САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

Соколов Андрей Владимирович

МАТЕМАТИЧЕСКИЕ МОДЕЛИ И АЛГОРИТМЫ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ДИНАМИЧЕСКИМИ СТРУКТУРАМИ ДАННЫХ

05.13.18 — Математическое моделирование,
численные методы и комплексы программ
05.13.17 — Теоретические основы информатики

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
доктора физико-математических наук

Санкт-Петербург 2006

Работа выполнена в Институте прикладных математических исследований Карельского научного центра РАН.

Официальные оппоненты:
доктор физ.-мат. наук, профессор Баранов Сергей Николаевич
доктор физ.-мат. наук, профессор Граничин Олег Николаевич
доктор физ.-мат. наук, профессор Серебряков Владимир Алексеевич

Ведущая организация:
Московский государственный университет им. М.В. Ломоносова.

Защита состоится "26" октября 2006 г. в 14 часов на заседании диссертационного совета Д 212.232.51 по защите диссертаций на соискание ученой степени доктора наук при СПбГУ по адресу: 198504, Санкт–Петербург, Старый Петергоф, Университетский пр., д. 28.

С диссертацией можно ознакомиться в научной библиотеке им. А.М. Горького СПБГУ по адресу: Санкт–Петербург, Университетская набережная., д. 7/9.

Автореферат разослан "15" сентября 2006 г.

Ученый секретарь
диссертационного совета
Д 212.232.51
д.ф.-м.н., профессор Б. К. Мартыненко.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Актуальность темы определяется потребностями индустрии программного и аппаратного обеспечения, особенно для мобильных устройств, встроенных систем, различных сетевых устройств, например, маршрутизаторов, и т. п. Такие устройства имеют жесткие ограничения на ресурсы памяти. Разработка программных систем для них требует особого внимания к алгоритмам управления памятью. В настоящее время достаточно хорошо развита теория страничной виртуальной памяти. Различные модели и алгоритмы оптимального замещения страниц исследовались в работах Ахо А., Деннинга Р, Ульмана Дж., Михновского С.Д., Шора Н.З., Авена О.И., Когана Я.А., Стояна Ю.А., Кутепова В.П., Пьянкова В.П. и многих других.

Однако по нашему мнению, разработанные специально под конкретную структуру данных алгоритмы будут работать лучше, чем универсальные алгоритмы замещения страничной виртуальной или кэш-памяти.

Этот вывод подтвердила и практика разработки, например, стековых компьютеров и RISC процессоров, где для управления стеками в двухуровневой памяти (регистры — оперативная память) использовались специальные алгоритмы, а не универсальные алгоритмы кэш-памяти. Еще один пример - алгоритмы работы с очередями, необходимые при разработке встроенных операционных систем, управляющих потоками пакетов в сетевых маршрутизаторах, таких, например, как Cisco IOS, где требования на время обработки пакетов маршрутизатором очень жесткие. Механизм страничной виртуальной памяти здесь не используется и вся работа происходит в нескольких пулах оперативной памяти.

Кроме того на практике достаточно широко используются системы с нестраничной организацией памяти. Теория нестраничной организации памяти развита недостаточно. Выводы большинства работ [Кнут Д., Грисс Д., Танненбаум Э., Цикритзис Д., Бернстайн Ф., Bays C.A., Fenton I.S, Paim P.W., Campbell I.A., Shore J. и др.] основаны на имитационном моделировании и часто противоречивы.

В связи с этим актуальной является задача разработки математического аппарата для анализа методов представления в памяти динами-

ческих структур данных.

Цель исследования. Целью диссертации является разработка и анализ новых математических моделей некоторых базовых структур данных, и разработка на основе этих моделей алгоритмов оптимального управления этими структурами данных.

Объекты исследования. Исследуются базовые линейные структуры — стеки, очереди, динамические таблицы, а также список свободных блоков переменной длины. В данной работе в задачах оптимального управления последовательными стеками в двухуровневой памяти мы будем иметь в виду чистые стеки, в которых разрешен доступ только к последнему включенному в стек элементу, а в задачах оптимального управления стеками в одноуровневой памяти доступ внутрь стека разрешен. Это замечание справедливо и по отношению к последовательным очередям.

В случае связанного представления стеков и очередей в памяти одного уровня мы будем иметь ввиду чистые стеки и очереди.

Практическая ценность. Результаты, выводы и рекомендации диссертации могут быть использованы разработчиками программных и аппаратных систем, в которых используются исследуемые структуры данных.

Стеки наиболее широко используются в задачах синтаксического анализа и трансляции, при организации обращений к подпрограммам. Многостековая иерархическая структура применяется для организации параллельных процессов и мультипрограммирования. Стеки также используются при разработке систем реального времени, алгоритмов комбинаторного анализа, дискретной оптимизации, и вообще везде, где процессы имеют рекурсивный характер.

Анализ ситуации переполнения стека также может быть полезен в связи с проблемой информационной безопасности компьютерных сетей, так как по некоторым данным, проблема переполнения стека (обычно, когда буфер переменной длины затирает адрес возврата, находящийся в вершине стека) оказывается источником ошибок, например, в программах под Unix и Windows примерно в 2/3 случаев.

Задачи управления стеками возникают при разработке RISC-процессоров. В этом случае для работы со скалярными аргументами и локальными переменными процедур организуются перекрывающиеся окна ре-

гистров, которые связаны в кольцо. По существу, в этом случае мы имеем один обычный стек с элементами типа "окно" в главной памяти, несколько верхних элементов которого образуют циклический стек на регистрах. Если вершина регистрового стека совпала с указателем начала с той или другой стороны, т. е. произошло переполнение (или опустошение) вершины стека, то необходимо какое-то число окон переписать в главную память из регистров или наоборот.

Очереди используются в алгоритмах машинной графики, поиска на графах, в качестве буфера ввода/вывода, в операционных системах, компьютерных сетях и других приложениях.

Алгоритмы работы со списком свободных блоков переменной длины используются в трансляторах, операционных системах, в системах моделирования.

Методы исследования. Для решения поставленных задач используется аппарат управляемых случайных блужданий, цепей Маркова, динамического программирования, разностных уравнений и комбинаторики, имитационное моделирование, численные эксперименты с разработанными программами.

Научная новизна. Все предложенные модели и алгоритмы являются новыми.

Результаты диссертации, выносимые на защиту. На защиту выносятся:

1. Математические модели и алгоритмы оптимального управления одним стеком в памяти, состоящей из двух уровней.

2. Математические модели, описывающие поведение двух стеков, растущих навстречу друг другу, в случае последовательной и параллельной организации операций работы со стеками. Алгоритмы оптимального управления двумя стеками, растущими навстречу друг другу, в двухуровневой памяти.

3. Математические модели и алгоритмы оптимального управления тремя последовательными стеками в памяти одного уровня, в случае, если два стека растут навстречу друг другу, а третий расположен отдельно, в случае, если два стека растут навстречу друг

другу с концов памяти, а третий стек растет навстречу им обоим одновременно, начиная с некоторой средней точки, а также в случае связанного и страничного представления трех стеков.

4. Математическая модель и алгоритм оптимального управления тремя стеками в двухуровневой памяти, в случае, если два стека растут навстречу друг другу, а третий расположен отдельно.

5. Математическая модель и алгоритм оптимального управления одной FIFO-очередью в двухуровневой памяти.

6. Математические модели и алгоритмы оптимального управления двумя последовательными циклическими FIFO-очередями в памяти одного уровня, когда в качестве критерия оптимальности выбрано среднее время до переполнения памяти, а также если целью оптимизации является минимизация доли потерянных элементов(пакетов) на бесконечном времени.

7. Математические модели, предназначенные для анализа связанного и страничного методов управления двумя FIFO-очередями в памяти одного уровня.

8. Новый метод управления несколькими последовательными FIFO-очередями, когда очереди двигаются по кругу друг за другом. Анализ данного метода с помощью разработанной имитационной программы.

9. Оценка минимального размера памяти, необходимого для динамического распределения сегментов разных длин.

10. Математические модели и алгоритмы, предназначенные для оценки эффективности стратегий обслуживания списка свободных блоков произвольной длины.

Связь работы с научными программами, темами. Часть результатов диссертации была получена в рамках исследований, выполнявшихся в ходе работы над госбюджетными темами Института прикладных математических исследований Карельского научного центра РАН. Некоторые результаты данной работы вошли в Основные результаты

в области естественных, технических, гуманитарных и общественных наук РАН за 2004 год.

Работа над данной темой была поддержана инициативными грантами РФФИ(№95-01-00800,№01-01-00113, №06-01-00303), грантом РФФИ на издание монографии "Математические модели и алгоритмы оптимального управления динамическими структурами данных" (№01-01-14013).

Апробация результатов диссертации. Результаты диссертации докладывались на семинарах кафедры "Автоматизации сложных систем" и научных конференциях факультета прикладной математики-процессов управления Ленинградского университета (Ленинград, 1974), на семинарах кафедры информатики и математического обеспечения Петрозаводского университета, на ученом совете Института прикладных математических исследований Карельского Научного Центра РАН, на научном семинаре по автоматизации программирования на факультете ВМК МГУ под руководством М.Р. Шура-Бура, на Всесоюзном симпозиуме "Проблемы системотехники" (Ленинград, 1977 г.), на конференции "Автоматизация производства и проектирования РЭА" (Харьков, 1979 г.), на Международной конференции "Развитие и применение открытых систем" (Петрозаводск, 1995 г.), на Семинаре "Неделя Финской Информатики" (Петрозаводск,1999 г.), на III Московской Международной Конференции по Исследованию Операций (Москва, 2001 г.). на I, II Всесоюзной и IV,V,VI Международной Петрозаводской научной конференции "Вероятностные методы в дискретной математике", на Пятом Всемирном Конгрессе по математическому моделированию (Дубна, 2002 г.), на III Всероссийском симпозиуме по прикладной и промышленной математике (Сочи, 2002 г.), на XIII Международной конференции "Проблемы теоретической кибернетики" (Казань, 2002 г.), на V международной конференции "Дискретные модели в теории управляющих систем" (Ратмино, 2003 г.), на Первой Всероссийской научной конференции "Методы и средства обработки информации" (Москва, 2003 г.), на IV Всероссийском симпозиуме по прикладной и промышленной математике (Летняя сессия. Петрозаводск, 2003 г.), (Осенняя сессия. Сочи, 2003 г.), на Восьмом международном семинаре "Дискретная математика и ее приложения" (Москва, 2004 г.), на Второй Всероссийской научной конференции "Методы и средства обработки

информации" (Москва, 2005 г.), а также были приняты к публикации на 13th International Symposium Fundamental of Computation Theory (Riga, FCT 2001, August 22-24), на Second Workshop on Intermediate Reprezentation Engineering for Virtual Machins and Inaugural Conference on the Principles and Practice of Progrmming in Java (Dublin, PPPJ 2002/IRE 2002, June 13-14), на XIV Международной конференции "Проблемы теоретической кибернетики" (Пенза 29 мая- 3 июня 2005 г.).

Публикация результатов. По теме диссертации опубликовано 19 статей, 19 тезисов докладов на международных, всероссийских и региональных конференциях, одна монография, раздел в коллективной монографии, одно учебное пособие.

Структура и объем диссертации. Диссертация состоит из введения, четырех глав, приложения и списка литературы. Общий объем диссертации составляет 301 страницу. Список литературы содержит 118 наименований.

СОДЕРЖАНИЕ РАБОТЫ

Во введении приводится обоснование актуальности темы диссертации, формулировка цели работы, основные научные положения, выносимые на защиту, и описание структуры диссертации.

Первая глава носит вспомогательный характер. В ней представлены последовательные и связанные методы реализации таких структур данных, как стек, очередь, список свободной памяти, бинарное и произвольное дерево, описан ряд алгоритмов из различных предметных областей, в которых используются динамические структуры данных, и предложен ряд новых структур данных, например, списки с 1,5 связями, а также последовательные циклические очереди, которые двигаются по кругу одна за другой.

Также в этой главе дан обзор известных методов реализации систем с виртуальной памятью, включая страничную, сегментную и сегментно-страничную организации виртуальной памяти, и описаны методы управления регистровой вершиной стека в стековых компьютерах и RISC-процессорах.

Во второй главе рассмотрены задачи оптимального управление стеками.

В первом параграфе рассмотрена задача оптимального управления одним стеком в двухуровневой памяти.

Пусть мы работаем со стеком, значительно превосходящим выделенный для него участок быстрой памяти размера m. В этом случае в быстрой памяти всегда хранится вершина стека, а остальная часть находится в памяти второго уровня. При переполнении и опустошении вершины стека мы производим перераспределение верхних элементов стека так, что в быстрой памяти остается x_0 элементов стека.

После перераспределения памяти начинается следующий этап работы. Нашей целью является нахождение оптимального значения x_0.

В качестве математической модели процесса рассмотрим случайное блуждание на состояниях -1, 0, 1,..., x,..., $m-1$, m, $m+1$, где x — число верхних элементов стека в быстрой памяти. Будем считать, что вероятность перехода из состояния x в $x+1$ (увеличение длины стека на единицу в результате включения нового элемента) равна p, а из x в $x-1$ (уменьшение длины стека на единицу в результате исключения одного элемента) равна $q = 1-p$. Блуждание начинается в состоянии x_0, а перераспределение происходит в состояниях -1 и $m+1$.

Если не учитывать стоимость обменов, то нашей целью является минимизация среднего числа обращений к памяти второго уровня. Следовательно, необходимо найти такое x_0, которое максимизирует математическое ожидание времени работы до перераспределения памяти при условии, что процесс начинается в этом состоянии.

Показано, что формула для оптимального значения вершины стека в этом случае имеет следующий вид:

$$x_0 = \begin{cases} \dfrac{m}{2}, & p = q = \dfrac{1}{2}, \\ \log_{\frac{q}{p}} \dfrac{((q/p)^{m+2}-1)p}{(m+2)q\ln(q/p)}, & p \neq q. \end{cases} \qquad (3)$$

Приведены некоторые результаты экспериментов с построенной моделью.

Во втором пункте параграфа рассмотрена задача определения оптимального значения x_0 с учетом затрат на перераспределение стека. Для $p = q = 1/2$ получены формулы для оптимального значения x_0 для нескольких вариантов стековых архитектур.

В параграфе 2 рассмотрена задача оптимального управления вершиной стека для немарковской модели поведения стека.

Пусть теперь параметры p_1 и p_2 определяют вероятность возвращения в то состояние, откуда мы перешли в заданное, в зависимости от того, исключение или включение элемента произошло.

В зависимости от того отслеживаем мы или нет, что произошло переполнение или опустошение вершины стека, мы будем иметь разные начальные вероятности переходов.

Для решения задачи перейдем от немарковской модели к цепи Маркова путем удвоения числа состояний. Будем рассматривать в качестве состояний не точки, а переходы из одной точки в другую. Каждому состоянию i поставим в соответствие пару состояний $(i, i+1)$ и $(i+1, i)$. Тогда состояния будут нумероваться так:

$(0,1)$ $(1,0)$ $(1,2)$ $(2,1)$ ... $(i, i{+}1)$ $(i{+}1, 1)$ ... $(m{-}1, m)$ $(m, m{-}1)$. Поглощающие состояния будут $(0, -1)$ и $(m, m+1)$. Начальное состояние будет обозначаться только номером этого состояния $0 \leqslant i \leqslant m$.

Сначала рассмотрена задача максимизации среднего времени работы до перераспределения памяти.

Для решения задачи строится матрица переходных вероятностей (матрица Q) из невозвратных состояний в невозвратные для новой цепи Маркова порядка $2m+1$ (так как у нас $2m$ переходов плюс начальное состояние).

Теорема 1 определяет вид матрицы Q.

Из экономии места вид этой и всех последующих матриц мы приводить не будем.

Для решения задачи требуется найти фундаментальную матрицу $N = (I - Q)^{-1}$. где I — единичная матрица.

Вторая задача состоит в определении такого значения x_0, чтобы среднее время доступа к вершине стека было минимальным.

Нам будет нужна матрица переходных вероятностей из невозвратных состояний в поглощающие — R размерности - $(2m+1, 2)$. Вид матрицы устанавливает Теорема 2.

Для решения данных задач были разработаны алгоритмы и программы, которые генерирует нужные матрицы и определяют оптимальное состояние.

В параграфе 3 приводится частичное решение одной задачи, поставленной Д. Кнутом.

Пусть в памяти объема m расположены два стека, растущие навстречу друг другу. В этом случае место их встречи можно рассматривать как случайную величину.

Пусть известно, что на каждом шаге с вероятностью p может произойти исключение элемента из одного из стеков и с вероятностью $1-p$ может произойти включение информации в один из стеков. Обозначим через $M(m,p)$ математическое ожидание случайной величины $max(k_1,k_2)$, где k_1 и k_2 — длины стеков при встрече. Д. Кнут поставил задачу разработать математическую модель этого процесса и установить вид функции $M(m,p)$. В (Соколов 1980, Yao 1981, Flajolet 1986, Maier 1991, Louchard 1990, 1994) была построена математическая модель процесса в виде двумерного случайного блуждания в треугольной области с двумя отражающими экранами и одним поглощающим экраном. В ряде работ исследовалось ассимптотическое поведение размеров стеков при переполнении и времени работы до переполнения. В нашей работе предложен алгоритм вычисления $M(m,p)$ для конечных значений m.

Обозначим через x_1 текущую длину первого стека, а через x_2 — второго. Тогда в качестве математической модели процесса мы имеем случайное блуждание по целочисленной решетке в области $x_1 \geqslant 0, x_2 \geqslant 0$, $x_1 + x_2 < m$ с вероятностями $p/2$ сдвинуться влево и вниз, $(1-p)/2$ — вправо и вверх. На границах $x_2 = 0$ и $x_1 = 0$ с вероятностью $p/2$, а в точке $x_1 = x_2 = 0$ с вероятностью p частица остается на месте. Для того чтобы исследовать $M(m,p)$, нужно получить распределение вероятностей поглощения на прямой $x_1 + x_2 = m$, если частица выходит из точки $x_1 = x_2 = 0$.

Пронумеруем состояния снизу вверх и справа налево, начиная с точки $(m,0)$. Теперь цепь имеет $m+1$ поглощающее состояние и $m(m+1)/2$ невозвратных состояний. Для этой цепи справедлива Теорема 3, определяющая вид матрицы переходных вероятностей данной цепи.

Матрица $B[1..m(m+1)/2, 1..m+1]$, определяемая формулой $B = (I-Q)^{-1}R$, дает вероятности поглощения блуждания. Отсюда можно вычислить и искомую величину $M(m,p)$.

Была разработана программа, которая для входных значений p и m

генерирует матрицы R и Q и вычисляет искомое значение $M(m,p)$.

В параграфе 4 рассмотрена задача оптимального управления двумя стеками, растущими навстречу друг другу в двухуровневой памяти.

В первом пункте решается задача минимизации среднего числа перераспределений стеков.

Пусть в памяти объема m расположены два стека, растущие навстречу друг другу и превосходящие объем быстрой памяти. В этом случае в быстрой памяти хранятся только вершины обоих стеков, а остальные части — в памяти второго уровня.

Если же вершина одного из стеков стала пустой или стеки заполнили всю быструю память, происходит перераспределение стеков и обмен с памятью второго уровня, так, что мы приходим к некоторому заданному состоянию памяти, после чего начинается следующий этап работы.

Пусть q_1, p_1 — вероятности исключения и включения в первый стек, q_2, p_2 — вероятности исключения и включения во второй стек, где $q_1 + p_1 + q_2 + p_2 = 1$.

Обозначим через x_1 текущую длину первого стека, а через x_2 — второго. Тогда в качестве математической модели процесса имеем случайное блуждание по целочисленной решетке в области $x_1 \geqslant 0$, $x_2 \geqslant 0, x_1 + x_2 \leqslant m$ с вероятностями из точки с координатами (x_1, x_2): q_1 — сдвинуться влево, p_1 — вправо, q_2 — вниз, p_2 — вверх. Цель — определить такое состояние $S_0 = (x_1^{(0)}, x_2^{(0)})$, при котором среднее время блуждания до поглощения на прямой $x_1 + x_2 = m + 1$ и на прямых $x_1 = -1$ и $x_2 = -1$ было максимальным при условии, что процесс начинается в состоянии S_0.

Пронумеруем поглощающие состояния по границе треугольника против часовой стрелки, начиная с точки $(m+1, 0)$, а невозвратные состояния — снизу вверх и справа налево, начиная с точки $(m, 0)$. В этом случае цепь имеет $(m+1)(m+2)/2$ невозвратных состояний и $3m+4$ поглощающих состояний.

Вид матрицы Q устанавливает Теорема 4.

Во втором пункте рассмотрена задача минимизации среднего времени доступа с учетом потерь на перераспределения стеков, если известны затраты на все разрешенные операции.

Наша цель теперь состоит в том, чтобы определить такое состояние $S_0 = (x_1^0, x_2^0)$, при котором средние затраты во время блуждания и

перераспределений памяти были бы минимальными при условии, что процесс начинается в состоянии S_0.

Для решения задачи потребуется матрица Q, вид которой установлен в предыдущей задаче и матрица R, вид которой устанавливает Теорема 5.

В третьем пункте рассмотрена задача оптимального управления двумя параллельными стеками.

Предполагается, что вершины двух стеков растут навстречу друг другу в быстрой памяти, к которой разрешен доступ нескольких параллельных процессоров. Параллельное выполнение операций подразумевает возможность одновременной работы с двумя стеками. Предполагаются заданными вероятности операций над стеками.

Ставится задача: в какое состояние, в зависимости от заданных вероятностей включения и исключения элементов в стеки, следует переходить после обращения к памяти второго уровня, чтобы среднее время работы до следующего перераспределения памяти было максимально, т. е. чтобы минимизировать среднее число обменов между уровнями.

Геометрически блуждание по целочисленной решетке можно теперь представить как сдвиги вдоль осей координат и по диагоналям. Наша задача состоит в том, чтобы определить такое начальное состояние $s_0 = (x_1^0, x_2^0)$, чтобы среднее время блуждания до поглощения на прямых $x_1 = -1, x_2 = -1$ и ломаной, которая задается уравнениями $x_1 + x_2 = m + 1$ и $x_1 + x_2 = m + 2$, было максимально.

Если пронумеровать сначала поглощающие состояния из точки $(m, 0)$ (против часовой стрелки), а затем невозвратные (снизу вверх, справа налево), то получим конечную однородную поглощающую цепь Маркова, количество невозвратных состояний в которой будет равно $\frac{(m+1)(m+2)}{2}$.

Доказана Теорема 6, определяющая структуру необходимой матрицы Q.

Для решения данных задач были разработаны алгоритмы и программы, определяющие оптимальное состояние (x_1^0, x_2^0).

В параграфе 5 рассмотрена задача оптимального расположения n стеков в памяти одного уровня.

Пусть в памяти объема m единиц требуется расположить n стеков, вероятностные характеристики которых следующие: p_i — вероятность

включения в i-й стек, q_i — вероятность исключения из i-го стека. Процесс начинается, когда стеки пустые. Исключение из пустого стека с вероятностью q_i означает остановку на месте.

Для $n \geqslant 2$ задача оптимального расположения стеков формулируется следующим образом: надо произвести выделение каждому i-му раздельному стеку по m_i единиц памяти и, возможно, какие-то стеки разбить на пары стеков, растущих навстречу друг другу, выделив каждой i-й паре по s_i единиц памяти, так что сумма s_i и m_i по всем раздельным стекам и по парам будет равна m.

Имеет место Теорема 7. Для любого $n \geqslant 2$ встречное расположение n стеков в виде пар стеков, растущих навстречу друг другу, лучше раздельного, если в качестве критерия оптимальности выбрано среднее время до переполнения памяти.

В параграфе 6 рассмотрены задачи оптимального распределения n стеков в памяти одного уровня.

Теперь в силу доказанного ранее мы рассматриваем n стеков, разбитых на пары растущих навстречу друг другу стеков (нечетный стек, если он есть, предполагается растущим навстречу пустому стеку). Ставится задача определить оптимальное начальное разбиение памяти, где оптимальность понимается как максимизация среднего времени работы со стеками до переполнения.

В данной работе рассмотрен случай n=3.

Пусть в памяти объема m единиц расположены три стека. Двум стекам, растущим навстречу друг другу, отведено s единиц памяти, а третьему стеку $m - s$ единиц. Обозначим через x_1, x_2, x_3 текущие длины стеков. В этом случае в качестве модели имеем трехмерное случайное блуждание в треугольной призме с тремя отражающими экранами $x_1 = -1, x_2 = -1, x_3 = -1$ и двумя поглощающими экранами $x_1 + x_2 = s + 1, x_3 = m - s + 1$.

Наша задача состоит в определении значения s и в определении стека, размещаемого отдельно, так, чтобы максимизировать среднее время блуждания внутри призмы до поглощения на ее границе при условии, что процесс блуждания начинается в начале координат.

Сначала рассмотрим задачу оптимального управления тремя стеками, допускающими только включения ($q_i = 0$). Введем новые обозначения $x = x_1 + x_2, y = x_3, p = p_1 + p_2, q = p_3$.

В качестве модели мы будем иметь случайное блуждание внутри прямоугольника $x \geqslant 0, y \geqslant 0,\ x < s+1,\ y < m-s+1$ с вероятностями переходов: из точки (x,y) в точку $(x+1,y)$ — p; из точки (x,y) в точку $(x,y+1)$ — $q = 1-p$. Процесс выходит из начала координат и поглощается на прямых $x = s+1$ и $y = m-s+1$. Нашей задачей является нахождение такого значения s и определение того, какое из p_i принять за q (какой стек расположить отдельно), чтобы среднее время блуждания до поглощения было максимальным.

Первый возможный подход заключается в непосредственном вычислении вероятностей времени работы до переполнения памяти. Можно показать, что среднее время блуждания до переполнения будет вычисляться по следующей формуле

$$E(s,q) = p^{s+1} \sum_{k=0}^{m-s} (s+1+k) C_{s+k}^{k} q^k + q^{m-s+1} \sum_{k=0}^{s} (m-s+1+k) C_{m-s+k}^{k} p^k.$$

Теперь нашу задачу можно сформулировать следующим образом: найти $0 \leqslant s \leqslant m$, и $q = p_i, p = 1-q$, доставляющие максимум функции $E(s,q)$. Аналитическое решение задачи оказалось затруднительно, так как нет явного вида функции $E(s,q)$.

Рассмотрим другой способ решения таких задач — метод разностных уравнений. Обозначим через $T(x,y)$ среднее время блуждания до поглощения, если начальной точкой блуждания была точка (x,y). Тогда

$$T(x,y) = pT(x+1,y) + qT(x,y+1) + 1,$$

так как в этом случае с вероятностью p мы переходим в состояние $x+1, y$, и тогда среднее время будет равно $T(x+1,y)$, с вероятностью q переходим в состояние $x, y+1$, и среднее время будет равно $T(x,y+1)$. При этом пройдет один шаг до переполнения памяти.

Попав на границы области блуждания, мы уже не двинемся ни на один шаг и, следовательно,

$$T(s+1,y) = 0, T(x,m-s+1) = 0.$$

Нашей задачей является нахождение такого значения s и выбор стека, который будет расположен отдельно, чтобы $T(0,0)$ было максималь-

ным. Были предложены алгоритмы и программы, решающие данную задачу обоими рассмотренными методами.

Результаты экспериментов показали, что оптимальные значения s не всегда совпадают с интуитивно ожидаемым $s = mp$. Например, для $m = 1000$ при $p_1 = p_2 = p_3 = 1/3$, $s = 661$, а не $s = 2/3m = 667$, а для $p_1 = 0.6, p_2 = 0.3, p_3 = 0.1$, $s = 889$, а не $s = 0.9m = 900$, причем в оптимальном варианте $T = 983.15$, а для $s = 900$, $T = 963.3$. Это значит, что если действовать по интуиции, а не по теории, то в среднем в данной ситуации мы будем производить на 20 операций включения в стеки меньше до переполнения памяти.

Задача оптимального управления четырьмя стеками, допускающими только включения сводится к рассмотренной задаче.

Далее рассмотрен общий случай управления тремя стеками.

Определен алгоритм нумерации невозвратных состояний и доказана Теорема 8, определяющая вид подматрицы Q матрицы переходных вероятностей данной цепи, соответствующая переходам из невозвратных состояний в невозвратные.

Была написана программа, решающая данную задачу.

Проведенные эксперименты позволяют сделать некоторые полезные для практики выводы. Когда один из стеков имеет нулевую вероятность исключения (т.е. является динамической таблицей), а остальные операции равновероятны, то лучше объединить эту таблицу с одним из стеков и выделить этой паре достаточно большой размер памяти. Например, в трансляторах в случае использования двух стеков и кучи надо кучу направить навстречу одному из стеков, а другой стек разместить отдельно,т.к. куча часто реализуется так, что ее верхний край может только возрастать. Если же мы работаем с двумя таблицами и стеком, то стек надо объединить с одной из таблиц и дать данной паре несколько больше половины памяти.

В следующем пункте рассмотрен другой способ работы с тремя (и более) стеками в памяти одного уровня. Здесь два стека мы направляем навстречу друг другу с концов участка памяти, а третий располагаем где-то в середине участка и включаем в него элементы попеременно с разных сторон. Операции push и pop для внутреннего стека здесь будут реализовываться несколько сложнее, так как нужно будет вычислять слева или справа от начала находится указатель на вершину стека.

Похожий метод применялся в реализации алгола-68, когда стек и куча росли навстречу друг другу, а между ними плавал пузырь, в котором размещались объекты, которые освобождались проще чем элементы кучи.

Пусть есть m единиц памяти, представим три стека, как четыре, т.е. память делится на две части k и $m-k$ элементов. Один из трех стеков начинает расти из точки k: при четном включении на единицу влево, при нечетном включении на единицу вправо. Два других стека растут навстречу этому стеку, из противоположных концов памяти. Задача состоит в том, чтобы определить, какой стек расположить по центру и из какой точки он должен начать движение, чтобы среднее время до переполнения было максимальным.

Можно рассмотреть 4 стека таких, что p_1 – вероятность включения в первый стек, $\frac{p_2}{2}$ – вероятность включения во второй стек, $\frac{p_2}{2}$ – вероятность включения в третий стек, p_3 - вероятность включения в четвертый стек, q_1 - вероятность исключения из первого стека, $\frac{q_2}{2}$ - вероятность исключения из второго стека, $\frac{q_2}{2}$ - вероятность исключения из третьего стека и q_3 - вероятность исключения из четвертого стека. Обозначим через x_1, x_2, x_3, x_4 текущие длины стеков. В качестве математической модели имеем четырехмерное блуждание в области $x_1 + x_2 + x_3 + x_4 \leqslant m, x_1 + x_2 \leqslant k, x_3 + x_4 \leqslant m - k$.

Можно представить это блуждание в виде цепи Маркова, если принять за состояния четверки чисел (x_1, x_2, x_3, x_4), где x_i - длина i-го стека, $x_1 + x_2 \leqslant k$, $x_3 + x_4 \leqslant m - k$. Если есть m единиц памяти и память поделена на две части: k и $m-k$, то количество невозвратных состояний данной цепи будет равно $\frac{(m-k+1)(m-k+2)(k+1)(k+2)}{4}$.

Предложен алгоритм нумерации состояний, который позволил построить матрицу Q переходных вероятностей соответствующей цепи Маркова, вид которой определяет Теорема 9.

Были разработаны алгоритм и программа, решающие задачу.

В следующем пункте строится математическая модель связанного метода представления трех стеков. Пусть мы работаем в памяти размера m единиц с тремя стеками, представленными в виде связанных списков.

В качестве математической модели имеем случайное блуждание в пирамиде: $x_1 + x_2 + x_3 \leqslant \frac{m}{2}$, т.к. половина памяти тратится на связи.

Пронумеруем состояния: начиная с плоскости $x_3 = 0$, $x_1+x_2+x_3 \leqslant \frac{m}{2}$ и т.д. наверх, в каждой из этих плоскостей нумерация состояний начинается вдоль прямой $x_1 + x_2 = \frac{m}{2} - x_3$ от оси x_1 к оси x_2 и т.д.

Теперь можно построить матрицу Q переходных вероятностей однородной поглощающей цепи Маркова. Для этой матрицы справедлива Теорема 10.

Была написана программа, которая реализует алгоритм построения матриц и вычисления среднего времени работы до переполнения памяти.

Рассматривался также случай, когда информационная часть имеет произвольный размер l, поле связи имеет единичную длину, размер памяти m кратен $l + 1$.

Последний рассмотренный в данном параграфе способ представления трех стеков - это страничный способ представления, когда память разделена на страницы одинакового размера. На связь тратится один элемент каждой страницы, поэтому потери на связи в случае страничного представления стеков будут меньше, чем в случае связанного представления.

Мы предполагаем, что существует механизм работы со списком свободных страниц, который позволяет при заполнении страницы предоставить стеку новую страницу. Если же вершина стека освободила страницу, то эта страница возвращается в список свободных страниц.

Получена формула для оптимального размера страницы, при котором среднее количество единиц памяти, которое тратится непосредственно под данные будет максимально, и найдены максимальная и минимальная оценки для среднего времени работы до переполения памяти.

Коротко просуммируем результаты экспериментов с моделями разных методов представления трех стеков в памяти одного уровня.

Среднее время до переполнения в случае последовательного оптимального представления, для большинства наборов вероятностей, лучше, чем среднее время во всех остальных способах представления. Однако, существуют некоторые случаи, когда максимальное среднее время достигается при представлении трех стеков, как четырех.

Для практики может быть интересен случай, когда память делится не оптимально, а поровну между стеками, т.е. в случае последователь-

ного представления $s \approx \frac{2m}{3}$, а в случае последовательного представления трех стеков, как четырех, $k \approx \frac{m}{2}$, Так логично поступать, когда вероятностные законы, описывающие поведение стеков неизвестны. В этом случае также оказалось, что среднее время в случае последовательного представления не всегда максимально, а иногда будет лучше способ представления трех стеков, как четырех.

Если же длина информационной части больше, чем длина поля связи, то при достаточно большом размере информационной части, связанный способ становится лучше, с точки зрения максимизации среднего времени, чем последовательный.

В параграфе 7 рассмотрена задача оптимального управления тремя стеками в двухуровневой памяти.

Пусть два стека расположены навстречу друг другу, и им выделяется память объема m_1 условных единиц, третий стек расположен отдельно, ему выделяется $m_2 = m - m_1$ условных единиц.

Пусть x_1, x_2, x_3 — текущие длины соответственно первого, второго и третьего стеков в быстрой памяти (в условных единицах), где $x_1, x_2, x_3 \geq 0$, $x_1 + x_2 \leq m_1$ и $x_3 \leq m_2$.

Будем предполагать, что при работе с тремя стеками в двухуровневой памяти переполнение или опустошение любого из стеков в быстрой памяти приводят к обмену данными между быстрой и внешней памятью, в результате которого в быстрой памяти остаются стеки длиной x_1^0, x_2^0, x_3^0 условных единиц соответственно, где $x_1^0, x_2^0, x_3^0 \geq 0$, $x_1^0 + x_2^0 \leq m_1$ и $x_3^0 \leq m_2$.

Таким образом, ставится задача определить, в какое состояние следует переходить после обмена данными между двумя уровнями памяти, чтобы среднее время работы со стеками в быстрой памяти до переполнения или опустошения одного из них было максимальным. Иначе говоря, требуется определить m_1 (m_2), x_1^0, x_2^0, x_3^0, при которых среднее время работы со стеками в быстрой памяти до следующего обмена будет максимальным.

При этом поглощающими состояниями являются плоскости $x_1 + x_2 = m_1 + 1$ и $x_3 = m_2 + 1$, и плоскости $x_1 = -1$, $x_2 = -1$ и $x_3 = -1$.

Доказана Теорема 11, определяюшая вид матрицы Q при выбранной нумерации состояний. Разработаны алгоритм и программа, решающая данную задачу.

В третьей главе рассмотрены задачи оптимального управления FIFO-очередями.

В параграфе 1 решается задача оптимального управления одной очередью в двухуровневой памяти.

Пусть мы работаем с одной последовательной циклической очередью, которая переполнила выделенный кусок памяти размера n. Предполагается, что свободных участков быстрой памяти больше нет, но есть память второго уровня, которую мы хотим задействовать для работы с очередью. В случае переполнения в быстрой памяти мы оставляем y_0 начальных элементов очереди, а $n - y_0$ конечных элементов переписываем в память второго уровня.

Нашей задачей будет определение такого значения y_0, чтобы среднее время работы между перераспределениями очереди было максимальным.

Обозначим через y текущее число начальных элементов в быстрой памяти, а через x — текущее число новых элементов, вставленных в конец части очереди, находящейся в быстрой памяти. Пусть p обозначает вероятность включения нового элемента в очередь, q — вероятность исключения элемента из очереди, где $p + q = 1$, $m = n - 1$ — максимальное число элементов в очереди, которое на 1 меньше, чем число свободных мест в быстрой памяти.

Тогда в качестве математической модели мы будем иметь двумерное случайное блуждание в области $x \geqslant 0$, $y \geqslant 0$, $x + y \leqslant m$. Блуждание начинается в точке $y = y_0, x = 0$. В каждый момент времени находясь в точке (x, y), мы можем попасть в точку $(x+1, y)$ с вероятностью p и в точку $(x, y - 1)$ с вероятностью q. Блуждание имеет два поглощающих барьера $x + y = m + 1$ и $y = -1$. Попадание на первый барьер означает переполнение быстрой памяти, а на второй барьер — исчерпание числа начальных элементов в быстрой памяти.

Наша задача формулируется следующим образом: какое значение y_0 следует выбрать, чтобы среднее время блуждания до поглощения было максимальным?

Для вычисления среднего времени блуждания использовался метод построения и численного решения разностного уравнения.

Вычисления показали, что в случае $p = q = 1/2$, $y_0 \neq m/2$, как могло показаться на первый взгляд, а $y_0 \approx 0.7m$, хотя вид точной формулы

для произвольного m неясен.

В параграфе 2 рассмотрена задача оптимального управления двумя последовательными циклическими очередями в случае раздельной реализации.

Пусть в памяти размера m мы должны работать с двумя последовательными циклическими очередями. Известны некоторые вероятностные характеристики очередей. Пусть p_1, q_1 обозначают вероятности включения и исключения элементов в первую очередь, p_2, q_2 – во вторую, а r обозначает вероятность операции, не изменяющей длины очередей (например, только чтение), где $p_1 + q_1 + p_2 + q_2 + r = 1$. Предполагается, что в очередях хранятся данные фиксированного размера. При попытке исключения информации из пустой очереди не происходит завершение работы.

Обозначим через x_1 и x_2 текущие длины очередей, через s максимальный размер первой очереди, через $n = m - 2$ максимальный размер двух очередей. Это значение на две единицы меньше размера памяти, так как в реализации циклической очереди один элемент памяти всегда остается свободным для того, чтобы отличить пустую очередь от переполненной. Иногда для этого предлагается хранить длину очереди, но это эквивалентно потере одной ячейки.

Тогда в качестве модели мы будем иметь двумерное блуждание по целочисленной решетке в области $-1 \leqslant x_1 \leqslant s+1$, $-1 \leqslant x_2 \leqslant n-s+1$ с вероятностями переходов: p_1 – вправо, q_1 – влево, p_2 - вверх, q_2 - вниз, r- остаться на месте. Блуждание начинается в точке $x_1 = x_2 = 0$, прямые $x_1 = -1$ и $x_2 = -1$ – отражающие экраны (попытка исключения из пустой очереди), а прямые $x_1 = s + 1$ и $x_2 = n - s + 1$ – поглощающие экраны (переполнение одной из очередей).

Ставится задача выбрать такое s, чтобы математическое ожидание времени блуждания до поглощения было максимальным.

Пронумеруем точки области блуждания сверху вниз и справа налево, нумерацию начнем с 0. Эти точки будут соответствовать невозвратным состояниям однородной цепи Маркова. Всего таких состояний будет $(s + 1)(n - s + 1)$.

Для решения задачи установлен вид матрицы переходных вероятностей из невозвратных состояний в невозвратные и разработаны алгоритм и программа, которая для заданных значений вероятностей и

размера памяти генерируют нужную матрицу и решает данную задачу.

В параграфе 3 предложена математическая модель связанного представления двух очередей.

Для связанного способа организации очередей достаточно иметь односвязанные списки элементов, упорядоченные от начала очереди к концу. Каждый элемент состоит из двух единиц памяти одинакового размера, одна из которых содержит хранимую информацию, а вторая содержит указатель (связь) на следующий элемент в списке. При таком представлении очередей $\frac{m}{2}$ единиц памяти затрачено на связи, т. е. для хранения данных мы можем использовать только $m - \frac{m}{2} = \frac{m}{2}$ единиц памяти.

Обозначим x_1 и x_2 – текущие длины очередей. В качестве математической модели будем иметь случайное блуждание по целочисленной решетке в треугольной области $x_1 + x_2 \leqslant \frac{m}{2} + 1$, где $x_1 + x_2 = \frac{m}{2} + 1$ - поглощающий экран, попадание на который означает переполнение одной из очередей, $x_1 = -1$ и $x_2 = -1$ - отражающие экраны, попадание на которые означает исключение элемента из пустой очереди.

Блуждание начинается в точке $x_1 = x_2 = 0$. Необходимо найти среднее время блуждания до поглощения при выходе из начального состояния $x_1 = x_2 = 0$.

В параграфе 4 предложена математическая модель страничного представления двух очередей.

В страничном методе организации очереди представляются в виде списка страниц размера x единиц памяти. Мы предполагаем, что существует механизм работы со списком свободных страниц, который позволяет при заполнении страницы предоставить очереди новую страницу. Если же начало очереди освободило страницу, то эта страница возвращается в список свободных страниц. Переполнение наступает тогда, когда список свободных страниц пуст, а какая-то из очередей переполнила страницу.

Заметим, что при $x = 2$ получаем связанное представление очередей, т. е. можно рассматривать связанное представление очередей как частный случай страничного.

Обозначим x_1 и x_2 – текущие длины очередей. При страничном представлении очередей $\frac{m}{x}$ единиц памяти затрачено на связи, т. к. всего $\frac{m}{x}$ страниц и у каждой страницы берем по ячейке. Для хранения

данных у каждой страницы используется $x-1$ единица памяти.

В качестве математической модели будем иметь случайное блуждание по целочисленной решетке в треугольной области

$$x_1 + x_2 \leqslant m - \frac{m}{x} + 1,$$

где $x_1 + x_2 = m - \frac{m}{x} + 1$ – поглощающий экран, а $x_1 = -1$ и $x_2 = -1$ отражающие экраны. Блуждание начинается в точке $x_1 = x_2 = 0$.

Наша задача состоит в том, чтобы найти среднее время блуждания до поглощения при выходе из начального состояния $x_1 = x_2 = 0$.

Структура матрицы Q для связанного представления совпадает со структурой матрицы Q для страничного представления очередей. Ее определяет Теорема 13.

Был разработан алгоритм и программа, которая вычисляет оценку сверху – T_{max}, и оценку снизу T_{min} для среднего времени блуждания в случае страничного представления, и среднее время для связанного представления очередей.

В параграфе 5 представлены результаты численных экспериментов с разработанными программами.

Из полученных данных можно сделать вывод о том, что для связанного представления очередей, т. е. когда размер страницы $x = 2$, среднее время блуждания до переполнения памяти всегда меньше среднего времени для последовательного представления.

Сравнивая среднее время блуждания для последовательного и страничного представления, можно заметить, что среднее время для последовательного представления в основном меньше оценки сверху T_{max} для страничного представления, но для некоторых наборов вероятностей среднее время для последовательного представления больше чем оценка сверху для страничного.

Для практики может представлять интерес также анализ последовательного представления очередей, когда память делится не оптимально в зависимости от вероятностных характеристик очередей, а просто пополам. Так логично поступать, если мы не знаем заранее вероятностных характеристик очередей. В этом случае последовательное представление двух FIFO-очередей с делением памяти на две равные доли может быть хуже связанного в том случае, когда алгоритмы или

устройства, которые отвечают за включение и исключение элементов в очереди, работают так, что при переполнении одной очереди вторая очередь с очень большой вероятностью является пустой.

В параграфе 6 проанализирован предложенный нами новый метод работы с несколькими последовательными циклическими очередями. В этом методе память заранее не делится между очередями, а две(или более) очереди двигаются по кругу друг за другом, начиная с некоторого начального места в памяти. В этом случае, если одна из очередей стала пустой, то вторая может занимать всю доступную область памяти пока не догонит сама себя. Если же оживет вторая очередь, то ее можно пустить с середины свободного участка или с некоторой оптимальной точки, которая зависит от вероятностных характеристик очередей. Заметим, что два стека, направленнные навстречу друг другу, это заведомо лучший метод в сравнении с раздельным хранением стеков. В случае же очередей, например, Д. Кнут считал, что эффективно работать в общей памяти с двумя очередями нельзя, однако, видимо, он не рассматривал возможности того, что две очереди могут двигаться по кругу одна за другой.

Построить математическую модель этого метода управления очередями не удалось и поэтому была разработана имитационная модель для анализа двух методов(нового предложенного нами и проанализированного в параграфе 2 раздельного метода представления) управления двумя FIFO-очередями в памяти одного уровня.

С помощью разработанной имитационной модели был проведен анализ двух данных методов. Методы сравнивались по таким критериям как среднее время до переполнения памяти, среднее число перемещаемых элементов очередей, средний процент потерянных элементов очередей при бесконечном времени работы.

Для каждой из очередей определены операции включения и исключения элементов, а также – "пустая" операция, т.е. операция не изменяющая длин очередей. Предполагались заданными вероятности всех операций, размер памяти $m > 0$, точка входа второй очереди s и количество итераций модели.

Эксперименты показали, что в случае работы на бесконечном времени раздельный метод хранения очередей лучше, чем наш метод (доля потерянных элементов меньше).

Что касается среднего времени до переполнения памяти, то здесь предложенный нами метод становится лучше, если вероятность включения в одну из очередей достаточно большая, а вероятности включения или исключения в другую равны, или больше вероятность исключения. Т.е. в этом случае в нашем методе часто возникают ситуации, когда одна из очередей пустая, и другая может занимать всю доступную память.

В параграфах 7 и 8 приведены постановки задач управления несколькими очередями, когда разрешено одновременное выполнение операций над очередями.

В параграфе 9 рассмотрена задача оптимального управления очередями в общей памяти в случае бесконечного времени блуждания.

Здесь рассмотрен другой способ управления очередями. Если в предыдущих задачах при переполнении одной из очередей работа прекращалась, то теперь процесс завершен не будет, а будет продолжен следующим образом: если очередь занимает всю предоставленную ей память, то все последующие элементы поступающие в нее отбрасываются до тех пор, пока не появится свободная память (т.е. до тех пор, пока не произойдет исключение элемента из очереди). Такая схема работы применяется в работе сетевых маршрутизаторов в том случае, когда по мере увеличения трафика на исходящем интерфейсе маршрутизатора очередь на нем заполняется пакетами. Такое поведение маршрутизатора называется "сбросом хвоста". Понятно, что процесс сброса пакетов приводит к нежелательному результату, поэтому, число таких ситуаций необходимо свести к минимуму.

Теперь перейдем непосредственно к постановке задачи. Сохранив все обозначения параграфа 2, в качестве модели мы будем иметь блуждание по целочисленной решетке в той же области, причем переходные вероятности для области, ограниченной прямыми $x_1 = -1$, $x_2 = -1$, $x_1 = s$, $x_2 = n - s$ останутся прежними.

Блуждание начинается в точке $x_1 = x_2 = 0$, прямые $x_1 = -1$ и $x_2 = -1$ – отражающие экраны (попытка исключения из пустой очереди). Определим теперь поведение процесса на прямых $x_1 = s + 1$ и $x_2 = n - s + 1$, которые и будут соответствовать ситуациям "сброса хвоста". Рассмотрим сначала прямую $x_1 = s + 1$. Пусть процесс находится в состоянии $(s + 1, x_2)$, т.е. первая очередь заняла всю пре-

доставленную ей память и произошла попытка включения еще одного элемента. С вероятностью p_1 мы остаемся на месте, т.к. при попытке включения элемента в переполненную очередь он будет потерян и для процесса блуждания ничего не изменится, с вероятностью q_1 процесс перейдет в состояние $(s-1, x_2)$, с вероятностью r - в состояние (s, x_2), с вероятностью p_2 – в (s, x_2+1) и с вероятностью q_2 – в (s, x_2-1). Аналогичные рассуждения можно применить для определения поведения процесса на прямой $x_2 = n-s+1$, только рассматривая состояние $(x_1, n-s+1)$. Таким образом, прямые $x_1 = s+1$ и $x_2 = n-s+1$ также можно рассматривать как отражающие экраны и данная модель описывает блуждание в прямоугольнике на бесконечном времени. Ставится задача выбрать такое s, чтобы доля времени, которое процесс проведет на прямых $x_1 = s+1$ и $x_2 = n-s+1$ была минимальной.

Был разработан алгоритм и программа, которая для заданных значений вероятностей и размера памяти генерирует необходимую матрицу и находит предельное распределение для нашей Марковской цепи. Далее, используя теоремы о законе больших чисел для конечных регулярных цепей Маркова, мы устанавливаем оптимальное разбиение памяти между очередями, так, чтобы время проведенное в состояниях, где теряются пакеты, было минимальным. Указанные теоремы устанавливают тот факт, что доли времени, которые процесс проводит в конкретных состояниях цепи на бесконечном времени, определяются вектором предельных вероятностей. По данному вектору для каждого разбиения памяти мы вычисляем суммарную долю времени, которую процесс проводит в тех состояниях, гле теряются элементы(пакеты), включаемые в очереди, то есть на границах прямоугольной области блуждания $x_1 = s+1$ и $x_2 = n-s+1$, которые соответствуют ситуациям "сброса хвоста". Разработаны алгоритм и программа для решения рассмотренной задачи.

В четвертой главе анализируются методы распределения памяти сегментами разных длин.

В параграфе 1 получена оценка минимального размера памяти, необходимого для динамического распределения сегментов разных длин.

Допустим, известно, что в любой момент времени суммарная длина занятых блоков не превышает M единиц памяти (слов), а длина любой заявки не превышает n слов. Предположим далее, что список

свободной памяти упорядочен по возрастанию адресов и для удовлетворения запросов на память используются стратегии "FIRST-FIT" или "BEST-FIT". Обозначим теперь через $N(M,n)$ такой минимальный размер памяти, который следует предоставить системе для того, чтобы при любой последовательности заявок и освобождений, удовлетворяющей приведенным выше ограничениям, не возникало переполнения памяти. В работе(Robson 1971) рассматривалась задача определения $N^*(M,n)$, где M и n имеют тот же смысл, а $N^*(M,n)$ обозначает минимальный необходимый размер памяти, который следует предоставить системе при условии, что посылает заявки и удаляет блоки из памяти некий разумный игрок (нападающий), его противник (защитник) размещает заявки в памяти, причем первый желает увеличить размер занимаемой памяти, а второй стремится помешать этому. В вышеупомянутой статье дана формула для $N^*(M,2)$ и приводятся нижняя и верхняя оценки в общем случае. Легко видеть (это следует из определения величин $N^*(M,n)$ и $N(M,n)$), что верхняя оценка и точная формула не могут быть использованы в том случае, когда для обслуживания свободного списка применяются стратегии "FIRST-FIT" или "BEST-FIT", а не стратегия некоего разумного защитника. Например, $N^*(7,2)=10$, а $N(7,2)=11$.

В этом параграфе получена формула для $N(M,2)$ и верхняя оценка в общем случае. Все следующие рассуждения можно обобщить для получения оценок величины $N(M,l_1,...,l_n)$, где l_i — размеры разрешенных сегментов.

Теорема 14.

$N(M,2)=(3M-1)/2$, если $Mmod4=1 (M \geqslant 5)$;

$N(M,2)=(3M+1)/2$, если $Mmod4=3 (M \geqslant 7)$;

$N(M,2)=3M/2$, если $Mmod2=0 (M \geqslant 4)$.

Из этой теоремы следует, что если используются блоки двух размеров, причем один больше другого в два раза, то в наихудшей ситуации суммарная длина свободных блоков примерно в два раза меньше суммарной длины занятых блоков. Этот результат хорошо согласуется с эмпирическим "правилом пятидесяти процентов" Кнута .

В случае произвольного n получена верхняя оценка для $N(M,n)$.

В параграфе 2 даны оценки эффективности методов динамического

распределения нестраничной памяти.

В пунктах 1 и 2 строятся модели методов "FIRST-FIT" и "BEST-FIT".

В пункте 3 приведена модель метода "OPT-FIT", при которой на каждом шаге помещаем запрос в свободную память так, чтобы максимизировать среднее время до переполнения памяти.

Оптимальная стратегия распределения памяти находится здесь методом динамического программирования.

В пункте 4 построена модель метода динамического распределения памяти основной памяти раздела, который использовался в широко известной в 1980-е годы операционной системе ОС ЕС ЭВМ (IBM 360).

Метод заключается в следующем. Организуется список свободных блоков, упорядоченный по убыванию адресов. Все запросы на память супервизор делит на два класса. Запросы первого класса удовлетворяются согласно стратегии "FIRST-FIT", а для удовлетворения запросов второго класса проводится поиск самого последнего подходящего блока в списке свободных блоков. Такой метод заведомо увеличивает время поиска свободного блока по сравнению с методом "FIRST-FIT". Интересно отметить, что время поиска в таком методе распределения памяти можно сократить, если связать свободные блоки в список с двумя связями так, что поиск можно начинать то с одного, то с другого конца списка свободной памяти. С другой стороны, следует выяснить, компенсируются ли затраты на поиск снижением степени фрагментации памяти.

Обозначим через $x_1, ..., x_n$ вектор длин свободных блоков, упорядоченный по убыванию адресов. Пусть p — вероятность того, что запрос будет размещаться в первом подходящем свободном блоке, а $1 - p$ – в последнем подходящем свободном блоке. Определена функция $T(x_1, ..., x_n)$, которая вычисляет среднее время до переполнения памяти, если начальным было состояние памяти $x_1, ..., x_n$.

В пункте 5 приведены результаты численных экспериментов с разработанными программами. Проведенные эксперименты по сравнению методов "FIRST-FIT" и "BEST-FIT" показали, что, например, для равномерного закона распределения вероятностей стратегия "BEST-FIT" эффективнее.

Для анализа метода динамического распределения памяти ОС ЕС ЭВМ было проведено несколько экспериментов (использовались равно-

мерный, биномиальный и экспоненциальный законы распределения). Для всех серий экспериментов время до переполнения оказывалось больше, если запросы удовлетворяются преимущественно с одного из концов списка свободных блоков. Самое худшее время имеем при $p = 0.5$, а лучшие времена при $p = 1$ (метод "FIRST-FIT") и при $p = 0$ (метод "последнего подходящего"). Если основываться на этих данных, то приходим к выводу, что метод динамического распределения памяти, принятый в ОС ЕС ЭВМ, увеличивая затраты на поиск свободных блоков, не снижал степени внешней фрагментации памяти. Следовательно, существовал весьма простой способ улучшения принятого в ОС ЕС ЭВМ метода распределения памяти, а именно: нужно был отказаться от деления супервизором всех запросов на классы и все запросы размещать согласно стратегии "FIRST-FIT". Практически это, видимо, свелось бы просто к исключению нескольких команд из супервизора ОС ЕС.

В Приложении приведены некоторые результаты численных экспериментов с моделями, разработанными для оптимального управления тремя стеками в памяти одного и двух уровней.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ ПО ТЕМЕ ДИССЕРТАЦИИ

1. *Мазалов В. В., Соколов А. В.* Об оптимальном динамическом распределении нестраничной памяти. Управление в динамических системах. Л., 1979,с.206-214. Деп. в ВИНИТИ 24.07.1979.

2. *Соколов А.В.* Оценка минимального размера памяти, необходимого для динамического распределения сегментов разных длин // Автоматизация эксперимента и обработки данных. Петрозаводск, 1980. С. 59–65.

3. *Соколов А. В.* О распределении памяти для двух стеков // Автоматизация эксперимента и обработки данных. Петрозаводск, 1980. С. 65–71.

4. *Соколов А. В.* Оптимальное динамическое распределение нестраничной виртуальной памяти: Дис. канд. физ.-мат. наук. ЛГУ, 1981.

5. *Соколов А. В.* Математические модели динамического распределения памяти // Тезисы докл. I Всесоюз. науч. конф. "Вероятностные методы в дискретной математике" / КФ АН СССР. Петрозаводск, 1983. С. 28–32.

6. *Соколов А. В.* Анализ метода динамического распределения памяти в ОС ЕС ЭВМ // Математическое обеспечение ЭВМ и систем управления. Петрозаводск, 1985. С. 28–32.

7. *Соколов А. В.* Оценка эффективности методов динамического распределения нестраничной памяти // Программирование. 1986. № 5. С. 65–71.

8. *Соколов А. В.* Об оптимальном управлении стековой памятью //Тезисы докл. I Всесоюз. науч. конф. "Вероятностные методы в дискретной математике" / КФ АН СССР. Петрозаводск, 1988. С. 115.

9. *Соколов А. В.* Оптимизация динамических структур данных. Учебное пособие. Петрозаводск: Изд-во ПетрГУ, 1993. 58 с.

10. *Соколов А. В.* Анализ эффективности алгоритмов динамического распределения нестраничной памяти // Труды II Междунар. конф. "Развитие и применение открытых систем". Петрозаводск, 1995. С. 76–77.

11. *Sokolov A. V.* On the problem of optimal stack control in two levels memory // Probabilistic methods in discrete mathematics: Proceedings of the Fourth International Petrozavodsk Conference. Utrecht, Netherlands: VSP, 1997. P. 349–351.

12. *Sokolov A. V.* On the problem of stack control // Proceedings' of Finnish Data Processing Week. 1999. Vol. 2. Petrozavodsk: Petrozavodsk University Press. P. 95–106.

13. *Соколов А. В.* Математические модели и алгоритмы оптимального управления динамическими структурами данных // Обозрение прикладной и промышленной математики. 2000. Т. 7. Вып. 1. С. 199–200.

14. *Соколов А. В.* Об оптимальном управлении очередями в ограниченной памяти // Обозрение прикладной и промышленной математики. 2000. Т. 7. Вып. 2. С. 419–421.

15. *Sokolov A. V.* Mathematical Models and Optimal Algorithms of Dynamic Data Structure Control // Fundamental of Computation Theory. 13th International Symposium, FCT 2001, Proceedings, In:Lecture Notes in Computer Science. vol. 2138, Springer-Verlag, p. 416–419.

16. *Sokolov A. V., Lemetti A. A.* On the problem of optimal stack control // Probabalistic methods in discrete mathematics: Proceedings of the Fifth International Conference. Utrecht, Netherlands: VSP, 2001. P. 351–366.

17. *A.V. Sokolov* Mathematical Models and Optimal Algorithms of Dynamic Data Structure Control // Abstracts. The 3rd Moscow International Conference On Operations Research (ORM2001) (Moscow, April 4-6, 2001) p. 109–110.

18. *Соколов А. В.* Создание персональной научной коллекции по теме "Математические модели и алгоритмы оптимального управления динамическими структурами данных" // Сборник аннотаций стендовых докладов III Всероссийской конференции по электронным библиотекам. Петрозаводск, 2001. С. 22–23.

19. *Соколов А. В., Тарасюк А. В.* Об оптимальном управлении циклическими очередями // Труды Института прикладных математических исследований КарНЦ РАН. 2001. Вып. 3. С. 190–195.

20. *А.В. Соколов* Вероятностные модели динамических структур данных // Труды третьего Всероссийского симпозиума по прикладной и промышленной математике. Сочи, 1-6 октября, 2002, Обозрение прикладной и промышленной математики. Москва 2002, С. 453.

21. *Соколов А.В.* Математические модели и алгоритмы оптимального управления динамическими структурами данных. Изд. ПГУ. 2002 г. 215 с.

22. *Соколов А.В.* Математические задачи оптимального управления динамическими структурами данных. Проблемы теоретической кибернетики. // Тезисы докладов XIII Международной конференции. (Казань, 27-31 мая 2002). Москва 2002. Изд. центра прикладных исследований при механико-математическом факультете МГУ. С. 169.

23. *Соколов А.В.* Математические задачи теории структур данных. Карелия и РФФИ // Тезисы докладов научной конференции, посвященной 10-летию РФФИ. Петрозаводск 2002. С. 96–97.

24. *Аксенова Е. А., Волкова О. В., Лазутина А. А., Соколов А. В.* Методы оптимального управления стеками в двухуровневой памяти // Труды Института прикладных математических исследований КарНЦ РАН. 2002. Вып. 3. С. 127–152.

25. *A.V. Sokolov, E.A. Aksenova, A.A. Lazutina, A.V. Tarasyuk* Mathematical models of dynamic data structure // V international congress on mathematical modelling. Book of abstract, vol. II, JINR, Dubna 2002, P. 127.

26. *Sokolov A.V.* Optimization strategies of stack control // Proceedings of the Second Workshop on Intermediate Reprezentation Engineering for Virtual Machins and Inaugural Conference on the Principles and Practice of Progrmming in Java. Trinity College Dublin, PPPJ 2002/IRE 2002, June 13-14, 2002. P. 151–156.

27. *Sokolov A.V.* // Optimization strategies of stack control. Chapter 22. In. Recent Advances in Java Technology. Theory, Application, Implementation. Intermediate Reprezentation Engineering. Computer Science Press, Trinity College Dublin 2002. P. 193–203.

28. *Е.А. Аксенова, А.А. Лазутина, А.В. Соколов, А.В. Тарасюк* Об оптимальном управлении динамическими структурами данных // Труды V международной конференции "Дискретные модели в теории управляющих систем." Ратмино, Изд. отдел ВМК МГУ им. М.В. Ломоносова, 2003. С. 5–6.

29. *Е.А. Аксенова, А.А. Лазутина, А.В. Соколов, А.В. Тарасюк* Оптимальные методы динамического распределения нестраничной памяти // Труды Первой Всероссийской научной конференции "Методы и средства обработки информации". Изд. отдел ВМК МГУ им. М.В. Ломоносова, 2003. С. 74–79.

30. *А.В. Соколов, А.В. Тарасюк.* Об оптимальном управлении двумя последовательными очередями // Обозрение прикладной и промышленной матем. 2003 т. 10, в. 1, с. 223–224.

31. *Соколов А.В. , Ислентьев Д.О., Колосов А.С.* Анализ нестандартных методов представления связных списков // Труды ИПМИ КарНЦ РАН. Методы математического моделирования и информационные технологии. Выпуск 4. 2003. С. 188–193.

32. *Е.А. Аксенова, А.А. Лазутина, А.В. Соколов* Об оптимальном распределении памяти для стеков // Обозрение прикладной и промышленной матем. 2003 т. 10, в. 1, с. 86–87.

33. *Е.А. Аксенова, А.А. Лазутина, А.В. Соколов* Об оптимальных методах представления динамических структур данных // Обозрение прикладной и промышленной матем. 2003 т. 10, в. 2. С. 375–376.

34. *Аксенова Е.А., Лазутина А.А. , Соколов А.В.* Исследование немарковской модели управления стеком в двухуровневой памяти // Программирование, 2004, № 1. С. 1–10.

35. *Е.А. Аксенова, А.В. Соколов* Об оптимальном управлении двумя FIFO очередями. Материалы VIII Международного семинара "Дискретная математика и ее приложения", с. 167-169. М.: Изд-во механико-математического факультета МГУ, 2004, 444 с.

36. *Соколов А.В., Соломатов В.А.* Анализ методов управления двумя FIFO очередями // Труды ИПМИ КарНЦ РАН. Методы математического моделирования и информационные технологии. Выпуск 5. 2004. С. 121–127.

37. *Е.А. Аксенова, А.А. Лазутина, А.В. Соколов.* Анализ методов представления динамических структур данных // Обозрение прикладной и промышленной математики. 2004, т.11, в.2. С. 233–234

38. *Е. А. Аксёнова, А. В.Соколов* Оптимальные методы управления FIFO-очередями в памяти одного уровня. Тезисы докладов XIV Международной конференции "Проблемы теоретической кибернетики"// Изд-во центра прикладных исследований при механико-математическом факультете МГУ, Москва 2005. С. 8.

39. *Лазутина А. А. , Соколов А.В.* Оптимальное управление тремя стеками в памяти одного уровня. Тезисы докладов XIV Международной конференции "Проблемы теоретической кибернетики"// Изд-во центра прикладных исследований при механико-математическом факультете МГУ, Москва 2005. С.84.

40. *Соколов А.В., Тарасюк А.В.* Об оптимальном управлении двумя FIFO-очередями на бесконечном времени. Тезисы докладов XIV Международной конференции "Проблемы теоретической кибернетики"// Изд.-во центра прикладных исследований при механико-математическом факультете МГУ, Москва 2005. С. 148.

41. *А.В. Соколов, А.В. Тарасюк.* Об оптимальном управлении циклическими FIFO-очередями // Системы управления и информационные технологии. 2005, № 3(20). С. 29–33.

42. *Е.А. Аксенова, А.В. Соколов,* Некоторые задачи оптимального управления FIFO очередями // Труды Второй Всероссийской научной конференции "Методы и средства обработки информации". Изд. отдел ВМК МГУ им. М.В. Ломоносова, 2005. С. 318–322.