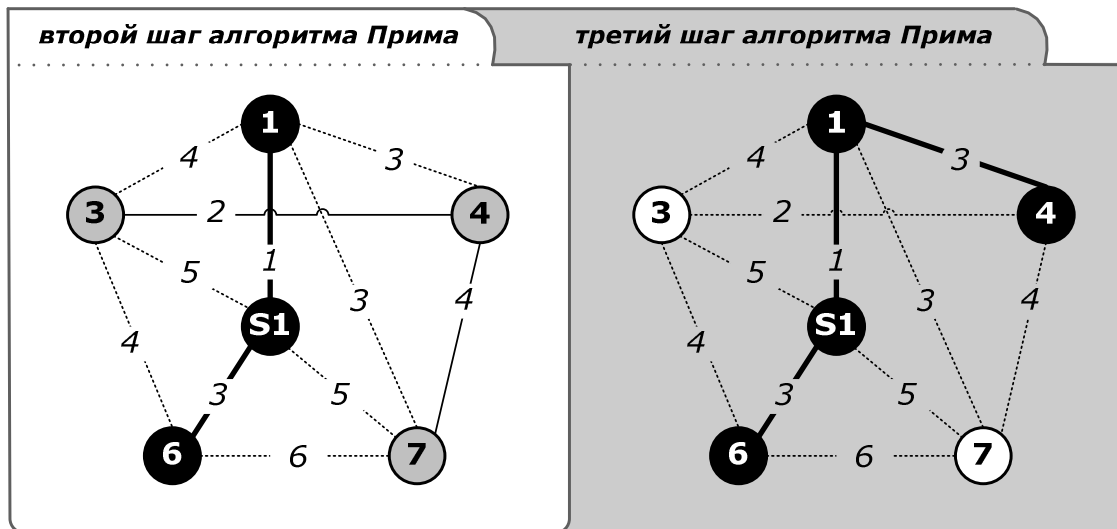


Рис. 34

На четвертом шаге алгоритма в дерево добавляется ребро с весом 2 и вершина 3; на пятом – ребро с весом 3 и вершина 7. Результат построения минимального связывающего дерева для выделенного графа подсети первого сервера приведен на рис. 35. Суммарный вес этого дерева равен 12.

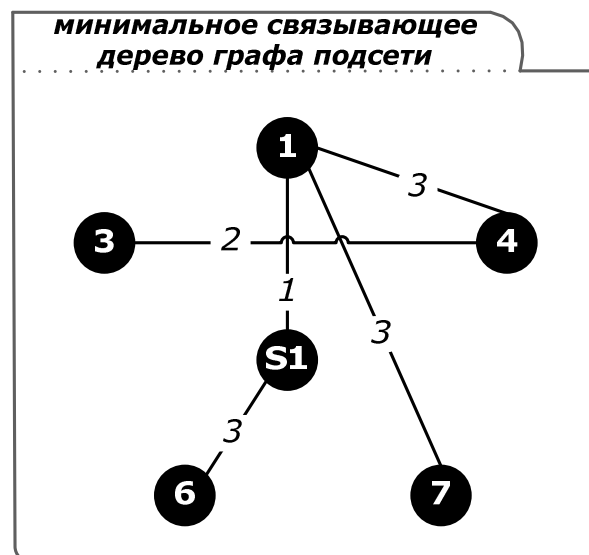


Рис. 35

Алгоритм Краскала, в отличие от алгоритма Прима, не требует прохода по всем вершинам графа. Если m – количество вершин графа, то для построения минимального дерева необходимо $(m - 1)$ ребер. Изначально все вершины графа окрашивают в различные цвета. Далее, на

каждом шаге алгоритма выбирают ребро минимального веса между смежными вершинами разного цвета. Выбор очередного ребра $r = (v_i, v_j)$, где v_i и v_j имеют разные цвета, возможен, если он не образует циклов в уже сформированном дереве. При выборе ребра $r = (v_i, v_j)$ вершина v_j и все, окрашенные в ее цвет (т.е. ранее уже включенные в построенные фрагменты дерева) перекрашиваются в цвет вершины v_i . Построение завершается, когда дерево включает все вершины графа, т. е. все вершины имеют общий цвет.

ПРИМЕР реализации алгоритма Краскала



Пример формальной декомпозиции информационной системы организации. Пусть информационная структура организации имеет вид, представленный на рис. 36.

Построим ориентированный граф информационной структуры организации, заменяя при построении двунаправленные каналы передачи информации симметричными дугами (рис. 37).

Формальные методы декомпозиции строятся на идентификации подсистем путем разбиения графа на максимальные сильно связанные подграфы. Орграф называется *сильно связным*, если для двух любых различных его узлов x_i и x_j существует, по крайней мере, один путь, соединяющий эти узлы. Это означает, что любые два узла сильно связного графа взаимно достижимы. Физический смысл сильной связности – наличие обратных связей между всеми узлами или взаимозависимость (взаимовлияние) всех переменных в графе. Важность понятия «сильная связность» вытекает из того, что практически все целенаправленные системы строятся на основе принципа обратной связи, когда информация о выходных параметрах системы, подсистемы или элемента используется для формирования управляющего воздействия (см. рис. 11).

Одним из методов разбиения графа на максимальные сильно связанные подграфы для формального выделения подсистем в структуре системы является *матричный метод*, который требует построения матриц достижимости и контрдостижимости.

Матрицей достижимости называется квадратная матрица $M_A = [m_{ij}]$, $i, j = \overline{1, m}$, где m – число узлов, т.е. $M_{m \times m}$, для которой:

- $m_{ij} = 1$, если узел x_j достижим из x_i ;
- $m_{ij} = 0$, если узел x_j недостижим из x_i .

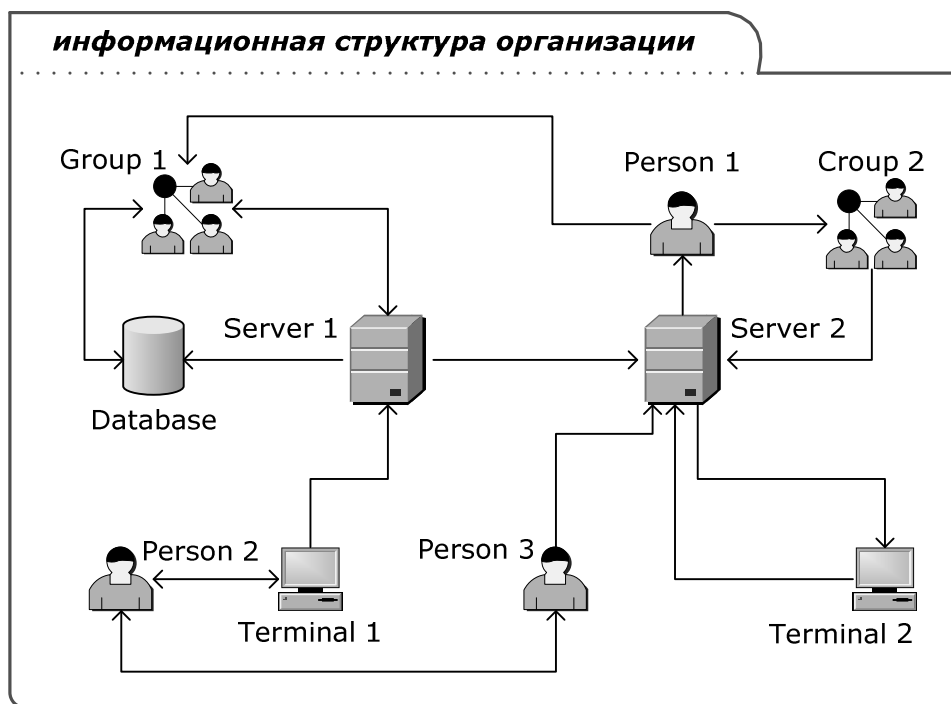


Рис. 36

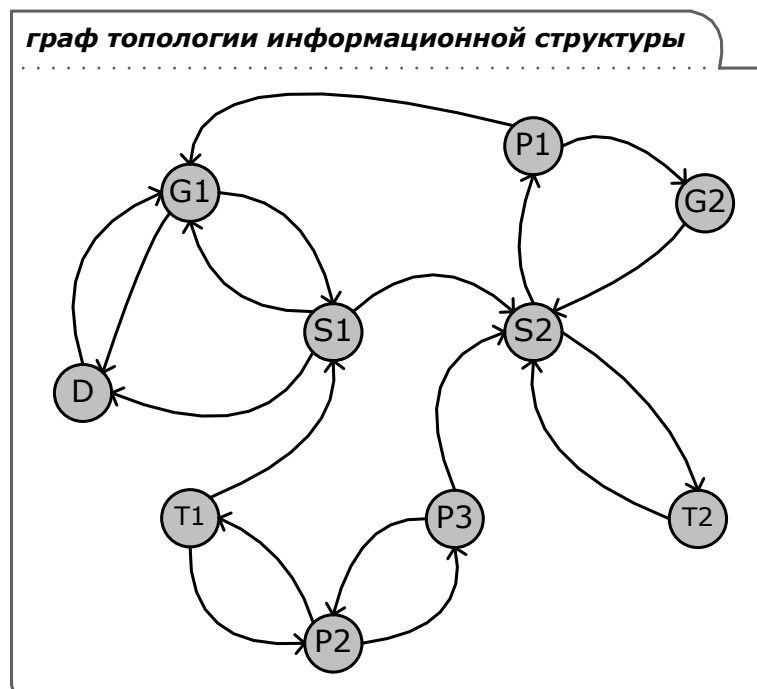


Рис. 37

Матрицей контрдостижимости называется транспонированная матрица достижимости.

Определить наличие или отсутствие в графе пути из x_i в x_j можно проанализировав матрицу смежности. Как уже отмечалось, если возвести матрицу смежности M_C в k -тую степень, то каждый элемент матрицы M_C^k будет равен числу путей из узла x_i в узел x_j длиной в k дуг. Так способ анализа тем более удобен для построения матрицы достижимости, так как для графов топологий реальных систем достижимость, как правило, определяется с учетом ограничения длины рассматриваемых путей.

Для решения задачи декомпозиции организационной структуры организации будем рассматривать достижимость при ограничении $k \leq 2$.

ПРИМЕРЫ реализации всех видов расчетов по декомпозиции структуры ИС матричным методом – см. файл примера к Лабораторной работе № 3



Матрица смежности и квадрат матрицы смежности графа топологии информационной структуры организации, изображенного на рис. 37, приведены на рис. 38.

матрица смежности										
	G1	P1	G2	S1	S2	D	T1	P3	T2	P2
G1	0	0	0	1	0	1	0	0	0	0
P1	1	0	1	0	0	0	0	0	0	0
G2	0	0	0	0	1	0	0	0	0	0
S1	1	0	0	0	1	1	1	1	0	0
S2	0	1	0	0	0	0	0	0	1	0
D	1	0	0	0	0	0	0	0	0	0
T1	0	0	0	1	0	0	0	0	0	1
P3	0	0	0	0	1	0	0	0	0	1
T2	0	0	0	0	1	0	0	0	0	0
P2	0	0	0	0	0	0	1	1	0	0

квадрат матрицы смежности										
	G1	P1	G2	S1	S2	D	T1	P3	T2	P2
G1	2	0	0	0	1	1	0	0	0	0
P1	0	0	0	1	1	1	0	0	0	0
G2	0	1	0	0	0	0	0	0	1	0
S1	1	1	0	1	0	1	0	0	1	0
S2	1	0	1	0	1	0	0	0	0	0
D	0	0	0	1	0	1	0	0	0	0
T1	1	0	0	0	1	1	1	1	0	0
P3	0	1	0	0	0	0	1	1	1	0
T2	0	1	0	0	0	0	0	0	1	0
P2	0	0	0	1	1	0	0	0	0	2

Рис. 38

Для построения матрицы достижимости при ограничении достижимости длиной пути, равной 1 или 2 дуги ($k \leq 2$) следует поэлементно сложить M_C и M_C^2 . В результате получится матрица, ненулевые элементы которой показывают наличие путей из узла x_i в узел x_j длиной в $k \leq 2$ дуг. Замена ненулевых элементов этой матрицы на 1 дает матрицу достижимости. Матрица достижимости графа топологии информационной структуры M_A и матрица контрдостижимости M_A^T приведены на рис. 39.

матрица достижимости										
	G1	P1	G2	S1	S2	D	T1	P3	T2	P2
G1	1	0	0	1	1	1	0	0	0	0
P1	1	0	1	1	1	1	0	0	0	0
G2	0	1	0	0	1	0	0	0	1	0
S1	1	1	0	1	1	1	0	0	1	0
S2	1	1	1	0	1	0	0	0	1	0
D	1	0	0	1	0	1	0	0	0	0
T1	1	0	0	1	1	1	1	1	0	1
P3	0	1	0	0	1	0	1	1	1	1
T2	0	1	0	0	1	0	0	0	1	0
P2	0	0	0	1	1	0	1	1	0	1

матрица контрдостижимости										
	G1	P1	G2	S1	S2	D	T1	P3	T2	P2
G1	1	1	0	1	1	1	1	0	0	0
P1	0	0	1	1	1	0	0	1	1	0
G2	0	1	0	0	1	0	0	0	0	0
S1	1	1	0	1	0	1	1	0	0	1
S2	1	1	1	1	1	0	1	1	1	1
D	1	1	0	1	0	1	1	0	0	0
T1	0	0	0	0	0	0	1	1	0	1
P3	0	0	0	0	0	0	1	1	0	1
T2	0	0	1	1	1	0	0	1	1	0
P2	0	0	0	0	0	0	1	1	0	1

Рис. 39

Матричный алгоритм формальной декомпозиции состоит в выполнении следующих действий:

1) строится квадратная матрица связности $M_L = [m_{ij}]$, $i, j = \overline{1, m}$, где m – число узлов, элементы которой получаются поэлементным логическим умножением матриц достижимости и контрдостижимости:

$$m_{ij} = (m_{ij} \in M_A) \wedge (m_{ij} \in M_A^T);$$

2) элементы, имеющие одинаковые строки и столбцы в матрице связности, группируются перестановкой строк и столбцов для получения блочно-диагональной матрицы, в которой элементы, равные 1, сгруппированы как можно более плотно вдоль главной диагонали матрицы. Каждая диагональная группа элементов в виде квадратного блока в перестановочной матрице связности и есть максимальный сильно связный подграф.

Результат выполнения матричного алгоритма формальной декомпозиции для рассматриваемого примера приведен на рис. 40.

матрица связности										
	G1	P1	G2	S1	S2	D	T1	P3	T2	P2
G1	1	0	0	1	1	1	0	0	0	0
P1	0	0	1	1	1	0	0	0	0	0
G2	0	1	0	0	1	0	0	0	0	0
S1	1	1	0	1	0	1	0	0	0	0
S2	1	1	1	0	1	0	0	0	1	0
D	1	0	0	1	0	1	0	0	0	0
T1	0	0	0	0	0	0	1	1	0	1
P3	0	0	0	0	0	0	1	1	0	1
T2	0	0	0	0	1	0	0	0	1	0
P2	0	0	0	0	0	0	1	1	0	1

перестановочная матрица связности										
	P3	P2	T1	S1	D	G1	S2	T2	G2	P1
P3	1	1	1	0	0	0	0	0	0	0
P2	1	1	1	0	0	0	0	0	0	0
T1	1	1	1	0	0	0	0	0	0	0
S1	0	0	0	1	1	1	0	0	0	1
D	0	0	0	1	1	1	0	0	0	0
G1	0	0	0	1	1	1	1	0	0	0
S2	0	0	0	0	0	1	1	1	1	1
T2	0	0	0	0	0	0	1	1	0	0
G2	0	0	0	0	0	0	1	0	0	1
P1	0	0	0	1	0	0	1	0	1	

Рис. 40

В первую очередь перестановке в матрице связности подвергаются полностью совпадающие строки (столбцы). В матрице связности на рис. 40 это строки узлов **T1**, **P3** и **P2**. Далее рассматриваются строки, в наибольшей степени совпадающие по элементам. К таким можно отнести строки узлов **S1**, **D** и **G1**, и с меньшей степенью соответствия – **S2**. Далее в перестановочную матрицу добавляются строки по неучтенным узлам, имеющим взаимную достижимость с **S2** – это узлы **T2** и **G2**. Последней добавляется строка **P1**.

Анализ перестановочной матрицы связности (см. рис. 40) позволяет выделить в составе графа топологии информационной структуры следующие максимально сильно связанные подграфы: (**P3**, **P2**, **P1**); (**S1**, **D**, **G1**) и (**S2**, **T2**). Остальные узлы выраженных общих связностей не имеют и могут рассматриваться как элементы системы, отличающиеся относительной изолированностью (независимостью функционирования) от других элементов (рис. 41).

Результат декомпозиции информационной структуры организации позволяет идентифицировать подсистемы, которые целесообразно выделить в отдельные подразделения, производственные отделы и т.п. [3, 19, 20, 22, 25].

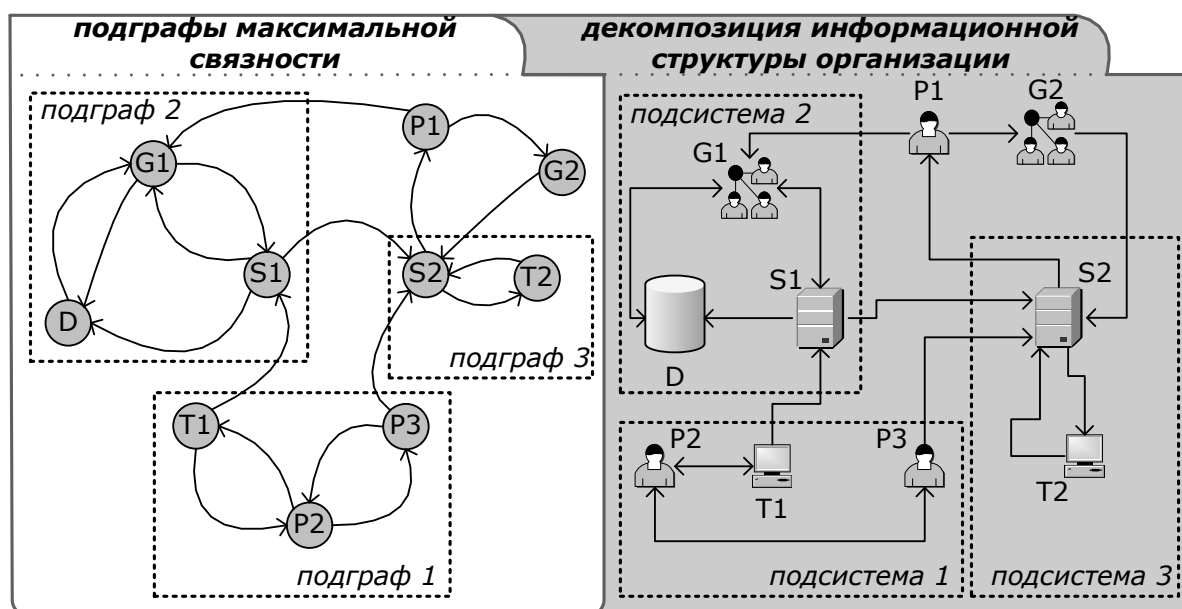


Рис. 41

Помимо решения задач декомпозиции топологические модели второго типа используются для определения структурных свойств системы. Как

уже отмечалось, структуры информационных систем тяготеют к структурам иерархического и/или сетевого типа (см. рис. 5 – 7).

К основным структурным свойствам информационных систем, определяемых с использованием теории графов, относятся:

- сетевая плотность;
- центральность;
- уязвимость;
- кластеризация.

Сетевая плотность (плотность связей) является мерой связанности между элементами структуры и определяется как отношение числа существующих к числу возможных связей. Максимально возможное число связей, т.е. максимально возможное суммарное число ребер в неориентированном графе, равно $m(m - 1)/2$, где m – количество вершин. Сетевая плотность неориентированного графа определяется по формуле

$$\Delta = \frac{L}{m(m - 1)/2} = \frac{2L}{m(m - 1)},$$

где L – число существующих связей в графе.

Максимально возможное число связей между узлами в орграфе в два раза больше, чем в неориентированном графе с тем же количеством вершин. Между любыми двумя узлами в орграфе с учетом направленности дуг может быть две разнонаправленные дуги, т.е. максимально возможное число связей орграфа равно $m(m - 1)$, где m – количество узлов. Сетевая плотность ориентированного графа определяется по формуле

$$\Delta_{or} = \frac{L}{m(m - 1)}.$$

Центральность характеризует значимость элемента в структуре системы. В анализе топологии систем с применением графов используются несколько мер центральности.

Первой простейшей мерой центральности вершины для неориентированного графа является степень вершины $d(v_i)$, равная количеству ребер, инцидентных вершине. Максимальное значение степени центральности вершины графа равно $(m - 1)$, где m – число вершин. Для сравнения степеней центральности не только внутри одного графа, но и между графами, в процессе сравнительного анализа различных структур систем рассчитывают *нормированную степень центральности* вершины графа:

$$c^*(v_i) = \frac{d(v_i)}{m - 1}.$$

Для орграфа *степень центральности* узла можно рассчитать как сумму полустепени исхода и полустепени захода $c(x_i) = d_{out}(x_i) + d_{in}(x_i)$. В этом случае нормированная степень центральности узла рассчитывается по отношению к максимально возможному числу связей обоих направлений:

$$c^*(x_i) = \frac{c(x_i)}{2(m-1)}.$$

В случае необходимости отдельно учитывать связи узла различной направленности в орграфе рассчитывают нормированные полустепени центральности исхода и захода. Критерий нормировки в этом случае, как и для ненаправленного графа, равен $(m-1)$.

Степень центральности графа в целом рассчитывается следующим образом:

$$C = \frac{\sum_{i=1}^m [c_{\max}(x_i) - c(x_i)]}{\max [c_{\max}(x_i) - c(x_i)]},$$

где $c_{\max}(x_i)$ – максимальная степень центральности узла (вершины) графа.

Степень центральности вершины (узла) является локальной характеристикой положения вершины в графе. Эта мера учитывает только смежные вершины, ближайшую окрестность и в этом смысле ограниченно информативна.

Второй мерой центральности является *плотность центральности*. Этой мерой оценивается эффективность взаимодействия элементов систем с точки зрения коммуникации. *Централью* называется элемент системы, из которого все остальные элементы достижимы с минимальным количеством шагов. На графе это узел (вершина), которая со всеми остальными узлами (вершинами) соединяется путями с минимальным числом дуг. Плотность центральности узла орграфа рассчитывается следующим образом:

$$c_{den}(x_i) = \frac{1}{\sum_{j=1}^m k_{ij}},$$

где k_{ij} – число дуг в наименьшем по количеству дуг пути $x_i x_j$.

Наилучшее значение плотности центральности обеспечивается в графе, у которого каждый узел (вершина) смежен со всеми, т.е. связан со всеми узлами путем длиной в одно ребро (дугу). Для такого графа плотность центральности максимальна и равна $1/(m-1)$. Поэтому *нормированная плотность центральности* $c_{den}^*(x_i)$ рассчитывается как

$$c_{den}^*(x_i) = \frac{c_{den}(x_i)}{1/(m-1)} = (m-1) \cdot c_{den}(x_i).$$

В третьем подходе центральность рассматривается как контроль кратчайших путей между узлами (вершинами) – путей длиной в две дуги (ребра). Так, если путь между узлами x_2 и x_3 проходит через $x_2x_4x_3$, то узел x_4 является контролирующими по отношению к узлам x_2 и x_3 и называется *посредником*. Идею этого подхода можно сформулировать следующим образом: узел тем более централен, чем больше количество других узлов, между которыми он находится (чем большее число кратчайших путей в графе он контролирует). Мерой центральности в этом подходе является *степень посредничества*:

$$c_{mediate}(x_i) = \sum_{j,k=1}^m \frac{g_{jk}(x_i)}{g_{jk}}; \quad i \neq j, k,$$

где $g_{jk}(x_i)$ – число кратчайших путей из x_j и x_k , которые проходят через x_i ; g_{jk} – общее число кратчайших путей из x_j и x_k . Таким образом, степень посредничества показывает долю путей из x_j и x_k , контролируемых x_i . Этот показатель нормализуется делением на максимально возможное число кратчайших путей в графе, равное $(m-1)(m-2)/2$. Нормированная степень посредничества равна:

$$c_{mediate}^*(x_i) = \frac{c_{mediate}(x_i)}{(m-1)(m-2)/2} = 2 \frac{c_{mediate}(x_i)}{(m-1)(m-2)}.$$

Уязвимость является структурным свойством, характеризующим устойчивость системы при утрате функциональности какого-либо из элементов системы или связи. Мерами уязвимости структуры являются:

- *связность*: минимальное число вершин (узлов) графа, удаление которых приводит к разделению графа на несвязанные подграфы;
- *реберная связность*: минимальное число ребер (дуг), удаление которых приводит к разделению графа на несвязанные подграфы.

Кластеризация это выявление в структуре системы кластеров: комплексов элементов, в которых любой элемент связан с каждым из оставшихся элементов. Если кластеризация выполняется без ограничений весов ребер в кластере, то используют алгоритмы построения минимального связывающего дерева. Если на максимально возможный вес ребра в кластере наложено ограничение, то применяют алгоритм выделения связных компонент. Связной компонентой графа называется подмножество его вершин, в котором любые две вершины можно соединить путем, целиком лежащим в этом подмножестве. При реализации алгоритма из графа удаля-

ются все ребра с весом больше ограничения и кластерами считают получившиеся в результате компоненты связности. Иногда этот алгоритм применяют от обратного, подбирая такое ограничение веса ребер $R \in (\max r_{ij}; \min r_{ij})$, при котором граф после удаления разделяется на несколько связанных компонент. Для подбора параметра R рекомендуется строить гистограмму распределения попарных расстояний r_{ij} . В системах с выраженной кластерной структурой гистограмма имеет два четких пика: в зоне небольших внутриклассовых расстояний и в зоне больших межклассовых расстояний. Параметр R задается как расстояние, соответствующее точке минимума между этими пиками [3, 20, 29-31].

ПРИМЕРЫ расчета и анализа структурно-топологических характеристик графов структур ИС и топологий ВС – см. Лабораторные работы № 3 и № 4



2.3.3. Использование теории графов в задачах исследования операций

Помимо топологического моделирования, теория графов в системном анализе используется в методах решения задач исследования операций для нахождения путей с заданными свойствами и оптимизации потоков. Рассмотрим использование теории графов в решениях перечисленных классов задач.

Как уже отмечалось (см. раздел 2.2.2. *Системотехнические задачи исследования операций*), в задачах календарного планирования рассматриваются два типа ограничений. Ограничения первого типа описывают взаимную зависимость работ. Ограничения второго типа определяют объем ресурса, который может быть выделен на реализацию проекта.

Ограничения первого типа носят логический характер: например, выполнению работы p_i должна предшествовать некоторая совокупность работ, без завершения которой она начаться не может. Такие ограничения легко представить в терминах теории графов: узлы должны соответствовать работам, а дуги задавать последовательность их выполнения.

На рис. 42 представлен пример задания условий предшествования для задачи составления расписаний в виде графа, узлами которого являются виды работ, а дуги задают частичный порядок выполнения. Одно из возможных решений задачи составления расписания по условию минимального времени окончания всех видов работ на 2-х параллельных устройствах обслуживания с равной длительностью выполнения всех видов работ τ , показано на диаграмме.

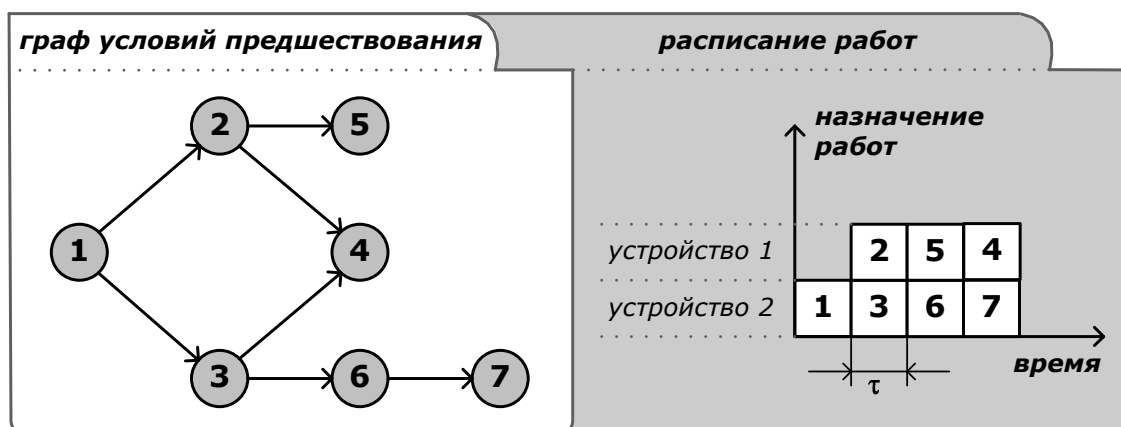


Рис. 42

В задачах сетевого планирования комплекс выполняемых операций также представляют в виде ориентированного графа, называемого *сетевым графиком*, или *сетью*. Операции изображают взвешенными дугами – вес дуги соответствует времени выполнения операции. Узлы представляют собой некоторые события – начало выполнения проекта, завершение выполнения очередной операции и т.п. Дуги, входящие в узел, соответствуют операциям, которые должны быть закончены раньше, чем можно будет начать операции, изображенные исходящими дугами. В сетевом графике всегда существует единственный исток (вход), соответствующий началу выполнения операций проекта, и единственный сток (выход), соответствующий завершению проекта. Также сетевой график должен удовлетворять следующему условию: в любой его узел существует путь из входа и из любого узла существует путь в выход.

ПРИМЕР сетевого графика (сети)



Ограничения условий предшествования в задачах календарного планирования могут иметь сложную форму: работы могут быть взаимозаменяемы, вестись параллельно и т.п.

Для представления и анализа условий взаимозаменяемости используются так называемые И/ИЛИ графы. И/ИЛИ граф является иерархической структурой и описывает развитие некоторого процесса от начального состояния (узла – истока графа) к множеству конечных состояний (узлов – стоков графа). Все промежуточные узлы графа делятся на два

типа: И и ИЛИ. Целью построения и анализа И/ИЛИ графа является доказательство существования одного или множества альтернативных путей от истока к стоку (стокам), т.е. доказательство возможности и определение числа вариантов осуществления определенной последовательности работ.

На каждом шаге построения И/ИЛИ графа исследуется *разрешимость* узлов рассматриваемого уровня – возможность прокладки через каждый узел пути от истока к стоку. Узел И/ИЛИ графа разрешим, если:

- разрешим хотя бы один из его дочерних (инцидентных на более низком уровне) ИЛИ-узлов;
- разрешимы все его дочерние И-узлы.

Рассмотрим построение И/ИЛИ графа модели процесса обработки данных в распределенной системе. Пусть элементарный объем данных – задача – требует выполнения трех операций обработки, причем порядок выполнения операций может быть любым, но выполнение первой и второй операции может быть осуществлено устройством №1, а выполнение третьей – устройством №2. Обозначим состояние задачи в процессе решения $S=(s_1, s_2, s_3)$, где s_i – состояние i -той операции по обработке (1 – выполнена, 0 – не выполнена). Введем в рассмотрение оператор f_i , действие которого изменяет общее состояние задачи в процессе решения путем смены состояния i -той операции с 0 на 1, например:

$$f_1(S) = f_1(0, s_2, s_3) = (1, s_2, s_3).$$

Построению И/ИЛИ графа начинается с вершины-истока, соответствующей состоянию начала обработки данных: (0, 0, 0). Дуги графа помечаются операторами смены состояния задачи в процессе выполнения операций обслуживания. Последовательное выполнение операций моделируется последовательно-смежными дугами и узлами типа ИЛИ. Если возможно одновременное выполнение нескольких операций, то при построении соответствующего пути используются параллельно-смежные дуги и вершины типа И. Полный И/ИЛИ граф рассматриваемого примера приведен на рис. 43 (пунктиром показаны параллельно-смежные дуги, соответствующие параллельным операциям обработки, т.е. используемые при разрешении вершин типа И).

Модели процессов обработки, последовательностей этапов в проектировании и т.п. в виде И/ИЛИ графов называются *перестановочными*, поскольку описывают множество структур процесса при изменении порядка выполнения операций.

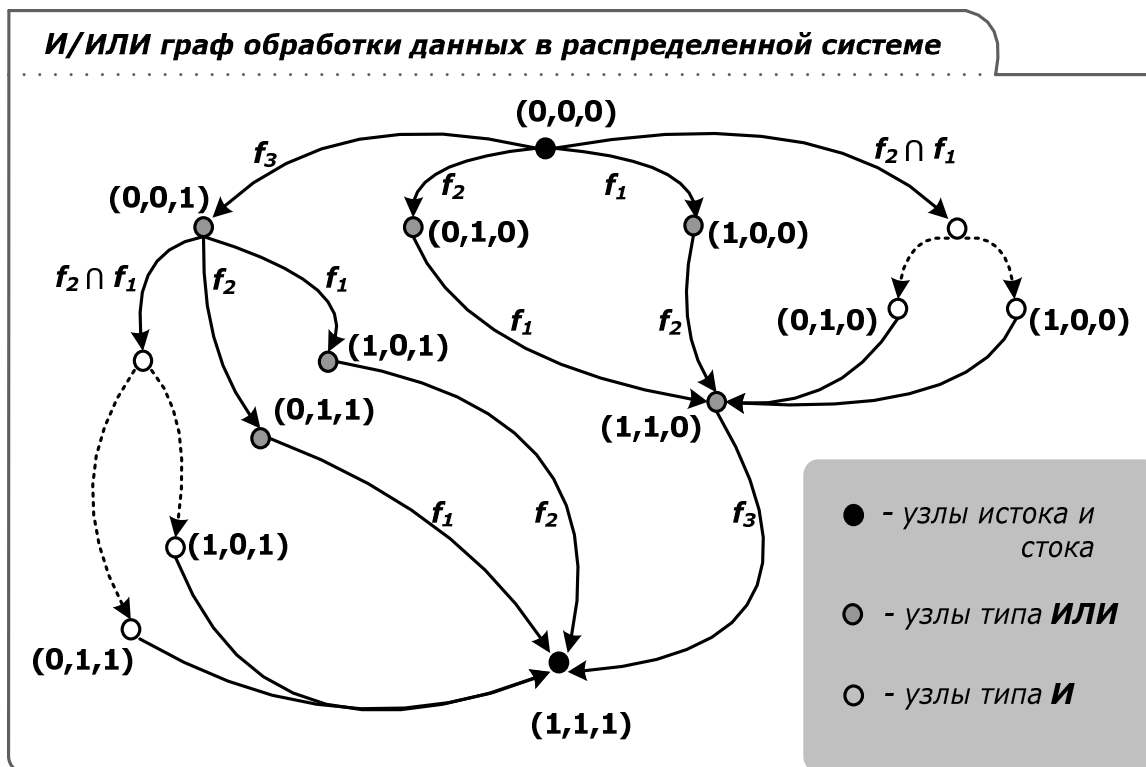


Рис. 43

Решение задач оптимизации (синтеза процесса с заданными свойствами, планирование проекта и т.п.) требует использования взвешенных графов, в которых каждой дуге соответствует не только оператор смены состояния, но и некоторая характеристика его выполнения, в простейшем случае – время выполнения операции.

В задачах сетевого планирования с заданными условиями предшествования на последовательно и параллельно выполняемые работы важным понятием является понятие *критического пути*. Отдельная операция считается критической, если задержка ее выполнения приводит к эквивалентной задержке всего проекта, т.е. к увеличению общего времени выполнения проекта T . Путь в сетевом графике от начального события к конечному, состоящий только из критически операций, носит название *критического*, и соответствует пути максимальной длины. Проще искать путь максимальной длины на сетевом графике с *правильной* нумерацией: узлу-истоку сетевого графика присваивается нулевой номер и далее узлы нумеруют так, чтобы для любой дуги (i, j) имело место $j > i$ (см. сетевой график на рис. 44).

Если t_{ij} – вес дуги (i, j) , то алгоритм построения критического пути в сетевом графике с правильной нумерацией имеет вид:

- 1) узел истока помечается индексом $\lambda_0 = 0$;
- 2) все узлы просматриваются в цикле в порядке возрастания номеров с присвоением узлу индекса $\lambda_j = \max(\lambda_i + t_{ij})$ до тех пор, пока не будет определен индекс стока (равен длине критического пути);
- 3) обратным ходом от стока к истоку строим критический путь, выбирая такие узлы, для которых $t_{ij} = \lambda_j - \lambda_i$.

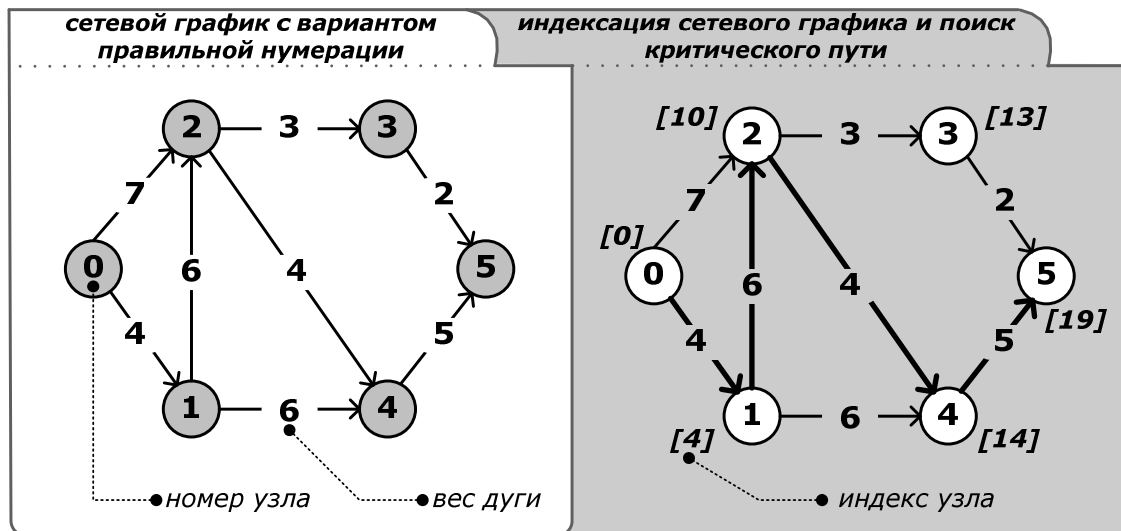


Рис. 44

В общем случае, при произвольной нумерации вершин, после первого шага алгоритма на каждом последующем шаге рассматривается множество узлов, смежных с уже размеченными узлами. Присвоение индексов выполняется по тому же правилу, что и в алгоритмах для сетевых графиков с правильной нумерацией.

Пример применения алгоритма для индексации графа и построения критического пути (выделен жирными линиями) показан на рис. 44.

С использованием аналогичного алгоритма решается задача о поиске пути минимальной длины: следует только заменить во втором шаге формулу присвоения значения индекса на $\lambda_j = \min(\lambda_i + t_{ij})$. Пример применения алгоритма для индексации графа и построения пути минимальной длины (выделен жирными линиями) показан на рис. 45.

Критический путь в сети также называют путем с нулевым резервом времени. Резерв времени операции (i, j) это интервал времени, в течение которого начало выполнения операции может задерживаться, не приводя к увеличению времени наступления последнего события, т.е. времени окончания всего комплекса операций T . Понятно, что дуги, входящие в критический путь не имеют резерва времени, т.е. задержка выполнения соответствующих

им операций недопустима. Для всех остальных операций резерв времени вычисляется как разница между самым поздним временем наступления конечного события в сети, не приводящим к увеличению общего времени выполнения проекта T , и самым ранним возможным временем наступления этого же события. Если и критический путь, и путь минимальной длины уже построен, то резерв времени выполнения операции вычисляется для соответствующей дуги как разница индексов конечной вершины в построении критического пути и пути минимальной длины. Так, резерв времени выполнения операции, соответствующей дуге (0, 1), составляет $[10] - [7] = 3$ (см. рис. 44 и рис. 45). В пределах резервного времени можно смещать начало выполнения соответствующих работ без изменения длительности критического пути.

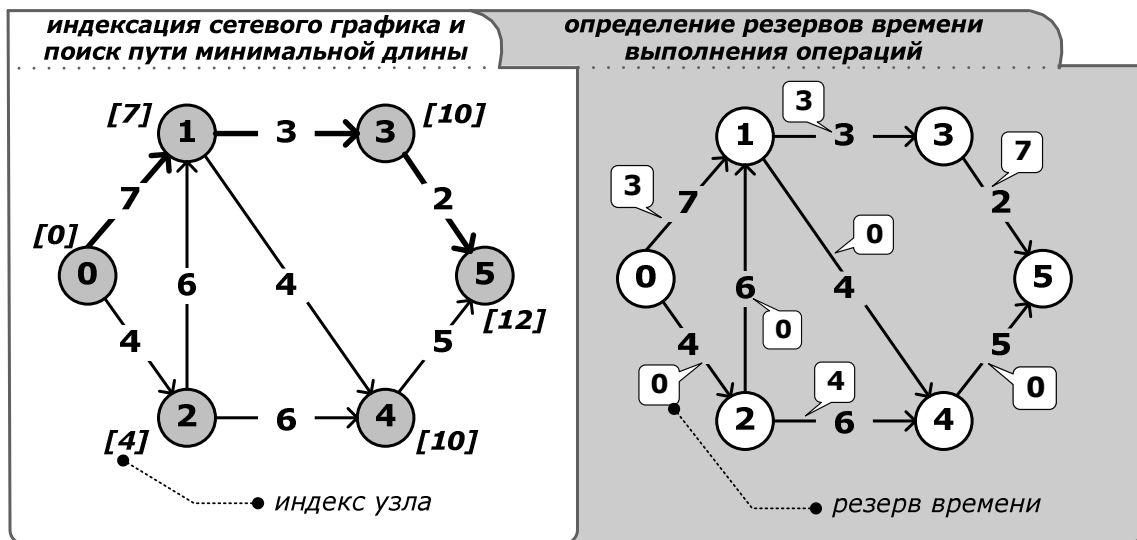


Рис. 45

В исследовании операций информационных систем и вычислительных сетей задача поиска пути часто формулируется как поиск пути *максимальной надежности*. В такой постановке вес дуги сетевого графика задает вероятность установления надежного соединения между узлами сети, и пути максимальной надежности соответствует путь максимальной длины. Аналогично решается задача поиска пути *максимальной эффективности* – в этом случае вес дуги интерпретируется как эффективность выполнения операции, равная отношению эффекта операции к затратам на ее осуществление.

В решениях транспортных задач в классической постановке и для анализа коммуникационных сетей, в т.ч. вычислительных, используются сети, у которых каждой дуге (i, j) приписывается положительное число c_{ij} , называемое *пропускной способностью*. Содержательной интерпретацией пропускной

способности для классической транспортной задачи является максимальное количество груза, которое можно перевести в единицу времени по транспортной магистрали, отображаемой дугой. Для вычислительных сетей пропускная способность дуги графа соответствует пропускной способности канала связи, описываемого данной дугой.

Для произвольного внутреннего узла x сети будем обозначать через $a^+(x)$ множество всех входящих в него дуг с размерностью, равной полустепени захода $d_{in}(x)$, а через $a^-(x)$ – множество выходящих дуг с размерностью, равной полустепени исхода $d_{out}(x)$. Очевидно, что для истока сети $a^+ = \emptyset$, а для стока, соответственно, $a^- = \emptyset$.

Потоком f сети называется множество значений потоков всех дуг сетевого графика $\{f_{ij}\}$, удовлетворяющих условиям:

- 1) $0 \leq f_{ij} \leq c_{ij}$ для каждой дуги $(i, j) \in A$ (A – множество всех дуг сети);
- 2) сумма потоков дуг, входящих во внутренние узлы сети, должна быть равна сумме потоков исходящих дуг (условие сохранения потока):

$$\sum_{a \in a^+(x)} f(x) = \sum_{a \in a^-(x)} f(x).$$

Величиной потока называется величина

$$M(f) = \sum_{a \in a^+(x_m)} f(x_m) = \sum_{a \in a^-(x_0)} f(x_0),$$

где x_0 – исток, а x_m – сток сети.

На рис. 46 показан пример сети, для которого число, вписанное в примыкающий к дуге квадрат, представляет пропускную способность, а вес дуги – заданную величину потока.

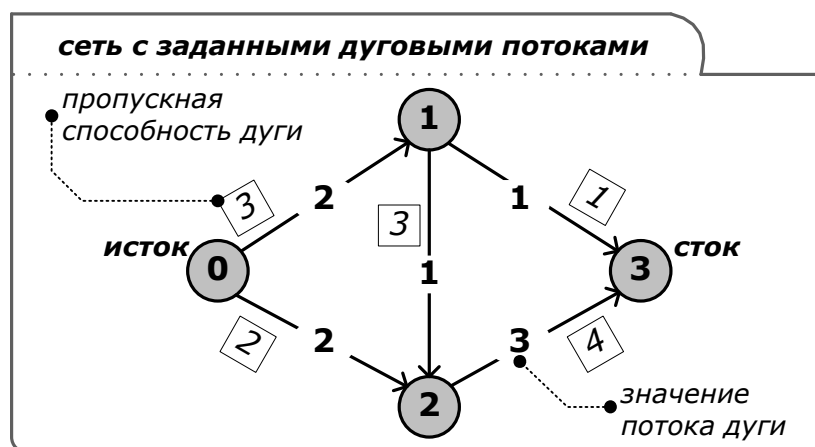


Рис. 46

Основной задачей оптимизации потоков является задача определения величин дуговых потоков f_{ij} так, чтобы обеспечить поток сети максимально возможной величины – задача о *максимальном потоке*. Как видно, для сети на рис. 46 величина потока $M(f) = 4$, однако поток с такой величиной не является максимальным: на графе существует направленный путь $(0, 1, 2, 3)$, все дуговые потоки которого меньше пропускных способностей дуг. Как видно, на каждой дуге этого пути поток можно увеличить на 1, т.е. направить по этому пути дополнительный поток.

Иногда поток можно увеличить и при отсутствии таких «недогруженных» направленных путей из истока в сток. Пример такого случая показан на рис. 47: в исходном варианте сети с величиной потока $M(f) = 4$ имеется последовательность разнонаправленных дуг, в которой дуговые потоки каждой дуги меньше пропускных способностей – (i, a, b, c, s) .

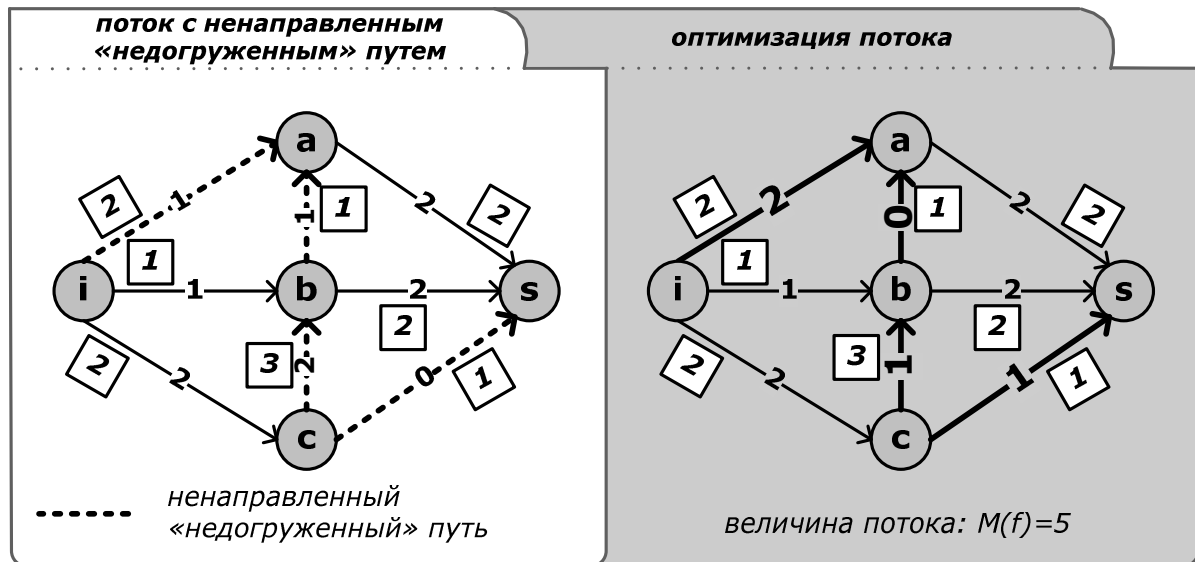


Рис. 47

Будем называть такую последовательность ненаправленным «недогруженным» путем. Результат увеличения потока до $M(f) = 5$ обеспечивается увеличением дуговых потоков на прямых дугах ненаправленного пути, т.е. дугах, направленных от истока к стоку: (i, a) и (c, s) , и согласованным уменьшением дуговых потоков обратных дуг: (a, b) и (b, c) , что необходимо для обеспечения условия сохранения потока в промежуточных вершинах. Понятно, что этот прием применим, только если все прямые дуги ненаправленного пути недогружены, а все обратные дуги имеют ненулевую величину дугового потока.

В целом можно утверждать, что если в сети с потоком f существует недогруженный путь любого вида, то этот поток *не максимален*. При идентификации такой ситуации возникают две задачи: определение значения максимально возможного потока в данной сети и построение максимального потока.

Определение значения максимально возможного потока в сети связано с понятием разреза. Рассмотрим произвольное множество узлов графа $X^* := \{x \mid X^* \subset X, \text{исток} \in X^*, \text{сток} \notin X^*\}$, т.е. любое множество узлов, содержащее исток и не содержащее сток. *Разрезом* сети называется множество дуг $A^* := \{a \mid (i, j) \in A, i \in X^*, j \notin X^*\}$, т.е. множество всех дуг, исходящих из узлов множества X^* . Это множество дуг называется разрезом (или сечением) сети, т.к. удаление всех дуг $a \in A^*$ из графа приводит к разрыву всех путей, ведущих из истока в сток. Сеть становится «разрезанной» на две части, первой из которых принадлежит исток, второй – сток.

Величина $C(A^*) = \sum_{(i,j) \in A^*} c_{ij}$ называется *пропускной способностью*

*разреза A^** . Разрез с минимальной пропускной способностью называется *минимальным разрезом*.

Теорема о максимальном потоке сети гласит: *величина максимального потока в сетевом графике равна пропускной способности минимального разреза*. Т.о., величину максимального потока можно найти, вычисляя пропускные способности всех разрезов и выбирая среди них минимальный. Однако этот результат имеет практическое значение только в качестве средства проверки решения задачи о максимальном потоке, поскольку не дает никакой информации о способе назначения величин дуговых потоков f_{ij} для обеспечения максимального потока.

Результат расчета пропускных способностей всех разрезов сети, показанной на рис. 46, приведен в табл. 2.

Таблица 2

Разрез	Множество узлов X^*	Множество дуг разреза A^*	Пропускная способность разреза $C(A^*)$
A^*_1	0	(0,1); (0,2)	$3 + 2 = 5$
A^*_2	0, 1	(0,2); (1,2); (1,3)	$2 + 3 + 1 = 6$
A^*_3	0, 2	(0,1); (2,3)	$3 + 4 = 7$
A^*_4	0, 1, 2	(1,3); (2,3)	$1 + 4 = 5$

Таким образом, максимально возможная величина потока в сети, приведенной на рис. 46, определяется двумя минимальными разрезами – A^*_1 и A^*_4 – и равна $M_{\max}(f) = 5$. Это подтверждает выдвинутое выше пред-

положение о возможности увеличения на единицу значений дуговых потоков на пути (0, 1, 2, 3).

Рассмотрим *алгоритм построения максимального потока* в сети. Алгоритм состоит в выполнении в цикле прямого и обратного ходов. Прямой ход состоит в индексации узлов сети, обратный – в изменении потока. Для начала выполнения алгоритма необходимо задать любой произвольный поток сети, например – с нулевыми значениями всех дуговых потоков.

При выполнении *прямого хода алгоритма* узлам сети присваиваются двухпозиционные индексные обозначения. Ход начинается с присвоения истоку специального индекса $[-, \infty]$. Далее, для каждой проиндексированного узла i с индексом $[x, e]$ рассматривается множество смежных с ней неиндексированных узлов, для которых индексы определяются по следующему правилу:

- если дуга направлена из i в j и поток этой дуги f_{ij} меньше ее пропускной способности c_{ij} , то узлу j присваивается индекс $[i+, \min(e, c_{ij}-f_{ij})]$;
- если дуга направлена из j в i и поток этой дуги f_{ji} больше нуля, то узлу j присваивается индекс $[i-, \min(e, f_{ji})]$.

Прямой ход алгоритма, т.е. процесс индексации узлов, заканчивается в двух случаях:

- больше ни один узел проиндексировать нельзя, но сток неиндексирован – это означает, что найденный поток является максимальным;
- определен индекс стока, что означает переход к выполнению обратного хода алгоритма.

При выполнении *обратного хода алгоритма* происходит изменение значений дуговых потоков. Процесс начинается со стока. Если сток имеет индекс $[k+, d]$, то дуговые потоки будут изменяться на величину d следующим образом:

- если рассматриваемый узел j имеет индекс $[i+, x]$, то к f_{ij} прибавляется d и далее рассматривается узел i ;
- если рассматриваемый узел j имеет индекс $[i-, x]$, то из f_{ij} вычитается d и далее рассматривается узел i .

Изменение потока продолжается до достижения истока. После этого индексация всех узлов отменяется и заново выполняется прямой ход алгоритма. Процесс повторяется до нахождения максимального потока.

Рассмотрим пример построения максимального потока в сети (рис. 48).

На первом шаге алгоритма при выполнении прямого хода после присвоения истоку индекса $[-, \infty]$ рассматриваются смежные с ним узлы – 1 и 2. Дуги в эти вершины направлены от истока, поэтому первое позиционное

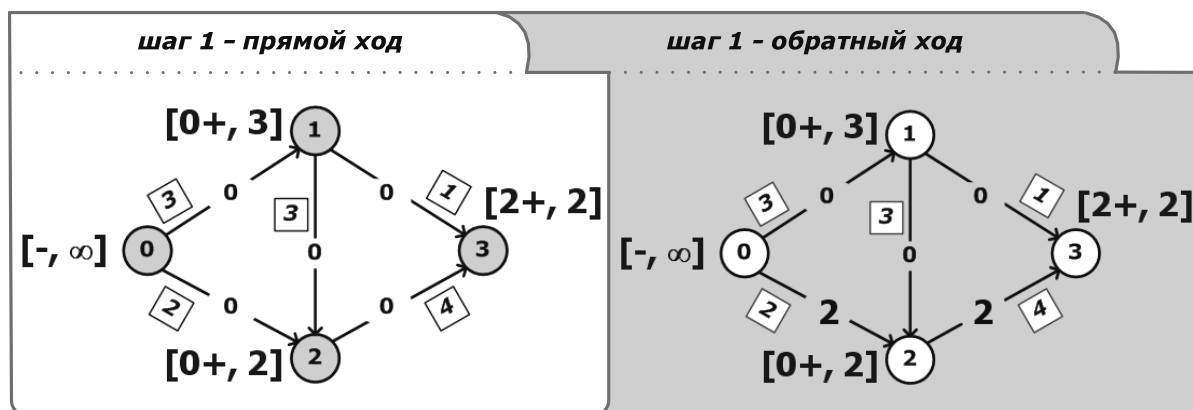


Рис. 48

обозначение индексов этих узлов имеет вид «0+» (узлы индексированы из узла 0). Второе позиционное обозначение определяется по правилу $\min(e, c_{ij}-f_{ij})$, но для истока $e = \infty$ и при выполнении первого значения дуговых потоков еще равны 0, поэтому для дуг (0, 1) и (0, 2) $\min(e, c_{ij}-f_{ij}) = c_{ij}-0 = c_{ij}$. После индексации узлов 1 и 2 узел 3 может быть равнозначно проиндексирован из этих узлов: в примере индексация осуществлена из узла 2, и для этого случая $\min(e, c_{23}-f_{23}) = \min(2, 4-0) = \min(2, 4) = 2$, таким образом, полный индекс узла 3 равен $[2+, 2]$. На обратном ходе алгоритма выполняется изменение значений дуговых потоков на величину d (d – вторая позиция индекса стока, полученная при индексации сети в результате прямого хода). В нашем случае $d = 2$, поэтому, согласно правилам обратного хода, с учетом того, что сток (узел 3) индексирован из узла 2, к начальной (нулевой) величине дугового потока f_{23} прибавляется 2 и далее рассматривается узел 2. Узел 2 индексирован из узла 0, поэтому к начальной (нулевой) величине дугового потока f_{02} также прибавляется 2, и выполнение первого шага алгоритма заканчивается.

На втором шаге алгоритма (рис. 49) при выполнении прямого хода узел 1 снова индексируется из истока, но индексация узла 2 из истока уже невозможна: дуга (0, 2) имеет значение дугового потока, равное пропускной способности. Единственный возможный вариант – проиндексировать узел 2 из узла 1: для этого случая $\min(e, c_{12}-f_{12}) = \min(3, 3-0) = \min(3, 3) = 2$, тогда полный индекс узла 2 равен $[1+, 3]$. После индексации узлов 1 и 2 узел 3 может быть равнозначно проиндексирован из этих узлов: в примере индексация осуществлена из узла 2, и для этого случая $\min(e, c_{23}-f_{23}) = \min(2, 4-2) = \min(2, 2) = 2$, и полный индекс стока равен $[2+, 2]$. На обратном ходе алгоритма выполняется изменение значений дуговых потоков дуг (2, 3), (1, 2) и (0, 1) на величину $d = 2$ (измененные значения показаны на рис. 49 крупным шрифтом).

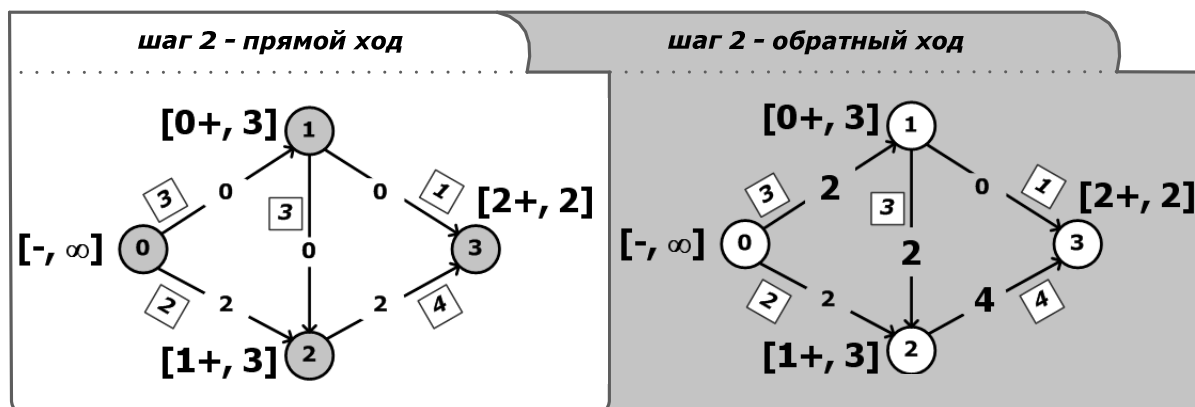


Рис. 49

Результат выполнения прямого и обратного ходов третьего шага алгоритма показан на рис. 50: как видно, в результате выполнения обратного хода произошло изменение значений дуговых потоков дуг (1, 3) и (0, 1). Попытка выполнения четвертого шага алгоритма показывает, что из истока выполнить индексацию ни одного узла невозможно: дуги смежных с ним узлов 1 и 2 имеют значения дуговых потоков, равные пропускной способности. Выполнение алгоритма завершено, а поток сети, построенный на обратном ходе третьего шага, является максимальным. Подтверждением максимальности построенного потока является выполненный ранее анализ разрезов данной сети (см. табл. 2), показавший, что $M_{\max}(f) = 5$. Именно эта величина потока и получена в результате построения: сумма потоков дуг, исходящих из истока, равно как и сумма потоков дуг, входящих в сток, составляет 5 (рис. 50).

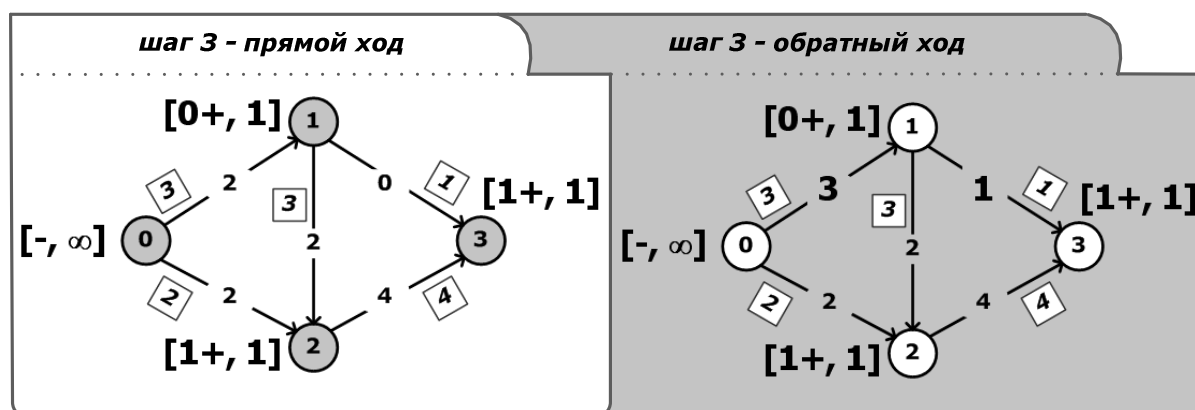


Рис. 50

Альтернативной задачей оптимизации потоков является *задача поиска потока минимальной стоимости*. В постановке этой задачи для каждой дуги сети $(i; j)$, помимо пропускной способности c_{ij} , задается параметр s_{ij} , величина которого интерпретируется как затраты (например, затраты на передачу единичного объема данных по каналу связи из узла i вычислительной сети в узел j). Задача поиска потока минимальной стоимости заключается в нахождении для заданной величины $M(f)$ сетевого потока такого назначения дуговых потоков f_{ij} , которое минимизировало бы сумму затрат [17, 32 – 34].

2.4. Дискретно-детерминированное моделирование систем абстрактными конечными автоматами

2.4.1. Задание конечных автоматов в терминах теории множеств

Абстрактный автомат задается как совокупность шести объектов:

- множества входных сигналов X (*входной алфавит автомата*);
- множества выходных сигналов Y (*выходной алфавит автомата*);
- множества состояний автомата A ;
- элемента $a_0 \in A$, называемого начальным состоянием автомата;
- функций переходов $\varphi(a, x)$ и выходов $\psi(a, x)$, задающих однозначные отображения множества (a, x) , где $a \in A$ и $x \in X$, в множества A и Y .

Абстрактный автомат функционирует в *дискретном времени*, в каждый момент этого времени имеет определенное состояние $a(t)$ из множества A состояний автомата. В каждый момент времени, отличный от начального, автомат способен воспринимать входной сигнал $x(t)$ – произвольную букву входного алфавита X и выдавать соответствующий выходной сигнал $y(t)$ – определенную букву выходного алфавита.

Закон функционирования автомата *первого рода* задается уравнениями вида:

$$a(t) = \varphi[a(t-1), x(t)];$$

$$y(t) = \psi[a(t-1), x(t)];$$

$$t = 1, 2, \dots;$$

Закон функционирования автомата *второго рода*:

$$a(t) = \varphi[a(t-1), x(t)];$$

$$y(t) = \psi[a(t), x(t)];$$

$$t = 1, 2, \dots$$

Также используют *автомат Мура*: частный случай конечных автоматов второго рода, у которого функция выходов $\psi(a, x)$ не зависит от переменной x .

Автомат называется *конечным*, если конечно число его состояний. Автоматы задают табличным способом или направленным графом. В первом случае строят матрицы *переходов* и *выходов*. Строки обеих этих таблиц обозначаются входными сигналами автомата, а столбцы – его состояниями. На пересечении строки и столбца таблицы переходов ставится соответствующее значение функции переходов $\phi(a, x)$, а в таблице выходов – значение $\psi(a, x)$. Для автомата Мура сдвинутая таблица выходов сводится к одной строке, поэтому часто в таблице переходов над каждым состоянием a_i автомата, обозначающим тот или иной столбец таблицы, ставят соответствующий этому состоянию выходной сигнал $\phi(a_i, x) = \psi(a_i)$.

По характеру отсчета дискретного времени автоматы делятся на синхронные и асинхронные. В *синхронных* конечных автоматах моменты времени, в которые автомат воспринимает входные сигналы, определяются принудительно синхронизирующими сигналами. После очередного синхронизирующего сигнала с учетом «считанного» входного сигнала и в соответствии с типом автомата происходит переход в новое состояние и выдача сигнала на выходе, после чего автомат может воспринимать следующее значение входного сигнала.

Асинхронный конечный автомат «считывает» входной сигнал непрерывно и поэтому может несколько раз изменять состояние, выдавая соответствующее число выходных сигналов, пока не перейдет в устойчивое состояние, которое уже не может быть изменено данным входным сигналом.

2.4.2. Задание конечных автоматов в терминах теории графов

При задании автомата с использованием направленного графа вершины графа отождествляются с состояниями автомата, а дуги – с выходными сигналами. Если входной сигнал x_i вызывает переход автомата из состояния a_j в состояние a_k , то на графе автомата этому сигналу соответствует помеченная буквой x_i дуга, соединяющая вершину, соответствующую состоянию a_j , с вершиной, соответствующей состоянию a_k . Для задания функции выхода дуги графа также помечаются соответствующими выходными сигналами. Если обозначенная входным сигналом x_i дуга соединяет вершину a_j с a_k , то в случае автоматов первого рода ей предписывается вы-

ходной сигнал $\psi(a_j, x_i)$, а в случае автоматов второго рода – выходной сигнал $\psi(a_k, x_i)$ (рис. 51).

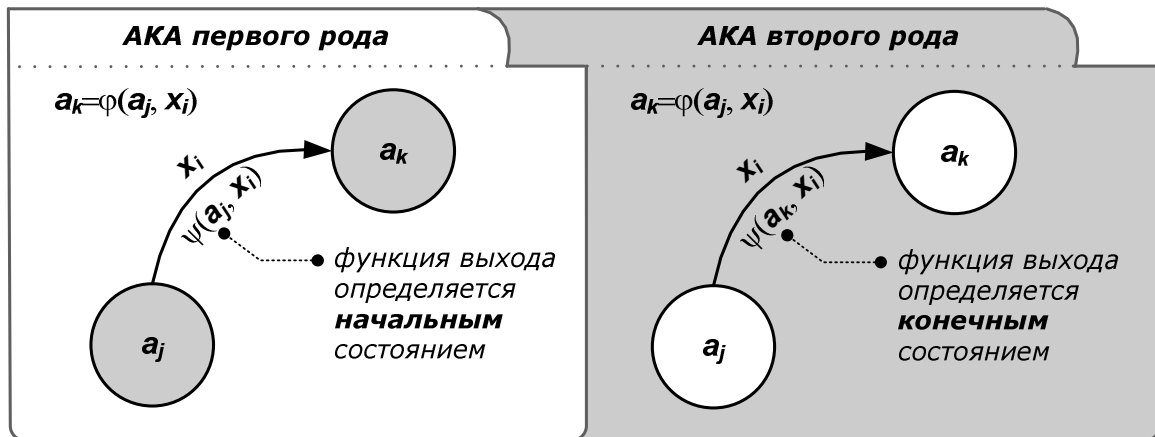


Рис. 51

Пример графа АКА первого рода представлен на рис. 52, а соответствующие ему матрицы переходов и выходов – в табл. 3 и 4.

Пример графа АКА второго рода представлен на рис. 53, а соответствующие ему матрицы переходов и выходов – в табл. 5 и 6.

Таблица 3

Матрица переходов

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
x_1	B	C	D	E	A
x_2	E	C	D	D	E
x_3	A	B	D	B	A

Таблица 4

Матрица выходов

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
x_1	y_1	y_2	y_3	y_4	y_5
x_2	y_2	y_3	y_2	y_4	y_5
x_3	y_5	y_1	y_2	y_5	y_1

Таблица 5

Матрица переходов

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
x_1	b	c	e	a	d
x_2	c	d	b	e	a

Таблица 6

Матрица выходов

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
x_1	y_5	y_1	y_2	y_3	y_3
x_2	y_5	y_4	y_1	y_2	y_4

В случае автомата Мура все дуги, входящие в одну и ту же вершину a_k , должны быть обозначены одним и тем же выходным сигналом. Поэтому принято обозначать выходными сигналами не дуги, а вершины, в которые эти дуги входят, т.е. на графе автомата Мура каждая вершина имеет два обозначения – одно, определяющее состояние автомата, и другое, обозначающее выходной сигнал.

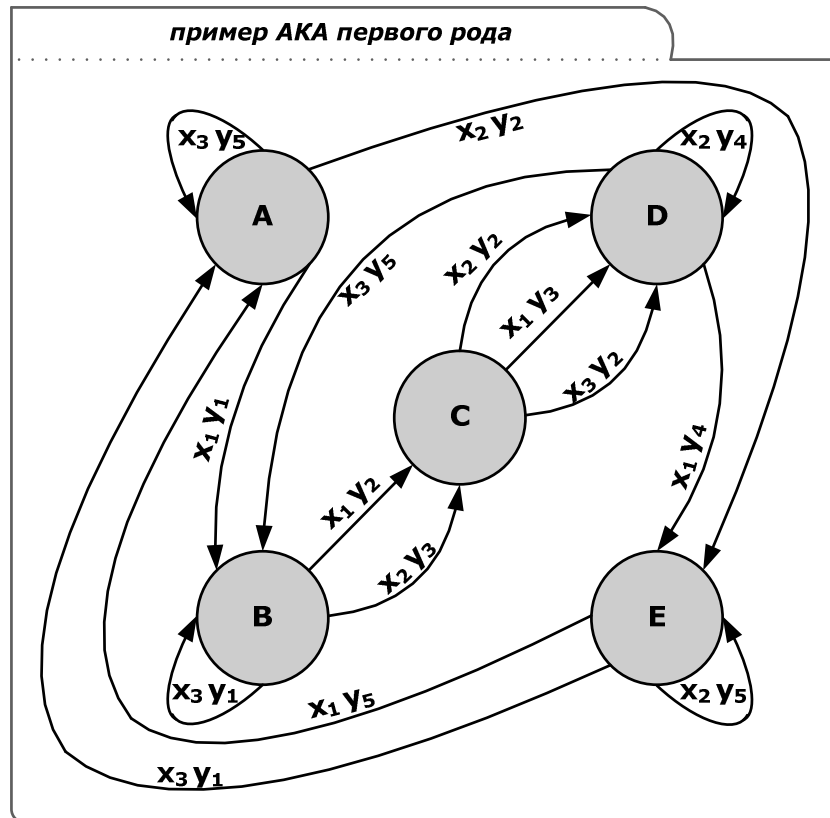


Рис. 52

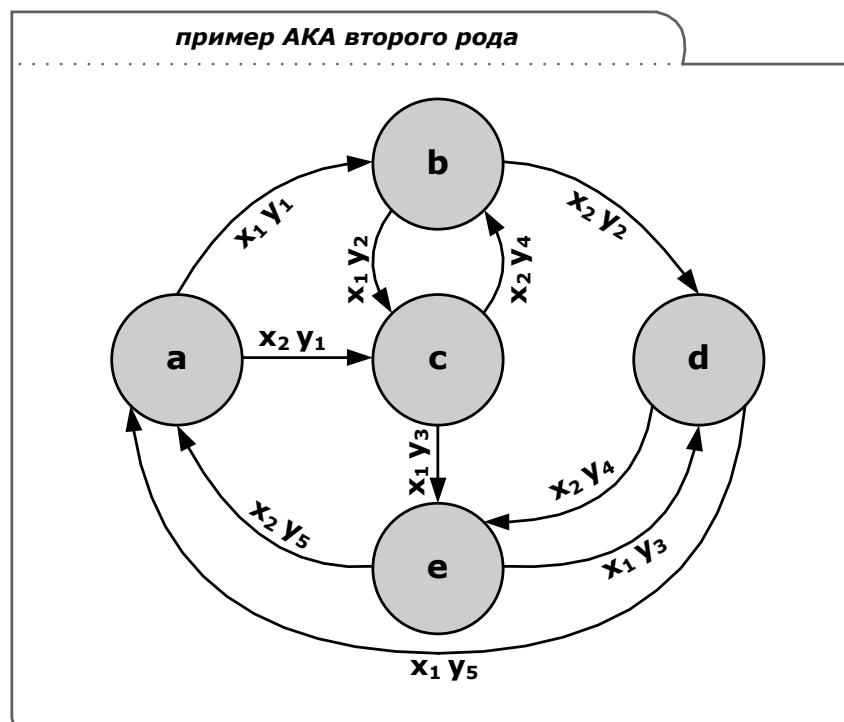


Рис. 53

Пример совмещенной матрицы переходов и выходов АКА Мура и соответствующей ей граф представлены на рис. 54.

Для моделирования элементов вычислительных систем и сетей, проявляющих статистически закономерное случайное поведение, можно использовать *вероятностные автоматы*. Вероятностный автомат определяется, в дополнение с семейству множеств X, Y, A конечного автомата, семейством матриц $[M(y/x)]$. Если $X(x_1...x_l)$ – входной алфавит, $Y(y_1...y_m)$ – выходной алфавит, $A(a_1...a_n)$ – множество состояний, то $M(y/x)$ – семейство $l \times m$ матриц размерностью $n \times n$. Элемент $\mu_{ij}(y_p/x_r)$ матрицы $M(y_p/x_r)$ есть вероятность $\mu_{ij}(y_p/x_r) = P(y_p, a_j/a_i, x_r)$, $i, j = 1...n$, того, что, находясь в состоянии a_i и получив входной сигнал x_r , автомат перейдет в состояние a_j , а выходной сигнал будет y_p . Если $\mu_{ij}(y_p/x_r)$ принимает только значения единицы или нуля, имеем частный случай вероятностного автомата – обычный детерминированный конечный автомат [35, 36].

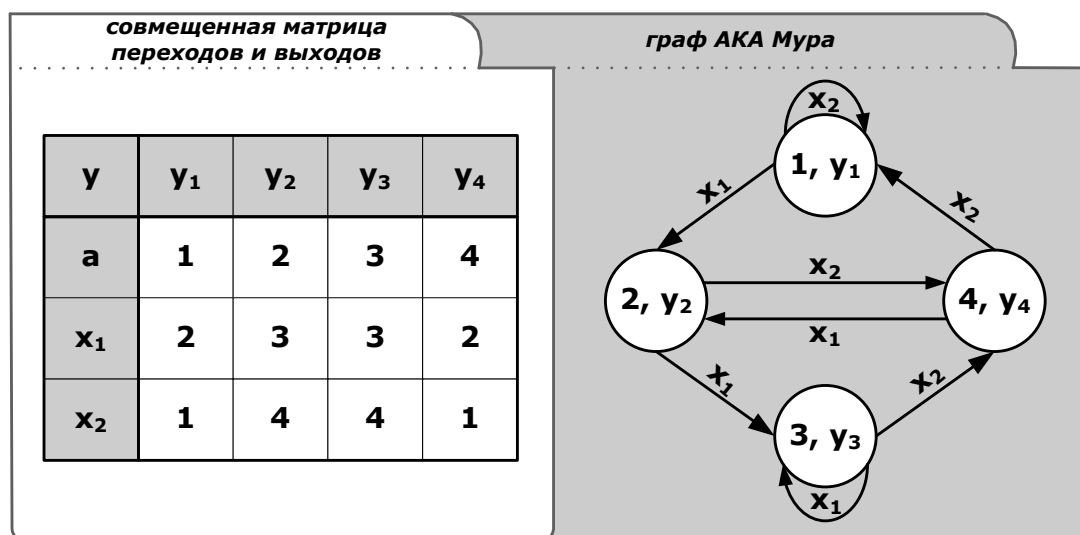


Рис. 54

2.4.3. Моделирование дискретных систем конечными автоматами

Как уже отмечалось (см. раздел 2.1.2), АКА является математической схемой дискретно-детерминированного вида, специально разработанной для формализации решения задач синтеза дискретных устройств, в котором все сигналы квантованы по уровню, а все действия квантованы по времени. В теории алгоритмов АКА решает задачу формализации записи алгоритмов, т.к. представляет собой средство описания элементарного шага алгоритма на максимально простой, но при этом точной и

подлежащей исследования математическими методами логической структуре [41].

Рассмотрим моделирование дискретных систем на примерах формального и функционального моделирования асинхронного RS -триггера, реализованного на элементах И-НЕ, абстрактными конечными автоматами.

Состояния RS -триггера, логическая схема которого и условное обозначение как системного элемента показаны на рис. 3, приведены в табл. 7.

Таблица 7

$\bar{R}$	$\bar{S}$	Q
0	1	0
1	0	1
1	1	не изменяется
0	0	не допускается

Триггер по основному сигнальному выходу имеет два устойчивых состояния ($Q = 0$ и $Q = 1$) и сохраняет любое из них сколь угодно долго после снятия внешнего напряжения, вызвавшего переход триггера из одного состояния в другое. Комбинация значений на сигнальных входах $\bar{R} = 0$, $\bar{S} = 0$ считается запрещенной, т.к. при этой комбинации схема находится в неопределенном состоянии.

Построим формальную модель RS -триггера в виде АКА Мура на основании таблицы его состояний для разрешенных комбинаций сигнальных входов $\bar{R}$ и $\bar{S}$. Зададим элементы описания АКА:

- множество выходных сигналов $Y = \{Q\} = \{1, 0\}$;
- множество состояний автомата Мура $A = Y$;
- множество входных сигналов $X = \{\bar{R} \bar{S}\} = \{11, 01, 10\} = \{x, x_1, x_2\}$.

С учетом введенных обозначений для комбинаций входных сигналов граф автомата Мура и совмещенная матрица переходов и выходов формальной автоматной модели RS -триггера будут иметь вид, показанный на рис. 55.

Формальная автоматная модель RS -триггера, полученная с использованием в качестве исходных данных «разрешенной» части таблицы его состояний, адекватна объекту моделирования с точки зрения процесса формирования статического вектора базисных координат – конечных установившихся значений выходов Q и $\bar{Q}$ в режиме нормальной эксплуатации.

Однако разработка функциональной модели *RS*-триггера, учитывающей специфику его работы, которая определяется перекрестными обратными связями, должна в качестве исходных данных рассматривать логическую схему *RS*-триггера (см. рис. 3), булевы функции элементов, входящих в ее состав, и связи между элементами. Наличие обратных связей обуславливает выбор в качестве математической схемы моделирования асинхронного конечного автомата. Учитывая, что мы строим модель электронной схемы, в которой уровни сигналов представлены двоичной кодировкой, множество выходных сигналов асинхронного АКА Мура полной функциональной модели *RS*-триггера следует определить как

$$Y = \{Q\bar{Q}\} = \{11, 10, 01, 00\}.$$

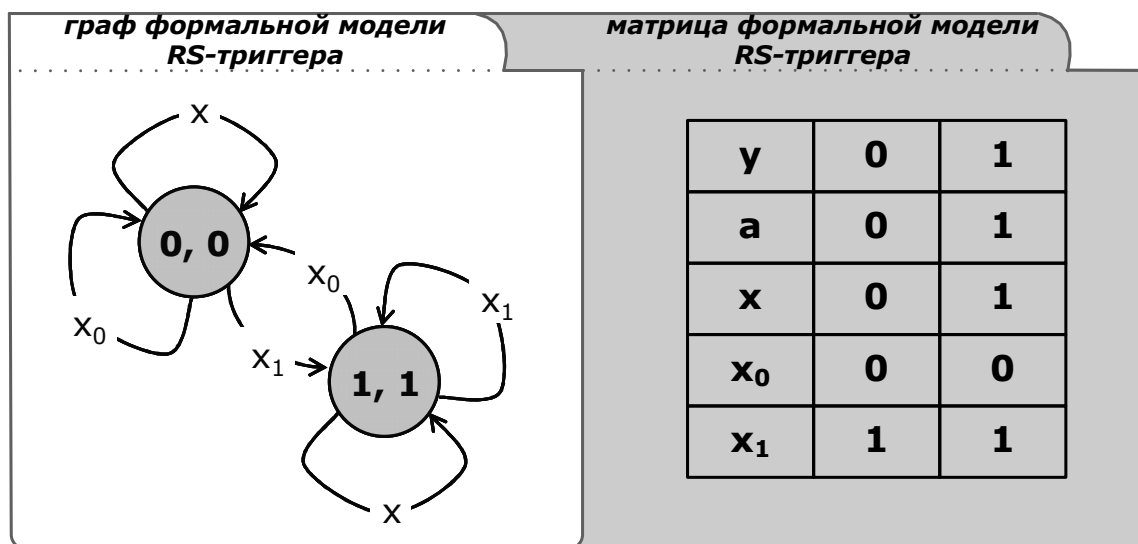


Рис. 55

Аналогично определим прочие элементы описания АКА Мура:

- множество состояний автомата Мура $A = Y$;
- множество входных сигналов $X = \{\bar{R}\bar{S}\} = \{11, 01, 10, 00\}$.

Заметим, что в данном случае мы не исключаем из рассмотрения «запрещенную» комбинацию входных сигналов.

При разработке формальной модели *RS*-триггера функции переходов и выходов были определены формально отображением множества входных сигналов в множество выходных по таблице состояний. В данном случае, при использовании в качестве исходных данных логической схемы триггера, функцию переходов можно записать с использованием булевых функций элементов И-НЕ, на которых реализован триггер:

$$a(t) = \phi[a(t-1), x(t)] = \begin{cases} Q(t) = \overline{S(t-1)} \wedge \overline{Q(t-1)} \\ \overline{Q}(t) = \overline{R(t-1)} \wedge Q(t-1) \end{cases}.$$

Функция выходов, с учетом принятого $A = Y$, имеет следующий вид:

$$y(t) = \psi[a(t)] = a(t).$$

Матрица переходов и выходов графа функциональной модели RS-триггера, элементы которой рассчитаны с использованием заданной функции переходов, имеет вид, представленный в табл. 8, а соответствующий ей граф показан на рис. 56. Ввиду равенства функций переходов и выходов для удобства анализа графа автомата в вершинах приведено одно значение, а входные сигналы, вызывающие эквивалентные переходы, указаны на дугах графа через запятую.

Таблица 8

$A=Y=Q\overline{Q}$ $X=\overline{R}\overline{S}$	01	10	11	00
11	01	10	00	11
01	01	11	01	11
10	11	10	10	11
00	11	11	11	11

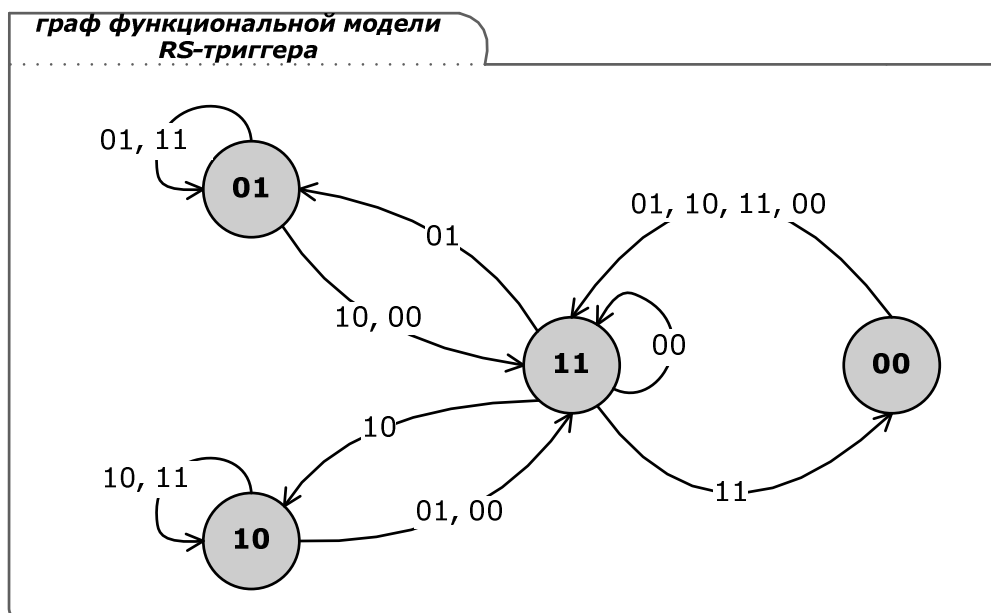


Рис. 56

Функциональная модель триггера, показанная графом асинхронного АКА на рис. 56, наглядно и адекватно отражает процесс смены состояний выходов триггера как в статике, так и в динамике. Видно, что переключение между его устойчивыми состояниями происходит через промежуточное неустойчивое состояние 11. Также очевидны условия вхождения триггера в режим генерации: сначала подачей «запрещенной» входной комбинации 00 его необходимо перевести в неустойчивое состояние, а затем установить и зафиксировать значения входов равным 11.

*ПРИМЕР автоматной модели управления индикацией
даты и времени*



2.5. Непрерывно-дискретное моделирование систем сетями Петри

Понятие конечного автомата является удобной математической схемой моделирования систем, для которых характерно наличие дискретных состояний и дискретный характер работы во времени. Однако для моделирования средств вычислительной техники и вычислительных сетей использование абстрактно-автоматной математической схемы встречает трудности несмотря на то, что данные объекты могут быть представлены как дискретные переходные системы. Это объясняется тем, что конечный автомат является последовательной синхронной алгоритмической схемой, а названные объекты могут выполнять свои функции параллельно и асинхронно, за исключением некоторых заданных точек, называемых точками взаимодействия параллельных процессов. Описание функционирования с использованием конечного автомата в таких случаях теоретически возможно, однако требует рассмотрения большого числа состояний и увеличения сложности модели. Данного недостатка лишены сети Петри.

Сеть Петри это графическое и математическое средство моделирования системной динамики, представляющее собой инструмент описания и исследования асинхронных, распределенных, параллельных, недетерминированных и/или стохастических систем. В качестве графического средства сети Петри используются для наглядного представления моделируемой системы, подобно блок-схемам, структурным схемам и сетевым графикам. Вводимое в этих сетях понятие маркеров позволяет моделировать динамику функционирования систем и параллельные процессы. В качестве математического средства аналитическое представление сети Петри позволяет исследовать динамику изменения и достижимость стационарных состоя-

ний систем. Моделирование в сетях Петри осуществляется на событийном уровне с учетом асинхронности процессов смены состояний путем упорядочивания событий.

В графическом представлении сеть Петри это ориентированный граф, содержащий позиции (вершины), определяющие условия, имеющиеся в системе, и переходы, отображающие связанные с этими условиями действия. В позициях проставляются метки, если соответствующее условие выполнено. Передвижение меток по сети определяет последовательность изменения состояний моделируемого объекта. Позиции изображаются кружками, переходы – планками. Позиции соединяют дугой с переходом, если выполнение заданного условия является необходимым для запуска связанного с данным переходом действия. Переход соединяют дугой с позицией, если связанное с ним действие порождает выполнение условия, представленного данной позицией. Соединение дугой позиции с позицией или перехода с переходом *не допускается*.

Сети Петри функционируют в непрерывном времени. Динамика функционирования определяется правилами срабатывания переходов. Изменение состояния сети связано с механизмом изменения маркировок позиций. В случае *простой временной* сети Петри:

- срабатывает только активизированный переход, т.е. такой, во всех входных позициях которого имеются метки;
- срабатывание перехода наступает через заданный конечный промежуток времени после его активизации, причем если возникает конфликт (одновременная активизация нескольких переходов, имеющих общие входные позиции), то срабатывает равновероятно только один из конфликтных переходов;
- в результате срабатывания перехода число меток в каждой входной позиции уменьшаются на единицу, а число меток во всех выходных позициях увеличивается на единицу.

В рассмотренных выше правилах срабатывания переходов простой временной сети Петри упомянуто состояние конфликта, являющееся фундаментальным в теории сетей Петри: невозможностью одновременного наступления нескольких событий ввиду ограничения общих ресурсов. Пример конфликта приведен на рис. 57. Здесь два активированных перехода t_1 и t_2 находятся в конфликте относительно позиции P_2 . В таком случае может быть сработать только один переход. При его срабатывании метка из общей входной позиции P_2 будет удалена и тем самым будет отменена активизация другого перехода. Как уже отмечалось, в случае конфликта в простых временных сетях Петри срабатывает равновероятно любой из конфликтных переходов.

Равновероятное разрешение конфликтов не всегда соответствует реальным закономерностям функционирования систем: некоторые процессы в системе могут иметь больший, в сравнении с остальными, приоритет, и, следовательно, в условиях недостаточности ресурсов выбор из числа конфликтных процессов будет не равновероятным, а однозначным – в пользу приоритетного процесса.

Для моделирования средств вычислительной техники и процессов обработки информации используются разновидности сетей Петри, различающиеся способами разрешения конфликтов. В *стохастических* сетях Петри вводятся вероятности срабатывания активных переходов. При таком типе разрешения конфликта в примере на рис. 57 сработает либо переход t_1 (с некоторой заданной вероятностью p_1), либо t_2 (с вероятностью $1-p_1$).

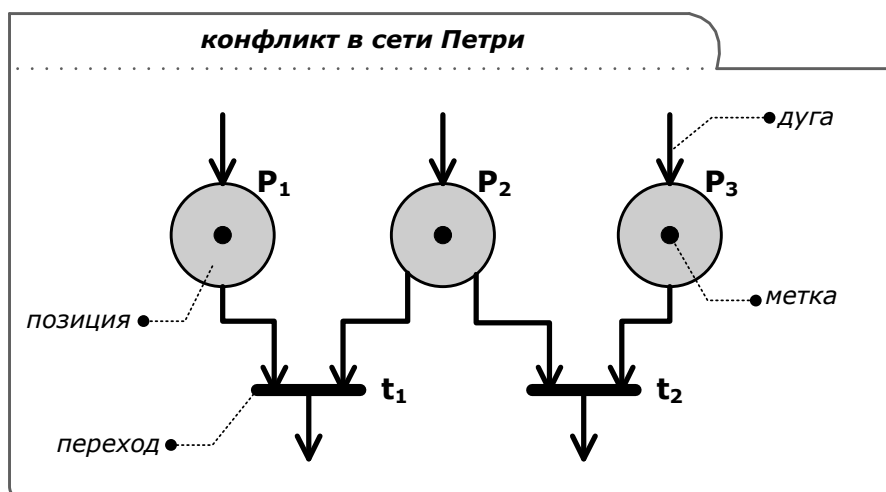


Рис. 57

В *приоритетных сетях* конфликтные ситуации разрешаются введением различных приоритетов для ветвей. Пусть в примере на рис. 57 приоритетным является переход t_1 , тогда конфликт всегда будет разрешаться в пользу этого перехода, а неприоритетный переход t_2 сможет сработать только в том случае, если, при наличии меток в позициях P_2 и P_3 , метки в позиции P_1 не окажется.

Элементарный цикл обслуживания моделируется простой временной сетью Петри, представленной на рис. 58.

С переходами на рис. 58 связаны времена выполнения следующих действий: t_1 – поступление заявки на обслуживание во входную очередь; t_2 – начало обслуживания; t_3 – конец обслуживания; t_4 – выход заявки из цикла обслуживания. Позиции этой сети Петри соответствуют условиям: P_1 – наличие заявки, ожидающей обслуживания, во входной очереди;

P_2 – наличие заявки на обслуживании в процессоре; P_3 – процессор свободен; P_4 – наличие обслуженной заявки в выходной очереди. Маркировка сети Петри, показанной на рис. 58, соответствует начальному состоянию системы обслуживания: заявок, ожидающих обслуживание, во входной очереди нет, и процессор свободен (позиция P_3 содержит метку).

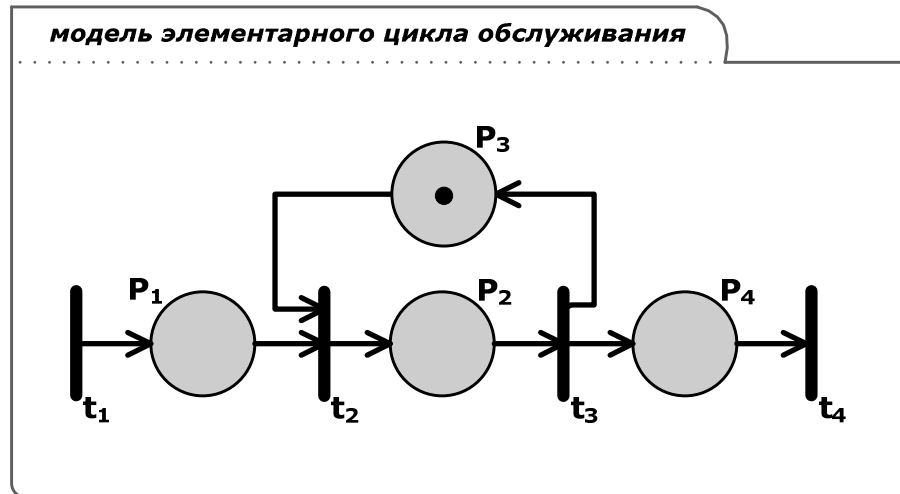


Рис. 58

Последовательность смены маркировок сети Петри, моделирующей элементарный цикл обслуживания, показана на рис. 59. Маркировка первого шага соответствует следующему состоянию: заявка ожидает обслуживания и процессор свободен: позиции P_1 и P_3 содержат метки, и, следовательно, переход t_2 активизирован. Моделирование дальнейших событий в пошаговом режиме показано на рис. 59. После срабатывания перехода t_2 метки из его входных позиций P_1 и P_3 будут изъяты, а в выходной позиции P_2 метка появится, что соответствует наличию заявки на обслуживание в процессоре. Метка в позиции P_2 активизирует переход t_3 , который сработает по истечению времени обслуживания. В результате срабатывания перехода t_3 метка изымается из позиции P_2 , а в позициях P_4 и P_3 метки появляются. Срабатывание перехода t_4 , соответствующее удалению обслуженной заявки из цикла, изымет метку из позиции P_4 , и система вернется к начальной маркировке (см. рис. 58).

В общем случае в позиции может быть более одной метки. Тогда, для срабатывания перехода вида n/m требуется наличие во входных позициях суммарного количества меток не менее n . При срабатывании перехода из всех входных позиций по числу исходящих дуг удаляются в сумме n меток, а во всех выходных позициях по числу выходных дуг появляются m меток (рис. 60).

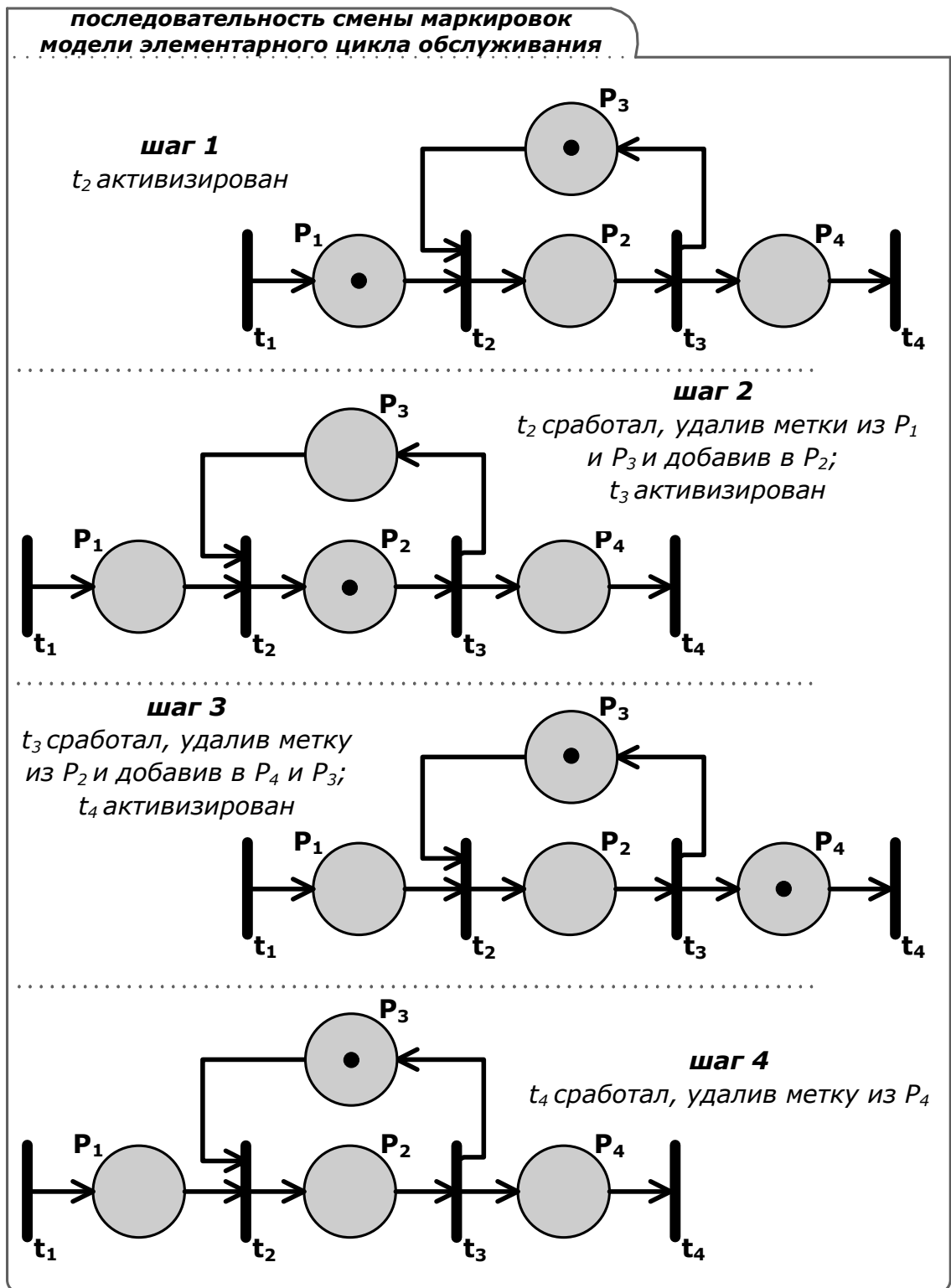


Рис. 59

Для удобства в графическом представлении сетей Петри и в программном обеспечении моделирования, как правило, при задании переходов типа n/t вместо множественных дуг используют одну, задавая ее вес.

Также из соображений удобства количество меток в позиции более 2-х указывают числом вместо множества меток. В таком представлении пример, приведенный на рис. 60, будет иметь вид, показанный на рис. 61.

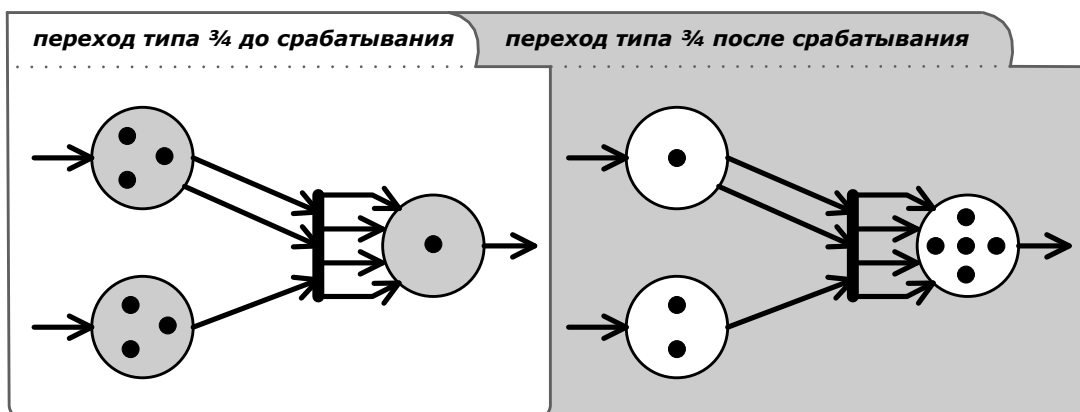


Рис. 60

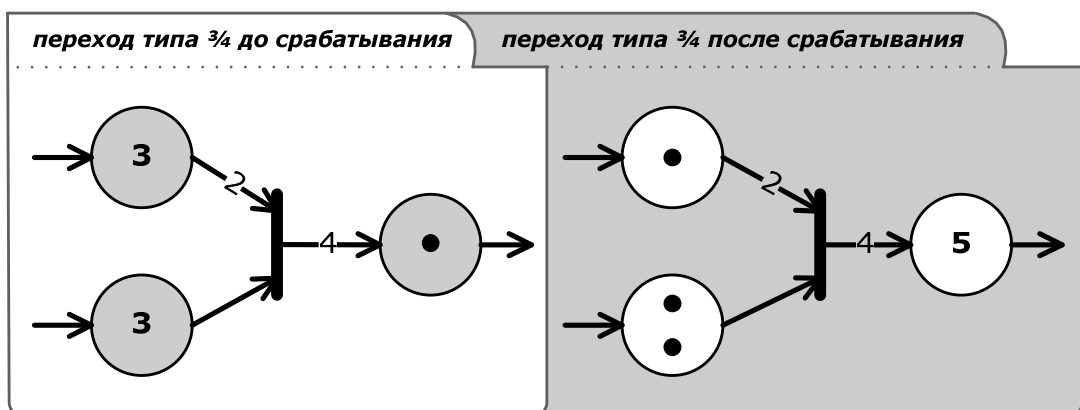


Рис. 61

Переходы типа n/m используются для моделирования процессов, сопровождаемых модификацией ресурсов: например, для моделирования процессов сборки в технологической системе, процессов пакетирования заявок в вычислительной сети (в этом случае $n > m$) или процесса разделения материальных потоков, процесса создания копий заявок ($n < m$).

Особой разновидностью сетей Петри являются *ингибиторные* сети, которые в дополнение к обычным дугам графа сети содержат запрещающие, так называемые ингибиторные дуги (обозначаются линиями с кружками на конце). Такая дуга запрещает срабатывание активизированного перехода до тех пор, пока в ее входной позиции имеются метки числом больше либо рав-

но весу дуги. Пример реализации простейшего цикла обслуживания с использованием ингибиторной сети Петри представлен на рис. 62. Здесь переход t_2 при наличии метки в позиции P_2 будет «заперт» не смотря на наличии метки в позиции P_1 до тех пор, пока метка не покинет P_2 через переход t_3 , что эквивалентно завершению очередного обслуживания [35, 37 – 40].

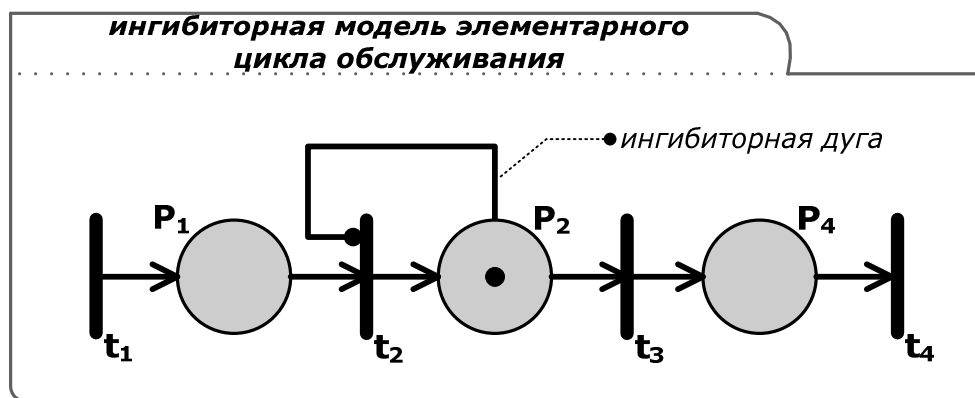


Рис. 62

ПРИМЕРЫ моделирования сетями Петри различных процессов и систем



Рассмотрим пример моделирования сетью Петри процессов возникновения отказов и устранения неисправностей в технологической системе с запасным блоком и дополнительным резервом. Пусть в технологической системе эксплуатируются три рабочих блока. С некоторой периодичностью в любом из рабочих блоков может наступить отказ. Для замены отказавшего блока изначально имеется один запасной. При наступлении отказа неисправный блок заменяется запасным, после чего неисправный блок подвергают восстановлению (ремонт) и он становится запасным. В системе кроме запасного блока имеется еще один, резервный. Его использование разрешено только в случае отказа всех трех основных рабочих блоков. Если резервный блок использован – он должен быть восстановлен в первую очередь. Все временные характеристики системы (время отказа, время замены, время восстановления) являются случайными величинами.

При функционировании данной системы могут иметь место следующие ситуации:

- все три рабочих блока исправны, запасной и резервный блоки готовы к использованию;

- один из рабочих блоков отказал и запасной готов к использованию (в этом случае будет произведена замена, и отказавший блок будет отправлен на восстановление для последующей передачи в запас);
- если запасной блок не готов к использованию, т.е. находится на восстановлении, возможны ситуации последовательного наступления отказов одного и второго рабочих блоков без замены;
- если запасной блок не готов к использованию, т.е. находится на восстановлении, а отказывают последовательно все три рабочих блока, то для замены будет использован резервный блок;
- если резервный блок уже использован, то после завершения восстановления очередного отказавшего блока он будет передан не в запас, а в резерв, т.к. восстановление резерва выполняется в первую очередь;
- если резервный блок уже восстановлен, то очередной отказавший блок будет отправлен на восстановление и, далее, в запас.

Заданную систему моделирует ингибиторная приоритетная сеть Петри, представленная на рис. 63.

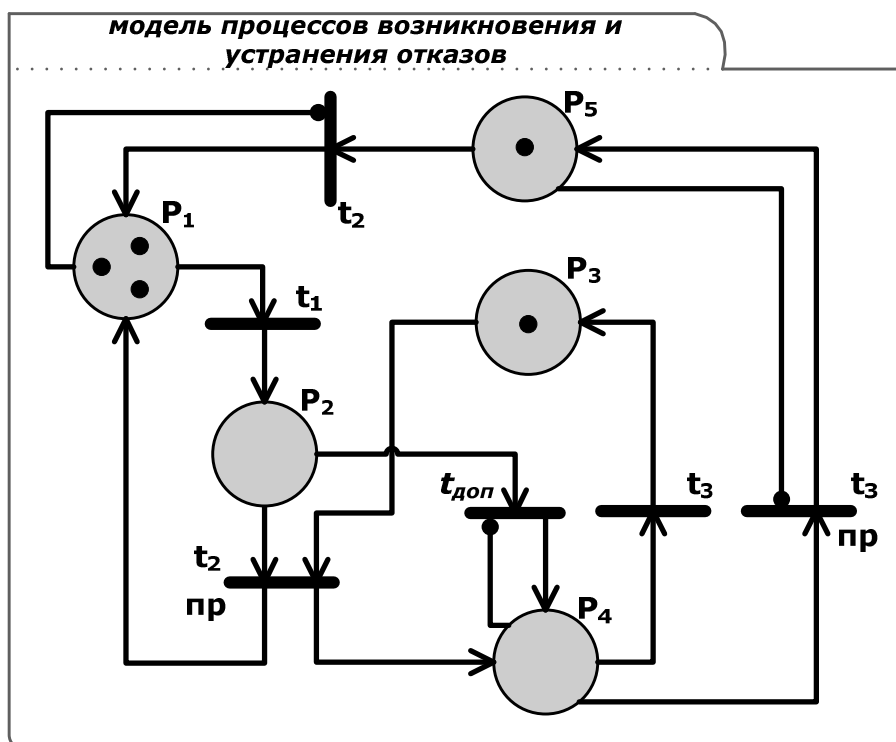


Рис. 63

В табл. 9 представлено описание всех позиций (вершин) сети с точки зрения условий, которые с ними связаны, т.е. выполняются при наличии в вершине метки-маркера.

Таблица 9

Вершина	Описание
P_1	Наличие в системе исправных рабочих блоков (количество меток соответствует количеству блоков)
P_2	Наличие в системе отказавших рабочих блоков (количество меток соответствует количеству блоков)
P_3	Наличие в системе исправного, готового к использованию для замены запасного блока
P_4	Наличие в системе неисправного, нуждающегося в восстановлении блока
P_5	Наличие в системе исправного, готового к использованию для замены резервного блока

Начальная маркировка сети, представленная на рис. 63, соответствует наличию в системе трех исправных рабочих блоков; исправного, готового к использованию для замены запасного блока и исправного, готового к использованию для замены резервного блока. Описание всех переходов с точки зрения связанных с ними событий представлено в табл. 10.

Таблица 10

Переход	Описание
t_1	Наступление отказа (время случайно)
t_2	Выполнение замены неисправного блока на запасной или резервный (время случайно)
t_3	Выполнение восстановления неисправного блока (время случайно). Два идентичных перехода реализованы для моделирования восстановления запасного и резервного блоков. Так как при необходимости восстановление резервного блока должно осуществляться в первую очередь, соответствующий переход имеет больший приоритет (помечен на рис. 54 «пр.») и в случае конфликта работает именно он
t_{don}	Отправление на восстановление одного из отказавших блоков без замены. Данный переход необходим, т.к. в случае первоочередного восстановления резерва метка в вершине P_3 отсутствует и переход t_2 сработать не может. В нормальном режиме работы, когда в системе имеется запасной блок (метка в вершине P_3) для разрешения конфликта переходу t_2 присвоен больший приоритет (помечен на рис. 54 «пр.»)

ПРИМЕРЫ моделирования системной динамики и систем массового обслуживания сетями Петри – см. лабораторные работы № 5 и № 6



ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Лабораторная работа № 1

Агрегирование и эквивалентные преобразования непрерывно-детерминированных моделей систем

Цель: агрегирование (композиция) замкнутой системы по заданной структуре и известным моделям непараметрической динамики элементов, представленных в виде передаточных функций; эквивалентное преобразование разомкнутой системы к виду последовательного соединения элементов; решение задачи динамики для заданного вида внешнего целенаправленного воздействия.

Программное обеспечение: VisSim, MathCAD.

Примеры выполнения работы: файлы *Пример моделирования входного воздействия.vsm*, *Пример агрегирования.vsm*.

Теория

Процесс последовательного деления системы на образующие ее элементы называется *декомпозицией*. Обратный процесс объединения отдельных элементов в систему – *агрегированием* (композицией).

Структура системы может быть охарактеризована по имеющимся в ней типам связей (см. рис. 7 в разделе 1.2 *Лекционного курса*).

Последовательным называется такое соединение элементов в системе, когда реакция первого элемента на внешнее воздействие является входным воздействием для второго элемента и т.д.

При *параллельном* соединении внешнее воздействие является входным для всех элементов структуры, а реакции элементов суммируются.

Обратная связь изменяет входное воздействие по результатам функционирования элемента, то есть внешнее воздействие модифицируется с учетом текущего состояния элемента системы.

Система в процессе функционирования взаимодействует с внешней средой. Воздействия называются *целенаправленными* (целесообразными), если они предусмотрены алгоритмом функционирования системы и изменяющими состояние системы заданным образом.

Поведение (динамика) системы – изменение состояния системы в процессе функционирования из-за внешних воздействий – исследуется с использованием моделей динамики различного вида. Одной из разновид-

ностей моделей непрерывно-детерминированного вида являются модели уровня *непараметрической динамики*. Эти модели имеют вид дифференциальных уравнений, описывающих закон функционирования системы без интерпретации коэффициентов уравнения через внутренние параметры системы (коэффициенты заданы как константы и физический смысл их неизвестен или на данном уровне не рассматривается).

Уравнение непараметрической динамики линейной системы или элемента имеет вид:

$$\begin{aligned} a_n \frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_0 y(t) = \\ = b_m \frac{d^m x(t)}{dt^m} + b_{m-1} \frac{d^{m-1} x(t)}{dt^{m-1}} + \dots + b_0 x(t), \end{aligned} \quad (1)$$

где $x(t)$ – входное воздействие; $y(t)$ – выходная величина (реакция); a_i и b_j – постоянные коэффициенты.

Для решения практических задач системного анализа чаще используют модели непараметрической динамики в виде передаточных функций. *Передаточной функцией* $W(s)$ называется отношение операторных изображений по Лапласу выходной и входной величин при нулевых начальных условиях. Переход от оригинала функций к изображениям по Лапласу для уравнения (1) может быть осуществлен формальной заменой дифференцирования на символическое умножение на оператор $s = d/dt$ и выражением отношения $y(s)/x(s)$:

$$a_n s^n y(s) + a_{n-1} s^{n-1} y(s) + \dots + a_0 y(s) = b_m s^m x(s) + b_{m-1} s^{m-1} x(s) + \dots + b_0 x(s) ;$$

$$y(s)(a_n s^n + a_{n-1} s^{n-1} + \dots + a_0) = x(s)(b_m s^m + b_{m-1} s^{m-1} + \dots + b_0) ;$$

$$W(p) = \frac{y(s)}{x(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_0} = \frac{\sum_{i=0}^m b_i s^i}{\sum_{j=0}^n a_j s^j},$$

где $s = d/dt$; $y(s)$ и $x(s)$ – изображения $y(t)$ и $x(t)$ соответственно.

При известных передаточных функциях элементов передаточные функции соединений элементов в составе системы определяются по формулам, приведенным в табл. 11.

Математические преобразования передаточных функций систем приводят к эквивалентным преобразованиям структуры системы.

Таблица 11

Тип и схема соединения	Формула расчета передаточной функции
последовательное	$W(s) = \prod_{i=1}^n W_i(s)$
параллельное	$W(s) = \sum_{i=1}^n W_i(s)$
с обратной связью	$W(s) = \frac{W_{\text{элемента}}(s)}{1 \pm W_{\text{элемента}}(s)W_{OC}(s)}$ (в формуле знак «плюс» соответствует отрицательной, а знак «минус» – положительной обратной связи)

Ход выполнения работы

Дано: структурная схема замкнутой системы; модели непараметрической динамики элементов системы в виде передаточных функций; вид внешнего целенаправленного воздействия.

1. Создать модель источника внешнего целенаправленного воздействия заданного вида в пакете VisSim.

2. Осуществить решение задачи агрегирования. Реализовать модель исходной системы в пакете VisSim. Получить решение задачи динамики для заданного вида внешнего целенаправленного воздействия. Выполнить анализ решения.

3. Осуществить решение задачи эквивалентных преобразований разомкнутой системы к последовательному соединению элементов. Подтвердить эквивалентность полученной системы путем решения задачи динамики для заданного типа внешнего целенаправленного воздействия.

4. Результаты выполнения работы по пунктам 1 – 3 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

Практические рекомендации

по выполнению работы и использованию программного обеспечения

Основные принципы работы с пакетом VisSim

Пакет VisSim разработан компанией Visual Solutions (USA). VisSim (Visual Simulator) – это визуальная среда моделирования технических объектов, систем и их элементов. Пакет имеет развитый графический интерфейс для создания моделей систем из виртуальных элементов; исследования и оптимизации моделей систем.

Запуск пакета VisSim осуществляется файлом Vissim32.exe или щелчком по ярлыку на рабочем столе. Рабочее окно программы включает в себя следующие основные элементы:

- заголовок окна, в котором указано название пакета VisSim и название редактируемого файла модели;
- строка команд основного меню;
- графическая панель инструментов с кнопками, дублирующими наиболее часто используемые команды основного меню, например, некоторые палитры типовых блоков.

Модель системы VisSim реализуется набором функциональных блоков, соединенных между собой линиями связи и отображаемых в рабочем окне редактора. Палитры функциональных блоков находятся в меню *Blocks*. Каждый блок моделирует динамические свойства некоторого функционального элемента систем. Выбор и размещение блоков в рабочем окне осуществляется левой кнопкой мыши, вызов окна редактирования свойств блока – правой.

Каждый блок имеет вид геометрического значка, внутри которого схематично изображено его функциональное назначение. С левой стороны блока находятся его входы, с правой – выходы. Количество входов и выходов у каждого блока различно и, как правило, неизменно.

Выделение блока или группы блоков осуществляется растягиванием рамки выделения при нажатой левой кнопке мыши. К выделенному блоку применимо удаление, копирование в буфер (*Ctrl+C* или пункт меню *Edit – Copy*) и вставка из буфера (*Ctrl+V* или пункт меню *Edit – Paste*).

Выделенную группу блоков можно объединить в единый блок – этот прием применяется для визуального упрощения структурно сложных моделей. Объединение выполняется командой меню *Edit – Create Compound Block*. В появившемся окне вводится имя объединенного блока. После нажатия ОК выделенная группа блоков заменяется одним новым блоком. Просмотр первоначальной структуры объединенных блоков вызывается щелчком правой кнопки мыши или двойным щелчком левой. Количество вложений блоков не ограничено. Обратная операция – разъединение объединенного блока на составляющие – осуществляется командой меню *Edit – Dissolve Compound Block*.

Соединения блоков в схеме осуществляется следующим образом:

- подведите указатель мыши к выводу блока, который необходимо соединить с другим выводом (вид указателя должен смениться на стрелку вида ↑);

- удерживая кнопку мыши, перетащите указатель с проводником к выводу блока назначения и в области подключаемого вывода отпустите кнопку мыши.

В момент выполнения операции VisSim генерирует мерцающую линию, которая является проводником. VisSim рисует проводники только под прямыми углами, поэтому проводник не повторяет путь курсора. Если взаимные пересечения проводников загромождают схему, то следует использовать блок *фиксатора проводника*. Блок выбирается из графической панели инструментов кнопкой , а при установке в поле редактора VisSim имеет вид .

Для удаления уже созданного соединения выполните обратные действия: подведите указатель мыши к конечному выводу соединения и при нажатой кнопке мыши отсоедините вывод.

Для разворота (зеркального отражения) блока в поле редактора VisSim используется команда меню *Edit – Flip Horizontal*.

Пример 1

Моделирование источников внешних воздействий кусочно-линейного вида

Для решения задачи формирования внешнего целенаправленного воздействия заданного вида в VisSim используется набор функциональных блоков из различных палитр (табл. 12).

Для визуализации графиков входных воздействий и выходных реакций системы (результатов решения задачи динамики) в VisSim используется *виртуальный запоминающий осциллограф*, добавление которого осуществляется кнопкой графической панели инструментов . В окне редактирования свойств осциллографа можно активировать вывод координатной сетки, подписи осей и т. п. Для корректного отображения заголовка на русском языке в меню *View – Fonts* следует выбрать шрифт с кириллическим набором символов. Двойной щелчок по заголовку полученного графика разворачивает его в полноэкранный режим. Двойной щелчок по полю построения эквивалентен нажатию правой кнопки мыши и вызывает окно редактирования свойств. Масштабирование окна осциллографа выполняется растягиванием рамки. Для перемещения окна указатель мыши следует переместить к центру окна (вид указателя изменится на ) и при нажатой левой кнопки мыши «перетащить» окно в нужное место.

Источники часто используемых типовых воздействий (константа, линейная функция, ступенчатая функция, импульсное воздействие и синусоида) вынесены на графическую панель инструментов: .

Таблица 12

Палитра	Блок	Описание
Arithmetic (арифметические функции)	* (multiply)	умножение двух сигналов
	-X (negate)	инверсия знака
	/ (divide)	деление
	abs	модуль
	gain	блок усиления сигнала (коэффициент усиления)
	summingJunction	сумматор (знак входа изменяется правой кнопкой мыши при нажатой клавише Ctrl)
Nonlinear (нелинейные элементы систем)	limit	ограничитель сигнала
	sampleHold	фиксатор мгновенного значения сигнала (фиксирует мгновенное значение сигнала на входе x в момент подачи сигнала на вход b и далее воспроизводит сигнал x на выходе)
Signal Producer (генераторы сигналов)	const	константа
	parabola	параболическое воздействие
	pulseTrain	импульсное воздействие
	ramp	линейное воздействие
	sinusoid	синусоидальный сигнал
Time Delay (задержки времени)	timeDelay	запаздывание (задерживает сигнал на входе x на время, заданное постоянным сигналом на входе t)

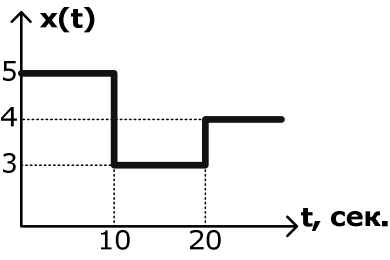
Настройка режима моделирования осуществляется командой меню *Simulate – Simulation Properties*. В окне настроек можно задать начальное время моделирования (*Start*), шаг приращения времени (*Step Size*) и конечное время моделирования (*End*). ВНИМАНИЕ! Для обеспечения достаточной точности моделирования величина шага приращения времени должна быть малой, например, при моделировании процесса функционирования системы в течение 25 с шаг следует выбрать равным 0,01 с или менее.

Запуск моделирования в режиме автоматического прогона осуществляется любым из следующих альтернативных способов:

- нажатием функциональной клавиши F5;
- командой меню *Simulate – Go*;
- нажатием кнопки в графической панели инструментов .

Рассмотрим примеры реализации моделей источников внешнего воздействия кусочно-линейного вида. Пример разложения сложного ступенчатого воздействия на сумму типовых сигналов приведен в табл. 13.

Таблица 13

Вид сигнала	Составляющие
	1. Постоянный сигнал ($\text{const} = 5$).
	2. Ступенчатая функция с параметрами: время задержки $t_0 = 10$ с; амплитуда $A = -2$.
	3. Ступенчатая функция с параметрами: время задержки $t_0 = 20$ с; амплитуда $A = 1$.

Реализация модели источника заданного ступенчатого сигнала с открытым окном настройки параметров третьей составляющей сигнала и выводом результата моделирования в VisSim имеет следующий вид (рис. 64).

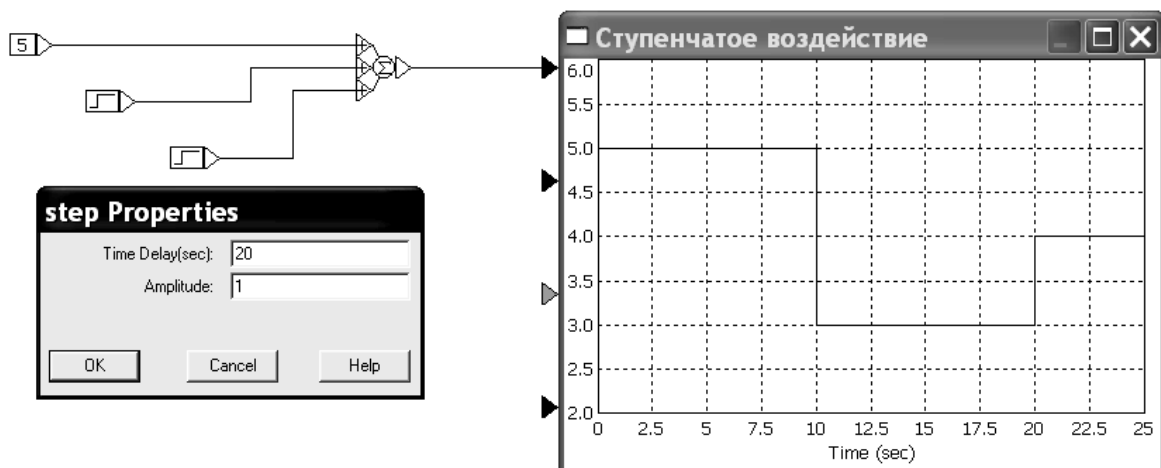


Рис. 64

Добавление в схему сумматора выполняется с использованием кнопки графической панели инструментов  или блоком *Blocks – Arithmetic – summingJunction*. У базовой модели сумматора два входа. Добавление/удаление дополнительных входов для настраиваемых блоков VisSim осуществляется кнопками панели инструментов  .

Для удобства дальнейшего использования выполним объединение схемы модели источника в единый блок. Выделим схему модели источника, применим команду *Edit – Create Compound Block* и в окне редактирования свойств объединенного блока введем его название и нажмем ОК (рис. 65).

Пример реализации линейно убывающего воздействия с ограничением с использованием типовых сигналов и функциональных блоков VisSim приведен в табл. 14.

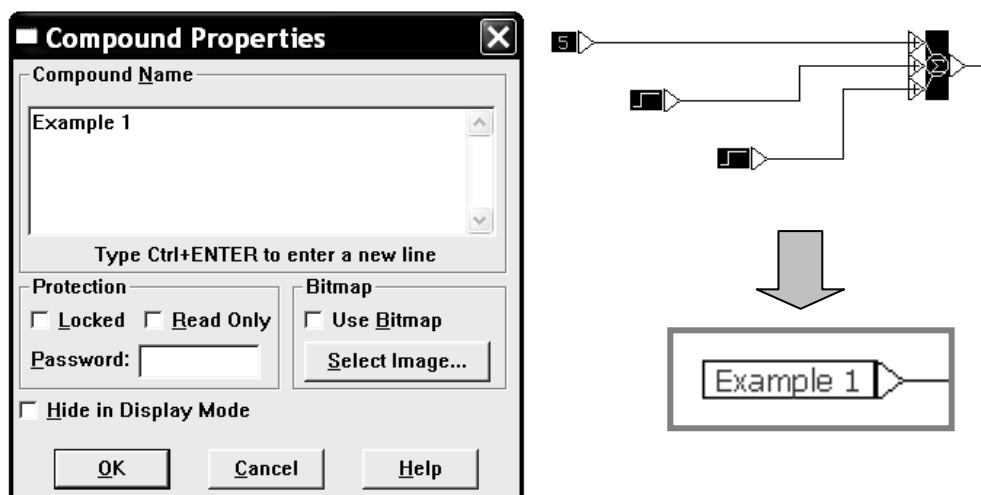


Рис. 65

Таблица 14

Вид сигнала	Составляющие и блоки
Graph showing signal $x(t)$ versus time t (sec). The signal is constant at 10 for $t < 10$ and then decreases linearly to 0 at $t = 20$.	1. Постоянный сигнал ($\text{const} = 10$).
	2. Линейно убывающая функция $x(t) = k \cdot (t - t_0)$ с параметрами: время задержки $t_0 = 10$ с; коэффициент $k = -1$.
	3. Ограничитель сигнала с нижним пределом ограничения, равным 0.

Реализация модели источника заданного сигнала с выводом результата моделирования в VisSim имеет следующий вид (рис. 66):

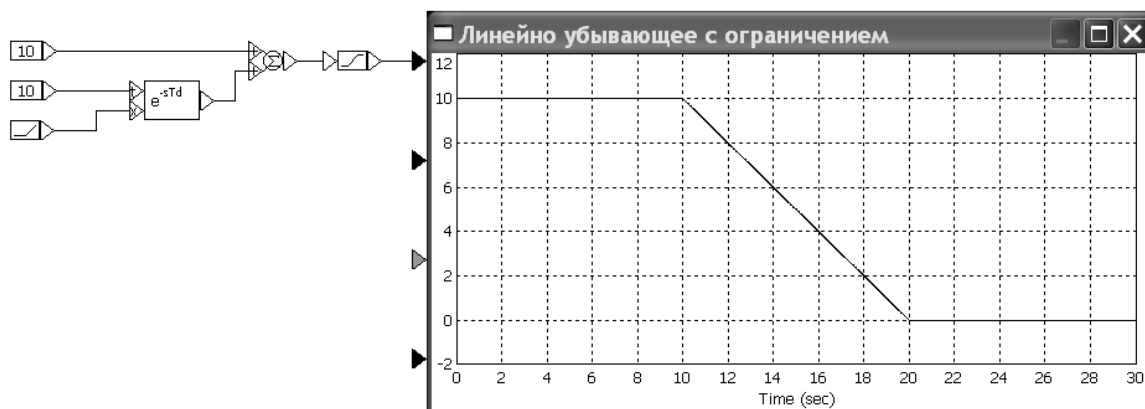


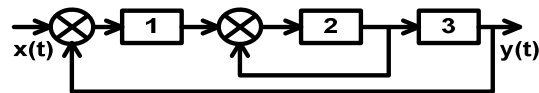
Рис. 66

Для задержки подачи на второй вход сумматора линейной функции использован блок *timeDelay*, который задерживает сигнал на входе x на время, заданное постоянным сигналом на входе t . Для удобства дальнейшего использования схему модели источника следует объединить в единый блок (в дальнейших примерах используется под именем *Example 2*).

Пример 2

Решение задач агрегирования линейных систем

Пусть задана структура системы № 1, состоящей из элементов, для которых известны модели в виде передаточных функций и задан вид внешнего целенаправленного воздействия (ступенчатая функция с параметрами: время задержки $t_0 = 0,5$ с и амплитудой $A = 1$).



$$W_1(s) = 1,8 \frac{s}{0,5s + 1}; \quad W_2(s) = 2 \frac{1}{0,1s + 1}; \quad W_3(s) = 2 \frac{1}{s(0,4s + 1)}.$$

Реализуем модель исходной системы в пакете VisSim. Вставка базовой модели элемента, заданного передаточной функцией, в рабочее поле редактора осуществляется с помощью блока *transferFunction* (*Blocks – Linear System – transferFunction*) (рис. 67, 68).

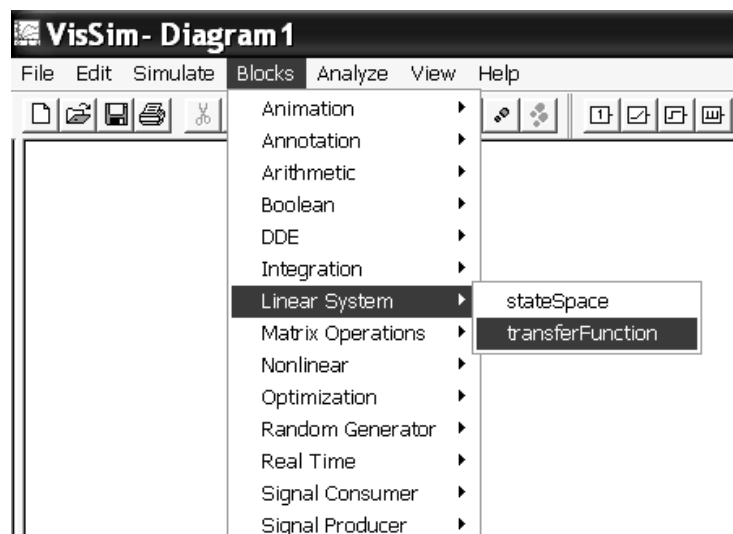


Рис. 67

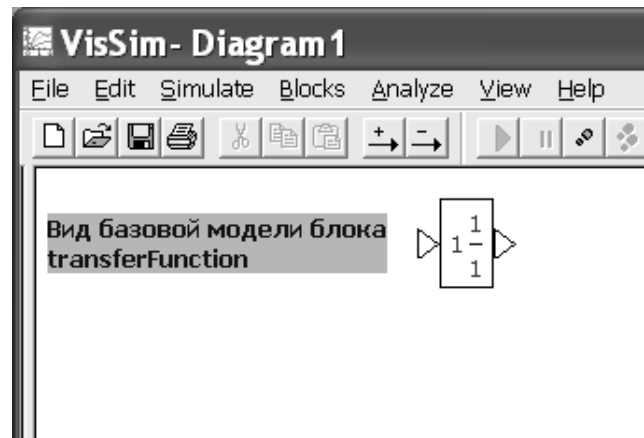


Рис. 68

В окне редактирования свойств блока вводятся параметры передаточной функции: общий коэффициент преобразования (усиления или ослабления) и коэффициенты полиномов числителя и знаменателя.

Модели в виде передаточных функций в VisSim имеют следующее ограничение: степень полинома числителя не может быть больше степени полинома знаменателя.

Примеры задания различных передаточных функций элементов представлены на рис. 69, 70.

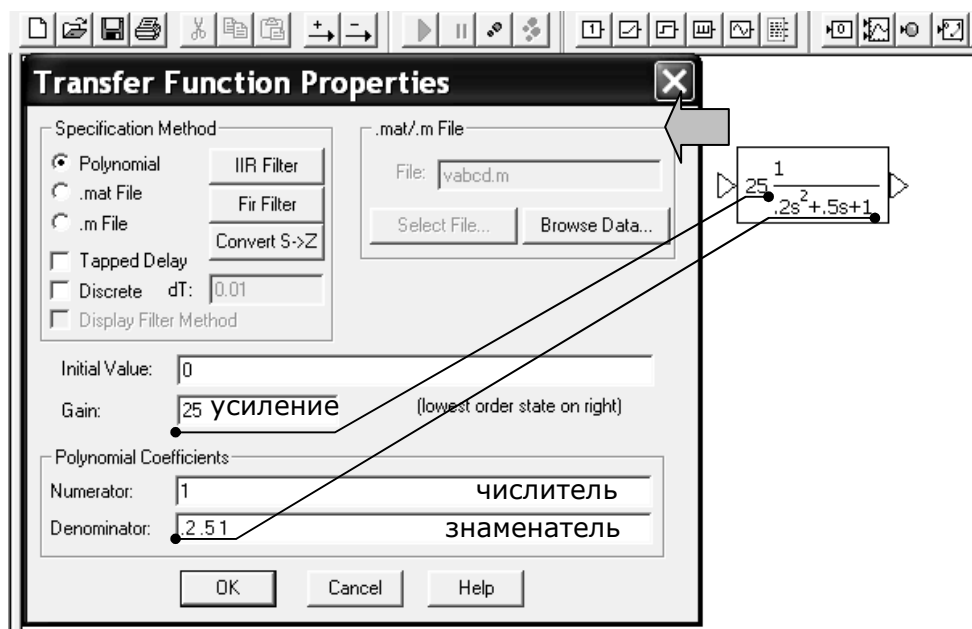


Рис. 69

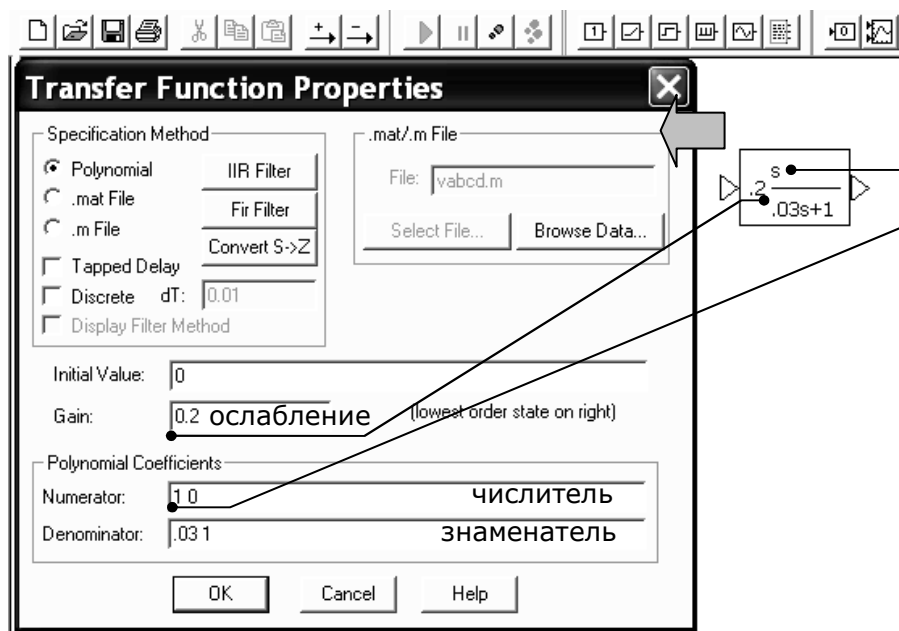


Рис. 70

Сомножитель передаточной функции вида $1/s$ называется *интегратором* и задается блоком из пункта меню *Blocks – Integration – Integrator*. Общий коэффициент преобразования также можно задать блоком *Blocks – Arithmetic – gain*.

Элемент сравнения, производящий определение рассогласования (разницы) входного воздействия и величины в цепи обратной связи, реализуется двухвходовым сумматором, у которого знак одного из входов изменен на минус. Изменение знака входа осуществляется щелчком правой кнопки мыши при нажатой клавише *Ctrl*.

Решение задачи агрегирования в VisSim для системы из рассматриваемого примера имеет вид, приведенный на рис. 71. Результат решения задачи динамики для заданного типа внешнего воздействия, полученный в виде графика реакции системы (изменения выходного параметра) показывает, что ступенчатое входное воздействие с $A = 1$ на выходе системы воспроизводится с ослаблением ($A_{\text{блх}} \approx 0,7$), причем время достижения системой стационарного состояния составляет приблизительно $3,5 - 0,5 = 2,5$ с, а переходной динамический процесс имеет слабоколебательную форму.

Выполним решение задачи динамики для сложного ступенчатого воздействия, модель источника которого была разработана в примере 1. С учетом того, что входной сигнал изменяется в интервале от 0 до 25 с, увеличим время моделирования до 30 с. Результат решения задачи динамики для данного типа внешнего воздействия (рис. 72) показывает, что сложное

ступенчатое входное воздействие на выходе системы воспроизводится также, как и единичная ступенька: с ослаблением амплитуды на 30 % в установившемся состоянии. Время достижения системой стационарного состояния после каждого изменения амплитуды входного сигнала составляет приблизительно 2,5 с, а переходной динамический процесс имеет слабоколебательную форму.

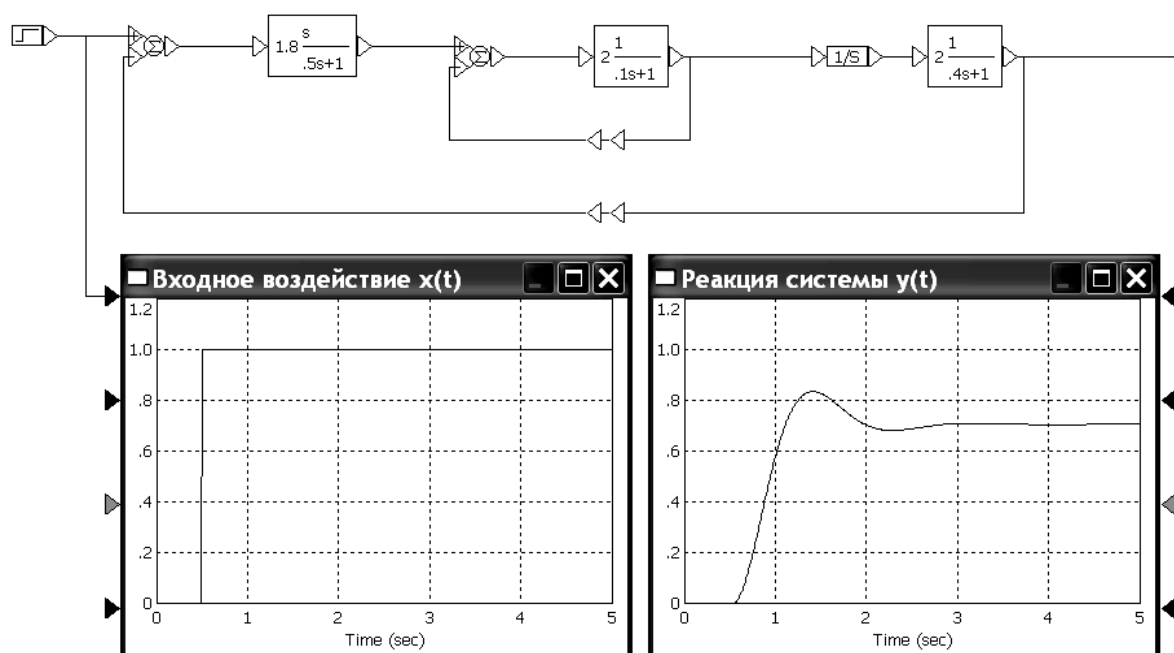


Рис. 71

Модель источника
из Примера 1

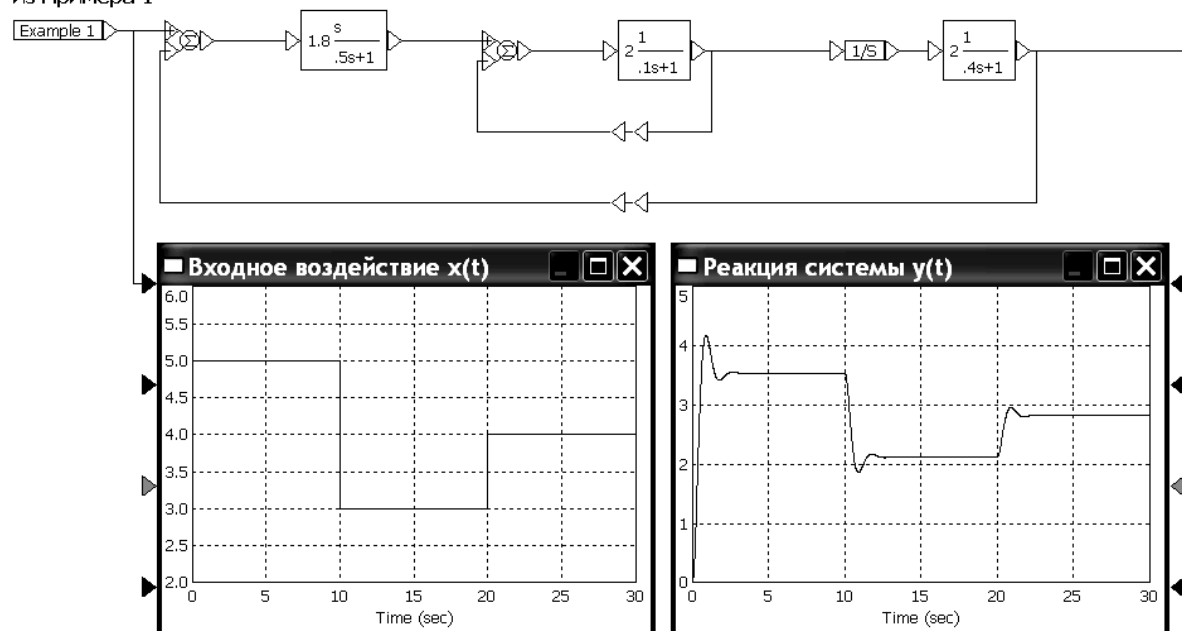
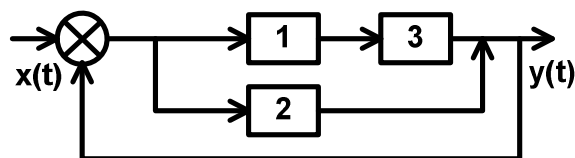


Рис. 72

Пусть задана структура системы № 2, состоящей из элементов, для которых известны модели в виде передаточных функций и вид внешнего целенаправленного воздействия (гармоническая функция с параметрами: период $T = 1$ с и амплитудой $A = 1$).



$$W_1(s) = \frac{1}{0,5s + 1}; \quad W_2(s) = \frac{1}{0,1s + 1}; \quad W_3(s) = \frac{2}{s(s^2 + 0,5s + 1)(2s + 1)}.$$

Результат агрегирования системы приведен на рис. 73, а. Решение задачи динамики в VisSim для данного примера, полученное в виде графика реакции системы (изменения выходного параметра), показывает, что система является неустойчивой: при подаче на вход гармонического воздействия с $A = 1$ и $T = 1$ с на выходе возникает колебательный процесс, не достигающий стационарности (постоянства амплитуды).

Выполним решение задачи динамики для линейно убывающего воздействия с ограничением, модель источника которого была разработана в примере 1.

Результат решения, приведенный на рис. 73, б, подтверждает неустойчивости системы: на выходе возникает колебательный процесс, не достигающий стационарности (постоянства амплитуды).

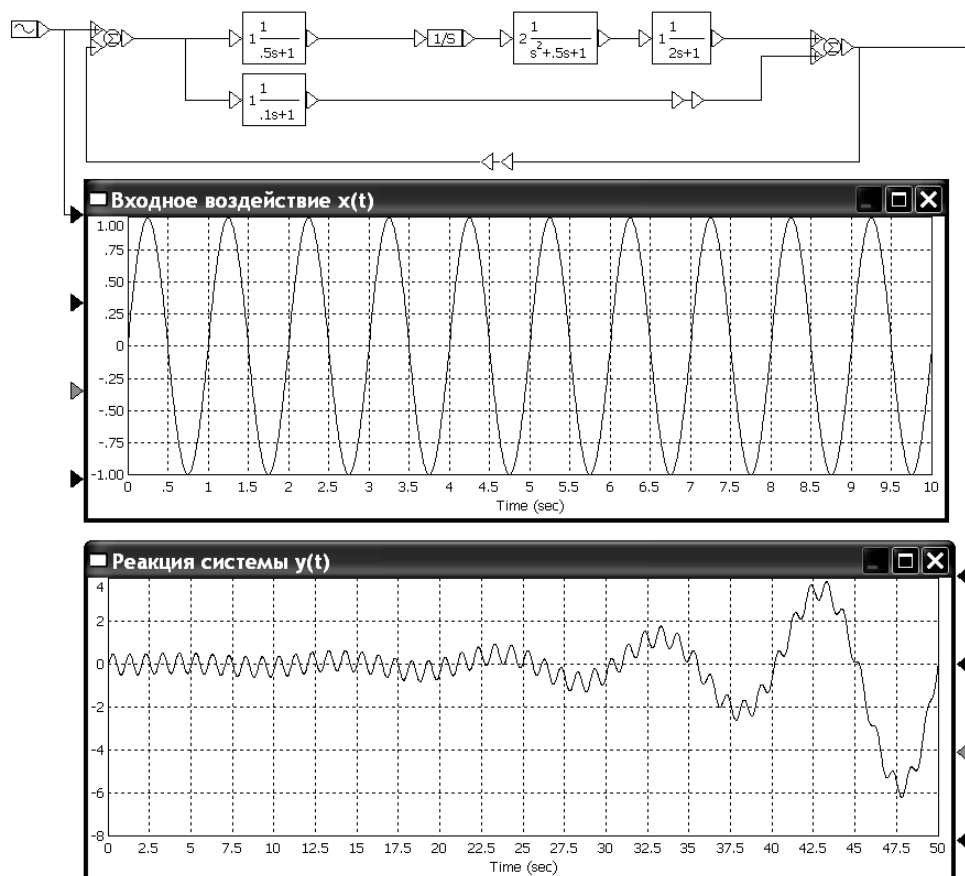
Пример 3

Моделирование источников внешних воздействий нелинейного вида

Для моделирования нелинейных источников внешних воздействий в VisSim используются генераторы параболического воздействия и синусоидального сигнала из палитры *Signal Producer*. Кроме того, для моделирования используются выходные сигналы блоков, заданных передаточной функцией *Blocks – Linear System – transferFunction*, при подаче на их вход единичного ступенчатого воздействия.

Например, сигнал вида $x(t) = 1 - e^{-\frac{t}{T}}$ для $t \geq t_0$ (экспоненциальная функция, асимптотически приближающаяся к 1) является выходом блока с передаточной функцией $W(s) = \frac{1}{T \cdot s + 1}$. Для $T = 2$ имеем вид воздействия, приведенный на рис. 74.

а



б

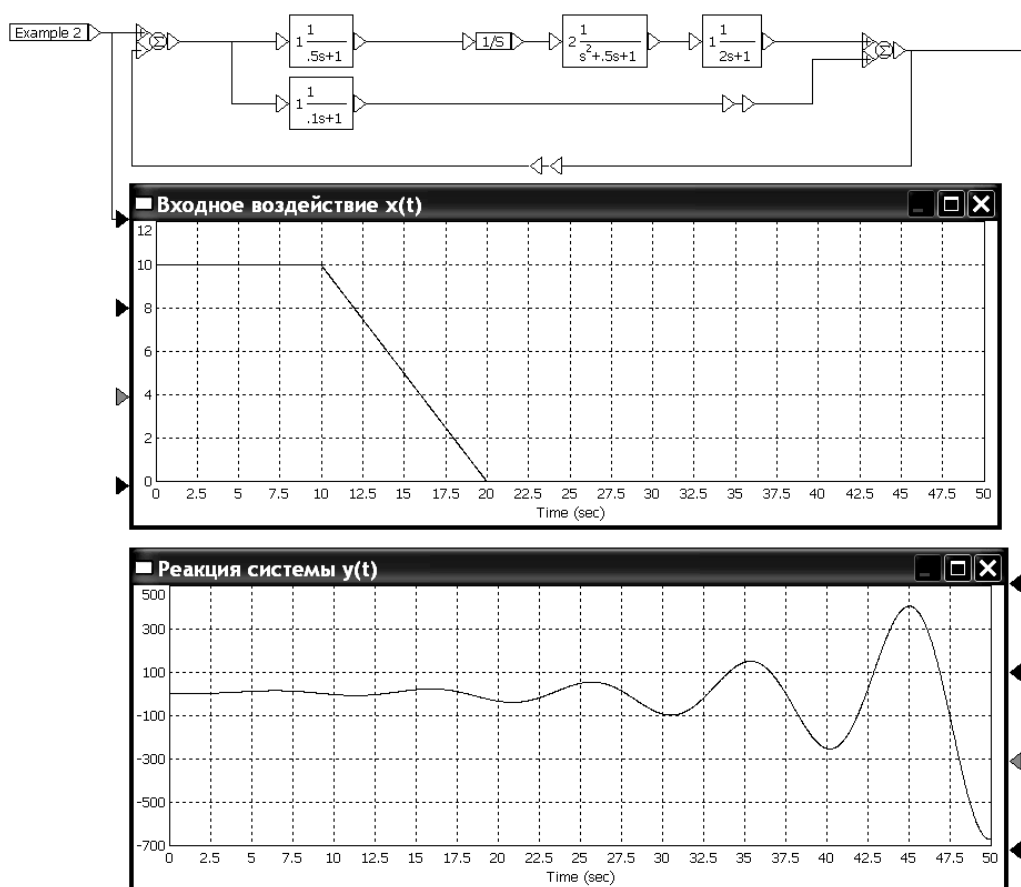


Рис. 73

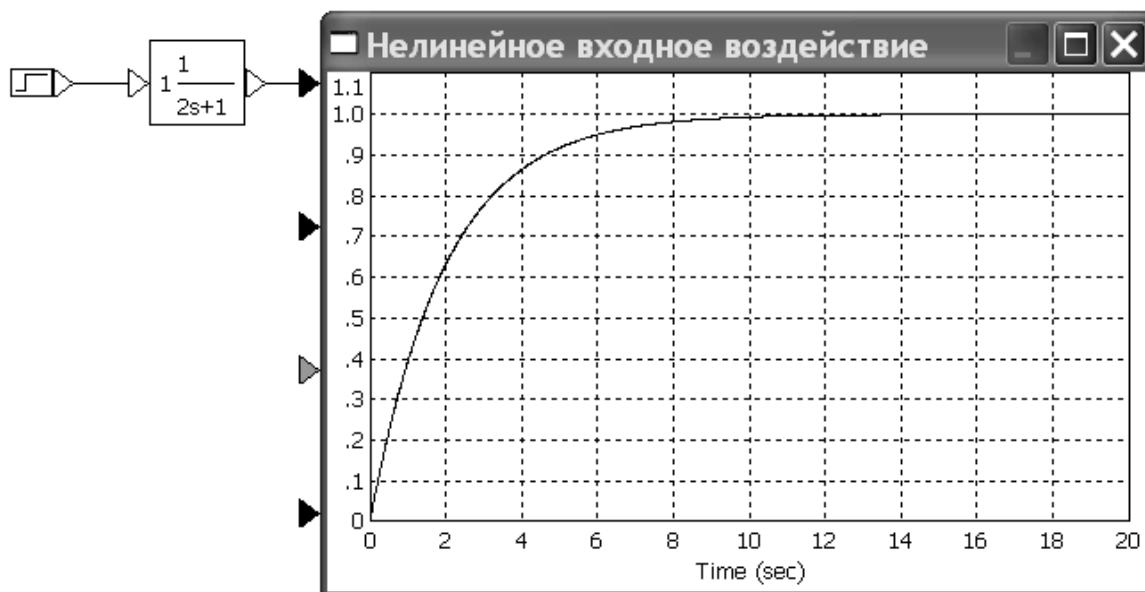


Рис. 74

При использовании в моделях источника блока, у которого полином знаменателя передаточной функции имеет второй порядок, можно получить нелинейное воздействие с точкой перегиба или вида затухающего колебательного сигнала (рис. 75, 76).

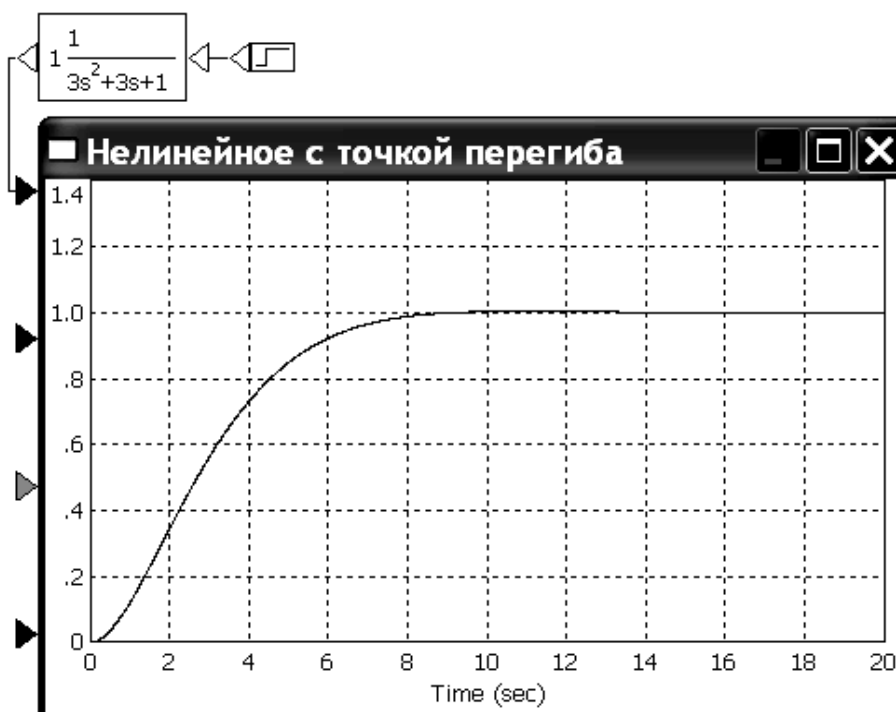


Рис. 75

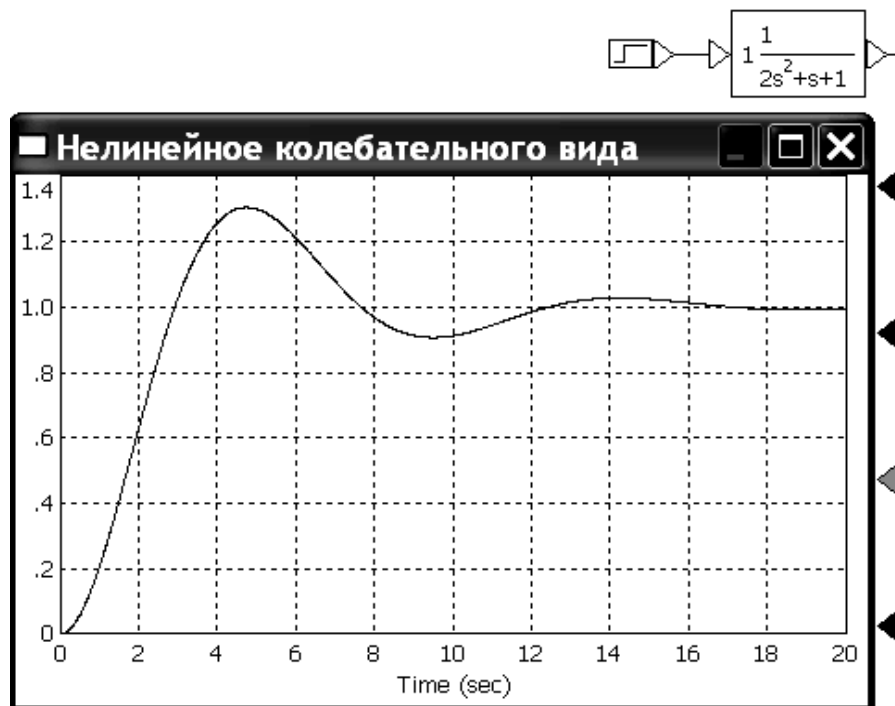
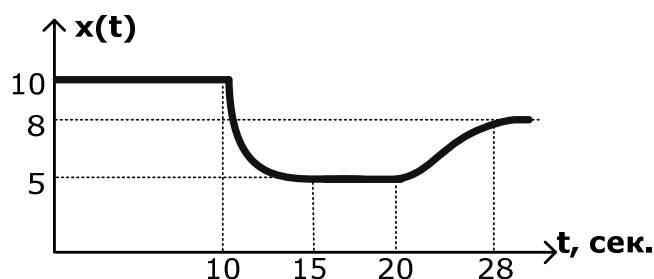


Рис. 76

Пусть требуется реализовать модель источника входного нелинейного воздействия системы вида



В VisSim реализовать модель источника воздействия такого вида можно с использованием нескольких составляющих. В течение первых десяти секунд воздействие представляет собой постоянную величину, равную 10. С 10 до 15 с воздействие по экспоненте убывает до значения 5 и это постоянное значение сохраняется до 20 с. Для реализации этой части воздействия из постоянного величины, равной 10, с задержкой в 10 с вычитается сигнал, являющийся выходом блока с передаточной функцией первого порядка. На вход этого блока подана ступенька с амплитудой, равной 5, т.к. требуется убывание амплитуды воздействия до значения 5. Постоянные коэффициенты знаменателя передаточной функции подбира-

ются таким образом, чтобы обеспечить время достижения воздействием минимального значения, равное приблизительно 5 с.

Далее, начиная с 20 с, воздействие плавно, по кривой с точкой перегиба, нарастает до значения 8. Для реализации этого нарастания к уже реализованному воздействию с момента времени 20 с следует добавить сигнал, являющийся выходом блока с передаточной функцией второго порядка. На вход этого блока следует подать ступеньку с амплитудой, равной 3, так как требуется нарастание значения воздействия от 5 до 8. Постоянные коэффициенты знаменателя передаточной функции подбираются таким образом, чтобы обеспечить время достижения воздействием значения 8 приблизительно за 8 с.

Реализация модели источника заданного сигнала с выводом результата моделирования в VisSim имеет вид, приведенный на рис. 77 (см. также файл *Пример моделирования входного воздействия.vsm*).

Пример 4. Решение задач эквивалентных преобразований разомкнутых линейных систем

Разомкнутая система (система без учета общей обратной связи) для системы № 1 из примера 2 представляет собой последовательное соединение первого элемента со вторым, охваченным единичной отрицательной обратной связью, и третьим. Передаточная функция разомкнутой системы в соответствии с определенными типами соединений элементов будет иметь вид:

$$\begin{aligned}
 W_{раз}(s) &= W_1(s) \frac{W_2(s)}{1 + W_2(s)} W_3(s) = 1.8 \frac{s}{0,5s + 1} \cdot 2 \frac{1}{s(0,4s + 1)} \cdot \frac{2 \frac{1}{0,1s + 1}}{1 + 2 \frac{1}{0,1s + 1}} = \\
 &= 3,6 \frac{s}{0,5s + 1} \cdot \frac{1}{s(0,4s + 1)} \cdot \frac{2 \frac{1}{0,1s + 1}}{1 + 2 \frac{1}{0,1s + 1}} = \frac{3.6}{(0,5s + 1) \cdot (0,4s + 1)} \cdot \frac{2}{1 + \frac{2}{0,1s + 1}}.
 \end{aligned}$$

Осуществим преобразования полученной передаточной функции разомкнутой системы для приведения ее к виду эквивалентного последовательного соединения элементов.

Автоматизировать эквивалентные преобразования моделей САУ можно с использованием MathCAD. Операции над математическими выражениями выполняются командами меню *Symbolics* (*Символика*).

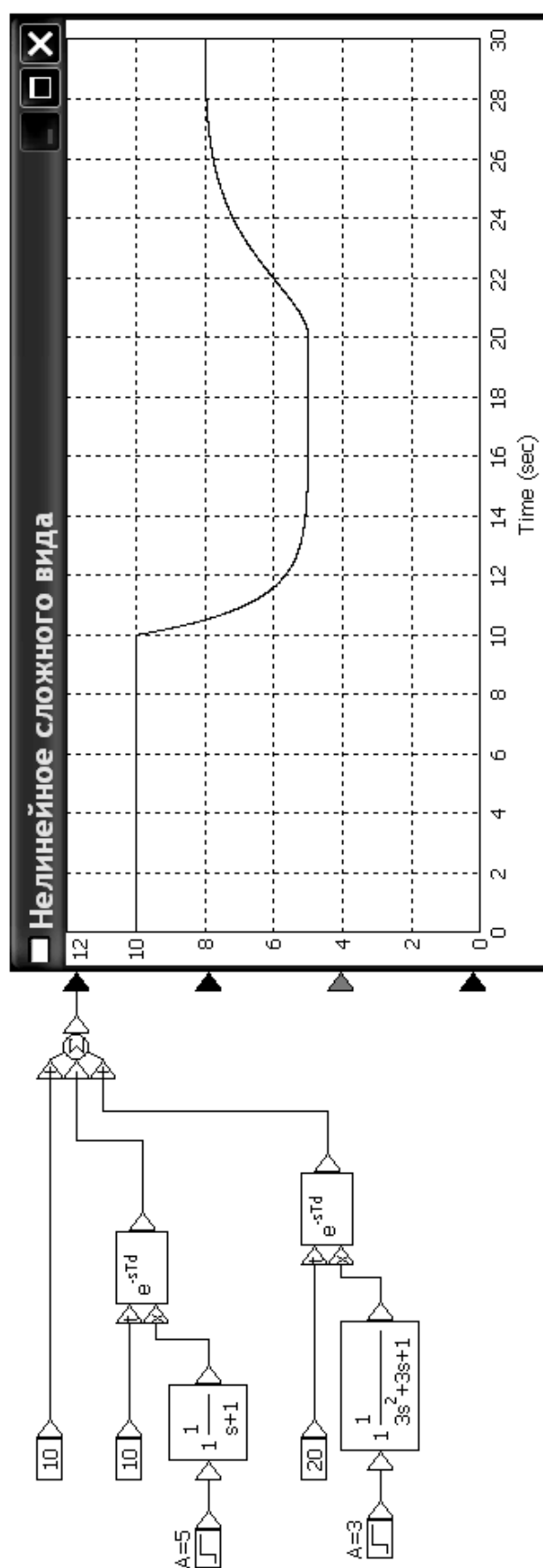


Рис. 77

Для выполнения операции выражение следует выделить. К сложным выражениям операции рекомендуется применять по частям.

Операции с выражениями:

- *Simplify (Упрощение)* – упростить выделенное выражение с выполнением таких операций, как приведение подобных слагаемых, приведение дробей к общему знаменателю и т.д.;
- *Expand (Расширить)* – раскрыть выражение: например, для $(x - y) \cdot (x + y)$ получим $(x^2 - y^2)$;
- *Factor (Фактор)* – разложить на простые множители: например, для $(x^2 - y^2)$ получим $(x - y) \cdot (x + y)$.

Осуществим эквивалентные преобразования с использованием пакета MathCAD. Введем в рабочем поле исходный вид передаточной функции разомкнутой системы. Ввод начинается с установки щелчком левой кнопки мыши курсора в произвольную позицию ввода. Формульный редактор MathCAD имеет следующие особенности:

- оператор умножения вводится как звездочка, но в рабочем поле отображается точкой в середине строки;
- оператор деления вводится как косая черта, но в рабочем поле отображается горизонтальной чертой;
- оператор возведения в степень вводится знаком $\wedge$, но число в степени в рабочем поле отображается в обычном виде – как верхний индекс.

Выделим часть выражения передаточной функции и выберем в меню *Symbolics (Символика)* операцию *Simplify (Упрощение)* (рис. 78).

Результат применения операции имеет вид (рис. 79).

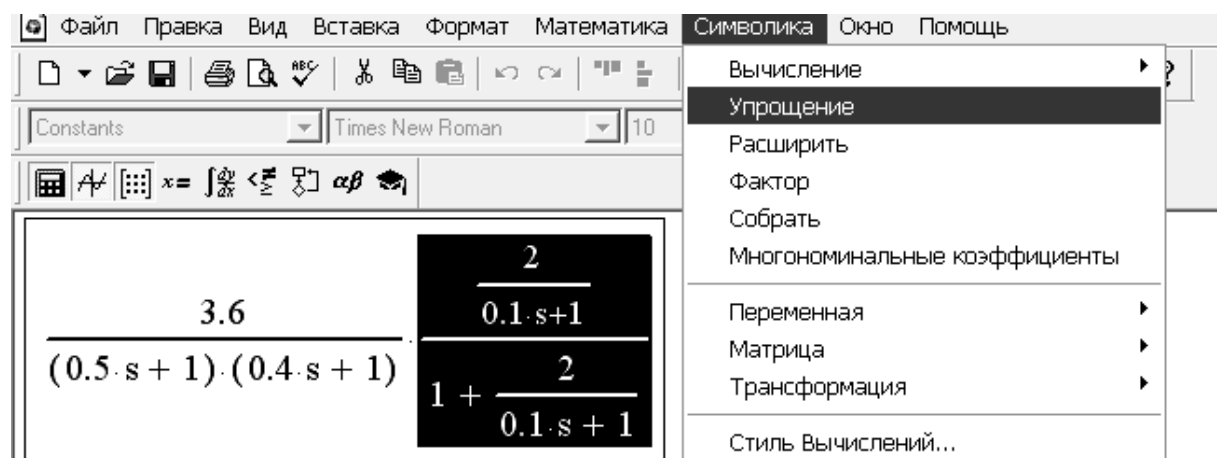
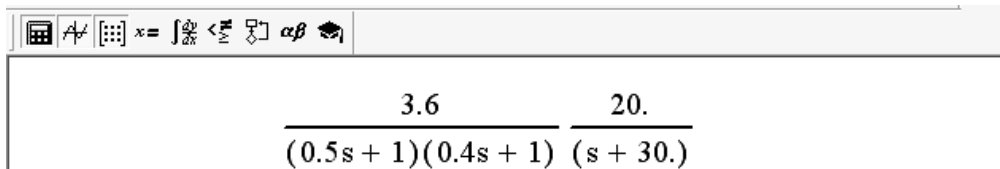


Рис. 78



$$\frac{3.6}{(0.5s + 1)(0.4s + 1)} \cdot \frac{20.}{(s + 30.)}$$

Рис. 79

С учетом того, что $3.6 \cdot 20 / 30 = 2.4$ и $1 / 30 = 0.033$ запишем конечный вид эквивалентной передаточной функции разомкнутой системы, приведенной к последовательному соединению элементов:

$$W_{раз}^*(s) = W_1^*(s) \cdot W_2^*(s) \cdot W_3^*(s) = \frac{2.4}{0.5s + 1} \cdot \frac{1}{0.4s + 1} \cdot \frac{1}{0.033s + 1}.$$

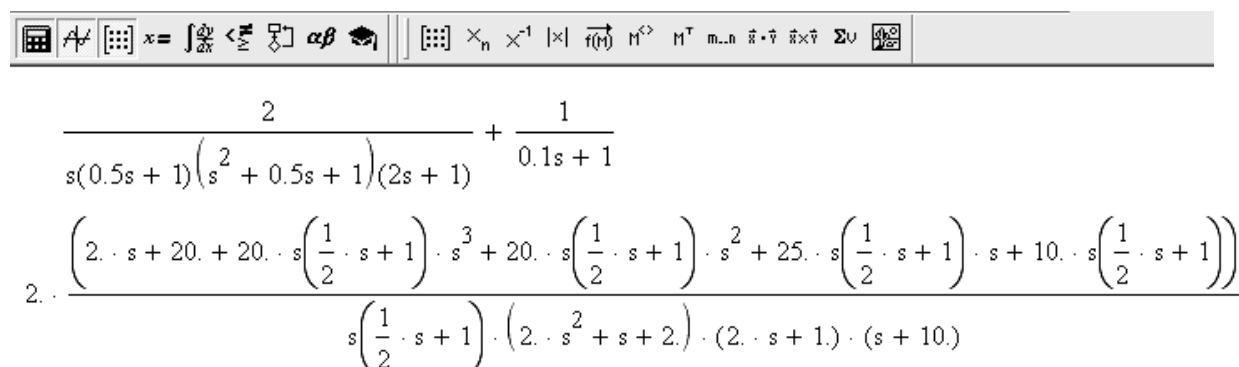
Реализуем полученную эквивалентную модель в пакете VisSim. Для удобства сравнения решений задачи динамики осуществим вывод реакции эквивалентной системы в поле осциллографа исходной системы (рис. 81).

Идентичность реакций исходной и эквивалентной систем подтверждается полным совпадением графиков решений $y(t)$ и $y^*(t)$.

Разомкнутая система (система без учета общей обратной связи) для системы №2 из Примера 2 представляет собой последовательное соединение первого элемента с третьим и к этому соединению параллельно подключен второй элемент. Передаточная функция разомкнутой системы, таким образом, имеет вид:

$$W_{раз}(s) = W_1(s)W_3(s) + W_2(s) = \frac{1}{0.5s + 1} \cdot \frac{2}{s(s^2 + 0.5s + 1)(2s + 1)} + \frac{1}{0.1s + 1}.$$

Осуществим преобразования полученной передаточной функции разомкнутой системы для приведения ее к виду эквивалентного последовательного соединения элементов. Введем передаточную функцию в рабочем поле MathCAD и выполним операцию *Simplify* (*Упрощение*) (рис. 80).



$$\frac{2}{s(0.5s + 1)(s^2 + 0.5s + 1)(2s + 1)} + \frac{1}{0.1s + 1}$$

$$2 \cdot \frac{\left(2 \cdot s + 20. + 20. \cdot s \left(\frac{1}{2} \cdot s + 1 \right) \cdot s^3 + 20. \cdot s \left(\frac{1}{2} \cdot s + 1 \right) \cdot s^2 + 25. \cdot s \left(\frac{1}{2} \cdot s + 1 \right) \cdot s + 10. \cdot s \left(\frac{1}{2} \cdot s + 1 \right) \right)}{s \left(\frac{1}{2} \cdot s + 1 \right) \cdot \left(2 \cdot s^2 + s + 2. \right) \cdot \left(2 \cdot s + 1. \right) \cdot \left(s + 10. \right)}$$

Рис. 80

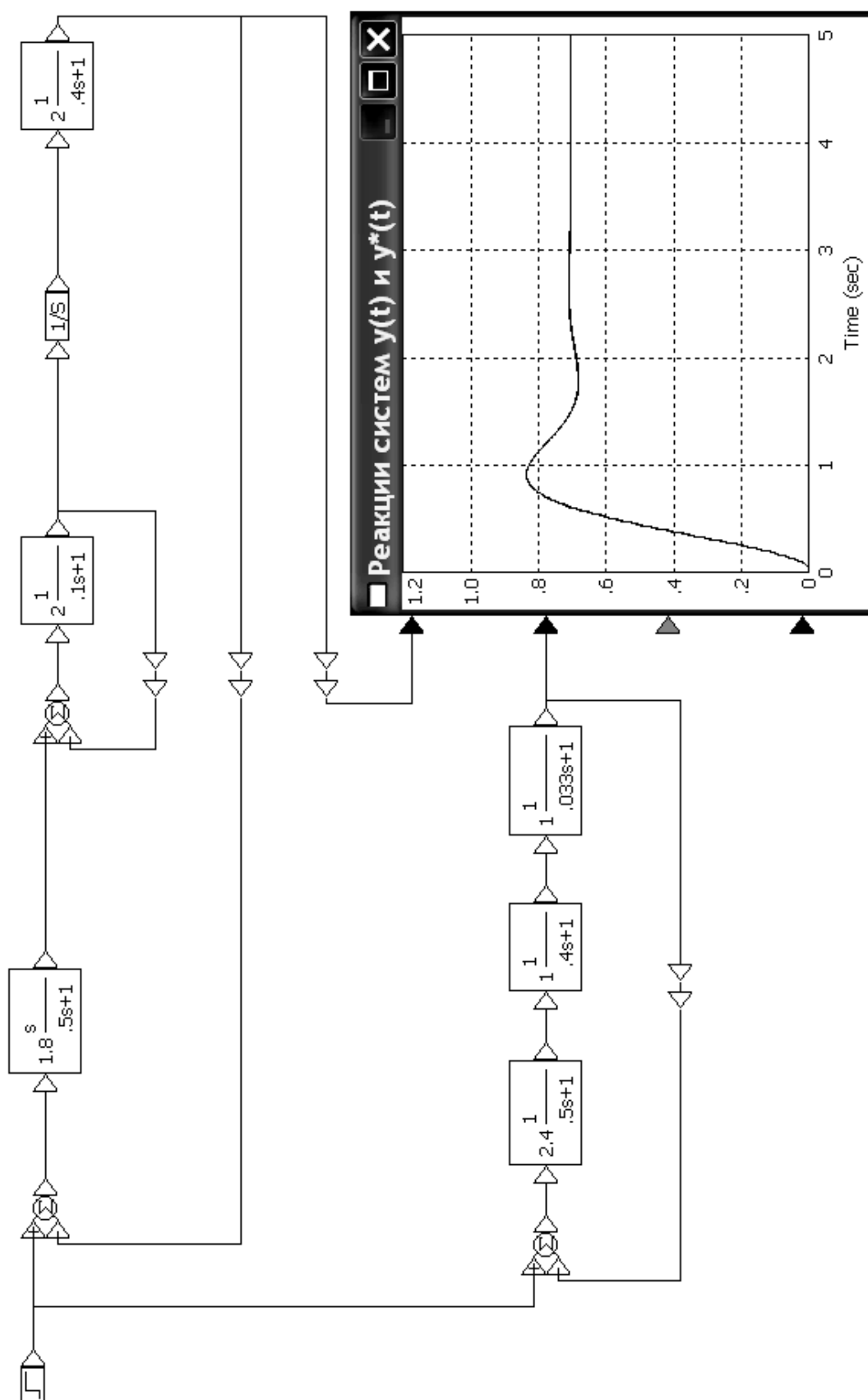
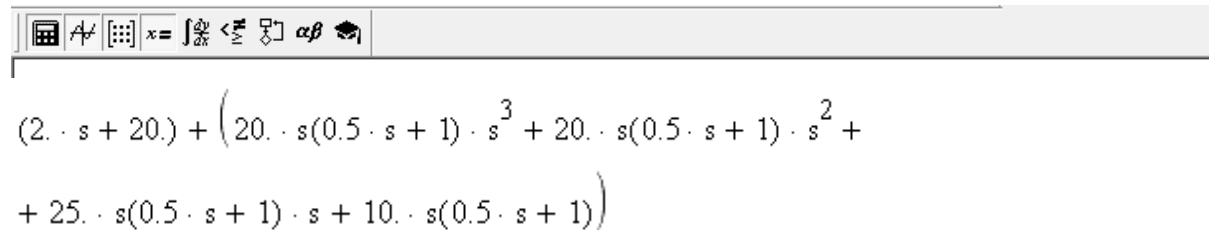


Рис. 81

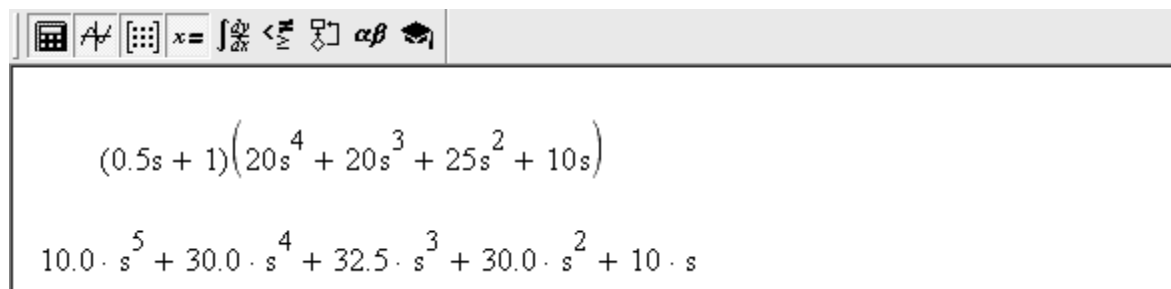
Далее, ввиду сложности выражения, будем выполнять его преобразование по частям. Числитель передаточной функции эквивалентной системы можно представить суммой двух многочленов (рис. 82).



$$(2 \cdot s + 20) + \left(20 \cdot s(0.5 \cdot s + 1) \cdot s^3 + 20 \cdot s(0.5 \cdot s + 1) \cdot s^2 + 25 \cdot s(0.5 \cdot s + 1) \cdot s + 10 \cdot s(0.5 \cdot s + 1) \right)$$

Рис. 82

Вынесем за скобки во втором многочлене общий множитель и выполним операцию *Expand* (*Расширить*) (рис. 83).

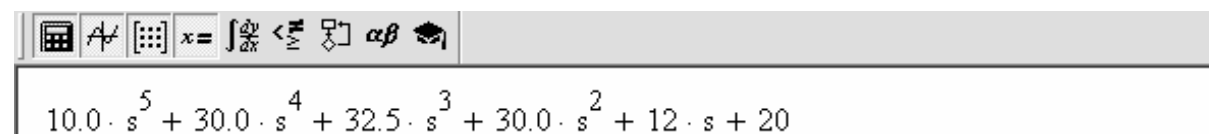


$$(0.5s + 1)(20s^4 + 20s^3 + 25s^2 + 10s)$$

$$10.0 \cdot s^5 + 30.0 \cdot s^4 + 32.5 \cdot s^3 + 30.0 \cdot s^2 + 10 \cdot s$$

Рис. 83

С учетом первого многочлена $(2s+20)$, числитель передаточной функции эквивалентной системы имеет вид (рис. 84).



$$10.0 \cdot s^5 + 30.0 \cdot s^4 + 32.5 \cdot s^3 + 30.0 \cdot s^2 + 12 \cdot s + 20$$

Рис. 84

В пакете VisSim полином числителя в описании блока *transferFunction* не может иметь степень больше, чем степень полинома знаменателя, поэтому приведем знаменатель передаточной функции эквивалентной системы к виду (рис. 85).

Выполним расчет коэффициентов полинома числителя и знаменателя и общего коэффициента передачи эквивалентной передаточной функции (рис. 86, 87).

$$s \left(\frac{1}{2} \cdot s + 1 \right) \cdot (2 \cdot s^2 + s + 2) \cdot (2 \cdot s + 1) \cdot (s + 10) \\ s(2.0 \cdot s^5 + 26.0 \cdot s^4 + 66.5 \cdot s^3 + 71.0 \cdot s^2 + 62.0 \cdot s + 20.)$$

Рис. 85

$$10.0 \cdot s^5 + 30.0 \cdot s^4 + 32.5 \cdot s^3 + 30.0 \cdot s^2 + 12 \cdot s + 20 \\ \frac{10}{20} = 0.5 \quad \frac{30}{20} = 1.5 \quad \frac{32.5}{20} = 1.625 \quad \frac{30}{20} = 1.5 \quad \frac{12}{20} = 0.6 \\ 20(0.5 \cdot s^5 + 1.5 \cdot s^4 + 1.625 \cdot s^3 + 1.5 \cdot s^2 + 0.6 \cdot s + 1)$$

Рис. 86

$$s(2.0 \cdot s^5 + 26.0 \cdot s^4 + 66.5 \cdot s^3 + 71.0 \cdot s^2 + 62.0 \cdot s + 20.) \\ \frac{2}{20} = 0.1 \quad \frac{26}{20} = 1.3 \quad \frac{66.5}{20} = 3.325 \quad \frac{71}{20} = 3.55 \quad \frac{62}{20} = 3.1 \\ 20s(0.1 \cdot s^5 + 1.3 \cdot s^4 + 3.325 \cdot s^3 + 3.55 \cdot s^2 + 3.1 \cdot s + 1)$$

Рис. 87

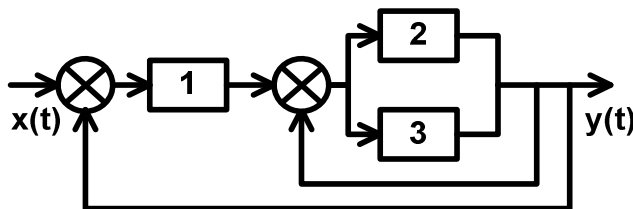
Таким образом, конечный вид эквивалентной передаточной функции разомкнутой системы, приведенной к последовательному соединению элементов:

$$\begin{aligned}
W_{раз}^*(s) &= 2 \cdot \frac{20(0,5s^5 + 1,5s^4 + 1,625s^3 + 1,5s^2 + 0,6s + 1)}{20s(0,1s^5 + 1,3s^4 + 3,325s^3 + 3,55s^2 + 3,1s + 1)} = \\
&= 2 \cdot \frac{(0,5s^5 + 1,5s^4 + 1,625s^3 + 1,5s^2 + 0,6s + 1)}{s(0,1s^5 + 1,3s^4 + 3,325s^3 + 3,55s^2 + 3,1s + 1)}; \\
W_{раз}^*(s) &= W_1^*(s) \cdot W_2^*(s) = \\
&= \frac{2}{s} \cdot \frac{(0,5s^5 + 1,5s^4 + 1,625s^3 + 1,5s^2 + 0,6s + 1)}{(0,1s^5 + 1,3s^4 + 3,325s^3 + 3,55s^2 + 3,1s + 1)}.
\end{aligned}$$

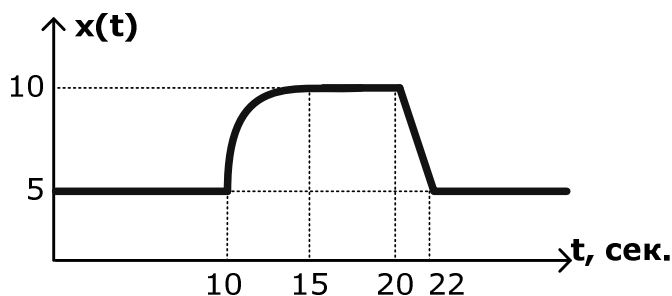
Реализуем полученную эквивалентную модель в пакете VisSim. Для удобства сравнения решений задачи динамики осуществим вывод реакции эквивалентной системы в поле осциллографа исходной системы (рис. 88). Идентичность реакций исходной системы $y(t)$ и эквивалентной разомкнутой системы $y^*(t)$ подтверждается практическим совпадением графиков решений. Имеющаяся погрешность объясняется округлениями при вычислении коэффициентов полиномов.

Варианты индивидуальных заданий

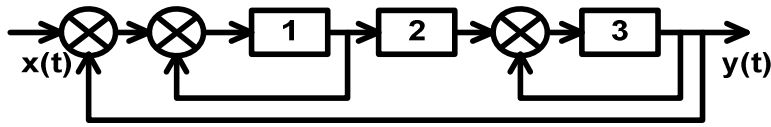
1.



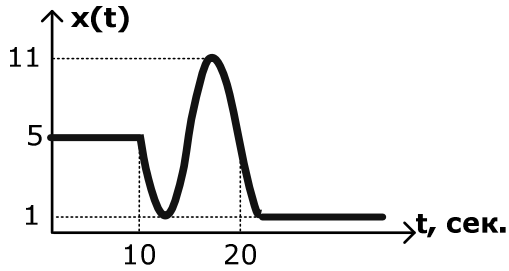
$$W_1(s) = 10 \frac{1}{0,2s + 1}; \quad W_2(s) = \frac{1}{0,4s + 1}; \quad W_3(s) = \frac{2}{0,01s^2 + 0,5s + 1}.$$



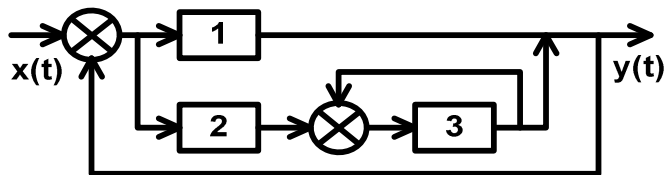
2.



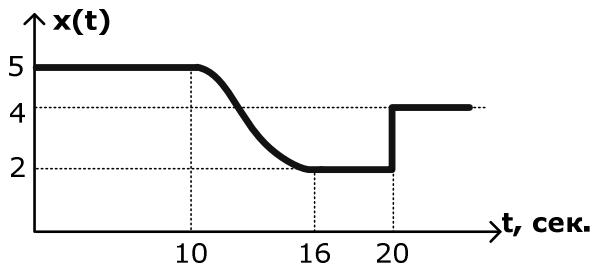
$$W_1(s) = 3 \frac{1}{0,4s + 1}; \quad W_2(s) = 2 \frac{1}{0,5s + 1}; \quad W_3(s) = 2 \frac{1}{0,01s^2 + 0,5s + 1}.$$



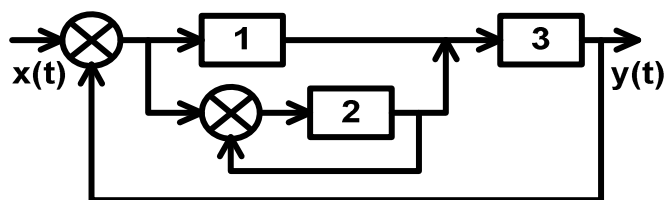
3.



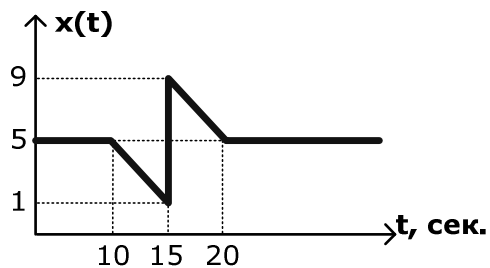
$$W_1(s) = 10 \frac{s}{0,2s^2 + 1}; \quad W_2(s) = \frac{1}{0,4s + 1}; \quad W_3(s) = \frac{1}{0,2s^2 + 0,5s + 1}.$$



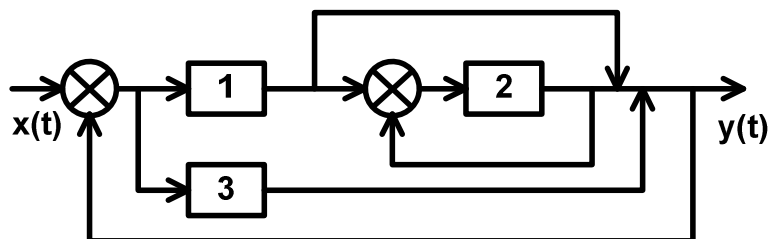
4.



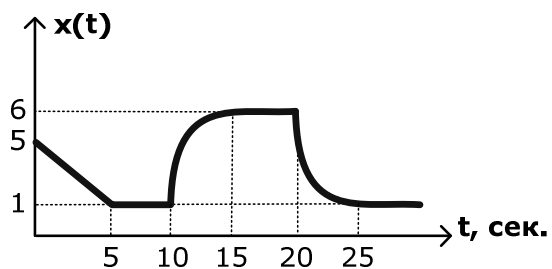
$$W_1(s) = 10 \frac{1}{s(0,2s^2 + 1)}; \quad W_2(s) = \frac{1}{0,4s + 1}; \quad W_3(s) = 2 \frac{1}{0,2s^2 + 0,5s + 1}.$$



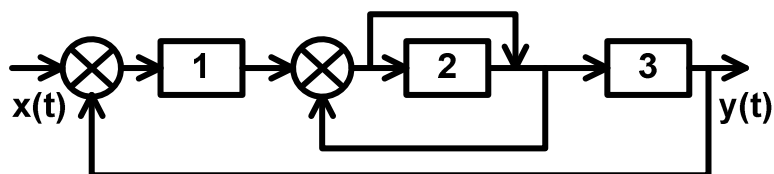
5.



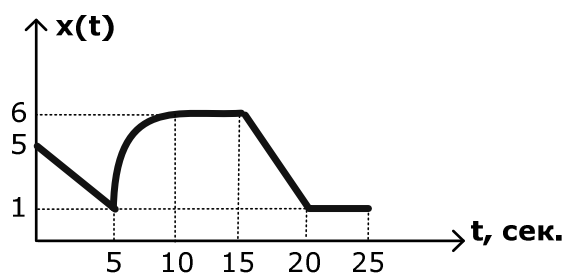
$$W_1(s) = 3 \frac{1}{0,4s+1}; \quad W_2(s) = 2 \frac{1}{0,8s+1}; \quad W_3(s) = 2 \frac{1}{2s+1}.$$



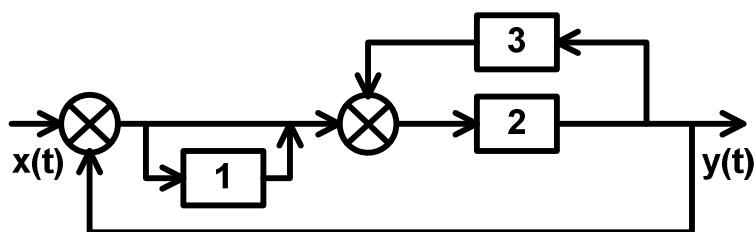
6.



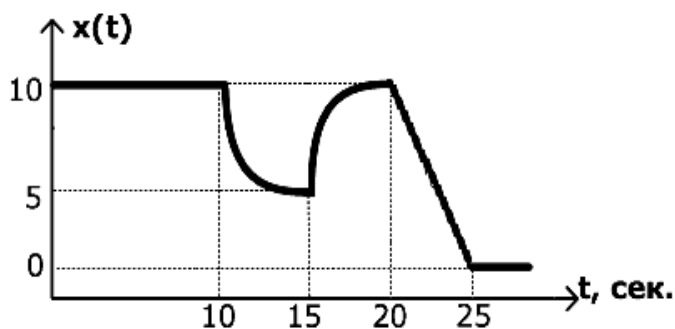
$$W_1(s) = 5 \frac{s}{0,4s^2+1}; \quad W_2(s) = 2 \frac{1}{(0,5s+1)}; \quad W_3(s) = 10 \frac{1}{0,01s^2+0,5s+1}.$$



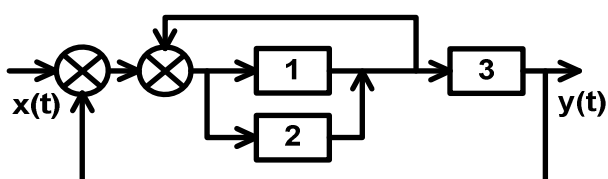
7.



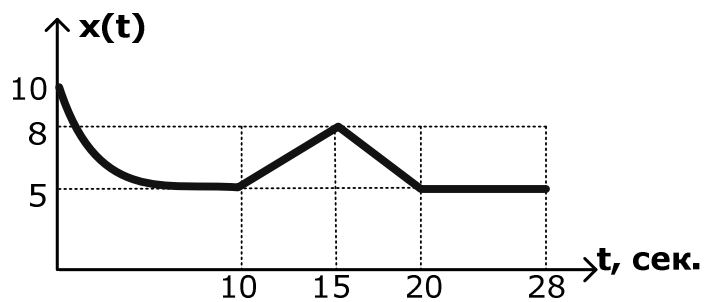
$$W_1(s) = 5 \frac{1}{s(0,2s+1)}; W_2(s) = \frac{s}{(s^2 + 0,2s+1)}; W_3(s) = \frac{1}{s}.$$



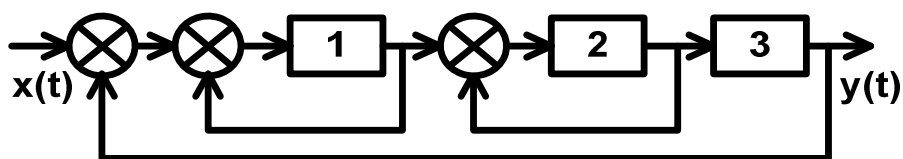
8.



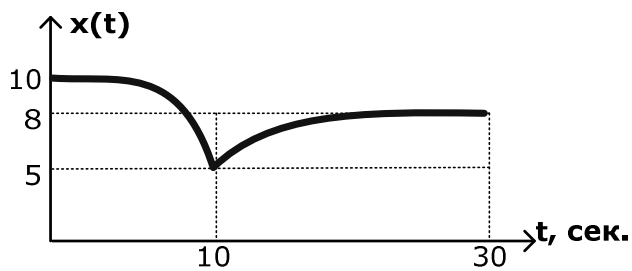
$$W_1(s) = 5 \frac{1}{(0,2s+1)}; W_2(s) = 20 \frac{s}{0,4s+1}; W_3(s) = \frac{1}{0,5s+1}.$$



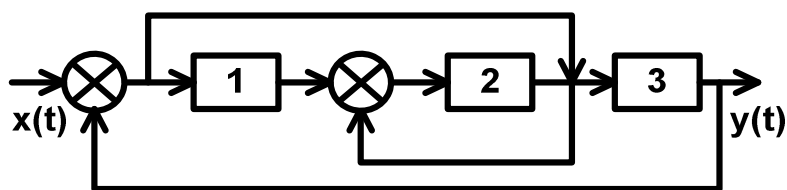
9.



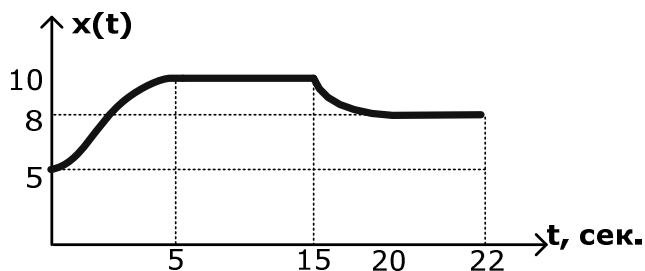
$$W_1(s) = 5 \frac{1}{s(0,2s+1)}; \quad W_2(s) = 20 \frac{s}{0,4s+1}; \quad W_3(s) = \frac{1}{0,5s+1}.$$



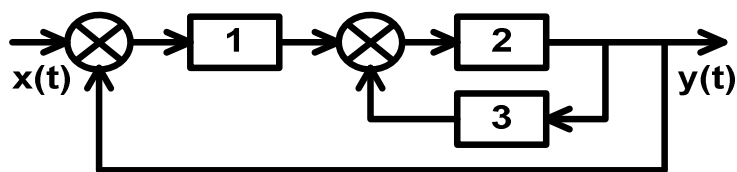
10.



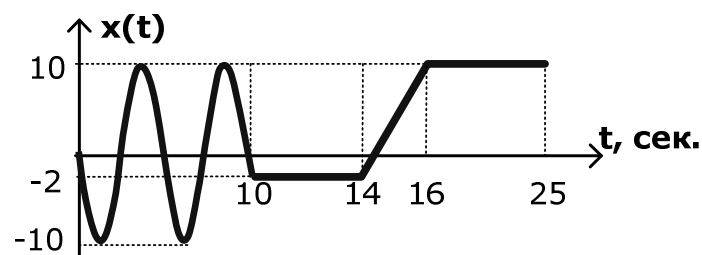
$$W_1(s) = \frac{s}{(s^2 + 0,2s + 1)}; \quad W_2(s) = \frac{1}{0,8s + 1}; \quad W_3(s) = 3 \frac{1}{s(0,5s + 1)}.$$



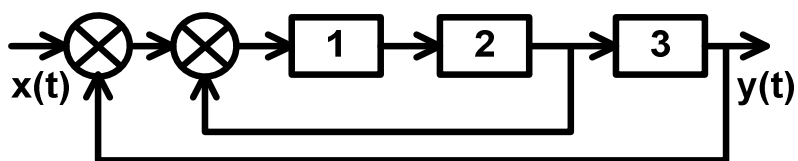
11.



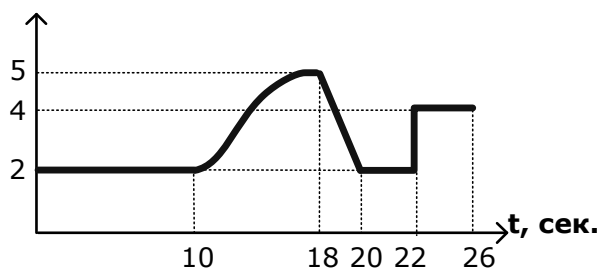
$$W_1(s) = \frac{1}{s}; \quad W_2(s) = 10 \frac{0,2s + 1}{s^2 + s + 1}; \quad W_3(s) = \frac{1}{0,1s + 1}.$$



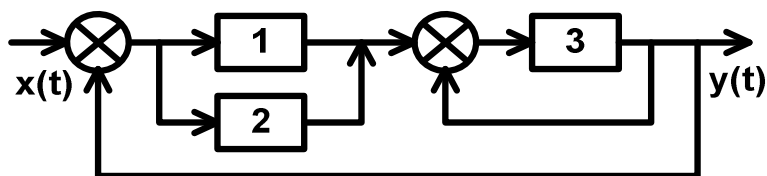
12.



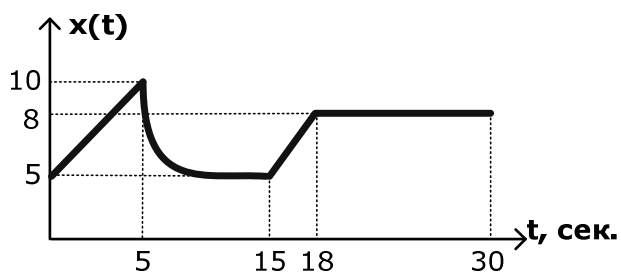
$$W_1(s) = 20 \frac{s}{0,4s+1}; \quad W_2(s) = 3 \frac{1}{s(0,5s+1)}; \quad W_3(s) = 10 \frac{1}{s(0,2s^2+1)}.$$



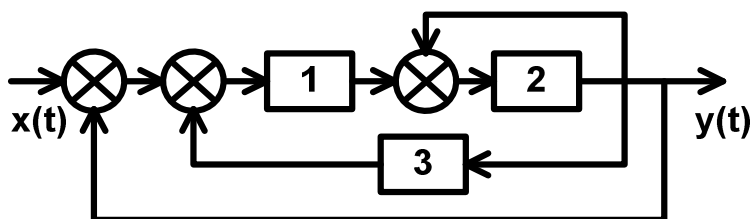
13.



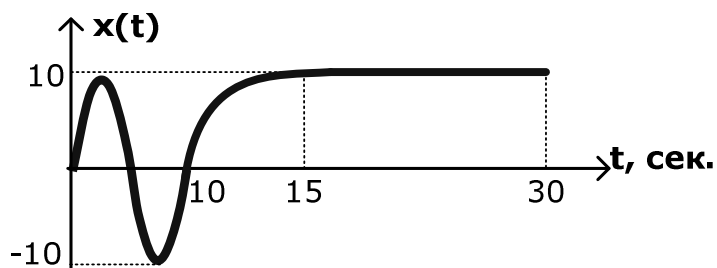
$$W_1(s) = 5 \frac{1}{s(0,2s+1)}; \quad W_2(s) = 10 \frac{0,5s^2+s}{s^2+s+1}; \quad W_3(s) = \frac{1}{0,5s+1}.$$



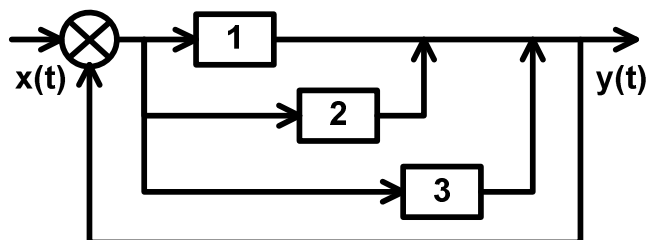
14.



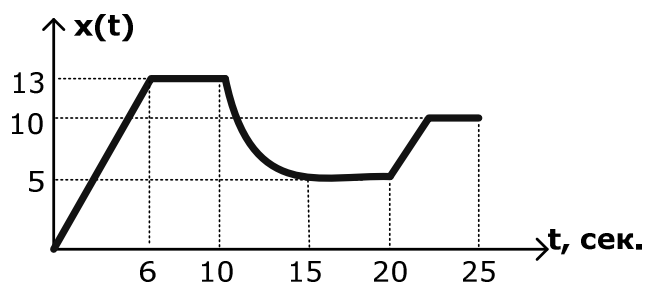
$$W_1(s) = 2 \frac{1}{0,4s+1}; \quad W_2(s) = 10 \frac{1}{s(0,2s^2+1)}; \quad W_3(s) = \frac{1}{s}.$$



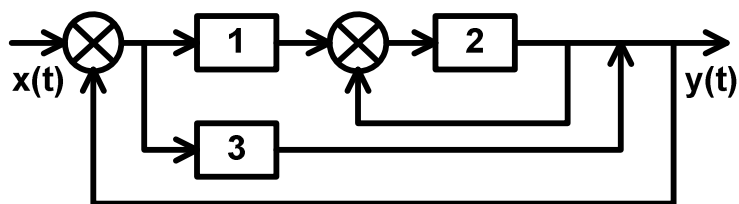
15.



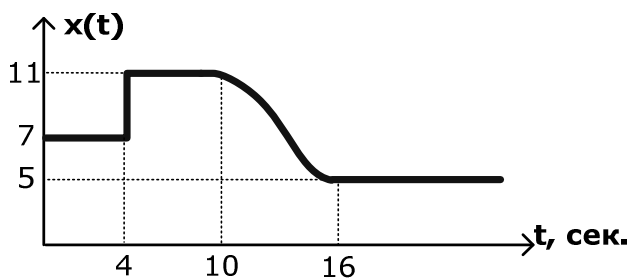
$$W_1(s) = 2 \frac{1}{0,4s+1}; \quad W_2(s) = 7 \frac{1}{s(0,5s+1)}; \quad W_3(s) = 10 \frac{1}{0,01s^2 + 0,5s+1}.$$



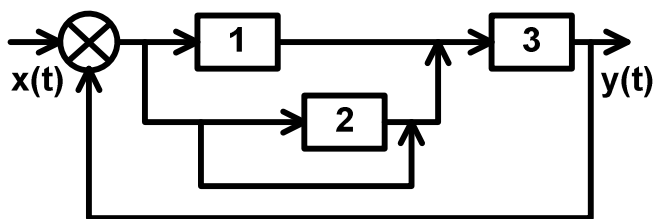
16.



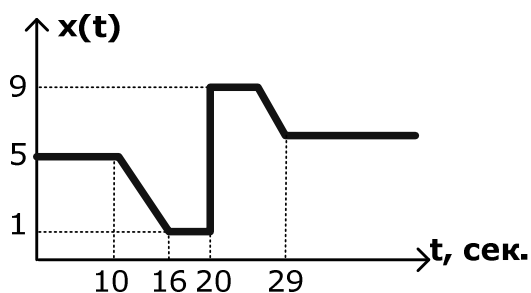
$$W_1(s) = 3 \frac{1}{0,4s+1}; \quad W_2(s) = 2 \frac{1}{0,8s^2 + 1}; \quad W_3(s) = 2 \frac{1}{s(s+1)}.$$



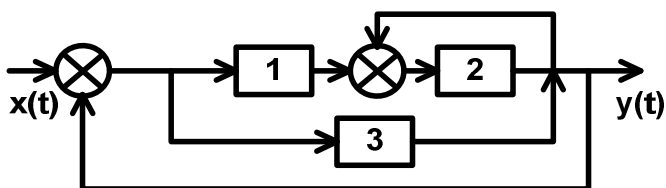
17.



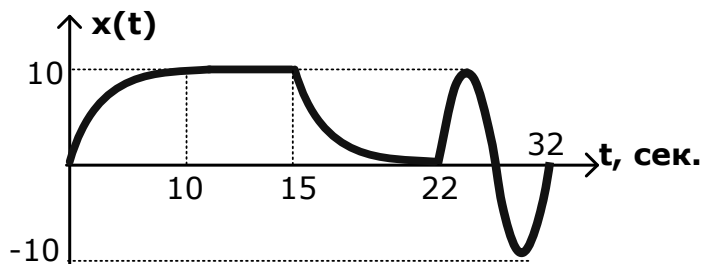
$$W_1(s) = 10 \frac{1}{0,2s^2 + 1}; \quad W_2(s) = \frac{1}{0,4s + 1}; \quad W_3(s) = 2 \frac{1}{0,2s^2 + 0,5s}.$$



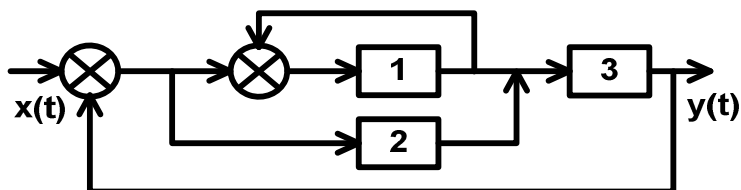
18.



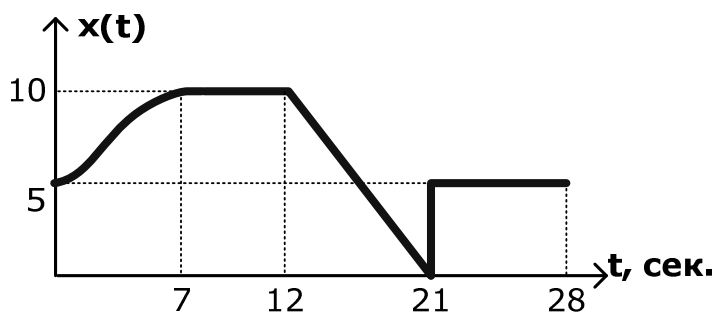
$$W_1(s) = 10 \frac{s}{0,2s + 1}; \quad W_2(s) = \frac{1}{0,4s + 1}; \quad W_3(s) = 5 \frac{1}{0,2s^2 + 0,5s + 1}.$$



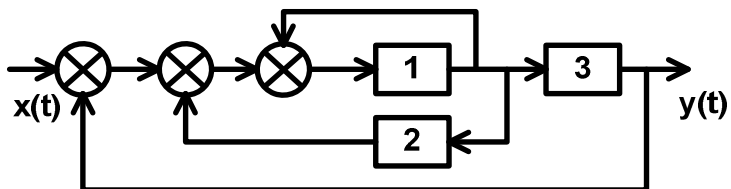
19.



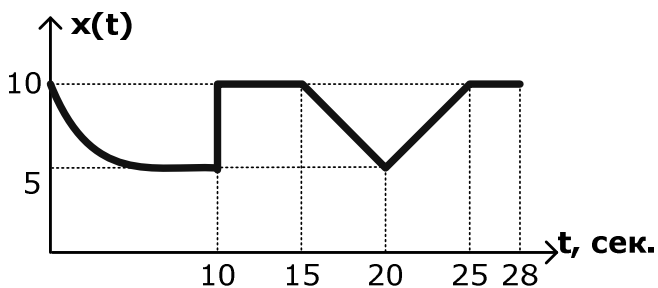
$$W_1(s) = \frac{1}{0,2s+1}; \quad W_2(s) = 2 \frac{s}{0,5s^2 + s + 1}; \quad W_3(s) = 2 \frac{1}{0,1s^2 + 0,5s + 1}.$$



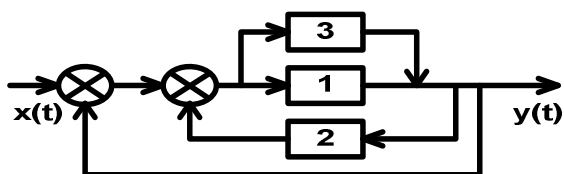
20.



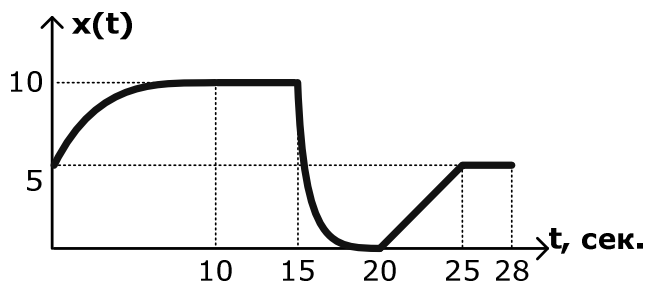
$$W_1(s) = \frac{2}{0,01s^2 + 0,5s + 1}; \quad W_2(s) = 10 \frac{1}{0,2s + 1}; \quad W_3(s) = \frac{1}{s}.$$



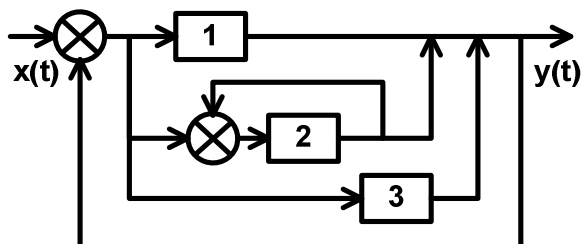
21.



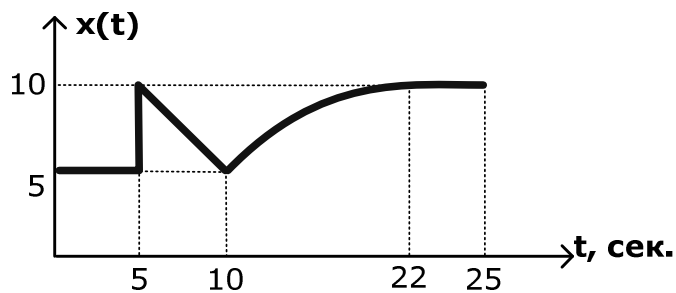
$$W_1(s) = 3 \frac{1}{0,4s + 1}; \quad W_2(s) = 2 \frac{1}{0,5s^2 + 1}; \quad W_3(s) = 20 \frac{1}{s}$$



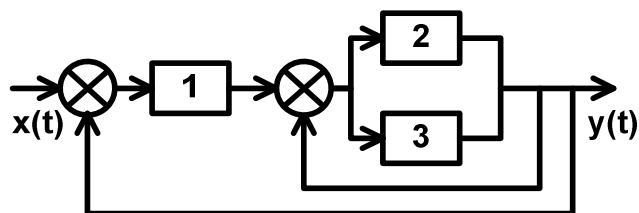
22.



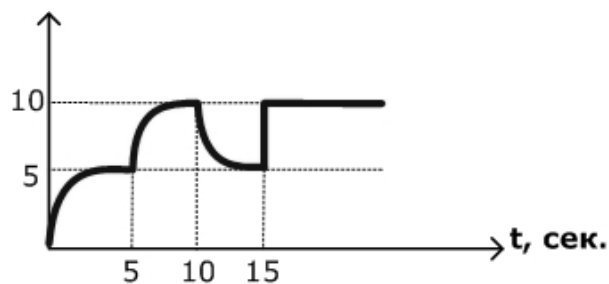
$$W_1(s) = 10 \frac{1}{s(0,2s+1)}; \quad W_2(s) = 10 \frac{1}{0,4s+1}; \quad W_3(s) = \frac{s}{0,01s^2 + 0,5s + 1}.$$



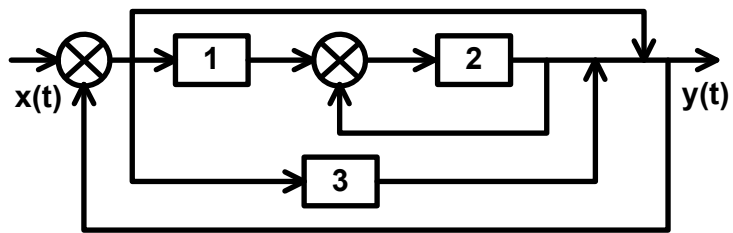
23.



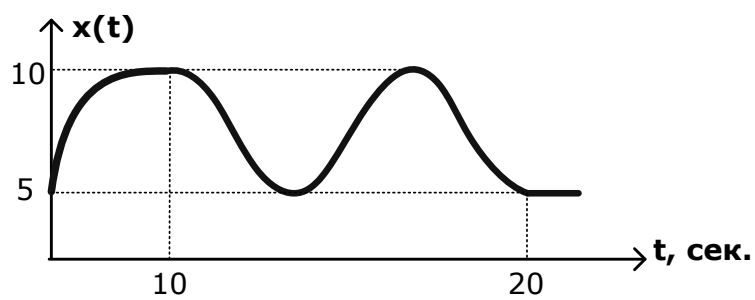
$$W_1(s) = 10 \frac{s}{s^2 + s + 1}; \quad W_2(s) = \frac{6}{s(5s+1)}; \quad W_3(s) = \frac{1}{0,8s+1}.$$



24.



$$W_1(s) = \frac{1}{0,8s+1}; \quad W_2(s) = \frac{1}{s^2+s+1}; \quad W_3(s) = \frac{1}{s(s+1)}.$$



Лабораторная работа № 2

Управление системными ресурсами

Цель: решение задачи разработки альтернативных вариантов распределения лимитированных ресурсов системы по нескольким функциям (видам работ) с учетом заданных ограничений по приоритетности выполнения работ и/или дополнительных условий распределения.

Программное обеспечение: MS Excel.

Примеры выполнения работы: файл *Управление ресурсами.xls*.

Теория

Задачи распределения ресурсов возникают, если есть некий набор запланированных работ, которые нужно выполнить, а наличных ресурсов для выполнения всех работ полностью не достаточно. Пусть для осуществления n видов работ необходимо m типов ресурсов, причем известно:

a_{ij} – количество i -того ресурса, необходимого для выполнения единичного объема j -того вида работ: $a_{ij} \geq 0$; $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$;

b_i – доступный объем (запас) i -того ресурса: $b_i > 0$;

c_j – вклад выполнения единичного объема j -того вида работ в общий результат целенаправленного функционирования системы: $c_j > 0$.

Пусть x_j – планируемый объем выполнения j -того вида работ. Тогда допустимым является такое планирование объемов выполняемых работ $x = (x_1, x_2, \dots, x_n)$, при котором суммарные затраты каждого i -того ресурса не превосходят его доступного объема (запаса):

$$\sum_{j=1}^n a_{ij} x_j \leq b_i; i = 1, \dots, m. \quad (11)$$

Кроме того, имеем следующее естественное ограничение:

$$x_j \geq 0; j = 1, \dots, n \quad (12)$$

Целевая функция $\sum_{j=1}^n c_j x_j$ отражает вклад каждого вида работ в общий результат. Задача распределения ресурсов ставится следующим образом: среди всех векторов x , удовлетворяющих ограничениям (11) и (12), найти такой, при котором величина $\sum_{j=1}^n c_j x_j$ принимает наибольшее или наименьшее значение. В случае максимизации целевая функция представ-

ляет собой положительный эффект (прибыль) от функционирования системы, в случае минимизации характеризует удельные затраты.

В ряде случаев вместо единой целевой функции могут быть заданы определенные приоритеты распределения или дополнительные ограничения по видам ресурсов. Тогда в постановке задачи система неравенств, учитывающих ограничения на ресурсы, дополняется условиями приоритетности или дополнительными ограничениями.

Также могут иметь место альтернативные постановки задачи распределения ресурсов:

1) заданы лишь наличные ресурсы, требуется определить, какой состав работ можно выполнить с этими ресурсами, чтобы обеспечить максимум некоторой меры эффективности. Например, задано предприятие с определенными производственными, материальными и трудовыми ресурсами, а требуется выбрать такой ассортимент изготавливаемой продукции, который обеспечит максимальную прибыль;

2) заданы работы, которые надо выполнить, требуется подобрать такие ресурсы, которые дают возможность выполнить их с минимальными производственными затратами. Например, известен перечень операций обслуживания, подлежащих выполнению в заданный интервал времени, а требуется подобрать необходимое число обслуживающих устройств, которые позволят выполнить план с минимальными затратами.

Ход выполнения работы

Дано: исходные данные для решения задачи распределения ресурсов; требования к двум альтернативным вариантам распределения в виде условий приоритетности определенных видов работ и/или дополнительных ограничений. Для *каждого* из заданных альтернативных вариантов:

1. Составить систему уравнений и ограничений, позволяющую получить заданное распределение ресурсов.

2. Разработать в MS Excel рабочий лист расчета распределения ресурсов и сценарий поиска решения. Выполнить сохранение модели (сценария) поиска решения. Окно поиска решений с заданным сценарием поиска вставить в отчет в виде скриншета.

3. Осуществить поиск решения задачи распределения ресурсов. Выполнить анализ решения.

4. Результаты выполнения работы по пп. 1 – 3 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

Практические рекомендации по выполнению работы и использованию программного обеспечения

Рабочий лист MS Excel представляет собой электронную таблицу, в ячейках которой могут содержаться числа, текст и формулы. Формула MS Excel начинается со знака «=» и представляет собой выражение, содержащее ссылки на ячейки, содержащие числа; операнды функций и знаки арифметических действий. Например, формула вида

$$=A21^2+\text{COS}(B21)$$

осуществит сложение второй степени числа, содержащегося в ячейке A21 и косинуса числа, содержащегося в ячейке B21. При вводе формула отображается в строке формул в верхней части таблицы. Ссылки на ячейки таблицы можно вводить вручную – при этом их следует набирать на латинском регистре, или добавлять в формулу щелчком мыши по соответствующей ячейке. По завершению ввода формулы при нажатии клавиши Enter в ячейке ввода отобразится результат расчета.

При копировании фрагментов таблицы вставка текста и чисел в новом месте осуществляется без изменений, а в формулах автоматически изменяются ссылки. Так, если формулу предыдущего примера из исходной ячейки, например, C21 скопировать в D25, то она изменится следующим образом:

$$=B25^2+\text{COS}(C25)$$

Если при копировании формулы необходимо защитить ссылку от автоматического изменения, перед буквой, и/или номером ячейки в зависимости от того, что требуется оставлять при копировании без изменений, следует поставить знак \$.

Для решения систем уравнений, задач оптимизационного и распределительного типа в MS Excel используют надстройку *Поиск решения* (меню *Сервис – Поиск решения*). Процедура поиска решения состоит в автоматическом определении по заданному сценарию значений влияющих ячеек, которые обеспечивают экстремум или заданное значение зависимой ячейки. Влияющие и целевая (целевые) ячейки должны быть связаны формулой, иначе при изменении первых значения вторых не будут изменяться.

Пример 1

Решение систем линейных уравнений в MS Excel и сохранение модели (сценария) поиска решения

Пусть требуется решить систему уравнений вида:

$$\begin{cases} x_1 + 5x_2 + 3x_3 = 3; \\ -2x_1 + 4x_2 + 2x_3 = 5; \\ 3x_1 - 2x_2 + 6x_3 = 2. \end{cases}$$

В матричном виде система имеет вид:

$$\begin{bmatrix} 1 & 5 & 3 \\ -2 & 4 & 2 \\ 3 & -2 & 6 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 2 \end{bmatrix},$$

где $\begin{bmatrix} 3 \\ 5 \\ 2 \end{bmatrix}$ – вектор-столбец ограничений; $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ – искомый вектор-столбец

аргументов. В MS Excel вектор-столбец аргументов при вводе системы уравнений заменяется произвольными начальными приближениями к искомому решению, например, все аргументы приравниваются к 1 (рис. 89).

	A	B	C	D	E	F	G	H
1								
2		Матрица коэффициентов				Аргументы		Расчет вектора ограничений
3		1	5	3		1		=F\$3*B3+F\$4*C3+F\$5*D3
4		-2	4	2		1		=F\$3*B4+F\$4*C4+F\$5*D4
5		3	-2	6		1		=F\$3*B5+F\$4*C5+F\$5*D5

Рис. 89

Результат расчета по начальным приближениям аргументов, разумеется, не соответствует заданному вектору-столбцу ограничений (рис. 90).

	A	B	C	D	E	F	G	H	I	J
1	Ввод исходной системы с начальными приближениями аргументов x_1-x_3 :									
2		Матрица коэффициентов				Аргументы		Расчет вектора ограничений		
3		1	5	3		1		9		
4		-2	4	2		1		4		
5		3	-2	6		1		7		

Рис. 90

Применим *Поиск решения* для решения системы уравнений. Зададим сценарий расчета, ак показано на рис. 91.

Обратите внимание на то, что значение первого элемента вектора-столбца ограничений, рассчитываемое в ячейке Н3, используется в качестве целевого значения, а равенство заданным значениям двух оставшиеся элементов обеспечивается заданием условий строгого равенства в окне ограничений. По нажатию кнопки *Выполнить* система производит расчет и выдает сообщение о найденном решении (рис. 92).

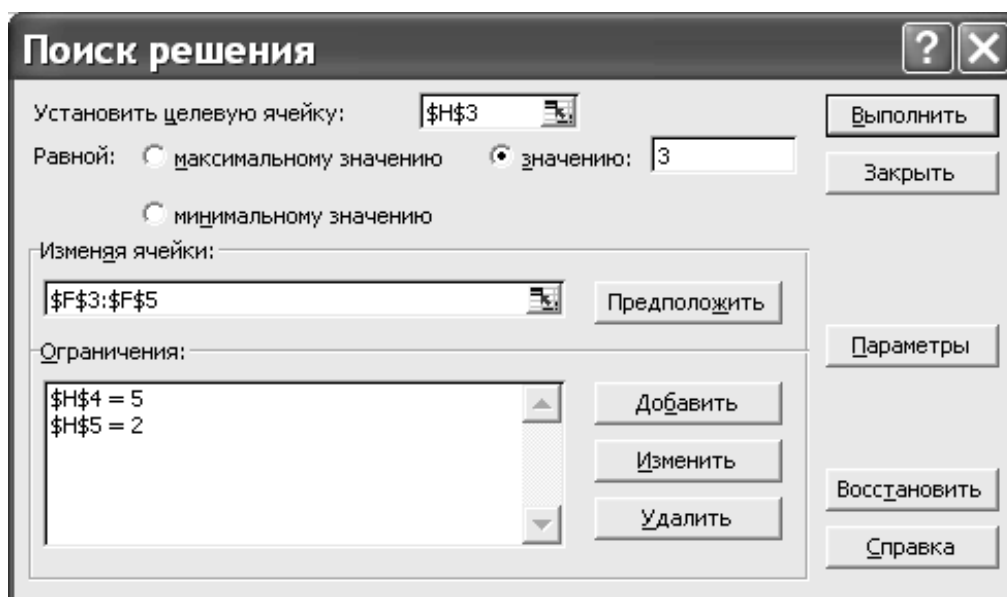


Рис. 91

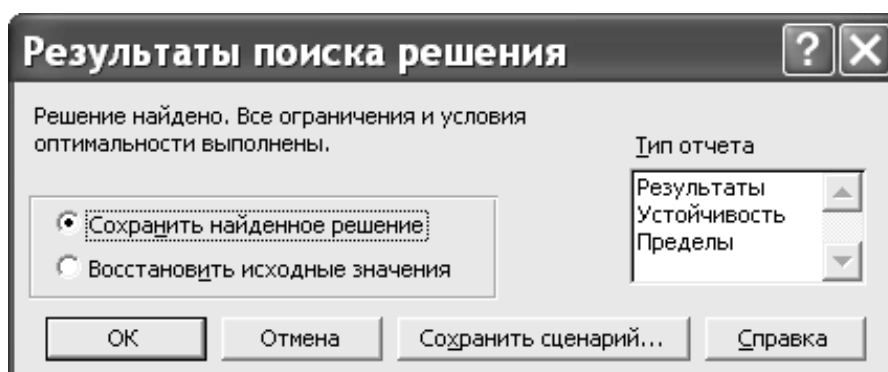


Рис. 92

При выборе *Сохранить найденное решение*, исходная таблица расчета изменится следующим образом (см. файл *Управление ресурсами.xls*, лист *Решение систем уравнений*), как показано на рис. 93.

	A	B	C	D	E	F	G	H	I	J
1										
2		Матрица коэффициентов				Аргументы		Расчет вектора ограничений		
3		1	5	3		-1,063829787		3		
4		-2	4	2		0,244680851		5		
5		3	-2	6		0,946808511		2		
6										

Рис. 93

Значения аргументов $x_1 - x_3$ найдены, система уравнений решена.

В некоторых случаях для упрощения сценария поиска решения удобнее вводить общую целевую ячейку поиска для всех ограничений. Вернемся к исходным значениям начальных приближений вектора-столбца аргументов и дополним исходную таблицу столбцом расчета квадратов отклонений реальных значений ограничений от заданных. Также введем ячейку расчета среднего значения квадратов отклонений (рис. 94).

	A	B	C	D	E	F	G	H	I	J		
23												
24		Матрица коэффициентов				Аргументы		Вектор ограничений				
25								расчет	заданные значения	квадрат отклонения		
26					1		5	3	1	9	3	36
27					-2		4	2	1	4	5	1
28		3	-2	6		1		7	2	25		
29								среднее значение:		20,66666667		
30												

Рис. 94

Теперь, при задании сценария поиска решения в качестве целевой ячейки можно использовать ячейку расчета среднего значения квадратов отклонений: так как при искомым значениях аргументов системы уравнений ограничения должны быть строго равны заданным значениям, то значение целевой ячейки следует задать равным нулю (рис. 95).

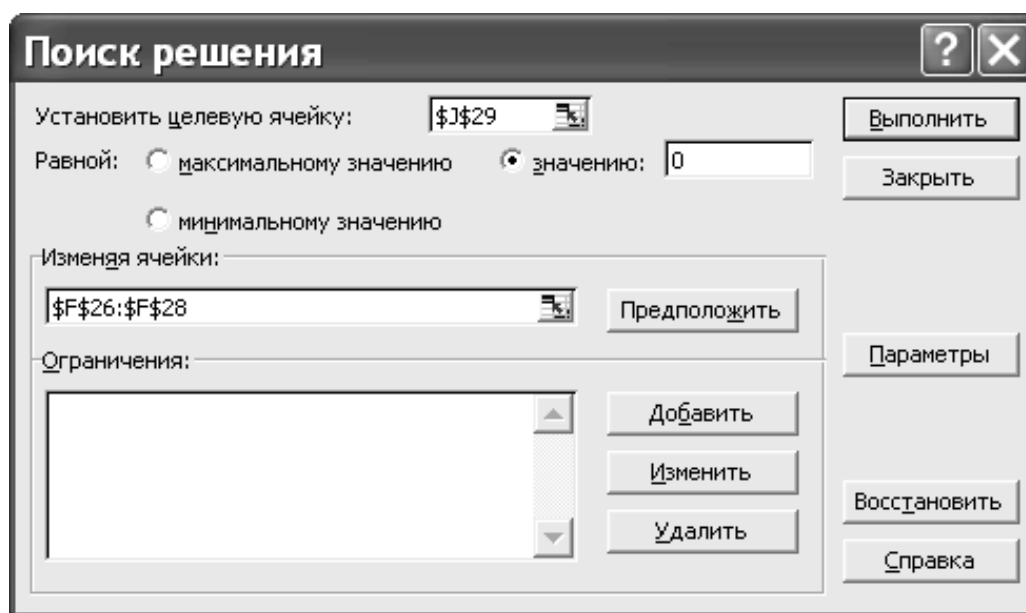


Рис. 95

Результат выполнения поиска решения совпадает с полученным ранее и имеет следующий вид (рис. 96).

	A	B	C	D	E	F	G	H	I	J
23										
24		Матрица коэффициентов				Аргументы		Вектор ограничений		
25								расчет	заданные значения	квадрат отклонения
26								3	3	0
27		1	5	3		-1,06382922		5	5	0
28		-2	4	2		0,244680658		2	2	0
29		3	-2	6		0,946808136		среднее значение:		0

Рис. 96

При каждом сохранении файла MS Excel после использования надстройки *Поиск решения* сохраняются и все значения, введенные перед сохранением в ее окне, т.е. MS Excel автоматически сохраняет последний набор параметров модели (сценария) поиска решения. Если на одном рабочем листе необходимо сохранить несколько наборов параметров сценариев поиска решения, следует в окне *Поиск решения* нажать *Параметры – Сохранить модель* и указать пустую ячейку. Сохраняемые параметры сценария будут вставлены на рабочем листе, начиная с указанной ячейки, и займут вниз несколько смежных ячеек, поэтому, перед сохранением сценария следует убедиться, что ячейки, расположенные ниже заданной, не содержат данных.

Для использования ранее сохраненного сценария следует нажать кнопку *Загрузить модель* в окне *Параметры*, задать выделением весь диапазон параметров модели (сценария поиска решения) и подтвердить изменение диапазона параметров нажатием *ОК* в диалоге загрузки и еще раз, в окне *Параметры*. В результате снова откроется окно *Поиск решения* с загруженным сценарием поиска.

В файле примера *Управление ресурсами.xls* на листах *Решение систем уравнений* диапазоны ячеек с сохраненными параметрами сценариев расчета выделены цветом.

Пример 2

Решение задачи распределения ресурсов системы при заданном приоритете выполнения определенных видов работ

Пусть система осуществляет несколько целенаправленных видов деятельности – выполняет несколько видов работ. Потребность в ресурсах для выполнения единичного объема каждого вида работ, доступные объе-

мы каждого вида ресурсов и планируемый объем выполнения каждого вида работ (число подлежащих выполнению единичных объемов работ) приведены в табл. 15.

Таблица 15

Наименование работ	План выполнения	Потребность в ресурсах для выполнения единичного объема работ			
		Ресурс № 1	Ресурс № 2	Ресурс № 3	Ресурс № 4
Работа №1	10	10	2	5	0
Работа №2	10	0	4	0	12
Работа №3	14	17	15	5	12
Работа №4	2	10	10	10	10
Работа №5	8	6	8	8	0
Доступный объем ресурса		300	300	150	200

Пусть требуется распределить ресурсы по видам выполняемых работ таким образом, чтобы план по работам № 1 и № 2 был выполнен на 100 %, а по остальным видам работ было выполнено максимально возможный равный объем работ в единичных объемах. Пусть также единичный объем является неделимой (целой) величиной.

Предварительный расчет (см. файл *Управление ресурсами.xls*, лист *Распределение ресурсов 1*) показывает, что доступного объема ресурсов для выполнения плановых объемов всех видов работ недостаточно. Поскольку работы № 1 и № 2 должны быть выполнены полностью, то из доступного объема каждого ресурса нужно вычесть объемы, необходимые для покрытия 100 % потребности по приоритетным работам. Распределению для обеспечения выполнения работ № 3 – 5 с учетом заданных ограничений подлежит остаток ресурсов: ресурс № 1 – 200; ресурс № 2 – 240; ресурс № 3 – 100 и ресурс № 4 – 80.

Если x_j – искомая реализация плана по j -тому виду оставшихся не-приоритетных работ ($j=3,4,5$); a_{ij} – количество i -того ресурса, необходимого для выполнения единичного объема j -того вида работ; b_i – доступный остаток i -того ресурса, то распределение ресурсов можно найти, решив систему неравенств вида:

$$\begin{cases} a_{13} \cdot x_3 + a_{14} \cdot x_4 + a_{15} \cdot x_5 \leq b_1; \\ a_{23} \cdot x_3 + a_{24} \cdot x_4 + a_{25} \cdot x_5 \leq b_2; \\ a_{33} \cdot x_3 + a_{34} \cdot x_4 + a_{35} \cdot x_5 \leq b_3; \\ a_{43} \cdot x_3 + a_{44} \cdot x_4 + a_{45} \cdot x_5 \leq b_4, \end{cases} \Rightarrow$$

$$\begin{bmatrix} a_{13} & a_{14} & a_{15} \\ a_{23} & a_{24} & a_{25} \\ a_{33} & a_{34} & a_{35} \\ a_{43} & a_{44} & a_{45} \end{bmatrix} \cdot \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \Rightarrow [A] \cdot [X] \leq [B].$$

Однако составленная система не учитывает дополнительных ограничений распределения, а именно требование выполнения максимально возможного равного объема запланированных неприоритетных видов работ и целочисленности этих объемов, поэтому в сценарии поиска распределения остатков ресурсов следует учесть условия:

$$\begin{cases} x_3, x_4, x_5 \rightarrow \max; \\ x_3 = x_4 = x_5 = \text{целое}. \end{cases}$$

Поскольку в сценарии поиска решений в MS Excel целевой может являться только одна ячейка, условие максимальности искомых реализаций $x_3 - x_5$ можно задать, рассчитав в целевой ячейке их сумму. Остальные условия учтем, задав соответствующие ограничения в сценарии поиска решения. В MS Excel система неравенств вводится как система уравнений, а вектор-столбец искомых реализаций планов $[X]$ заменяется произвольными начальными приближениями к искомым решениям, например, все аргументы приравниваются к 1 (рис. 97).

	A	B	C	D	E
34	[A]				[X]
35					
36	17	10	6		1
37	15	10	8		1
38	5	10	8		1
39	12	10	0	Σ	3

G	H
[B]	
расчет	доступный остаток
=E36*A36+E37*B36+C36*E38	200
=A37*E36+B37*E37+C37*E38	240
=A38*E36+B38*E37+C38*E38	100
=A39*E36+B39*E37+C39*E38	80

Рис. 97

Результат расчета по начальным приближениям аргументов, разумеется, не соответствует заданным ограничениям распределения ресурсов (рис. 98).

	A	B	C	D	E
34	[A]				[X]
35					
36	17	10	6	x_3	1
37	15	10	8	x_4	1
38	5	10	8	x_5	1
39	12	10	0	Σ	3

	G	H	I
	[B]		
	расчет	доступный остаток	неизрасходованный остаток
b_1	33	200	167
b_2	33	240	207
b_3	23	100	77
b_4	22	80	58

Рис. 98

Зададим сценарий расчета распределения доступных остатков ресурсов по неприоритетным видам работ, ак показано на рис. 99, 100 (полный перечень ограничений приведен в двух окнах из-за фиксированного размера окна поиска решений).

Поиск решения

Установить целевую ячейку:

Равной: ☒ максимальному значению ☐ значению:
☐ минимальному значению

Изменяя ячейки:

Ограничения:

Рис. 99

Поиск решения

Установить целевую ячейку:

Равной: ☒ максимальному значению ☐ значению:

☐ минимальному значению

Изменяя ячейки:

Ограничения:

\$E\$37 = целое	Добавить Изменить Удалить
\$E\$38 = целое	
\$G\$36 <= \$H\$36	
\$G\$37 <= \$H\$37	
\$G\$38 <= \$H\$38	
\$G\$39 <= \$H\$39	

Рис. 100

Результат поиска распределения остатков ресурсов по неприоритетным видам работ с учетом всех заданных ограничений имеет вид (рис. 101).

	A	B	C	D	E
34	[A]				[X]
35					
36	17	10	6	x ₃	3
37	15	10	8	x ₄	3
38	5	10	8	x ₅	3
39	12	10	0	Σ	9

F	G	H	I
	[B]		
	расчет	доступный остаток	неизрасходованный остаток
b ₁	99	200	101
b ₂	99	240	141
b ₃	69	100	31
b ₄	66	80	14

Рис. 101

Как следует из полученного варианта расчета, доступного остатка ресурсов достаточно для выполнения трех единичных объемов работ № 3 – 5. Правильность решения можно проверить подстановкой в вектор-столбец

[X] значений, равных 4. Видно, что в этом случае доступного остатка ресурса №4 для выполнения суммарного объема работ уже недостаточно (разница между доступным остатком ресурса № 4 и расчетной величиной потребного потребности в этом ресурсе, выводимая в графе «неизрасходованный остаток», становится отрицательной) (рис. 102).

34	[A]				[X]
35					
36	17	10	6	x_3	4
37	15	10	8	x_4	4
38	5	10	8	x_5	4
39	12	10	0	Σ	12

	[B]		
	расчет	доступный остаток	неизрасходованный остаток
b_1	132	200	68
b_2	132	240	108
b_3	92	100	8
b_4	88	80	-8

Рис. 102

Однако найденное решение по распределению доступного остатка ресурсов для выполнения максимально возможного равного объема не-приоритетных работ в размере трех единичных объемов не соответствует исходному плану выполнения работ: согласно нему плановый объем выполнения работы № 4 равен 2.

Следовательно, окончательный оптимальный вариант распределения ресурсов обеспечивается при выполнении в соответствии с условиями распределения на 100 % объема плановых задания по приоритетным работам № 1 и № 2, а также выполнения в максимально возможном (равном 2) объеме не-приоритетных работ № 3 – 5.

Пример 3

Решение задачи распределение ресурсов системы при наличии дополнительных условий и ограничений распределения

Выполним распределение ресурсов системы по видам выполняемых работ для исходных данных из примера 2 с учетом следующих условий

распределения: необходимо выполнить на 100 % максимально возможное число работ, а остаток ресурсов распределить между оставшимися работами пропорционально плану.

Данную задачу можно решить по частям. Сначала определим, какие виды работ можно с учетом ограниченного объема ресурсов выполнить на 100 % при условии выполнения максимального возможного числа таких работ. Введем в рассмотрение коэффициент k_j , равный доле выполнения плана по j -тому виду работ. В решении первой части задачи этот k_j может иметь значение 1 или 0, причем сумма коэффициентов должна быть максимально возможной. Если x_j – заданная реализация плана по j -тому виду работ ($j = 1, 2, \dots, 5$); a_{ij} – количество i -того ресурса, необходимого для выполнения единичного объема j -того вида работ; b_i – имеющийся объем i -того ресурса, то распределение ресурсов можно найти, решив относительно k_j систему неравенств вида:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 + a_{14} \cdot x_4 + a_{15} \cdot x_5 \leq b_1; \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + a_{23} \cdot x_3 + a_{24} \cdot x_4 + a_{25} \cdot x_5 \leq b_2; \\ a_{31} \cdot x_1 + a_{32} \cdot x_2 + a_{33} \cdot x_3 + a_{34} \cdot x_4 + a_{35} \cdot x_5 \leq b_3; \\ a_{41} \cdot x_1 + a_{42} \cdot x_2 + a_{43} \cdot x_3 + a_{44} \cdot x_4 + a_{45} \cdot x_5 \leq b_4, \end{cases} \Rightarrow$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{bmatrix} \cdot \begin{bmatrix} k_1 \cdot x_1 \\ k_2 \cdot x_2 \\ k_3 \cdot x_3 \\ k_4 \cdot x_4 \\ k_5 \cdot x_5 \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix},$$

с учетом следующих ограничений:

$$\begin{cases} k_j \in \{0,1\}; \\ \sum_{j=1}^5 k_j \rightarrow \max. \end{cases}$$

В MS Excel система неравенств вводится как система уравнений, а искомые значения коэффициентов k_j задаются начальными приближениями, например, все аргументы приравниваются к 0. Для возможности последующего выполнения поиска решения с учетом заданных ограничений в таблицу дополнительно вводится ячейка суммы всех значений k_j (рис. 103).

Зададим сценарий расчета распределения имеющихся в наличии ресурсов по видам работ для обеспечения максимально возможного числа

работ, выполненных на 100 %, как показано на рис. 104, 105 (полный перечень ограничений приведен в двух окнах из-за фиксированного размера окна поиска решений).

	A	B	C	D	E	F	G
12	[A]					[X]	k_j
13							
14	10	0	17	10	6	10	0
15	2	4	15	10	8	10	0
16	5	0	5	10	8	14	0
17	0	12	12	10	0	2	0
18						8	0
19						Σ	0

Н							I
[B]							
расчет							наличие
=F\$14*\$G\$14*A14+F\$15*\$G\$15*B14+C14*\$F\$16*\$G\$16+F\$17*\$G\$17*D14+F\$18*\$G\$18*E14							300
=F\$14*\$G\$14*A15+F\$15*\$G\$15*B15+C15*\$F\$16*\$G\$16+F\$17*\$G\$17*D15+F\$18*\$G\$18*E15							300
=F\$14*\$G\$14*A16+F\$15*\$G\$15*B16+C16*\$F\$16*\$G\$16+F\$17*\$G\$17*D16+F\$18*\$G\$18*E16							150
=F\$14*\$G\$14*A17+F\$15*\$G\$15*B17+C17*\$F\$16*\$G\$16+F\$17*\$G\$17*D17+F\$18*\$G\$18*E17							200

Рис. 103

Поиск решения

Установить целевую ячейку:

Равной: ☒ максимальному значению ☐ значению:

☐ минимальному значению

Изменяя ячейки:

Ограничения:

\$G\$14 = двоичное	▲ ▼	Добавить Изменить Удалить
\$G\$15 = двоичное		
\$G\$16 = двоичное		
\$G\$17 = двоичное		
\$G\$18 = двоичное		
\$H\$14 <= \$I\$14		

Рис. 104

Поиск решения

Установить целевую ячейку:

Равной: ☒ максимальному значению ☐ значению:

☐ минимальному значению

Изменяя ячейки:

Ограничения:

\$G\$17 = двоичное	▲	Добавить Изменить Удалить
\$G\$18 = двоичное		
\$H\$14 <= \$I\$14		
\$H\$15 <= \$I\$15		
\$H\$16 <= \$I\$16		
\$H\$17 <= \$I\$17	▼	

Рис. 105

Результат поиска распределения ресурсов по видам работ с учетом всех заданных ограничений показан на рис. 106 (см. файл *Управление ресурсами.xls*, лист *Распределение ресурсов 2*).

	A	B	C	D	E	F	G	H	I	J
12	[A]					[X]	k_j	[B]		
13								расчет	наличие	неизрасходованный остаток
14	10	0	17	10	6	10	1	168	300	132
15	2	4	15	10	8	10	1	144	300	156
16	5	0	5	10	8	14	0	134	150	16
17	0	12	12	10	0	2	1	140	200	60
18						8	1			
19						Σ	4			

Рис. 106

Как следует из результатов расчета, наличных ресурсов хватит на полное выполнение планов по четырем видам работ: № 1, № 2, № 4 и № 5 – расчетные значения k_j для них оказались равными 1. Следовательно, постановка второй части задачи распределения требует изменения: из-за того, что в результате первоначального распределения не выполненной оказалась только работа № 3, весь неизрасходованный остаток

ресурсов следует распределить для обеспечения максимально возможного объема выполнения плана по этому виду работы. Зададим таблицу расчета распределения оставшихся ресурсов с заданием начального значения расчетного объема выполнения плана по 3-му виду работ, равным 1 (рис. 107).

	A	B	C	E
40	[A]	[B]		x_3 (расчет)
41		расчет	наличие	
42	17	=A42*\$E\$42	132	1
43	15	=A43*\$E\$42	156	
44	5	=A44*\$E\$42	16	
45	12	=A45*\$E\$42	60	

Рис. 107

Зададим сценарий расчета с учетом ограничения по целочисленности значения объема выполнения плана (рис. 108).

Поиск решения

Установить целевую ячейку:

Равной: ☒ максимальному значению ☐ значению: ☐ минимальному значению

Изменяя ячейки:

Ограничения:

-
-
-
-
-
-

Рис. 108

Результат выполнения поиска решения имеет вид, приведенный на рис. 109 (см. файл *Управление ресурсами.xls*, лист *Распределение ресурсов 2*).

	A	B	C	D	E
40	[A]	[B]			x_3 (расчет)
41		расчет	наличие	остаток	
42	17	51	132	81	3
43	15	45	156	111	
44	5	15	16	1	
45	12	36	60	24	

Рис. 109

Таким образом, при заданных условиях распределения ресурсов, после полного обеспечения потребности в ресурсах работ №1, №2, №4 и №5 из 14 плановых объемов выполнения работы №3 остаточных ресурсов хватит на выполнение только 3-х объемов. Видно, что лимитирующим является ресурс №4: при значительной величине остатков по остальным ресурсам, остаток ресурса №4, равный 1, больше не позволяет обеспечить потребность ни одного единичного объема выполнения работы №3.

Пример 4

Использование формул массива для выполнения операций над матрицами коэффициентов в MS Excel

При решении задач распределения ресурсов в процессе составления систем неравенств с ограничениями распределения и при анализе результатов распределения часто требуется выполнение действий над матрицами коэффициентов (значений a_{ij}), векторами ограничений (значений b_i) и т. д. К таким действиям относятся:

- транспонирование матрицы;
- умножение матрицы на вектор-столбец;
- умножение матрицы на постоянный коэффициент.

Для выполнения этих действий можно использовать формулы массива MS Excel. Правила создания и использования формул массива:

- при создании формулы массива выделяется не ячейка, а диапазон результатов вычислений;

- для вычисления результатов по формуле массива используется комбинация клавиш *Ctrl+Shift+Enter*. После нажатия комбинации MS Excel заключает формулу в фигурные скобки, которые указывают на то, что формула является формулой массива. **ВАЖНО:** не вводите фигурные скобки вручную! MS Excel воспримет такой ввод как текстовое значение;

- в диапазоне массива нельзя редактировать отдельные ячейки, так как диапазон массива является единым объектом. Для редактирования массив необходимо выделить целиком и щелкнуть в строке редактирова-

ния формул (фигурные скобки вокруг формулы при этом исчезнут). После редактирования нужно вновь завершить ввод нажатием *Ctrl+Shift+Enter*;

- для быстрого выделения массива щелкните на любой из его ячеек и нажмите комбинацию *Ctrl+|* (клавишу «|» нажимайте на цифровой клавиатуре).

Транспонированной называется матрица, строками которой являются столбцы исходной матрицы. Для выполнения операции транспонирования в MS Excel используется функция ТРАНСП. Функция ТРАНСП должна быть введена как формула массива в диапазон, который имеет столько же строк и столбцов, сколько столбцов и строк имеет аргумент массив – исходная матрица. Введем произвольную матрицу коэффициентов в MS Excel размером 4×3, выделим диапазон ячеек 3×4 и введем формулу транспонирования (рис. 110).

	A	B	C	D	E	F	G	H
3		[A]				[A ^T]		
4	1	5	9					
5	2	6	10					
6	3	7	11					
7	4	8	12					

	A	B	C	D	E	F	G	H
3		[A]				[A ^T]		
4	1	5	9			=ТРАНСП(A4:C7)		
5	2	6	10					
6	3	7	11					
7	4	8	12					

Рис. 110

После завершения ввода необходимо нажать *Ctrl+Shift+Enter*. Результат транспонирования исходной матрицы имеет вид (см. файл *Управление ресурсами.xls*, лист *Использование формул массива*) (рис. 111). Обратите внимание, что после вычисления в любой из ячеек массива формула имеет вид {=ТРАНСП(A4:C7)}.

	A	B	C	D	E	F	G	H
3		[A]				[A ^T]		
4	1	5	9		1	2	3	4
5	2	6	10		5	6	7	8
6	3	7	11		9	10	11	12
7	4	8	12					

Рис. 111

Вторым способом транспонирования матрицы в MS Excel является использование специального режима транспонирования при вставке фрагмента, скопированного в буфер обмена. Скопируем подлежащую транспонированию матрицу в буфер обмена, выдели произвольную ячейку, вправо и вниз от которой будет осуществлена вставка транспонированной матри-

цы, и нажмем правую кнопку мыши. В окне диалога *Специальная вставка* следует выбрать опцию «транспонировать» и нажать ОК (рис. 112).

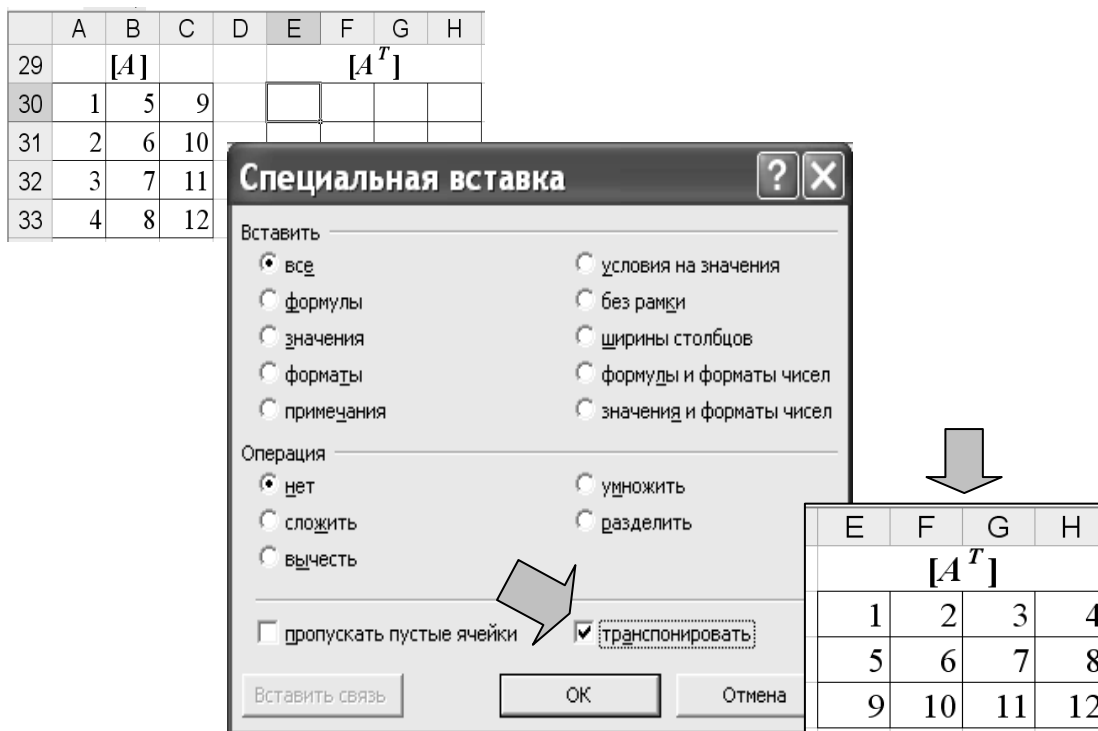


Рис. 112

Умножение матрицы на постоянный коэффициент может потребоваться при вычислении долевой потребности в ресурсах, например, определение потребностей по всем видам ресурсов, обеспечивающих 80% выполнения заказа и т. п. Для реализации такого расчета вводится формула массива следующего вида, для вычисления которой необходимо нажать *Ctrl+Shift+Enter* (рис. 113).

	A	B	C	D	E	F	G	H	I	J
19		[A]			k			k·[A]		
20	1	2	3		0,8			=(A20:C23)*E20		
21	4	5	6							
22	7	8	9							
23	10	11	12							

	A	B	C	D	E	F	G	H	I
19		[A]			k			k·[A]	
20	1	2	3		0,8		0,8	1,6	2,4
21	4	5	6				3,2	4	4,8
22	7	8	9				5,6	6,4	7,2
23	10	11	12				8	8,8	9,6

Рис. 113

Варианты индивидуальных заданий

1. Модернизация системы связана с выполнением работ по нескольким договорам подряда, каждый из которых требует использования нескольких типов расходных материалов. Разработать варианты распределения ресурсов в соответствии со следующими условиями: 1) максимальным приоритетом по выполнению обладает договор № 3, договор № 2 необходимо выполнить как минимум на 50 %, а остальные – по возможности; 2) все расходные материалы должны быть распределены по договорам пропорционально соотношению потребностей лимитирующего ресурса.

Расходные материалы	Наличие	Потребность по договору			
		№ 1	№ 2	№ 3	№ 4
тип 1	70	25	14	18	19
тип 2	15	1	8	8	0
тип 3	90	14	21	21	27
тип 4	75	15	12	12	45
тип 5	85	15	16	14	50
тип 6	30	5	10	5	10
тип 7	4	2	2	2	0

2. На модернизацию системы, состоящую в выполнении четырех видов работ, выделен ограниченный объем финансирования: 16 000 у.е. Разработать варианты распределения финансовых средств в соответствии со следующими условиями: 1) финансирование должно быть распределено пропорционально соотношению суммарных стоимостей выполнения 100 % объема работ по видам; 2) в первую очередь завершению подлежит вид работ, выполненных более чем на 50 %.

Ресурсы	Стоимость единичного объема ресурса, у.е.	Потребность по видам работ (до завершения)			
		№ 1	№ 2	№ 3	№ 4
тип 1	25	45	80	41	50
тип 2	8,24	19	24	26	26
тип 3	140	20	20	20	20
тип 4	10	12	15	15	10
тип 5	9,95	10	10	10	50
тип 6	8,2	36	36	14	18
тип 7	4,5	0	0	21	20
Процент выполнения работ		10%	25%	55%	25%

3. В системе параллельно осуществляется пять процессов с различной потребностью в расходовании шести типов ресурсов. Разработать варианты планов осуществления максимально возможного числа процессов в системе в соответствии со следующими условиями: 1) минимум остатков неизрасходованных ресурсов типов № 5 и № 2; 2) расходование ресурсов должно осуществляться пропорционально начальному наличию.

Процесс	Потребность в ресурсах					
	№ 1	№ 2	№ 3	№ 4	№ 5	№ 6
№ 1	15	2	0	6	21	2
№ 2	12	4	5	0	0	12
№ 3	23	2	2	2	4	8
№ 4	2	2	4	13	10	6
№ 5	0	0	7	2	2	2
Наличие ресурсов	800	130	200	250	350	380

4. В системе выполняется пять видов работ, причем работы 1 – 3 имеют альтернативные варианты расходования ресурсов. Распределить имеющиеся в наличие ресурсы с использованием альтернативных вариантов расходования ресурсов по приоритету максимально возможного выполнения плана с учетом целочисленности показателей плана.

Вид работ	План выполнения	Потребность в ресурсах для выполнения единичного объема работ					
		1	2	3	4	5	6
1.1	50	15	16	5	2	55	58
1.2		10	12	10	0	60	40
2.1	50	21	20	10	3	3	40
2.2		15	35	5	0	6	52
3.1	20	12	10	5	5	10	12
3.2		8	15	2	2	8	15
4	20	3	0	10	10	15	40
5	100	0	0	15	15	25	10
Наличие ресурса		1800	2100	2300	1500	5300	6200

5. В предыдущем месяце в производственной системе допущен перерасход электроэнергии на обеспечение технологических процессов. Скорректировать план текущего месяца в соответствии со следующими условиями: 1) пропорциональное (в процентах) уменьшение плана выпуска по процессу до совпадения расхода с планом; 2) пропорциональное снижение затрат электроэнергии по статьям процессов № 5 и № 6 без изменения пла-

на выпуска при максимально возможном равном объеме выполнения плана по остальным процессам.

Процесс	План выполнения	Расход электроэнергии на единичное осуществление процесса, по статьям			
		№ 1	№ 2	№ 3	№ 4
№ 1	100	25	10	2	10
№ 2	100	10	25	5	5
№ 3	150	20	10	2	0
№ 4	100	10	35	20	2,5
№ 5	180	5	5	5	1,8
№ 6	200	14,2	4	4	5,2
По плану, всего		10 000	9500	4500	2500

6. Разработать варианты распределения расчетной нагрузки сотрудников предприятия по видам работ с учетом ограничений по количеству рабочих часов в соответствии со следующими условиями: 1) максимально возможная равная загрузка каждого сотрудника по каждому виду работ; 2) предпочтительное выполнение работы, связанной с клиентами (работа в секциях и дежурство в офисе).

Виды работ	Расчетная нагрузка сотрудника, часов в неделю						
	№ 1	№ 2	№ 3	№ 4	№ 5	№ 6	№ 7
Секция № 1	6	10	5	8	8	12	12
Секция № 2	6	10	5	6	8	7	12
Секция № 3	6	10	5	8	8	20	6
Дежурство в офисе	10	6	6	9	8	0	6
Дежурство на производстве	5	6	6	5	4	4	6
Дежурство на складе	5	6	6	5	4	2	6
Ограничение нагрузки, не более	36	36	30	36	36	34	36

7. Модернизация восьми подсистем системы требует установки нового оборудования и использования расходных материалов. Разработать варианты распределения оборудования и расходных материалов в соответствии со следующими условиями: 1) оборудование и расходные материалы должны быть распределены пропорционально суммарной стоимости полной модернизации подсистем; 2) модернизация должна быть завершена на 100 % для максимального числа подсистем.

Подсистема	Оборудование, шт.			Расходные материалы		
	тип 1	тип 2	тип 3	тип А	тип Б	тип В
№ 1	10	52	10	100	150	25
№ 2	12	50	12	120	150	25
№ 3	12	40	14	100	144	25
№ 4	5	0	16	120	128	30
№ 5	5	14	16	100	140	25,5
№ 6	4	15	16	100	152	16
№ 7	7	16	16	100	160	8,5
№ 8	12	36	22	280	300	0
Стоимость за единицу, у.е.	1230	550	850	30	16	85
Наличие оборудования и материалов	60	200	110	1000	1200	140

8. Расходные материалы распределяются по четырем видам работ. Разработать варианты распределения расходных материалов в соответствии со следующими условиями: 1) распределение должно быть выполнено пропорционально размерам заказов по видам работ; 2) распределение должно быть выполнено пропорционально наличию расходных материалов по наименованиям.

Расходные материалы	Заказы материалов по видам работ				Наличие
	1	2	3	4	
Наименование 1	15	16	14	19	58
Наименование 2	22	32	45	12	100
Наименование 3	8	9	17	15	45
Наименование 4	14	24	21	22	70
Наименование 5	23	21	21	20	80
Наименование 6	100	115	89	55	400
Наименование 7	45	12	17	18	85
Наименование 8	32	35	35	30	100

9. Укомплектовать заказы на поставку ресурсов в соответствии с наличием наименований по следующим условиям: 1) максимально возможное равное комплектование заказов; 2) максимально возможное число заказов, укомплектованных полностью, при допустимом минимуме укомплектования неукомплектованных заказов, равном 30 %.

Тип заказа	План выполнения заказов	Состав заказов по наименованиям ресурсов, шт.				
		№ 1	№ 2	№ 3	№ 4	№ 5
Заказ 1	10	15	16	14	17	22
Заказ 2	10	2	4	5	6	6
Заказ 3	25	25	25	24	20	6
Заказ 4	20	34	31	5	5	5
Заказ 5	40	10	10	10	2	4
Заказ 6	30	40	40	10	10	10
Наличие ресурса		3100	3000	1500	1200	1000

10. Завершение модернизации восьми подсистем системы требует установки нового оборудования и использования расходных материалов. Разработать варианты распределения оборудования и расходных материалов в соответствии со следующими условиями: 1) оборудование и расходные материалы должны быть распределены для обеспечения максимально возможной равной степени завершения модернизации; 2) модернизация должна быть выполнена в максимально возможном объеме в подсистемах со степенью завершения равной 25 %.

Подсистема	Оборудование, штук			Расходные материалы			Степень завершения
	тип 1	тип 2	тип 3	тип А	тип Б	тип В	
№ 1	10	52	10	100	150	25	25%
№ 2	12	50	12	120	150	28,5	25%
№ 3	12	40	14	100	144	25	25%
№ 4	5	0	16	120	128	30	35%
№ 5	5	14	16	100	140	24	25%
№ 6	4	15	16	100	152	18,5	75%
Наличие ресурсов	30	150	100	500	600	100	

11. Выполнение плана по заказам требует расходования пяти видов ресурсов. Разработать варианты распределения ресурсов в соответствии со следующими условиями: 1) приоритет выполнения заказов убывает с увеличением общей стоимости заказа; 2) в первую очередь должны быть по возможности выполнены три заказа, суммарная стоимость которых максимальна, а остаточные ресурсы должны быть распределены пропорционально плану.

Заказ	Ресурсы					План по заказам
	№ 1	№ 2	№ 3	№ 4	№ 5	
№ 1	10	2	12	1	15	50
№ 2	12	35	13	2	15	50
№ 3	12	3	12	1	15	45
№ 4	12	6	45	2	24	45
№ 5	12	4	68	1	22	62
№ 6	20	18	36	3	21	20
Цена за единицу, у.е.	150	350	50	1000	25	
Наличие ресурса	3000	3000	8500	500	5100	

12. В системе параллельно осуществляется пять видов работ с различной потребностью в расходовании пяти типов ресурсов. Разработать варианты распределения ресурсов в соответствии со следующим условиями: 1) максимально возможный равный объем выполнения плана; 2) максимально возможное пропорциональное выполнение плана работ № 1– 3 и № 5 при допуском максимуме выполнения плана по работе № 4, равном 25 %.

Вид работ	План выполнения	Ресурсы				
		№ 1	№ 2	№ 3	№ 4	№ 5
№ 1	10	10	6	2	30	8
№ 2	15	10	6	2	30	10
№ 3	15	10	4	8	30	10
№ 4	12	15	2	20	50	10
№ 5	8	18	0	20	45	12
Наличие ресурса		550	100	450	1300	400

13. Освоение новых видов работ в системе требует дополнительного финансирования. Выделенный лимит финансирования составляет 3,8 тыс. у.е. Разработать варианты распределения финансирования в соответствии со следующими условиями: 1) финансирование распределяется пропорционально суммарной стоимости освоения работ по видам, при обязательном условии освоения наименее затратного вида работ не менее, чем на 80 % и наиболее затратного вида работ не более, чем на 35 %; 2) приоритет освоения видов работ убывает с возрастанием порядкового номера при обязательном условии обеспечения работ финансированием как минимум на 20 %.

Вид работ	Статьи затрат (% от суммарной стоимости)				Суммарная стоимость освоения, тыс. у.е.
	внедрение	сертификация	опытный прогон	расходные материалы	
№ 1	60,2	10	12	17,8	0,52
№ 2	55	10	13,4	21,6	0,89
№ 3	45,9	5	15	34,1	1,5
№ 4	70	15	11	4	1,1
№ 5	23	44	8	25	0,8

14. Завершение модернизации системы требует выполнения нескольких видов работ с расходом ресурсов двух типов: материалов и оборудования. Разработать варианты распределения оборудования и расходных материалов в соответствии со следующими условиями: 1) максимально возможное равное обеспечение ресурсами всех видов работ при условии расхождения в обеспечении одноименными ресурсами и оборудованием не более, чем на 10 %; 2) первоочередное максимально возможное обеспечение ресурсами работ, выполненных не менее, чем на 25 % и не более, чем на 55 %.

Вид работ	Степень завершения модернизации	Требуемые ресурсы					
		Материалы			Оборудование, штук		
		тип А	тип В	тип С	А	В	С
№ 1	10 %	20	12,2	56	20	3	4
№ 2	25 %	52	20	58	20	5	8
№ 3	50 %	36	20	58	23	5	9
№ 4	55 %	38,5	31	59	23	6	12
№ 5	75 %	60	10	45	25	0	10
№ 6	5 %	120	45	150	55	10	10
Наличие ресурсов		300	115	408	150	20	38

15. В системе параллельно осуществляется пяти видов работ с различной потребностью в расходовании четырех типов ресурсов. Разработать варианты распределения ресурсов по видам работ в соответствии со следующим условиями: 1) максимально возможное удовлетворение потребности в ресурсах по видам работ пропорционально потребности в третьем ресурсе; 2) минимум суммарного остатка неизрасходованных ресурсов 2 и 4.

Вид работ	План выполнения	Требуемые ресурсы			
		Ресурс 1	Ресурс 2	Ресурс 3	Ресурс 4
№ 1	5	25	68	91	10
№ 2	5	0	12	10	3
№ 3	2	12	10	6	10
№ 4	1	50	0	45	30
№ 5	4	2	2	2	2
Наличие ресурса		60	100	120	35

16. В системе параллельно осуществляется шесть видов работ с различной потребностью в расходовании пяти типов ресурсов. Разработать варианты распределения ресурсов по видам работ в соответствии со следующим условиями: 1) обеспечение минимальных остатков неизрасходованных ресурсов; 2) выполнение на 92 % максимально возможного числа видов работ.

Вид работ	План выполнения	Требуемые ресурсы				
		№ 1	№ 2	№ 3	№ 4	№ 5
№ 1	25	2	3	4	5	6
№ 2	25	1	1	2	3	4
№ 3	20	3	4	2	7	8
№ 4	10	2	4	3	5	6
№ 5	25	2	3	2	2	2
№ 6	32	3	4	5	2	0
Наличие ресурса		275	400	425	620	550

17. Выполнение пяти видов работ в системе требует расходования ресурсов двух типов: материалов и оборудования. Разработать варианты распределения оборудования и расходных материалов в соответствии со следующими условиями: 1) распределение ресурсов пропорционально плану при условии обязательного выполнения работ № 2 и № 4 на 80 % от плана; 2) выполнение планов на 100 % по максимально возможному числу работ при возможности дополнительного расходования на три наиболее ресурсоемких вида работ до 24 тыс. у.е.

Вид работ	План выполнения	Требуемые ресурсы					
		Материалы			Оборудование, шт.		
		тип 1	тип 2	тип 3	А	В	С
№ 1	10	45	12	14	2	9	10
№ 2	12	18	15	18	5	8	10
№ 3	15	12	16	15	6	7	8
№ 4	10	15	14	12	3	15	9
№ 5	8	14	10	13	9	12	15
Наличие ресурсов		650	480	440	100	250	280
Стоимость за единицу, у.е.		25	45	60	850	1200	980

18. В системе параллельно осуществляется пять видов работ с различной потребностью в расходовании пяти типов ресурсов на выполнение единичных объемов. Разработать варианты распределения ресурсов по видам работ в соответствии со следующим условиями: 1) выполнение максимально возможного равного числа единичных объемов всех видов работ при обязательном сохранении в неиспользованном остатке не менее 20 % от начального объема всех ресурсов; 2) выполнение максимально возможного числа единичных объемов всех видов работ при условии первоочередного распределения ресурсов типов А и В на работы № 1 и 2, а ресурсов типа Б и Г – на работу № 3.

Вид работ	Ресурсы				
	Тип А	Тип Б	Тип В	Тип Г	Тип Д
№ 1	180	25	52	64	100
№ 2	222	35	63	64	100
№ 3	60	36	41	64	55
№ 4	60	35	85	80	55
№ 5	200	35	69	100	80
Наличие ресурса	650	120	250	300	245

19. В системе параллельно осуществляется пять видов работ с различной потребностью в комплектах четырех типов ресурсов на выполнение единичных объемов работ. Разработать по возможности равномерное распределение ресурсов по комплектам. Разработать комплектование видов работ пропорционально потребностям в комплектах по видам, указанным в соответствующих строках таблицы видов работ, при условии максимально полного укомплектования первых трех видов.

Вид работ	Потребность в комплектах				Состав комплектов	Ресурсы			
	№ 1	№ 2	№ 3	№ 4		тип А	тип Б	тип В	тип Г
№ 1	5	2	2	10	Комплект № 1	25	480	60	19
№ 2	5	3	2	11	Комплект № 2	25	340	50	19
№ 3	5	2	1	12	Комплект № 3	25	300	55	20
№ 4	12	4	3	11	Комплект № 4	10	120	43	25
№ 5	2	3	0	9	Наличие	1300	24000	4000	1800

20. В системе параллельно осуществляется пять видов работ с различной потребностью в затратах по четырем статьям. Общий объем финансирования ограничен 18 000 у.е. Разработать варианты распределе-

ния выделенного объема финансирования в соответствии со следующим условиями: 1) распределение выделенного объема финансирования пропорционально суммарным затратам по видам работ № 2– 4 при условии выделения не менее 20 % необходимого финансирования работы № 1 и 80 % финансирования работы № 5; 2) распределение выделенного объема финансирования по видам работ пропорционально потребностям по статьям с приоритетом, возрастающим в порядке перечисления видов работ, при обязательном минимуме финансирования неприоритетных работ не менее 30 %.

Вид работ	Плановый объем затрат по статьям, у. е.			
	№1	№2	№3	№4
№1	800	200	1500	600
№2	500	150	6000	120
№3	2500	900	5000	500
№4	3600	100	1000	0
№5	800	150	200	500

21. В системе параллельно осуществляется шесть видов работ с различной потребностью в затратах на расходные материалы шести типов. Общий объем финансирования, выделенный на закупку расходных материалов, ограничен 60 500 у. е. Разработать варианты распределения выделенного объема финансирования для выполнения объемов работ в соответствии со следующими условиями: 1) приоритетное выполнение работ с максимальной прибылью на единицу затрат при обязательном соотношении объемов выполнения наиболее приоритетной работы к наименее приоритетной, равному пять к одному; 2) обеспечение максимально возможного равного объема выполнения работ.

Вид работ	Плановый объем затрат по типам расходных материалов, у.е.						Прибыль от выполнения единичного объема работ, у.е.
	1	2	3	4	5	6	
№ 1	40	50	50	40	35	20	100
№ 2	100	100	120	34	89	100	120
№ 3	35	40	50	50	62	100	120
№ 4	23	34	45	56	60	60	115
№ 5	45	45	50	55	55	45	120
№ 6	120	120	110	110	100	100	400

22. Расходные материалы распределяются по четырем видам работ. Разработать варианты распределения расходных материалов в соответствии со следующими условиями: 1) минимум неизрасходованных остатков; 2) по возможности полное выполнение максимального числа заказов при условии обязательного обеспечения всех заказов не менее, чем на 35 %.

Расходные материалы	Заказы материалов по видам работ				Наличие
	1	2	3	4	
Наименование 1	15	16	14	19	58
Наименование 2	22	32	45	12	100
Наименование 3	8	9	17	15	45
Наименование 4	14	24	21	22	70
Наименование 5	23	21	21	20	80
Наименование 6	100	115	89	55	400
Наименование 7	45	12	17	18	85
Наименование 8	32	35	35	30	100

23. В системе параллельно осуществляется шесть видов работ с различной потребностью в расходовании пяти типов ресурсов на выполнение единичных объемов. Разработать варианты распределения ресурсов по видам работ в соответствии со следующим условиями: 1) максимально возможное расходование ресурсов № 4 и № 5; 2) максимально возможное пропорциональное выполнение плана с учетом возможности дополнительных затрат в объеме 1000 у.е. на закупку дополнительного объема ресурсов (цена за единицу ресурса типов № 1 – 3 равна 6 у.е., типов № 4, 5 – 10 у.е.).

Вид работ	План выполнения	Требуемые ресурсы				
		№ 1	№ 2	№ 3	№ 4	№ 5
№ 1	10	15	4	7	1	0
№ 2	10	9	2	2	16	11
№ 3	14	10	2	15	11	0
№ 4	2	10	5	5	5	0
№ 5	8	6	8	8	0	10
№ 6	9	5	4	8	2	2
Наличие ресурса		350	100	300	120	300

24. В системе параллельно осуществляется шесть видов работ с различной потребностью в расходовании четырех типов ресурсов на выпол-

нение единичных объемов. Помимо потребностей в ресурсах выполнение единичного объема каждой работы требует определенных затрат. Разработать варианты распределения ресурсов по видам работ при ограничении на общие затраты по выполнению работ (1000 у.е.) в соответствии со следующим условиями: 1) максимально возможное расходование ресурсов пропорционально наличию; 2) выполнение максимально возможного равного числа единичных объемов работ № 1 – 3 при обязательном сохранении в неиспользованном остатке не менее 10 % от начального объема всех ресурсов.

Вид работ	План выполнения	Затраты на выполнение единичного объема работы, у.е.	Требуемые ресурсы			
			№ 1	№ 2	№ 3	№ 4
тип 1	25	10	45	80	41	50
тип 2	20	15	19	24	26	26
тип 3	24	8	20	20	20	20
тип 4	17	8	12	15	15	10
тип 5	15	10	10	10	10	50
тип 6	25	15	36	36	14	18
Наличие ресурса			2400	2100	2600	2850

Лабораторная работа № 3

Декомпозиция информационных систем с использованием теории графов

Цель: идентификация подсистем в структуре информационной системы путем формальной декомпозиции (матричным методом разбиения графа на максимальные сильно связные подграфы); оценка структурных свойств системы в целом и выделенных подсистем.

Программное обеспечение: MS Excel.

Примеры выполнения работы: файл *Пример декомпозиции структуры ИС.xls*.

Теория

Процесс последовательного деления системы на образующие ее подсистемы и элементы называется *декомпозицией*. Для решения задач декомпозиции используют топологические модели систем, получаемые методами теории графов.

Основные понятия теории графов, способы их задания и правила построения матрицы смежности приведены в разделе 2.3.1 *Основные понятия и способы задания графов*. Реализация формальной декомпозиции структуры системы с использованием матричного метода теории графов рассмотрена в разделе 2.3.2 *Создание и анализ топологических моделей систем* (см. начиная с рис. 32 и далее).

Формальные методы декомпозиции строятся на идентификации подсистем путем разбиения графа на максимальные сильно связные подграфы. Орграф называется *сильно связным*, если для двух любых различных его узлов x_i и x_j существует, по крайней мере, один путь, соединяющий эти узла. Это означает, что любые два узла сильно связного графа взаимодостижимы. На практике при решении задач декомпозиции достижимость ограничивается длиной пути, т.е. например, при ограничении достижимости величиной $k \leq 3$ связность между двумя узлами орграфа считается сильной, если существуют пути, соединяющие эти узлы, длиной не более 3-х дуг.

Помимо решения задач декомпозиции топологические модели используются для определения структурных свойств системы, среди которых основными для информационных систем являются:

- *сетевая плотность (плотность связей)* – мера связанности между элементами структуры, определяемая как отношение числа существующих к числу возможных связей;

- *плотность центральности* – мера эффективности взаимодействия элементов систем с точки зрения коммуникации.

Плотность центральности оценивают для определения *централи* системы – элемента, из которого все остальные элементы достижимы за минимальное количество шагов. Такой элемент является коммуникационным центром системы, надежность функционирования которого в основном определяет надежность связей в системе. Централь на графе структуры определяется как узел, который со всеми остальными узлами соединяется путями с минимальным числом дуг.

Формулы расчета характеристик структурных свойств систем приведены в разделе 2.3.2 «Создание и анализ топологических моделей систем».

Ход выполнения работы

Дано: структура информационной системы; ограничение достижимости (для вариантов с *четным* номером $k \leq 2$, с *нечетным* – $k \leq 3$).

1. Построить ориентированный граф структуры информационной системы. Построить матрицу смежности графа.

2. Решить задачу декомпозиции структуры информационной системы матричным методом с учетом ограничения достижимости. В случае отсутствия решения задачи декомпозиции для заданного ограничения достижимости повторить решение для большего значения k . Сделать выводы и наличии/отсутствии в структуре системы *а)* элементов, отличающихся относительной изолированностью (независимостью функционирования); *б)* подсистем.

3. Доказать правильность решения задачи декомпозиции путем расчета и сравнения сетевых плотностей полного графа системы и выделенных подсистем. Выявить централь информационной системы путем расчета и сравнения плотностей центральности узлов графа.

4. Результаты выполнения работы по пунктам 1 – 3 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

Практические рекомендации по выполнению работы и использованию программного обеспечения

Применение формул массива для выполнения операций над матрицами в MS Excel – см. *Пример 4 к Лабораторной работе № 2*.

Пример 1

Построение матриц графа структуры ИС, необходимых для решения задачи формальной декомпозиции

Матрица смежности это квадратная матрица $M_C = [m_{ij}]$, $i, j = \overline{1, m}$, где m – число узлов графа, т.е. $M_{m \times m}$, для которой:

- $m_{ij} = 1$, если существует дуга из узла x_i в узел x_j ;
- $m_{ij} = 0$, если такой дуги не существует.

Матрица смежности графа структуры ИС для примера, рассмотренного в разделе 2.3.2 «Создание и анализ топологических моделей систем» (см. файл *Пример декомпозиции структуры ИС.xls*), показана на рис. 114.

	A	B	C	D	E	F	G	H	I	J	K	L
3	Матрица смежности графа структура ИС											
4		G1	P1	G2	S1	S2	D	T1	P3	T2	P2	
5	G1	0	0	0	1	0	1	0	0	0	0	
6	P1	1	0	1	0	0	0	0	0	0	0	
7	G2	0	0	0	0	1	0	0	0	0	0	
8	S1	1	0	0	0	1	1	0	0	0	0	
9	S2	0	1	0	0	0	0	0	0	1	0	
10	D	1	0	0	0	0	0	0	0	0	0	
11	T1	0	0	0	1	0	0	0	0	0	1	
12	P3	0	0	0	0	1	0	0	0	0	1	
13	T2	0	0	0	0	1	0	0	0	0	0	
14	P2	0	0	0	0	0	0	1	1	0	0	
15												

Рис. 114

Пусть в рассматриваемой задаче декомпозиции достижимость ограничена величиной $k \leq 2$, поэтому для построения матрицы достижимости необходимо определить для всех пар узлов наличие или отсутствие путей из x_i в x_j длиной не более 2-х дуг. Уже построенная матрица смежности показывает наличие путей из x_i в x_j длиной в одну дугу. Если возвести матрицу смежности M_C во 2-ю степень, то каждый элемент матрицы M_C^2 будет равен числу путей из узла x_i в узел x_j длиной в 2 дуги. Аналогично, при ограничении $k \leq 3$ необходимо рассчитать M_C^3 и т.д.

Для расчета M_C^2 в MS Excel используется формула массива МУМНОЖ, которая для случая, когда матрица смежности занимает диапазон ячеек B5:K14, имеет следующий синтаксис:

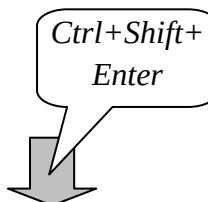
$$=\text{МУМНОЖ}(\text{B5:K14};\text{B5:K14})$$

Помните, что ввод формулы массива осуществляется в левую верхнюю ячейку выделенного диапазона необходимого размера (для вывода результата

расчета) и после завершения ввода формул массива необходимо нажать комбинацию клавиш *Ctrl+Shift+Enter*. Подтверждением правильности ввода формулы массива служит распространение результата вычисления на весь выделенный диапазон и вывод формулы в строке формул в фигурных скобках.

Вычисление *квадрата матрицы смежности* для рассматриваемого примера показана на рис 115 (см. файл *Пример декомпозиции структуры ИС.xls*).

	A	B	C	D	E	F	G	H	I	J	K
21	Квадрат матрицы смежности графа структуры ИС										
22		G1	P1	G2	S1	S2	D	T1	P3	T2	P2
23	G1	=МУМНОЖ(B5:K14;B5:K14)									
24	P1										
25	G2										
26	S1										
27	S2										
28	D										
29	T1										
30	P3										
31	T2										
32	P2										
33											



	A	B	C	D	E	F	G	H	I	J	K
21	Квадрат матрицы смежности графа структуры ИС										
22		G1	P1	G2	S1	S2	D	T1	P3	T2	P2
23	G1	2	0	0	0	1	1	0	0	0	0
24	P1	0	0	0	1	1	1	0	0	0	0
25	G2	0	1	0	0	0	0	0	0	1	0
26	S1	1	1	0	1	0	1	0	0	1	0
27	S2	1	0	1	0	1	0	0	0	0	0
28	D	0	0	0	1	0	1	0	0	0	0
29	T1	1	0	0	0	1	1	1	1	0	0
30	P3	0	1	0	0	0	0	1	1	1	0
31	T2	0	1	0	0	0	0	0	0	1	0
32	P2	0	0	0	1	1	0	0	0	0	2
33											

Рис. 115

Для построения *матрицы достижимости* при ограничении достижимости длиной пути, равной 1 или 2 дуги следует поэлементно сложить M_C и M_C^2 . В результате получится матрица, ненулевые элементы которой показывают наличие путей из узла x_i в узел x_j длиной меньше либо равно 2 дуги. Замена ненулевых элементов этой матрицы на 1 дает матрицу достижимости.

Для расчета матрицы достижимости в MS Excel используется логическая формула ЕСЛИ, которая имеет синтаксис

=ЕСЛИ (лог_выражение;значение_если_истина;значение_если_ложь)

и возвращает одно значение, если заданное условие при вычислении дает значение ИСТИНА, и другое значение, если вычисление условия дает значение ЛОЖЬ.

Логика расчета матрицы достижимости имеет следующий вид: если поэлементная сумма матриц M_C и M_C^2 равна нулю, то пути длиной не более 2-х дуг из x_i в x_j не существует, следовательно соответствующий элемент матрицы достижимости равен 0. Если поэлементная сумма матриц M_C и M_C^2 не равна нулю, то существует хотя бы один путь длиной не более 2-х дуг из x_i в x_j , следовательно соответствующий элемент матрицы достижимости равен 1. Реализация такого расчета для рассматриваемого примера с использованием формулы ЕСЛИ как формулы массива показана на рис. 116 (см. файл *Пример декомпозиции структуры ИС.xls*).

35	Матрица достижимости										
36		G1	P1	G2	S1	S2	D	T1	P3	T2	P2
37	=ЕСЛИ((B23:										
38	K32)+(B5:										
39	K14)=0;0;1)										
40	S1										
41	S2										
42	D										
43	T1										
44	P3										
45	T2										
46	P2										
47											

Ctrl+Shift
+Enter

35	Матрица достижимости										
36		G1	P1	G2	S1	S2	D	T1	P3	T2	P2
37	G1	1	0	0	1	1	1	0	0	0	0
38	P1	1	0	1	1	1	1	0	0	0	0
39	G2	0	1	0	0	1	0	0	0	1	0
40	S1	1	1	0	1	1	1	0	0	1	0
41	S2	1	1	1	0	1	0	0	0	1	0
42	D	1	0	0	1	0	1	0	0	0	0
43	T1	1	0	0	1	1	1	1	1	0	1
44	P3	0	1	0	0	1	0	1	1	1	1
45	T2	0	1	0	0	1	0	0	0	1	0
46	P2	0	0	0	1	1	0	1	1	0	1
47											

Рис. 116

Матрицей контрдостижимости называется транспонированная матрица достижимости. Для выполнения операции транспонирования в MS Excel используется функция ТРАНСП или специальный режим транспонирования при вставке фрагмента, скопированного в буфер обмена (см. Пример 4 к лабораторной работе № 2).

Реализация расчета матрицы контрдостижимости для рассматриваемого примера с использованием формулы массива ТРАНСП показана на рис. 117 (см. файл *Пример декомпозиции структуры ИС.xls*).

35	Матрица достижимости												Матрица контрдостижимости											
36		G1	P1	G2	S1	S2	D	T1	P3	T2	P2			G1	P1	G2	S1	S2	D	T1	P3	T2	P2	
37	G1	1	0	0	1	1	1	0	0	0	0													
38	P1	1	0	1	1	1	1	0	0	0	0													
39	G2	0	1	0	0	1	0	0	0	1	0													
40	S1	1	1	0	1	1	1	0	0	1	0													
41	S2	1	1	1	0	1	0	0	0	1	0													
42	D	1	0	0	1	0	1	0	0	0	0													
43	T1	1	0	0	1	1	1	1	1	0	1													
44	P3	0	1	0	0	1	0	1	1	1	1													
45	T2	0	1	0	0	1	0	0	0	1	0													
46	P2	0	0	0	1	1	0	1	1	0	1													
47																								
35	Матрица достижимости												Матрица контрдостижимости											
36		G1	P1	G2	S1	S2	D	T1	P3	T2	P2			G1	P1	G2	S1	S2	D	T1	P3	T2	P2	
37	G1	1	0	0	1	1	1	0	0	0	0			G1	1	1	0	1	1	1	0	0	0	
38	P1	1	0	1	1	1	1	0	0	0	0			P1	0	0	1	1	1	0	0	1	1	0
39	G2	0	1	0	0	1	0	0	0	1	0			G2	0	1	0	0	1	0	0	0	0	0
40	S1	1	1	0	1	1	1	0	0	1	0			S1	1	1	0	1	0	1	1	0	0	1
41	S2	1	1	1	0	1	0	0	0	1	0			S2	1	1	1	1	1	0	1	1	1	1
42	D	1	0	0	1	0	1	0	0	0	0			D	1	1	0	1	0	1	1	0	0	0
43	T1	1	0	0	1	1	1	1	1	0	1			T1	0	0	0	0	0	0	1	1	0	1
44	P3	0	1	0	0	1	0	1	1	1	1			P3	0	0	0	0	0	0	1	1	0	1
45	T2	0	1	0	0	1	0	0	0	1	0			T2	0	0	1	1	1	0	0	1	1	0
46	P2	0	0	0	1	1	0	1	1	0	1			P2	0	0	0	0	0	0	1	1	0	1
47																								

Рис. 117

Основным средством решения задачи формальной декомпозиции является матрица связности $M_L = [m_{ij}]$, $i, j = \overline{1, m}$, где m – число узлов, элементы которой получаются поэлементным логическим умножением матриц достижимости и контрдостижимости: $m_{ij} = (m_{ij} \in M_A) \wedge (m_{ij} \in M_A^T)$. Для организации вычисления матрицы связности в MS Excel используется формула И, которая имеет следующий синтаксис:

И(логическое_значение1; логическое_значение2; ...)

Формула И возвращает значение ИСТИНА, если все аргументы имеют значение ИСТИНА и значение ЛОЖЬ, если хотя бы один аргумент

имеет значение ЛОЖЬ. Если аргументами формулы являются логические 0 и 1, то 0 эквивалентен значению ЛОЖЬ, а 1 – ИСТИНА. Если необходимо, чтобы результат вычисления формулы И также имел логическое значение 0 или 1, то синтаксис формулы должен иметь вид:

И(логическое_значение1; логическое_значение2; ...)*1.

Так, например, результат вычисления по формуле =И(B37;N37)*1 будет равен 0, если хотя бы в одной из ячеек аргументов формулы (B37 или N37) содержится 0. Расчет матрицы связности для рассматриваемого примера приведен в файле *Пример декомпозиции структуры ИС.xls*.

Пример 2

Решение задачи формальной декомпозиции для структуры ИС

Для решения задачи формальной декомпозиции осуществляется построение *перестановочной матрицы связности*. Узлы, имеющие одинаковые строки (столбцы) в матрице связности, группируются перестановкой строк (столбцов) для получения блочно-диагональной матрицы, в которой элементы, равные 1, сгруппированы как можно более плотно вдоль главной диагонали матрицы. Каждая диагональная группа элементов в виде квадратного блока в перестановочной матрице связности является максимально сильно связным подграфом. Он соответствует фрагменту структуры ИС, который по признаку сильной связности целесообразно выделить в отдельную подсистему.

При построении перестановочной матрицы для рассматриваемого примера в нее в первую очередь были включены узлы, строки которых в матрице связности полностью идентичны. Далее рассматривались строки с наибольшим числом совпадающих элементов, следом – с меньшим и т.д. В файле *Пример декомпозиции структуры ИС.xls* представлен результат анализа, предшествующий построению перестановочной матрицы.

В матрице связности идентичные строки имеют узлы **T1**, **P3** и **P2** (первый порядок включения). Помечаем первые три строки и, соответственно, три столбца обозначениями этих узлов и заполняем фрагмент перестановочной матрицы. На втором шаге рассматриваем строку узла **S1** – она совпадает со строками **D** и **G1** по 3-м связям, поэтому следующие за уже заполненными столбцы и строки перестановочной матрицы помечаем обозначениями этих узлов и продолжаем заполнение (рис. 118).

Далее в матрицу следует включить строку узла **S2** – этот узел связан с узлом **G1**, строка которого включена в матрицу последней, и, кроме того, **S2** имеет максимальное число связей по сравнению с оставшимися узлами,

еще не добавленными в матрицу. Последними в матрицу включаются оставшиеся строки третьего и четвертого порядков. Результат построения перестановочной матрицы показан на рис. 119 (см. файл *Пример декомпозиции структуры ИС.xls*).

50	Первый шаг построения											Второй шаг построения														
51		P3	P2	T1	?	?	?	?	?	?			P3	P2	T1	S1	D	G1	?	?	?	?				
52	P3	1	1	1	0	0	0	0	0	0			P3	1	1	1	0	0	0	0	0	0				
53	P2	1	1	1	0	0	0	0	0	0			P2	1	1	1	0	0	0	0	0	0				
54	T1	1	1	1	0	0	0	0	0	0			T1	1	1	1	0	0	0	0	0	0				
55	?	0	0	0									S1	0	0	0	1	1	1							
56	?	0	0	0									D	0	0	0	1	1	1							
57	?	0	0	0									G1	0	0	0	1	1	1							
58	?	0	0	0									?	0	0	0										
59	?	0	0	0									?	0	0	0										
60	?	0	0	0									?	0	0	0										
61	?	0	0	0									?	0	0	0										

Рис. 118

	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK
63		Перестановочная матрица связности										
64		P3	P2	T1	S1	D	G1	S2	T2	G2	P1	
65	P3	1	1	1	0	0	0	0	0	0	0	
66	P2	1	1	1	0	0	0	0	0	0	0	
67	T1	1	1	1	0	0	0	0	0	0	0	
68	S1	0	0	0	1	1	1	0	0	0	1	
69	D	0	0	0	1	1	1	0	0	0	0	
70	G1	0	0	0	1	1	1	1	0	0	0	
71	S2	0	0	0	0	0	1	1	1	1	1	
72	T2	0	0	0	0	0	0	1	1	0	0	
73	G2	0	0	0	0	0	0	1	0	0	1	
74	P1	0	0	0	1	0	0	1	0	1	0	
75												

Рис. 119

Анализ перестановочной матрицы связности позволяет выделить в составе графа топологии информационной структуры следующие максимально сильно связанные подграфы: (P3, P2, P1); (S1, D, G1) и (S2, T2). Остальные узлы выраженных общих связностей не имеют и могут рассматриваться как элементы системы, отличающиеся относительной изолированностью (независимостью функционирования) от других элементов (см. рис. 41 в разделе 2.3.2 *Создание и анализ топологических моделей систем*).

Пример 3

Решение задачи формальной декомпозиции для произвольного графа

Рассмотрим еще один пример решения задачи формальной декомпозиции с использованием матричного метода для произвольного графа. Пусть граф структуры некоторой системы имеет вид, приведенный на рис. 120, а дополнительным условием к решению задачи декомпозиции является ограничение достижимости величиной $k \leq 3$.

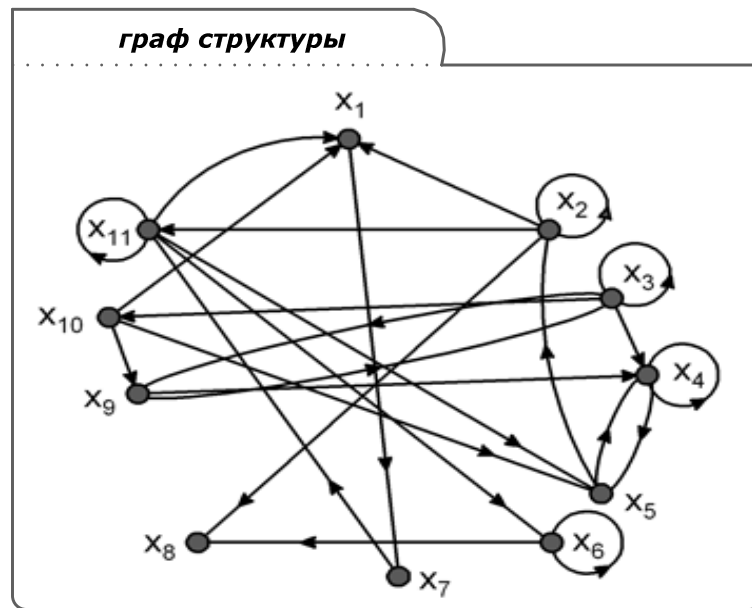


Рис. 120

Построение и расчет матриц, необходимых для решения задачи формальной декомпозиции для заданного графа структуры, выполнены в файле *Пример декомпозиции структуры ИС.xls* (рис. 121).

Как следует из анализа матрицы связности, идентичные строки имеют следующие группы узлов (в файле *Пример декомпозиции структуры ИС.xls* выделены цветом): X_1 , X_7 и X_{11} ; X_3 , X_9 и X_{10} ; X_4 и X_5 . Строки этих групп узлов включаются в перестановочную матрицу в первую очередь. Строки узлов X_2 и X_{16} содержат по одному значимому элементу, а узлу X_8 соответствует нулевая строка – этот узел является стоком. Добавление строк этих узлов в перестановочную матрицу завершает построение (см. файл *Пример декомпозиции структуры ИС.xls*).

Результат построения перестановочной матрицы и декомпозиция заданного графа структуры (выделение сильно связных подграфов G_1 , G_3 и G_4 и изолированных узлов G_2 , G_5 и G_6), приведены на рис. 122.

	A	B	C	D	E	F	G	H	I	J	K	L
3	Матрица смежности графа структуры											
4		X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁
5	X ₁	0	0	0	0	0	0	1	0	0	0	0
6	X ₂	1	1	0	0	0	0	0	1	0	0	1
7	X ₃	0	0	1	1	0	0	0	0	1	1	0
8	X ₄	0	0	0	1	1	0	0	0	0	0	0
9	X ₅	0	0	0	1	0	0	0	0	0	0	0
10	X ₆	0	0	0	0	0	1	0	1	0	0	0
11	X ₇	0	0	0	0	0	0	0	0	0	0	1
12	X ₈	0	0	0	0	0	0	0	0	0	0	0
13	X ₉	0	0	1	0	0	0	0	0	0	0	0
14	X ₁₀	1	0	0	1	1	0	0	0	1	0	0
15	X ₁₁	1	0	0	0	1	1	0	0	0	0	1

	A	B	C	D	E	F	G	H	I	J	K	L
17	Квадрат матрицы смежности графа структуры											
18		X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁
19	X ₁	0	0	0	0	0	0	0	0	0	0	1
20	X ₂	2	1	0	0	1	1	1	1	0	0	2
21	X ₃	1	0	2	3	2	0	0	0	2	1	0
22	X ₄	0	0	0	2	1	0	0	0	0	0	0
23	X ₅	0	0	0	1	1	0	0	0	0	0	0
24	X ₆	0	0	0	0	0	1	0	1	0	0	0
25	X ₇	1	0	0	0	1	1	0	0	0	0	1
26	X ₈	0	0	0	0	0	0	0	0	0	0	0
27	X ₉	0	0	1	1	0	0	0	0	1	1	0
28	X ₁₀	0	0	1	2	1	0	1	0	0	0	0
29	X ₁₁	1	0	0	1	1	2	1	1	0	0	1

	A	B	C	D	E	F	G	H	I	J	K	L
31	Третья степень матрицы смежности графа структуры											
32		X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁
33	X ₁	1	0	0	0	1	1	0	0	0	0	1
34	X ₂	3	1	0	1	2	3	2	2	0	0	4
35	X ₃	1	0	4	8	4	0	1	0	3	2	0
36	X ₄	0	0	0	3	2	0	0	0	0	0	0
37	X ₅	0	0	0	2	1	0	0	0	0	0	0
38	X ₆	0	0	0	0	0	1	0	1	0	0	0
39	X ₇	1	0	0	1	1	2	1	1	0	0	1
40	X ₈	0	0	0	0	0	0	0	0	0	0	0
41	X ₉	1	0	2	3	2	0	0	0	2	1	0
42	X ₁₀	0	0	1	4	2	0	0	0	1	1	1
43	X ₁₁	1	0	0	2	2	3	1	2	0	0	2

	A	B	C	D	E	F	G	H	I	J	K	L
45	Матрица достижимости											
46		X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁
47	X ₁	1	0	0	0	1	1	1	0	0	0	1
48	X ₂	1	1	0	1	1	1	1	1	0	0	1
49	X ₃	1	0	1	1	1	0	1	0	1	1	0
50	X ₄	0	0	0	1	1	0	0	0	0	0	0
51	X ₅	0	0	0	1	1	0	0	0	0	0	0
52	X ₆	0	0	0	0	0	1	0	1	0	0	0
53	X ₇	1	0	0	1	1	1	1	1	0	0	1
54	X ₈	0	0	0	0	0	0	0	0	0	0	0
55	X ₉	1	0	1	1	1	0	0	0	1	1	0
56	X ₁₀	1	0	1	1	1	0	1	0	1	1	1
57	X ₁₁	1	0	0	1	1	1	1	1	0	0	1

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
45	Матрица контрдостижимости												
46		X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	
47	X ₁	1	1	1	0	0	0	1	0	1	1	1	
48	X ₂	0	1	0	0	0	0	0	0	0	0	0	
49	X ₃	0	0	1	0	0	0	0	0	1	1	0	
50	X ₄	0	1	1	1	1	0	1	0	1	1	1	
51	X ₅	1	1	1	1	1	0	1	0	1	1	1	
52	X ₆	1	1	0	0	0	1	1	0	0	0	1	
53	X ₇	1	1	1	0	0	0	1	0	0	1	1	
54	X ₈	0	1	0	0	0	1	1	0	0	0	1	
55	X ₉	0	0	1	0	0	0	0	0	1	1	0	
56	X ₁₀	0	0	1	0	0	0	0	0	1	1	0	
57	X ₁₁	1	1	0	0	0	0	1	0	0	1	1	

	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
	Матрица связности											
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	
47	X ₁	1	0	0	0	0	0	1	0	0	0	1
48	X ₂	0	1	0	0	0	0	0	0	0	0	0
49	X ₃	0	0	1	0	0	0	0	0	1	1	0
50	X ₄	0	0	0	1	1	0	0	0	0	0	0
51	X ₅	0	0	0	1	1	0	0	0	0	0	0
52	X ₆	0	0	0	0	0	1	0	0	0	0	0
53	X ₇	1	0	0	0	0	0	1	0	0	0	1
54	X ₈	0	0	0	0	0	0	0	0	0	0	0
55	X ₉	0	0	1	0	0	0	0	0	1	1	0
56	X ₁₀	0	0	1	0	0	0	0	0	1	1	0
57	X ₁₁	1	0	0	0	0	0	1	0	0	0	1

Рис. 121

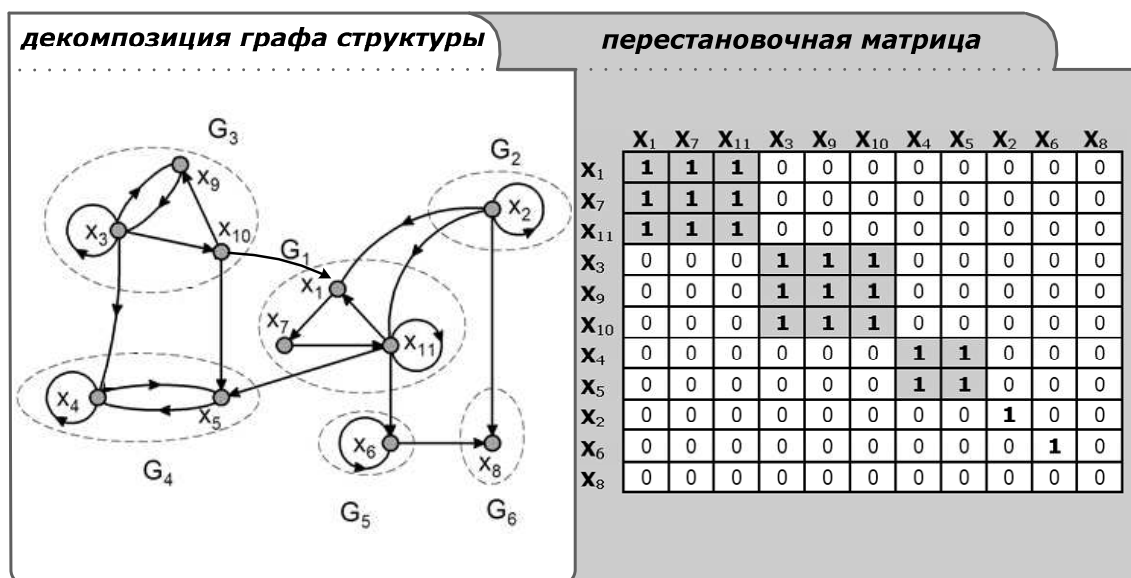


Рис. 122

Пример 4

Расчет и анализ структурных свойств систем

Доказательством корректности решения задачи декомпозиции может служить повышение значения сетевой плотности (плотности связей) для подсистем в сравнении со значением той же характеристики полного графа структуры. Рассчитаем сетевую плотность графа структуры информационной системы, декомпозиция которой рассматривалась в *Примере 2*, по формуле $\Delta_G = \frac{L}{m(m-1)}$, где m – число вершин полного графа структуры;

L – число существующих связей в графе (вычисляется как сумма всех элементов матрицы смежности). Результат расчета имеет вид:

$$\Delta_G = \frac{18}{10(10-1)} = 0,2$$

Аналогично рассчитывается сетевая плотность подсистем (выделенных подграфов), для которых предварительно необходимо построить собственные матрицы смежности. Окончательный результат расчета имеет вид (см. файл *Пример декомпозиции структуры ИС.xls*):

	G	G_1	G_2	G_3
m	10	3	3	2
L	18	4	5	2
Δ	0,20	0,67	0,83	1,00

Анализ результатов расчета подтверждает правильность решения задачи декомпозиции: сетевая плотность всех выделенных подсистем значительно выше сетевой плотности полного графа структуры.

Для выявления централи необходимо рассчитать плотности центральности узлов графа по формуле

$$c_{den}(x_i) = \frac{1}{\sum_{j=1}^m k_{ij}},$$

где k_{ij} – число дуг в наименьшем по количеству дуг пути из узла x_i в узел x_j (длина минимального пути между узлами x_i и x_j). При решении задачи декомпозиции достижимость между узлами графа структуры была ограничена $k \leq 2$. При решении задачи выявления централи это ограничение необходимо снять, т.к. при нарушении работоспособности кратчайших каналов связи в какой-либо подсистеме целостность коммуникаций в системе будет, при наличии возможности, обеспечена связями неограниченной длины, проходящими через централь.

Первым шагом расчета плотностей центральности узлов графа является расчет *матриц существования путей* различной длины. Для $k = 1$ такой матрицей является матрица смежности графа. В *Примере 1* рассчитывался и квадрат матрицы смежности. Каждый элемент матрицы M_c^2 равен числу путей из узла x_i в узел x_j длиной в 2 дуги. Однако, в данном случае, нас не интересует количество путей заданной длины, а интересует только факт наличия или отсутствия между парами узлов графа путей длиной в 2 дуги. Т.е. элемент матрицы существования путей для $k = 2$ должен быть равен 1, если существует хотя бы один путь такой длины между парой узлов, и 0, если путь такой длины не существует. Реализовать расчет такой матрицы в MS Excel можно с использованием формулы массива, которая поэлементно проверяет результат возведения в квадрат матрицы смежности на равенство/неравенство 0 и по результату проверки присваивает элементу значение, соответственно, 0 или 1. В рассматриваемом примере, в котором матрица смежности занимает диапазон ячеек B32:K41, эта формула имеет следующий синтаксис (см. файл *Пример декомпозиции структуры ИС.xls*):

$$=ЕСЛИ(МУМНОЖ(B32:K41;B32:K41)>0;1;0).$$

Результат расчета матрицы существования путей для $k = 2$ в рассматриваемом примере имеет вид (рис. 123).

Расчет матриц существования путей для других значений k выполняется аналогично.

	M	N	O	P	Q	R	S	T	U	V	W
30	Матрица существования путей $k=2$										
31		G1	P1	G2	S1	S2	D	T1	P3	T2	P2
32	G1	1	0	0	0	1	1	0	0	0	0
33	P1	0	0	0	1	1	1	0	0	0	0
34	G2	0	1	0	0	0	0	0	0	1	0
35	S1	1	1	0	1	0	1	0	0	1	0
36	S2	1	0	1	0	1	0	0	0	0	0
37	D	0	0	0	1	0	1	0	0	0	0
38	T1	1	0	0	0	1	1	1	1	0	0
39	P3	0	1	0	0	0	0	1	1	1	0
40	T2	0	1	0	0	0	0	0	0	1	0
41	P2	0	0	0	1	1	0	0	0	0	1

Рис. 123

Всякий раз после расчета матрицы существования путей для нового значения k следует выполнять расчет матрицы достижимости (см. *Пример 1*). Критерием завершения расчета является выполнение одного из четырех условий [42]:

- 1) рассчитана матрица достижимости для $k = m - 1$, т.е. для значения, равного длине максимально возможного пути в графе с m узлами;
- 2) равенство единице всех элементов матрицы для текущего значения k , означающее взаимную достижимость всех узлов графа на пути длиной, равной k ;
- 3) идентичность матриц k -той степени и степени $k + 1$, означающая, что длина пути для всех попарно достижимых узлов найдена, и дальнейшее увеличение k приводит к нахождению длин путей, увеличенных на одну дугу за счет петель графа;
- 4) чередование через одну, две и т.д. идентичных матриц разных степеней, означающее, что длина пути для всех попарно достижимых узлов найдена и дальнейшее увеличение k приводит к нахождению длин путей, увеличенных за счет замкнутых контуров графа на две, три дуги и т.д.

Третий вариант критерия завершения означает, что в графе есть пары узлов, в принципе не достижимые друг для друга. Именно такой вариант завершения расчета имеет место в рассматриваемом примере: при выборе $k = 6$ и более матрица достижимости не изменяется по сравнению с вариантом, полученным для $k = 5$. Следовательно, те пары узлов, на пересечении строк и столбцов которых в матрице достижимости для $k \leq 5$ стоит значение 1, взаимно достижимы при помощи путей длиной от 1 до 5 дуг, а для тех пар, на пересечении которых стоит 0, путей взаимной достижимости не

существует. Таким образом, в данном случае, окончательный вариант матрицы достижимости имеет вид, показанный на рис. 124 (см. файл *Пример декомпозиции структуры ИС.xls*).

	A	B	C	D	E	F	G	H	I	J	K
44	Матрица достижимости для k≤5										
45		G1	P1	G2	S1	S2	D	T1	P3	T2	P2
46	G1	1	1	1	1	1	1	0	0	1	0
47	P1	1	1	1	1	1	1	0	0	1	0
48	G2	1	1	1	1	1	1	0	0	1	0
49	S1	1	1	1	1	1	1	0	0	1	0
50	S2	1	1	1	1	1	1	0	0	1	0
51	D	1	1	1	1	1	1	0	0	1	0
52	T1	1	1	1	1	1	1	1	1	1	1
53	P3	1	1	1	1	1	1	1	1	1	1
54	T2	1	1	1	1	1	1	0	0	1	0
55	P2	1	1	1	1	1	1	1	1	1	1

Рис. 124

Из полученной матрицы достижимости следует, например, что пути, соединяющего узел **G1** с узлом **T1**, не существует. В этом легко убедиться, проанализировав граф топологии информационной структуры (см. файл *Пример декомпозиции структуры ИС.xls*).

Вторым шагом расчета плотностей центральности узлов графа является расчет *матрицы минимальных путей*, попарно соединяющих узлы графа. Реализовать расчет такой матрицы в MS Excel можно с использованием формул ЕСЛИ (см. *Пример 1*) и МИН. Последняя возвращает минимальное значение из списка аргументов, заданного ссылками на соответствующие ячейки, и имеет следующий синтаксис:

МИН(ссылка1;ссылка2; ...).

Логика расчета матрицы минимальных путей следующая: если сумма одноименных ячеек всех матриц существования путей равна 0, то пути, соединяющего пару узлов, не существует, и ячейке матрицы минимальных путей также присваивается значение 0. Если сумма одноименных ячеек всех матриц существования путей не равна 0, то существует один или несколько путей разной длины, соединяющих соответствующую пару узлов, и из множества этих длин следует выбрать минимальную. Однако, чтобы такой выбор, реализованный с использованием формулы МИН, не вернул значение 0 в том случае, если пути какой-то конкретной длины не существует, предварительно с использованием формул ЕСЛИ следует заменить

нулевые значения, полученные ссылками на ячейки матриц существования путей, каким-нибудь большим числом, например, 100. Пример реализации полной формулы расчета значений матрицы минимальных путей приведен в файле *Пример декомпозиции структуры ИС.xls*.

В завершении расчета плотностей центральности узлов графа выполняется расчет суммы k_{ij} (число дуг в наименьшем по количеству дуг пути из узла x_i в узел x_j) для каждого узла графа, т.е. для каждой строки матрицы минимальных путей, и собственно расчет значений $c_{den}(x_i)$.

Матрица минимальных путей и результат расчета плотностей центральности узлов графа структуры информационной системы для данного примера показаны на рис. 125 (см. файл *Пример декомпозиции структуры ИС.xls*).

	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
43													$\sum_{j=1}^m k_{ij}$		
44	Матрица минимальных путей													$c_{den}(x_i)$	
45		G1	P1	G2	S1	S2	D	T1	P3	T2	P2				
46	G1	2	3	4	1	2	1	0	0	3	0		16	0,063	
47	P1	1	3	1	2	2	2	0	0	3	0		14	0,071	
48	G2	3	2	3	4	1	4	0	0	2	0		19	0,053	
49	S1	1	2	3	2	1	1	0	0	2	0		12	0,083	
50	S2	2	1	2	3	2	3	0	0	1	0		14	0,071	
51	D	1	4	5	2	3	2	0	0	4	0		21	0,048	
52	T1	2	3	4	1	2	2	2	2	3	1		22	0,045	
53	P3	3	2	3	3	1	4	2	2	2	1		23	0,043	
54	T2	3	2	3	4	1	4	0	0	2	0		19	0,053	
55	P2	3	3	4	2	2	3	1	1	3	2		24	0,042	

Рис. 125

Наибольшую плотность центральности из всех узлов графа имеет узел **S1**, соответственно он и является централью данной системы. Однако построчный анализ матриц достижимости и минимальных путей показывает, что из узла **S1** ряд узлов графа в принципе не достижимы: это узлы **T1**, **P3** и **P2**. Поэтому в подграфе (**T1**, **P3**, **P2**) следует выделить собственную централь. Наибольшее значение плотности центральности в подграфе (**T1**, **P3**, **P2**) имеет узел **T1**: $c_{den}(\mathbf{T1}) = 0,45$, следовательно для этого подграфа он является централью.

Варианты индивидуальных заданий

Во всех вариантах структур информационных систем (ИС) однонаправленные связи обозначены линиями со стрелками, двунаправленные – линиями без стрелок. При построении графа структуры двунаправленная связь заменяется двумя симметричными дугами (см. рис. 19 *Лекционного курса*).

№	Структура ИС	№	Структура ИС
1		2	
3		4	
5		6	

№	Структура ИС	№	Структура ИС
7		8	
9		10	
11		12	
13		14	

№	Структура ИС	№	Структура ИС
15		16	
17		18	
19		20	
21		22	
23		24	

Лабораторная работа № 4

Анализ структурно-топологических характеристик топологий вычислительных сетей

Цель: сравнительный анализ и оптимизация вариантов топологий ВС по результатам расчета структурно-топологических характеристик с учетом дополнительных ограничений.

Программное обеспечение: MS Excel.

Примеры выполнения работы: файл *Пример анализа СТХ.xls*.

Теория

Основные понятия теории графов, способы их задания и правила построения матрицы смежности приведены в разделе 2.3.1 *Основные понятия и способы задания графов*.

Топологические модели систем используются для решения задач формальной декомпозиции и анализа структурных свойств. В *Лабораторной работе № 3* были рассмотрены способы оценки двух структурных характеристик ИС: *сетевой плотности (плотности связей)*, являющейся мерой связанности между элементами структуры, и *плотности центральности* – меры эффективности взаимодействия элементов систем с точки зрения коммуникации. Помимо этих показателей для сравнительной оценки топологий вычислительных систем на ранней стадии проектирования рассчитывают дополнительно еще ряд характеристик. Основой расчетов структурно-топологических характеристик ВС является матрица смежности ориентированного (в случае однонаправленных связей), неориентированного (в случае использования для каждого соединения параллельных линий связи, передающих информацию в противоположных направлениях) или смешанного графа топологии.

Число единиц в матрице M_C неориентированного графа равно $2n$, где n – число ребер графа. Для орграфа число единиц в матрице M_C равно числу дуг n . Для смешанного типа число единиц зависит от числа каждой из разновидностей связей. Для анализа графы смешанного типа удобнее представлять орграфами, для чего каждое ребро графа, т.е. каждая двунаправленная связь заменяется двумя симметричными дугами

Рассмотрим структурно-топологические характеристики топологий ВС.

1. *Структурная избыточность R* графа топологии рассчитывается в долях или процентах и показывает превышение общего числа связей над

минимально необходимым для обеспечения связности между всеми вершинами. Для неориентированного графа минимально необходимое число связей равно $m - 1$, где m – число вершин графа, т.к. для обеспечения связности всех элементов при использовании двунаправленных связей достаточно реализовать их последовательное (линейное) соединение. Для орграфа топология, обеспечивающая связность при наименьшем числе дуг, представляет собой однонаправленное соединение по кольцу, т.е. для орграфа минимально необходимое число связей равно m , где m – число узлов графа). Т.о., формулы расчета структурной избыточности для неориентированного и ориентированного графа имеют вид:

$$R_{nor} = \frac{n}{m-1} - 1, \quad R_{or} = \frac{n}{m} - 1.$$

Для несвязанных систем $R < 0$. Вычислительные сети с бóльшей избыточностью потенциально более надежны, но реализация избыточности требует дополнительных затрат.

2. *Неравномерность распределения связей* ϵ^2 характеризует недоиспользование заданной топологии, имеющей n ребер/дуг и m вершин/узлов, в достижении максимальной связности. Для систем, критичных к равномерности распределения связей, из альтернативных вариантов топологий выбирается структура с наименьшим значением ϵ^2 .

Равномерное распределение связей графа характеризуется средней степенью графа $\bar{d} = \frac{2n}{m}$ и отклонением Δ_i действительной степени вершины d_i (число ребер или дуг, смежных с i -ой вершиной) от $\bar{d}$: $\Delta_i = d_i - \bar{d}$. Неравномерность распределения связей рассчитывается по формуле

$$\epsilon^2 = \sum_{i=1}^m (\Delta_i)^2 = \sum_{i=1}^m (d_i - \bar{d})^2 = \sum_{i=1}^m \left(d_i - \frac{2n}{m} \right)^2.$$

Равномерное распределение связей в графе топологии характеризуется значением $\epsilon^2 = 0$.

Следует отметить, что с использованием матрицы смежности M_C неориентированного графа степень вершины можно рассчитать как сумму элементов столбца или строки вершины, т.к. M_C неориентированного графа симметрична. Степень вершины орграфа равна сумме полустепеней исхода $d_{out}(x_i)$ и захода $d_{in}(x_i)$. Полустепенью исхода $d_0(x_i)$ называется количество дуг, исходящих из узла x_i . Полустепенью захода $d_t(x_i)$ называется количество дуг, входящих в этот узел. Следовательно, при использовании для

расчета степени вершины орграфа матрицы смежности M_C требуется суммировать значения элементов столбца i строки конкретного узла.

3. Структурная компактность $Q_{отн}$ и диаметр структуры D характеризуют близость элементов топологии ВС между собой. Близость элементов i и j определяется через минимальную длину пути для орграфа и цепи для неориентированного, и обозначается k_{ij} . Структурная компактность рассчитывается по формуле

$$Q_{отн} = \frac{Q}{m(m-1)} - 1 = \frac{\sum_{i=1}^m \sum_{j=1}^m k_{ij}}{m(m-1)} - 1; \quad i \neq j;$$

$$D = \max_{i,j} \{k_{ij}\}.$$

где $Q = \sum_{i=1}^m \sum_{j=1}^m k_{ij}$ – сумма всех элементов матрицы минимальных путей графа без учета значений, стоящих на главной диагонали.

Показатели $Q_{отн}$ и D характеризуют инерционность информационных процессов в системе. Если в сравниваемых альтернативных вариантах топологий значения характеристик ϵ^2 и R равны, то предпочтительным является вариант структуры с меньшими значениями $Q_{отн}$ и D , т.к. возрастание этих характеристик отражает увеличение разделяющих связей в структуре, характеризуя тем самым снижение общей надежности. У *полного графа*, т.е. у графа, в котором любая пара вершин смежна, диаметр структуры имеет минимально возможное значение $D = 1$. Такой граф обладает максимальной компактностью.

4. Индекс централизации графа δ является мерой равномерности распределения связей в графе, альтернативной ϵ^2 . Для топологий с равномерным распределением связей δ , как и ϵ^2 , равно 0; для систем с максимальной степенью централизации $\delta = 1$. Рассчитывается δ по формуле

$$\delta = \frac{(m-1)(2Z_{\max} - m)}{Z_{\max}(m-2)}; \quad Z_{\max} = \max\{Z_i\},$$

где Z_i – степень централизации i -той вершины:

$$Z_i = \frac{Q}{m \sum_{j=1}^m k_{ij}}; \quad i \neq j.$$

Ход выполнения работы

Дано: два варианта топологии ВС; дополнительная задача анализа или оптимизации топологии.

1. Построить неориентированный граф топологии ВС. Построить матрицу смежности графа.

2. Выполнить расчет структурно-топологических характеристик (СТХ) топологий ВС. Выполнить сравнительный анализ вариантов топологии по следующим критериям:

- идентичность/не идентичность вариантов;
- соответствие/несоответствие вариантов топологии полному графу;
- надежность вариантов топологии;
- равномерность распределения связей в вариантах топологии.

3. Выполнить решение дополнительной задачи анализа или оптимизации топологии в соответствии с индивидуальным заданием варианта.

4. Результаты выполнения работы по пп. 1 – 3 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

Практические рекомендации по выполнению работы и использованию программного обеспечения

Пример 1

Расчет и анализ СТХ топологий ВС

Пусть заданы два варианта топологии ВС – последовательная линейная (не путать с топологией типа «шина», в которой все компьютеры параллельно подключаются к общей магистрали) и кольцевая (рис. 126). Оба варианта топологии построены с использованием двух параллельных линий связи, передающих информацию в противоположных направлениях.

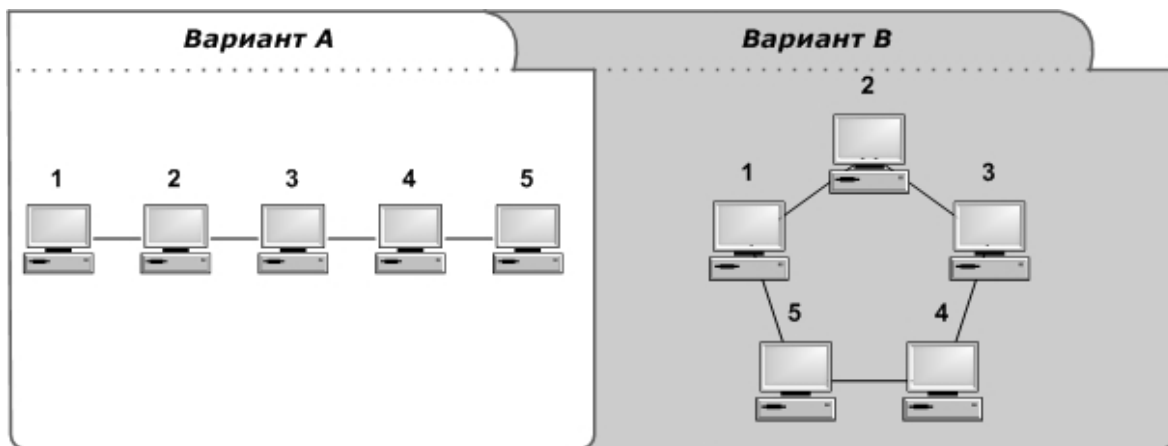


Рис. 126

Неориентированные графы заданных вариантов топологии и их матрицы смежности показаны на рис. 127 (см. файл *Пример анализа CTX.xls*).

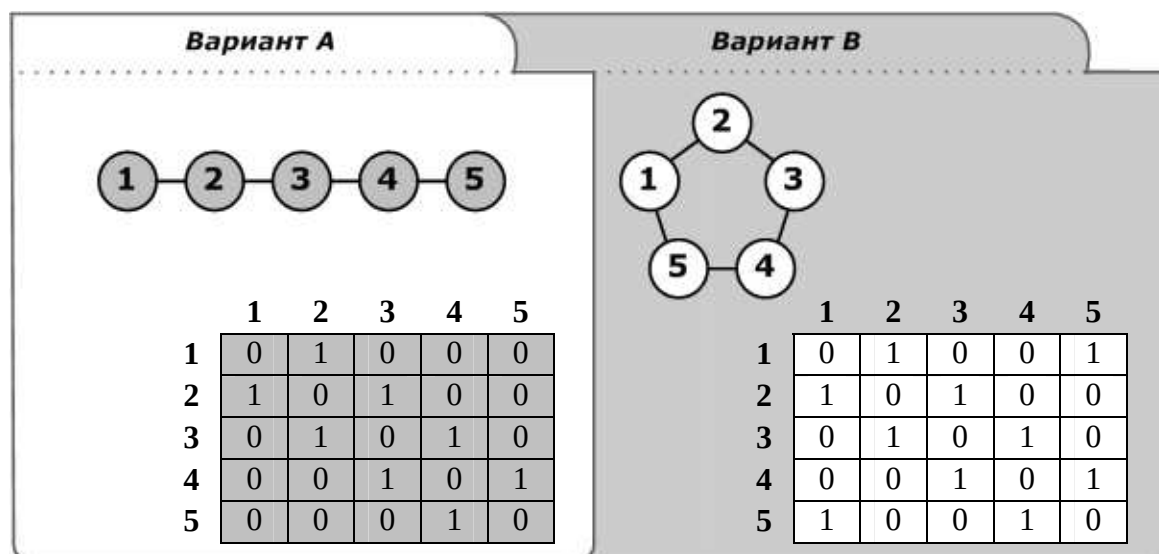


Рис. 127

Пример расчета СТХ вариантов топологии по построенным матрицам смежности, выполненный с использованием формул, приведенных в разделе *Теория* данной лабораторной работы – см. файл *Пример анализа CTX.xls*. Применение формул массива для выполнения операций над матрицами в MS Excel – см. *Пример 4 к Лабораторной работе № 2*.

Для расчета структурной компактности Q , диаметра структуры D и индекса централизации δ необходимо выполнить расчет матриц существования в графе путей заданной длины и построение матрицы минимальных путей. Способ выполнения такого расчета был рассмотрен в *Примере 4 к Лабораторной работе № 3* и в данном случае выполняется аналогично, с учетом следующей особенности: при построении матрицы минимальных путей не заполняется главная диагональ.

Результат расчета СТХ для сравнительного анализа заданных вариантов топологий ВС приведен в табл. 16.

Таблица 16

Вариант	Структурная избыточность R	Неравномерность распределения связей ϵ^2	Структурная компактность $Q_{отн}$	Диаметр структуры D	Степень централизации δ
А	0	1,2	1	4	0,67
В	0,25	0	0,5	2	0

Сравнительный анализ СТХ заданных топологий позволяет утверждать следующее:

- предложенные варианты топологий не идентичны – рассчитанные значения СТХ не совпадают;
- ни один из вариантов топологии не соответствует полному графу, т. к. диаметры структур $D \neq 1$;
- вариант топологии **B** обеспечит бóльшую надежность функционирования ВС, чем вариант **A**, ввиду наличия структурной избыточности ($R_B=0,25$) и меньших значений $Q_{отн}$ и D .
- вариант топологии **B** обеспечивает максимально возможную равномерность распределения связей;
- вариант топологии **A** характеризуется существенной неравномерностью распределения связей ($\epsilon_A^2 = 1,2$) и значительной централизацией: $\delta_A = 0,67$, что близко к максимально возможному значению – единице.

Таким образом, с точки зрения надежности и равномерности распределения связей предпочтительным является вариант **B**, что вполне объяснимо – в отсутствии других альтернатив кольцевая топология ВС предпочтительнее последовательной.

Пример 2

Решение задачи оптимизации топологии

Пусть в соответствии с дополнительными условиями проектирования на второй элемент сети требуется возложить функции управления решением общих сетевых задач. Решить поставленную задачу можно с использованием ранее выбранного варианта топологии **B**, однако для его реализации потребуется $2n=10$ соединений. Выполним сравнительный анализ вариантов реализации такого условия: стандартной кольцевой однонаправленной топологии (число соединений равно числу $n=5$) и топологии «активная звезда» с использованием второго компьютера в качестве сервера (число соединений равно числу $2n = 8$). Графы таких вариантов топологии и их матрицы смежности показаны на рис. 128.

Реализация расчета СТХ для предлагаемых вариантов топологий **C** и **D** (рис. 128) выполнена в файле *Пример анализа СТХ.xls*. Результат расчета СТХ в сравнении со значениями, рассчитанными в *Примере 1* для топологии **B**, приведены в табл. 17.

Сравнительный анализ СТХ заданных топологий позволяет утверждать следующее:

- предложенные варианты топологий не идентичны – рассчитанные значения СТХ не совпадают;
- ни один из вариантов топологии не соответствует полному графу, т.к. диаметры структур $D \neq 1$;

- вариант топологии **C** и **D** не обладают структурной избыточностью ($R = 0$), однако значения структурной компактности $Q_{отн}$ и диаметра структуры D у варианта **D** меньше, следовательно общая надежность у этого варианта больше;
- вариант топологии **C** обеспечивает максимально возможную равномерность распределения связей;
- вариант топологии **D** характеризуется очень большой неравномерностью распределения связей ($\epsilon_D^2 = 7,2$) и максимально возможной централизацией: $\delta_D = 1$.

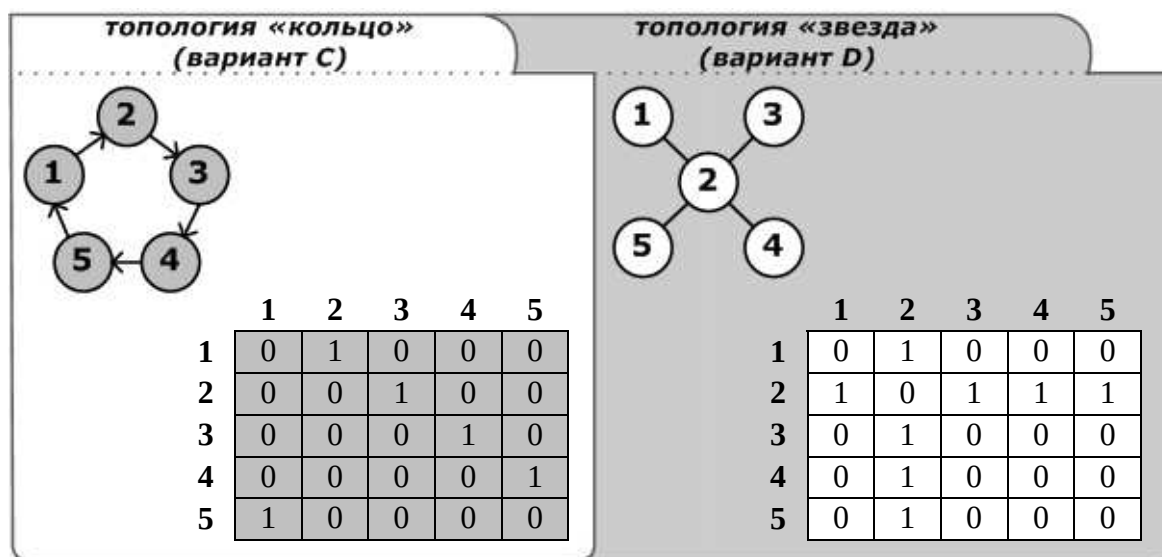


Рис. 128

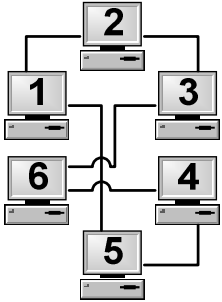
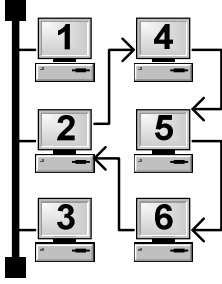
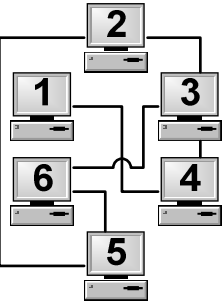
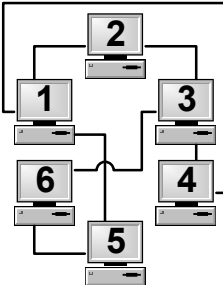
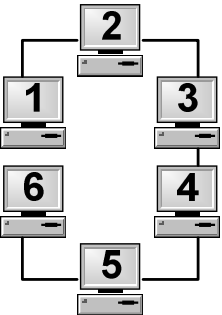
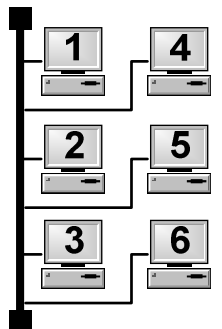
Таблица 17

Вариант	Структурная избыточность R	Неравномерность распределения связей ϵ^2	Структурная компактность $Q_{отн}$	Диаметр структуры D	Степень централизации δ
<i>B</i>	0,25	0	0,5	2	0
<i>C</i>	0	0	1,5	4	0
<i>D</i>	0	7,2	0,6	2	1

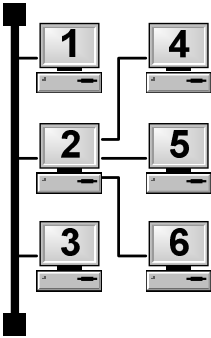
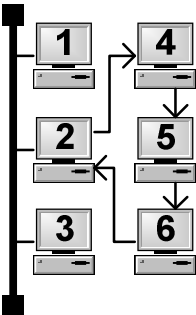
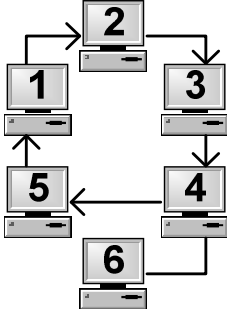
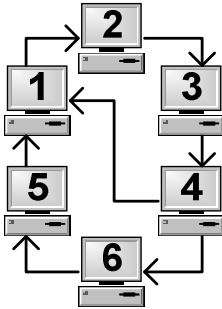
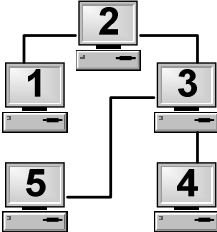
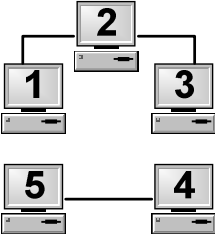
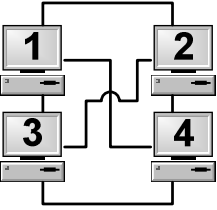
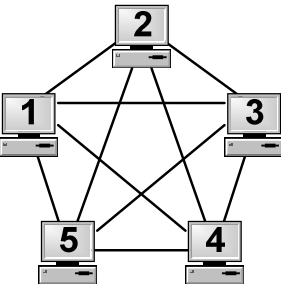
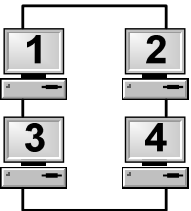
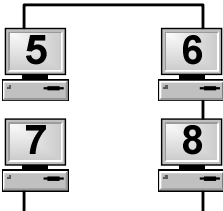
Таким образом, если число соединений является критическим показателем, и требуется высокая степень управляемости с точки зрения общесетевых задач, обеспечиваемая максимальной централизацией, то предпочтительной реализацией топологии является вариант **D**. Если у общесетевых задач нет большого приоритета и нет ограничения на число используемых соединений, то предпочтительным является вариант **B**.

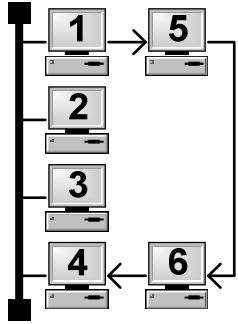
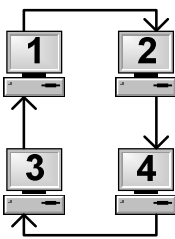
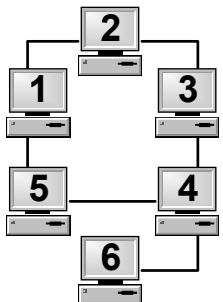
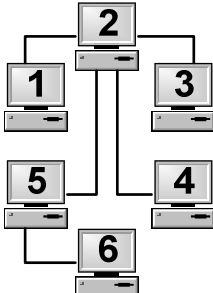
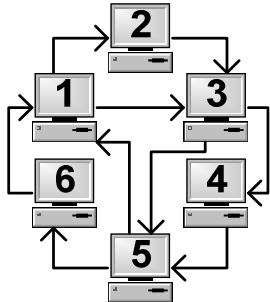
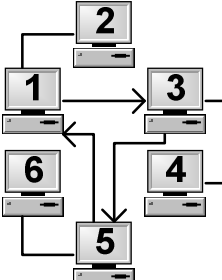
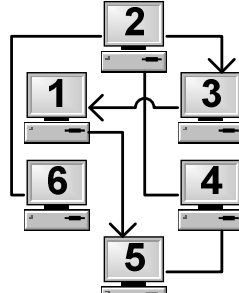
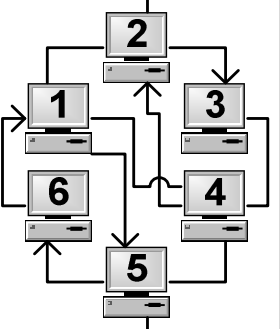
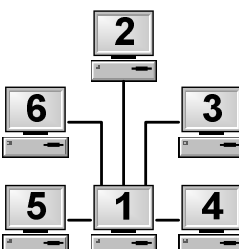
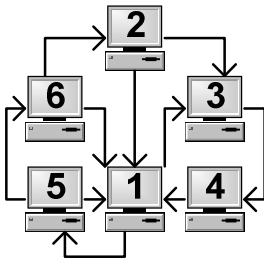
Варианты индивидуальных заданий

Во всех вариантах топологий ВС однонаправленные связи обозначены линиями со стрелками, двунаправленные – линиями без стрелок. Варианты со смешанным типом связей следует представлять оргграфом: при его построении двунаправленная связь заменяется двумя симметричными дугами (см. рис. 19 *Лекционного курса*). Общая шина обозначается жирной линией. При модификации топологий для решения дополнительной задачи, кроме указанных в качестве критерия оптимизации, рассчитываются и анализируются все СТХ.

№	Вариант топологии А	Вариант топологии В	Дополнительная задача анализа или оптимизации
1			Определить, какой из вариантов топологии можно модифицировать до обеспечения максимальной компактности с использованием меньшего числа дополнительных связей
2			Для варианта с лучшей равномерностью распределения связей исследовать изменение характеристик надежности при добавлении ребер в граф по одному до построения полного графа
3			Для варианта топологии с меньшей структурной избыточностью предложить модификацию с использованием минимального числа дополнительных связей, обеспечивающую равномерное распределение связей

№	Вариант топологии <i>A</i>	Вариант топологии <i>B</i>	Дополнительная задача анализа или оптимизации
4			Граф топологии с меньшей компактностью без добавления дополнительных связей модифицировать так, чтобы степень централизации 6 вершины была максимальной
5			Лучший с точки зрения надежности вариант топологии модифицировать с целью обеспечения минимально возможного диаметра структуры без добавления дополнительных связей
6			Для более надежного варианта топологии предложить две модификации (с и без добавления связей) с учетом необходимости использовать компьютер 2 как сервер в подсети 2-4-5
7			Две подсети с заданной топологией объединить без добавления связей в общую сеть для обеспечения максимальной компактности и возможности использовать компьютер 5 как сервер в каждой из подсетей
8			Две подсети с заданной топологией объединить без добавления связей в общую сеть с обеспечением максимально возможной надежности

№	Вариант топологии <i>A</i>	Вариант топологии <i>B</i>	Дополнительная задача анализа или оптимизации
9			Лучший с точки зрения надежности вариант топологии модифицировать до отсутствия структурной избыточности с обеспечением максимально возможной равномерности распределения связей
10			Для более надежного варианта топологии предложить две модификации (с и без добавления связей) с учетом необходимости использовать компьютер 1 как сервер для двух подсетей: 2-3-4 и 4-5-6
11			С использованием элементов двух заданных топологий построить топологию в виде полного орграфа с максимально возможным числом узлов и определить диаметр структуры
12			Для обоих вариантов топологии исследовать изменение всех СТХ при удалении связей по одной до достижения отсутствия структурной избыточности
13			Две подсети с заданной топологией объединить без добавления связей в общую сеть с обеспечением минимально возможного диаметра структуры

№	Вариант топологии <i>A</i>	Вариант топологии <i>B</i>	Дополнительная задача анализа или оптимизации
14			С использованием элементов двух заданных топологий построить топологию в виде полного графа с максимально возможным числом вершин (узлов) и определить диаметр структуры
15			Для варианта с бóльшей надежностью исследовать изменение характеристик равномерности распределения связей при добавлении ребер в граф по одному до построения полного графа
16			Граф топологии с меньшим диаметром структуры модифицировать без добавления дополнительных связей так, чтобы степень централизации 2 вершины была максимальной
17			Выполнить сравнительный анализ обеих топологий с полным кольцом (неориентированный граф в виде кольца, соединяющего 6 вершин)
18			Вариант топологии с меньшей структурной избыточностью с использованием имеющихся связей модифицировать так, чтобы на 5 элемент структуры можно было возложить функции управления решением общесетевых задач

№	Вариант топологии <i>A</i>	Вариант топологии <i>B</i>	Дополнительная задача анализа или оптимизации
19			Для варианта с бóльшей надежностью исследовать изменение характеристик надежности при удалении дуг по одной до достижения отсутствия структурной избыточности
20			С использованием элементов двух заданных топологий построить сеть из 8 компьютеров с максимально возможной равномерностью распределения связей
21			Для варианта с бóльшей структурной избыточностью с использованием имеющихся связей предложить вариант построения топологии с двумя подсетями (1-2-3 и 3-4-5) с максимально возможной централизацией каждой из подсетей
22			Для варианта с бóльшей надежностью исследовать изменение характеристик надежности при удалении дуг по одной до достижения отсутствия структурной избыточности
23			Лучший с точки зрения надежности вариант топологии модифицировать с целью обеспечения максимально возможного диаметра структуры без добавления дополнительных связей
24			Две подсети с заданной топологией объединить без добавления связей в общую сеть для обеспечения максимальной надежности при максимальной степени централизации компьютера 2

Лабораторная работа № 5

Моделирование системной динамики сетями Петри

Цель: моделирование процессов целенаправленного функционирования системы с учетом ограничения ресурсов системы и разной приоритетности процессов.

Программное обеспечение: PetriNets InProject.

Примеры выполнения работы: файлы *Параллельные узлы.rpe*, *Ограниченная очередь.rpe*, *Ограниченный ресурс.rpe*.

Теория

Основные сведения по моделированию систем различной природы сетями Петри приведены в разделе 2.5 *Непрерывно-дискретное моделирование систем сетями Петри*.

Ход выполнения работы

Дано: описание закономерностей функционирования системы с заданием ограничений на ресурсы и приоритетов или вероятностей осуществления процессов.

1. По варианту индивидуального задания разработать сеть Петри, моделирующую динамику функционирования системы, и реализовать ее в пакете PetriNets InProject. В отчете представить:

- начальную (нулевую) маркировку сети;
- описание всех позиций (вершин) сети с точки зрения условий, которые с ними связаны;
- описание всех переходов с точки зрения связанных с ними процессов и времен их осуществления.

2. Осуществить моделирование функционирования системы в автоматическом режиме и режиме пошагового прогона. В отчете представить:

- схемы маркировок сети Петри и их описание для *всех значимых ситуаций* функционирования СМО, полученные в режиме пошагового прогона в PetriNets InProject;
- схему конечной маркировки сети Петри, соответствующую конечному состоянию системы, если такое состояние возможно по условиям функционирования системы.

3. Результаты выполнения работы по пп. 1 – 2 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

Практические рекомендации по выполнению работы и использованию программного обеспечения

Основные принципы работы с пакетом PetriNets InProject

PetriNets InProject обеспечивает:

- создание выбранного типа сети Петри с использованием встроенного визуального редактора графа;
- параметризация графа сети Петри путем задания начальной маркировки сети, определения свойств позиций и переходов;
- моделирование динамики функционирования сети в виде последовательной смены маркировок в режиме автоматического прогона или пошагового выполнения.

В программе реализован пошаговый алгоритм управления модельным временем. Одному шагу приращения модельного времени равно минимально возможное время реализации действий и наступления событий в системе. Минимально возможное время срабатывания перехода также равно 1 шагу модельного времени.

Запуск ПО осуществляется файлом PetriNets_InProject.exe. Рабочее окно программы имеет вид (рис. 129).

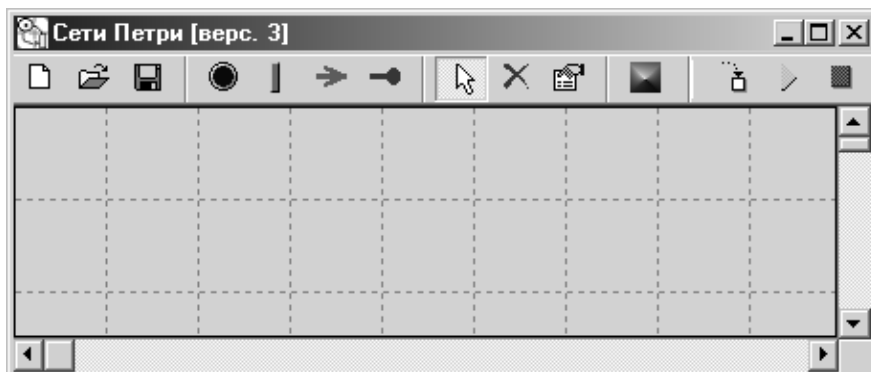


Рис. 129

Кнопки графической панели инструментов снабжены всплывающими подсказками. Кнопка  – «Создать» – предназначена для создания нового графа сети Петри (модели). При использовании этой кнопки в режиме редактирования модели программа запрашивает необходимость сохранения текущей модели в файл и в случае отказа от сохранения удаляет текущую модель. Кнопка  – «Открыть» – предназначена для открытия файла ранее сохраненной модели. Также файлы можно открывать методом

drag&drop. Описание функционального назначения кнопок графического редактора PetriNets InProject приведены в таблице 18.

Таблица 18

Вид	Назначение
	Кнопка для работы с позицией (вершиной) на рабочем поле. Размещение позиции задается щелчком левой клавиши мыши на пустой клетке рабочего поля. В нажатом положении щелчком левой клавишей мыши в позицию можно добавлять метки, правой – удалять. При наличии в позиции более двух меток их количество отображается числом
	Кнопка для работы с переходом на рабочем поле. Размещение перехода задается щелчком левой клавиши мыши на пустой клетке рабочего поля. В нажатом положении при щелчке левой клавишей мыши над переходом происходит смена положения (горизонтальная/вертикальная ориентация)
	Кнопка для работы с простой дугой. В нажатом положении на рабочем поле дуга единичного веса задается соединением позиции с переходом или перехода с позицией при удерживании нажатой левой клавиши мыши. Соединение дугой двух позиций или двух переходов между собой невозможно. Если необходимо задать дугу с весом более 1 для задания перехода типа n/m , то можно соединить элементы столько раз, каков требуемый вес, или изменить вес дуги в окне редактирования свойств элемента сети Петри, из которого она выходит
	Кнопка для работы с ингибиторной дугой. Режимы работы аналогичные режимам работы с простой дугой, но разрешенное направление данной дуги – только от позиции к переходу
	Кнопка для режима перемещения элементов. Для перемещения одного элемента нужно щелкнуть левой клавишей мыши над его клеткой и перемещать, удерживая клавишу нажатой. Для перемещения нескольких элементов необходимо выделить их рамкой. Для сброса выделения следует щелкнуть левой клавишей мыши на свободной клетке или правой клавишей на любом месте рабочего поля
	Кнопка для удаления выделенных элементов сети. Также удаление можно произвести нажатием клавиши <i>Delete</i> на клавиатуре. Осуществление удаления с удерживанием клавиши <i>Shift</i> избавляет от необходимости подтверждать удаление в окне диалога
	Кнопка включения режима редактирования свойств для элемента сети Петри. При нажатой кнопке щелчок левой клавиши мыши по позиции или переходу открывает окно редактирования свойств
	Сохранение графа сети Петри в графический файл формата «PNG»
	Запуск моделирования в пошаговом режиме. Перед запуском необходимо задать начальную (нулевую) разметку сети. В процессе моделирования происходит динамическая смена маркировок сети в результате срабатывания активизированных переходов, которые в момент активизации выделяются цветом
	Запуск моделирования в автоматическом режиме
	Остановка моделирования

Окна для редактирования свойств переходов и позиций имеют вид, показанный на рис. 130, 131.

The dialog box is titled "Переход - elem1". It contains the following fields and controls:

- Имя:** A text field containing "elem1".
- Приоритет:** A numeric field with a value of 2 and up/down arrows.
- Постоянное время срабатывания:** A numeric field with a value of 2 and up/down arrows.
- Вероятность:** A checkbox labeled "задать" is unchecked, followed by a numeric field with a value of 0.
- Выходные дуги:**
 - Тип:** A dropdown menu set to "Простая".
 - Target:** A dropdown menu showing "1 K elem4".
 - Стиль:** A dropdown menu set to "Изгиб вверх".
 - Вес:** A numeric field with a value of 1 and up/down arrows.
 - Buttons:** "Удалить" (Delete) and "Задать" (Set).
- Bottom Buttons:** "Да" (Yes) and "Нет" (No).

Рис. 130

The dialog box is titled "Позиция - elem0". It contains the following fields and controls:

- Имя:** A text field containing "elem0".
- Маркеры:** A numeric field with a value of 0 and up/down arrows.
- Выходные дуги:**
 - Тип:** A dropdown menu set to "Ингибиторная".
 - Target:** A dropdown menu showing "2 K elem1".
 - Стиль:** A dropdown menu set to "Нормальный".
 - Вес:** A numeric field with a value of 4 and up/down arrows.
 - Buttons:** "Удалить" (Delete) and "Задать" (Set).
- Bottom Buttons:** "Да" (Yes) and "Нет" (No).

Рис. 131

В окне редактирования свойств позиции задаются:

- ее имя (не более 20 символов) и начальное количество маркеров-меток (от 0 до 1000);
- для исходящих дуг при выборе их из списка – вес и стиль отображения (в виде прямой линии, ломаной с изгибом вверх или вниз);
- удаление отдельных исходящих дуг (при необходимости).

ВНИМАНИЕ! Изменение свойств *каждой* выходной дуги необходимо подтверждать нажатием кнопки *Задать* – при этом заданные изменения отображаются на графе сети. Для завершения редактирования и сохранения всех изменений используется кнопка *Да*.

Для изменения значений в полях ввода можно использовать:

- ввод числа вручную с клавиатуры;
- скроллинг мышью при нахождении курсора в поле ввода;
- клавиши управления курсором на клавиатуре (вверх/вниз);
- элементы управления, расположенные справа от поля ввода.

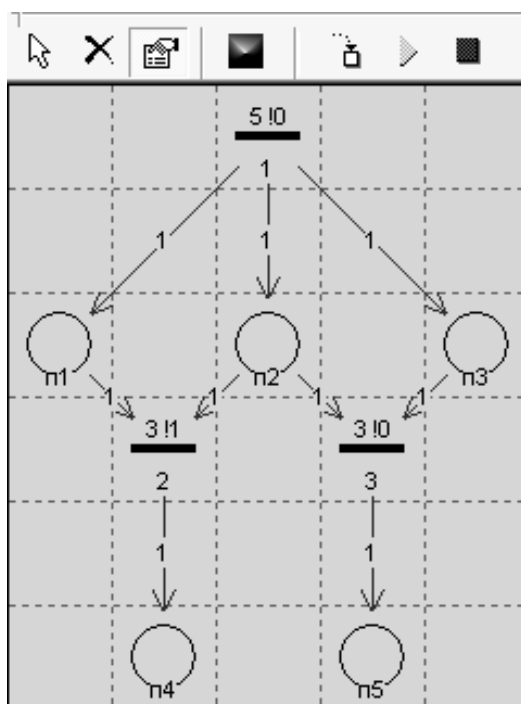
В окне редактирования свойств перехода, кроме того, задаются:

- времени срабатывания (от 1 до 1000 единиц модельного времени);
- значение приоритета (от 0 до 99) или вероятности срабатывания (от 0 до 1 с точностью до четырех знаков после запятой).

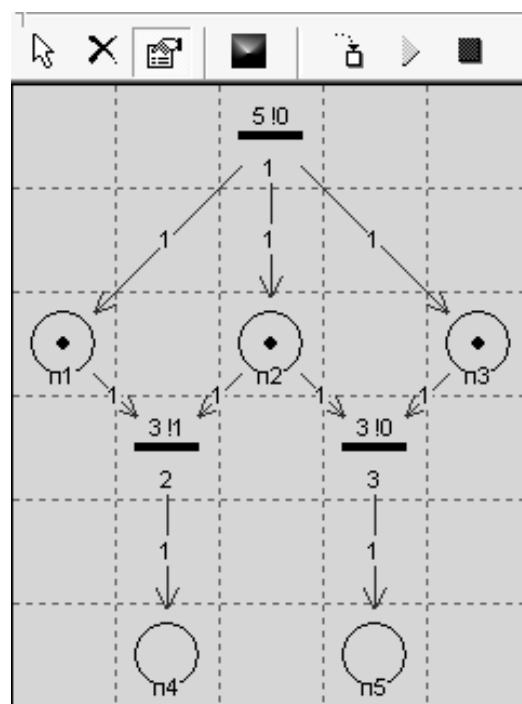
ВНИМАНИЕ! Для корректной работы вероятностной сети Петри в процессе разрешения конфликтов необходимо, чтобы сумма вероятностей срабатывания конфликтных переходов была равна 1.

Также следует отметить особенность разрешения конфликтов с использованием приоритетных переходов в PetriNets_InProject. Безусловный выбор перехода с бóльшим приоритетом осуществляется только в случае, если маркеры во входные позиции конфликтных переходов поступают *одновременно*. В этом случае активизируется приоритетный переход. Так, в примере, показанном на рис. 132, реализован фрагмент сети Петри с одновременной активизацией переходов с разными приоритетами. Задание бóльшего приоритета одного из переходов приводит к его безусловному срабатыванию в конфликте, что подтверждается пошаговым прогоном.

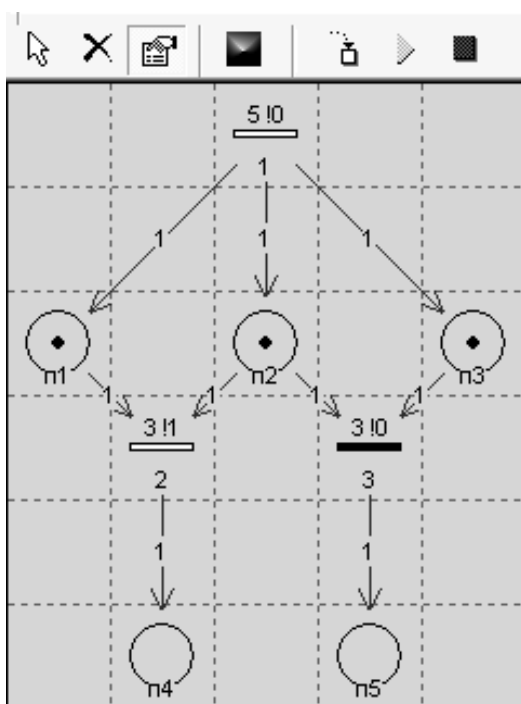
Если же в сети сначала происходит активизация перехода с низким приоритетом, т.е. начинается отсчет времени срабатывания этого перехода (время выполнения некоторого действия, связанного с переходом), а затем происходит изменение маркировки так, что создается ситуация конфликта с более приоритетным переходом, срабатывания более приоритетного перехода не произойдет, т.к. выполнение действия, связанного с ранее активизированным переходом уже началось.



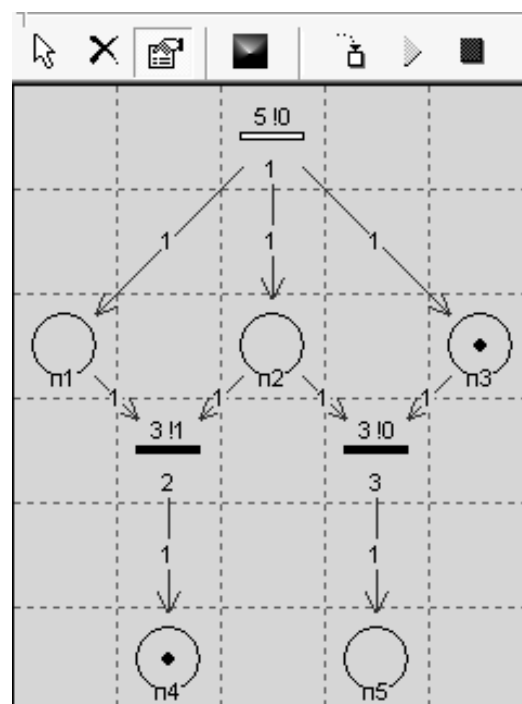
1. Запуск сети Петри.



2. После срабатывания перехода 1 метки одновременно поступают в позиции п. 1 – п. 3 возникает конфликт переходов 2 и 3.



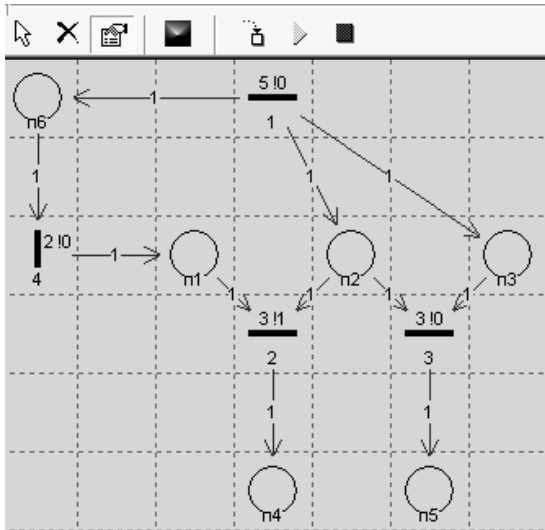
3. Конфликт разрешен в пользу приоритетного перехода 2 – он активизирован.



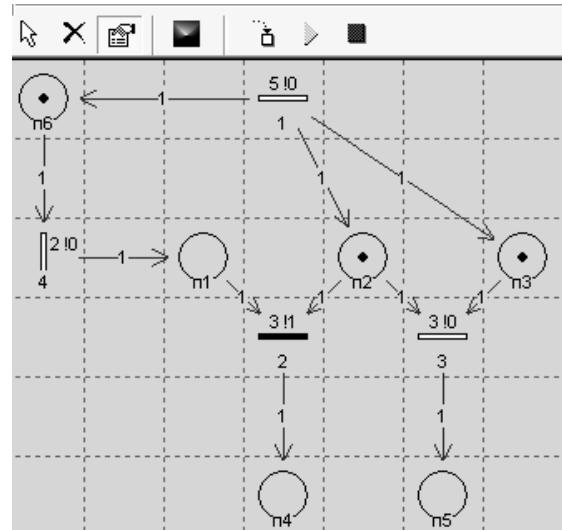
4. Результат срабатывания перехода 2.

Рис. 132

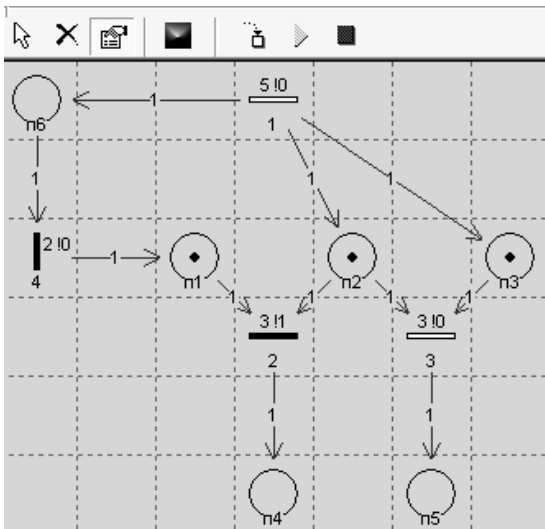
В случае одновременного поступления меток во входные позиции нескольких конфликтных переходов срабатывает тот переход, который был активизирован ранее. Так, например, в примере, показанном на рис. 133, реализован фрагмент сети Петри с *разновременной* активизацией переходов с разными приоритетами. Пошаговый прогон показывает, что в этом случае срабатывает тот переход, который был активизирован ранее, не зависимо от приоритета.



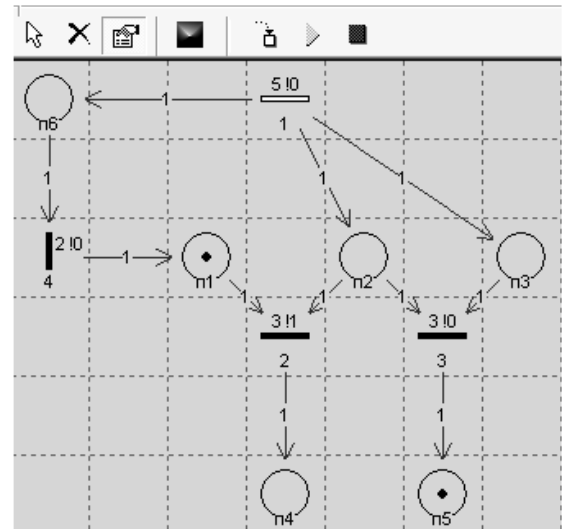
1. Запуск сети Петри.



2. Из-за одновременного прохождения меток по сети первым активизирован переход 3 (с низким приоритетом).



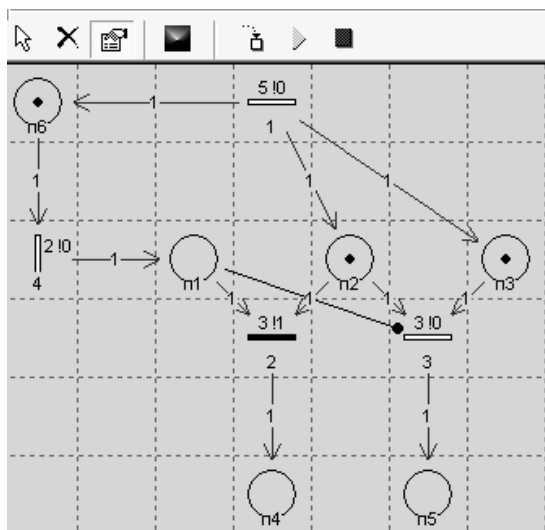
3. Метки имеются во всех входных вершинах конфликтных переходов. Но переход 3 уже активизирован, поэтому активизации приоритетного перехода 2 не происходит.



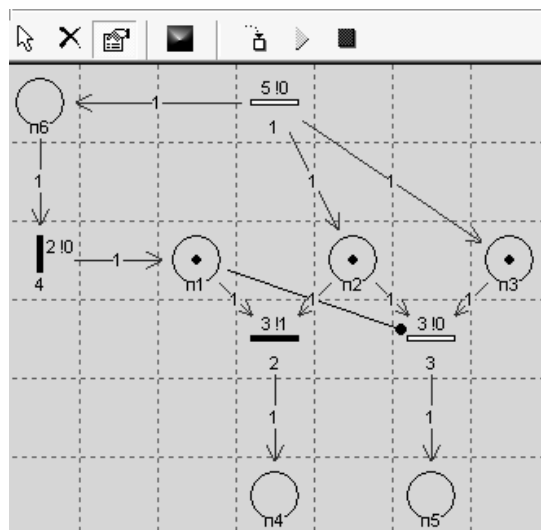
4. Результат срабатывания перехода 3.

Рис. 133

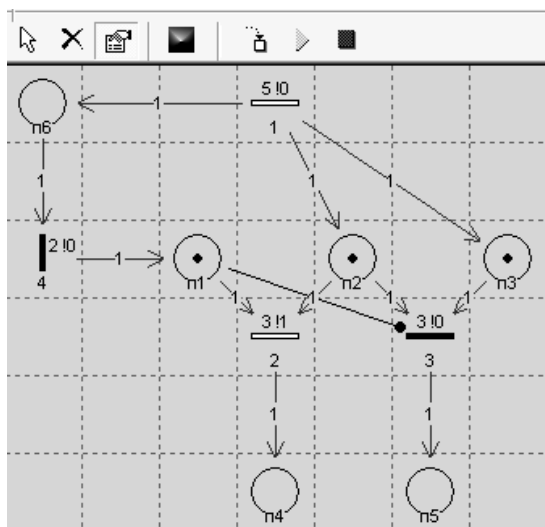
В случае, если при одновременном поступлении меток по входные вершины конфликтных переходов требуется прервать действие, связанное с менее приоритетным переходом, т.е. обеспечить безусловное первоочередное срабатывание перехода с бóльшим приоритетом, следует использовать ингибиторную ветвь, запирающую низкоприоритетный переход (рис. 134).



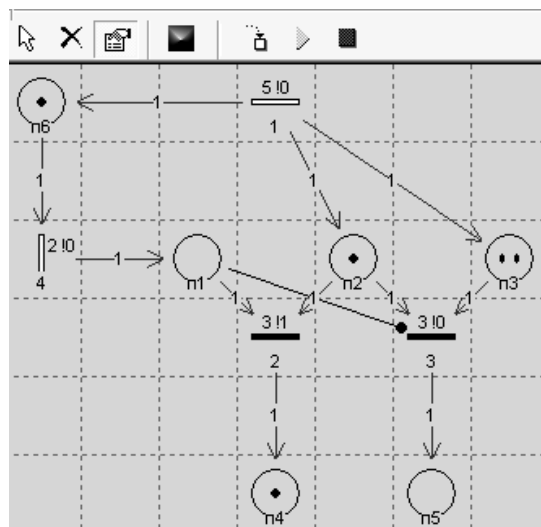
1. Из-за одновременного прохождения меток по сети первым активизирован переход 3 (с низким приоритетом).



2. Поступление метки в позицию **п1** за-
прещает срабатывание перехода 3.



3. Активизация перехода 3 отменена;
активизирован переход 2.



4. Сработал переход 2. Дополнительные метки в позициях **п6**, **п2** и **п3** появились в результате очередного срабатывания входного перехода 1.

Рис. 134

Пример 1

Моделирование динамики функционирования системы с параллельным осуществлением процессов

Любая элементарная операция, осуществляемая в процессе функционирования отдельным элементом системы, моделируется в терминах сетей Петри простейшим циклом обслуживания в том случае, если:

- операции выполняются по одной, в порядке очереди;
- выполнение операции не требует ресурсов, кроме времени осуществления, или ресурсы системы, расходуемые на выполнение операции, не ограничены.

Если элементарная операция функционирования в системе может осуществляться несколькими альтернативными элементами параллельного действия (устройствами обслуживания, подсистемами, узлами и т.п.), то каждый из таких элементов моделируется собственным циклом обслуживания. Пусть общий поток запросов, поступающих в вычислительную сеть с интервалом в 5 с, обслуживают два узла параллельного обслуживания, причем один из них тратит на обслуживание запроса 5 ± 1 с, а второй – 4 ± 1 с. Из общей входной очереди запросы передаются на обслуживание в любой из узлов за 1 с.

В PetriNets InProject возможно только детерминированное моделирование, т.е. величина времени срабатывания перехода является постоянной. Поэтому для исследования динамики работы системы необходимо реализовать равновероятное распределение времен путем использования нескольких переходов. Например, если время некоторого процесса составляет 4 ± 1 , то это моделируется тремя равновероятными переходами, имеющими время срабатывания 3, 4 и 5 с соответственно.

Сеть Петри, моделирующая заданную систему, показана на рис. 135 (см. файл *Параллельные узлы.pne*).

Приведенная маркировка сети Петри соответствует следующему состоянию сети: 9 запросов уже завершили обслуживание, причем 4 из них были обслужены в первом узле (4 метки в позиции «вых. очередь U1»), а 5 – во втором узле (5 меток в позиции «вых. очередь U2»). Предпоследний из поступивших в систему запросов был отправлен на обслуживание в первый узел. Последний из поступивших в систему запросов ожидает передачи на обслуживание во входной очереди (метка в позиции «входная очередь»).

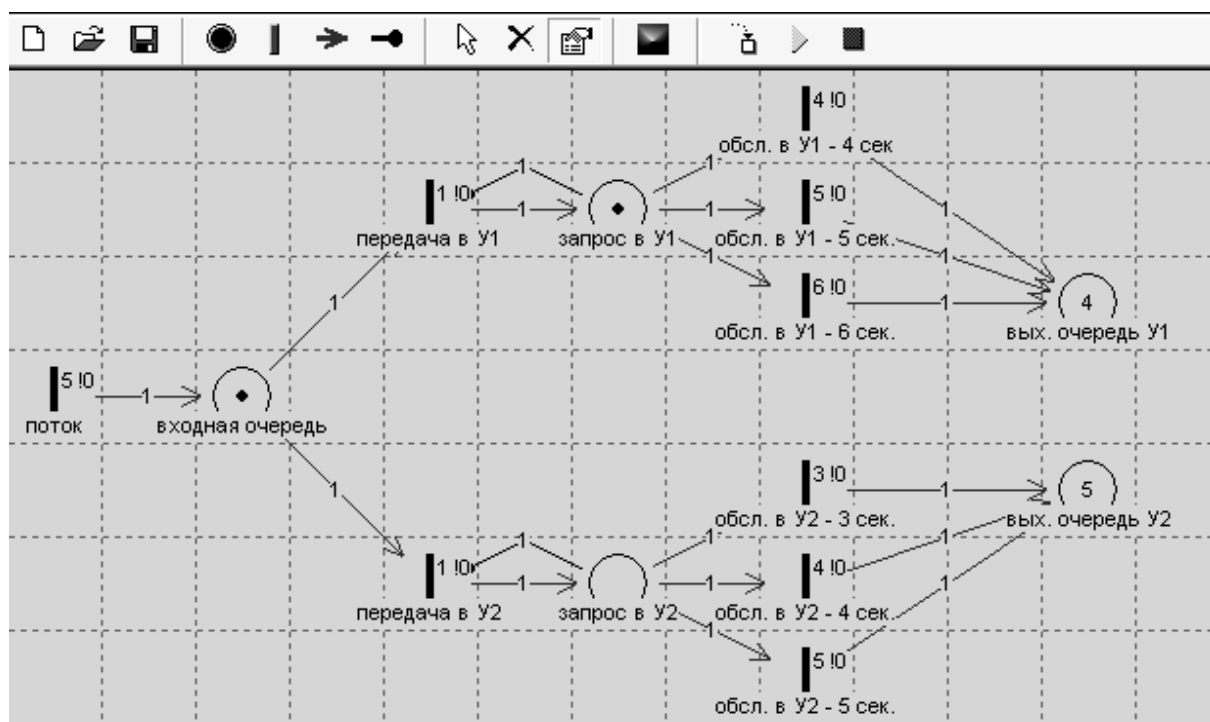


Рис. 135

Пример 2

Моделирование динамики функционирования системы с ограничением ресурса длины очереди

Пусть требуется смоделировать процесс обслуживания в системе 100 запросов входного потока, поступающих в систему с интервалом в 5 с. Для осуществления обслуживания одного запроса система расходует 2 единицы дополнительного ресурса. При наличии достаточного объема ресурса за 1 с осуществляется подготовка к обслуживанию, после чего обслуживание завершается в течение еще 1 с. Дополнительный ресурс не ограничен и поступает в систему поштучно из внешнего источника с интервалом в 4 ± 2 с. Число мест во входной очереди потока запросов ограничено и составляет 2 места. Запросы, не поместившиеся во входной очереди, теряются без обслуживания. Начальная маркировка сети Петри, моделирующей заданную систему, показан на рис. 136 (см. файл *Ограниченная очередь.pne*).

Число моделируемых запросов на обслуживание задано в исходной маркировке 100 метками в позиции «огр. 100 запросов». Для реализации заданного условия по ограничению входной очереди запросов используются позиция «входная очередь» и два конфликтных перехода с временем срабатывания 5 с – «поступление» и «потеря», причем первый из этих переходов имеет больший приоритет (равен 1). Величина приоритета отображается числом после восклицательного знака возле перехода; по умолчанию равна 0.

Ингибиторная дуга, исходящая из позиции «*входная очередь*», имеет вес 2 и «запрещает» срабатывание приоритетного перехода «*поступление*» при заполнении входной очереди запросов. Только в этом случае, из-за низшего приоритета, срабатывает переход «*потеря*» и метка, соответствующая запросу, поступившему в систему в условиях заполнения входной очереди и потерянного по этой причине, появляется в позиции «*число потерь*».

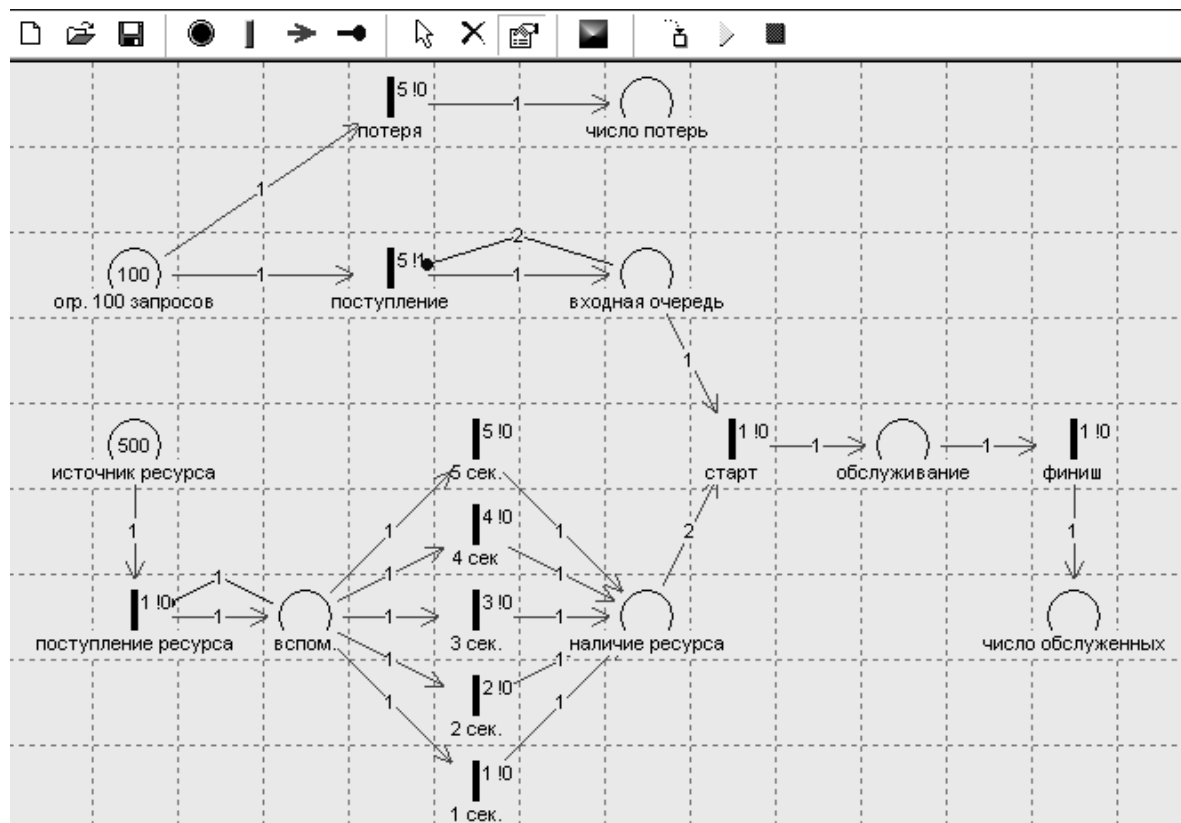


Рис. 136

Дополнительный ресурс поступает в систему неограниченно, поэтому в предлагаемой реализации его источник задан позицией с начальной маркировкой, гарантирующей безусловную достаточность единиц ресурса для обслуживания заданного числа запросов (500). Время поступления ресурса в систему, равное 4 ± 2 с, моделируется одним вспомогательным переходом с постоянным временем срабатывания, равным 1 с, и пятью переходами для задания равномерного распределения времени. Общий вспомогательный переход с позицией, «запирающей» его ингибиторной дугой, необходим для моделирования последовательного поступления ресурса в систему. Без такой реализации пять равновероятных переходов срабатывали бы независимо друг от друга, как пять асинхронных источников меток.

Для начала (подготовки) к обслуживанию запроса входной очереди необходимо наличие 2-х единиц дополнительного ресурса, поэтому из по-

зиции «наличие ресурса» в переход «старт» идет дуга с весом, равным 2, т. е. данный переход будет активизирован только при условии наличия 1 запроса во входной очереди (одна метка в соответствующей позиции) и не менее 2-х меток в позиции «наличие ресурса».

Конечная маркировка данной сети Петри, полученная в режиме автоматического прогона на некотором интервале времени, имеет вид (рис. 137).

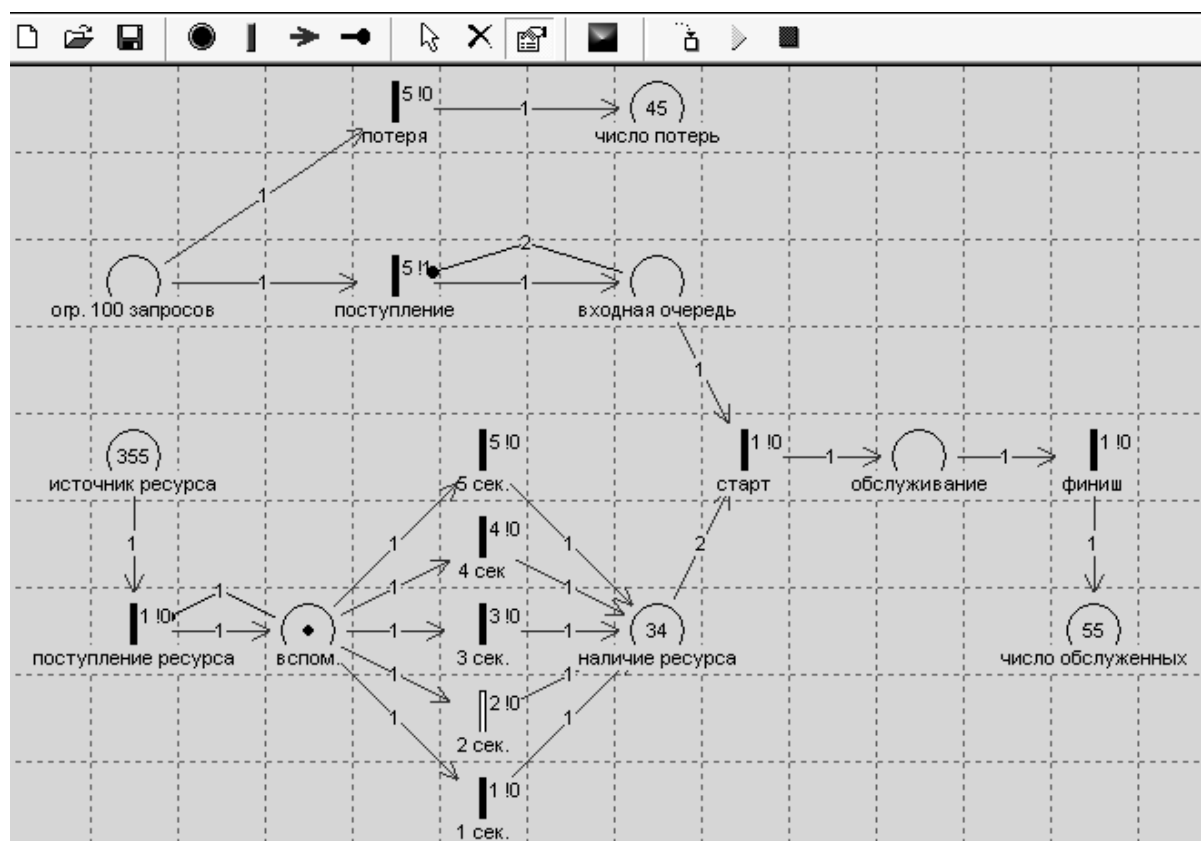


Рис. 137

Результат моделирования показывает, что из 100 запросов были обслужены только 55 – по количеству меток в выходной позиции «число обслуженных». Оставшиеся 45 запросов были потеряны по причине того, что на момент их поступления входная очередь была заполнена. На корректное обслуживание запросов было истрачено 110 единиц дополнительного ресурса, что подтверждается остаточным наличием в количестве 390 единиц: 355 меток в позиции «наличие ресурса»; 1 метка во вспомогательной позиции и 34 метки в позиции «наличие ресурса». Такая маркировка обусловлена тем, что после завершения обслуживания заданных 100 запросов до остановки процесса моделирования поступления ресурса из источника продолжалось без его израсходования.

Пример 3

Моделирование динамики функционирования вероятностной системы с ограничением используемого ресурса

Задана система, выполняющая два типа работ по запросам соответствующего типа. Запросы каждого типа поступают в систему через 3 ± 1 с. Для выполнения работ по запросу каждого типа расходуется единица ресурса, запас которого в системе ограничен 100 единицами. Время выполнения запроса с расходом ресурса составляет 1 с, завершение выполнения – еще 1 с. Конфликт запросов с вероятностью 0,8 разрешается в пользу запроса первого типа, т.е. при одновременном поступлении запросов он с заданной вероятностью выполняется в первую очередь. С остаточной вероятностью в первую очередь выполняется запрос второго типа. После израсходования ресурса работа системы прекращается. Начальная маркировка сети Петри, моделирующей заданную систему, показана на рис. 138 (см. файл *Ограниченный ресурс.pne*).

Начальный запас ресурса моделируется начальной маркировкой сети – 100 меток в позиции «ресурс». Правило разрешения конфликтов запросов моделируется двумя конфликтными переходами «*start1*» и «*start2*» с общей входной позицией «ресурс». В случаях одновременного поступления запросов двух типов в соответствующие очереди с вероятностью 0,8 срабатывает один из конфликтных переходов и с остаточной вероятностью 0,2 – второй. В результате срабатывания любого из конфликтных переходов одна метка появляется в позиции, соответствующей передаче запроса конкретного типа на завершение обслуживания (позиции «*в процессе 1*» и «*в процессе 2*» соответственно), и еще одна метка появляется в позиции – счетчике израсходованных ресурсов. По факту накопления в позиции «*израсходовано*» 100 меток, исходящие ингибиторные ветви «запирают» переходы «*тип 1*» и «*тип 2*», имитирующие поступление запросов соответствующих типов.

Вариант конечной маркировки сети, полученной в режиме автоматического прогона, имеет вид (рис. 139).

Как следует из конечной маркировки, в процессе функционирования системы с ограниченным запасом расходуемого ресурса и вероятностным разрешением конфликта очередности обслуживания запросов двух типов, были выполнены работы по 65 запросам первого типа и 35 запросам второго типа. При этом на момент исчерпания ресурса в очередях невыполненных запросов первого и второго типов остались, соответственно, один и 24 запроса. Следует отметить, что ввиду вероятностной природы сети Петри конечные маркировки, полученные в результате разных прогонов для заданной начальной маркировки, будут различаться в пределах допустимых статистических отклонений.

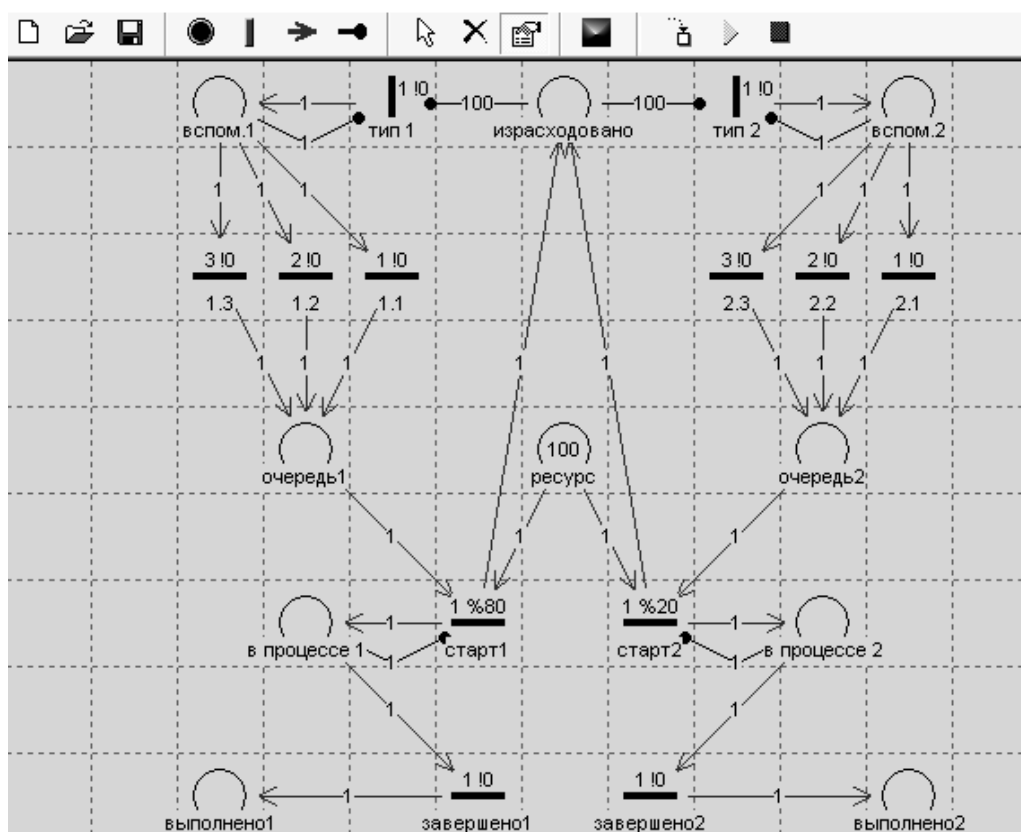


Рис. 138

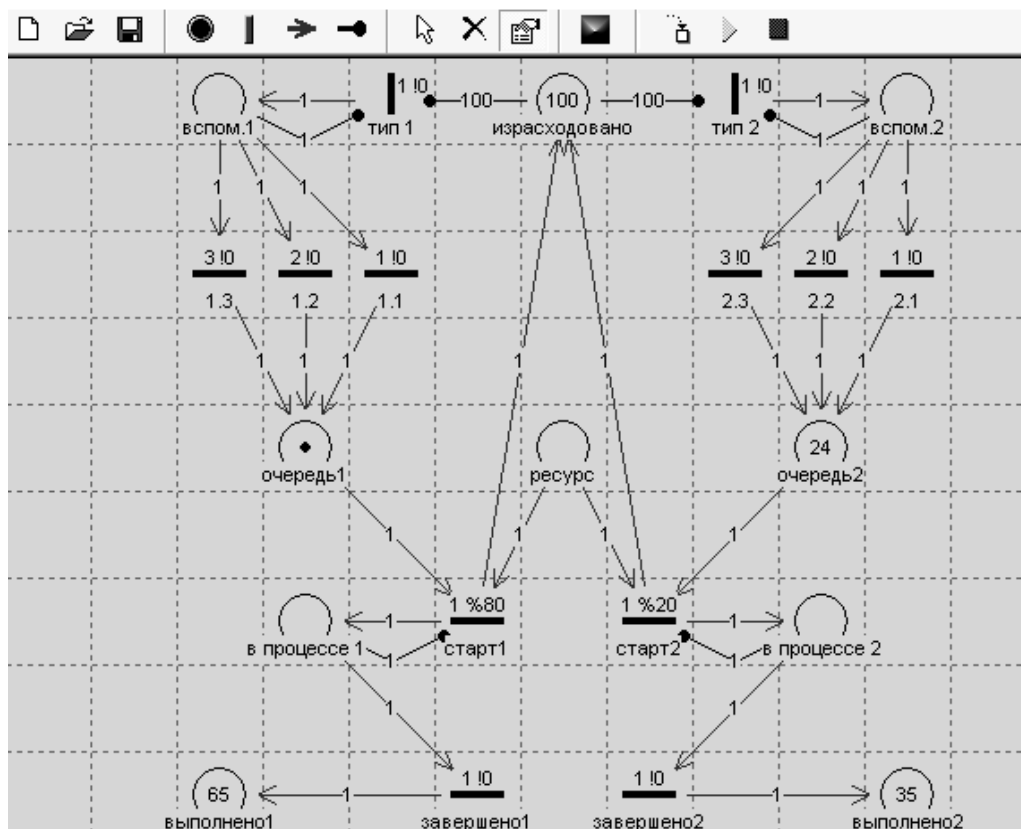


Рис. 139

Варианты индивидуальных заданий

1. Согласно плану системой должны быть выполнены 50 единичных объемов работ первого типа и 30 единичных объемов работ второго типа. На выполнение единичного объема работ любого типа затрачивается единица ресурса А и 2 единицы ресурса В. Доступный объем ресурса А составляет 65 единиц; ресурса В – 140 единиц. Согласно условиям функционирования сначала выполняется 10 единичных объемов работ первого типа, после чего работы обоих типов выполняются по очереди. Время выполнения работы с расходом ресурса составляет 1 с, завершение выполнения – еще 3 ± 2 с. Работы в системе прекращаются после исчерпания любого из ресурсов.

2. Согласно плану системой должны быть выполнены по 20 единичных объемов работ первого и второго типа. На выполнение единичного объема работ любого типа затрачивается по одному единичному объему ресурса А и ресурса В. Доступный объем ресурса А составляет 45 единиц; ресурса В – 40 единиц. Время выполнения любой работы с расходом ресурса составляет 1 с, завершение выполнения работы первого типа – еще 3 ± 2 с; работы второго типа – еще 3 ± 1 с. Работы выполняются параллельно и независимо друг от друга. После завершения выполнения плановых объемов работ по остаточному объему неизрасходованного ресурса производится дополнительный заказ такого же количества другого ресурса из внешнего неограниченного источника.

3. В систему поступают два независимых потока запросов на выполнение работ двух типов. Запросы первого типа поступают через 6 с, второго типа – через 5 ± 2 с, и обслуживаются в порядке поступления в очередь. При одновременном поступлении запросов, запрос первого типа принимается к обслуживанию в первую очередь. Выполнение работ по запросу первого типа требует расходования 2 единиц ресурса А; второго типа – единицы ресурса В. Время выполнения работ по запросу любого типа с расходом ресурса составляет 1 с, завершение выполнения – еще 1 с. Запас ресурса А в системе составляет 100 единиц; ресурса В – 120. По факту исчерпания ресурса А система перестает принимать запросы первого типа. По факту исчерпания ресурса типа В из внешнего неограниченного источника в систему дополнительно поставляется 20 единиц ресурса, после исчерпания дополнительной поставки система перестает принимать запросы второго типа.

4. Система непрерывно выполняет плановые работы, каждая из которых состоит из 3-х операций, выполняемых независимо друг от друга на

3-х параллельных устройствах обслуживания. Время подготовки выполнения любой операции составляет 1 с, завершение выполнения операций – еще 3 ± 1 с. Работа считается выполненной после завершения всех операций. После завершения очередной работы через 3 с начинается выполнение следующей. С периодичностью 40 ± 2 с в системе возникает отказ одного из устройств, равновероятно любого. Время устранения отказа занимает 10 с. Во время недоступности устройства, операция, которая должна выполняться отказавшим устройством, выполняется резервным устройством, для которого время подготовки выполнения любой операции составляет 2 с, завершение выполнения операций – еще 4 ± 1 с. После наступления в системе 15-го отказа ее функционирование прекращается.

5. В систему через 4 ± 1 с поступает потока запросов на выполнение работ. Выполнение работ по запросу требует расходования 2 единиц ресурса. Время выполнения работ по запросу с расходованием ресурса составляет 1 с, завершение выполнения – еще 1 с. Доступный объем ресурса составляет 100 единиц. После исчерпания ресурса для выполнения каждого очередного запроса из двух дополнительных источников неограниченного объема выполняется поставка необходимого числа единиц ресурса, причем с вероятностью 0,75 поставка осуществляется из первого дополнительного источника, а с остаточной вероятностью – из второго. При этом, во втором дополнительном источнике хранится такая модификация ресурса, которая требует увеличения времени завершения работы до 4 ± 1 с. Работа системы прекращается после выполнения 80 запросов.

6. В систему через 4 ± 1 с поступают запросы на выполнение двух последовательных операций обслуживания. Для первой операции за 1 с осуществляется подготовка к обслуживанию, после чего обслуживание завершается в течение еще 1 с. Выполнение второй операции может быть осуществлено за 2 с на любом из двух параллельных устройств обслуживания с общей очередью и требует расходования 2 единиц ресурса. У каждого устройства имеется собственный запас ресурса, который для первого устройства составляет 25 единиц; для второго – 38 единиц. После исчерпания ресурса одного из устройств второе продолжает обслуживание; после исчерпания всех ресурсов система прекращает прием запросов. При этом все запросы, уже поступившие в систему, т.е. находящиеся в устройствах и/или очередях, поступают в буфер незавершенных работ.

7. В систему через 6 ± 2 с поступают запросы на выполнение двух последовательных операций обслуживания. Параметры обеих операций следующие: за 1 с осуществляется подготовка к обслуживанию, после чего обслуживание завершается в течение еще 3 ± 1 с. Очередь на выпол-

нение первой операции ограничена и может содержать только 1 запрос на обслуживание. Если нет места в очереди первой операции – запрос покидает систему без обслуживания. В системе организован счетчик запросов, обслуженных с минимально возможным временем выполнения обеих операций. Работа системы прекращается после выполнения 25 операций.

8. В систему поступает поток ресурса, причем очередная поставка может содержать 3 ± 2 единицы ресурса. Поставки происходят через 10 с. Всего запланировано 50 поставок. Параллельно система принимает поток запросов, поступающих через 3 ± 1 с. Запрос равновероятно имеет тип 1 или 2. Запросы предварительно накапливаются во входной очереди; после поступления в очередь 5 запросов прием запросов прекращается до полного завершения работ по уже имеющимся запросам. Время выполнения работ по запросу с расходом ресурса составляет 1 с, завершение выполнения – еще 1 с. Для выполнения работ по запросу типа 1 расходуется 2 единицы ресурса; по запросу 2 – одна единица.

9. В кластерную систему, состоящую из 3-х устройств обслуживания, с интервалом в 3 ± 1 с поступают задачи. Поступившая задача разбивается на части по числу доступных устройств системы или ожидает обслуживания в очереди длиной не более 2-х задач, если доступных устройств нет. Если очередь заполнена, вновь поступающие задачи теряются. Направление части задачи на обслуживание занимает 1 с, обслуживание части в устройстве – еще 4 ± 2 с. Функционирование системы прекращается после завершения первым устройством 45-го обслуживания.

10. Согласно плану система должна выполнить по 20 единичных объемов работ 2-х типов. Работа каждого типа состоит в выполнении 2-х операций. Первая операция каждого типа работ выполняется на собственном устройстве обслуживания, причем подготовка выполнения операции занимает 1 с, завершение выполнения – для первой работы 3 ± 1 с, для второй – 4 ± 1 с. Для выполнения второй операции в каждой работе в системе имеется общее устройство с временем выполнения 2 с и расходом на работу первого типа 2-х единиц, а на работу второго типа – 3-х единиц общего ресурса. В случае конфликта одновременности в первую очередь выполняется вторая операция по работе первого типа. Объем внутрисистемного запаса ресурса составляет 50 единиц. В случае уменьшения запаса до 5 единиц система формирует заказ на пополнение ресурса еще на 50 единиц из неограниченного внешнего источника. Время формирования заказа составляет 1 с, время поставки ресурса – 20 с.

11. Система непрерывно выполняет плановые работы. Каждая работа требует выполнения операций с общей суммарной длительностью не менее 6 с. Операции выполняются на 3-х устройствах обслуживания; первое устройство обеспечивает обслуживание длительностью 1 с, второе – 2 с, третье – 3 с. Работа в цикле направляется равновероятно на любое из устройств до тех пор, пока суммарная длительность обслуживания не станет больше либо равно 6 с. После этого работа считается выполненной. После завершения очередной работы через 2 с начинается выполнение следующей. Функционирование системы прекращается после выполнения 40 работ.

12. В состав системы входит 3 устройства обслуживания, соединенных по кольцу. В момент запуска в системе находится одна заявка, которая циркулирует по системе. Передача заявки по каналу связи между устройствами занимает 1 с, обслуживание в каждом устройстве 3 ± 1 с. В отдельный накопитель с внешним потоком в систему поштучно поступает дополнительный ресурс с интервалом поступления 5 ± 1 с. Если на момент поступления заявки в какое-либо устройство в системе в наличии имеется 2 единицы ресурса, то это количество расходуется в процессе обслуживания и на обслуживание в этом случае затрачивается 1 с. Функционирование системы прекращается после израсходования 40 единиц ресурса.

13. Заявки на обслуживание поступают в систему с интервалом 5 с, проходят первую операцию обслуживания в первом устройстве за 1 с и равновероятно направляются на вторую операцию обслуживания в одно из трех параллельных устройств. Направление на вторую операцию обслуживания занимает 1 с, обслуживание – еще 1 с. Системной политикой в узле параллельных устройств запрещено использование два раза подряд одного и того же устройства: если, например, обслуживание предыдущей заявки было выполнено устройством 1 этого узла, то следующая заявка равновероятно направляется в устройство 2 или 3. Функционирование системы прекращается в момент поступления в систему 85-ой заявки.

14. Согласно плану системой должны быть выполнены по 20 единичных объемов работ первого и второго типа, выполнение планового объема работ первого типа является приоритетным. На выполнение единичного объема работ любого типа затрачивается по одному единичному объему ресурса А и равновероятно один или два единичных объема ресурса В. Доступный объем ресурса А составляет 35 единиц; ресурса В – 55 единицы. Время выполнения любой работы с расходом ресурса составляет 1 с, завершение выполнения – еще 3 ± 1 с. В системе одновременно может выполняться один единичный объем работы первого типа или

два объема работы второго типа – в этом случае время завершения выполнения одинаково для обоих объемов.

15. В вычислительную систему через 6 с поступают задачи. Для каждой задачи создается 2 копии, после чего задача и ее копии направляются в три параллельных устройства. Создание копии и направление на обслуживание занимает по 1 с, решение задачи в устройстве – еще 3 ± 1 с. На выходе из системы реализован счетчик решенных задач, причем значение счетчика увеличивается на 1 в момент завершения первого из 3-х обслуживаний. «Лишние» решения удаляются после их завершения. Функционирование системы прекращается после завершения решения 50-ти задач.

16. В систему с интервалом в 3 ± 1 с поступают запросы на выполнение работ. Работа состоит в последовательном выполнении 3-х операций, причем первая и третья операция сопровождается расходом единичного объема ресурса. Общий запас ресурса в системе составляет 60 единиц. Время выполнения операции с расходом ресурса составляет 1 с, завершение выполнения – еще 4 ± 1 с. Время выполнения второй операции составляет 5 с. Поступление запросов в систему приостанавливается после поступления каждого 5-го запроса и возобновляется после того, как работа по последнему из поступивших запросов будет полностью завершена.

17. В кластерную систему, состоящую из 3-х устройств обслуживания, с интервалом в 10 секунд поступают запросы на обслуживание. Синхронно с запросом поступает команда, которая с вероятностью 0,3 задает индивидуальный режим обслуживания запроса, и с остаточной вероятностью – режим предварительного разделения запроса на 2 части. Запрос в индивидуальном режиме и части запроса в режиме разделения равновероятно направляются на любое из устройств системы. Время направления составляет 1 с, обслуживание – еще 3 ± 1 с. На выходе системы части разделенного запроса объединяются. Функционирование системы прекращается после поступления в систему 20-ой команды на разделение запроса.

18. В систему поступают три независимых потока запросов на выполнение работ. Интервал поступления запросов каждого потока: 5 ± 1 с. Запросы первого типа являются первоочередными. Запросы второго и третьего типа в случае одновременного поступления или накопления в очередях могут попарно объединяться и далее выполняться как один запрос. Выполнение работы по запросу первого типа сопровождается расходом единичного объема ресурса типа А; по запросам второго и третьего типов, а также парного запроса – ресурса типа Б. Время выполнения работы по запросу с расходом ресурса составляет 1 с, завершение выполнения – еще 2 ± 1 с. Запасы ресурсов типа А и Б равны и составляют

35 единиц. В случае исчерпания одного из видов ресурса осуществляется поштучный обмен единичных объемов оставшегося ресурса на требуемый. Функционирование системы прекращается после исчерпания ресурсов.

19. Согласно плану системой должны быть выполнены 20 единичных объемов работ первого типа и 30 единичных объемов работ второго типа. Для выполнения единичного объема работы первого типа необходимо 2 единицы ресурса типа А и 3 единицы ресурса типа В; для второго типа – 3 единицы ресурса типа А и 2 единицы ресурса типа В. Ресурсы каждого типа поступают в систему поштучно отдельным потоком в собственный накопитель. Интервал поступления ресурса типа А – 4 ± 1 с, типа В – 4 ± 2 с. Работы выполняются системой в порядке, соответствующем накоплению достаточного количества ресурсов каждого типа. Если в какой-то момент времени количество ресурсов в накопителях системы оказывается достаточным для выполнения обоих типов работ, приоритетом первоочередности обладает работа второго типа. Время выполнения работы с расходом ресурса составляет 1 с, завершение выполнения – еще 2 ± 1 с.

20. В систему с периодичностью 8 ± 2 с поступают запросы на выполнение единичного объема работ. Для выполнения единичного объема работы необходимо по единице ресурсов типа А и В. Ресурсы каждого типа поступают в систему поштучно отдельным потоком в собственный накопитель. Интервал поступления каждого ресурса – 3 ± 1 с. Поступление ресурсов каждого типа прекращается после 60-ой единицы ресурса. Емкости накопителей ограничены 3-мя местами хранения. При заполнении накопителя поступающие ресурсы перенаправляются в дополнительные накопители за 2 с. Время выполнения работы с расходом ресурса из основного накопителя составляет 1 с, завершение выполнения – еще 2 ± 1 с. По исчерпании ресурсов из основных накопителей работы продолжают с использованием дополнительных накопителей. Доставка ресурса из дополнительного накопителя занимает 2 с.

21. Система непрерывно выполняет плановые работы, каждая из которых состоит из 3-х операций, выполняемых независимо друг от друга на 3-х параллельных устройствах обслуживания. Время подготовки выполнения любой операции составляет 1 с, завершение выполнения операций 1 и 2 – еще 3 ± 1 с, операции 3 – еще 3 ± 2 с. Работа считается выполненной после завершения всех операций. После завершения очередной работы через 2 с начинается выполнение следующей. С периодичностью 30 ± 2 с в систему поступают запросы на выполнение внеплановой работы. Дополнительная работа выполняется вне очереди, требует выполнения только операции 1 и расходования 2-х единиц ресурса. Расходование ресурса происходит за

время подготовки выполнения операции. Внутрисистемный источник содержит 20 единиц ресурса. Функционирование системы прекращается после исчерпания ресурса.

22. В систему с периодичностью 4 ± 1 с поступают запросы на выполнение единичного объема работ, каждая из которых состоит из операций: с вероятностью 0,6 работа требует выполнения 4-х операций, с остаточной вероятностью – 3-х. Операции выполняются на 2-х параллельных устройствах обслуживания. Если работа требует выполнения 4-х операций, каждое из устройств выполняет по очереди 2 операции. Если работа требует выполнения 3-х операций, то, после выполнения 2-х операций двумя устройствами, 3-я выполняется на любом из устройств. Время подготовки выполнения операции составляет 1 с, завершение выполнения операций – еще 2 ± 1 с. Работа считается выполненной после завершения всех операций. После завершения очередной работы выполнение следующей начинается после перерыва длительностью 1 с. Функционирование системы прекращается после выполнения 50-ти работ.

23. В систему с периодичностью 5 ± 1 с поступают запросы на формирование комплекта из ресурсов наименований А, В и С (по одному единичному объему), и с периодичностью 5 ± 2 с запросы на формирование комплекта из ресурсов наименований В, С и D (по два единичного объема). Запросы удовлетворяются со строгим чередованием: первый тип, второй тип, первый тип и т. д. Время подготовки комплектования составляет 1 с, собственно комплектование – еще 4 ± 1 с. Если в очереди запросов любого типа скапливаются два запроса, то поступление запросов этого типа прерывается на 8 с. Внутрисистемные источники ресурсов всех типов неограниченны. После обслуживания первых 30-ти запросов одного типа поступление запросов этого типа прекращается. Функционирование останавливается после завершения комплектования 30-ти запросов другого типа.

24. В систему с интервалом в 5 ± 1 с поступают запросы на выполнение работ. Работа состоит в последовательном выполнении 3-х операций, причем первая и третья операция сопровождается расходом единичного объема ресурса. Общий запас ресурса в системе составляет 60 единиц. Время доставки ресурса из накопителя составляет 1 с, выполнение операции с расходом – еще 4 ± 1 с. Единовременно в системе может осуществляться только одна доставка ресурса; операции с расходом ресурса могут выполняться одновременно. Время выполнения второй операции составляет 3 с. Поступление запросов в систему прекращается после исчерпания ресурса.

Лабораторная работа № 6

Моделирование систем массового обслуживания сетями Петри

Цель: моделирование многоканальных систем массового обслуживания неоднородных потоков заявок с потерями.

Программное обеспечение: PetriNets InProject.

Примеры выполнения работы: файлы *Пример 1 (ограничение времени ожидания).rne*, *Пример 2 (приоритетное обслуживание).rne*, *Пример 3 (отказ и остановка по условию).rne*.

Теория

Основные сведения по моделированию систем различной природы сетями Петри приведены в разделе 2.5 *Непрерывно-дискретное моделирование систем сетями Петри*.

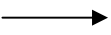
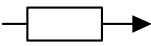
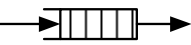
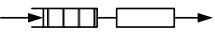
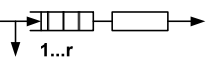
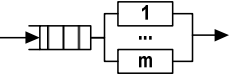
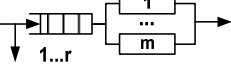
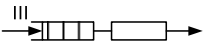
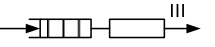
В теории массового обслуживания изучаются системы, на вход которых поступает случайный поток заявок, т.е. поток заявок характеризуется законом распределения случайной величины – интервала времени между поступлением «соседних» заявок. Поступившая заявка обслуживается в системе путем предоставления ей некоторых ресурсов на какое-то время и, будучи в той или иной мере обслуженной, покидает систему. Помимо случайного появления заявок на обслуживание и случайной длительности обслуживания каждой заявки для систем массового обслуживания характерным является наличие очередей, в которых заявки ждут момента освобождения ресурсов, занятых обслуживанием других заявок. Такие системы называются *системами массового обслуживания (СМО)*.

События, происходящие в ВС, носят случайный характер, поэтому для их моделирования используются модели теории массового обслуживания. В системах, моделируемых в виде СМО, различают *статические* и *динамические* объекты. Статические объекты – обслуживающие аппараты (ОА) – моделируют средства обработки информации (аппаратные и программные), при использовании которых происходят затраты времени на выполнение операций обслуживания. Динамические объекты – заявки (запросы, требования) – моделируют решаемые в ВС задачи. Поток заявок физически представляет собой явления одной природы, например попытки модемного соединения, запросы к базе данных и т.д.

СМО бывают одно- и многоканальными в зависимости от числа ОА параллельной обработки; одно- и многофазными в зависимости от числа последовательно включенных ОА.

В структурных схемах СМО используются обозначения и узлы, приведенные в табл.19.

Таблица 19

Наименование	Графическое обозначение	Описание
Поток заявок		Поток заявок (запросов, требований, транзактов и т.п.)
Обслуживающий аппарат		Устройство обслуживания. Моделирует затраты времени на выполнение отдельной операции по обслуживанию
Очередь обслуживающего аппарата		В очереди заявки, поступающее со входным потоком, ожидают освобождения ОА от обслуживания ранее поступивших заявок
Одноканальная СМО с ожиданием		Один ОА с бесконечной очередью
Одноканальная СМО с ограниченной очередью		Один ОА с конечным числом мест в очереди. Если число заявок превышает число мест в очереди, то лишние заявки теряются
Многоканальная СМО с ожиданием		Несколько параллельно работающих ОА с общей бесконечной очередью
Многоканальная СМО с ограниченной очередью		Несколько параллельно работающих ОА с общей очередью, число мест в которой ограничено
Одноканальная СМО с групповым поступлением заявок		Один ОА с бесконечной очередью. Заявки поступают в очередь сгруппированными в пакеты по определенному правилу
Одноканальная СМО с групповым обслуживанием заявок		Один ОА с бесконечной очередью. Заявки обслуживаются пакетами, составляемыми по определенному правилу

Потери заявок в процессе обслуживания в СМО могут иметь место по причине:

- возникновения отказов ОА или ошибок в процессе обслуживания;
- ограничения количества мест в очереди;
- ограничения времени ожидания (времени с момента поступления до момента выхода заявки из очереди);
- ограничения времени пребывания (времени с момента поступления до момента выхода заявки из системы или некоторого ее узла; склады-

вается из времен ожидания и обслуживания заявки в процессе прохождения по системе);

- приоритетного обслуживания (при поступлении заявки с бóльшим абсолютным приоритетом ранее начатое обслуживание прерывается, и заявка с прерванным обслуживанием теряется или после завершения приоритетного обслуживания прерванное обслуживание завершается/начинается заново).

В случае относительного приоритета прерывания начатого обслуживания заявки с низким приоритетом не происходит, но заявки с бóльшим приоритетом принимаются к обслуживанию после завершения начатого в первую очередь.

Согласно теории вероятностей, если некоторое случайное событие A наступает как следствие какого-либо из n_A событий при общем числе n возможных событий (несовместимых и равновероятных), то вероятностью события A называют число: $P_A = \lim_{n \rightarrow \infty} \frac{n_A}{n}$. Невозможному событию соот-

ветствует вероятность 0, достоверному – вероятность 1. Вероятность любого иного события имеет значение между 0 и 1. По результатам статистических испытаний, полученных имитационным моделированием на достаточно большом числе прогонов, вероятностные характеристики функционирования могут быть рассчитаны как частота наступления соответствующего события.

Ход выполнения работы

Дано: структурная схема СМО с описанием потоков и особенностей обслуживания (приоритетов заявок различных типов, вероятностей распределения заявок на обслуживание между параллельными ОА или каналами, условий пакетирования и т. п.).

1. По варианту индивидуального задания разработать сеть Петри, моделирующую заданную СМО, и реализовать ее в пакете PetriNets InProject. В отчете представить:

- начальную (нулевую) маркировку сети;
- описание всех позиций (вершин) сети с точки зрения условий, которые с ними связаны (см. пример в табл. 9 в разделе 2.5);
- описание всех переходов с точки зрения связанных с ними процессов и времен их осуществления (см. пример в таблице 10 в разделе 2.5).

2. Осуществить моделирование функционирования СМО в автоматическом режиме и режиме пошагового прогона. В отчете представить:

- схемы маркировок сети Петри и их описание для *всех значимых ситуаций* функционирования СМО, полученные в режиме пошагового прогона в PetriNets InProject;

- схему конечной маркировки сети Петри, соответствующую конечному состоянию системы, если такое состояние возможно по условиям функционирования системы.

3. Осуществить моделирование функционирования СМО в автоматическом режиме с остановкой моделирования по заданному условию. По результатам моделирования оценить вероятность потерь заявок в системе по каждой из возможных причин.

4. Результаты выполнения работы по пп. 1 – 3 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

Практические рекомендации по выполнению работы и использованию программного обеспечения

Основные навыки работы с программным обеспечением PetriNets InProject, примеры моделирования систем с ограниченной очередью и приоритетного обслуживания – см. *Лабораторную работу № 5*. Особое внимание следует обратить на особенности моделирования приоритетных процессов в PetriNets InProject в случаях одно- и разновременного поступления меток во входные вершины конфликтных переходов с разными приоритетами, рассмотренными в разделе *Основные принципы работы с пакетом PetriNets InProject* методических указания к *Лабораторной работе № 5*.

Пример 1

Моделирование устройства СМО с ограничением времени ожидания

Пусть устройство СМО обслуживает однородный поток заявок – все заявки в потоке имеют тип **A**. Время поступления заявок составляет 3 с. Время передачи заявки на обслуживание из очереди в устройство занимает 1 с, обслуживания заявок в устройстве составляет 3 ± 2 с. Время ожидания обслуживания заявок типа **A** в очереди устройства ограничено 10-ю секундами, при нарушении ограничения заявки теряются.

Сеть Петри, моделирующая заданный узел СМО, имеет вид, показанный на рис. 140 (см. файл *Пример 1 (ограничение времени ожидания).pne*).

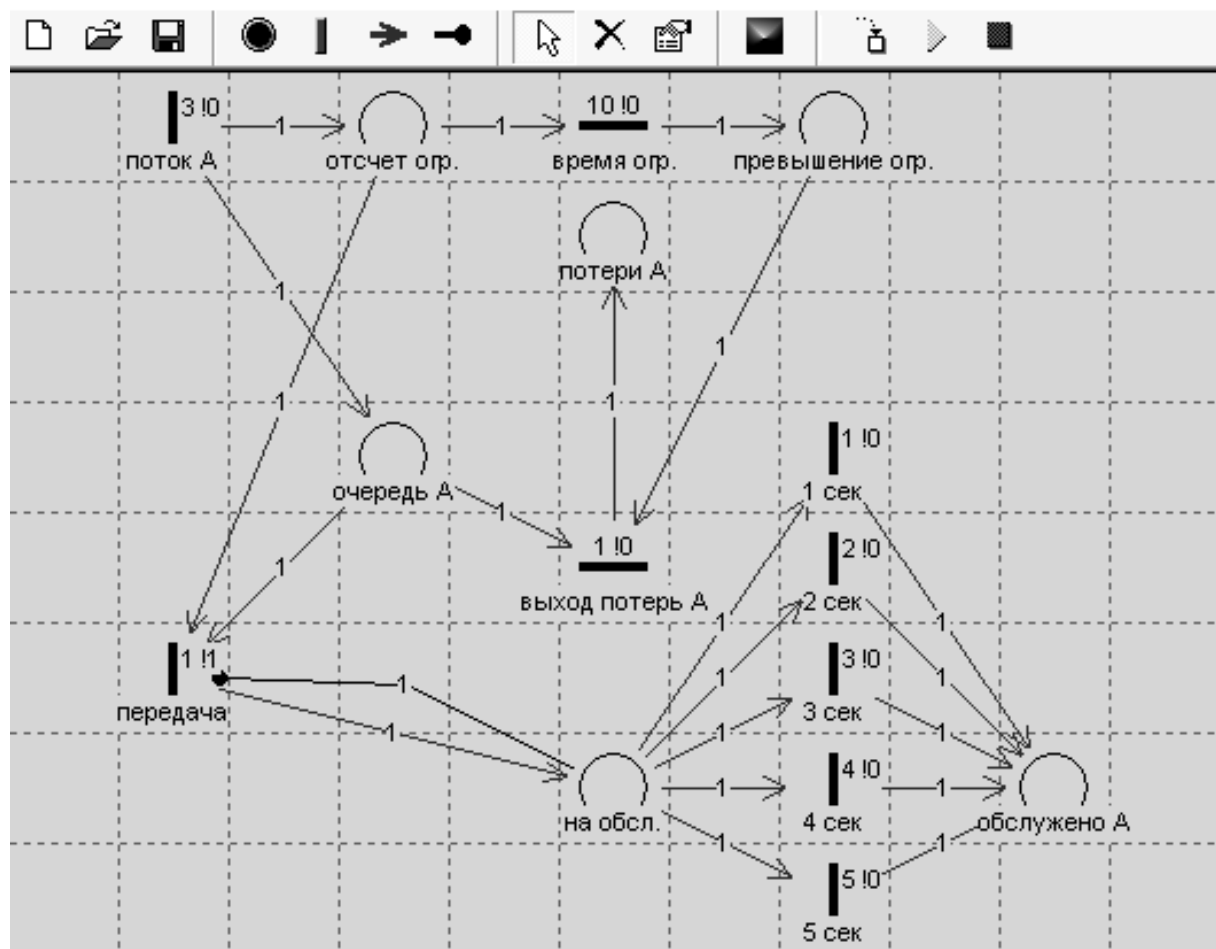


Рис. 140

Переход «*поток А*» моделирует поступление заявок с интервалом в 3 с. Одновременно с поступлением метки-заявки в позицию «*очередь А*» метка-инициатор отсчета времени появляется в позиции «*отсчет огр.*» и активизирует переход «*время огр.*», с которым связано ограничение времени ожидания. Если время ожидания (время нахождения метки-заявки в позиции «*очередь А*») превысит 10 с, то в результате срабатывания перехода «*время огр.*» метка появится в позиции «*превышение огр.*», что приведет к активации перехода «*выход потерь А*». В результате срабатывания перехода метка-заявка из позиции «*очередь А*» и метка-признак превышения времени из позиции «*превышение огр.*» удаляются, и метка поступает в позицию-счетчик потерь. Маркировка, соответствующая ситуации превышения времени ограничения имеет вид, приведенный на рис. 141.

Видно, что предыдущее обслуживание еще не завершено, а время пребывания метки-заявки в позиции «*очередь А*» уже превысило ограни-

чение: метка отсчета времени в результате срабатывания перехода «*время огр.*» переместилась в позицию «*превышение огр.*». В результате активирован переход «*выход потерь А*», и, в результате его срабатывания метки из входных позиций будут удалены, а счет потерь увеличится на 1 (рис. 142).

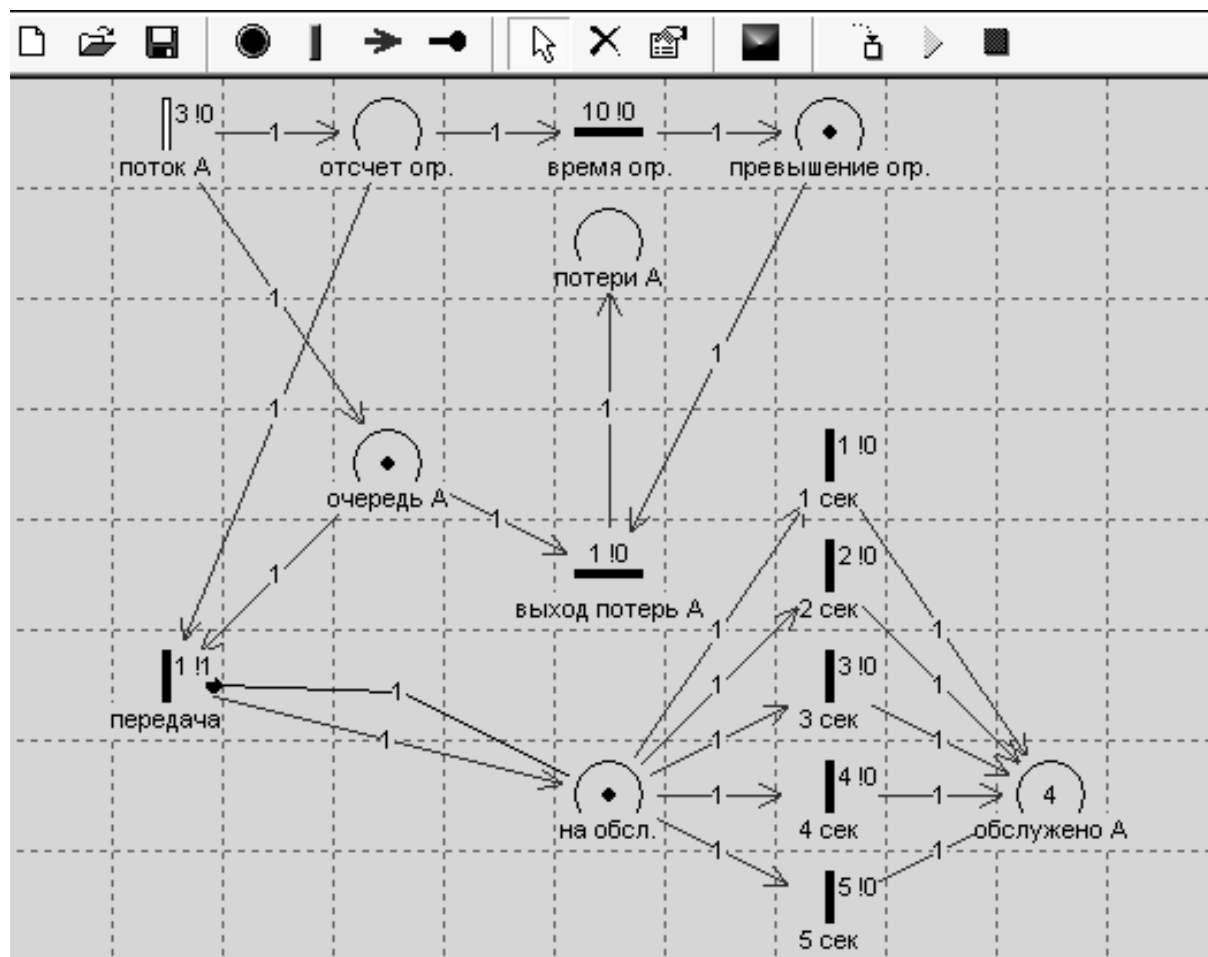


Рис. 141

Если ограничение не нарушается и заявка принимается к обслуживанию до истечения времени ограничения, т. е. после завершения предыдущего обслуживания, то метка, соответствующая предыдущей заявке, удаляется из позиции «*на обл.*» и активизируется переход «*передача*», до сих пор «запертый» ингибиторной ветвью. В этом случае происходит сброс отсчета времени ограничения: метка-инициатор отсчета удаляется из позиции «*отсчет огр.*» в результате срабатывания переход «*передача*». Переходу «*передача*» присвоен больший приоритет для разрешения конфликта с переходом «*время огр.*».

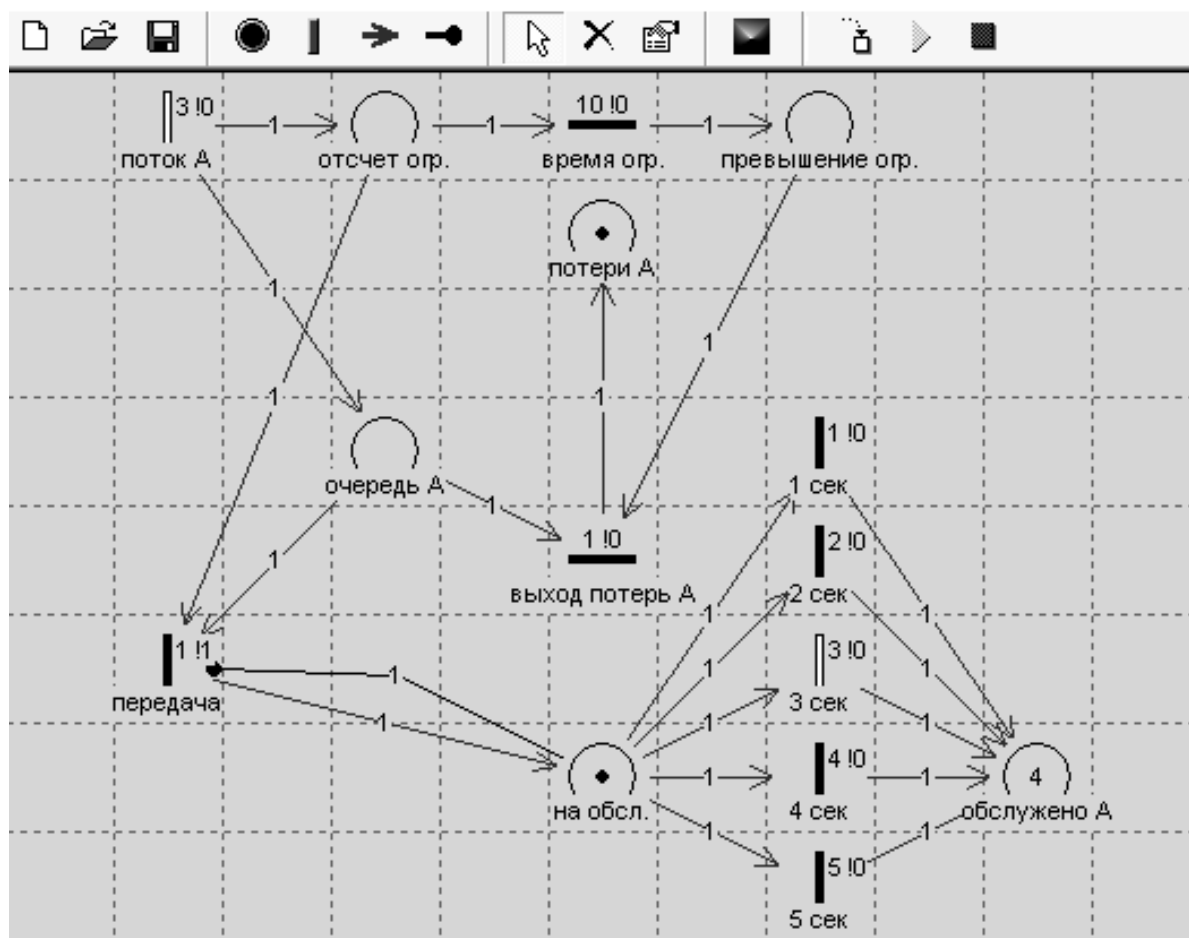
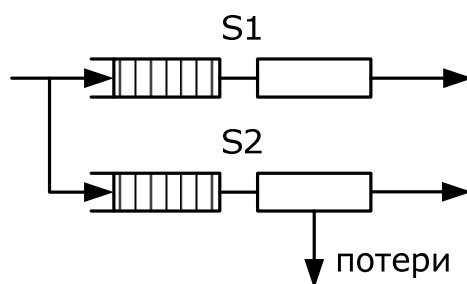


Рис. 142

Пример 2

Моделирование узла СМО с неоднородным входным потоком и абсолютным приоритетом обслуживания

Пусть узел СМО задан структурной схемой вида:



На вход системы поступает поток заявок двух типов (А и В) с вероятностями поступления 0,6 и 0,4 соответственно. Время поступления составляет, равновероятно, 5 или 6 с. Заявки типа А распределяются на обслуживание следующим образом: 70% – в устройство S1; 30% – в S2. Все

заявки типа В обслуживаются в S2 и пользуются абсолютным приоритетом: если при поступлении на обслуживание заявки этого типа в S2 имеется ранее начатое обслуживание заявки типа А, то оно прерывается, а заявка теряется. Время выполнения промежуточных операций в СМО (передача из очереди на обслуживание, передача между каналами) составляет 1 с, собственно обслуживание в устройствах – 2 с.

Сеть Петри, моделирующая заданный узел СМО, реализованная в файле *Пример 2 (приоритетное обслуживание).rne*, имеет вид, показанный на рис. 143. Для моделирования равновероятного распределения времени поступления заявок входного потока использован переход «*вх.поток*» с временем срабатывания 1 с и вспомогательный узел распределения меток между переходами с разным временем срабатывания. Распределение заявок входного потока по типам (из позиции «*неопр. тип*») и распределение заявок типа А по каналам реализовано с использованием парных вероятностных переходов «*поступление А*»/«*поступление В*» и «*передача А в S1*»/«*передача в канал*» соответственно. Вероятности срабатывания задаются в свойствах перехода в долях, а отображаются рядом с обозначением перехода в процентах.

Наличие приоритетных заявок типа В, ожидающих обслуживания или уже обслуживаемых в устройстве S2 (метки в позициях «*вх. очередь В*» и «*приор. В в S2*» соответственно), должно блокировать передачу заявок типа А на обслуживание в S2. Эта блокировка реализована ингибиторными ветвями из названных вершин к переходу «*передача А в S2*». Если же на момент поступления в очередь приоритетного обслуживания заявки типа В обслуживание заявки типа А в S2 уже началось, то это обслуживание должно быть прервано. Для этого, при срабатывании перехода «*передача А в S2*» создается дополнительная метка-флаг начатого не-приоритетного обслуживания и помещается в позицию «*непр. обл. А*». При появлении метки-заявки в позиции «*вх. очередь В*» ингибиторной ветвью блокируется переход «*S2 1 сек. А*» и активизируется переход, моделирующий прерывание, в результате срабатывания которого заявка типа В незамедлительно начинает обслуживаться – соответствующая метка появляется в позиции «*приор. В в S2*», а заявка типа А теряется. Маркировка сети Петри, предшествующая прерыванию не-приоритетного обслуживания, имеет вид, показанный на рис. 144.

На следующем шаге в пошаговом прогоне срабатывает переход «*прер.*», и маркировка принимает вид, показанный на рис. 145.

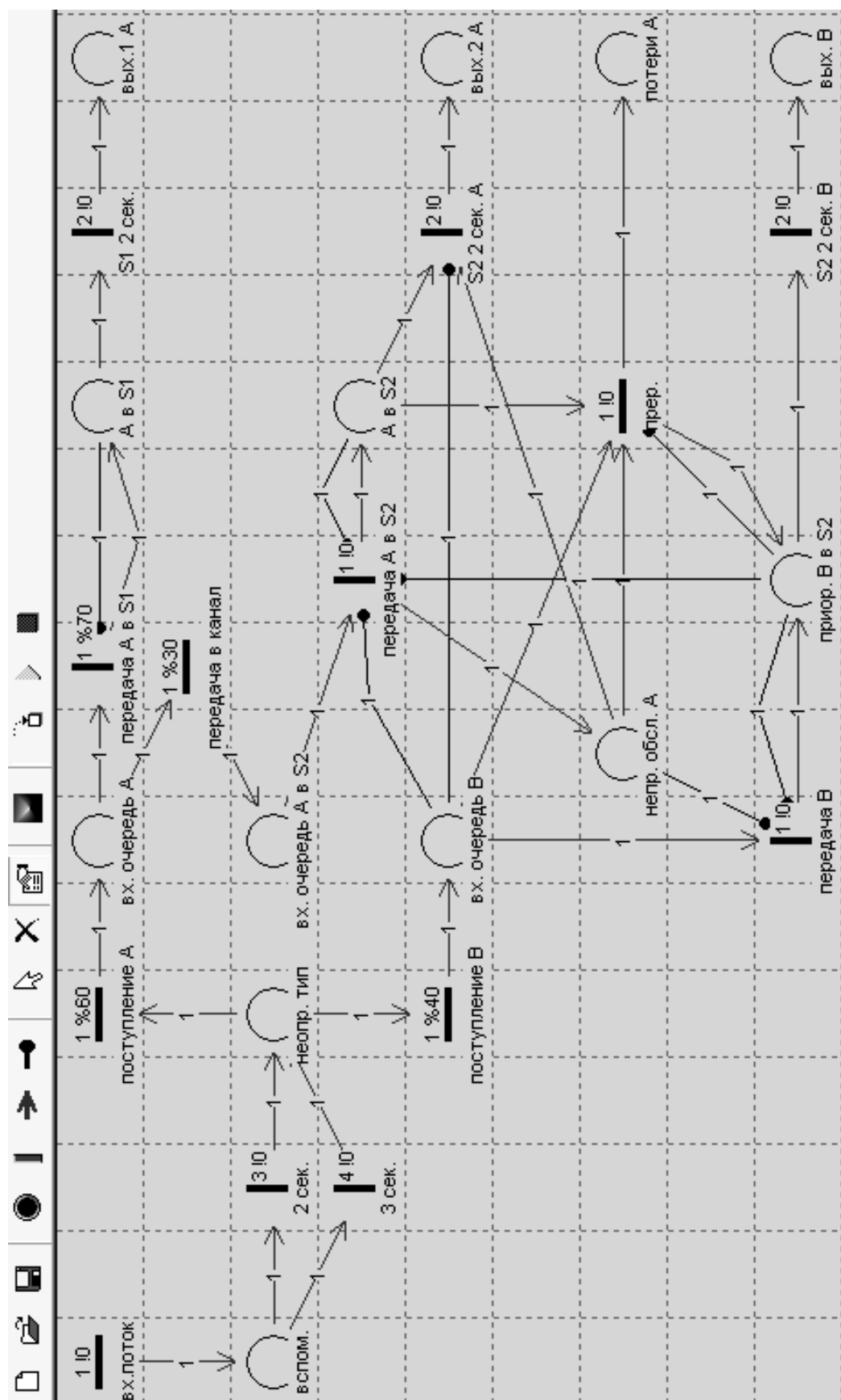


Рис. 143

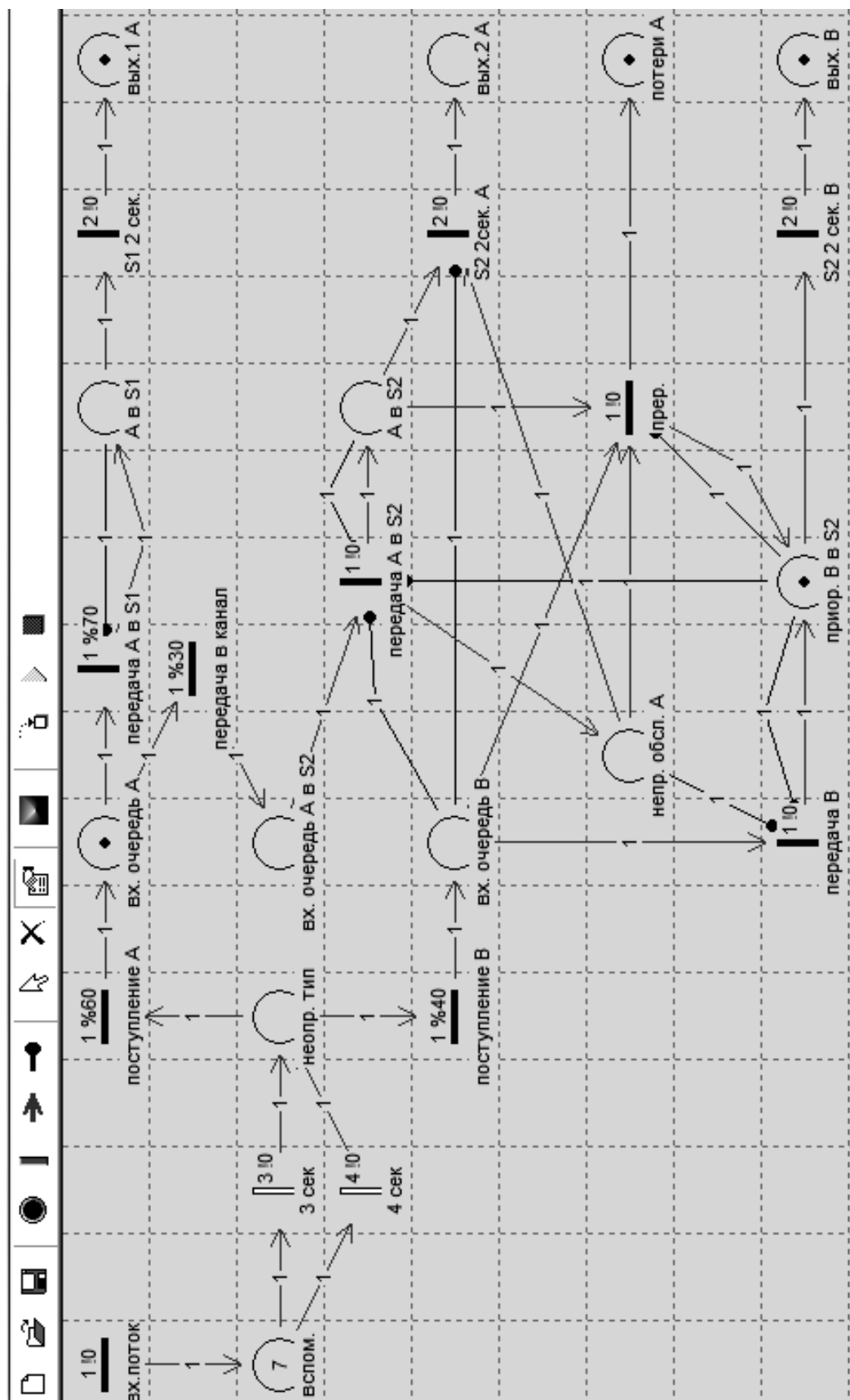


Рис. 145

Если же начатое неприоритетное обслуживание заявки типа А завершается в штатном режиме, без прерывания, то метка-флаг «сбрасывается» при срабатывании перехода «S2 1 сек. А».

Пусть для заданного узла требуется реализовать остановку моделирования по факту поступления с входным потоком 100-ой заявки и оценить вероятность потери заявок типа А по конечной статистике моделирования. Конечная маркировка сети Петри, дополненной блокировкой перехода-генератора входного потока ингибиторной ветвью от позиции-счетчика поступивших заявок, после остановки моделирования по условию поступления 100-ой заявки имеет вид, показанный на рис. 146.

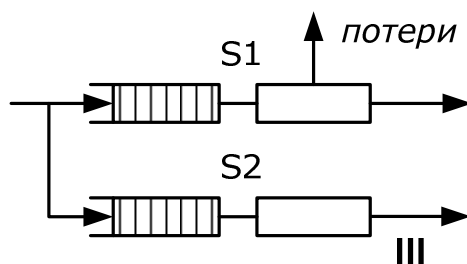
Как следует из конечной маркировки, на момент поступления в систему 100-той заявки завершили обслуживание 47 заявок типа В, $33+9=42$ заявок типа А и обслуживание 6-ти заявок типа А было прервано из-за поступления более приоритетной заявки. Кроме того, на момент остановки моделирования обслуживание одной заявки типа В не завершилось (метка в позиции «*приор. В в S2*»), и по две заявки типа А ожидало обслуживания в очередях обоих устройств. Таким образом, число заявок типа А покинувших систему, составляет $N_A = 42 + 6 = 58$. На основании полученной статистики рассчитаем вероятность потери заявки типа А в рассматриваемом узле СМО:

$$P_{\text{потери } A} = \frac{N_{\text{потерь } A}}{N_A} = \frac{6}{58} = 0,103.$$

Пример 3

Моделирование узла СМО с отказом основного устройства и остановкой по условию

Пусть узел СМО задан структурной схемой вида:



На вход СМО поступает однородный входной поток с периодичностью поступления 2 ± 1 с. Приоритетным является обслуживание заявок в устройстве S1, но, если устройство занято, то заявки входного потока перенаправляются в S2, где объединяются для обслуживания в пакеты по 2. С периодичностью 50 ± 1 с происходит отказ устройства S1.

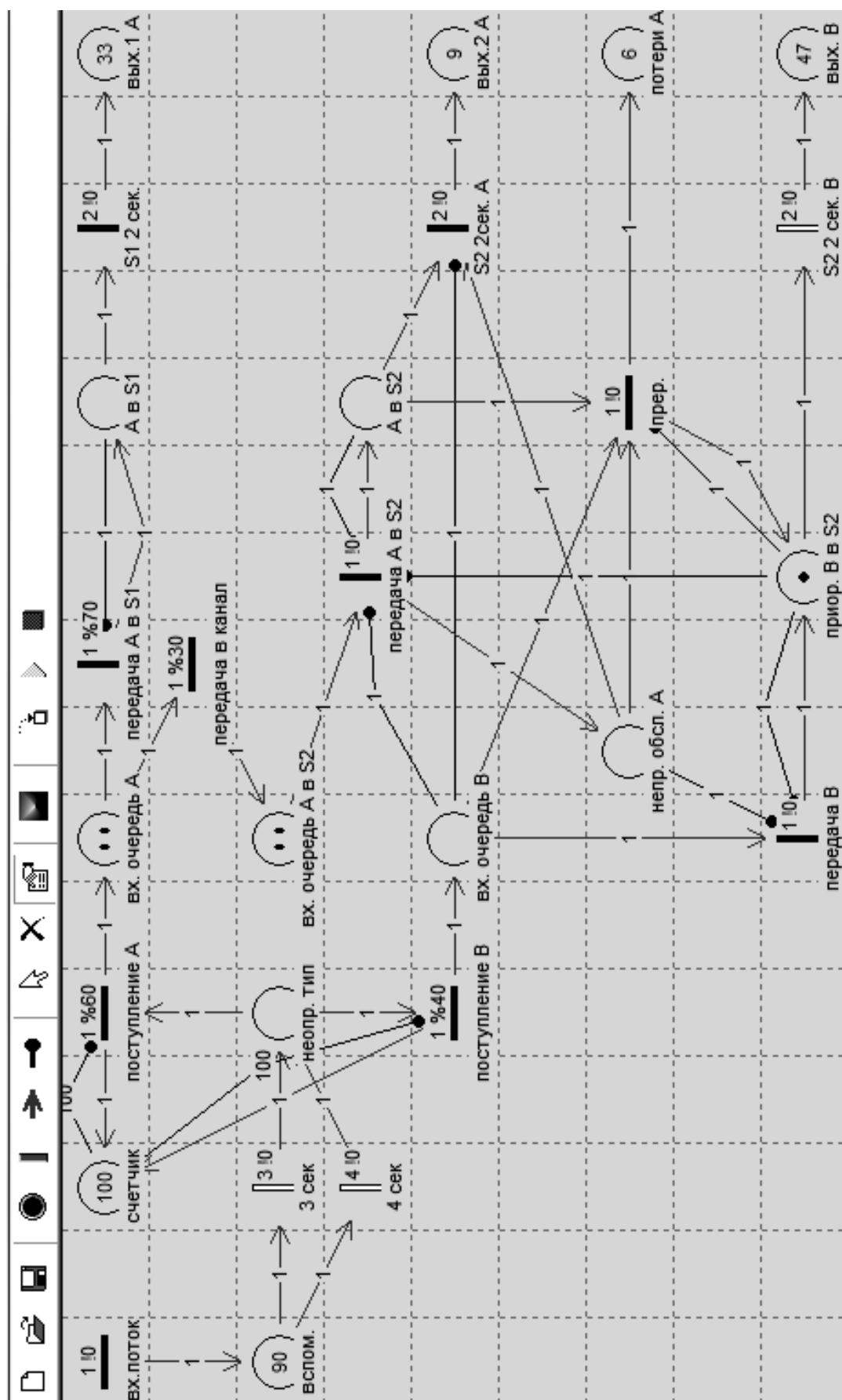


Рис. 146

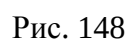
Заявка, обслуживание которой было прервано отказом, теряется. Во время восстановления S1, которое занимает 15 с, все заявки входного потока обслуживаются в S2. Время обслуживания в S1 составляет 4 с, в S2 – 1 с, время передачи заявок и пакетов на обслуживание из очередей в устройство и между каналами занимает 1 с. Требуется осуществить моделирование заданной СМО за 300 с поступления заявок входного потока и оценить вероятность потерь заявок.

Начальная маркировка сети Петри, моделирующая заданный узел СМО, имеет вид, показанный на рис. 147. (см. также файл *Пример 3 (отказ и остановка по условию).pne*).

Особенностью реализации источника входного потока и источника отказов является использование меток-инициаторов, возвращающихся в соответствующую позицию после срабатывания промежуточных переходов. Для реализации условия остановки моделирования через 300 с использован переход, генерирующий метку останова, «запирающую» ингибиторными ветвями переходы в узле реализации входного потока. В нормальном режиме метки-заявки поступают в очередь S1, и, если это устройство не занято, принимаются к обслуживанию, а противном случае, через переход «в канал» перенаправляются в очередь S2. Для разрешения конфликта в случае одновременной активизации переходов «в S1» и «в канал» первому присвоен больший приоритет.

В результате срабатывания любого из переходов в узле реализации потока отказов генерируются две метки, поступающие, соответственно, в позиции «отказ» и «прерывание». Удаление метки-флага отказа по истечении времени восстановления работоспособности устройства S1 происходит через переход «восст.1», и, если на момент возникновения отказа устройство S1 было свободно от обслуживания, то через парный переход «восст.2» происходит удаление метки-флага прерывания. Из позиции «отказ» выходит ингибиторная дуга, запрещающая срабатывание перехода «в S1» во время отказа. Если на момент возникновения отказа в S1 выполняется обслуживание (метка в позиции «обсл. S1»), то моделирование прерывания будет обеспечено срабатыванием перехода «потеря», активизированного благодаря наличию метки-флага прерывания в позиции «прерывание». В этом случае для обеспечения безусловного срабатывания перехода потеря «потеря» конфликтные переходы «восст.2» и «S1 4 сек.» блокируются ингибиторными ветвями от входных позиций перехода «потеря» – «обсл. S1» и «прерывание».

Конечная маркировка рассматриваемой сети Петри после остановки моделирования по условию истечения 300 с модельного времени имеет вид, показанный на рис. 148.

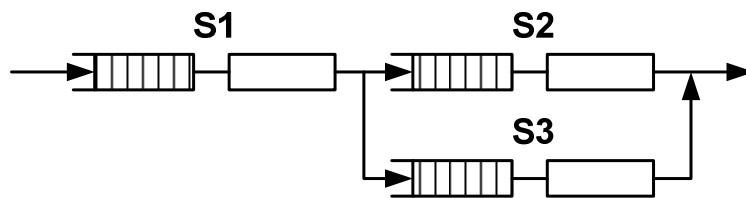
$$P_{номеру} = \frac{N_{номеру}}{N} = \frac{4}{145} = 0,0276.$$


Варианты индивидуальных заданий

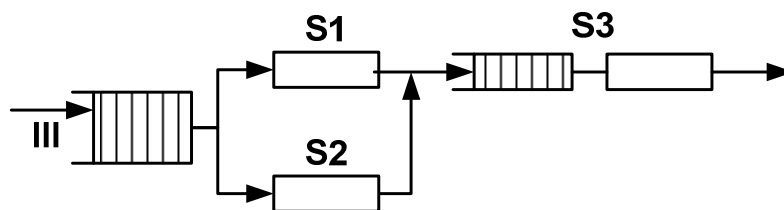
Индивидуальный вариант задан исходными данными в двух частях: первая – описание процессов функционирования СМО и схема, вторая – таблица значений времен поступления и обслуживания в устройствах (см. в конце, после описания схем всех вариантов).

Во всех вариантах время выполнения вспомогательных операций (передачи заявок и/или пакетов на обслуживание из очередей в устройства, между каналами и т. п.) принимать равным 1 с. Время поступления заявок входных потоков и обслуживания в устройствах является целочисленным, может быть либо постоянной величиной, либо величиной, равномерно распределенной в заданном интервале. Потери в схемах СМО условно не показаны.

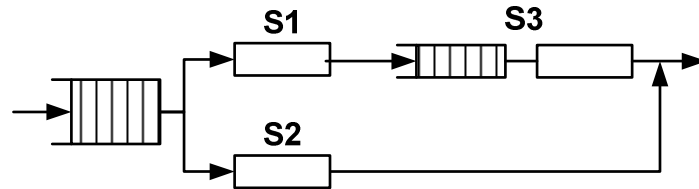
1. Время пребывания заявок входного потока с момента поступления в очередь S1 и до выхода из устройства S1 ограничено 4 с. Заявки, обслуженные без нарушения ограничения, далее обслуживаются в S2; с нарушением ограничения – в S3. С вероятностью 0,05 обслуживание в S2 оказывается ошибочным. Заявка с ошибочным обслуживанием теряется. Остановка моделирования: поступление 100-ой заявки во входную очередь.



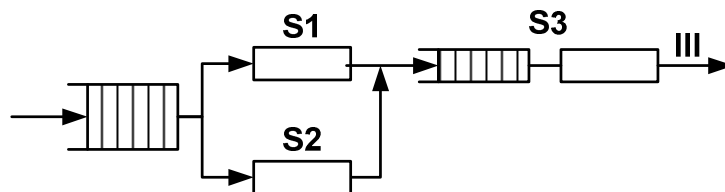
2. Заявки входного потока поступают в систему с вероятностью 0,68 пакетом по 2; 0,32 – по 3. Входная очередь ограничена 3-мя местами; в случае заполнения очереди пакеты теряются. Пакеты по 2 при обслуживании в S1-S2 пользуются относительным приоритетом. Для обслуживания они разукomплектовываются: одна заявка отправляется в S1, вторая – в S2. Пакеты по 3 направляются на обслуживание без разукomплектования, равновероятно в S1 и S2. В S3 заявки и пакеты обслуживаются в порядке поступления в очередь. Остановка моделирования: потеря 50-го пакета.



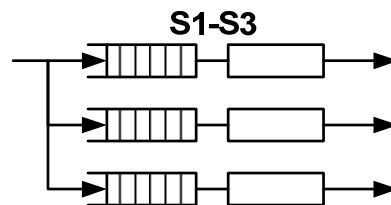
3. Входной поток разнородный, с вероятностью 0,6 поступают заявки типа А (обслуживаются в S1); с остаточной – типа В (обслуживаются в S2). Из общей входной очереди заявки на обслуживание извлекаются с чередованием типов. Время пребывания заявок типа А в своем канале ограничено 9 с; заявки, нарушившие ограничение, на выходе теряются. Остановка моделирования: поступление во входную очередь 100-ой заявки типа В.



4. В нормальном режиме работы заявки обслуживаются в S1-S3, причем в S3 – с пакетированием по 3. Через 50 ± 2 с наступает отказ S1. Обслуживание, прерванное отказом, теряется. Восстановление S1 происходит за 8 с. Во время восстановления заявки обслуживаются в резервном устройстве S2. Остановка моделирования: завершение 80-го обслуживания в S2.

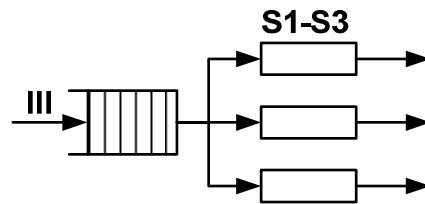


5. Входной поток разнородный, с вероятностями 0,3 поступают заявки типа А и типа В; с остаточной – типа С. Заявки каждого типа обслуживаются в «своем» устройстве. Через 30 ± 1 с наступает отказ S1. Обслуживание, прерванное отказом, теряется. Восстановление S1 происходит за 10 с. Во время восстановления заявки типа В равновероятно направляются на обслуживание в «не свои» устройства, где на обслуживание этого типа тратится двойное время. Остановка моделирования: через 400 с.

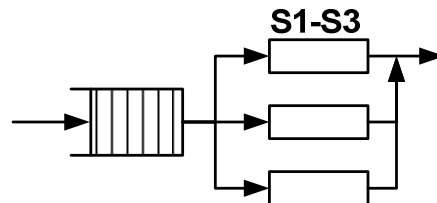


6. Входной поток разнородный, с вероятностями по 0,3 поступают одиночные заявки и пакеты по 2; с остаточной – пакеты по 3. Все пакеты разуккомплектовываются; заявки равновероятно обслуживаются в S1-S3. Через 25 ± 2 с равновероятно в любом из устройств возникает ошибка об-

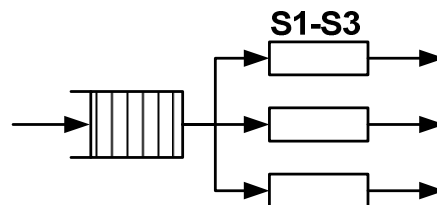
служивания. Если на момент возникновения ошибки устройство было занято обслуживанием – оно прерывается с потерей заявки. Остановка моделирования: поступление 100-го пакета по 3.



7. Входной поток разнородный, с вероятностью 0,6 поступают заявки типа А; с остаточной – типа В. В нормальном режиме работы заявки обслуживаются в S1. Тип В пользуется относительным приоритетом, для А время ожидания в очереди ограничено 2 с. Заявки, нарушившие ограничение, теряются. Через 40 ± 1 с наступает отказ S1. Обслуживание, прерванное отказом, теряется. Восстановление S1 происходит за 6 с. Во время восстановления заявки типа А обслуживаются в S2; типа В – в S3. Остановка моделирования: через 300 с.

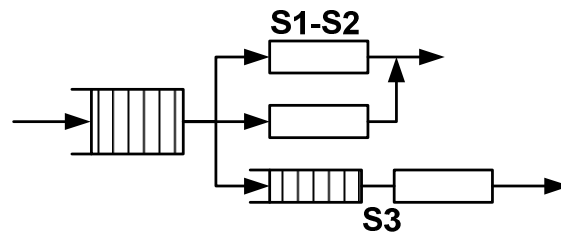


8. Входной поток разнородный, с вероятностями 0,25 поступают заявки типа А и типа В; с остаточной – типа С. Заявки принимаются очередью только если на момент поступления есть свободное устройство, иначе – теряются. Если два устройства одновременно заняты обслуживанием заявки типа С, то очередная поступившая заявка этого типа теряется даже если свободно 3-е устройство. Остановка моделирования: поступление 100-ой заявки типа А.

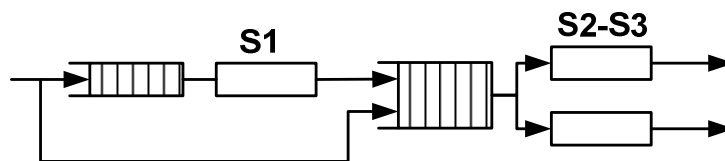


9. Заявки входного потока в нормальном режиме с вероятностью 0,7 направляются в устройство S1; с остаточной – в S2. Через 40 ± 1 с выполняется профилактика узла S1-S2, которая занимает 7 ± 1 с. Обслуживание в S1, прерванное профилактикой, теряется. Заявка из S2, обслуживание которой прерывается профилактикой, перенаправляется в очередь S3, где на

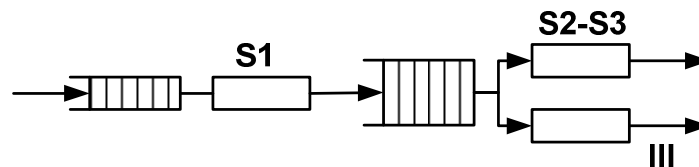
завершение обслуживания тратиться 1 с. Заявки входного потока, поступающие во время выполнения профилактики, обслуживаются в S3. Остановка моделирования: завершение обслуживания в S3 20-ой заявки.



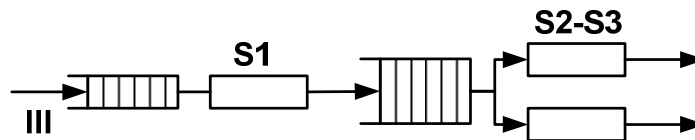
10. Входной поток разнородный, с вероятностью 0,3 поступают заявки типа А; с остаточной – типа В, которые сразу перенаправляются в общую очередь и на обслуживание в S2-S3 (равновероятно). Через 50 ± 2 с возникает отказ S1, восстановление занимает 9 ± 1 с. Обслуживание, прерванное отказом, теряется. Заявки типа А во время восстановления перенаправляются в общую очередь и на обслуживание в S2-S3, причем пользуются абсолютным приоритетом. Остановка моделирования: поступление в общую очередь обслуживания в S2-S3 20-ой заявки типа А.



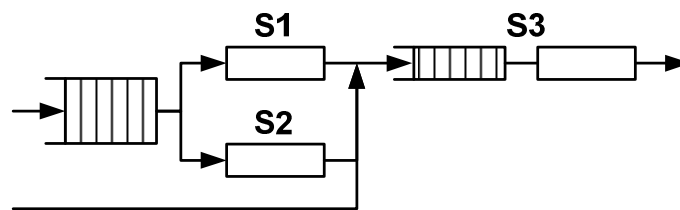
11. Входной поток разнородный, с вероятностью 0,65 поступают заявки типа А; с остаточной – типа В. Время ожидания обслуживания во входной очереди ограничено 5 с, при нарушении ограничения заявки теряются. Из общей очереди одиночные заявки направляются на обслуживание в S2, но, если есть возможность создать пакет из 2-х заявок разных типов, то он обслуживается в S3. Остановка моделирования: потеря 40-ой заявки типа А.



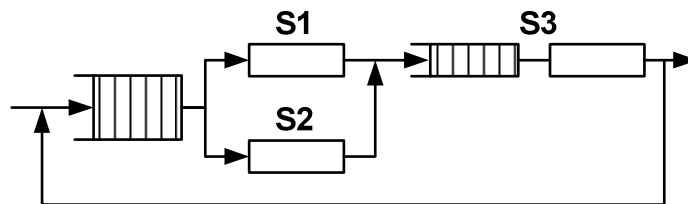
12. Заявки типов А и В поступают на обслуживание пакетами по 2, причем с равной вероятностью это пакеты комплектации вида АА, АВ, ВА, ВВ. Пакеты разделяются на составляющие, и, после обслуживания в S1, в S2-S3 обслуживаются по типам. Время ожидания обслуживания в очереди S2-S3 ограничено 3 с, при нарушении ограничения заявки теряются. Остановка моделирования: через 300 с.



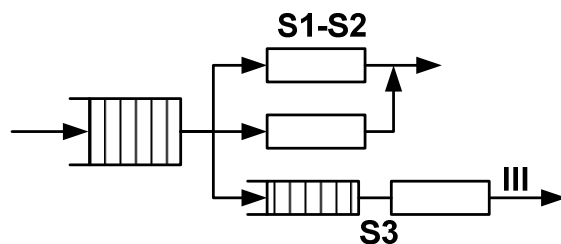
13. В СМО поступают два входных потока с одинаковой интенсивностью. Заявки первого потока с вероятностью 0,65 обслуживаются в S1, с остаточной – в S2. Заявки второго входного потока при обслуживании в S3 пользуются относительным приоритетом. Для заявок первого входного потока время ожидания в очереди S3 ограничено 3 с, при нарушении ограничения заявки теряются. Остановка моделирования: поступление 200-ой заявки второго входного потока на обслуживание в S3.



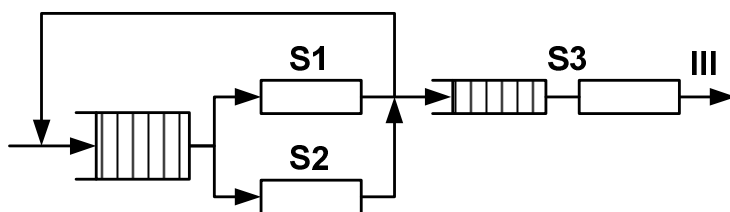
14. Заявки с вероятностью 0,55 обслуживаются в S1, с остаточной – в S2, и, далее – в S3. Через 30 ± 2 с возникает отказ S1, восстановление занимает 8 с. Заявка, обслуживание которой было прервано отказом, с вероятностью 0,35 теряется; с остаточной – продолжает обслуживание в S3, но с выхода S3 возвращается на вход СМО. Во время восстановления все заявки входного потока обслуживаются в S2. Остановка моделирования: через 400 с.



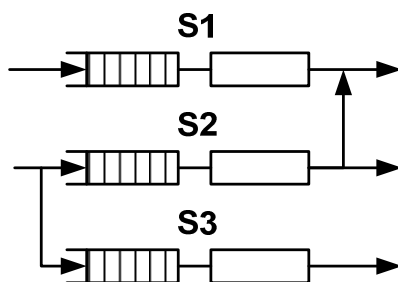
15. Входной поток разнородный, с вероятностью 0,65 поступают заявки типа А; с остаточной – типа В. Заявки входного потока вне зависимости от типа в нормальном режиме с вероятностью 0,7 направляются в устройство S1; с остаточной – в S2. Через 45 ± 1 с равновероятно возникает отказ S1 или S2, восстановление занимает 10 с. Обслуживание, прерванное отказом, теряется. Во время восстановления заявки входного потока направляются в очередь S3, при этом заявки, нарушающие чередование типов, теряются. В S3 заявки обслуживаются пакетами по 2. Остановка моделирования: поступление с входным потоком 200-ой заявки типа В.



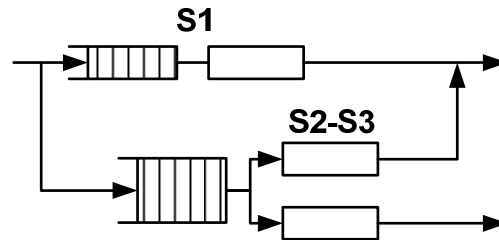
16. Входной поток заявок равномерно распределяется на обслуживание между S1 и S2. Для обслуживания в S3 заявки пакетируются по 3. Через 20 ± 2 с равномерно возникает ошибка обслуживания: равномерно в одном из устройств S1, S2 или в обоих сразу. Если на момент возникновения ошибки устройство занято обслуживанием и с момента последней ошибки прошло менее 20 с, то обслуживание прерывается и заявка теряется. Если с момента последней ошибки прошло больше либо равно 20 с, заявка возвращается во входную очередь для повторного обслуживания. Остановка моделирования: через 400 с.



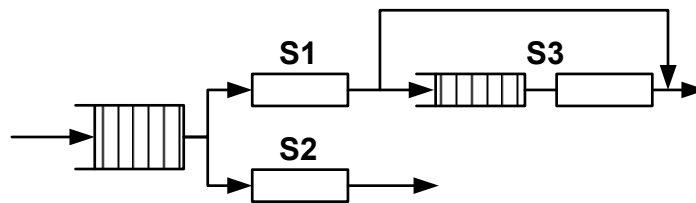
17. На вход СМО поступают два потока с одинаковым распределением времени между поступлениями заявок. С первым потоком поступают только заявки типа А; второй – разнородный: с вероятностями 0,45 поступают заявки типа А и В, с остаточной вероятностью – заявка типа С. Тип А второго потока обслуживается в S2; тип В – в S3; тип С распределяется между устройствами равномерно и пользуется абсолютным приоритетом (прерванное обслуживание теряется), причем обслуживании в любом устройстве заявки этого типа вызывает ошибку обслуживания следующей заявки в этом устройстве и ее потерю. Остановка моделирования: нормальное завершение обслуживания в системе 200-ой заявки типа А.



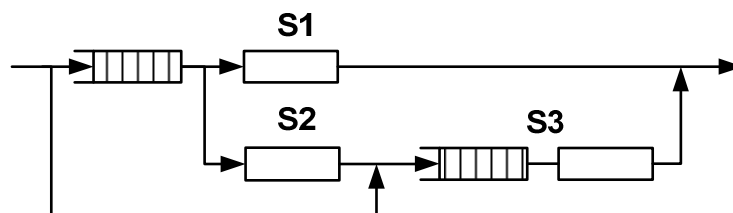
18. Входной поток разнородный: с вероятностями 0,45 поступают заявки типа А и В, с остаточной вероятностью – заявка типа С. Тип А обслуживается в S1, время ожидания для этого типа ограничено 3 с, при нарушении ограничения заявки теряются. Тип В и С обслуживаются в S2-S3, причем поступление заявки типа С переключает направления заявок: заявки обслуживаются в S2; после обслуживания очередной заявки типа С начинают направляться в S3; после обслуживания очередной заявки типа С – снова в S2 и так далее. Остановка моделирования: через 400 с.



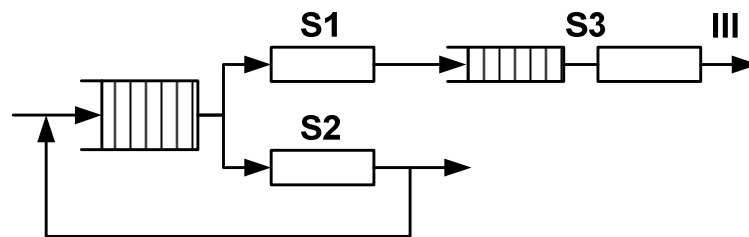
19. Входной поток разнородный, с вероятностью 0,65 поступают заявки типа А; с остаточной – типа В. Тип В обслуживается в S2; через 45 ± 1 с возникает отказ S2, восстановление занимает 5 с. Обслуживание, прерванное отказом, и все заявки типа В, поступающие во время восстановления, теряются. Заявки типа А, обслуживание которых в S1 заняло менее 3 с, теряются; от 3 до 5 с – направляются в S3; более 5 с – сразу покидают систему. Остановка моделирования: потеря 100-ой заявки.



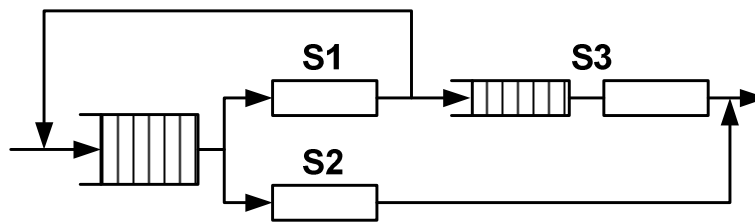
20. Входной поток разнородный: с вероятностями 0,4 поступают заявки типа А и В, с остаточной вероятностью – заявка типа С. Тип А обслуживается в S1; В – в S2 и S3; С – сразу в S3. Время ожидания обслуживания заявок типов А и В в общей очереди ограничено для заявок типа А – 2 с, для В – 3 с; при нарушении ограничения заявки теряются. Остановка моделирования: поступление в систему 80-ой заявки типа С.



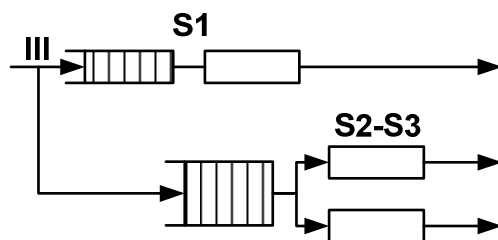
21. Входной поток разнородный, с вероятностью 0,6 поступают заявки типа А (обслуживаются в S1); с остаточной – типа В (обслуживаются в S2). Через 35 ± 1 с равновероятно возникает отказ S1 или S2, восстановление занимает 10 с. Обслуживание заявки типа А, прерванное отказом, теряется; заявка типа В после прерывания возвращается во входную очередь с меньшим относительным приоритетом. Во время восстановления заявки принимаются к обслуживанию «не своим» устройством, при этом на их обслуживание затрачивается двойное время, и они также имеют меньший относительный приоритет. В S3 заявки обслуживаются пакетами по 2. Остановка моделирования: через 600 с.



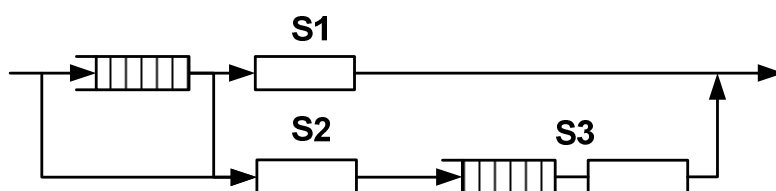
22. Заявки входного потока заявок направляются на обслуживание в S1 и S2 поочередно. Время ожидания во входной очереди ограничено 3 с. Если заявка, нарушившая ограничение, должна по очереди обслуживаться в S2, она теряется; если в S1 – принимается к обслуживанию, но после завершения возвращается во входную очередь. Остановка моделирования: поступление в очередь S3 350-ой заявки.



23. Входной поток разнородный, с вероятностью 0,6 поступают одиночные заявки (обслуживаются в S1); с 0,3 – пакеты по 2, остаточной – пакеты по 3. Пакеты по 2 разукomплектовываются и заявки поступают на обслуживание S2-S3 (равновероятно). Пакеты по 3 также разукomплектовываются, но одна из заявок сразу теряется. Через 25 ± 1 с равновероятно возникает отказ S2 или S3, восстановление занимает 6 с. Обслуживание заявки типа, прерванное отказом, теряется. Во время восстановления заявки принимаются к обслуживанию исправным устройством. Остановка моделирования: возникновение в системе 13-го отказа.



24. Заявки входного потока заявок направляются на обслуживание в S1 и S2 равновероятно. Если на момент поступления с входным потоком очередной заявки очередь равна 3-м заявкам, то вновь поступившей присваивается абсолютный приоритет и она направляется в S2 минуя очередь. Обслуживание, прерванное приоритетной заявкой, теряется. Устройство S3 после обслуживания каждой 10-ой заявки прекращает обслуживание на 1 с. Остановка моделирования: поступление в очередь S3 350-ой заявки.



№	Поступление	Обслуживание		
		S1	S2	S3
1	2	3±1	4±1	5
2	4±2	2	2	1
3	2±1	2	2	2±1
4	3±1	2	2	7±1
5	5±2	2	2	2
6	2±1	5±2	2	2
7	1	4±2	2	1
8	2±1	9±1	9±2	2
9	3±2	2	2	2
10	4±2	3±1	3±1	1
11	3±2	2	2	2
12	1	5±1	5±1	2

№	Поступление	Обслуживание		
		S1	S2	S3
13	2±1	2	1	2±1
14	2±1	7±2	8±2	3
15	2±1	3±1	3	3±1
16	3±2	2±1	2	3±1
17	3±2	2	2	2
18	6±1	2±1	2±1	2±1
19	2±1	2±1	1	1
20	3±1	9	8±1	7
21	3±1	1	2	2±1
22	2±1	2±1	1	1
23	1	4±2	8±2	2
24	3	2±1	4±2	4±2

Лабораторная работа № 7

Моделирование систем массового обслуживания с визуализацией характеристик системной динамики

Цель: моделирование системы массового обслуживания неоднородного потока заявок с визуализацией и анализом операционных характеристик обслуживания.

Программное обеспечение: система имитационного моделирования Arena.

Примеры выполнения работы: файлы *Пример 1 (визуализация динамики функционирования).doe*, *Пример 2 (общая очередь).doe*, *Пример 3 (ограниченная очередь).doe*.

Теория

Основные сведения о системах массового обслуживания – см. раздел *Теория* в *Лабораторной работе № 6*.

В зависимости от специфики функционирования системы с вероятностной природой при ее моделировании системой массового обслуживания для описания закона распределения длительности интервалов времени между поступлением в СМО динамических объектов (заявок, запросов, транзактов и т.п.) и законов распределения длительности интервалов обслуживания в обслуживающих аппаратах (устройствах) используют следующие типы распределений:

- равномерное вида $a \pm b$;
- экспоненциальное с интенсивностью λ , математическим ожиданием $1/\lambda$ и плотностью вероятности $f(x) = \lambda \cdot e^{-\lambda x}$;
- гамма-распределение (распределение Эрланга с целочисленным параметром) с интенсивностью λ и целочисленным параметром b , математическим ожиданием b/λ и плотностью вероятности $f(x) = \lambda \frac{(\lambda x)^{b-1}}{(b-1)!} e^{-\lambda x}$;
- нормальное распределение с математическим ожиданием m , коэффициентом масштаба σ (σ^2 равно дисперсии случайной величины) и плотностью вероятности $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right)$.

В программном обеспечении моделирования СМО реализация перечисленных законов распределения случайных величин осуществляется с использованием программных генераторов псевдослучайных чисел.

Основной операционной характеристикой устройства обслуживания S является *коэффициент использования устройства*, равный доле суммарного времени работы устройства по обслуживанию заявок от общего времени функционирования СМО T :

$$КИ_S = \frac{\sum_{i=1}^n t_{обсл_i}}{T} = \frac{\sum_{i=1}^n t_{обсл_i}}{\sum_{i=1}^n t_{обсл_i} + \sum_{i=1}^n t_{простоя_i}}.$$

Построение функциональной модели системы осуществляется от общего к частному – сначала строится концептуальная модель, затем шаг за шагом все более и более подробно описываются конкретные процессы функционирования системы. Такой подход весьма эффективен, однако на уровне наибольшей детализации, когда рассматриваются отдельные операции обслуживания, для оптимизации этих операционных характеристик функциональной модели может оказаться недостаточно. В этом случае целесообразно использовать имитационное моделирование.

Имитационное моделирование – это метод, позволяющий строить модели, учитывающие время выполнения процессов и их операций. Полученную модель можно многократно «проиграть» во времени и получить статистику происходящих процессов так, как это было бы в реальности. В имитационной модели изменения процессов и данных ассоциируются с событиями. «Проигрывание» модели заключается в последовательном переходе от одного события к другому. Обычно имитационные модели строятся для поиска оптимального решения в условиях ограничений, когда другие математические модели оказываются слишком сложными.

Ход выполнения работы

1. Реализовать имитационную модель заданного варианта СМО в системе имитационного моделирования Arena с визуализацией динамики функционирования: назначением различных графических обозначений для заявок разного типа; индикацией в реальном времени текущих уровней значений коэффициентов использования устройств и длин очередей.

2. Осуществить имитационное моделирование СМО. По результатам моделирования оценить следующие характеристики СМО:

- коэффициент использования и число обслуженных заявок для устройств обслуживания;

- среднюю длину и среднее время ожидания для отдельных очередей устройств;

- процент потерь заявок (если есть).

3. Проанализировать результаты моделирования. Оценить эффективность СМО с точки зрения загрузки устройств, задержек заявок в очередях, потерь заявок из-за переполнения очередей (если есть).

4. Предложить и реализовать вариант структурной оптимизации СМО, гарантирующей значение коэффициента использования наиболее загруженного устройства в системе равное 0,5. По результатам моделирования оценить характеристики СМО.

5. Предложить и реализовать вариант параметрической модификации СМО, гарантирующей значение коэффициента использования наиболее загруженного устройства в системе равное 0,5. По результатам моделирования оценить характеристики СМО.

6. Результаты выполнения работы по пп. 1 – 5 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

Практические рекомендации по выполнению работы и использованию программного обеспечения

Arena является эффективным программным средством имитационного моделирования. В совокупности с поставляемыми с ней инструментальными средствами (Arena Viewer, Input Analyzer, Output Analyzer, OptQuest), она представляет собой среду работы системного аналитика и позволяет моделировать производственные процессы, системы массового обслуживания, социально-экономические процессы и т.д.

При моделировании процессов в Arena, используются три основных области окна программы. Левая колонка рабочей области – Project Bar – содержит панели с основными типами объектов, необходимых для моделирования:

- Basic Process: содержит модули, используемые для имитации процессов и событий в системе;

- Reports: содержит сообщения и отчеты по результатам прогонов модели;

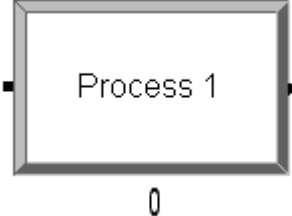
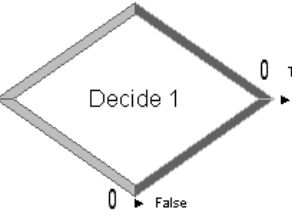
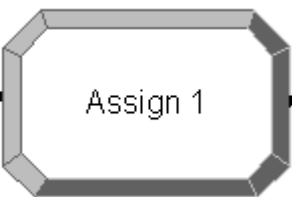
- Navigate: позволяет показывать различные представления вашей модели, включая иерархию подсистем (подмоделей).

Добавлять/удалять панели можно из меню File → Template Panel → Attach/Detach или с использованием кнопки  графической панели инструментов. Для выполнения работы дополнительно потребуется панель Advanced process.

Модули в Arena это элементы потоковой диаграммы и объекты данных, определяющие процесс моделирования. В модулях содержится вся информация относительно предметной области, ее компонент и параметров. В панели Basic Process содержатся основные модули, которые размещаются в рабочее пространство модели методом drag&drop.

Для моделирования СМО по заданиям данной работы потребуются модули, приведенные в табл. 20.

Таблица 20

Вид модуля	Назначение и описание
	Генерация потока заявок. Появление очередной заявки определяется расписанием или интервалом времени, генерируемым по заданному в параметрах модуля закону распределения. Численный параметр блока – количество сгенерированных за время моделирования заявок
	Выход (удаление) заявок из системы. Предназначен для сбора статистики по количеству заявок, покидающих модель. Численный параметр блока – количество заявок, покинувших систему через данный выход за время моделирования
	Обработка заявки определенным устройством обслуживания, выполняемая в течение времени, генерируемого по заданному в параметрах модуля закону распределения. Численный параметр блока – количество обслуживаемых в настоящий момент заявок
	Имитация процесса принятия решения по выбору одного из двух или более выходов для заявки, основанному на одном или нескольких условиях или вероятностях. Численные параметры блока – количество заявок, выбравших каждый из возможных выходов за время моделирования
	Модуль присваивания в процессе моделирования новых значений переменным, параметрам объекта, типу объекта и другим характеристикам

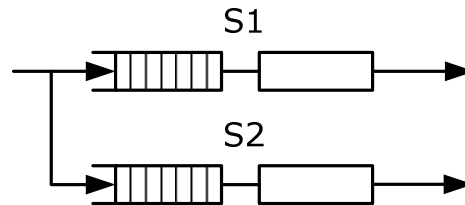
Помимо рассмотренных модулей типа «Flowchart» (элементов потоковой диаграммы), в панели Basic Process содержатся модули типа «Data»

– модули данных, которые позволяют задать различные характеристики элементам моделирования – объектам, ресурсам, очередям и др. Данные представляются в виде таблиц в нижней части окна модели.

Пример 1

Моделирование узла СМО с визуализацией динамики функционирования

Пусть узел СМО задан структурной схемой вида:



На вход системы поступает поток заявок двух типов (А и В) с вероятностями поступления 0,6 и 0,4 соответственно. Длительность интервалов времени между поступлениями заявок распределена по экспоненциальному закону с интенсивностью $\lambda = 1$. Заявки типа А обслуживаются в устройстве S1, типа В – в S2. Время обслуживания в обоих устройствах составляет 2 ± 1 с.

Реализуем модель заданного узла в Arena. Для моделирования входного потока заявок используется модуль Create панели Basic Process. В окне редактирования свойств модуля задается имя; тип распределения и параметры закона распределения длительности интервалов времени между поступлениями заявок (вводятся в скобках после обозначения закона распределения вместо обозначений параметров – так, для экспоненциального закона параметром является величина математического ожидания Mean); единица измерения времени (рис. 149).

Для реализации распределения заявок входного потока по типам с заданными значениями вероятностей поступления используется модуль Decide панели Basic Process. В настройках модуля задается тип реализуемого перехода: по условию или по вероятности. В данном варианте использования необходим вероятностный переход на 2 выхода. В этом случае задается значение вероятности перехода в процентах для выхода типа True, а значение вероятности перехода по выходу False вычисляется автоматически. В результате, после прохождения этого модуля общий поток заявок неопределенного типа разделяется на два независимых потока: поток заявок типа А покидает модуль через выход True, а типа В – через выход False (рис. 150).

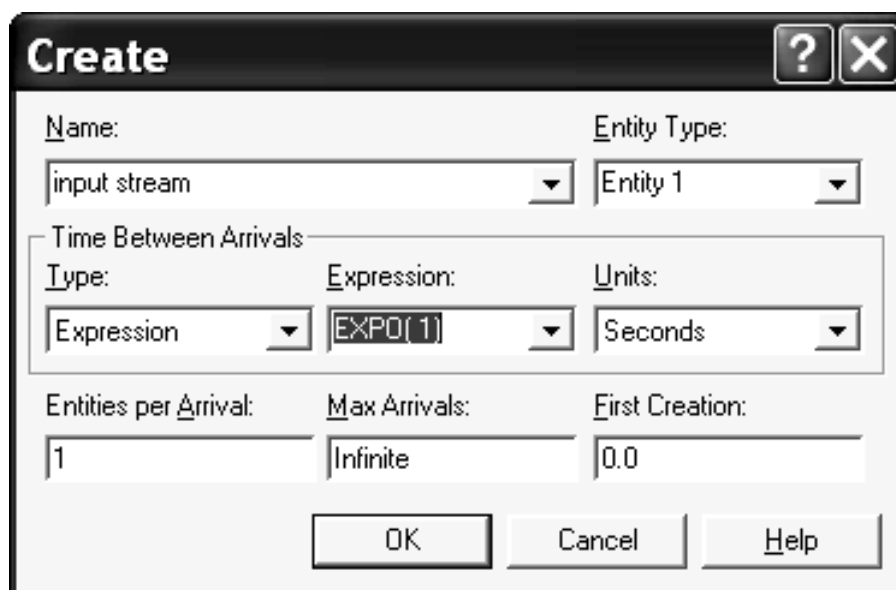


Рис. 149

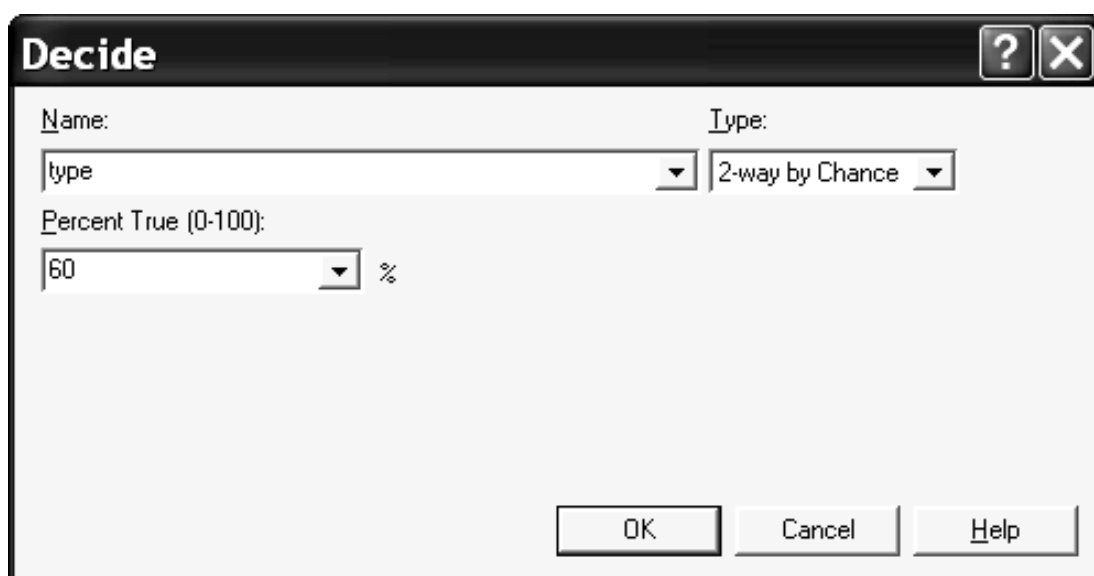


Рис. 150

Для визуализации динамики функционирования требуется использовать различные графические обозначения для заявок разного типа. Для изменения параметров заявок в Arena используется модуль Assign панели Basic Process. В окне редактирования свойств модуля следует выбрать режим добавления позиции в список параметров заявки (опция Add...), в режиме добавления задать тип изменяемого параметра – Entity Picture и выбрать вид графического обозначения для заявки конкретного типа (рис. 151).

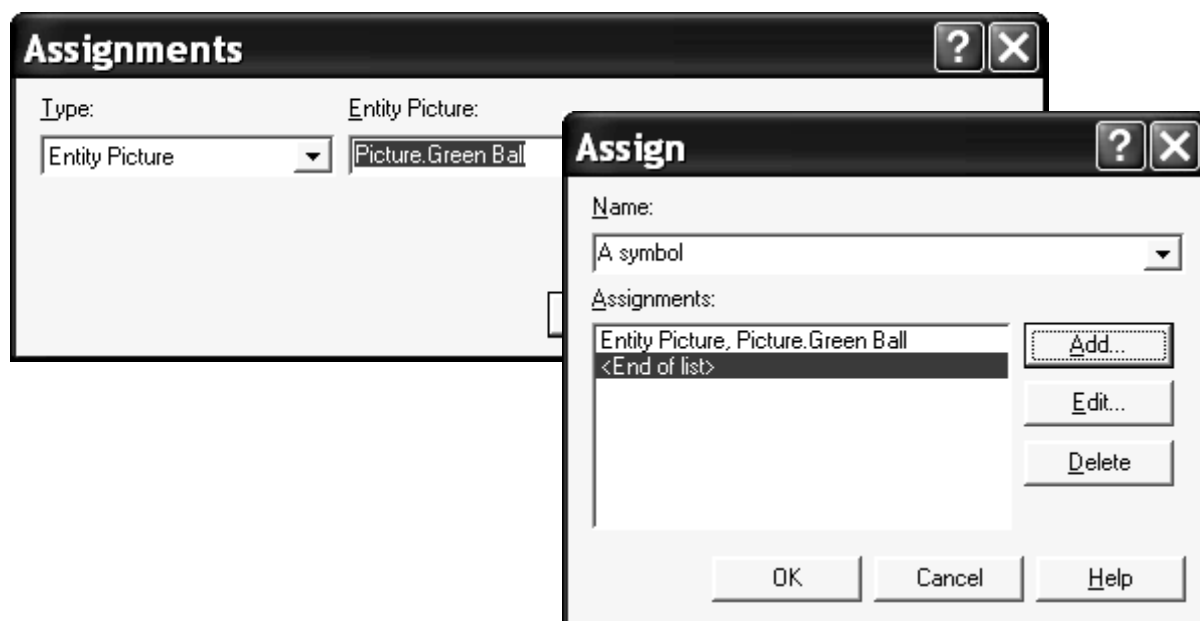


Рис. 151

Для моделирования обслуживания заявок по типам в устройствах S1 и S2 используется модуль Process панели Basic Process. В окне редактирования свойств модуля задается тип выполняемого действия: в случае реализации устройством СМО простейшего обслуживания в порядке поступления заявок в очередь выбирается тип Seize Delay Release (занять устройство обслуживания, задержать заявку на сгенерированное время обслуживания, освободить устройство по завершению обслуживания). Кроме того, в свойствах модуля задается тип распределения и параметры закона распределения длительности обслуживания; единица измерения времени; список используемых при обслуживании ресурсов и их количество (рис. 152).

Для организации выходов обслуженных заявок из СМО используется модуль Dispose.

Реализованная в редакторе Arena схема модели заданного узла СМО имеет вид, показанный на рис. 153, а (см. также файл *Пример 1 (визуализация динамики функционирования).doe*).

Для получения и анализа статистики функционирования моделируемого узла СМО необходимо реализовать достаточно большое число прогоном имитационной модели. Задать необходимое число прогонов можно несколькими способами: указать число генерируемых источником входного потока заявок (параметр Max Arrivals модуля Create), задать время моделирования; задать остановки моделирования по условию и т.п. В рассматриваемом примере определим число прогонов модели заданием времени моделирования равным 1000 с в пункте меню Run Setup (рис. 154).

The image shows two overlapping dialog boxes from a software application. The 'Resources' dialog is in the background, and the 'Process' dialog is in the foreground.

Resources Dialog:

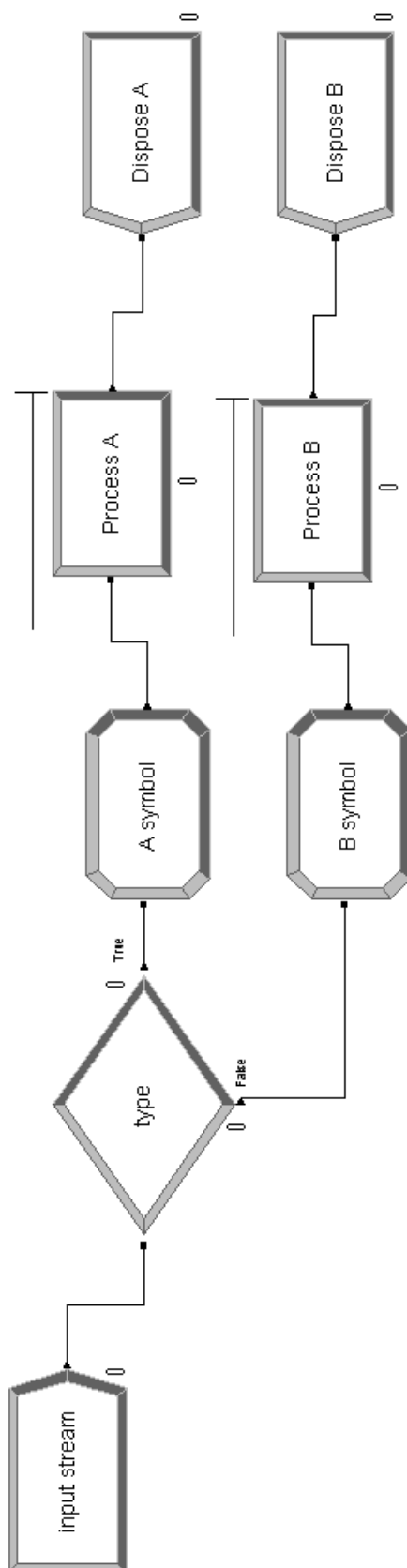
- Type:** Resource
- Resource Name:** S1
- Quantity:** 1

Process Dialog:

- Name:** Process A
- Type:** Standard
- Logic:**
 - Action:** Seize Delay Release
 - Priority:** Medium(2)
 - Resources:**
 - Resource, S1, 1
 - <End of list>
 - Buttons:** Add..., Edit..., Delete
- Delay Type:** Uniform
- Units:** Seconds
- Allocation:** Value Added
- Minimum:** 1
- Maximum:** 3
- ☒ Report Statistics
- Buttons:** OK, Cancel, Help

Рис. 152

a



б

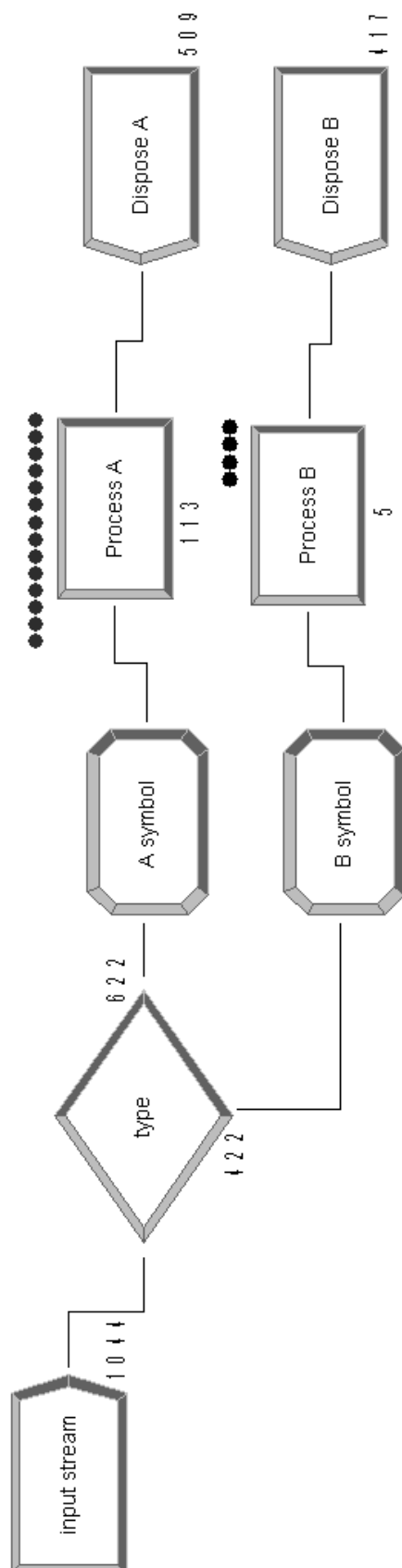


Рис. 153

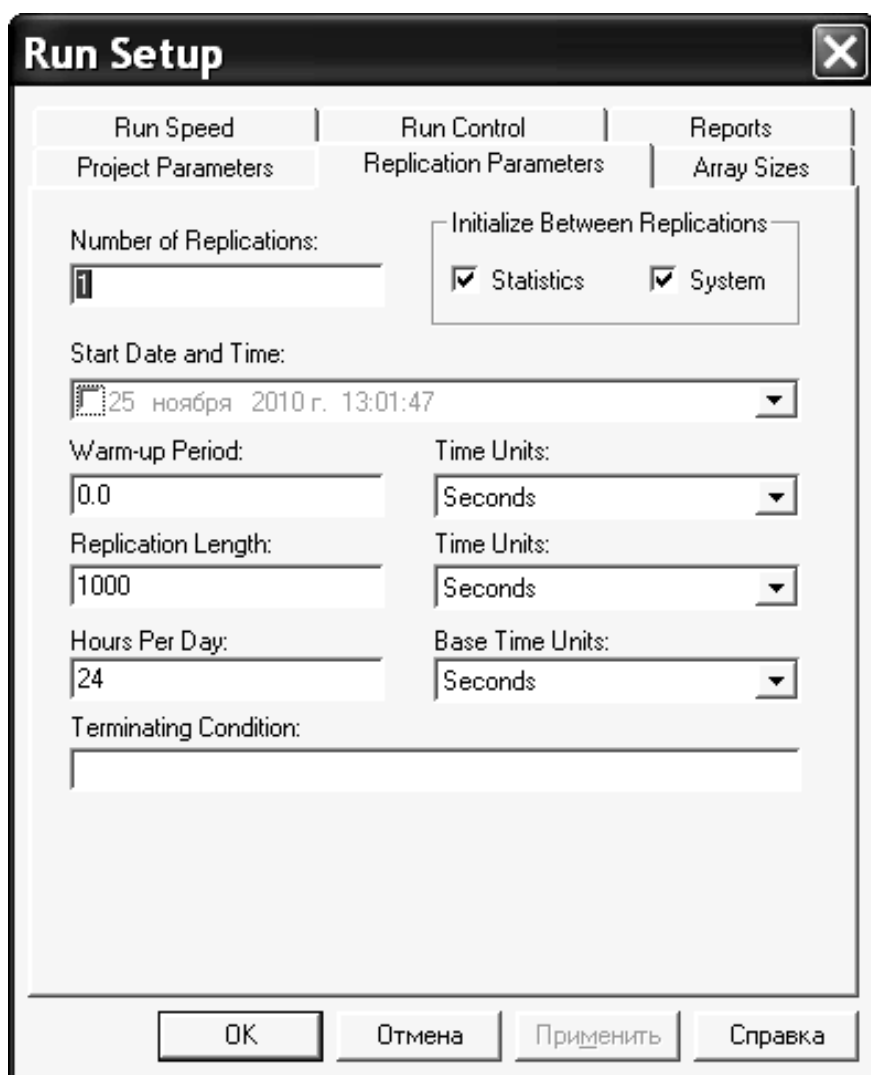


Рис. 154

Запуск моделирования осуществляется кнопкой Go  графической панели инструментов. Скорость моделирования настраивается шкалой скорости с перемещаемым маркером справа от панели запуска: используйте ее для уменьшения скорости моделирования с целью проверки визуализации процессов обслуживания. Для пошагового прогона модели используется кнопка Step .

Результат имитационного моделирования заданного узла СМО имеет вид (см. рис. 153, б). Как видно, за заданное время моделирования с входным потоком в узел СМО поступило всего 1044 заявок, из которых 622 оказались заявками типа А и 422 – типа В. На момент завершения моделирования 509 заявок типа А покинули систему обслуженными в устройстве S1, 112 оказались задержанными в очереди устройства S1 и одна заявка

находилась в процессе обслуживания. Аналогично, 417 заявок типа В покинули систему обслуженными в S2, 4 оказались задержанными в очереди устройства S2 и одна заявка находилась в процессе обслуживания. Операционные статистические характеристики обслуживания заявок в моделируемом узле СМО рассчитываются автоматически и приводятся в генерируемых по результатам моделирования отчетах в разделе Reports. Результаты анализа отчетов сведены в табл. 21.

Таблица 21

Параметр	Значения параметра	Раздел стандартного отчета
Среднее время ожидания заявок в очередях, с		
Устройство S1	81,65	Queues
Устройство S2	4,24	
Среднее число заявок в очередях		
Устройство S1	52,3	Queues
Устройство S2	1,8	
Коэффициенты использования устройств		
Устройство S1	0,99	Resources
Устройство S2	0,81	

Дополним реализованную модель индикацией в реальном времени текущих уровней значений коэффициентов использования устройств и длин очередей. Индикатор текущих значений заданных параметров выбирается с использованием кнопки  графической панели инструментов. Зададим индикацию в процессе моделирования текущего значения длины очереди устройства S1. Для этого в поле Expression окна редактирования свойств индикатора в контекстном меню (вызывается правой кнопкой мыши) выберем пункт Build Expression, в списке параметров выберем текущую длину очереди (Current Number In Queue), а в списке очередей устройств системы – имя очереди устройства S1 (рис. 155).

Добавим в настройках индикации заголовок индикатора и автомасштабирование значений в процессе моделирования (рис. 156).

Аналогично задается индикация коэффициентов использования устройств: в списке параметров Basic Processes Variables следует выбрать Resource/Usage/Scheduled Utilization. Промежуточный результат моделирования с индикацией текущих значений операционных характеристик обслуживания для заданного примера имеет вид, показанный на рис. 157. Индикацию текущих значений операционных характеристик обслуживания следует использовать для анализа корректности решения задачи оптимизации/модификации системы в соответствии с заданием индивидуального варианта.

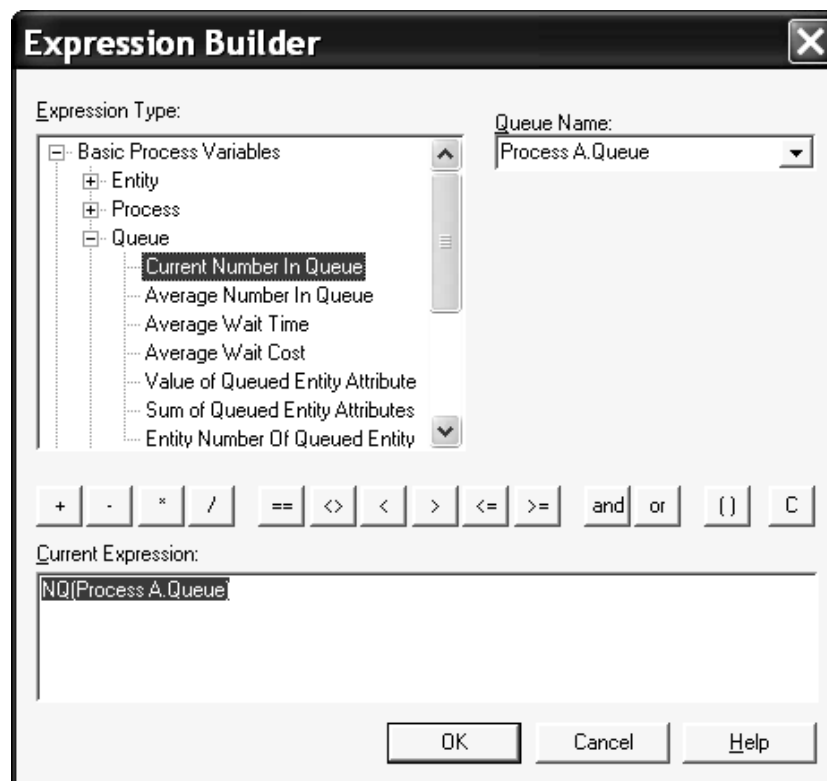


Рис. 155

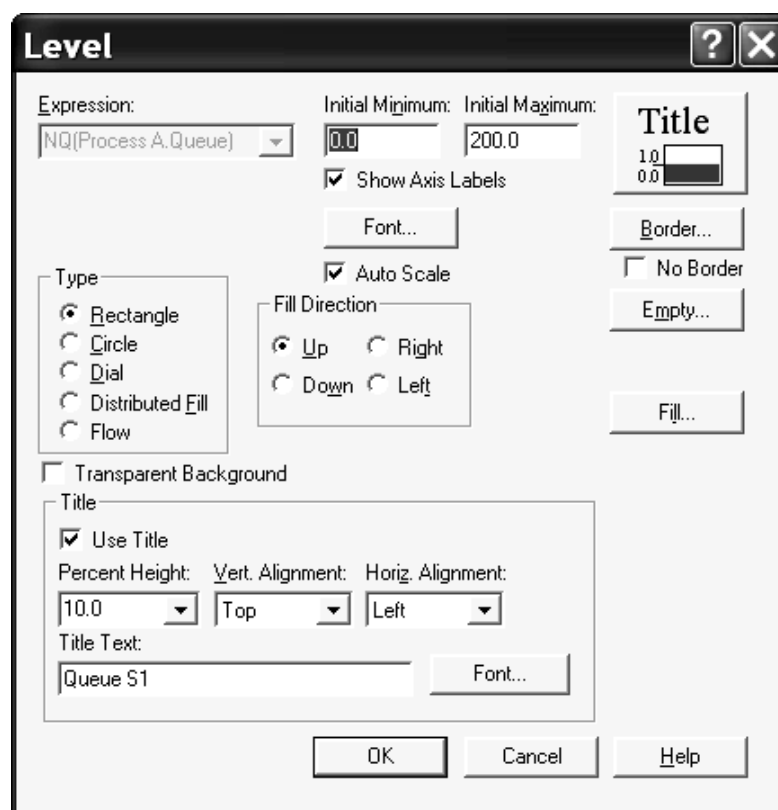


Рис. 156

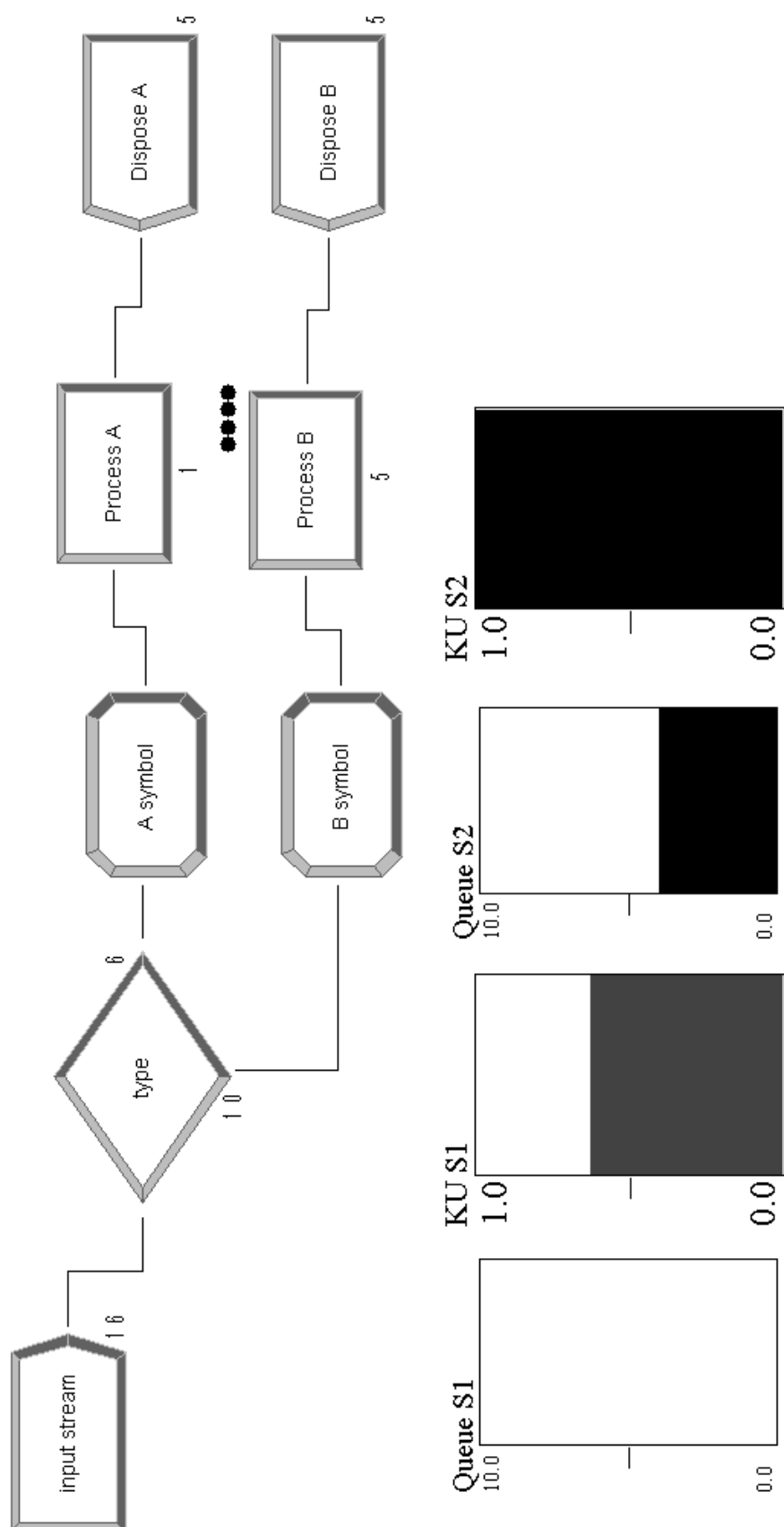
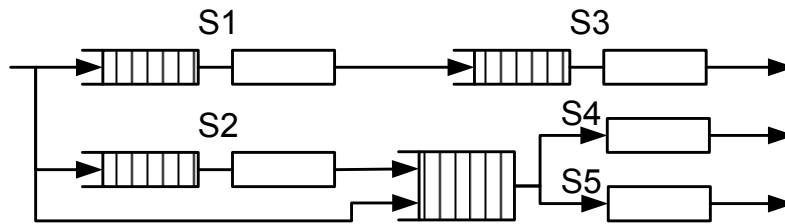


Рис. 157

Пример 2

Моделирование узла СМО с общей очередью

Пусть СМО задана структурной схемой вида:



На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,3; 0,3 и 0,4 соответственно. Заявки первого типа обслуживаются в S1, второго – равновероятно в S1 и S2, третьего – в S2. Вероятность обслуживания заявок всех типов в S4 равна 0,7; в S5 – 0,2, с остаточной вероятностью заявки теряются. Законы и параметры распределений времен поступления/обслуживания заявок приведены в табл. 22.

Таблица 22

объект \ характеристика	входной поток	S1	S2	S3	S4	S5
закон распределения	гамма	равн.	эсп.	эсп.	эсп.	эсп.
параметры	$\lambda=1$; $b=4$	$1 \pm 0,5$	$\lambda=0,2$	$\lambda=0,2$	$\lambda=0,2$	$\lambda=0,2$

Как следует из структурной схемы СМО устройства S4 и S5 имеют общую очередь обслуживания. Для ее реализации в Arena, после построения основной модели, в панели Basic Process был выбран модуль данных Queue (см. файл *Пример 2 (общая очередь).doe*). При выборе этого модуля в нижней части экрана под схемой в таблицу очередей была добавлена новая очередь Queue4_5 и задан режим общего доступа (Shared) (рис. 158).

Queue - Basic Process				
	Name	Type	Shared	Report Statistics
1	S1.Queue	First In First Out	☐	☒
2	S2.Queue	First In First Out	☐	☒
3	S3.Queue	First In First Out	☐	☒
4	Queue4_5	First In First Out	☒	☒

Double-click here to add a new row.

Рис. 158

После этого, для организации режима поступления заявок в общую очередь в панели Advanced Process были выбраны модули Seize (задают

режим «занять ресурс», в данном случае – очередь), параметры которых заданы следующим образом (рис. 159).

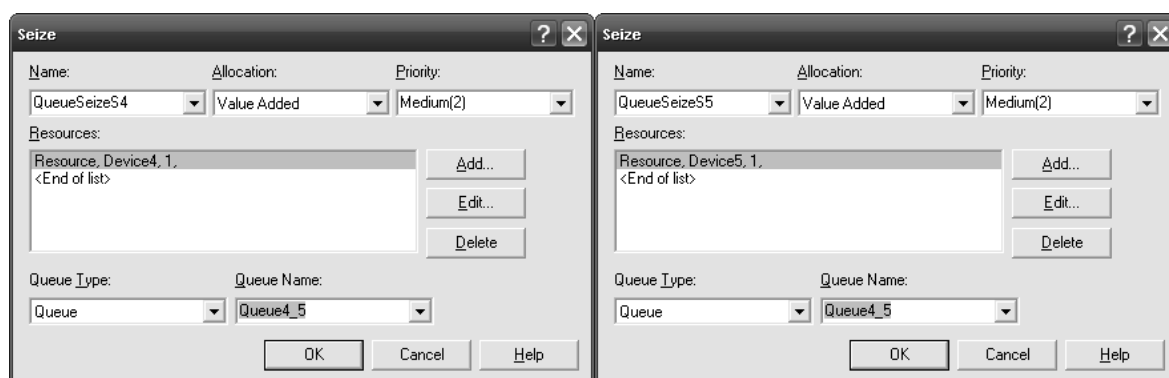


Рис. 159

Обратите внимание на то, что после реализации отдельным блоком Seize «захвата» одного из устройств с общей очередью для моделирования работы собственно устройства следует использовать модуль Process типа Delay Release.

Промежуточный результат моделирования с визуализацией содержимого очередей с использованием различных графических обозначений для заявок разных типов, подтверждающий корректность реализации режима использования общей очереди, для данного примера имеет вид (рис. 160).

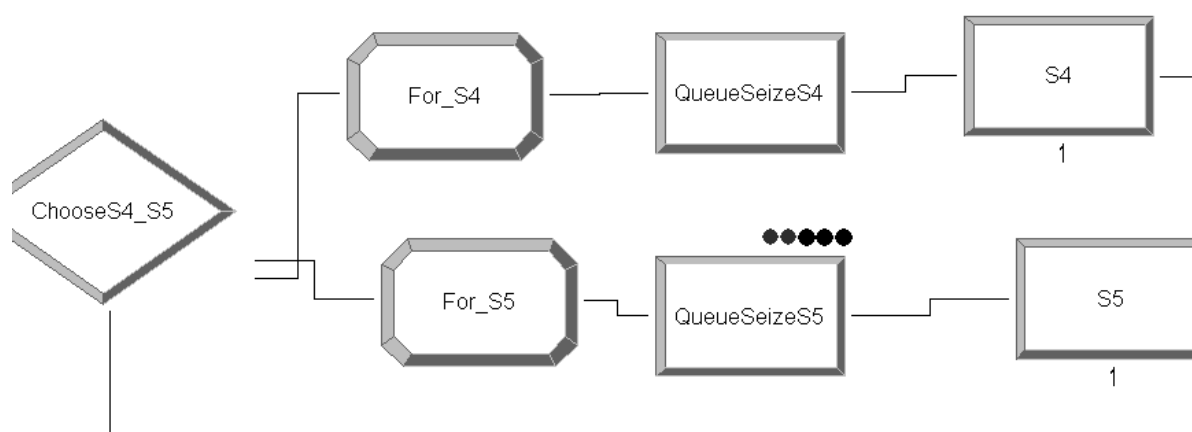


Рис. 160

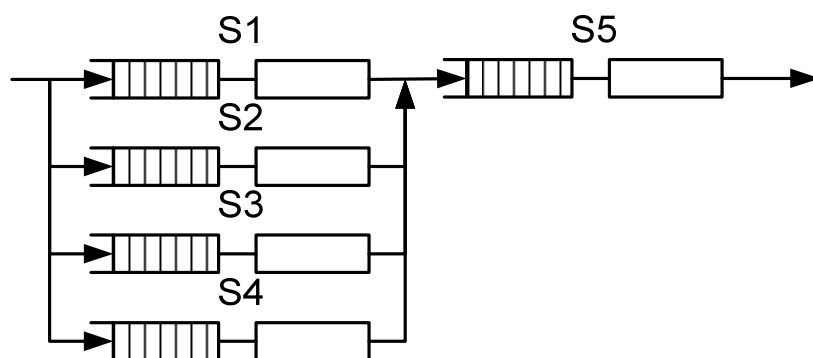
В общей очереди устройства находятся заявки, предназначенные с заданной вероятностью для обслуживания в устройствах S4 и S5: в этом можно убедиться в режиме пошагового прогона модели *Примера 2* (в за-

медленном прогоне легко наблюдать корректность извлечения заявок из общей очереди и передачи из в устройства S4 и S5). В примере для бóльшей наглядности были дополнительно использованы графические обозначения разного цвета для заявок, распределяемых между устройствами S4 и S5 (реализовано с использованием вспомогательных модулей типа Assign с именами For_S4 и For_S5).

Пример 3

Моделирование СМО с ограниченной очередью

Пусть СМО задана структурной схемой вида:



На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,3; 0,45 и 0,25 соответственно. Заявки первого типа обслуживаются в S1 и с вероятностью 0,5 – в S2; второго типа – в S2 и с вероятностью 0,5 – в S3; третьего типа – в S3 и с вероятностью 0,5 – в S4. Очередь S5 ограничена 3-мя заявками, при переполнении очереди заявки теряются. Законы и параметры распределений времен поступления/обслуживания заявок приведены в табл. 23.

Таблица 23

характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	гамма	эксп.	эксп.	эксп.	эксп.	равн.
параметры	$\lambda=1$; $b=3$	$\lambda=0,4$	$\lambda=0,5$	$\lambda=0,5$	$\lambda=0,1$	10 ± 2

Как следует из описания варианта СМО устройство S5 имеет ограниченную очередь обслуживания. Для ее реализации в Arena, после построения основной модели (см. файл *Пример 3 (ограниченная очередь).doe*), в панели Basic Process был выбран модуль Decide. В параметрах модуля задается переход заявок по условию (опция 2-way by Condition). Выражение для проверки условия задается в поле Value вызовом меню ре-

дактирования Build Expression с заданием в списке параметров текущей длины очереди (Current Number In Queue) устройства S5 и заданием ограничения длины (рис. 161).

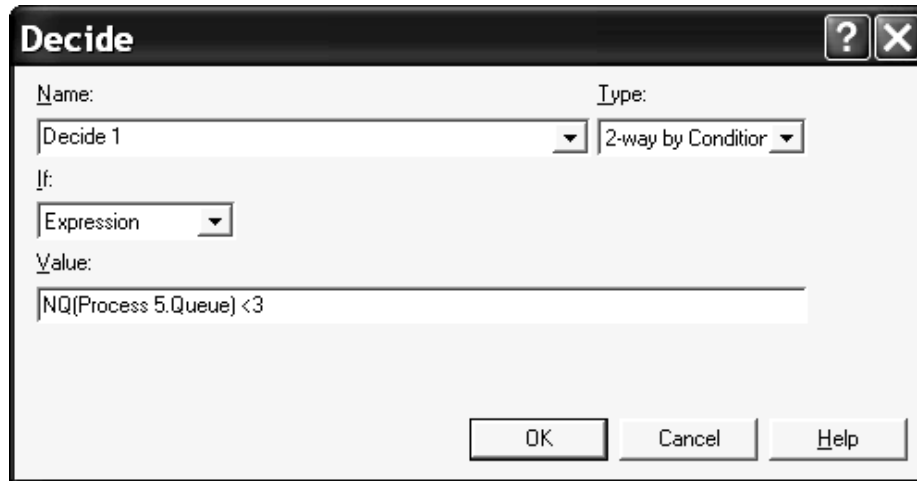
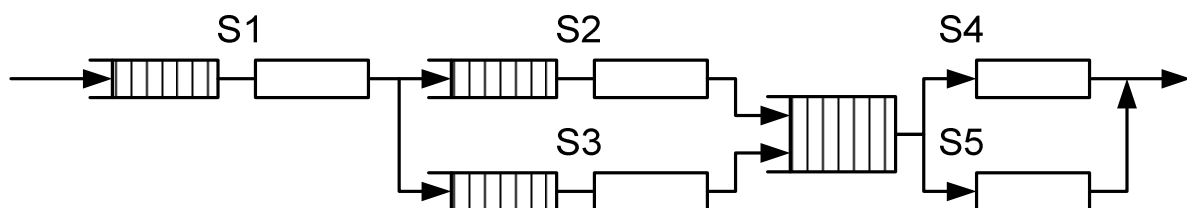


Рис. 161

В режиме пошагового прогона или автоматического моделирования с малой скоростью можно убедиться в корректности реализации ограничения очереди: заявки, претендующие на обслуживание в S5, при наличии одной обслуживаемой заявки в устройстве и 3-х в очереди, теряются, покидая систему через выход Dispose 2 (см. файл *Пример 3 (ограниченная очередь).doe*).

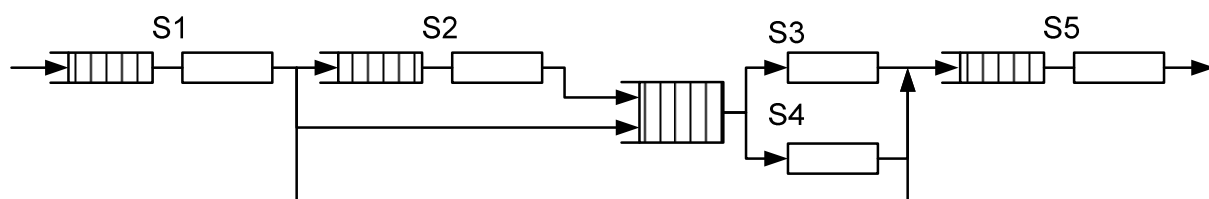
Варианты индивидуальных заданий

1. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,4 и 0,6 соответственно. После обслуживания в S1, заявки первого типа обслуживаются в S2, а второго – в S3. Общая очередь S4-S5 ограничена 3-мя заявками, при переполнении заявки теряются. Вероятности обслуживания в S4 – 0,3; в S5 – 0,7.



объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	эсп.	равн.
параметры	$\lambda=2$	$1\pm0,5$	$2\pm0,5$	$2\pm1,5$	$\lambda=0,13$	6 ± 1

2. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,3 и 0,7 соответственно. Заявки первого типа обслуживаются в S1 и S2; второго – после обслуживания в S1 с вероятностью 0,7 поступают в общую очередь устройств S3-S4, а с вероятностью 0,3 – сразу в очередь устройства S5.



объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	эсп.	эсп.	равн.	равн.	равн.
параметры	0.5 ± 0.2	$\lambda=0,2$	$\lambda=0,25$	3 ± 1	$2\pm0,5$	6 ± 2

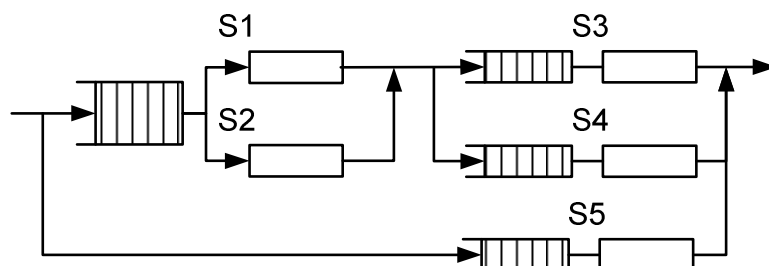
3. Схема – см. вариант 2. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,6 и 0,4 соответственно. Очередь S2 ограничена 3-мя заявками. При переполнении очереди заявки второго типа теряются, первого с вероятностью 0,2 в общую очередь устройств S3-S4, а с вероятностью 0,8 – сразу в очередь устройства S5.

объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	эсп.	гамма	равн.	равн.
параметры	$1\pm0,5$	$1,2\pm0,5$	$\lambda=0,2$	$\lambda=0,5; b=3$	$2\pm0,5$	$2\pm0,5$

4. Схема – см. вариант 1. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,55 и 0,45 соответственно. Очередь S2 ограничена 2-мя заявками; заявки покинувшие очередь из-за ограничения, поступают на обслуживание в S3. Общая очередь S4-S5 ограничена 3-мя заявками, при переполнении заявки теряются. Заявки первого типа обслуживаются в S4, второго – в S5.

объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	эсп.	равн.
параметры	$\lambda=1$	$1\pm0,2$	2 ± 1	12 ± 4	$\lambda=0,07$	8 ± 1

5. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,35 и 0,65 соответственно. Общая очередь S1-S2, где заявки обслуживаются согласно типу, ограничена 3-мя заявками; заявки всех типов покинувшие очередь из-за ограничения, поступают на обслуживание в S5. Обслуживание в S3-S4 заявок обоих типов равновероятно.

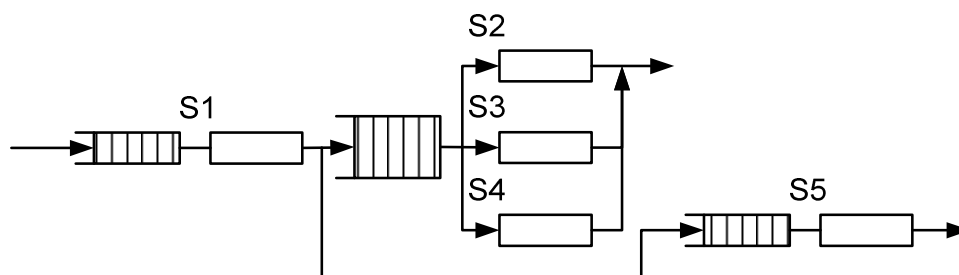


объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эксп.	равн.	равн.	равн.	равн.	эксп.
параметры	$\lambda=1$	8 ± 1	4 ± 1	4 ± 1	$2\pm 0,5$	$\lambda=0,2$

6. Схема – см. вариант 5. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,7 и 0,3 соответственно. Заявки второго типа обслуживаются в S5, очередь которого ограничена 2-мя заявками; при переполнении очереди заявки теряются. Заявки первого типа в S1-S2 обслуживаются равновероятно, и далее – в S3, очередь которого ограничена 3-мя заявками. Заявки, покинувшие очередь из-за переполнения, завершают обслуживание в S4.

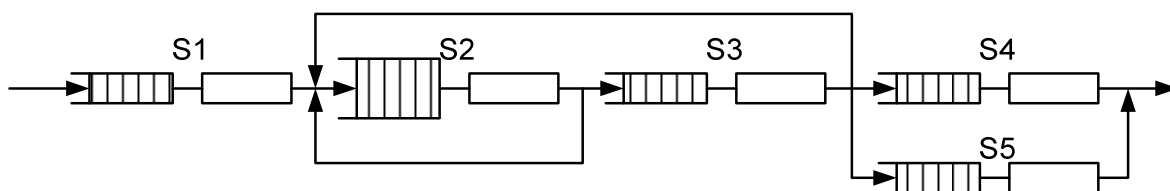
объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	гамма	эксп.	равн.	равн.	равн.	равн.
параметры	$\lambda=0,5; b=3$	$\lambda=0,5$	$1\pm 0,5$	15 ± 5	20 ± 5	16 ± 2

7. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,2; 0,3 и 0,5 соответственно. Общая очередь S2-S4, где заявки обслуживаются согласно типу, ограничена 3-мя заявками. Заявки первого и второго типов при переполнении очереди теряются; третьего типа – завершают обслуживание в S5.



объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	равн.	эсп.	эсп.
параметры	$1 \pm 0,5$	$1 \pm 0,5$	12 ± 2	10 ± 5	$\lambda = 0,2$	$\lambda = 0,2$

8. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,3 и 0,7 соответственно. После обслуживания в S2 заявки второго типа с вероятностью 0,1 возвращаются на повторное обслуживание. После обслуживания в S3 заявки всех типов с вероятностью 0,1 возвращаются на повторное обслуживание. В S4-S5 заявки обслуживаются согласно типу.

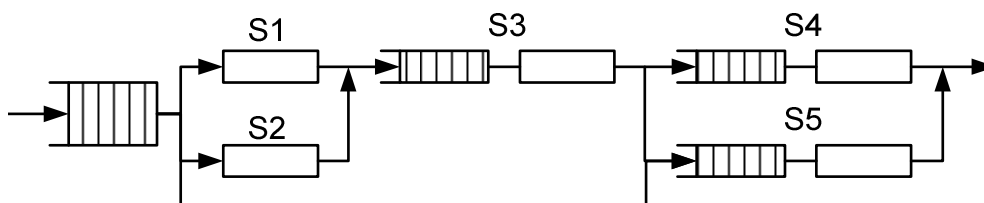


объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	гамма	равн.	равн.
параметры	$\lambda = 0,2$	5 ± 2	$2 \pm 0,5$	$\lambda = 0,5; b = 3$	15 ± 5	10 ± 2

9. Схема – см. вариант 8. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,2; 0,3 и 0,5 соответственно. После обслуживания в S2 заявки первого типа с вероятностью 0,05 возвращаются на повторное обслуживание. После обслуживания в S3 заявки второго типа с вероятностью 0,1 возвращаются на повторное обслуживание. Очередь S4 ограничена 2-мя заявками. В S5 обслуживаются все заявки, покинувшие очередь S4 из-за переполнения, и половина всех заявок третьего типа.

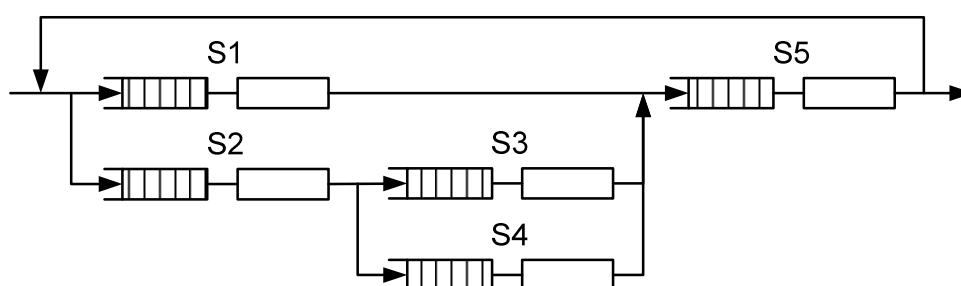
объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	гамма	равн.	равн.	эсп.	эсп.	равн.
параметры	$\lambda = 1; b = 3$	$1 \pm 0,5$	3 ± 1	$\lambda = 0,5$	$\lambda = 0,15$	$2 \pm 0,5$

10. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,3; 0,45 и 0,25 соответственно. Заявки первого и второго типов обслуживаются в S1 и S2 соответственно, заявки третьего типа направляются в S5, где также обслуживаются заявки, покинувшие очередь S4 из-за переполнения (ограничена 3-мя заявками).



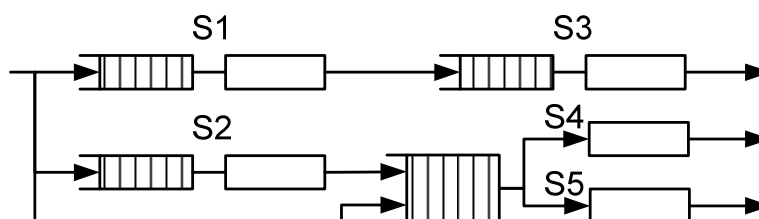
характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	эсп.	равн.	эсп.	равн.
параметры	$\lambda=0,5$	5 ± 1	$\lambda=0,25$	2 ± 1	$\lambda=0,125$	$1\pm 0,5$

11. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,1; 0,4 и 0,5 соответственно. Заявки первого типа обслуживаются в S1, второго и третьего – в S2, и, затем, соответственно типу, в S3-S4. Очередь S4 ограничена 2-мя заявками, заявки, покинувшие очередь из-за переполнения, теряются. С вероятностью 0,1 заявки всех типов с выхода системы возвращаются на вход.



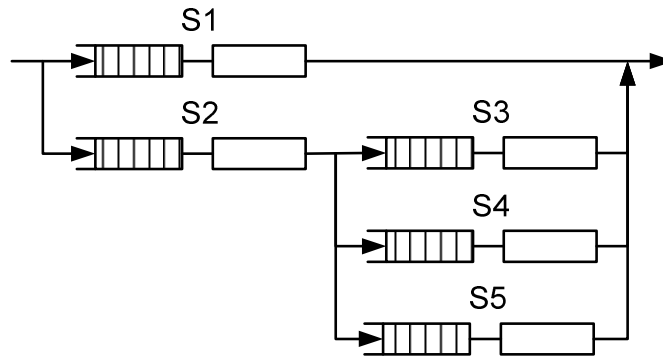
характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	эсп.	равн.	равн.	равн.
параметры	$\lambda=0,5$	5 ± 1	$\lambda=0,5$	8 ± 2	8 ± 2	8 ± 2

12. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,4 и 0,6 соответственно. Заявки первого типа обслуживаются в S1, очередь которого ограничена 3-мя заявками. Заявки, покинувшие очередь из-за переполнения, отправляются в общую очередь S4-S5. Заявки второго типа обслуживаются в S2 и, далее, в S5.



характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	эсп.	равн.	равн.	равн.	эсп.
параметры	$1\pm 0,5$	$\lambda=0,25$	10 ± 2	10 ± 2	10 ± 2	$\lambda=0,5$

13. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,3; 0,45 и 0,25 соответственно, и обслуживаются с вероятностью 0,4 в S1, с вероятностью 0,6 – в S2. Кроме того, очередь S1 ограничена 3-мя заявками; заявки, покинувшие очередь из-за переполнения, также направляются в S2. В S3-S5 обслуживание осуществляется соответственно типу.

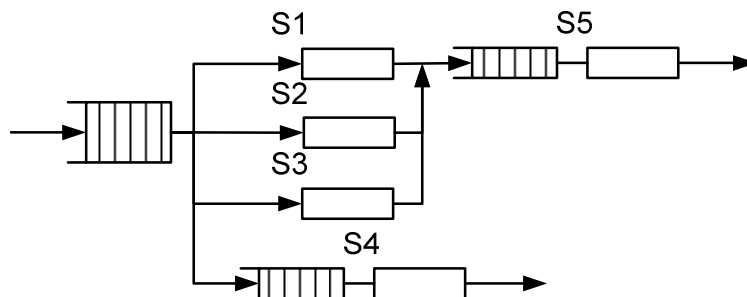


объект / характеристика	входной поток	устройство 1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	равн.	эсп.	эсп.
параметры	$1 \pm 0,5$	9 ± 1	5 ± 1	5 ± 1	$\lambda = 0,5$	$\lambda = 0,1$

14. Схема – см. вариант 13. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,4 и 0,6 соответственно. Очередь S1 ограничена 2-мя заявками. Заявки, покинувшие очередь из-за переполнения, направляются в S2, и далее – в S3, очередь которого ограничена 3-мя заявками. Заявки первого типа, покинувшие очередь из-за переполнения, обслуживаются в S4, второго типа – в S5.

объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	равн.	эсп.
параметры	$\lambda = 1$	$4 \pm 0,5$	$2 \pm 0,5$	$15 \pm 0,5$	$2 \pm 0,5$	$\lambda = 0,8$

15. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,25; 0,5 и 0,25 соответственно. Очередь системы S1-S3, в которой заявки обслуживаются соответственно типу, ограничена 5-ю заявками. Заявки, покинувшие очередь из-за переполнения, направляются в S4 с вероятностью 0,87; остальные – теряются.

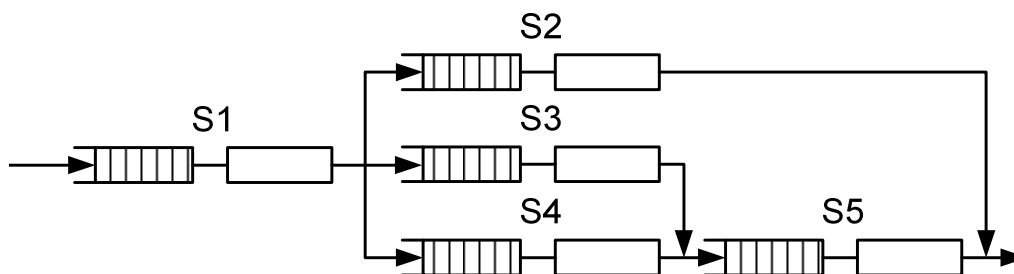


объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	гамма	равн.	равн.	эсп.	эсп.	равн.
параметры	$\lambda=1; b=3$	20 ± 5	13 ± 2	$\lambda=0,028$	$\lambda=0,15$	$2\pm0,5$

16. Схема – см. вариант 15. На вход СМО поступает поток заявок 4-х типов, с вероятностями поступления 0,1; 0,2; 0,3 и 0,4 соответственно. Заявки первого типа с вероятностью 0,8 обслуживаются в S1, и с вероятностью 0,2 – в S2. Заявки второго типа равновероятно обслуживаются в S2 и S3. Заявки третьего типа – в S3; четвертого – в S4, очередь которого ограничена 2-мя заявками, при переполнении очереди заявки теряются.

объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	гамма	эсп.	эсп.	равн.	равн.	равн.
параметры	$\lambda=2; b=3$	$\lambda=0,2$	$\lambda=1$	12 ± 1	16 ± 1	7 ± 1

17. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,15; 0,25 и 0,6 соответственно. Заявки первого типа обслуживаются в S2, заявки второго типа равновероятно в S2 и S3, заявки третьего типа – в S3. Очередь S3 ограничена 3-мя заявками; при переполнении очереди заявки отправляются на обслуживание в S4.

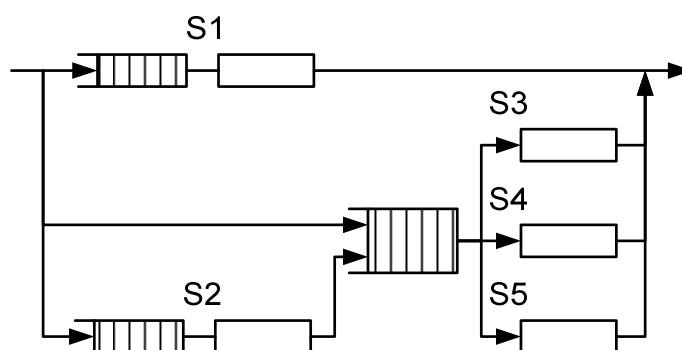


объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	эсп.	эсп.
параметры	$\lambda=1$	$1\pm0,5$	15 ± 2	25 ± 5	$\lambda=0,2$	$\lambda=0,2$

18. Схема – см. вариант 17. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,25 и 0,75 соответственно. Заявки всех типов обслуживаются в S1 и, далее, в S2, очередь которого ограничена 5-ю заявками. При переполнении очереди заявки первого типа отправляются в S3, второго – в S4. Очередь S5 ограничена 3-мя заявками, при переполнении очереди заявки теряются.

объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	эксп.	эксп.	равн.	равн.	равн.
параметры	$1 \pm 0,5$	$\lambda=2$	$\lambda=0,12$	$3 \pm 0,5$	$1,2 \pm 0,2$	$4,2 \pm 1,5$

19. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,42; 0,48 и 0,1 соответственно. Заявки первого типа обслуживаются в S1, заявки второго типа сразу поступают на вход многоканальной системы с общей очередью, третьего типа – предварительно обслуживаются в S2. В многоканальной системе заявки направляются на обслуживание в свободное устройство с наименьшим номером.



объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эксп.	эксп.	эксп.	равн.	равн.	равн.
параметры	$\lambda=0,5$	$\lambda=0,5$	$\lambda=0,05$	12 ± 2	12 ± 2	12 ± 2

20. Схема – см. вариант 19. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,32; 0,3 и 0,38 соответственно. Заявки всех типов с вероятностью 0,3 обслуживаются в S1, с вероятностью 0,5 сразу поступают на вход многоканальной системы с общей очередью, с остаточной вероятностью предварительно обслуживаются в S2. В многоканальной системе обслуживание заявок выполняется согласно типу.

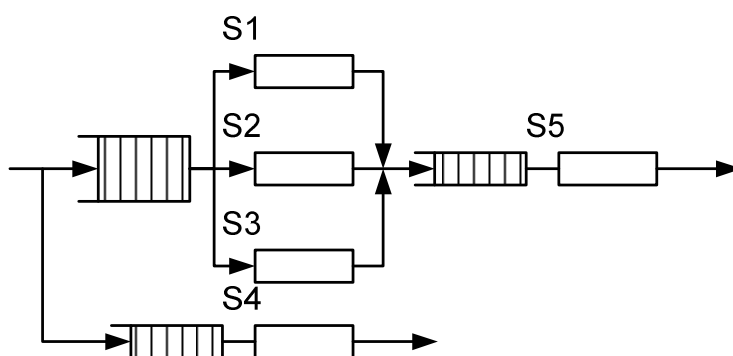
объект \ характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	гамма	эксп.	эксп.	эксп.
параметры	$1 \pm 0,2$	3 ± 1	$\lambda=0,4; b=3$	$\lambda=0,5$	$\lambda=0,5$	$\lambda=0,5$

21. Схема – см. вариант 19. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,52 и 0,48 соответственно. Заявки второго типа обслуживаются в S2, заявки первого типа – в S1, очередь которого ограничена 3-мя заявками. При переполнении очереди заявки первого типа поступают на вход многоканальной системы с общей очередью.

В многоканальной системе заявки с вероятностями 0,4 обслуживаются в S3 и S4, с остаточной – в S5.

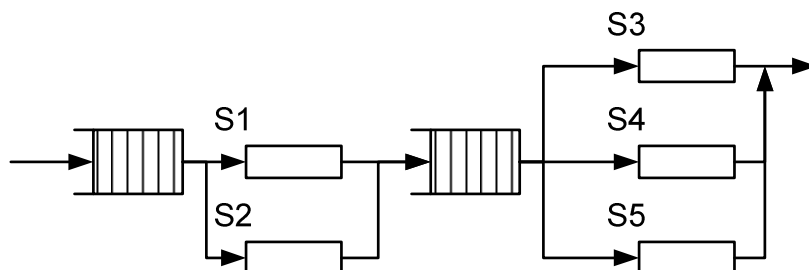
характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	эсп.	гамма	равн.	равн.	равн.
параметры	$1 \pm 0,5$	$\lambda=0,1$	$\lambda=0,35; b=2$	2 ± 1	2 ± 1	2 ± 1

22. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,22; 0,38 и 0,4 соответственно. Заявки второго типа обслуживаются в S4, первого типа – в S1, третьего типа – равновероятно в S2 и S3. Очередь S5 ограничена 5-ю заявками, при переполнении очереди заявки теряются.



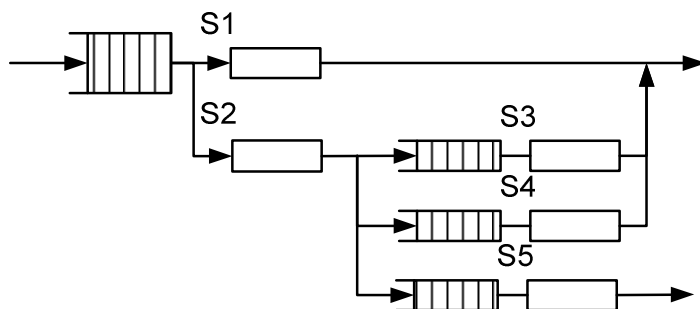
характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	эсп.	равн.	эсп.
параметры	2 ± 1	4 ± 1	$2 \pm 0,5$	$\lambda=0,25$	$5 \pm 0,5$	$\lambda=0,045$

23. На вход СМО поступает поток заявок 4-х типов, с вероятностями поступления 0,25; 0,3; 0,22 и 0,23 соответственно. Заявки с нечетными типами обслуживаются в устройствах с нечетными номерами, во второй многоканальной системе – равновероятно. Заявки с четными типами обслуживаются равновероятно в любых устройствах.



характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	эсп.	эсп.	эсп.
параметры	$1 \pm 0,5$	2 ± 1	2 ± 1	$\lambda=0,25$	$\lambda=0,5$	$\lambda=0,75$

24. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,4 и 0,6 соответственно. Общая очередь ограничена 5-ю заявками, при переполнении очереди заявки теряются. В S1-S2 заявки обслуживаются равновероятно; в S3-S4 согласно типу. Очереди S3-S4 ограничены 3-я заявками, при переполнении очереди заявки обслуживаются в S5.



объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
характеристика	эксп.	эксп.	эксп.	равн.	равн.	гамма
закон распределения	эксп.	эксп.	эксп.	равн.	равн.	гамма
параметры	$\lambda=1$	$\lambda=0,12$	$\lambda=0,1$	27 ± 5	22 ± 5	$\lambda=1,2; b=3$

КОНТРОЛЬ УСПЕВАЕМОСТИ

*«Каждый успех наших знаний ставит
больше проблем, чем решает.»*

Де Бройль, французский физик,
лауреат Нобелевской премии

Первая аттестация

Первая аттестация по курсу проводится на 6-ой учебной неделе для студентов дневной формы обучения. Решение об аттестации или не аттестации студента принимается на основании его индивидуального рейтинга, который складывается из полученных к моменту аттестации баллов рейтинга по всем видам выполняемых работ и результата аттестационного тестирования. Правила проведения и оценивания аттестационного тестирования, а также граничные условия аттестации – см. *Введение в курс* для студентов дневной формы обучения.

В аттестационный тест входят вопросы по всему пройденному к моменту аттестации теоретическому материалу и выполненным работам лабораторного практикума. В вопросах требуется выбрать один или несколько верных ответов из предложенных вариантов. Если в тексте вопроса есть указание на возможную множественность правильных ответов, т. е. если вопрос начинается со слов «выберите все верные утверждения...» или «выберите из перечня характеристики (одну или несколько)...», то правильными ответами для него могут быть один, два и так далее, в пределах – все варианты ответов, приведенные в вопросе. Примеры вопросов первого аттестационного теста:

Пример 1. В системе передачи сообщений используется способ двукратной избыточной последовательной передачи каждого символа. Анализ сбоев передачи осуществляется по контролю четности в переданном сообщении. Выберите верные в отношении этой системы утверждения (одно или несколько):

- А) это система без последствий;
- В) это система с последствием;
- С) эта система требует реализации простого закона управления;
- Д) эта система требует реализации сложного закона управления.

Пример 2. Система, на входе которой сравниваются заданное и текущее значение выходного параметра, реализована по принципу:

- А) обратной связи;
- В) компенсации;
- С) линейного управления;
- Д) управления по возмущению.

Пример 3. В структуре ЛВС организации выделены подсети отделов. Выполненная декомпозиция является:

- А) материальной;
- В) функциональной;
- С) алгоритмической;
- Д) временной.

Вторая аттестация

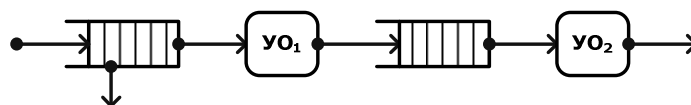
Вторая аттестация по курсу проводится на 12-ой учебной неделе для студентов дневной формы обучения. В аттестационный тест входят вопросы по всему пройденному с момента первой аттестации теоретическому материалу и выполненным работам лабораторного практикума. Кроме вопросов выборочного типа в аттестационный тест входят две задачи: решение двухкритериальной задачи методом графического построения множества Парето и разработка математической модели решения задачи распределения ресурсов по заданным условиям и ограничениям (см. примеры в *Типах экзаменационных задач*).

Примеры вопросов второго аттестационного теста:

Пример 1. Четыре множества заданы разрешающими процедурами. Выберите те множества (одно или несколько), для которых верным является равенство $A=\emptyset$:

- А) $A:=\{x \mid \forall x, x < 2 \cup x > 3\}$;
- В) $A:=\{x \mid \forall x, x < 2 \cap x > 3\}$;
- С) $A:=\{x \mid \exists(x,y), x+y=3\}$;
- Д) $A:=\{x \mid \forall x \exists y, x+y=3\}$.

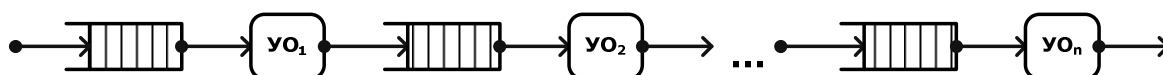
Пример 2. Рассматривается система следующего вида:



Заявки входного потока поступают в систему и последовательно обслуживаются в двух устройствах. Если интервал между поступлением заявок в первую очередь составляет менее заданной величины, то заявки теряются. Целью оптимизации системы является максимизация коэффициентов использования устройств и обеспечение не более 20% потерь заявок из числа поступающих в систему путем варьирования интенсивности входного потока. Выберите все утверждения, верные в отношении поставленной задачи:

- А) это однокритериальная задача;
- В) это многокритериальная задача;
- С) можно использовать метод уступок;
- Д) множество управляющих переменных это $X=\{x_n\}$;
- Е) множество управляющих переменных это $X \cup Q_{cont}$, где $X=\{x_n\}$ и $Q_{cont} \in Q=\{Q_m\}$;
- Ф) можно использовать обычный поиск условного экстремума.

Пример 3. Рассматривается система следующего вида:



Время поступления заявок входного потока и время обслуживания в каждом УО – случайные величины, распределенные по заданному непрерывному закону. Для моделирования данной системы следует использовать типовую математическую...

- А) D-схему;
- В) P-схему;
- С) F-схему;
- Д) Q-схему.

Экзамен

Правила организации, проведения и оценивания экзаменационной работы, а также условия допуска и учет семестрового рейтинга в итоговой оценке освоения курса – см. *Введение в курс* для соответствующей формы обучения.

Вопросы к экзамену

Звездочкой отмечены вопросы, являющиеся обязательными для студентов заочной формы обучения (включены в *часть I* билетов по курсу – см. *Введение в курс* для заочной формы обучения).

Для студентов дневной формы обучения обязательным является полный перечень вопросов.

1. *Определение элемента системы, его функции и связей. Определение системы и ее свойств. Параметризация системы.
2. *Структура системы. Агрегирование и декомпозиция. Виды декомпозиции систем. Пример декомпозиции любого вида¹.
3. Типы системных структур. Природа и функции элементов обратных связей.
4. Принципы системного управления. Управление по отклонению – функциональная схема, пример системы.
5. Принципы системного управления. Управление по возмущению – функциональная схема, пример системы.
6. Принципы системного управления. Комбинированное управление – функциональная схема, пример системы.
7. Особенности информационных систем с точки зрения управления. Управление системами с последствием. Примеры систем с последствием и без.
8. Принципы системного подхода. Пример реализации принципов иерархичности и многомодельности.
9. *Процедуры системного подхода. Задача синтеза систем.
10. *Алгоритм итерационного проектирования систем. Характеристика методов модификации проектов систем.
11. *Базисные множества и концептуальная модель системы в терминах теории множеств.
12. *Типовые математические схемы моделирования систем.
13. *Постановка одно- и многокритериальной задачи поиска и принятия решений².
14. Решение многокритериальных задач сведением к однокритериальной и методом поиска условного экстремума.

¹ Здесь и далее: во всех ответах на вопросы, требующих рассмотрения примеров, следует приводить конкретные примеры требуемого (системы, решения задачи, реализации алгоритма и т.д.) из числа рассмотренных на лекциях или описанных в дополнительной литературе.

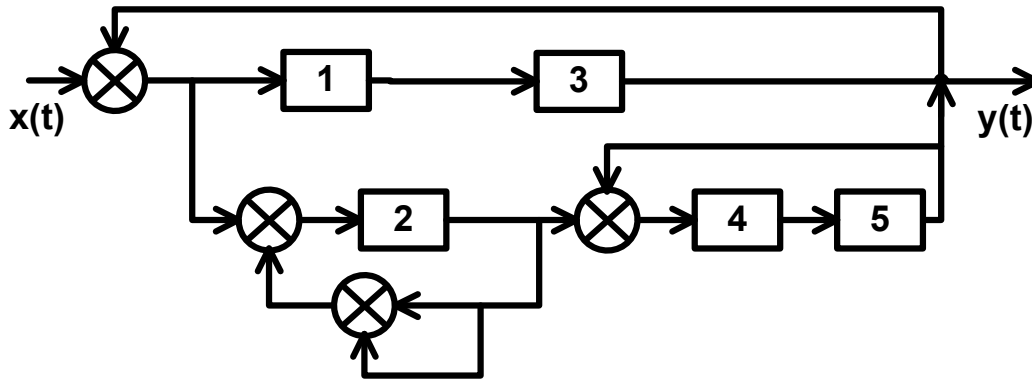
² В ответах на вопросы № 13 – 15 допускается приводить описание постановки задачи определенного вида и методов ее решения в общем виде **или** конкретного примера постановки/решения задачи из числа рассмотренных на лекциях или описанных в дополнительной литературе.

15. Решение многокритериальных задач методом поиска множества Парето. Метод границ.
16. Постановка задач календарного планирования. Диаграммы Ганта.
17. Матрица инцидентности. Пример матрицы инцидентности произвольного орграфа. Матрица структуры связей.
18. Матрица смежности. Пример матрицы смежности произвольного орграфа. Степень матрицы смежности.
19. Топологические модели систем. Алгоритм табличного метода *или* метода узловых потенциалов. Пример получения топологической модели выбранным методом.
20. *Оптимизация структур связей методом построения минимальных связывающих деревьев. Алгоритм Прима *или* Краскала. Пример реализации выбранного алгоритма.
21. Алгоритм формальной декомпозиции систем по методу разбиения графа на максимально сильно связанные подграфы.
22. Графы условий предшествования в задачах составления расписаний и сетевых графиков в задачах сетевого планирования.
23. *Алгоритм построения критического пути сетевого графика. Пример реализации.
24. *Алгоритм построения пути минимальной длины сетевого графика. Определение резервов времени выполнения операций сетевого графика. Примеры реализации.
25. *Определение потока сети. Величина потока. Определение максимальной возможной величины потока в сети. Пример определения.
26. *Алгоритм построения максимального потока в сети. Пример реализации прямого хода алгоритма для произвольного шага алгоритма.
27. *Алгоритм построения максимального потока в сети. Пример реализации обратного хода алгоритма для произвольного шага алгоритма.
28. Определение абстрактного конечного автомата. Пример моделирования дискретной системы абстрактным конечным автоматом.
29. Простые временные сети Петри. Динамика функционирования. Моделирование элементарного цикла обслуживания простой временной сетью Петри.
30. Ингибиторные сети Петри. Моделирование элементарного цикла обслуживания ингибиторной сетью Петри.
31. Определение конфликта в сети Петри. Способы разрешения конфликтов в сетях Петри различных типов. Пример моделирования системы с заданием способа разрешения конфликта.

Типы экзаменационных задач

1. Идентификация типов связей элементов систем и агрегирование.

Пример условия задачи: задана структурная схема системы. Выполнить идентификацию типов связей и агрегирование.



2. *Операции над множествами.* Пример условия задачи: запишите с использованием операций над множествами способ получения множества $X = \{1, 3, 5, 6\}$ из двух исходных множеств $A = \{2, 4, 6\}$ и $B = \{1, 2, 3, 4, 5\}$.

3. *Многокритериальная оптимизация методом построения множества Парето.* Пример условия задачи: построить множество Парето для двухкритериальной задачи вида $F_1(x) \rightarrow \min; F_2(x) \rightarrow \min; x \in \{1, 2, 3, 4, 5, 6, 7\}$. Решение представить графически в приведенной системе координат и записать в виде множества. Исходные данные приведены в таблице.

x	1	2	3	4	5	6	7
$F_1(x)$	1	2	3	2	3	4	3
$F_2(x)$	6	6	6	5	5	5	3

4. *Распределение ресурсов.* Пример условия задачи: потребность системы в двух типах ресурсов для выполнения единичного объема трех видов услуг приведена в таблице.

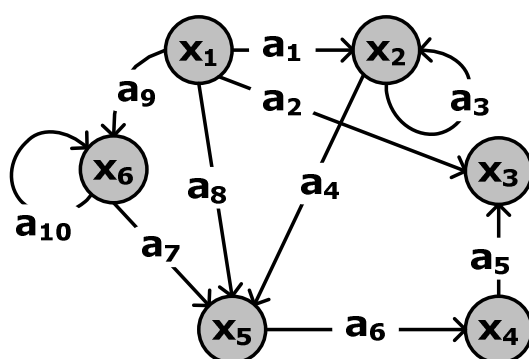
услуга \ ресурс	ресурс	
	P1	P2
Y1	1	3
Y2	2	6
Y3	5	4

Запасы P1 составляют 180 ед., P2 – 200 ед. Доход системы от оказания услуг 1 и 2 составляет 2 у.е., услуги 3 – 3 у.е. Разработать математическую модель для решения задачи распределения ресурсов на выполнение

возможного объема услуг по критериям: максимизация дохода системы при условии обязательного выполнения услуг 1 и 2 не менее, чем на 10% от плана выполнения услуги 3.

5. *Элементы описаний и способы задания графов.* Пример условия задачи: для графа, приведенного на рисунке, укажите элементы описания в соответствии с заданным списком, если такие элементы существуют, и постройте матрицу инцидентности графа (вариант – матрицу смежности).

1. Две любые симметричные дуги графа _____
2. Две любые кратные дуги графа _____
3. Любой путь графа, проходящий через узел x_1 _____

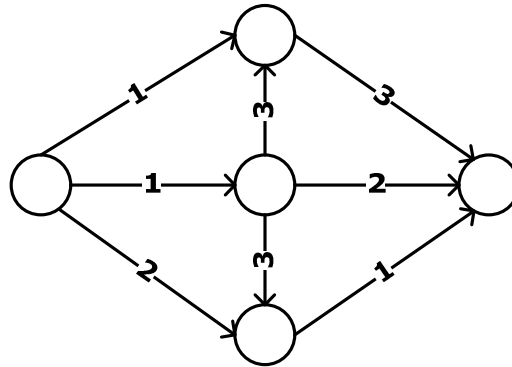


6. *Топологическое моделирование систем методами теории графов.* Пример условия задачи: дана схема электрическая принципиальная. Построить топологическую модель схемы с использованием табличного метода теории графов (вариант – с использованием метода узловых потенциалов). Представить результат топологического моделирования в матричной форме и в форме системы уравнений.

Альтернативное условие задачи данного типа: дана матрица контуров и сечений (вариант – матрица «узел-ветвь»). Требуется изобразить схему электрическую принципиальную и ее граф, и построить фундаментальное дерево графа, отличное от того, с использованием которого была получена исходная матрица контуров и сечений (вариант для заданной матрицы «узел-ветвь» – произвольное фундаментальное дерево).

7. *Структурная оптимизация систем методами теории графов.* Пример условия задачи: дан граф топологии компьютерной сети с заданными удельными приведенными затратами на передачу данных по каналам связи (веса ребер). Выполнить оптимизацию структуры сети по критерию минимума приведенных затрат с использованием алгоритма Прима (вариант – алгоритма Краскала).

8. *Определение операционных резервов времени и критических путей в сетевом планировании методами теории графов.* Пример условия задачи: задан сетевой график. Веса дуг равны времени выполнения операций. Определить операцию (операции) с максимальным резервом времени.



9. *Построение перестановочных моделей сетевого планирования.* Пример условия задачи: построить И/ИЛИ граф процесса обработки запросов в распределенной системе с учетом следующих данных:

- обработка запроса требует выполнения 4-х операций;
- в начале процесса обработки должны быть выполнены операции 1 и 2 в любом порядке и только после выполнения этих операций могут быть выполнены операции 3 и 4 в указанном порядке;
- в системе два устройства; операции 1 и 2 выполняются любым устройством; 3 и 4 – только вторым устройством.

10. *Оптимизация сетевых потоков.* Пример условия задачи: дана сеть с заданным потоком и пропускными способностями дуг, а также индексация данной сети, полученная в результате выполнения прямого хода алгоритма построения максимального потока. Выполнить обратный ход алгоритма для текущего шага. Определить максимально возможную величину потока в заданной сети.

Альтернативное условие задачи данного типа: дана сеть с заданным потоком и пропускными способностями дуг. Выполнить прямой ход алгоритма построения максимального потока в сети для текущего шага. Определить максимально возможную величину потока в заданной сети.

ЭЛЕКТРОННЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ «СИСТЕМОТЕХНИКА»

Помимо данного учебно-методического комплекса для успешного освоения курса «Системотехника», выполнения лабораторных работ и контрольной работы студентов заочного отделения рекомендуется дополнительно ознакомиться с материалами, которые выдаются в электронном виде. Характеристика материалов приведена в табл. 24.

Таблица 24

Папка	Материалы	Описание
ПО лабораторного практикума 	 PetriNets_InProject	Программа моделирования простых временных, ингибиторных, приоритетных и вероятностных сетей Петри PetriNets InProject. Используется в лабораторных работах № 5, № 6 и при выполнении контрольной работы
	 VisSim	Дистрибутив среды моделирования технических объектов, систем и их элементов VisSim. Используется в лабораторной работе № 1
	 Arena	Дистрибутив среды имитационного моделирования систем массового обслуживания Arena. Используется в лабораторной работе № 7 и при выполнении контрольной работы
ПРИМЕРЫ к лабораторному практикуму 	 Пример моделирования входного воздействия Файл "VSM"	Примеры выполнения лабораторной работы № 1. Файлы реализаций в системе VisSim моделей источников воздействий внешней среды на систему и решения задач агрегирования линейных систем
	 Пример агрегирования Файл "VSM" 2 КБ	
ПРИМЕРЫ к лабораторному практикуму 	 Управление ресурсами Лист Microsoft Excel 166 КБ	Пример выполнения лабораторной работы № 2. Реализация в MS Excel альтернативных вариантов распределения лимитированных ресурсов системы по нескольким функциям с учетом заданных ограничений по приоритетности выполнения работ и дополнительных условий

Папка	Материалы	Описание
	 Пример анализа СТХ Лист Microsoft Excel 1 207 КБ	Пример выполнения лабораторной работы № 3. Реализация в MS Excel расчета структурно-топологических характеристик систем и идентификации типов структур
ПРИМЕРЫ к лабораторному практикуму 	 Пример декомпозиции структуры ИС Лист Microsoft Excel 722 КБ	Пример выполнения лабораторной работы № 4. Реализация в MS Excel решения задачи формальной декомпозиции структуры ИС; расчета и анализа структурных свойств системы
	 Ограниченная очередь Файл "PNE" 2 КБ  Ограниченный ресурс Файл "PNE" 3 КБ  Параллельные узлы Файл "PNE" 2 КБ	Примеры выполнения лабораторной работы № 5. Файлы реализаций в программе PetriNets InProject сетей Петри для моделирования динамики функционирования систем с параллельным осуществлением процессов; с ограничением ресурса длины очереди; с ограничением используемого ресурса
	 Пример 1 (ограничение времени ожидания) Файл "PNE"  Пример 2 (приоритетное обслуживание) Файл "PNE"  Пример 3 (отказ и остановка по условию) Файл "PNE"	Примеры выполнения лабораторной работы № 6. Файлы реализаций в программе PetriNets InProject моделей устройств СМО с ограничением времени ожидания; с неоднородным входным потоком и абсолютным приоритетом обслуживания; с отказом основного устройства и остановкой по условию
	 Пример 1 (визуализация динамики функционирования) Arena Model Document  Пример 2 (общая очередь) Arena Model Document 346 КБ  Пример 3 (ограниченная очередь) Arena Model Document 319 КБ	Примеры выполнения лабораторной работы № 7. Файлы реализаций в пакете Arena моделей узла системы массового обслуживания (СМО) с визуализацией динамики функционирования; узла СМО с общей очередью; СМО с ограниченной очередью

Литература

1. Bertalanffy, L. General System Theory – A Critical Review // L. Bertalanffy // General Systems. – 1962. – vol. VII. – P. 1 – 20.
2. Норенков, И.П. Автоматизированное проектирование: учеб. пособие / И.П. Норенков. – М.: Высш. шк., 2000. – 188 с.
3. Веревкин, А.П. Теория систем: учеб. пособие / А.П. Веревкин, О.В. Кирюшин. – Уфа: УГНТУ, 2003. – 100 с.
4. Токарев, М. Популярная системотехника. Что надо знать, чтобы правильно построить и использовать искусственную систему? / М. Токарев // Кейс. – 2006. – № 2[6]. – С. 44 – 49.
5. Хачатурова, С.М. Математические модели системного анализа: учеб. пособие / С.М. Хачатурова. – Новосибирск: НГТУ, 2002. – 62 с.
6. Казиев, В.М. Введение в анализ, синтез и моделирование систем / В.М. Казиев // Интернет-Университет Информационных Технологий [Электронный ресурс]. – 2007. – Режим доступа: <http://www.intuit.ru/department/expert/intsys/>. – Дата доступа: 01.05.2008.
7. Черняховская, Л.Р. Моделирование систем: конспект лекций / Л.Р. Черняховская. – Уфа: УГАТУ, 2007. – 81 с.
8. Анхимюк, В.Л. Теория автоматического управления / В.Л. Анхимюк, О.Ф. Опейко, Н.Н. Михеев. – Минск: Дизайн ПРО. 2000. – 352 с.
9. Александров, А.А. Системный подход / А.А. Александров // СЮ, 2007. – № 4. – С. 42 – 46.
10. Шалагинов, А.В. Моделирование: экскурс в дисциплину / А.В. Шалагинов. – Новосибирск: НГТУ, 1998. – 24 с.
11. Громов, Ю.Ю. Системный анализ в информационных технологиях: учеб. пособие / Ю.Ю. Громов [и др.]. – Тамбов: ТГУ, 2004. – 176 с.
12. Романов, В.Н. Системный анализ для инженеров / В.Н. Романов. – СПб.: СЗГЗТУ, 2006. – 186 с.
13. Хэмди, А. Таха. Введение в исследование операций / А. Таха Хэмди. – Киев: Диалектика, 2007. – 912 с.
14. Рубцов, С. Вчера и сегодня исследования операций / С.Рубцов // БОСС, 2000. – № 12. – С. 38 – 42.
15. Зайченко, Ю.П. Исследование операций / Ю.П. Зайченко. – Киев: ИПСАНТУУ «КПИ», 2004. – 170 с.
16. Шестопалова, О.Е. Моделирование: учебно-методический комплекс / О.Е. Шестопалова. – Новополюцк: ПГУ, 2009. – 302 с.
17. Костюкова, О.И. Исследование операций: учеб. пособие / О.И. Костюкова. – Минск: БГУИР, 2003. – 94 с.
18. Мухин, О.И. Моделирование систем: курс лекций / О.И. Мухин. – Пермь: ПГТУ, 2000. – 325 с.
19. Лапшева, Е.Е. Введение в теорию графов / Е.Е. Лапшева // Метод. материалы Поволжского центра новых информац. технологий при Саратов. гос. ун-те [Электронный ресурс]. – 2008. – Режим доступа: <http://www.sgu.ru/prcnit/teach/2.php>. – Дата доступа: 24.06.2009.
20. Князьков, В.С. Введение в теорию графов / В.С. Князьков, Т.В. Волоченская // Интернет-Университет Информационных Технологий [Электронный ресурс]. – 2006. – Режим доступа: <http://www.intuit.ru/department/algorithms/ingrth/lit.html>. – Дата доступа: 29.04.2008.

21. Рабкин, Е.Л. Дискретная математика: булевы функции и элементы теории графов / Е.Л. Рабкин, Ю.Б. Фарфоровская // Санкт-Петербург. гос. ун-т телекоммуникаций [Электронный ресурс]. – Режим доступа: <http://dvo.sut.ru/libr/himath/w163rabk/index.htm> – Дата доступа: 29.06.2009.
22. Норенков, И.П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры / И.П. Норенков, В.Б. Маничев. – М.: Радио и связь, 2000. – 334 с.
23. Прилуцкий, М.Х. Методические указания для самостоятельной работы студентов по курсу «Моделирование сложных систем» при изучении темы «Распределение ресурсов в многоиндексных иерархических системах» / М.Х. Прилуцкий, Л.Г. Афраймович. – Нижний Новгород.: НГУ, 2006. – 18 с.
24. Прилуцкий, М.Х. Распределение однородного ресурса в иерархических системах древовидной структуры / М.Х. Прилуцкий // Идентификация систем и задачи управления SICPRO: труды междунар. конф., Москва, 26 – 28 сент. 2000 г. – М., 2000. – С. 2038-2049.
25. Липский, В.А. Комбинаторика для программистов / В.А. Липский. – М.: Мир, 1988. – 200 с.
26. Сурмин, Ю.П. Теория систем и системный анализ: учеб. пособие / Ю.П. Сурмин. – К.: МАУП, 2003. – 368 с.
27. Советов, Б.Я. Моделирование систем / Б.Я. Советов, А.М. Яковлев. – М.: Высш. шк., 2001. – 374 с.
28. Курс лекций по дисциплине «Основы системного подхода и системного анализа» // Междунар. ин-т подготовки кадров и информации [Электронный ресурс]. – 2008. – Режим доступа: http://www.mipki.spb.ru/topics/Mipki_sysan2.pdf – Дата доступа: 26.10.2008.
29. Градосельская, Г.В. Современные подходы к измерению сетевых взаимодействий / Г.В. Градосельская // Материалы интернет-конференции «Сетевые формы межфирменной кооперации: стратегические вызовы и конкурентные преимущества новых организаций XXI века» [Электронный ресурс]. – 2009. – Режим доступа: <http://www.ecsocman.edu.ru/db/msg/152288.html> – Дата доступа: 28.08.2009.
30. Матула, Д.В. Методы теории графов в алгоритмах кластер-анализа / Д.В. Матула. – М.: Мир, 1980. – 121 с.
31. Воронцов, К.В. Алгоритмы кластеризации и многомерного шкалирования / К.В. Воронцов // Материалы курса лекций «Машинное обучение» МФТИ [Электронный ресурс]. – 2008. – Режим доступа: <http://www.ccas.ru/voron/download/Clustering.pdf> – Дата доступа: 21.09.2009.
32. Алексеев, В.Е. Графы и алгоритмы / В.Е. Алексеев, В.А. Таланов // Интернет-Университет Информационных Технологий [Электронный ресурс]. – 2007. – Режим доступа: <http://www.intuit.ru/departments/algorithms/gaa>. – Дата доступа: 12.12.2009.
33. Костюкова, Н.И. Графы и их применение / Н.И. Костюкова // Интернет-Университет Информационных Технологий [Электронный ресурс]. – 2006. – Режим доступа: <http://www.intuit.ru/departments/algorithms/graphsuse>. – Дата доступа: 12.12.2009.
34. Бурков, В.Н., Заложнев А.Ю., Новиков Д.А. Теория графов в управлении организационными системами / В.Н. Бурков, А.Ю. Заложнев, Д.А. Новиков. – М.: Синтег, 2001. – 124 с.
35. Блюмин, С.Л. Дискретное моделирование систем автоматизации и управления / С.Л. Блюмин, А.М. Корнеев. – Липецк: ЛЭГИ, 2005г. – 124 с.

36. Зюбин, В.Е. Программирование информационно-управляющих систем на основе конечных автоматов: учеб.-метод. пособие / В.Е. Зюбин. – Новосибирск: НГУ, 2006. – 96 с.
37. Murata, T. Petri Nets: Properties, Analysis and Applications / T. Murata // Proceedings of the IEEE. – 1989. – Voll. 77. – № 4.
38. Khoromskaia, V. Petri Nets Based Modelling of Control Flow / V. Khoromskaia // Вестник РУДН, сер. «Прикладная и компьютерная математика». – 2004. – № 1. – С. 74 – 84.
39. Shestopalova, O.E. Ingproject – Petri Nets Software Tool for Queuing Systems Simulation / O.E. Shestopalova, S.A. Skriplenok // Проблемы проектирования и производства РЭС: сб. материалов междунар. научно-техн. конф, Секция «Информатика», ПГУ, Новополоцк, 10 – 12 мая 2008 г. – Новополоцк: ПГУ, 2008. – С. 91 – 94.
40. Шестопалова, О.Е. Защита программных средств с использованием сетей Петри / О.Е. Шестопалова, С.А. Скрипленок // Вестн. Полоцк. гос. ун-та, Сер. С, Фундаментальные науки. – Новополоцк: ПГУ, 2009. – № 9. – С. 13 – 19.
41. Зюбин, В.Е. Программирование информационно-управляющих систем на основе конечных автоматов / В.Е. Зюбин. – Новосибирск: НГТУ, 2006. – 97 с.
42. Шестопалова, О.Е. Модификация матричного алгоритма разбиения графов для решения задач декомпозиции / О.Е. Шестопалова, П.В. Кизин // Вестн. Полоцк. гос. ун-та, сер. С, Фундаментальные науки. – Новополоцк: ПГУ, 2011. – № 4. – С. 26 – 33.

Учебное издание

ШЕСТОПАЛОВА Ольга Евгеньевна

СИСТЕМОТЕХНИКА

Учебно-методический комплекс
для студентов специальности
1-40 02 01 «Вычислительные машины, системы и сети»

Редактор *Т. А. Дарьянова*

Дизайн обложки *В. А. Виноградовой*

Подписано в печать 31.11.11. Формат 60х84 1/16. Гарнитура Таймс. Бумага офсетная.
Ризография. Усл. печ. л. 17,28. Уч.-изд. л. 16,32. Тираж 27 экз. Заказ 1934.

Издатель и полиграфическое исполнение –
учреждение образования «Полоцкий государственный университет»

ЛИ № 02330/0548568 от 26.06.09

ЛП № 02330/0494256 от 27.05.09

211440, г. Новополоцк, ул. Блохина, 29