МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Юго-Западный государственный университет

(ЮЗГУ)

Кафедра вычислительной техники

УТВЕРЖДАЮ

Проректор по учебной работе

________ О.Г. Локтионова

«___» ________ 2016 г.

МЕТРИЧЕСКАЯ ТЕОРИЯ ЭВМ И ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Методические указания по выполнению лабораторных работ
для студентов направления подготовки 09.03.01

Курск 2016

УДК 621.3
Составитель: Э.И. Ватутин

Рецензент
Кандидат технических наук, доцент *А.Г. Спеваков*

Метрическая теория ЭВМ и вычислительных систем: методические указания по выполнению лабораторных и практических работ по дисциплине «Теоретические основы организации многопроцессорных комплексов и систем» / Юго-Зап. гос. ун-т; сост.: Э.И. Ватутин; Курск, 2016. 31 с.

Излагаются методические основы последовательно выполняемых на ЭВМ лабораторных работ, позволяющих получить практические навыки применения метрической теории вычислительных систем, математических методов программно-аналитического и имитационного моделирования для определения численных значений системной производительности вычислительной системы, а также определению вычислительной сложности алгоритмов входных заданий вычислительной системы, дисциплин их обслуживания, требуемого быстродействия процессора, оптимального числа процессоров.

Предназначены для студентов направления подготовки 09.03.01 «Информатика и вычислительная техника».

Текст печатается в авторской редакции

Подписано в печать _________. Формат 60x84 1/16.
Усл. печ. л.   Уч. – изд.л.   Тираж 30 экз. Заказ   . Бесплатно.
Юго-Западный государственный университет
305040, Курск, ул. 50 лет Октября, 94.

Содержание

ЛАБОРАТОРНАЯ РАБОТА № 1. ОПРЕДЕЛЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ АЛГОРИТМА ПРИКЛАДНОЙ ПРОГРАММЫ 6
1. ЦЕЛЬ РАБОТЫ 6
2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПОСТРОЕНИЮ МАРКОВСКОЙ МОДЕЛИ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ АЛГОРИТМА 6
3.ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ........................... 10
4.СОДЕРЖАНИЕ ОТЧЕТА 12
5.КОНТРОЛЬНЫЕ ВОПРОСЫ 12
ЛАБОРАТОРНАЯ РАБОТА № 2. ВЫБОР БЫСТРОДЕЙСТВИЯ ПРОЦЕССОРА И ДИСЦИПЛИН ОБСЛУЖИВАНИЯ ВХОДНЫХ ЗАДАНИЙ.. 14
1.ЦЕЛЬ РАБОТЫ 14
2.МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРИМЕНЕНИЮ ТЕОРИИ СМО ДЛЯ ОЦЕНКИ ХАРАКТЕРИСТИК ПРОИЗВОДИТЕЛЬНОСТИ ВС .. 14
3.ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ........................... 27
4.СОДЕРЖАНИЕ ОТЧЕТА 28
5.КОНТРОЛЬНЫЕ ВОПРОСЫ 29
БИБЛИОГРАФИЧЕСКИЙ СПИСОК 30

Лабораторная работа №1

ОПРЕДЕЛЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ АЛГОРИТМА ПРИКЛАДНОЙ ПРОГРАММЫ

1. ЦЕЛЬ РАБОТЫ

Приобретение практических навыков определения с помощью ЭВМ основных параметров рабочей нагрузки ВС, создаваемой прикладной программой, и освоение методики подготовки исходных данных для программно-аналитической модели вычислительной сложности алгоритма.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПОСТРОЕНИЮ МАРКОВСКОЙ МОДЕЛИ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ АЛГОРИТМА

Основными параметрами рабочей нагрузки ВС, создаваемом прикладной программой ее входного задания являются:

- время обработка программы процессором:

$$\upsilon = \frac{\theta}{B},$$

где θ – среднее значение вычислительной сложности (трудоемкости) алгоритма программы с размерностью $\left[\theta\right]$ = эквивалентных операций (ЭО); B – быстродействие процессора с размерностью $\left[B\right]$ = ЭО/С;

- размер рабочей области, занимаемой программой и ее данными в оперативной памяти $V_{ОЗУ}$;

- среднее время обращения к ВЗУ за время выполнения программы:

$t_{ВЗУ} = \theta_{ф} t_{o}$,

где $\theta_{ф}$ – средний объем информации, передаваемый между ОЗУ и ВЗУ при обращении к файлам в ходе выполнения программы, с размерностью $\left[\theta_{ф}\right]$ = байт; t_{o} – время обращения к ВЗУ для передачи одного байта информации между ОЗУ и ВЗУ;

- среднее время ввода информации:

$$t_{вз} = \theta_{вв} t_{вип},$$

где $\theta_{вв}$ – средний объем информации, вводимой в ОЗУ за время выполнения программы, с размерностью $\left[\theta_{вв}\right]$ = байт; $t_{вип}$ – среднее время ввода и передачи в ОЗУ одного байта информации;

- среднее время вывода информации:

$$t_{выв} = \theta_{ф} t_{пи},$$

где $\theta_{ф}$ – средним объем выводимой информации в ходе выполнения программы с размерностью $\left[\theta_{ф}\right]$ = байт; $t_{пи}$ – среднее время передачи информации из ОЗУ при выводе одного байта информации.

При детальном анализе производительности и профиля загрузки устройств ЭВМ и ВС [1] могут потребоваться следующие дополнительные параметры рабочей нагрузки: среднее время ввода информации с внешнего источника (например, HDD или SSD); среднее время обращения к ОЗУ (чаще всего учитывается в составе названного выше параметра и); среднее время обращения к ПЗУ; среднее время передачи информации по каналу ввода-вывода или по системной шине под управлением контролера прямого доступа в память; среднее время передачи информации через адаптер межпроцессорной связи, адаптер канал-канал или по логическому каналу связи между прикладными программами в сети ЭВМ или локальной вычислительной сети и т.п.

Значительная часть исходных данных $\theta, \theta_{ф}, \theta_{вв}$, требуемых для вычисления параметров рабочей нагрузки, может быть найдена в результате аналитическою (теоретического) анализа процессоров в ходе случайного функционирования алгоритмов прикладных программ. Случайное функционирование алгоритма моделируется вероятностными цепями Маркова [1–3]. Алгоритм представляется в виде частично взвешенного графа Маркова со следующими тремя видами вершин-операторов.

Функциональный оператор (рис. lа) отображает затраты времени на арифметико-логическое преобразование данных и взвешивается числом эквивалентных операции (ЭО) K_i, требуемых для осуществления названного преобразования:

$$K_i = \frac{\sum_{j=1}^{m_i} t_{ij}}{t_{эо}},$$

где m_i – число операций в данном i-м преобразовании; t_{ij} – время выполнения процессором j-й операции в i-м преобразовании; $t_{эо}$ – время выполнения базовой эквивалентной операции, в качестве которой выбирается любая короткая операция из системы команд процессора (например, сложение). Данный тип вершины имеет любое число входов и только один выход.

Оператор ввода-вывода (рис. 1б) отображает затраты времени на обращение к определенному файлу с целью передачи (записи или чтения) требуемого по исследуемой программе количества информации l_i. Вершина взвешивается числом l_i в байтах, соответствующим однократному обращению к файлу; имеет любое число входов и только один выход.

Оператор перехода (рис. 1в) отображает правило выбора одного из возможных путей дальнейшего развития вычислительного процесса в зависимости от значения предиката и диапазонов изменения его аргументов. Он имеет любое число входов и не менее двух выходных дуг, каждая из которых взвешивается вероятностью перехода. Сумма вероятностей по всем выходным дугам должна равняться единице. Для отображения вычислительной сложности предиката используется дополнительный функциональный оператор, включаемый в граф непосредственно перед соответствующим оператором перехода (рис. 1.1г).

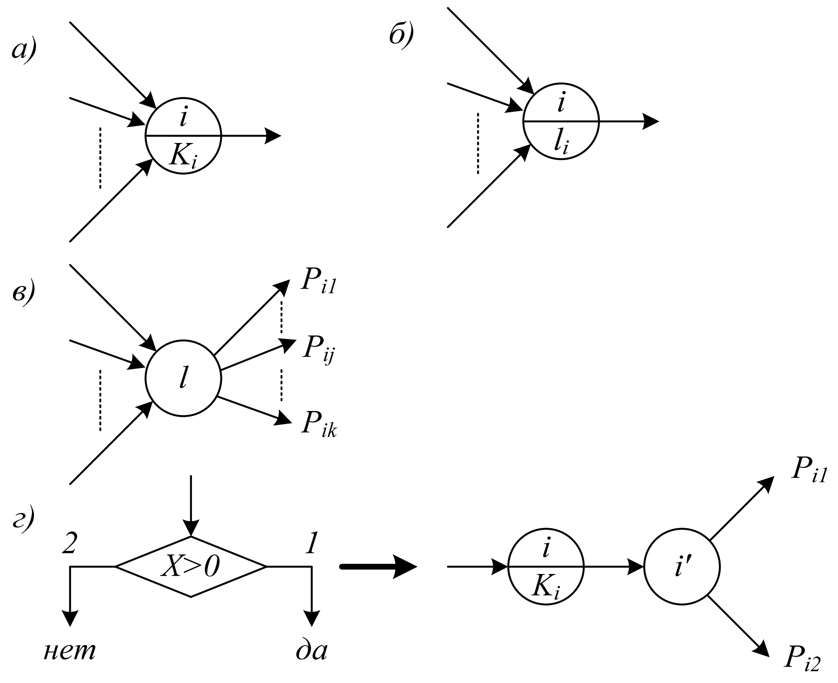

Рис. 1.1. Условные обозначения вершин операторов Маркова

Вероятности переходов определяются свойствами имитируемого предиката, например, если предикат имеет вид:

$$x < 0 \rightarrow да,$$
$$x \geq 0 \rightarrow нет,$$

где x – переменная с диапазоном изменения, найденным путем анализа исследуемой программы, например:

$$-1 < x < 3,$$

тогда, полагая, что переменная x подчиняется равномерному закону распределения случайных величин, получим соответствующие значения вероятностей переходов:

$$P_1 = \frac{1}{4}, P_2 = \frac{3}{4}.$$

Если предикат применен для формирования программного цикла с параметром N, определяющим число прохождений цикла (например, $N \leq 9$), то, учитывая, что полная группа случайных событий в данном случае состоит из 10 событий, значения вероятностей переходов для вхождения в цикл и выхода из цикла соответственно равны:

$$P_1 = 0{,}9,\ P_2 = 0{,}1.$$

Кроме названных вершин-операторов в графе Маркова используются начальная вершина без входов и только одним выходом и конечная вершина с любым числом входов и без выхода.

Алгоритм любой программы может быть представлен в виде графа Маркова. Наиболее наглядный путь его построения состоит в следующем:

- текст моделируемой программы на алгоритмическом языке заменяется эквивалентной блок-схемой, или граф-схемой, ее алгоритма (ГСА);

- каждый арифметический оператор или группа операторов на линейном участке программы заменяется функциональным оператором графа Маркова;

- каждый оператор условного перехода ГСА замещается в графе Маркова оператором, имитирующим затраты времени на вычисление значения предиката, и оператором перехода;

- все процедуры ввода, вывода и обращений к ВЗУ отображаются в графе Маркова соответствующими операторами ввода-вывода.

При допущении, что все вероятности переходов являются постоянными величинами и не зависят от хода выполнения алгоритма в прошлом до данного перехода, процесс выполнения алгоритма может быть представлен в виде марковского дискретного случайного процесса. Если в графе Маркова общее количество вершин равно *K*, то марковский процесс описывается *K* состояниями $S_1, S_2, ..., S_K$, каждое из которых соответствует событию пребывания в одной из вершин графа $V_1, V_2, ..., V_K$, где V_i – условное обозначение одной вершины графа Маркова. Состояния $S_1, S_2, ..., S_{K-1}$ – невозвратные, S_K – конечное поглощающее состояние.

Описанная марковская модель алгоритма позволяет найти средние значения чисел $n_1, n_2, ..., n_{K-1}$ пребывания марковского процесса в невозвратных состояниях $S_1, S_2, ..., S_{K-1}$ как решение следующей системы линейных алгебраических уравнений, описывающих граф Маркова [3]:

$$n_i = \delta_{1i} + \sum_{j=1}^{K-1} P_{ij} n_j, \; i = \overline{1, K-1}, \tag{1.1}$$

где P_{ij} – вероятность перехода из вершины j в вершину i; δ_{ij} – символ Кронскера, который в общем случае определяется в виде:

$$\delta_{ij} = \begin{cases} 1, i = j, \\ 0, i \neq j. \end{cases}$$

В данной системе уравнений $j = 1$, и поэтому символ δ_{1i} равен единице только при $i = 1$, т.е. $\delta_{11} = 1$. Это соответствует случаю, когда алгоритм начинается с вершины V_1, и поэтому число пребываний в начальном состоянии S_1 сразу же будет равно единице:

$$n_1 = 1.$$

В остальных $(K-2)$-х невозвратных состояниях процесс попадает в i-е состояние S_i (в вершину V_i) из других $(K-2)$-х вершин под номерами

$$j = 1, 2, ..., i-1, i+1, ..., K-1$$

с вероятностями P_{ij}. Поэтому, если процесс находился в состоянии S_j в среднем n_j раз, то он переходил из состояния S_j в состояние S_i в среднем $P_{ji} n_j$ раз. Тогда для получения среднего числа n_i, пребываний процесса в состоянии S_i (вершина V_i) достаточно просуммировать произведения $P_{ji} n_j$ по всем $(K-2)$-м состояниям S_j, $j \neq i$, из которых процесс может перейти в данное состояние S_i.

Найденные описанным выше способом средние числа $\{n_i\}$ пребывания алгоритма в вершинах $V_1, ..., V_{K-1}$ за один его прогон (транзакцию) позволяют достаточно просто вычислить следующие характеристики его вычислительной сложности [3]:

• среднее число эквивалентных операций, выполняемых за один прогон (транзакцию) алгоритма:

$$\theta = \sum_{V_i \in S_\phi} n_i k_i ,$$

где S_ϕ – подмножество функциональных операторов графа Маркова;

• среднее количество информации, передаваемой между ОЗУ и ВЗУ при обращении к файлу имени F_h за один прогон алгоритма:

$$\theta_{o\phi} = \sum_{V_i \in S_h} n_i l_i ,\, h = \overline{1, H} ,$$

где S_h – подмножество вершин ввода-вывода, в которых отображено обращение к файлу F_h; H – общее число имен файлов, к которым происходит обращение при выполнении моделируемой программы;

• среднее количество информации, передаваемой при одном обращении к файлу имени F_h:

$$\theta_h = \frac{1}{\sum_{V_i \in S_h} n_i} \sum_{V_i \in S_h} n_i l_i .$$

Порядок системы уравнений (1.1), описывающих реальные алгоритмы, может оказаться достаточно высоким. Поэтому в инженерной практике широко применяются программно-аналитические марковские модели алгоритмов, одна из которых ("Алгоритм") используется в данной лабораторной работе. Основные задачи пользователя программно-аналитической модели:

- освоение описанной выше методики построения марковских моделей алгоритмов;

- освоение технологии обработки информации программой "Алгоритм";

- подготовка данных для определения на ПЭВМ вычислительной сложности алгоритма;

- интерпретация и обработка результатов вычислений.

3. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

3.1. Подготовка исходных данных:

3.1.1. Выбрать одну из стандартных прикладных программ в соответствии с индивидуальным вариантом. Для получения достаточно большой вычислительной сложности рекомендуется выбирать программу, содержащую циклы и вложенные подциклы, назначая большое количество итераций.

3.1.2. По тексту программы составить блок-схему (граф-схему) се алгоритма (ГСА).

3.1.3. Конкретизировать численные значения входных параметров исследуемой стандартной программы. Численные значения параметров циклов следует выбирать достаточно большими для получения вычислительной сложности, целесообразной для обработки исследуемой программы на ЭВМ.

3.1.4. На основании ГСА построить граф Маркова. Пронумеровать его вершины $(K-1)$-го невозвратных состояний.

3.1.5. Вычислить весовые коэффициенты K_i, L_i, P_{ij}, вершин и дуг переходов. Веса указать в вершинах и на дугах схемы графа Маркова.

3.2. Выполнение расчетов на ЭВМ:

3.2.1. Запустить программу "Алгоритм".

3.2.2. Ввести, пользуясь меню программы, параметры графа Маркова. Число вершин *K* вводить на единицу больше помеченных на графе $(K-1)$-го невозвратных состояний. При вводе параметров оператора перехода вначале указывается номер соседней вершины графа по направлению перехода, а затем – значение вероятности перехода. Номер последней *K*-й поглощающей вершины вводить цифрой нуль. Направление перехода к последней *K*-й поглощающей вершине указывается ее действительным *K*-тым номером, а при вводе описания этой последней *К*-й вершины величина ее номера указывается равной нулю.

3.2.3. Выполнить вычисления на ЭВМ.

3.2.4. Вывести из меню результаты расчетов на экран. Зарегистрировать полученные результаты.

3.2.5. Интерпретировать полученные результаты, сравнивая предполагаемые и полученные на ЭВМ числа пребываний $\{n_i\}$ в вершинах графа Маркова.

3.3. Предъявить преподавателю зарегистрированные результаты и их интерпретацию.

4. СОДЕРЖАНИЕ ОТЧЕТА

4.1. Краткое описание методики составления марковской модели алгоритма прикладной программы.

4.2. Текст на алгоритмическом языке моделируемой прикладной программы. Словесное и математическое описание принципа ее функционирования.

4.3. Блок-схема (ГСА) алгоритма исследуемой программы.

4.4. Схема графа Маркова с указанием номеров и весов его вершин и вероятностей переходов на его дугах. Расчеты значений весов вершин и вероятностей переходов.

4.5. Зарегистрированные результаты вычислений на ЭВМ и их интерпретация.

4.6. Расчеты основных параметров рабочей нагрузки, создаваемой в ЭВМ исследуемой программой, при параметрах процессора и других устройств, взятых из справочников [5,6].

4.7. Список использованных источников.

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

5.1. Что такое рабочая нагрузка вычислительной системы?

5.2. Основные параметры рабочей нагрузки?

5.3. Табличный способ описания рабочей нагрузки многомерного потока входных заданий ВС?

5.4. Как увеличить объем вычислительной работы (загрузку) системы при неизменных параметрах рабочей нагрузки входных заданий?

5.5. Что такое классификация рабочей нагрузки? Зачем она применяется? Как вводится случайный разброс параметров рабочей нагрузки после ее классификации?

5.6. От каких характеристик вычислительной сложности алгоритма и как зависят параметры рабочей нагрузки?

5.7. Почему функционирование алгоритма представляется случайным дискретным процессом?

5.8. При каких допущениях дискретный случайный процесс считается марковским?

5.9. Как вычисляется вес функциональной вершины оператора графа Маркова?

5.10. Как зависят веса выходных дуг вершины-оператора перехода от свойств предиката? Чему равно значение вероятности перехода по выходным дугам функциональных вершин и вершин ввода-вывода?

5.11. Как имитируется в графе Маркова оператор условного перехода ГСА алгоритма?

5.12. Зачем в систему уравнений, описывающих граф Маркова, введен символ Кронекера?

5.13. Почему в системе уравнений, описывающих граф Маркова, суммирование выполняется только по $(K-2)$-м вершинам, исключая данную i-ю вершину?

5.14. Как, используя результаты марковского моделирования алгоритма программы, вычислить средний объем информации, передаваемой между ОЗУ и ВЗУ за одну транзакцию.

5.15. Как определить размер рабочей области ОЗУ, требуемый для хранения прикладной программы и данных?

Лабораторная работа № 2
ВЫБОР БЫСТРОДЕЙСТВИЯ ПРОЦЕССОРА И ДИСЦИПЛИН ОБСЛУЖИВАНИЯ ВХОДНЫХ ЗАДАНИЙ

1. ЦЕЛЬ РАБОТЫ

Изучение элементов теории систем массового обслуживания (СМО) и методики ее применения для обоснованного выбора быстродействия процессора, видов и уровней приоритетов заданий во входной очереди. Практическое освоение интерактивного режима работы с программно-аналитической моделью центрального ядра вычислительной системы (ВС) при решении прямой и обратной задач анализа ВС.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРИМЕНЕНИЮ ТЕОРИИ СМО ДЛЯ ОЦЕНКИ ХАРАКТЕРИСТИК ПРОИЗВОДИТЕЛЬНОСТИ ВС

Теория систем массового обслуживания (СМО) – основной математический аппарат, применяемый для аналитического описания случайных процессов функционирования ВС [2, 3]. Случайный характер процессов в ВС обусловлен колебаниями интенсивности потока входных заданий, существенными различиями параметров рабочей нагрузки разных прикладных программ, колебаниями вычислительной сложности алгоритмов программ в зависимости от изменения значений исходных данных. На основе простейших математических моделей ВС в виде СМО получен ряд важных аналитических формул, позволяющих приближенно вычислять характеристики ВС как математические ожидания неслучайных характеристик случайных величин. Для анализа поведения ВС при вариации функции распределения в широком диапазоне, названные формулы выведены в теории СМО для трех видов стандартных законов распределения случайных величин:

- экспоненциального

$$p(\tau)=1-e^{-\lambda\tau},$$

$$p(\tau_0)=1-e^{-\mu\tau_0},$$

с плотностью распределения

$$P(\tau)=\lambda e^{-\lambda\tau},$$

$$P(\tau_0)=\mu e^{-\mu\tau},$$

где $\lambda=\dfrac{1}{T}$ – интенсивность потока заданий (заявок); $\mu=\dfrac{1}{v}$ – интенсивность обслуживания (обработки) заявок, τ – случайная величина интервала времени между соседними заявками, $T=MO(\tau)$, τ_0 – случайная величина длительности обслуживания одной заявки, $v=MO(\tau_0)$.

Коэффициент вариации времени обслуживания:

$$\upsilon=\frac{\delta_0}{v},$$

где $\delta_0=\sqrt{D(\tau_0)}$ – СКО случайной величины τ_0.

Для экспоненциального закона распределения $\upsilon=1, \delta_0=v$;

– равномерного

$$p(\tau_0)=\frac{\tau_0-a}{b-a}$$

с плотностью распределения

$$P(\tau_0)=\frac{1}{b-a}=\text{const},$$

коэффициент вариации

$$\upsilon<\frac{1}{\sqrt{3}};$$

– постоянного (детерминированного) для неслучайных величин как частного случая равномерного закона:

$$\tau_0 = v = \text{const},$$
$$a = b = \tau_0 = \text{const},$$
$$\upsilon = \delta_0 = 0.$$

В теории СМО чаще всего применяется показательный экспоненциальный закон распределения как худший из названных выше по вариации $\upsilon = 1$, а равномерный и постоянный – для проверки диапазона изменения характеристик СМО в широком диапазоне функций распределения с вариациями: $\upsilon = 0...1$.

В связи с отсутствием на начальных стадиях системотехнического проектирования ВС достаточно точных значений параметров ВС применяется приближенное математическое моделирование системы в виде трех типов простейших СМО:

– одноочередной одноканалыюй СМО с одномерным потоком входных заявок – ОООС (рис. 2.1);

– одноочередной одноканальной СМО с многомерным потоком входных заявок – ООМС (рис. 2.2);

– одноочередной многоканальной СМО с многомерным потоком входных заявок – ОММС (см. рис. 3.1 в описании лабораторной работы № 3).

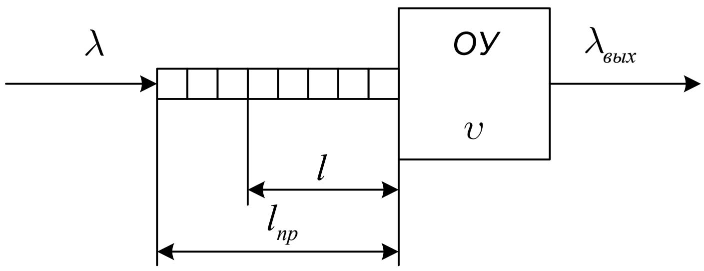

Рис. 2.1. ОООС

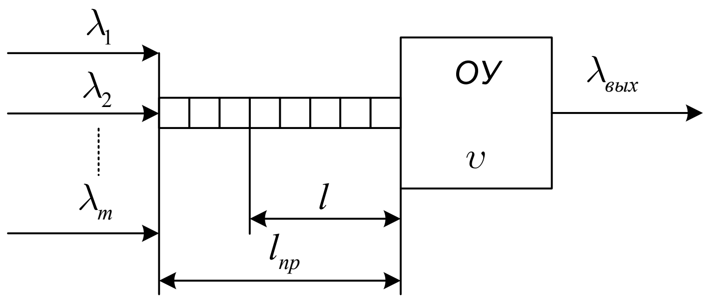

Рис. 2.2. ООМС

2.1. Аналитические формулы теории СМО для модели ОООС (рис. 2.1).

• загрузка обслуживающего устройства (ОУ):

$$\rho = \frac{T_n}{T} = \frac{v}{T} = \lambda v = \frac{\lambda}{\mu},$$

где T_n – полезное время работы ОУ; T – общее время работы системы с учетом простоев;

• коэффициент простоя ОУ:

$$\eta = 1 - \rho,$$

• время ответа, равное времени пребывания заявки в системе.

$$t_{отв} = u = w + v = \frac{v}{1-\rho},$$

где w – среднее значение времени ожидания заявки в очереди;

• средняя длина очереди:

$$l = \lambda w = \frac{\rho^2}{1-\rho},$$

• время ожидания заявки в очереди:

$$w = \rho u = \frac{\rho v}{1-\rho},$$

• среднее число заявок, пребывающих в системе:

$$h = \lambda u = 1 + \rho = \frac{\rho}{1-\rho}.$$

Условие существования стационарного режима работы системы без перегрузки, при котором названные выше характеристики ВС нс зависят от времени:

$$\rho < 1.$$

В этом режиме l и $l_{пр}$ являются конечными величинами, $\lambda_{вых} = \lambda$ (рис. 2.1), вероятность потери заявок $P_n = 0$.

Нестационарный режим наблюдается при перегрузках ВС, когда $\rho > 1$. При этом, если очередь неограничена, когда $l_{пр} = \infty$ $\lambda_{вых} < \lambda$, а при ограниченной очереди, когда $l_{пр}$ – конечная величина, $P_n \neq 0$.

2.2. Аналитические формулы теории СМО для модели ООМС (рис. 2.2.)

• загрузка системы:

$$R = \sum_{i=1}^{m} \rho_i = \sum_{i=1}^{m} \lambda_i v_i,$$

где m – количество разных типов заявок во входном многомерном потоке; λ_i, v_i, ρ_i – интенсивность, время обработки, загрузка ОУ по каждой заявке *i*-го типа в отдельности;

• суммарная интенсивность входного потока заявок:

$$\lambda = \sum_{i=1}^{m} \lambda_i ,$$

• средняя длина очереди:

$$l = \sum_{i=1}^{m} l_i = \sum_{i=1}^{m} \lambda_i w_i,$$

где w_i – время ожидания в общей очереди заявки *i*-го типа;

• среднее число заявок, пребывающих в системе:

$$n = \sum_{i=1}^{m} n_i = \sum_{i=1}^{m} \lambda_i u_i = l + R ,$$

где u_i – время пребывания в системе заявки *i*-го типа;

• средне-взвешенное время обработки одной заявки в ОУ эквивалентной СМО типа ОООС:

$$v_{cp} = \sum_{i=1}^{m} \frac{\lambda_i}{\lambda} v_i = \frac{R}{\lambda},$$

• средне-взвешенное время ожидания одной заявки в очереди эквивалентной СМО типа ОООС:

$$w_{cp} = \sum_{i=1}^{m} \frac{\lambda_i}{\lambda} w_i = \frac{1}{\lambda},$$

• средне-взвешенное время ответа на одну заявку (время ее пребывания в системе) эквивалентной СМО типа ОООС:

$$t_{отв} = u_{cp} = \sum_{i=1}^{m} \frac{\lambda_i}{\lambda} u_i = \frac{n}{\lambda}.$$

Условие существования стационарного режима работы системы:

$$R < 1.$$

Зависимость от $\{\rho_i\}$ и $\{v_i\}$ времени ожидания w_i в очереди по каждому *i*-му типу входных заявок для модели ООМС определяется видом дисциплины обслуживания при выборке их из очереди и постановке на обслуживание:

• для бесприоритетной дисциплины обслуживания (FIFO. LIFO, случайная выборка из очереди) оно одинаково для всех *i*-х типов заявок [3]:

$$w = \frac{\sum_{i=1}^{m} \rho_i v_i \left(1 + \upsilon_i\right)^2}{2(1 - R)},$$

где υ_i – коэффициент вариации случайной длительности обслуживания в ОУ *i*-го типа входной заявки.

Для двух альтернативных законов распределения времени обслуживания заявок: экспоненциального и постоянного время ожидания соответственно равно:

$$w_э = \frac{\sum_{i=1}^{m} \rho_i v_i}{1-R},$$

$$w_п = \frac{\sum_{i=1}^{m} \rho_i v_i}{2(1-R)},$$

• для относительных приоритетов (RP) при выборке из очереди без прерывания обработки находящейся на обслуживании заявки любого уровня приоритета [3]:

$$w_{кэ} = \frac{\sum_{i=1}^{m} \rho_i v_i}{(1-R_{k-1})(1-R_k)}, k = \overline{1, m},$$

где k – уровень приоритета заявки,

$$R_k = \sum_{i=1}^{k} \rho_i \text{ , } \rho_i = \lambda_i v_i,$$

• для абсолютных приоритетов (AP) при выборке из очереди с прерыванием обработки заявки более низкого приоритета, находящейся на обслуживании, при экспоненциальном и постоянном распределении времен обслуживания и дообслуживания прерванных заявок время ожидания заявки k-го приоритета соответственно равно [3]:

$$w_{кэ} = \frac{R_{k-1} v_k}{1-R_{k-1}} + \frac{\sum_{i=1}^{k} \rho_i v_i}{(1-R_{k-1})(1-R_k)},$$

$$w_{кп} = \frac{R_{k-1} v_k}{1-R_{k-1}} + \frac{\sum_{i=1}^{k} \rho_i v_i}{2(1-R_{k-1})(1-R_k)}.$$

По приведенным формулам вычисляются и строятся зависимости:

$$w_k = f(k, B), w_k = f(k, R), k = \overline{1,m},,$$

где B – быстродействие процессора, которое может быть введено в указанные формулы с учетом выражения: $v_i = \dfrac{\theta_i}{B}$, если θ_i – вычислительная сложность алгоритма обработки i-й заявки.

В результате сравнительного анализа названных зависимостей при разных видах и уровнях приоритетов обосновывается выбор вида дисциплины обслуживания и уточняется выбранное быстродействие процессора. Критерием выбора может быть минимизация времен ожидания заявок в очереди при достижении удовлетворительной загрузки процессора $R = 0,7...0,9$.

Обоснование желательного диапазона изменения величины быстродействия B процессора при построении и анализе зависимостей $w_k = f(k, B), k = \overline{1, m}$ выполняется по следующим формулам для оценки величины B, которые составлены методами теории СМО для ВС оперативной обработки информации по критерию минимизации потерь из-за задержек в обслуживании заявок и простоев процессора [3]:

- при неограниченном времени ожидания, когда в явном виде не накладываются названные ограничения:

$$B'_{\min} \geq \sum_{i=1}^{m} \lambda_i \theta_i ,$$

- при наложении явных ограничений на времена ожиданий заявок в очереди $(w_i \leq w_i^*)$:

$$B''_{\min} \geq \frac{1}{2}\sum_{i=1}^{m} \lambda_i \theta_i + \sqrt{\frac{1}{4}\sum_{i=1}^{m} \lambda_i \theta_i \left(\sum_{i=1}^{m} \lambda_i \theta_i + 2\frac{\sum_{i=1}^{m} \lambda_i \theta_i^{(2)}}{\sum_{i=1}^{m} \lambda_i \theta_i w_i^*} \right)},$$

где $\theta_i, \theta_i^{(2)}$ – вычислительная сложность алгоритма обработки *i*-й заявки и ее второй начальный момент, который в общем случае равен $\theta_i^{(2)} = D(\theta_i) + \theta_i^2$, а при экспоненциальном распределении времени обработки $\theta_i^{(2)} = 2\theta_i^2$.

- оптимальное быстродействие для случая неограниченного времени ожидания и бесприоритетной дисциплины обслуживания [3]:

$$B_{opt} = \sum_{i=1}^{m} \lambda_i \theta_i + \frac{1}{2\sum_{i=1}^{m} \lambda_i \theta_i} \left\{ \kappa\lambda \sum_{i=1}^{m} \lambda_i \theta_i^{(2)} + \sqrt{\left(2\left(\sum_{i=1}^{m} \lambda_i \theta_i \right)^2 + \kappa\lambda \sum_{i=1}^{m} \lambda_i \theta_i^{(2)} \right) \kappa\lambda \sum_{i=1}^{m} \lambda_i \theta_i^{(2)}} \right\},$$

где $\lambda = \sum_{i=1}^{m} \lambda_i$ – суммарная интенсивность заявок во входном многомерном потоке, κ – постоянный коэффициент, отражающий уровень требований к степени уменьшения времен ожидания $\{w_i\}$.

Например, при $\kappa = 0$: $B_{opt} = B'_{\min} = \sum_{i=1}^{m} \lambda_i \theta_i$. Чаще всего принимают $\kappa = 0{,}1$.

После подбора подходящей величины быстродействия B одного из стандартных процессоров, вида и уровней приоритетов входных заявок следует проверить полученную степень загрузки процессора, выполнив проверочный ее расчет по формуле для модели ООМС (рис. 2.2):

$$R = \sum_{i=1}^{m} \rho_i = \sum_{i=1}^{m} \lambda_i v_i = \frac{1}{B} \sum_{i=1}^{m} \lambda_i \theta_i. \qquad (2.1)$$

Если полученная загрузка не удовлетворяет условию стационарности: $R < 1$, или она недопустимо мала, например, $R < 0{,}5$, следует изменить величину быстродействия B или значения интенсивностей входных заявок и повторить расчеты

$$w_k = f(k, B), k = \overline{1, m}.$$

Циклы расчетов повторяют до тех пор, пока одновременно не будет удовлетворен названный выше критерий выбора дисциплины обслуживания и достигнута удовлетворительная загрузка процессора (2.1).

При назначении видов и уровней приоритетов следует учитывать, что они предназначаются не только для достижения экономии ресурсов ПС, обеспечиваемой в результате данного анализа, но и для решения следующих вопросов организации вычислительных процессов:

- ускорение решения срочных задач;

- разрешение конфликтов между вычислительными процессами при мультипрограммной обработке.

Следовательно, уровни приоритетов некоторых заданий во входном многомерном потоке не могут быть выбраны произвольно. Названная выше экономия ресурсов обусловлена тем, что относительные и абсолютные приоритеты позволяют гарантировать для наиболее приоритетных заявок достижение конечных малых величин времен их ожидания в очереди даже в периоды, когда ВС работаем с перегрузкой при $R > 1$. Это позволяет выбрать более дешевый процессор с меньшим быстродействием, намерено его перегружая (2.1). По закону сохранения времени ожидая заявок в очереди СМО типа ООМС [3]:

$$\sum_{i=1}^{m} \rho_i w_i = \text{const}$$

названный положительный эффект достигается за счет существенного увеличения времени ожидания наименее приоритетных заявок. Удовлетворительное время ответа по таким заявкам может быть получено либо при значительных колебаниях загрузки, когда они обслуживаются в периоды недогрузки ВС, либо за счет введения динамических приоритетов, позволяющих уменьшить среднее значение времен их ожидания в очереди.

2.3. Методика интерактивного решения на ЭВМ обратной задачи анализа ВС

Рассмотренный математический аппарат может быть применен не только для решения на ЭВМ описанной выше (п.2.2) прямой задачи анализа ВС, завершаемой выбором подходящей величины быстродействия процессора *B*, но и более важной для инженерной практики обратной задачи анализа при заданной величине быстродействия *B* выбранного стандартного процесса. В последнем случае необходимо подобрать допустимые интенсивности поступления входных заданий $\{\lambda_i\}$ и, следовательно, – допустимую пропускную способность ВС:

$$ПС = \lambda = \sum_{i=1}^{m} \lambda_i = ?,$$

а также выбрать дисциплину их обслуживания при известных быстродействии B и вычислительных сложностях алгоритмов входных заданий $\{\theta_i\}$.

Обратная задача анализа может быть решена на ЭВМ в интерактивном режиме обработки методом последовательных приближений путем многократного решения описанной выше (п.2.2) прямой задачи с вариацией ее исходных данных $\{\lambda_i\}, \{w_i\}$.

Для ускорения сходимости процесса последовательных приближений рекомендуется следующий способ выбора начального приближения при заданных B и $\{\theta_i\}$:

- вычислить множество времен обслуживания (обработки процессором) входных заданий:

$$v_i = \frac{\theta_i}{B}, i = \overline{1, m},$$

- принять допущение, что ограничения w_i^* на времена ожидания заявок в очереди не должны превышать времен их обслуживания: $w_i^* \le v_i$, и ввести в ЭВМ в качестве начального приближения:

$$\left(w_i^*\right)_{нач} = v_i, i = \overline{1, m},$$

- учитывая, что в стационарном режиме работы ВС время пребывания i-й заявки в системе равно:

$$u_i = w_i + v_i, i = \overline{1, m},$$

выбрать средние значения интервалов времени между соседними входными заявками, соблюдая неравенства:

$$T_i \ge u_i, i = \overline{1, m},$$

и задать начальное приближение по $\{\lambda_i\}$ в виде:

$$\lambda_i^{нач} = \frac{1}{T_i} \approx \frac{1}{2v_i}, i = \overline{1,m}.$$

В конце каждого *K*-го шага интерактивных вычислений на ЭВМ сравнивается найденное значение $B'_{\min,k}$, соответствующее неограниченному времени ожидания, с заданным исходным значением *B*, и в зависимости от результатов сравнения корректируются величины $\{w_{ik}^*\}$, вводимые в ЭВМ в следующем $(K+1)$-м шаге вычислений.

Если найденное в данном *K*-м шаге

$$B'_{\min,k} > B,$$

то использованные в *K*-м шаге величины $\{w_{ik}^*\}$ недопустимо занижены, и их следует пропорционально увеличить при вводе в ЭВМ в $(K+1)$-м шаге приближений:

$$w_{i,k+1}^* \approx \left(\frac{B'_{\min,k} - B}{B} + 1\right) w_{ik}^*, i = \overline{1,m}.$$

Если наблюдается обратное неравенство:

$$B'_{\min,k} < B,$$

то величины ограничений на времена ожидания следует по аналогии пропорционально уменьшить:

$$w_{i,k+1}^* \approx \left(1 - \frac{B'_{\min,k} - B}{B}\right) w_{ik}^*, i = \overline{1,m}.$$

Следует соответственно изменить и вторую группу исходных данных $\{\lambda_{i,k+1}\}$, вводимых в $(K+1)$-м шаге решения:

$$\lambda_{i,k+1} \approx \frac{1}{v_i + w_{i,k+1}^*}, i = \overline{1,m}.$$

Последовательные шаги интерактивного решения обратной задачи путем решения в каждом *K*-м шаге прямой задачи завершаются, когда очередное найденное значение $B'_{\min,k}$ приблизится к заданному быстродействию B выбранного стандартного процессора: $B'_{\min,k} \approx B$. Этот критерий косвенно соответствует обеспечению достаточно высокой загрузки процессора $R \approx 0{,}7...0{,}9$. В заключение необходимо выполнить поверочный расчет полученной фактической величины загрузки (2.1). а также весовых коэффициентов: $\kappa_i = \dfrac{\lambda_i}{\lambda}, i = \overline{1,m}$,которые необходимы для дальнейших вычислений средневзвешенных временных характеристик ВС по модели типа ООМС.

Трудоемкие многократные расчеты по приведенным выше формулам автоматизируются с помощью программно-аналитической модели "Выбор". Основные задачи пользователя программы "Выбор":

- предварительное определение вычислительных сложностей $\{\theta_i\}$ программ входных заданий ВС с помощью изученной в лаб. раб. №1 программы "Алгоритм";

- выбор величины быстродействия B одного из стандартных процессов по формуле:

$$B \geq \sum_{i=1}^{m} \lambda_i^{ж} \theta_i \,,$$

где $\lambda_i^{ж}$ – такое желательное значение интенсивности i-го типа входного задания, сумма которых обеспечивает требуемую по техническому заданию пропускную способность ВС:

$$\sum_{i=1}^{m} \lambda_i^{ж} \geq ПС \,,$$

при ориентировочных значениях весовых коэффициентов:

$$K_i^{ж} = \frac{\lambda_i^{ж}}{\sum_{j=1}^{m} \lambda_j^{ж}} \approx \frac{\theta_i^{-1}}{\sum_{j=1}^{m} \theta_j^{-1}} \,,$$

- вычисление параметров начального приближения;

- выполнение на ЭВМ интерактивных вычислений методом последовательных приближений с пересчетом в каждом шаге итераций исходных данных:

$$\{\lambda_i\} \text{ и } \{w_i^*\},$$

- проверка подобранных в результате интерактивного расчета на ЭВМ значений $\{\lambda_i\}$ по условию:

$$\sum_{i=1}^{m} \lambda_i^{ж} \geq ПС,$$

и, если это неравенство не выполняется, увеличение исходной величины B и повторение всех расчетов;

- проверочный расчет по формуле (2.1) фактически полученной величины загрузки R процессора и корректировка быстродействия B при неудовлетворительной загрузке;

- путем сравнения кривых $w_k = f(k, B)$, полученных на ЭВМ, выбор подходящей дисциплины обслуживания и уровней приоритетов входных заданий;

- вычисление весовых коэффициентов $\{\kappa_i\}$, требуемых для нахождения средневзвешенных значений характеристик ВС.

3. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

3.1. Запустить программу "Выбор".

3.2. Выбрать величину быстродействия B процессора и значения вычислительной сложности $\{\theta_i\}$ трех входных заданий. Вычислить параметры начального приближения $(w_1^*)_{нач}, (w_2^*)_{нач}, (w_3^*)_{нач}$ и интенсивностей поступления входных заданий $\lambda_1^{нач}, \lambda_2^{нач}, \lambda_3^{нач}$ по приведенным выше (п.2.3) формулам.

3.3. Ввести $\{\lambda_i^{нач}\}, \{\theta_i\}, \{(w_i^*)_{нач}\}$ соответственно в первую, вторую и третью колонки исходных данных.

3.4. Выполнить вычисление на ЭВМ результатов первого приближения и зарегистрировать их. Сравнить найденное значение $B'_{\min,1}$, соответствующее неограниченному времени ожидания, с исходной величиной B. Откорректировать третью и первую колонки в таблице исходных данных, вычисляя их но формулам для $w^{*}_{i,k+1}$ и $\lambda_{i,k+1}$ приведенным выше (п.2.3).

3.5. Выполнить вычисление на ЭВМ результатов второго приближения. Зарегистрировать новое значение $B'_{\min,2}$ и сравнить его с исходной величиной B. Аналогично п.3.4 откорректировать третью и первую колонки в таблице исходных данных.

3.6. Повторять вычисления на ЭВМ с коррекцией исходных данных аналогично п.п.3.4, 3.5 до тех пор, пока очередное новое значение $B'_{\min,k}$ не станет примерно равным B: $B'_{\min,k} \approx B$. Зарегистрировать все результаты вычислений последнего приближения $B'_{\min}$, $B''_{\min}$, B_{opt} и кривые $w_k = f\left(k, B\right)$ для трех дисциплин обслуживания.,

3.7. Выбрать наиболее подходящую дисциплину обслуживания и уровни приоритетов входных заданий согласно рекомендациям п.2.2.

3.8. Выполнить поверочный расчет фактически полученной величины загрузки процессора по формуле (2.1) и значения весовых коэффициентов $\kappa_i = \dfrac{\lambda_i}{\lambda}, i = \overline{1,3}$.

3.9. Предъявить полученные результаты преподавателю.

4. СОДЕРЖАНИЕ ОТЧЕТА

4.1. Краткое описание методики интерактивного решения на ЭВМ обратной задачи анализа ВС но определению пропускной способности ВС и выбору дисциплины обслуживания входных заданий при заданном быстродействии B стандартного процессора.

4.2. Схема СМО типа ООМС.

4.3. Расчет параметров начального приближения.

4.4. Зарегистрированные результаты вычислений на ЭВМ первого приближения.

4.5. Последовательность зарегистрированных значений $B'_{\min,k}$ и расчетов исходных данных прямой задачи $\left\{w^*_{i,k+1}\right\}$ и $\left\{\lambda_{i,k+1}\right\}$ корректируемых в каждом k-м шаге интерактивных вычислений на ЭВМ.

4.6. Зарегистрированные результаты вычислений на ЭВМ последнего приближения: $B'_{\min}$, $B''_{\min}$, B_{opt} и кривые $w_k = f\left(k, B\right)$, $k = \overline{1,3}$ для всех трех дисциплин обслуживания.

4.7. Обоснование выбора вида дисциплины обслуживания и уровней приоритетов входных заданий.

4.8. Расчет фактически полученной величины загрузки и весовых коэффициентов.

4.9. Выводы с указанием достигнутой величины пропускной способности ВС, фактических величин времен ожидания в очереди и времен ответа по каждому типу входных заданий и описанием экономического выигрыша от введения выбранной дисциплины обслуживания.

4.10. Список использованных источников.

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

5.1. Почему аналитическая теория СМО основана на использовании показательного экспоненциального закона распределения случайных величин?

5.2. Чем объяснить случайный характер процессов функционирования ВС?

5.3. Как средняя длина очереди и время ожидания в очереди зависят от загрузки обслуживающего устройства СМО типа ОООС?

5.4. Почему средние значения временных характеристик СМО типа ООМС должны вычисляться как средне-взвешенные?

5.5. Чем отличаются стационарный и нестационарный режим работы ВС?

5.6. В чем состоит экономический выигрыш от применения приоритетных дисциплин обслуживания?

5.7. Назначение приоритетных дисциплин обслуживания заявок в очередях современных ВС?

5.8. В чем состоят критерии выбора дисциплин обслуживания и уровней приоритетов?

5.9. Чем отличаются прямая и обратная задачи анализа ВС?

5.10. Зачем требуется предварительная ориентировочная оценка быстродействия процессора при решении обратной задачи анализа ВС?

5.11-. Закон сохранения времени ожидания?

5.12. Обосновать способ выбора параметров начального приближения при интерактивном решении на ЭВМ обратной задачи анализа ВС?

5.13. Почему с уменьшением быстродействия процессора увеличивается его загрузка?

5.14. Как связаны между собой допустимая интенсивность потока входных заданий и пропускная способность ВС?

5.15. Как вычислить производительность ВС, зная ее пропускную способность?

Библиографический список

1. Ларионов А.М., Майоров С.А., Новиков Г.И. Вычислительные системы, комплексы и сети: Учебн. для вузов. – Л.: Энергоатомиздат, 1987. – 288 с.

2. Альянах И.Н. Моделирование вычислительных систем. – Л.: Машиностроение, 1988. - 223 с.

3. Основы теории вычислительных систем / Под ред. С.А. Майорова: Учебн. пособие для вузов. – М.: Высшая школа, 1978. – 408 с.

4. Дьяконов В.П. Справочник по алгоритмам и программам па языке бейсик для персональных ЭВМ: Справочник. – М.: Наука. Гл. ред. физ.-мат. лит., 1987. –249 с.

5. Заморин А.П., Мячев А.А., Селиванов Ю.П. Вычислительные машины, системы, комплексы: Справочник. – М.: Энергоиздат, 1985. – 264 с.

6. Грубов В.И. и др. Справочник по ЭВМ / Отв. ред. Г.Г. Пухов. – Киев: Наук. думка, 1989. – 544 с.

7. Кузьмин И.В. и др. Синтез вычислительных алгоритмов управления и контроля. – Киев: Техника, 1975. – 248 с.