На правах рукописи

Назарьянц Елена Геворговна

ПРЕОБРАЗОВАНИЯ ГРАФОВ И КОМБИНАТОРНЫХ СОЧЕТАНИЙ НА ОСНОВЕ ВИДОИЗМЕНЕНИЯ ФОРМУЛ ВИЕТА И АЛГОРИТМОВ СОРТИРОВКИ С МИНИМИЗАЦИЕЙ ВРЕМЕННОЙ СЛОЖНОСТИ В ПРИЛОЖЕНИИ К ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

Специальность:
05.13.17 – Теоретические основы информатики

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Таганрог – 2018

Работа выполнена в Таганрогском институте имени А.П. Чехова (филиал) ФГБОУ ВО "Ростовский государственный экономический университет (РИНХ)"

Научный руководитель:	доктор технических наук, профессор **Ромм Яков Евсеевич**
Официальные оппоненты:	доктор технических наук, профессор **Ерусалимский Яков Михайлович**
	доктор технических наук, профессор **Бутакова Мария Александровна**
Ведущая организация:	**ФГАНУ НИИ «СПЕЦВУЗАВТОМАТИКА»**

Защита состоится « 27 » июня 2018 г. в ____на заседании диссертационного совета Д 999.065.02 созданного на базе ФГАОУ ВО "ЮФУ" и ФГБОУ ВО ЮРГПУ (НПИ) им. М.И. Платова по адресу: 347928, г. Таганрог, пер. Некрасовский, 44, ауд. Д- 406.

С диссертацией можно ознакомиться в Зональной научной библиотеке Южного федерального университета по адресу: 344000, г. Ростов-на-Дону, ул. Зорге, 21 ж и на сайте:

Автореферат разослан «   »   2018 г.

Ученый секретарь
диссертационного совета Д 999.065.02
доктор технических наук, профессор   Целых А.Н.

Актуальность темы исследований. Решения задач комбинаторной оптимизации широко применяются в информатике, технике, логистике, криптографии, экономике, планировании. Несмотря на большое количество результатов в этой области, потребность в дальнейших исследованиях не уменьшается. Это связано с постоянным появлением новых задач, а также со сложностью их решения. Наиболее важные задачи комбинаторной оптимизации относят к NP-трудным, построение их оптимального решения требует значительных временных затрат. Возникает необходимость в глубоком анализе алгоритмов решения таких задач методами теории сложности, в оценках перспективы разработки алгоритмов с заданными характеристиками (время вычислений, требуемое количество процессоров, точность решения и др.), чтобы на этой основе правильно построить и выбрать алгоритм. Для таких задач наиболее актуальна проблема нахождения оптимального решения за минимально возможное время. Большой вклад в развитие методов решения комбинаторных задач внесен отечественными учеными, в числе которых Бабаев Ф.В., Вайнштейн А.Д., Валеева А.Ф., Гладков Л.А., Курейчик В.В., Курейчик В.М., Колпаков Р.М., Посыпкин М.А., Сигал И.Х., Кузюрин Н.Н., Мухачева Э.А, а также зарубежными исследователями, среди которых Dyckhoff H., Festa P., Resende M.G.C., Laporte G., Martello S., Zhao X., Shen H.

Во всех случаях существующие алгоритмы дискретной оптимизации требуют ускоренной вычислительной обработки, тем более актуальной, чем больше значение имеет количество входных данных. Большинство известных методов являются приближенными, а точные методы не дают гарантированно оптимального решения. При этом создание методов ускорения, в частности на основе параллельных алгоритмов преобразования графов задач дискретной оптимизации, не является окончательно или во многих случаях приемлемо решенной задачей. В частности, это относится к алгоритмам преобразования графов применительно к задачам коммивояжера, а также одномерной и двумерной упаковки. Предложенные решения будут связаны между собой единообразной алгоритмической основой преобразования графов и комбинаторных сочетаний с минимизацией временной сложности.

Целью диссертационной работы является разработка и исследование детерминированных алгоритмов преобразования графов и комбинаторных сочетаний в приложении к задачам дискретной оптимизации с минимизацией временной сложности их решения.

Для достижения цели в диссертационной работе решаются следующие **задачи**:

1. Разработать алгоритм преобразования матричной формы графа, представляющего метод ветвей и границ, который на основе устойчивой адресной сортировки снижал бы временную сложность решения задачи коммивояжера до значения оценки полиномиального вида.

2. Разработать алгоритм преобразования графа метода ветвей и границ с обработкой одновременно нескольких ветвей с целью снижения временной сложности решения задачи коммивояжера.

3. Представить алгоритм генерации комбинаторных сочетаний на основе модификации формул Виета восстановления коэффициентов многочлена по его действительным корням с использованием устойчивой адресной сортировки. Применительно к задаче об одномерном булевом рюкзаке разработать на этой основе детерминированные алгоритмы с линейными и логарифмическими оценками временной сложности решения, достигаемыми ценой роста числа компонентов оборудования.

4. На основе устойчивой адресной сортировки и видоизменения формул Виета для случая комплексных корней многочлена предложить алгоритм генерации комбинаторных сочетаний, который применительно к двумерной упаковке позволял бы достигать локальной оптимальности с полиномиальной оценкой временной сложности.

5. На этой же основе представить разновидности детерминированных алгоритмов двумерной упаковки, достигающих глобального оптимума с линейной оценкой временной сложности и оценить рост оборудования.

6. Выполнить численный и программный эксперимент с целью верификации правильности построения алгоритмов.

Методы исследования включают элементы теории графов и комбинаторного анализа, синтез и анализ параллельных алгоритмов упорядочения и обработки данных, элементы теории сложности.

Достоверность полученных результатов вытекает из конструктивного построения формализованных алгоритмов, корректного математического обоснования с применением комбинаторного анализа, оценок временной сложности с учетом количества компонентов вычислительной системы, подтверждается результатами программного и численного эксперимента.

Научная новизна. В диссертационной работе получены следующие новые научные результаты:

1. Предложен алгоритм преобразования матричной формы графа метода ветвей и границ на основе устойчивой адресной сортировки, отличающийся от известных аналогов сокращением количества последовательных шагов, что позволяет реализовать определение нижней границы каждой ветви с единичной оценкой временной сложности. Полностью метод ветвей и границ реализуется с полиномиальной оценкой количества последовательных шагов за счет роста числа требуемых параллельных компонентов вычислительной системы, что позволяет получать решение NP-сложной задачи за допустимое время ценой роста затрат оборудования (С. 53 – 61).

2. Для случая преобразования графа метода ветвей и границ с обработкой одновременно нескольких ветвей временная сложность решения задачи коммивояжера оценивается из соотношения $T\left(\frac{n^6}{6}\right)=O\left(n^5\log_2 n\right)$. Численное моделирование предложенного алгоритма подтверждает правильность его работы и оценку числа шагов (С. 61 – 75).

3. Для генерации комбинаторных сочетаний предложены модификации формул Виета восстановления коэффициентов многочлена по его корням. На

данной основе с использованием устойчивой адресной сортировки разработаны детерминированные алгоритмы решения задачи об одномерном булевом рюкзаке, которые отличаются от известных по структуре и позволяют достигать оптимума с линейной и логарифмической временной сложностью ценой экспоненциального роста оборудования (С. 80 –91, 112 – 118).

4. На основе устойчивой адресной сортировки и видоизменения формул Виета применительно к генерации сочетаний разработан детерминированный алгоритм двумерной упаковки. Алгоритм отличается от известных аналогов структурой и не более чем линейным количеством последовательных шагов, что позволяет достигать локальной оптимальности на каждом последовательном шаге при экспоненциальном росте оборудования (С. 128 – 147).

5. На этой же основе даны разновидности детерминированных алгоритмов двумерной упаковки, одна из которых позволяет достигать глобального оптимума с линейной временной сложностью за счет экспоненциального с квадратичным множителем количества задействованных компонентов вычислительной системы, что отличает алгоритмы от известных и позволяет выполнить двумерную упаковку с линейной оценкой времени ценой существенного роста оборудования (С. 147 – 154).

Основные положения, выносимые на защиту

1. Предложен алгоритм преобразования матричной формы графа метода ветвей и границ на основе устойчивой адресной сортировки, позволяющий реализовать определение нижней границы каждой ветви с единичной оценкой временной сложности. Полностью метод ветвей и границ реализуется с полиномиальной оценкой количества последовательных шагов за счет роста числа требуемых параллельных компонентов вычислительной системы.

2. Для преобразования графа метода ветвей и границ с обработкой одновременно нескольких ветвей временная сложность решения задачи коммивояжера оценивается из соотношения $T\left(\frac{n^6}{6}\right)=O\left(n^5\log_2 n\right)$. Численное моделирование предложенного алгоритма подтверждает правильность его работы.

3. Для генерации комбинаторных сочетаний предложены модификации формул Виета восстановления коэффициентов многочлена по его корням. На данной основе с использованием устойчивой адресной сортировки разработаны детерминированные алгоритмы решения задачи об одномерном булевом рюкзаке, которые позволяют достигать оптимума с линейной и логарифмической временной сложностью ценой экспоненциального роста оборудования.

4. На основе устойчивой адресной сортировки и видоизменения формул Виета применительно к генерации сочетаний разработан детерминированный алгоритм двумерной упаковки, достигающий локальной оптимальности на каждом последовательном шаге при экспоненциальном росте оборудования.

5. На этой же основе предложены разновидности детерминированных алгоритмов двумерной упаковки, одна из которых позволяет достигать глобального оптимума с линейной временной сложностью за счет экспоненциального с

квадратичным множителем количества задействованных компонентов вычислительной системы.

Теоретическая значимость работы заключается в преобразовании графов и комбинаторных сочетаний на основе видоизменения формул Виета, алгоритмов сортировки и в создании на этой основе детерминированных алгоритмов решения NP-трудных задач комбинаторной оптимизации с полиномиальными, линейными и логарифмическими оценками временной сложности.

Практическая ценность исследования заключается в том, что на основе разработанных алгоритмов достигается глобально оптимальное решение практических задач комбинаторной оптимизации. В частности, это относится к задачам логистики. Кроме того, разработанная программа решения задачи коммивояжера может быть использована в навигационных системах и системах маршрутизации. Предложенное решение задачи одномерной булевой упаковки применимо для построения шифров криптографии. Предложенная двумерная упаковка практически пригодна для погрузки контейнеров с целью транспортировки, а также для оптимизации раскроя при стандартизированной штамповке. При этом разработанные алгоритмы решают задачи за реально достижимое, практически приемлемое время.

Внедрение и использование результатов работы. Следующие результаты диссертационной работы приняты к использованию в АО «Научно-производственный комплекс «Бортовые интеллектуальные системы»»: алгоритм решения задачи коммивояжера на основе метода ветвей и границ с применением устойчивой адресной сортировки, имеющий полиномиальную временную сложность, достигаемую за счет роста вычислительных ресурсов; детерминированные алгоритмы решения задачи об одномерном булевом рюкзаке с логарифмической временной сложностью за счет роста числа процессоров; детерминированные алгоритмы двумерной упаковки на основе устойчивой адресной сортировки и видоизменения формул Виета с полиномиальной временной сложностью при экспоненциальном росте оборудования. Помимо того, результаты диссертационной работы использованы в учебном процессе на кафедре информатики Таганрогского института имени А.П. Чехова (филиала) «Ростовского государственного экономического университета (РИНХ)»: в качестве учебного материала в лекционных курсах и при проведении лабораторных работ по дисциплинам «Теория алгоритмов», «Исследование операций и методы оптимизации»; «Программирование», «Основы математической обработки информации», «Методы и средства защиты информации» Использование подтверждено соответствующими актами об использовании, приведенными в приложении к диссертационной работе.

Апробация работы. Основные положения и результаты диссертационной работы докладывались и обсуждались на конференциях: XXVII Международная конференция: «Актуальные проблемы в современной науке и пути их решения» // Евразийский Союз Ученых (ЕСУ) (г.Москва, 2016); Международная научно-практическая конференция «Информационные технологии: актуальные вопросы, перспективы и инновации» (г.Пенза, 2016); XIII Международная научно-техническая конференция «Новые информационные технологии и системы» (Пенза, НИТиС-2016); XVII Всероссийский симпозиум по прикладной и промышленной математике (осенняя открытая сессия) (г. Сочи–Дагомыс, 2016; Всероссийская научно-практическая конференция с международным участием «Аспекты развития науки, образования и модернизации промышленности» (Политехнический институт (филиал) ДГТУ в г. Таганроге, 2017).

Публикации. По теме диссертационной работы опубликовано 12 печатных работ общим объемом около 12,75 печатных листов, в том числе 3 статьи в рецензируемых журналах, рекомендуемых ВАК РФ для опубликования материалов кандидатских диссертаций по специальности 05.13.17.

Личный вклад автора. В совместно опубликованных работах Назарьянц Е.Г. выполнила построение параллельных алгоритмов решения некоторых задач комбинаторной оптимизации, в работах приведен анализ предложенных алгоритмов, вычислена временная сложность их. Представлено численное моделирование предложенного алгоритма решения задачи коммивояжера.

Структура и объем работы. Диссертационная работа состоит из введения, 3 глав основного раздела, заключения, списка литературы и приложения. Основное содержание работы изложено на 197 страницах, включая 7 таблиц, 10 рисунков и список литературы из 216 наименований.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обоснована актуальность темы диссертационного исследования, представлен краткий обзор научных работ, посвященных задачам дискретной и комбинаторной оптимизации, охарактеризовано современное состояние вопроса. На основе обзора с учетом нерешенных задач определены цель и задачи исследования, изложена научная новизна, сформулированы основные положения, выносимые на защиту, охарактеризована структура диссертации.

В **первой главе** предложен алгоритм преобразования матричного представления графа решения задачи коммивояжера на основе метода ветвей и границ с полиномиальной оценкой временной сложности, которая достигается с помощью применения сортировки. Обсуждаются разновидности этого алгоритма и их временные оценки. Предварительно выполнен краткий обзор различных модификаций постановок задачи и методов ее решения. В главе задача коммивояжера рассматривается в наиболее простой постановке: коммивояжер должен выйти из первого города, посетить по одному разу в неизвестном порядке города $2, 3, 4, ..., n$ и вернуться в первый город; расстояния между всеми городами известны; в каком порядке следует обходить города, чтобы замкнутый путь коммивояжера был кратчайшим?

Применительно к решению задачи на основе метода ветвей и границ используется определение нижних границ подмножеств и разбиения множества маршрутов на подмножества (ветвление) с помощью матрицы, задающей граф всевозможных путей:

	x_1	x_2	…	x_j	…	x_n
x_1	∞	a_{12}	…	a_{1j}		a_{1n}
x_2	a_{21}	∞	…	a_{2j}		a_{2n}
…	…	…	…	…	…	…
x_n	a_{n1}	a_{n2}	…	a_{nj}	…	∞

$x_1, x_2, ..., x_n$ – вершины; $a_{ij} \geq 0$ – длина дуги; ∞ – обозначается путь из вершины x_k в x_k, $1 \leq k \leq n$. Алгоритм решения разбит на три этапа, которые воспроизводятся в цикле с логарифмическим числом шагов, –

1. *Нахождение нижней границы длин всевозможных маршрутов*. Основная операция нахождения минимального элемента выполняется с помощью максимально параллельного варианта устойчивой адресной сортировки подсчетом. Остальные операции преобразования матрицы к форме приведенной по строкам и столбцам основаны на естественном параллелизме. Для вычисления константы приведения применяется схема сдваивания. Константа выбирается в качестве нижней границы длины маршрутов.

2. *Разбиение множества маршрутов на подмножества*. Для выделения претендентов на включение в множество дуг, по которым производится ветвление, в приведенной матрице с помощью той же сортировки идентифицируются все нулевые элементы. Сортировка в явном виде задает взаимно однозначное соответствие между индексами входных и выходных элементов, на этой основе с сохранением максимального параллелизма идентифицируются нули с их индексами, выполняется поиск минимальных элементов, не совпадающих с нулями на пересечении. Максимальная степень нуля определяется также с помощью сортировки: для нахождения минимальных элементов в строке и в столбце, содержащих нулевой элемент, применяется рассматриваемая сортировка, после этого минимальные элементы суммируются. Эти операции производятся для всех найденных нулей одновременно. После определения всех степеней Θ_{ij} нулевых элементов этой матрицы, с применением той же сортировки находится максимальная степень нуля. Фиксируется маршрут, которому принадлежат дуги с максимальной степенью нуля. Этот маршрут с наибольшей вероятностью (предположительно) является искомым.

3. *Исключение маршрутов*. Для полученной матрицы маршрутов, включающей дугу (i, j), вычеркивается строка i и столбец j, и, чтобы не допустить образования цикла в маршруте, заменяется элемент, замыкающий текущую цепочку, на бесконечность (∞). С учетом симметрии множество всех маршрутов, не включающих дугу (i, j), получается заменой элемента a_{ji} на ∞.

4. *Проверка маршрутов* на минимальность и повтор предыдущих этапов воспроизводится до тех пор, пока не будет найдет минимальный маршрут.

Охарактеризованный конструктивный алгоритм решения задачи коммивояжера с матричным представлением графа пути и применением сортировки, в зависимости от разновидности, достигает временной сложности, представленной следующими двумя теоремами.

Теорема 1. Временная сложность параллельного алгоритма решения задачи коммивояжера на основе метода ветвей и границ, без учета возвратов к оборванным ветвям, составляет $T\left(\frac{n^4}{6}-\frac{n^3}{4}\right)=O\left(n^5\log_2 n\right)$, с учетом возвратов – $T\left(\frac{n^4}{6}-\frac{n^3}{4}\right)=O\left(n^7\log_2 n\right)$.

Здесь и в дальнейшем в скобках слева – требуемое количество процессоров.

Теорема 2. Временная сложность того же алгоритма при обработке одновременно нескольких ветвей составляет $T\left(\frac{n^6}{6}\right)=O\left(n^5\log_2 n\right)$.

Оценки выполняются на модели неветвящихся параллельных программ без учета обмена. Доказательства строятся на основе детализированного описания шагов в конструктивной форме, подсчета числа операций на шаге и учета количества шагов алгоритма. Более сложный случай учета всех оборванных ветвей в главе не рассматривается. Выполнено численное моделирование данного алгоритма в последовательной интерпретации на языке Object Pascal в среде Delphi. Результаты моделирования непосредственно совпадают с эталонными решениями на всех рассмотренных примерах. Пусть, например, входные данные заданы таблицей расстояний:

i, j	1	2	3	4	5
1	∞	90	80	40	100
2	60	∞	40	50	70
3	50	30	∞	60	20
4	10	70	20	∞	50
5	20	40	50	20	∞

Результат решения вручную: $(1,4)$, $(4,3)$, $(3,5)$, $(5,2)$, $(2,1)$. Длина оптимального маршрута равна $f=180$. Результат программного решения: $(1,4)$, $(4,3)$, $(3,2)$, $(2,5)$, $(5,1)$. Длина оптимального маршрута снова $f=180$. При этом сами пути могут иметь различия.

Таким образом, предложен алгоритм решения задачи коммивояжёра методом ветвей и границ, который основан на взаимной независимости операций на каждом этапе нахождения нижней границы. В итоге достигаются оценки временной сложности полиномиального вида за счет роста числа процессоров.

Во второй главе рассматривается задача об одномерном булевом рюкзаке (кратко – одномерной упаковки). Приведен обзор постановок и методов решения задачи, рассматривается практическое применение в информационных системах. Затем предложен детерминированный алгоритм на основе модификации формул Виета для выражения коэффициентов многочлена по его корням с применением той же сортировки, которая использовалась в гл. 1 . Алгоритм и его разновидности решают задачу о булевом рюкзаке в следующей постановке. Дано конечное множество предметов $Q=\{q_1,q_2,...q_n\}$, для каждого q_i определена стоимость p_i и вес w_i, нужно максимизировать целевую функцию $\sum_{i=1}^{n} p_i \cdot x_i$ ($\sum_{i=1}^{n} p_i \cdot x_i \to \max$) при ограничении $\sum_{i=1}^{n} w_i \cdot x_i \le w$, где w – вместимость рюкзака, $x_i=1$, если предмет взят для загрузки и $x_i=0$, если он не взят. Формальная запись:

$$\begin{cases} f(x)=\sum\limits_{i=1}^{n} p_i x_i \to \max, \\ \sum\limits_{i=1}^{n} w_i x_i \le w, \\ x_i \in \{0,1\}, 1 \le i \le n. \end{cases}$$

Алгоритм воспроизводит структуру матричной формулы восстановления коэффициентов многочлена по его нулям, с точностью до выполнения операций, – сами арифметические операции при этом выполнять не потребуется. Матричное представление коэффициентов многочлена $P_n(x)=d_n x^n + d_{n-1}x^{n-1} + d_{n-2}x^{n-2} + \ldots + d_1 x + d_0$ записывается в виде:

$$\begin{pmatrix} d_n \\ d_{n-1} \\ \vdots \\ d_0 \end{pmatrix} = \left(\underbrace{ \left.\begin{matrix} 1 & 0 \\ x_{n-1} & 1 \\ 0 & x_{n-1} \\ 0 & 0 \\ \dots & \dots \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix}\right\} n+1 \quad \cdots \quad \begin{matrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 0 \\ x_{n-1} & 1 \\ 0 & x_{n-1} \end{matrix} }_{n} \right) \times \left(\underbrace{ \left.\begin{matrix} 1 & 0 \\ x_{n-2} & 1 \\ 0 & x_{n-2} \\ 0 & 0 \\ \dots & \dots \\ 0 & 0 \\ 0 & 0 \end{matrix}\right\} n \quad \cdots \quad \begin{matrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 0 \\ x_{n-2} & 1 \\ 0 & x_{n-2} \end{matrix} }_{n-1} \right) \times \ldots \times \begin{pmatrix} 1 & 0 \\ x_1 & 1 \\ 0 & x_1 \end{pmatrix} \times \begin{pmatrix} 1 \\ x_0 \end{pmatrix}, \qquad (1)$$

где x_i – i-й корень многочлена, $i = 0,1,\ldots,n-1$, при этом $d_n=1$. Собственно формула Виета для коэффициентов многочлена имеет вид:

$$\left. \begin{array}{l} d_n = 1, \\ d_{n-1} = -(x_0 + x_1 + x_2 + ... + x_{n-1}), \\ d_{n-2} = (x_0 \cdot x_1) + (x_0 \cdot x_2) + ... + (x_0 \cdot x_{n-1}) + ... + (x_{n-2} \cdot x_{n-1}), \\ d_{n-3} = -(x_0 \cdot x_1 \cdot x_2 + x_0 \cdot x_1 \cdot x_3 + ... + x_{n-3} \cdot x_{n-2} \cdot x_{n-1}), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ d_{n-l} = (-1)^{n-l} \cdot (x_0 \cdot x_1 \cdot x_2 \cdot x_3 ... \cdot x_{l-1} + ... + x_{n-l-1} \cdot ... \cdot x_{n-1}), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ d_0 = (-1)^n \cdot (x_0 \cdot x_1 \cdot x_2 \cdot ... \cdot x_{n-2} \cdot x_{n-1}). \end{array} \right\} \qquad (2)$$

Левые части формул (1), (2) одинаковы, следовательно, равны правые части. При этом в правых частях (2) – всевозможные сочетания корней, сочетания не повторяются независимо от порядка расположения корней в сочетании. Отсюда формула (1) может служить алгоритмом генерации всех возможных сочетаний корней, если не принимать во внимание операции умножения и знаки сумм. С этими оговорками веса предметов в задаче о рюкзаке можно интерпретировать как корни многочленов, тогда формула (2), аналогично, формула (1) даст все возможные сочетания весов из n по m. При этом требуется не произведение всех весов, а их сумма, и не нужен знак слагаемых. В итоговой записи наборов (2) можно заменить знак умножения на знак сложения, а знак суммы – на любой знак, обозначающий только сочетание элементов (в качестве такого знака выбрана логическая связка «или»). Формула (2) перейдет в соотношение:

$$\left.\begin{array}{l}
d_n \mapsto 1, \\
d_{n-1} \mapsto x_0 \vee x_1 \vee x_2 \vee \ldots \vee x_{n-1}, \\
d_{n-2} \mapsto (x_0 + x_1) \vee (x_0 + x_2) \vee (x_0 + x_3) \vee \ldots \vee (x_{n-2} + x_{n-1}), \\
d_{n-3} \mapsto (x_0 + x_1 + x_2) \vee (x_0 + x_1 + x_3) \vee \ldots \vee (x_{n-3} + x_{n-2} + x_{n-1}), \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
d_{n-l} \mapsto (x_0 + x_1 + x_2 + x_3 \ldots + x_{l-1}) \vee \ldots \vee (x_{n-l-1} + \ldots + x_{n-1}), \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
d_0 \mapsto (x_0 + x_1 + x_2 + \ldots + x_{n-2} + x_{n-1}).
\end{array}\right\} \qquad (3)$$

Соответственно формула (1) перейдет в соотношение вида:

$$\begin{pmatrix} d_n \\ d_{n-1} \\ d_{n-2} \\ \vdots \\ d_0 \end{pmatrix} \mapsto
\left(\underbrace{\left.\begin{matrix} 1 & 0 \\ x_{n-1} & 1 \\ 0 & x_{n-1} \\ 0 & 0 \\ \ldots & \ldots \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix}\right\} n+1 \quad \cdots \quad \begin{matrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 0 \\ x_{n-1} & 1 \\ 0 & x_{n-1} \end{matrix}}_{n} \right)
\otimes
\left(\underbrace{\left.\begin{matrix} 1 & 0 \\ x_{n-2} & 1 \\ 0 & x_{n-2} \\ 0 & 0 \\ \ldots & \ldots \\ 0 & 0 \\ 0 & 0 \end{matrix}\right\} n \quad \cdots \quad \begin{matrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 0 \\ x_{n-2} & 1 \\ 0 & x_{n-2} \end{matrix}}_{n-1} \right)
\otimes \ldots \otimes
\begin{pmatrix} 1 & 0 \\ x_1 & 1 \\ 0 & x_1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ x_0 \end{pmatrix} \qquad (4)$$

где знак «$\otimes$» соответствует алгебраическому умножению матриц, но при этом собственно арифметическая операция умножения не производится, а заменяется на сложение $(+)$, арифметическое суммирование заменяется знаком «$\vee$» и интерпретируется по аналогии с тем, как отмечено для соотношения (3). Для дальнейшей обработки суммы отделяются скобками. На основе (4) алгоритм решения задачи о рюкзаке строится следующим образом. Соответственно умножению матриц в (2) справа налево на каждом шаге (4) берутся все возможные сочетания предметов, упорядочиваются посредством сортировки по суммарному весу предметов каждого сочетания, все сочетания веса, превосходящего заданное ограничение, из рассмотрения исключаются – отсекаются из множества допустимых.

1. Первый этап алгоритма. Запись данных в виде (4), где

d_{n-1} – вариант набора рюкзака, состоящий из одного предмета,

d_{n-2} – варианты наборов рюкзака, состоящих из двух предметов,

$\ldots$,

d_0 – варианты набора рюкзака, состоящий из всех предметов,

$x_0, x_1, \ldots, x_{n-1}$ – веса предметов.

2. Второй этап. «Умножение» матриц выполняется попарно последовательно справа-налево (параллельный вариант отдельно оговаривается в дальнейшем). На каждом шаге получается столбец (вектор), который снова «умножается» на предшествующую матрицу, в результате получим новый столбец. Число таких шагов равно n. Первый шаг, например, приведет к соотношению:

$$\begin{pmatrix} d_n \\ d_{n-1} \\ \vdots \\ d_0 \end{pmatrix} \mapsto \begin{pmatrix} \underbrace{\left.\begin{matrix} 1 & 0 \\ x_{n-1} & 1 \\ 0 & x_{n-1} \\ 0 & 0 \\ \dots & \dots \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix}\right\} n+1 \quad \dots \quad \begin{matrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 0 \\ x_{n-1} & 1 \\ 0 & x_{n-1} \end{matrix}}_{n} \end{pmatrix} \otimes \dots \otimes \begin{pmatrix} 1 & 0 & 0 \\ x_2 & 1 & 0 \\ 0 & x_2 & 1 \\ 0 & 0 & x_2 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ x_1 + x_0 \\ x_1 \cdot x_0 \end{pmatrix} \qquad (5)$$

По ходу каждого такого шага с помощью описанной ранее сортировки производится отсечение наборов, превосходящих ограничение по весу. Сортируются не сами веса наборов предметов, а их разности с границей максимально допустимого веса в рюкзаке. Тогда индекс перехода элемента отсортированной последовательности с неположительного на положительный является индексом отсечения всех положительных элементов, – тех, которые соответствуют превышению границы максимально допустимого веса. Соответственно, эти действия в дальнейшем используют новое обозначение: знак умножения «$\cdot$» в (5) заменяется знаком арифметического сложения «$+$», а знак «$+$» заменяется на знак логического сложения «$\vee$».

3. Третий этап. Вычисление стоимостей производится в порядке отсортированных по весу сочетаний предметов (за исключением недопустимых). Последовательность стоимостей сортируется по неубыванию. На выходе сортировки получается оптимальный (последний в массиве отсортированных элементов) по стоимости набор (сочетание) допустимых по весу предметов.

Алгоритм дан в конструктивной форме и является детерминированным. Каждый шаг разлагается на взаимно независимые операции соответственно естественному параллелизму матрично-векторных операций и сортировки. Временная сложность оценивается из теоремы:

Теорема 3. В рассматриваемой постановке задача о рюкзаке может быть решена с помощью детерминированного параллельного алгоритма с временной сложностью $T(2^{2n-3}) = O(n)$.

Использования схемы сдваивания для «умножения» матриц в (4) приводит к параллельной форме алгоритма с временной сложностью, приводимой в теореме:

Теорема 4. Рассматриваемая задача за счет роста числа процессоров может быть решена с помощью максимально параллельного аналога детерминированного алгоритма предыдущей теоремы с оценкой $T\left(2^{2n-1}\right) = O\left(\log_2 2n\right)$.

В главе даны примеры решения рассматриваемой задачи с детализацией предложенной и известных схем, которые во всех случаях приводят к правильным значениям оптимумов. Приводится сравнение полученных оценок временной сложности с оценками известных алгоритмов решения данной задачи. От известных аналогов предложенный алгоритм отличается по построению, а также детерминированным точным решением задачи с временной сложностью $O(n)$ либо

$O(\log_2 n)$, в зависимости от варианта алгоритма. Существующие оценки абстрактно улучшаются ценой роста числа процессоров.

В третьей главе рассматривается задача двумерной упаковки, приведен обзор постановок и методов решения задачи. Далее, предложен детерминированный алгоритм двумерной упаковки на основе той же модификации формул Виета, однако для выражения коэффициентов многочлена по его комплексным корням. В алгоритме по-прежнему используется та же сортировка. Задача решается в следующей постановке: дано n прямоугольных предметов с размерами $w_i \times h_i$, $i = 1, 2, \ldots, n$. Требуется найти упаковку в положительном квадранте с минимальной площадью окаймляющего прямоугольника. Повороты запрещены. Стороны предметов параллельны осям координат. Формальная запись:

$$\begin{cases} n - \textit{количество прямоугольников}; \quad w_i, h_i - \textit{длина и ширина } i-\textit{го прямогольника}; \; i = 1, 2, \ldots, n; \\ \qquad W, H - \textit{длина и ширина окаймляющего прямоугольника}; \quad W \times H \to \min \end{cases}$$

Пусть $z_i = x_i + I\, y_i$, $i = 1, 2, \ldots, n$, $I = \sqrt{-1}$. Коэффициенты многочлена $P_n(z) = d_n z^n + d_{n-1} z^{n-1} + d_{n-2} z^{n-2} + \ldots + d_1 z + d_0$ выражаются через его корни в виде (3) с точностью до замены x на z. Если прямоугольники со сторонами a_j, b_j интерпретировать как взаимно однозначно сопоставленные комплексным корням многочлена $z_j = a_j + I b_j$ так, что a_j – длина основания, b_j – высота, $j = 0, 1, \ldots, n-1$, то из (1), (2) следуют все возможные сочетания прямоугольников со сторонами a_j (основание), b_j (высота). Все сочетания из n по m получаются в соответствии коэффициенту многочлена d_{n-m}. Как и раньше, с целью реализации следует заменить знак умножения на знак «$\otimes$», который понимается как перечисление (без выполнения каких бы то ни было операций), знак суммы – на любой знак, обозначающий сочетание прямоугольников, в таком качестве выбран знак «$\vee$». С этими заменами получаются все сочетания из n по m прямоугольников:

$$\left.\begin{array}{l} d_n \mapsto 1, \\ d_{n-1} \mapsto z_0 \vee z_1 \vee z_2 \vee \ldots \vee z_{n-1}, \\ d_{n-2} \mapsto (z_0 \otimes z_1) \vee (z_0 \otimes z_2) \vee (z_0 \otimes z_3) \vee \ldots \vee (z_{n-2} \otimes z_{n-1}), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ d_{n-l} \mapsto (z_0 \otimes z_1 \otimes x_2 \otimes z_3 \ldots \otimes z_{l-1}) \vee \ldots \vee (z_{n-l-1} \otimes \ldots \otimes z_{n-1}), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ d_0 \mapsto (z_0 \otimes z_1 \otimes x_2 \otimes \ldots \otimes z_{n-2} \otimes z_{n-1}). \end{array}\right\} \qquad (6)$$

на основе

$$\begin{pmatrix} d_n \\ d_{n-1} \\ \vdots \\ d_0 \end{pmatrix} \Leftrightarrow \left(\underbrace{ \left.\begin{matrix} 1 & 0 \\ x_{n-1} \oplus y_{n-1} & 1 \\ 0 & x_{n-1} \oplus y_{n-1} \\ 0 & 0 \\ \ldots & \ldots \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix}\right\} n+1 \quad \cdots \quad \begin{matrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 0 \\ x_{n-1} \oplus y_{n-1} & 1 \\ 0 & x_{n-1} \oplus y_{n-1} \end{matrix} }_{n} \right) \otimes \left(\underbrace{ \left.\begin{matrix} 1 & 0 \\ x_{n-2} \oplus y_{n-2} & 1 \\ 0 & x_{n-2} \oplus y_{n-2} \\ 0 & 0 \\ \ldots & \ldots \\ 0 & 0 \\ 0 & 0 \end{matrix}\right\} n \quad \cdots \quad \begin{matrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 0 \\ x_{n-2} \oplus y_{n-2} & 1 \\ 0 & x_{n-2} \oplus y_{n-2} \end{matrix} }_{n-1} \right) \otimes \cdots$$

$$\ldots \otimes \begin{pmatrix} 1 & 0 \\ x_1 \oplus y_1 & 1 \\ 0 & x_1 \oplus y_1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ x_0 \oplus y_0 \end{pmatrix} \qquad (7)$$

где предполагается $d_n = 1$.

Описание алгоритма.

Этап I. Установление явной формы взаимно однозначного соответствия между элементами (прямоугольниками) и включающими их сочетаниями.

1.1. Все прямоугольники фиксируются в той последовательности, в которой задаются условием задачи, выбирается и фиксируется m, $m = \text{const}$, $m < n$

1.2. Последовательно каждому прямоугольнику взаимно однозначно сопоставляется свой целочисленный вес. Это выполняется для того, чтобы в дальнейшем идентифицировать каждый отдельный прямоугольник и осуществить выборку сочетаний с различными взвешенными элементами.

Конкретно, все входные прямоугольники нумеруются от единицы, и каждому пронумерованному прямоугольнику сопоставляется вес i равный его номеру: $i = 1, 2, \ldots, n$. Выбираются первые $\tilde{n} = m\left[\dfrac{n}{m}\right]$ номеров i, фиксируется $n - m\left[\dfrac{n}{m}\right] = n - \tilde{n}$ оставшихся номеров, которые на этом шаге не будут подаваться на вход алгоритма генерации сочетаний. Элементы в остатке от целочисленного деления не пересекаются с предыдущими, они группируются в одно отдельное сочетание (остаточное) из $n - \tilde{n}$ элементов и будут обрабатываться отдельно.

Этап II. Генерация и выборка сочетаний

2.1. На основании (6) генерируются все сочетания из $\tilde{n}$ по m. Остаточное сочетание образуется в порядке расположения его элементов автоматически. Структура преобразования матриц (7) сохраняется с заменой n на $\tilde{n}$, $\tilde{n} = \left[\dfrac{n}{m}\right]m$, что влечет обозначения:

$d_{\tilde{n}-1}$ – все сочетания из $\tilde{n}$ по 1, независимо от порядка,

$d_{\tilde{n}-2}$ – все сочетания из $\tilde{n}$ по 2, независимо от порядка, … ,

$d_{\tilde{0}}$ – сочетание из $\tilde{n}$ по $\tilde{n}$, независимо от порядка,

$x_j \oplus y_j$ – размеры j-го прямоугольника (x_j – длина основания, y_j – высота), $j = 0, 1, \ldots, \tilde{n} - 1$. Сочетания генерируются в порядке последовательного умножения алгебраических матриц справа налево (аналог параллельного умножения отдельно оговаривается в дальнейшем). Преобразования выполняются, пока не останется один результирующий столбец матрицы с числом строк, равным числу компонентов столбца (вектора) в левой части (7). В рассматриваемой трактовке вектору левой части сопоставлены наборы сочетаний, объединенных знаком $\vee$, элементы каждого отдельного сочетания объединяются знаком $\otimes$.

2.2. В строке, сопоставленной $d_{\tilde{n}-j}$, располагаются все сочетания элементов (прямоугольников) из $\tilde{n}$ по j. В каждой строке все сочетания нумеруются в порядке расположения слева направо, начиная от единицы. В строке, сопоставленной $d_{\tilde{n}-j}$, число сочетаний равно $C_{\tilde{n}}^{j}$, $j=\overline{1,\tilde{n}}$. Таким образом, $d_{\tilde{n}-j} \Leftrightarrow C_{\tilde{n}}^{j}$.

2.3. Выбор различных сочетаний из $\tilde{n}$ по m, не содержащих общих элементов.

2.3.1. С целью требуемого выбора в строке, соответственной $d_{\tilde{n}-m}$, все элементы переводятся в аналог двумерного массива: первый «индекс» – вес элемента (от единицы до $\tilde{n}$) (на самом деле это не индекс, поскольку он может повторяться у элементов разных сочетаний, в сочетании это именно назначенный априори вес выбранного элемента). Второй индекс (он действительно индекс) – номер сочетания. Затем выполняется сортировка по не строгому возрастанию, причем сортируются все $\tilde{n}$ элементов и только по их весу: ключ, по которому выполняется сортировка, совпадает с весом элемента сочетания. В силу устойчивости сортировки все равные веса в отсортированном массиве расположатся друг за другом по неубыванию. Вторые индексы элементов переставятся вместе с элементами, так что элементы с равными весами, следуя подряд друг за другом, образуют вторыми индексами последовательность номеров (вообще говоря, неупорядоченную) различных сочетаний, в которые эти элементы входят.

2.3.2. Из последовательности, полученной при действии 3.1 этапа II (2.3.1 – второй этап, действие 3.1), выбираются элементы, вес которых равен единице ($i=1$). Каждому такому элементу приписываются элементы сочетания, в которое этот элемент входил, – в сочетание с ним объединяются элементы, имеющие одинаковый с ним второй индекс.

Далее, каждое выбранное сочетание оптимально упаковывается (независимо друг от друга) способом, вариант которого представлен в дальнейшем. После упаковки всех сочетаний с элементом единичного веса фиксируются окаймляющие прямоугольники каждого сочетания с минимальной вследствии упаковки площадью. При этом после упаковки сохраняется второй (истинный) индекс сочетания.

Все вычисленные значения минимальных площадей сохраняются вместе со своими сочетаниями и затем сортируются по неубыванию с соответствующим смещением индекса сочетания. На выходе получается неубывающая последовательность площадей окаймляющих прямоугольников сочетаний, эти сочетания содержат элемент веса равного единице.

Первый элемент отсортированной таким образом последовательности – наименьшая из отсортированных площадь. Соответствующее этому элементу упакованное сочетание идентифицируется по его второму (истинному) индексу, перемещенному вместе с ним в процессе выполнения сортировки.

Для выбранного таким образом сочетания фиксируется его (приоритетное) местоположение. Принимая во внимание единичный вес определившего сочетание элемента и второй его индекс, определиший все элементы сочетания, обозначим данную пару индексов $(1, i_1)$.

Вложенный алгоритм для нахождения совпадающих элементов в последовательности весов, полученных при действии 2.3.2. этапа II).

Фиксируется сочетание $(1, i_1)$ и его элементы (прямоугольники, каждый со своим весом и индексом). Это фиксированное сочетание $(1, i_1)$ поэлементно (по каждому входящему в это сочетании прямоугольнику) по весу сравнивается с последовательностью весов, полученных при действии 2.3.2 (с одномерным массивом отсортированных весов). Совпадающие по весу элементы удаляются из числа оставшихся в линейке отсортированных весов, полученных при выполнении действия 2.3.2, причем с данной парой индексов (в различных упаковках не должно быть одних и тех же элементов). При этом по второму индексу ищутся элементы с индексом равным второму, они также удаляются из дальнейшего рассмотрения (таким образом, из рассмотрения удаляется не один элемент, а полностью все сочетание, содержащее вес выбранного ранее элемента).

Далее рассматривается линейка не удаленных весов с ближайшим весом к рассмотренному (на первом шаге – к единичному), для определенности она обозначается $(\tilde{k}, i_{\tilde{k}})$, $\tilde{k} = const$. Если элементов с весом $\tilde{k}$ несколько, то к полученной линейке заново применяются все операции этапа 2.3.2, с тем условием, что выбираются все сочетания, содержащие вес $\tilde{k}$, и каждое такое сочетание отдельно упаковывается, выбирается сочетание с наименьшей окаймляющей площадью и применяется вложенный алгоритм, а сравнение весов будет теперь выполняться с весом $\tilde{k}$. Сохраненное сочетание оптимизировано по площади окаймляющего все его элементы прямоугольника.

Аналогично, обрабатываются остальные сочетания, элементы которых упорядочены по весу k, $k = 3, 4, \ldots, \tilde{n}$. Все в совокупности обработанные таким образом, сохраненные и оптимизированные по площади окаймляющего прямоугольника сочетания, индексируются (k, i_k), $k = 1, 2, \ldots, \tilde{n}$. В результате выполнения последовательности данных шагов останется последовательность упакованных сочетаний прямоугольников с индексами (k, i_k), $k = 1, 2, \ldots, \tilde{n}$, среди которых некоторые значения k могут отсутствовать. Остаточное сочетание из $n - \tilde{n}$ элементов также подвергается упаковке (синхронно с другими) и индексируется $(\tilde{n} + 1, i_{\tilde{n}+1})$.

На выходе последнего шага оказываются в наличии n прямоугольников, так как удалению подвергались лишь те сочетания, в которых были совпадающие элементы, и останется набор сочетаний, включающий все различные прямоугольники. Для максимально параллельного отсечения недопустимых комбинаций упаковки используется сортировка.

2.3.3 При выполнении предыдущего действия 2.3.2 получена последовательность упакованных сочетаний прямоугольников с индексами (k, i_k), $k = 1, 2, \ldots, \tilde{n} + 1$ и минимизированные площади окаймляющих эти сочетания прямоугольников.

Именно окаймляющие эти сочетания прямоугольники вместе с входящими в них упакованными элементами интерпретируются теперь как новые элементы для только что описанной обработки (рис.1, 2).

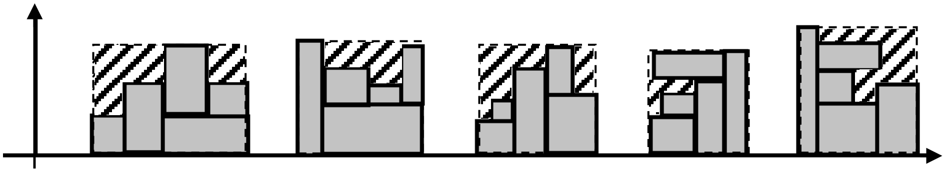

Рис. 1. Результат упаковки на шаге i

На шаге $i+1$ выходные окаймляющие контуры шага i будут интерпретироваться как новые входные прямоугольники для данного шага (шага $i+1$) (рис. 2).

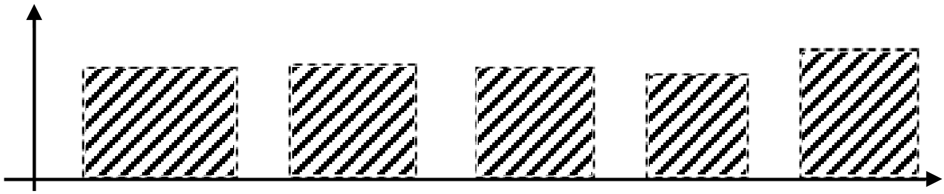

Рис. 2. Входные прямоугольники на шаге $i+1$

В качестве входных на шаге $i+1$ окаймляющие прямоугольники заново получают веса от единицы и второй индекс по сочетанию. При этом сочетания генерируются по тому же алгоритму, что и на предыдущем шаге, – на основе формулы (6).

Иными словами, по отношению к новым элементам (окаймляющим прямоугольникам, внутри которых фиксированы оптимизированные сочетания исходных элементов-прямоугольников) применяются этап I и все действия на нем 1.1., 1.2, затем этап II и все действия на нем 2.1., 2.2., 2.3.1., 2.3.2, 2.3.3.

Шаг $i=\lceil \log_m n \rceil$ окажется завершающим, поскольку число сочетаний убывает от шага как шагу обратно пропорционально m с единичным коэффициентом пропорции. Отсюда число рассматриваемых шагов не превзойдет $\sim \log_m n$.

Поскольку $n \le m^{\lceil \log_m n \rceil}$, число шагов алгоритма $\tilde{K} \le \lceil \log_m n \rceil$. Пусть $\tilde{t}$ – временная сложность упаковки одного сочетания прямоугольников. С учетом максимального параллелизма используемой сортировки, числа шагов алгоритма и последовательности операций получается:

$$T(R) \le q\frac{m}{m-1} \times (n + \log_m n + 1) \times (1 + \tilde{t}),\ q = \text{const},\ R \le \frac{\left(C_n^m m\right)^2}{2}. \qquad (7)$$

Лемма 1. Двумерная упаковка на основе сортировки и видоизменения формул Виета может быть выполнена с помощью детерминированного параллельного алгоритма с временной сложностью (7).

Следствие 1. Глобально оптимальная упаковка получится при $m = n$ При $m \ne n$ глобальный оптимум не является гарантированным, но наилучшая упаковка достигается в каждом отдельном сочетании прямоугольников на каждом шаге.

Теорема 5. Если каждое сочетание из m прямоугольников упаковывается параллельным перебором всех 2^m вариантов, то сохраняются утверждения леммы 1 и следствия 1, при этом временная сложность выполнения всего алгоритма составит

$$T(R) \le 2q\frac{m}{m-1}\times\left(n+\log_m n+1\right)\tau,\ q=\mathrm{const}, R\le 2^m\times\frac{\left(C_n^m m\right)^2}{2}, \qquad (8)$$

где τ – время бинарного сравнения, в оценке R учитывается число процессоров для максимально параллельной упаковки каждого рассматриваемого сочетания.

Следствие 2. Выполнение данного алгоритма параллельно по всем $m \le n$ даст выбор наилучшего по m варианта упаковки с временной сложностью

$$T(R) \le 2q\frac{m}{m-1}\times\left(n+\log_m n+1\right)\tau,\ q=\mathrm{const},\ R\le\left(2^n-2\right)\sum_{m=2}^{n}\left(C_n^m m\right)^2.$$

Следствие 3. Глобально наилучшая упаковка получится при $m=n$ с временной сложностью, $T(R)\le 2q\dfrac{n}{n-1}\times(n+2)\tau$, $q=\mathrm{const}, R\le\left(2^n-2\right)n^2$, или, $T\left(\left(2^n-2\right)n^2\right)=O(n)$.

Следствие 4. При $m=\lceil n/m\rceil$, или, $m\sim\sqrt{n}$, упаковка заканчивается за 2 шага и согласно (8) $T(R)\le 2q\dfrac{\sqrt{n}}{\sqrt{n}-1}\times(n+3)\tau$, $q=\mathrm{const}, R\le 2^{\sqrt{n}-1}\left(C_n^{\sqrt{n}}\right)^2 n$, так что $T(R)\le O(n)$, $R\le 2^{\sqrt{n}-1}\left(C_n^{\sqrt{n}}\right)^2 n$. При этом все элементы каждого сочетания на первом шаге будут упакованы оптимально, равно как и на втором, однако глобальный оптимум на выходе алгоритма не является гарантированным.

Следствие 5. Если выполнять упаковку по алгоритму леммы 1, при этом по тому же алгоритму (а не перебором) выполнять упаковку каждого сочетания на текущем шаге, то (6) перейдет в оценку

$$T(R)\le q\frac{m}{m-1}\times\left(n+\log_m n+1\right)\times\left(1+q\frac{\tilde m}{\tilde m-1}\times\left(m+\log_{\tilde m} m+1\right)\right)\tau,\ q=\mathrm{const},\ R\le R_0+\frac{\left(C_n^m m\right)^2}{2}\times\frac{\left(C_m^{\tilde m}\tilde m\right)^2}{2}$$

, где $\tilde m<m$, $\tilde m=\mathrm{const}$, – число элементов сочетания из m по $\tilde m$. В частности, при $m\sim\sqrt{n}$ и $\tilde m\sim\sqrt{m}$ получится: $T\left(R_0+\dfrac{1}{4}\left(C_n^{\sqrt{n}}\right)^2\left(C_{\sqrt{n}}^{\sqrt[4]{n}}\right)^2 n^{\frac{3}{2}}\right)=O\left(n^{\frac{3}{2}}\right)$.

Последняя оценка означает, что время возрастает, сохраняя полиномиальный рост порядка $n^{\frac{3}{2}}$ при снижении количества процессоров до $R_0+\dfrac{1}{4}\left(C_n^{\sqrt{n}}\right)^2\left(C_{\sqrt{n}}^{\sqrt[4]{n}}\right)^2 n^{\frac{3}{2}}$. Такую рекурсивную вложенность алгоритма можно было бы продолжить, однако ввиду отсутствия гарантии глобального оптимума упаковки на выходе алгоритма целесообразность более глубоких рекурсивных вложений не очевидна.

Временная сложность генерации сочетаний согласно (6) с точностью до постоянного коэффициента не повлияет на порядок роста данных выше оценок.

Представленные алгоритмы отличаются от известных аналогов по своему построению, детерминированностью достижения точного решения и полиномиальными оценками временной сложности, которые в случае упаковки улучшаются до линейных и логарифмических. Сравнение оформлено в виде таблиц, одна из которых дана непосредственно ниже:

Таблица 1

Временная сложность параллельной двумерной упаковки

Авторы	Алгоритм	Оценка временной Сложности	Достижение оптимума
Anderson R.J., Mayr E.W., Warmuth M.K.	Параллельный алгоритм на основе FFD алгоритма	$O(n)$	Не указывается
García L., Coromoto L., Miranda G., Rodrıguez C.	Параллельный алгоритм - на основе Вишванатан и алгоритма Багчи (VB) - решение двумерной задачи раскроя (2DCSP).	Не приводится	Значение оптимума экспериментально близкое к глобальному
Zhao X., Shen H.	Квадратные элементы упаковываются в контейнеры, при этом стороны квадратов не больше 1	$\Theta(n)$, верхняя граница не превосходит (9/4), используется 32 процессора	Асимптотическая ошибка в худшем случае не превосходит верхнюю границу (9/4)
Han X., Ye D, Zhoux Y.	Упаковка в контейнеры гиперкубов	$O(n\log n)$	Оптимальность алгоритма не Достигается
Fernández A., Gil C., Baños R., Montoya M.G.	Упаковка с поворотами	Не приводится	Оптимальность алгоритма не достигается.
Leon C., Miranda G., Rodrıguez C., Segura C.	Ортогональный раскрой	Не приводится	Оптимальность алгоритма достигается не всегда
Baz D.E., Hifi M., Saadi T.	Раскрой	Не приводится	Оптимальность алгоритма не доказана.
Thiebaut D.	2D-упаковка изображений	Приводятся результаты численного эксперимента	Оптимальность не доказывается
Ромм Я.Е., Назарьянц Е.Г.	Ортогональная упаковка с вложенностями	$T(R) \le q\dfrac{m}{m-1}\times$ $\times(n+\log_m n+1)\times(1+\tilde{t})$, $q=\text{const}$, $R \le R_0 + \dfrac{(C_n^m m)^2}{2}$	Локальный оптимум достигается на каждом этапе

Ромм Я.Е., Назарьянц Е.Г.	Двумерная упаковка с вложенностями	$T\left(R_0+\frac{1}{4}\left(C_n^{\sqrt{n}}\right)^2\left(C_{\sqrt{n}}^{\sqrt[4]{n}}\right)^2 n^{\frac{3}{2}}\right)=O\left(n^{\frac{3}{2}}\right)$	Локальный оптимум достигается на каждом этапе
Ромм Я.Е., Назарьянц Е.Г.	Двумерная упаковка без вложенностей	$T\left(\left(2^n-2\right)n^2\right)=O(n)$	Достигается глобальный оптимум

Таким образом, предложен детерминированный параллельный алгоритм двумерной упаковки и его разновидности на основе устойчивой адресной сортировки и видоизменения формул Виета. Временная сложность алгоритма имеет полиномиальную оценку, близкую к линейной, однако упаковка оптимальна в каждом сочетании на каждом последовательном шаге, но глобально оптимальной она оказывается лишь в случае экспоненциального роста числа процессоров.

В заключении сформулированы основные научные результаты диссертационной работы и выводы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

Основной результат диссертации состоит в том, что разработаны алгоритмы преобразований графов и комбинаторных сочетаний на основе видоизменения формул Виета и алгоритмов сортировки в приложении к задачам дискретной оптимизации. Представленные алгоритмы достигают локально и глобально оптимальных решений с оценками временной сложности не выше полиномиальных за счет роста затрат числа компонентов вычислительной системы.

Научной новизной отличаются следующие результаты.

1. Предложен алгоритм преобразования матричной формы графа метода ветвей и границ на основе устойчивой адресной сортировки, отличающийся от известных аналогов сокращением количества последовательных шагов, что позволяет реализовать определение нижней границы каждой ветви с единичной оценкой временной сложности. Полностью метод ветвей и границ реализуется с полиномиальной оценкой количества последовательных шагов за счет роста числа требуемых параллельных компонентов вычислительной системы, что позволяет получать решение NP-сложной задачи за допустимое время ценой роста затрат оборудования (С. 53 – 61).

2. Для случая преобразования графа метода ветвей и границ с обработкой одновременно нескольких ветвей временная сложность решения задачи коммивояжера оценивается из соотношения $T\left(\frac{n^6}{6}\right)=O\left(n^5\log_2 n\right)$. Численное моделирование предложенного алгоритма подтверждает правильность его работы и оценку числа шагов (С. 61 – 75).

3. Для генерации комбинаторных сочетаний предложены модификации формул Виета восстановления коэффициентов многочлена по его корням. На данной основе с использованием устойчивой адресной сортировки разработаны детерминированные алгоритмы решения задачи об одномерном булевом рюкзаке,

которые отличаются от известных по структуре и позволяют достигать оптимума с линейной и логарифмической временной сложностью ценой экспоненциального роста оборудования (С. 80 –91, 112 – 118).

4. На основе устойчивой адресной сортировки и видоизменения формул Виета применительно к генерации сочетаний разработан детерминированный алгоритм двумерной упаковки. Алгоритм отличается от известных аналогов структурой и не более чем линейным количеством последовательных шагов, что позволяет достигать локальной оптимальности на каждом последовательном шаге при экспоненциальном росте оборудования (С. 128 – 147).

5. На этой же основе даны разновидности детерминированных алгоритмов двумерной упаковки, одна из которых позволяет достигать глобального оптимума с линейной временной сложностью за счет экспоненциального с квадратичным множителем количества задействованных компонентов вычислительной системы, что отличает алгоритмы от известных и позволяет выполнить двумерную упаковку с линейной оценкой времени ценой существенного роста оборудования. (С. 147 – 154).

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Публикации в ведущих научных журналах из списка ВАК

1. Ромм Я.Е., Назарьянц Е.Г. Параллельные алгоритмы решения булевой задачи о рюкзаке на основе сортировки и видоизменения формул Виета. // Фундаментальные исследования. – 2015. – № 2. (часть 12). – С. 2575-2580.

2. Ромм Я.Е., Назарьянц Е.Г. Полиномиальная сложность параллельной формы метода ветвей и границ решения задачи коммивояжера // Известия ЮФУ. Технические науки. – 2015. – № 4 (165). – С. 44-55.

3. Ромм Я.Е., Назарьянц Е.Г. Параллельная двумерная упаковка на основе видоизменения формул Виета и сортировки // Современные наукоемкие технологии. – 2016. – № 6 (часть 2) – С. 280-287.

Другие работы по теме диссертации

4. Ромм Я.Е., Назарьянц Е.Г. Преобразование метода ветвей и границ для решения задачи коммивояжера на основе максимально параллельной сортировки / ТГПИ. – Таганрог, 2013. – 25 с. Деп. в ВИНИТИ 30.09.2013, № 279-В2013.

5. Ромм Я.Е., Дзюба А.С., Назарьянц Е.Г. Преобразование и численное моделирование метода ветвей и границ для решения задачи коммивояжера на основе максимально параллельной сортировки / ТГПИ. – Таганрог, 2014. – 49 с. Деп. в ВИНИТИ 30.09.2014, № 279-В2013.

6. Ромм Я.Е., Назарьянц Е.Г. Детерминированный параллельный алгоритм решения задачи об одномерном булевом рюкзаке на основе сортировки и видоизменения формул Виета / ТИ имени А.П. Чехова (филиал) ФГБОУ ВПО "РГЭУ (РИНХ)". – Таганрог, 2015. – 45 с. Деп. в ВИНИТИ 18.02.2015, № 32-В2015.

7. Ромм Я.Е., Назарьянц Е.Г. Параллельный детерминированный алгоритм двумерной упаковки на основе сортировки и видоизменения формул Виета / ТИ

имени А.П. Чехова (филиал) ФГБОУ ВО "РГЭУ (РИНХ)". – Таганрог, 2016. – 39 с. Деп. в ВИНИТИ 14.04.2016, № 61-В2016.

8. Ромм Я.Е., Назарьянц Е.Г. Одномерная и двумерная параллельная упаковка // XXVII Международная конференция: «Актуальные проблемы в современной науке и пути их решения» (Россия, г. Москва, 23 июня 2016 г.) // Евразийский Союз Ученых (ЕСУ) – № 6 (27) Часть 2. – г. Москва, 2016. – С. 50-52.

9. Ромм Я.Е., Назарьянц Е.Г. Параллельное решение задачи коммивояжера методом ветвей и границ с применением сортировки // Международная научно-практическая конференция «Информационные технологии: актуальные вопросы, перспективы и инновации», секция – технические науки – 15 августа 2016 г., г. Пенза, издательство «Наука и Просвящение», г. Пенза, 2016. – С. 20 – 28.

10. Ромм Я.Е., Назарьянц Е.Г. Параллельная двумерная упаковка // XVII всероссийский симпозиум по прикладной и промышленной математике (осенняя открытая сессия) – 1-8 октября, №4, том 23, 2016 г., г. Сочи-Дагомыс. – С. 383-384.

11. Ромм Я.Е., Назарьянц Е.Г. Параллельные детерминированные алгоритмы одномерной упаковки с линейной и логарифмической временной сложностью // XIII Международная научно-техническая конференция «Новые информационные технологии и системы» – 23-25 ноября, 2016 г., НИТиС-2016, г.Пенза, С. 272 – 276.

12. Ромм Я.Е., Назарьянц Е.Г. Параллельные алгоритмы решения некоторых задач комбинаторной оптимизации // материалы Всероссийской начно-практической конференции с международным участим. Технические науки. «Аспекты развития науки, образования и модернизации промышленности» – 20-21 апреля, 2017г., г. Таганрог – издательство Донской гос. тех. ун-т. – Ростов-на-Дону, ДГТУ, 2017. – С. 35 – 41.

В работах, которые выполнены в соавторстве, автору принадлежат:

– [1, 3, 6, 8, 11] – применение модифицированных формул Виета для генерации сочетаний; применение сортировки для численной оптимизации целевой функции; модификации алгоритмов локальной и глобальной оптимизации с оценками временной сложности;

– [2, 4, 5, 9] – локальная оптимизация с помощью сортировки на каждом шаге параллельной обработки ветвей; полиноминальные оценки временной сложности одновременной обработки нескольких ветвей; численное моделирование параллельного алгоритма;

– [7, 10, 12] – доказательства локальной и глобальной оптимальности предложенных алгоритмов; построение примеров для аналитической проверки с целью верификации предложенных алгоритмов.