

Эффективные алгоритмы и их программные реализации

Н.Н. Кузюрин, В.А. Захаров, А.Н. Холодов,
Л.В. Шабанов, А.В. Шокуров

1. Введение

Настоящая статья посвящена результатам, полученным в последнее время рядом сотрудников отдела математических методов и алгоритмов ИСП РАН в области разработки эффективных алгоритмов и их программных реализаций. Прежде чем описывать эти результаты, постараемся бросить общий взгляд на проблематику, связанную с разработкой и анализом эффективных алгоритмов, и отметить некоторые характерные тенденции, присущие ей в последнее десятилетие.

При расчете любой математической модели явно или неявно присутствуют два этапа: построение математического (и по возможности эффективного) алгоритма для произведения ее расчета и его реализация в виде конкретного программного обеспечения [6]. Снижение вычислительных затрат возможно на любом из этих двух этапов, однако используемые при этом методы различны.

В англоязычной литературе задачи построения и анализа алгоритмов с точки зрения их эффективности и практической целесообразности объединяются в раздел, называемый теорией алгоритмов (см., например, [41]), в то время как более абстрактными и математическими вопросами существования алгоритмов с теми или иными характеристиками занимается теория сложности вычислений (см., например, [51]). В отечественной литературе эти две смежные дисциплины традиционно объединяются в одну, также называемую теорией сложности вычислений. Чтобы подчеркнуть, что выбор конкретной программной реализации выходит за рамки теории сложности, строящиеся в литературе эффективные алгоритмы обычно описываются на наиболее примитивном псевдоалгоритмическом языке.

Проблема создания эффективной программной реализации данного алгоритма на конкретной вычислительной архитектуре называется отображением алгоритма на архитектуру вычислительной системы. Учет особенностей вычислительной системы (параллельность, конвейерность, наличие быстрого кэша и т.п.) позволяет иногда выиграть (или проиграть при неудачном их использовании) в эффективности в десятки или сотни раз.

Эти проблемы зачастую не имеют решающего значения ввиду быстрого роста

производительности процессоров и объемов памяти и выступают на первый план лишь при решении трудоемких задач. К их числу относятся, например, NP-трудные задачи дискретной оптимизации, задачи линейного программирования большой размерности, задачи анализа огромных массивов информации (data mining) и т.п. Даже простая задача сортировки для огромных файлов становится сложной вычислительной задачей, требующей иногда десятков и сотен часов для ее решения и, соответственно, использования эффективных алгоритмов (подробнее об этом см. в разд. 2 настоящей работы). Кроме того, потребность в эффективности выступает на первый план при создании стандартного программного обеспечения, хотя и не требующего больших вычислительных затрат, но в силу постоянного и частого использования влияющего на общую производительность вычислительной системы. Эти вопросы затрагиваются в разд. 5 настоящей статьи, в котором приводятся результаты исследования эффективности отображения на архитектуру современных рабочих станций с конвейерной архитектурой программ прикладной статистики.

Основная тенденция последних лет в разработке эффективных алгоритмов заключается в быстром развитии области дискретных алгоритмов. Их доля стремительно растет в последние годы. Проводится ежегодный симпозиум по дискретным алгоритмам (Symposium on Discrete Algorithms – SODA), на двух крупнейших ежегодных симпозиумах по проблемам Theoretical Computer Science (IEEE FOCS и ACM STOC) дискретные алгоритмы также бывают представлены достаточно широко.

В последнее десятилетие главный прогресс в разработке эффективных дискретных алгоритмов (и не только их) связан с вероятностными алгоритмами, т.е. алгоритмами, использующими в процессе своей работы результат случайного бросания монеты (или, как говорят, случайные биты информации) [53]. Такие алгоритмы, хотя и не подпадают под классическое определение алгоритма [1], являются, на самом деле, весьма практичными. Для огромного числа задач удается разработать более эффективные вероятностные алгоритмы (по сравнению с детерминированными) или, по крайней мере, гораздо более простые по структуре и реализации [53].

Одними из наиболее широко распространенных математических моделей, описывающих самые разные процессы реальности, являются линейное и целочисленное программирование [26]. Теория двойственности и симплекс-метод составляют краеугольные камни в этой области, опираясь на которые удалось решить огромное число практических задач оптимизации. Однако, несмотря на эти успехи, в последние годы линейное программирование (ЛП) переживает период стремительного развития. Это началось с осознания [46] того факта, что, несмотря на свою практическую, симплекс-метод не является эффективным на любых исходных данных, а только на подавляющей доле их (или другими словами не является полиномиальным алгоритмом). Первый полиномиальный алгоритм для задачи линейного программирования предложил Л.Г. Хачиян [29]. Его работа послужила стимулом для создания большого количества полиномиальных алгоритмов решения ЛП-задач, использующих так называемый метод внутренней точки [25], [45] и конкурирующих на практике с симплекс-методом.

Затем были разработаны еще более эффективные алгоритмы, которые находят приближенные решения с гарантированными оценками точности, причем точность выступает как параметр алгоритма [66]. Для задач ЛП с неотрицательными исходными данными были разработаны параллельные алгоритмы, трудоемкость которых уже близка к неулучшаемой [52], [37]. Как отмечается в разделе 2 настоящей работы, эти алгоритмы (или их модификации) обладают еще одним замечательным свойством – являются коммуникационно-эффективными.

Значимость новых эффективных алгоритмов для задач ЛП еще более возросла, когда было понято, что для решения существенно более сложных задач ЛП с ограничением, что переменные целочисленны (т.е. задач целочисленного линейного программирования – ЦЛП), можно с успехом применять метод вероятностного округления нецелочисленного решения их линейной релаксации (то есть той же задачи ЛП без ограничения целочисленности переменных) до целочисленного. Впервые этот метод построения эффективных вероятностных алгоритмов с гарантированными оценками точности был предложен в [56], а вскоре был разработан метод их дерандомизации, т.е. конвертирования таких вероятностных алгоритмов в детерминированные [57], [18]. Этот метод стал весьма широко использоваться и получил название *метод условных вероятностей*. Применительно к ЦЛП эти методы рассматривались в работах [18], [57], [19], [64]. На их основе для задач ЦЛП с неотрицательными данными удалось построить приближенные алгоритмы с неулучшаемыми гарантированными оценками точности. Однако для

задач ЦЛП общего вида эти вопросы еще мало исследованы и представляют весьма интересное и перспективное направление. Этой проблематике посвящен разд. 4 настоящей статьи. Там же рассматриваются и близкие вопросы о явных конструкциях различных объектов, понимаемых как эффективные алгоритмы их построения.

Одним из современных направлений исследований является разработка так называемых коммуникационно-эффективных алгоритмов [42]. Его актуальность объясняется широким распространением параллельных и распределенных вычислительных систем различных типов и тем фактом, что коммуникационные затраты в них зачастую намного превосходят собственно вычислительные. Этим вопросам с точки зрения коммуникационной сложности [47] посвящен разд. 2 настоящей работы.

Во многих приложениях возникает задача нахождения коэффициентов разложения функции по некоторому базису, если известно разложение этой функции по другому базису. По-видимому, самым известным и практически используемым разложением такого рода является дискретное преобразование Фурье. Наличие эффективного алгоритма или быстрого преобразования Фурье сделало его использование чрезвычайно широким. Имеется, однако, много других классических базисов из ортогональных многочленов, также требующих разработки соответствующих эффективных алгоритмов разложения. Этой проблематике и достигнутым в ней результатам посвящен разд. 6 настоящей работы.

В заключительном разделе кратко рассмотрены алгоритмические вопросы формального анализа программ.

2. Коммуникационно-эффективные алгоритмы

В связи с широким распространением глобальных телекоммуникационных сетей актуальность приобретает разработка эффективных методов решения задач оптимизации, когда различные части данных могут находиться в весьма удаленных друг от друга местах. Одним из естественных подходов к решению таких задач является использование локальных алгоритмов. Теория локальных алгоритмов была разработана Ю.И. Журавлевым еще в конце 50-х годов в связи с исследованием сложности решения так называемых переборных задач [9]-[11] и доказательством их неразрешимости в классах алгоритмов с ограничениями. В последние годы этот подход использовался при нахождении точности, которой можно достигнуть локальными алгоритмами в задаче линейного программирования [55].

2.1 Оценка точности распределенного решения задач ЛП и ЦЛП с точки зрения коммуникационной сложности

Мы исследовали подход к оценке точности распределенного решения задач ЛП и ЦЛП с точки зрения коммуникационной сложности [47]. При этом предполагается, что исходные данные распределены между двумя пользователями, каждый из которых знает только свою часть данных. Под коммуникационной сложностью задачи понимается минимальное по всем алгоритмам, обеспечивающим нахождение решения, число пересылаемых битов. Важно отметить, что нами рассматривались приближенные алгоритмы и изучался вопрос соотношения точности и коммуникационной сложности.

Рассматривалась задача целочисленного линейного программирования

$$\max cx \quad (1)$$

$$Ax \leq b, x \geq 0 - \text{целые}$$

с $m \times n$ матрицей ограничений A , состоящей из неотрицательных элементов, пронормированных так, что $a_{ij} \leq 1$. Линейная релаксация задачи (1) отличается от нее отсутствием требования целочисленности переменных

$$\max cx \quad (2)$$

$$Ax \leq b, x \geq 0.$$

Предположим, что каждый из пользователей знает лишь половину матрицы A (а именно, 1-й пользователь знает столбцы с номерами $\{1, \dots, n/2\}$, а второй – с номерами $\{n/2+1, \dots, n\}$), и оба знают целевую функцию и правую часть ограничений. Как они могут решить эту распределенную задачу, по возможности минимально общаясь между собой? Иначе говоря, вопрос заключается в определении точности, с которой распределенную задачу можно решить с заданной коммуникационной сложностью.

Назовем число r мультипликативной точностью алгоритма, если всякое решение задачи, полученное с его помощью, отличается от оптимума не более чем в r раз. Алгоритм называется ε -оптимальным, если он гарантирует мультипликативную точность $1 + \varepsilon$.

Имеется тривиальная стратегия, при которой исходные данные от второго пользователя пересылаются к первому, который и находит точное решение задачи. Коммуникационная сложность при этом определяется пересылкой половины элементов матрицы ограничений. Имеются также простые стратегии (назовем их стратегиями 1 и 2 соответственно). Стратегия 1 заключается в том, что каждый пользователь решает усеченную задачу (1) со своим блоком столбцов, своими переменными и правой частью b , а затем из найденных значений выбирается максимальное. Остальные переменные при этом полагаются равными нулю. Тогда допустимость полученного решения очевидна, а

коммуникационная сложность есть $O(\log(1+z))$, где z – величина оптимума (1). Стратегия 2 заключается в том, что каждый пользователь решает усеченную задачу (2) со своим блоком столбцов, своими переменными и правой частью $b/2$. Допустимость полученного решения очевидна, а коммуникационная сложность при этом равна нулю. При условии неотрицательности исходных данных нетрудно доказать следующее

Утверждение. Стратегия 1 (соответственно, 2) распределенного решения гарантирует для задачи (1) (соответственно, (2)) мультипликативную точность 2.

Таким образом, имеется стандартная стратегия пересылки всех данных из одной части в другую, которая дает точное решение, но имеет большую коммуникационную сложность. И есть простые стратегии (1 и 2), которые имеют небольшую коммуникационную сложность, но не гарантируют нахождение решения с наперед заданной точностью.

В результате наших исследований доказано существование стратегий решения задач ЛП и ЦЛП с неотрицательными данными, существенно более экономных по критерию коммуникационной сложности, чем стандартные. Сформулируем некоторые из полученных результатов.

В [20] доказаны следующие утверждения.

1. Любая стратегия решения распределенной задачи (2), имеющая коммуникационную сложность не более t , не может гарантировать мультипликативную точность менее 2. Таким образом, имеется нижняя граница для всех алгоритмов. Оказывается, что существенно улучшить ее нельзя.
2. Для любого фиксированного $\varepsilon > 0$ существует ε -оптимальный алгоритм решения задачи (2), имеющий коммуникационную сложность $O(m (\log nm)^{const})$.

Эта верхняя оценка отличается от нижней оценки t только полилогарифмическим множителем. Таким образом, алгоритм имеет близкую к минимальной коммуникационную сложность, т.е. является коммуникационно-эффективным. Оказалось, что свойством коммуникационной эффективности обладают и новые параллельные ЛП-алгоритмы [52], [37]. Модификация последнего алгоритма, существенно снижающая общее число выполняемых операций, предложена в [27].

Для задачи ЦЛП (1) в [22] построен детерминированный алгоритм нахождения приближенного целочисленного решения с гарантированными оценками точности и коммуникационной сложности. Более точно, доказано следующее:

3. Для любого фиксированного $\varepsilon > 0$ существует полиномиальный алгоритм решения распределенной задачи (1), который при условии (при $n \rightarrow \infty$) $\log t = o(\min_i b_i)$

гарантирует мультипликативную точность $1+\epsilon$ и имеет коммуникационную сложность $O(m(\log nm)^{const})$.

Этот алгоритм основан на процедуре независимого (в каждой части) вероятностного округления нецелочисленного решения линейной релаксации до целочисленного с последующей дерандомизацией. Отметим, что условие, при котором доказано существование коммуникационно-эффективного алгоритма решения задачи ЦЛП (1), является стандартным достаточным условием близости целочисленного оптимума и оптимума линейной релаксации [18], [65], [36].

2.2 Коммуникационно-эффективный алгоритм внешней сортировки

Сортировка больших файлов всегда была важной практической задачей обработки информации. В последние годы к категории "больших файлов" принято относить файлы, имеющие размер в несколько сотен мегабайт. Теперь к реальным задачам относится и сортировка "сверхбольших файлов", размер которых составляет один и более терабайт. Некоторое представление о возможностях современной вычислительной техники дает время сортировки (2,5 часа) файла, содержащего 10000000000 100-байтных записей (ключи длиной в 10 байт имеют равномерное распределение). Эта сортировка была выполнена на 32 процессорах параллельной машины Origin2000. При этом использовалось 8 GB основной памяти и 559 4 GB дисков.

Высокая скорость сортировки на современных параллельных машинах (распределенных вычислительных системах, системах с массовым параллелизмом, машинах с общей памятью и др.) может быть достигнута благодаря использованию эффективных параллельных алгоритмов сортировки. Особое место среди них занимают алгоритмы внешней сортировки. На эффективность алгоритмов параллельной сортировки сильное влияние оказывают коммуникационные затраты, которые неизбежны на фазе слияния отсортированных блоков данных и которые могут оказаться преобладающими в алгоритмах сортировки слияниями.

Построению и анализу эффективных в смысле коммуникационных расходов алгоритмов сортировки посвящено немало научных статей (например, [42]). Интересно, что вопросы минимизации коммуникационных затрат в параллельных сортировках близки к вопросам минимизации количества обменов с внешней памятью (как правило, с памятью на дисках) для внешних сортировок. Последнее обстоятельство мотивировало проведение нашего исследования подходов к построению эффективных внешних сортировок.

В результате исследований способов построения коммуникационно-эффективных внешних сортировок Л.В.Шабановым была предложена схема внешней сортировки, в которой при некотором заданном размере основной памяти количество промежуточных записей и считываний ключей (и связанных с ними данных) было сведено к минимуму. В предложенной схеме использован только один вспомогательный файл, размер которого равен размеру исходного (сортируемого) файла. Сортировка проходит в два шага. На первом производится считывание исходного файла, разбиение его на блоки равной длины, внутренняя сортировка каждого блока и запись упорядоченных по ключам блоков в один промежуточный файл. Известно, что эффективность внутренней сортировки существенно влияет на эффективность внешней сортировки. Предварительный анализ и эксперименты показали, что в случае 32-разрядных целочисленных ключей значительным преимуществом (более чем в два раза) по сравнению с другими традиционными методами (быстрая сортировка, пирамидальная сортировка и др.) обладает поразрядная сортировка. Для 64-разрядных ключей преимущество менее значительно (примерно в 1,5 раза).

На втором шаге внешней сортировки производится слияние упорядоченных блоков. Для слияния Л.В.Шабановым предложена простая схема, которая реализует поиск и считывание сравнительно коротких упорядоченных последовательностей ключей из промежуточного файла в специальный буфер и слияние этих последовательностей с помощью динамической структуры данных (AVL-дерева). Таким образом производится слияние и вывод (запись в выходной файл) небольших упорядоченных по ключам последовательностей записей. Подкачка данных в буфер производится по сигналу схемы слияния в моменты, когда для выполнения слияния необходима новая порция данных, содержащихся в промежуточном файле. Следует отметить, что вместо AVL-деревьев для слияния отсортированных подпоследовательностей можно использовать другие способы и средства. Например, некоторая модификация поразрядной сортировки будет более эффективным способом реализации слияния для случая сортировки больших файлов, когда длины сливаемых последовательностей сравнительно велики (десятки и сотни тысяч элементов).

Результаты проведенных нами экспериментов показали, что предложенная Л.В.Шабановым схема внешней сортировки позволяет эффективно сортировать большие исходные файлы с использованием одного промежуточного файла при минимально возможном количестве промежуточных записей и считываний (одна запись и одно считывание на ключ). При этом для сортировки достаточно выделить небольшой

сегмент основной памяти. Высокая эффективность сортировки сохраняется, например, в случае когда размер выделяемой основной памяти в 50 раз меньше размера в байтах исходного файла. Экспериментальное исследование также показало, что по производительности предложенный алгоритм не уступает лучшим коммерческим продуктам в данной области.

Основной задачей дальнейшего исследования предложенной схемы внешней сортировки является ее оптимизация, т.е. выбор параметров, доставляющих наибольшую эффективность при заданных ограничениях (например, при заданном размере выделяемой для внешней сортировки основной памяти).

3. Методы дерандомизации, явные конструкции и эффективные алгоритмы

Исследования, проводимые в последние десятилетия, продемонстрировали исключительную силу вероятностных методов и вероятностных алгоритмов [34], [53]. Идея аппроксимации различных классов целочисленных линейных программ путем нового способа округления – *вероятностного округления*, оказавшаяся чрезвычайно продуктивной, была предложена в [56]. Для положительного вещественного числа v идея состоит в том, чтобы рассматривать его дробную часть как вероятность, округляя v до $\lfloor v \rfloor + 1$ с вероятностью $v - \lfloor v \rfloor$, и округляя v до $\lfloor v \rfloor$ с вероятностью $1 - (v - \lfloor v \rfloor)$. Важным свойством такого округления является то, что математическое ожидание при этом равно v .

Как это можно использовать для нахождения приближенных решений задачи ЦЛП (1)? Пусть найдено решение ее линейной релаксации $x^* = (x_1, \dots, x_n)$ и значение оптимума этой линейной программы есть q . Положим $x_i = x_i/k$, где k – некоторый параметр, значение которого будет выбрано далее. Определим случайный вектор $z \in Z^n$ следующим образом. Независимо для каждого i положим z_i равным $\lfloor x_i \rfloor + 1$ с вероятностью $p_i = x_i - \lfloor x_i \rfloor$, и $x_i - \lfloor x_i \rfloor$ с вероятностью $1 - p_i = 1 - (x_i - \lfloor x_i \rfloor)$.

Используя такие округления, для задач ЦЛП с неотрицательными данными удалось построить приближенные алгоритмы с неулучшаемыми оценками точности [18], [57], [19], [64]. Однако для задач ЦЛП общего вида эти вопросы еще предстоит исследовать. Исследование возможностей вероятностных методов в задачах ЦЛП общего вида и выделение классов задач ЦЛП, для которых вероятностные округления дают нетривиальные гарантированные оценки точности, является важным как с теоретической, так и с практической точек зрения. Некоторые результаты в этом направлении получены в [19] и [35].

В [57], [18] независимо был разработан метод дерандомизации, т.е. конвертирования

соответствующих вероятностных алгоритмов в детерминированные. Этот метод в последние годы широко и успешно использовался при построении детерминированных алгоритмов для различных задач и получил название метод условных вероятностей (см. [53]). Поскольку алгоритмы вычисления условных вероятностей (на которых базируется построение детерминированного алгоритма) в [57] и [18] кардинально различны, интерес представляет нахождение близких к оптимальным алгоритмов вычисления условных вероятностей в задачах ЦЛП. Алгоритм из [57] применим только для задач ЦЛП с 0–1 коэффициентами и имеет сложность $O(mn)$ (в модели unit-cost RAM, т.е. в числе арифметических операций) и не переносится прямо на задачи более общего вида. Алгоритм сложности $O(mn^2 \log m)$ для вычисления условных вероятностей по схеме [57] для задач ЦЛП с произвольными неотрицательными коэффициентами был предложен в [65]. Алгоритм из [18] (см. также [21]) для тех же задач имеет сложность $O(mn^2)$ (при любой фиксированной точности вычислений). Любой прогресс в уменьшении сложности вычисления условных вероятностей и приближение к сложности вида $O(mn)$ представляет несомненный интерес и перспективное направление в области разработки эффективных алгоритмов.

Как мы уже упоминали, ввиду широкого применения вероятностных методов, многие результаты в комбинаторике, теории чисел, теории графов являются неконструктивными. Поэтому все большее внимание уделяется построению так называемых *явных конструкций* и технике дерандомизации [54].

Обычно под явной конструкцией некоторого множества понимается эффективный алгоритм построения этого множества (или произвольного его элемента). Таким образом, здесь мы также сталкиваемся с необходимостью разработки эффективных алгоритмов. Это направление находится в стадии зарождения и успехи его еще не столь заметны. Однако потребность в нем стимулируется практикой, что сделало его весьма популярным [34], [54]. В частности, в теории конечных полей, являющихся базой современной криптографии и теории кодирования, большинство конструкций не являются явными в этом смысле (например, неизвестны эффективные алгоритмы нахождения по данному числу близкого к нему простого, построения конечного поля заданного размера и т.п. [62]). Поэтому обычно используют заранее найденные таблицы (например, таблицу простых чисел, не превосходящих заданного числа), что приводит к *неуниформности* конструкции. К тому же, этот подход накладывает естественные и довольно жесткие ограничения на величину рассматриваемых чисел.

В [63] (см. также [62]) получен интересный результат, который открывает новые возможности в построении явных конструкций, использующих конечные поля. Для данного числа Q за полиномиальное время (от $\log Q$) там построено конечное поле размера q , причем $q \sim Q$ при $Q \rightarrow \infty$. Однако области применения этого результата еще предстоит найти, что представляется интересной проблемой для исследования. Ведь все конструкции, использующие большие простые числа, являются неuniformными (например, широко известная криптосистема RSA), и нахождение uniformных аналогов (или явных конструкций) представляет несомненный интерес. В этой связи упомянем результат [50], дающий явную конструкцию так называемых асимптотически хороших упаковок – классических комбинаторных объектов, существование которых было доказано в 1985 г. В. Редлем (см. [34]).

В заключение этого раздела, посвященного возможностям вероятностных методов и алгоритмов, упомянем новое направление – вероятностные алгоритмы для *квантовых компьютеров* – сравнительно новой вычислительной модели, введенной в рассмотрение в 1985 г. В отличие от многих других моделей вычислений квантовые компьютеры в принципе физически реализуемы. В то же время, есть много указаний на то, что по своей силе (вычислительным возможностям) они могут значительно превосходить обычные. Особенно яркий пример этого содержится в [61], где предложены вероятностные алгоритмы нахождения дискретных логарифмов и факторизации целых чисел (разложения на простые множители), имеющие полиномиальную по числу бит трудоемкость. Эти две задачи обычно рассматриваются как вычислительно трудные (для них неизвестны полиномиальные алгоритмы) и на них базируются многие современные криптосистемы (например, криптосистема RSA). Нахождение других задач, эффективно решаемых вероятностными алгоритмами на квантовых компьютерах (типа проверки изоморфизма графов или какой-либо NP-полной задачи), является, несомненно, одной из важных и интересных проблем в области разработки эффективных алгоритмов.

4. Опыт решения задач вычислительной геометрии

4.1. Задача линейного программирования фиксированной размерности

За последние годы было предложено большое число эффективных рандомизированных алгоритмов решения задач комбинаторной оптимизации и вычислительной геометрии. Один из таких алгоритмов – это алгоритм Кларксона [40], решающий задачу линейного программирования с n ограничениями и d

переменными в среднем за время $O(n)$, при условии, что d является константой.

Этот алгоритм и его модификации использовались нами как основа для разработки эффективных алгоритмов и программ для решения ряда задач вычислительной геометрии: задачи нахождения для заданного политопа вписанного шара максимального объема, нахождения крайних точек множества в d -мерном пространстве и т.д.

Рассматривается задача линейного программирования фиксированной размерности

$$\begin{aligned} cx &\rightarrow \min \\ Ax &\geq b, \end{aligned} \quad (3)$$

где c, x – d -мерные векторы, A – $n \times d$ матрица.

Алгоритм Кларксона вычисляет оптимум (3) следующим образом. Если $n \leq c_d$, то оптимум находится с помощью симплекс-метода. Если $n > c_d$, то повторяется следующее с V_0 (первоначально пустым): пусть S обозначает множество линейных неравенств в (3), $R \subset S \setminus V_0$ – случайное подмножество размера r . Пусть $x_0 \leftarrow x_0 (R \cup V_0)$ – решение задачи ЛП с множеством неравенств-ограничений $R \cup V_0$, определенное рекурсивно, и пусть V – множество ограничений, нарушаемых x_0 . Если $|V| > 2\sqrt{n}$, то сделать новую попытку с новым подмножеством R , повторяя до тех пор, пока $|V| \leq 2\sqrt{n}$. Включить заключительное множество V в V_0 . Когда V пусто, остановиться и выдать x_0 в качестве искомого оптимума $x_0(S)$. В противном случае повторять вычисления в цикле.

Модифицированный алгоритм [48] решения ЛП-задачи (3) имеет схожую структуру. Основное отличие заключается в выборе и добавлении только одного неравенства, максимизирующего невязку среди невыполненных ограничений. Как показано в [48], этот алгоритм допускает естественное распараллеливание и весьма эффективен на случайных данных.

4.2. Построение диаграммы Вороного и нахождение крайних точек в пространствах фиксированной размерности

Разработанный С.А. Мартишиным [23] быстрый алгоритм нахождения крайних точек множества в D -мерном евклидовом пространстве основан на использовании модифицированного алгоритма Кларксона [40], [48] решения задачи линейного программирования в пространствах фиксированной размерности и использовании некоторых просто проверяемых достаточных условий, позволяющих существенно повысить эффективность алгоритма. Программная реализация и эксперименты показали, что в пространствах размерности до 10–15 крайние точки множеств находятся гораздо эффективнее, чем при использовании стандартных алгоритмов, основанных на построении выпуклой оболочки.

Разработан также новый динамический алгоритм построения и вычисления некоторых

характеристик ячеек диаграммы Вороного на плоскости с ограничениями, весьма эффективный для данных, распределение которых близко к равномерному [24].

5. Эффективная реализация пакета прикладной статистики на конвейерных ЭВМ

Статистические методы решения прикладных задач – от моделирования плазмы до проверки простоты чисел – используют генераторы случайных последовательностей и требуют большого объема вычислений. Поэтому генераторы таких чисел должны работать с максимальной производительностью. Основное препятствие на этом пути – использование точных целочисленных вычислений с многоразрядными числами. В [30], [31] разработаны быстрые алгоритмы генерирования псевдослучайных последовательностей, использующие по возможности минимальное число операций целочисленного умножения.

Разработаны [30] методы повышения эффективности решения задач статистической обработки больших объемов данных и нахождения функции распределения по данным. Предложена схема сдвигания для сложения массивов и исследована ее эффективность при отображении на конвейерные архитектуры.

Важной компонентой большинства статистических пакетов является эффективное вычисление функций распределения случайных величин. В [30] предложен алгоритм быстрого вычисления функции распределения, основанный на таблично-аппроксимационном методе [32].

Предложенные алгоритмы позволили решить задачу эффективной реализации на языке FORTRAN пакета математической статистики на RISK процессоре ALPHA EV-4. В работе [30] описаны методы ускорения программных реализаций алгоритмов, учитывающие архитектурные особенности используемого процессора. Процессор EV-4 предоставляет возможность параллельного использования операций, исполняемых различными функциональными устройствами. Устройства целочисленной и плавающей арифметических операций конвейеризованы. Имеется также быстрая кэш-память. Однако не все арифметические операции конвейеризованы. Среди таких операций – целочисленное умножение и деление чисел с плавающей запятой.

Максимальная производительность программ достигается за счет максимальной загрузки конвейеров функциональных устройств и максимального использования возможностей иерархической памяти (регистров и быстрой кэш-памяти).

В работе [30] описаны результаты оптимизации программ на языке FORTRAN для некоторых задач

математической статистики на процессоре ALPHA EV-4. Приведенные результаты временных тестов для оценки производительности программных реализаций показывают значительный рост производительности оптимизированных программ.

6. Быстрые алгоритмы разложения полинома по последовательностям ортогональных полиномов

Во многих приложениях возникает задача нахождения коэффициентов разложения функции по некоторому базису, если известно разложение этой функции по другому базису. Наиболее важными являются следующие примеры: полиномиальная интерполяция, преобразования Фурье и другие ортогональные преобразования, разложение по собственным векторам ленточных матриц, разложение по собственным функциям циклических операторов.

Все эти проблемы могут быть сведены к проблеме умножения $N \times N$ матрицы на N -мерный вектор. В общем случае для этого необходимо $O(N^2)$ арифметических операций [1]. Однако для некоторых матриц, зависящих от $O(N)$ параметров, существуют быстрые алгоритмы умножения с $O(M \log^a N)$ арифметическими операциями. Такими матрицами являются, например: матрицы дискретного преобразования Фурье, матрицы ортогональных преобразований, циркулянты, теплицевы матрицы, матрицы Вандермонда и матрицы Коши.

Обычно быстрый алгоритм основан на тождествах для умножаемой матрицы. Например, алгоритм быстрого преобразования Фурье основан на некоторых тождествах для матриц Фурье.

Быстрый алгоритм для вычисления значений полинома и значений его производных основан на следующем разложении треугольника Паскаля:

$$\|C_j^i\| = \text{Diag}(n!) \|t_{ij}\| \text{Diag}(n!)^{-1}$$

где $\text{Diag}(n!)$ является диагональной матрицей $\text{Diag}(n!)_{i,j} = i! \delta_{i,j}$, а $\|t_{ij}\|$ является теплицевой матрицей с элементами $t_{ij} = 1/(i-j)!$, если $i \geq j$; $t_{ij} = 0$, если $i < j$.

Напомним, что матрица называется теплицевой, если ее элементы, расположенные на прямых, параллельных главной диагонали, равны.

Быстрый $O(M \log N)$ -алгоритм для умножения теплицевой матрицы на вектор основан на том, что теплицева матрица представляется как произведение двух циркулянтов. Циркулянт называется такой матрицей C , для которой $c_{ij} = c_{pq}$, если $i-j \equiv p-q \pmod N$.

Задача умножения циркулянта C на вектор x эквивалентна вычислению циклической свертки

$$\text{Conv}(c, x) = F_N^{-1}(F_{Nc} \odot F_N x),$$

где \odot – операция покомпонентного умножения векторов.

Быстрый алгоритм разложения по собственным векторам якобиевой матрицы основан на представлении такой матрицы как самосопряженной одноранговой модификации прямой суммы двух якобиевых матриц половинного порядка. Это представление приводит к $O(M \log^3 N)$ -алгоритму умножения вектора на матрицу собственных векторов якобиевой матрицы. Похожее соотношение для самосопряженных ленточных матриц дает быстрый алгоритм умножения вектора на матрицу ее собственных векторов.

Быстрый алгоритм для разложения полинома по ортогональным многочленам следует из упомянутых выше быстрых алгоритмов и из того факта, что такие ортогональные полиномы удовлетворяют трехчленному рекуррентному соотношению и поэтому являются собственными векторами якобиевой матрицы.

Все упомянутые выше задачи являются специальными случаями следующей задачи разложения по полиномиальному базису. Пусть $R[x]$ обозначает векторное пространство полиномов над полем R нулевой характеристики. Тогда любая последовательность полиномов $\{p_n(x)\}$, $n \geq 0$ с условием $\deg p_n(x) = n$ является базисом этого векторного пространства. Задача разложения по полиномиальному базису состоит в следующем: дано разложение полинома $p(x) = \sum a_n p_n(x)$, найти разложение $p(x) = \sum b_n q_n(x)$. Обычно либо $p_n(x) = x^n$, либо $q_n(x) = x^n$.

Последовательность полиномов $\{s_n(x)\}$ с производящей функцией $S(x,t) = \sum s_n(x)t^n$ называется последовательностью Боаса-Бака [38], [58], если и только если $S(x,t) = g(t)\Psi(xf(t))$, где $g(t)$, $\Psi(t)$, $f(t)$ – формальные ряды такие, что $f(0) = 0$, $f'(0) = g(0) = 1$. Этот класс последовательностей полиномов часто называется последовательностями обобщенного типа Аппеля [5], [7], [60].

Многие известные последовательности полиномов являются последовательностями Боаса-Бака [5], [38], [58], [59]. В частности, классические ортогональные полиномы (вместо полиномов Якоби $P_n^{\alpha,\beta}(x)$ необходимо рассмотреть полиномы $P_n^{\alpha,\beta}(x+1)$), полиномы Бесселя, полиномы Бернулли и многие другие имеют производящую функцию Боаса-Бака. Теневой анализ изучает общие свойства последовательностей полиномов Боаса-Бака ([7], [60], [59]). Задача разложения полинома $p(x)$ по последовательности Боаса-Бака $\{s_n(x)\}$ может быть сведена к задаче умножения вектор-строки коэффициентов многочлена на теневую матрицу $M(f(t),g(t))$. Элементами k -го столбца теневой матрицы являются коэффициенты ряда $g(t)f(t)^k$.

Последовательность полиномов Боаса-Бака с $f(t) = t$ называется последовательностью Бренке [38]. Для таких последовательностей теневая матрица является теплицевой. Это позволило

получить результат [28], что сложность разложения данного многочлена по последовательности полиномов Бренке равна $O(M \log N)$.

С использованием того факта, что последовательности полиномов Эрмита и Лагерра являются последовательностями Бренке, в [28] получены быстрые алгоритмы разложения данного полинома по полиномам Эрмита и Лагерра.

В общем случае пока не найден быстрый алгоритм умножения вектор-строки на теневую матрицу. Однако в [28] найден приближенный $O(M \log^2 N)$ -алгоритм умножения теневой матрицы на вектор-столбец, который находит искомое произведение с любой наперед заданной точностью.

7. Алгоритмические вопросы формального анализа программ

В последнее время В.А.Захаровым получен ряд результатов, касающихся применения методов математической логики для анализа поведения параллельных распределенных программных систем.

В [16, 67] рассмотрены две наиболее употребительные семантики параллельных вычислений – семантика чередования и семантика частичного порядка. При использовании логических методов верификации параллельных и распределенных программ множеству вычислений в зависимости от выбора семантики сопоставляется модель (или множество моделей) в одной из неклассических логик – темпоральной логике или логике причинности. Поскольку эффективность алгоритмов анализа спецификаций параллельных вычислений существенно зависит от размера модели, выбор подходящей логики следует осуществить таким образом, чтобы объем рассматриваемой модели был наименьшим. В данных работах исследуются необходимые и достаточные условия, при которых множество вычислений параллельной программы в семантике чередования может быть полностью и однозначно описано единственной моделью меньшего объема в семантике частичного порядка.

В [68] разработан новый подход к проверке выполнимости формул пропозициональной темпоральной логики линейного времени (PLTL). Начиная с основополагающих работ Манни и Пнуели, эта логика широко применяется для спецификации распределенных систем программ. Благодаря значительным выразительным возможностям этой логики многие свойства, описывающие поведение сложных реагирующих программных систем могут быть представлены формулами PLTL. Таким образом, задача верификации взаимодействующих программных систем может быть сведена к проблеме выполнимости в PLTL. В предложенном алгоритме проверка выполнимости формул PLTL осуществляется путем символьной манипуляции с

монотонными дизъюнктивными нормальными формами, используя технику представления ДНФ контактными схемами.

В [17] рассматривается задача проверки выполнимости формул логики ветвящегося времени СТЛ на конечных моделях и предлагается новый оптимальный по быстродействию алгоритм ее решения. Практическая применимость алгоритма продемонстрирована на примере анализа системы взаимодействующих процессов, осуществляющих сортировку.

В.А. Захарову в серии работ [12, 13, 14, 15, 16, 67, 68, 17] удалось разработать общий подход к построению эффективных (в том числе полиномиальных) алгоритмов разрешения проблемы эквивалентности в различных классах программ.

Проблема эквивалентности программ явно или опосредованно возникает при решении многих задач системного программирования, к числу которых относятся задачи трансляции, оптимизации и верификации вычислительных программ, разработка методов частичных вычислений, задачи специализации и повторного использования программ и др. Всякий раз, когда возникает необходимость изменить синтаксическую составляющую вычислительной программы (текст программы), сохраняя при этом в той или иной мере ее семантическую компоненту (поведение, или функционирование), приходится иметь дело с отношением эквивалентности на множестве программ и, следовательно, с проблемой эквивалентности программ. Необходимость исследования свойства программных вычислений, наряду с проблемами синтаксического анализа и перевода, послужила источником многочисленных моделей вычислений, получивших широкое распространение и применение в современном программировании.

Наиболее общая формулировка проблемы эквивалентности такова. Для произвольной пары программ в заданной модели вычислений требуется выяснить, обладают ли эти программы одинаковым поведением. Уточняя понятия вычислительной модели, программы и ее поведения, мы получаем многочисленные вариации проблемы эквивалентности.

Известно, что проблема эквивалентности программ неразрешима в тех случаях, когда модель вычислений является универсальной. В то же время, начиная с основополагающей работы Ю.И. Янова, удалось выделить обширные классы программ с разрешимой проблемой эквивалентности. В течение долгого времени основные усилия в этой области были сосредоточены на выявлении самого факта алгоритмической разрешимости эквивалентности программ. Подавляющее большинство полученных разрешающих процедур имели высокую сложность

и были мало пригодны для практического применения.

Суть предложенного В.А. Захаровым подхода такова. Заданной вычислительной модели программ сопоставляется специальная полугруппа – критериальная система. С ее помощью для любой пары анализируемых программ можно построить систему переходов, описывающую всевозможные пары совместных вычислений этих программ. При этом для подтверждения эквивалентности программ необходимо и достаточно убедиться в том, что ни одна из особо выделенных опровергающих вершин не достижима из начального состояния. Оказывается также, что для проверки этого условия можно ограничиться анализом лишь некоторого начального фрагмента построенной системы переходов, размер которого полиномиально зависит от размеров исследуемых программ. В тех случаях, когда всякий начальный фрагмент системы переходов может быть построен эффективно, мы получаем удобный и быстрый алгоритм, разрешающий эквивалентность программ. При помощи указанного подхода удалось построить полиномиальные по сложности алгоритмы, разрешающие эквивалентность в отдельных классах пропозициональных операторных программ с перестановочными [12, 14, 15, 67] и частично-перестановочными [15, 16], [67] операторами, унарных рекурсивных программ [13], первопорядковых операторных программ [67].

В заключение авторы благодарят С.А. Фомина за помощь и многочисленные полезные замечания при написании этой статьи. Работа выполнена при поддержке РФФИ, проекты 99-01-00210, 99-01-00211.

Литература

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
2. Бухштабер В.М., Холодов А.Н. Группы формальных диффеоморфизмов суперпрямой, производящие функции для последовательностей полиномов и функциональные уравнения. Изв. АН СССР. Сер. Матем. 53(5) (1989) 944-970.
3. Бухштабер В.М., Холодов А.Н. Структуры Боаса-Бака на последовательностях полиномов. Функциональный анализ. 23(4) (1989) 11-23.
4. Бэйтмен Г., Эрдейи А. Высшие трансцендентные функции. Т. 2. М.: Наука, 1974.
5. Бэйтмен Г., Эрдейи А. Высшие трансцендентные функции. Т. 3. М.: Наука, 1967.
6. Вентцель Е.С., Исследование операций. Задачи, принципы, методология, М., Наука, 1988.
7. Висков О.В. Операторная характеристика обобщенных полиномов Аппеля. ДАН СССР. 225 (1975) 749-752.
8. Гэри М., Джонсон Д.С. Вычислительные машины и труднорешаемые задачи, М.: Мир, 1982.
9. Журавлев Ю.И. О невозможности построения минимальных дизъюнктивных нормальных форм функций алгебры логики в одном классе алгоритмов, ДАН СССР, 1960, т. 132, N 3, С. 504-506.

10. Журавлев Ю.И. Локальные алгоритмы вычисления информации. Кибернетика. 1965, N 1, С. 12-19.
11. Журавлев Ю.И. Теоретико-множественные методы алгебры логики. Проблемы кибернетики. вып. 8, М.: Физматгиз, 1962, С. 5-44.
12. Захаров В.А. О проблеме эквивалентности пропозициональных программ над полугруппами, Kurosh Algebraic Conference'98, Abstracts of Talks, 1998, с.170-172.
13. Захаров В.А., Подловченко Р.И., Полиномиальный по сложности алгоритм, распознающий коммутативную эквивалентность схем программ, Доклады РАН, 1998, т.362, N 6.
14. Захаров В.А. Быстрые алгоритмы разрешения эквивалентности операторных программ на уравновешенных шкалах, Математические вопросы кибернетики, Физматлит, 1998, вып.7.
15. Захаров В.А. Быстрые алгоритмы разрешения эквивалентности пропозициональных операторных программ на упорядоченных полугрупповых шкалах, Вестник Московского Университета, сер.15, Вычислительная математика и кибернетика, 1999, N 2.
16. Захаров В.А., Спанопуло В.В. О взаимосвязи двух семантик параллельных вычислений, Программирование, РАН, 1997, N 5, с.36-48.
17. Захаров В.А., Царьков Д.В. Эффективные алгоритмы проверки выполнимости формул темпоральной логики STL на модели и их применение для верификации параллельных программ, Программирование, РАН, 1998, N 5.
18. Кузюрин Н.Н. Асимптотически точные полиномиальные алгоритмы в целочисленном линейном программировании, Дискретная математика, 1989, т. 1, N 2, С. 78-85.
19. Кузюрин Н.Н. Метрические аспекты теории целочисленного линейного программирования, Дискретная математика, 1994, т. 4, N 6, С. 499-518.
20. Кузюрин Н.Н. Распределенное принятие решений, Сб. Методы комбинаторной оптимизации, М., ВЦ РАН, 1997, С. 30-36.
21. Кузюрин Н.Н. Оценки оптимумов задач ЦЛП типа покрытия, Дискретные модели и методы, вып. 2, М.: ВЦ РАН, 1996.
22. Кузюрин Н.Н. Коммуникационно-эффективные алгоритмы решения задач линейного и целочисленного программирования, Вопросы кибернетики. Приложения системного программирования, М., ВИНТИ, 1998, вып. 4, С. 162-174.
23. Мартишин С.А. Алгоритм нахождения крайних точек множества в D-мерном евклидовом пространстве с использованием линейного программирования, Вопросы кибернетики. Приложения системного программирования, вып. 1, М.: ВИНТИ, 1995.
24. Мартишин С.А. Динамический алгоритм вычисления некоторых характеристик ячеек диаграммы Вороного на плоскости с ограничениями, Вопросы кибернетики. Приложения системного программирования, вып. 3, М.:ВИНТИ, 1997, С. 240-257.
25. Нестеров Ю.Е. Метод линейного программирования с трудоемкостью $O(n^3L)$ операций, Экономика и математические методы, 1988, т. 24, N 16, С. 174-176.
26. Схрейвер А. Теория линейного и целочисленного программирования (в двух томах), М.: Мир, 1991.
27. Фомин С.А. Исследование эффективности реализации нового приближенного алгоритма для задачи положительного линейного программирования, Вопросы кибернетики. Приложения системного программирования, М.: ВИНТИ, вып. 4, 1998, С. 175-185.
28. Фрумкин М.А., Холодов А.Н. Об алгоритмах разложения данного полинома по последовательностям полиномов Боаса-Бака, Вопросы кибернетики. Приложения системного программирования, вып. 4, М.: ВИНТИ, 1998, С. 144-161.
29. Хачиян Л.Г. Полиномиальный алгоритм в линейном программировании, ДАН СССР, 1979, т. 244, С. 1093-1096.
30. Шокуров А.В. Об эффективной реализации статистических программ на конвейерных ЭВМ, Вопросы кибернетики. Приложения системного программирования, вып. 4, М.: ВИНТИ, 1998, С. 119-143.
31. Шокуров А.В., Захарова Н.И. Генерирование случайных последовательностей на векторно-конвейерной ЭВМ, Вопросы кибернетики. Базовое программное обеспечение суперЭВМ, М.: ВИНТИ, 1990, С. 59-78.
32. Шокуров А.В. Вычисление функций на векторно-конвейерной ЭВМ, Кибернетика и вычислительная техника. вып.4, М.: Наука, 1988, С. 7-27.
33. Alizadeh F. Interior point methods in semidefinite programming with applications to combinatorial optimization, SIAM Journal Optimization, 1995, v. 5, N 1, 13-51.
34. Alon N., Spencer J.H. The probabilistic method. John Wiley and Sons inc., New York, 1992.
35. Asratian A.S., Kuzjurin N.N. Approximations of packing-covering integer programs via linear relaxations, Research Report 1998-18, Department of Mathematics, Lulea University of Technology.
36. Awerbuch B., Azar Y. Local optimization of global objectives: competitive distributed deadlock resolution and resource allocation, 26th ACM STOC-94, 1994, 240-249.
37. Bartal Y., Byers J.W., Raz D. Global optimization using local information with applications to flow control, 38th IEEE FOCS-97, 1997, 303-312.
38. Boas R.P. and Buck R.C. Polynomial expansion of analytic functions. Berlin: Springer, 1958.
39. Chihara T.S. Orthogonal polynomials with Brenke type generating functions. Duke Math. J. 35(3) (1969) 505-517.
40. Clarkson K.L. Las Vegas algorithm for linear programming when the dimension is small. Proceedings of Ann. 29th IEEE FOCS, 1988, pages 452-457.
41. Cormen T.H., Leiserson C.E., Rivest R.L. Introduction to Algorithms, The MIT Electrical Engineering and Computer Science Series, 1997.
42. Goodrich M.T., Communication-Efficient Parallel Sorting, Proc.30th ACM Symposium on Theory of Computing (STOC), 1996.
43. Hayman J.M., Lazar A.A., Pacifici G. Joint scheduling and admission control for ATS-based switching nodes, Proc. ACM SIGCOMM, 1992, 223-234.
44. Jabotinsky E. Analytic iteration. Trans. Amer. Math. Soc. 108 (1963) 457-477.
45. Karmarkar N. A new polynomial-time algorithm for linear programming, Combinatorica, 1984, v. 4, p. 373-395.

46. Klee V., Minty G.J. How good is the simplex algorithm? Inequalities. III, Academic Press, New York, 1972, p. 159-175.
47. Kushilevitz E., Nisan N. Communication Complexity, Cambridge University Press, 189 pp., 1997.
48. Kuzjurin N.N., Shabanov L.V. A parallel algorithm for fixed-dimensional linear programming, Parallel Algorithms and Applications, 1995, v. 5, N 34, P. 17-24.
49. Kuzjurin N.N. On the difference between asymptotically good packings and coverings. – European J. Comb. 16 (1995) 35 – 40.
50. Kuzjurin N.N. Locally explicit constructions of Rodl's asymptotically good packings, Lecture Notes in Computer Science, 1998, v. 1450, Proc. MFCS-98, Brno, Czech Republic, August 1998, pp. 194-202.
51. Jan van Leeuwen, editor. Handbook of Theoretical Computer Science, vol. A. Algorithms and Complexity. Elsevier Science Publishers B.V. and The MIT Press, 1990.
52. Luby M., Nisan N. A parallel approximation algorithm for positive linear programming, Proc. 5th ACM-STOC, 1993, 448-457.
53. Motwani R., Raghavan P. Randomized Algorithms, Cambridge University Press, 1995.
54. Naor M., Shulman L.J., Srinivasan A. Splitters and near-optimal derandomization, Proc. 36th Ann. IEEE FOCS, 1995, 182-191.
55. Papadimitriou C.H., Yannakakis M. Linear Programming Without the Matrix, Proc. 25th ACM STOC-93, 1993, 121-129.
56. Raghavan P., Tompson C.D. Randomized rounding: a technique for provably good algorithms and algorithmic proofs, Combinatorica, 37 (1987) 365-374.
57. Raghavan P. Probabilistic constructions of deterministic algorithms: approximating packing integer programs. J. Comput. and Syst. Sci., 1988, v. 37, N 4, pp. 130-143.
58. Rainville D.E. Special functions. Macmillan Co. N.-Y. 1960.
59. Roman S. The umbral calculus. N.-Y. A.P., 1984.
60. Roman S. The theory of the umbral calculus, I. J. Math. Anal. Appl. 87 (1982) 58-115.
61. Shor P.W. Algorithms for quantum computations: discrete logarithm and factoring, Proceedings of the 35th IEEE FOCS, 1994, pp. 124-134.
62. Sparlinski I. Approximate constructions in finite fields, In Finite Fields and Applications, London Math. Soc., Lect. Notes Ser., v. 233, Cambridge Univ. Press, Cambridge, 1996, 313-332.
63. Sparlinski I. Finding irreducible and primitive polynomials, Appl. Algebra in Engin., Commun. and Computing, 4 (1993) 263-268.
64. Srinivasan A. Improved Approximation Guarantees for Packing and Covering Integer Programs, Proc. ACM Symposium on Theory of Computing, 1995, 268-276.
65. Srivastav A., Stangier P. Algorithmic Chernoff-Hoeffding inequalities in integer programming, Random Structures and Algorithms, 8, N 1 (1996) 27-58.
66. Plotkin S., Shmoys D.B., Tardos E. Fast approximation algorithms for fractional packings and covering problems, Proc. 32nd Annual Symp. IEEE Found. on Computer Sci., 1991, 495-504.
67. Zakharov V.A., Spanopulo V.V. To the Relationship between the Interleaving and Causal Models of Parallel Computations, Advances in Modal Logic'96, CSLI Publications, 1997, pp. 221-232.
68. Zakharov V.A. On the verification of PLTL formulae by means of monotone disjunctive normal forms, Lecture Notes in Computer Science, Springer-Verlag, 1997, v.1234, p.419-429.