

1609465

**Московский Государственный Технический Университет
им. Н.Э. Баумана**

На правах рукописи

Попов Алексей Юрьевич

**Модели и алгоритмы автоматизированной декомпозиции
схем ЭВМ**

Специальность 05.13.12 - Системы автоматизации проектирования.

**Автореферат диссертации на соискание ученой степени кандидата
технических наук.**

Москва - 2003

1603465

Научный руков

ор Овчинников В.А.

Р 1603465

Официальные с

Голов А.Ю.

ор Юркевич Е.В.

модели и алгоритмы
автом. диалог. систем

в С.В.

Ведущая орган

17.03 35
10.11.06

Документ
Акт

АРГОН"

Защита д
заседании д
государственнс
107005, 2-я Бау

2 2003 г. в 16 часов на
11.10 в Московском
Э. Баумана по адресу:

гью, просьба

ВОЗВРАТИТЕ КНИГУ НЕ ПОЗЖЕ
обозначенного здесь срока

Московского

ов С.Р.

Объем _____

Актуальность темы. Автоматизированное проектирование (АП) широко применяется при создании ЭВМ. Качество результатов конструкторского проектирования ЭВМ, т.е. технический уровень готового изделия, в большей мере зависит от степени использования и эффективности систем АП. На всех этапах, начиная с эскизного проектирования и кончая выпуском конструкторской документации, решаются задачи структурного синтеза комбинаторно-оптимизационного характера. Многие из этих задач относятся к классу NP-полных. В связи с этим, актуальным является разработка высокопроизводительных алгоритмов их решения, сравнение эффективности которых возможно на основании оценок вычислительных и емкостных сложностей, а также качества результатов проектирования.

Задача декомпозиции схем возникает при схемно-топологическом проектировании в процессе определения метрических параметров размещения элементов схем на типовых конструкциях и топологии линий связи, определяемых в ходе компоновки, размещения и трассировки. В большинстве случаев компоновка сводится к декомпозиции (разрезанию) схем на подсхемы с заранее заданным или неизвестным количеством элементов в них.

Декомпозиция схем успешно применяется при проектировании полужаказных СБИС на базовых матричных кристаллах (БМК), для которых определяется состав макроэлементов. Это позволяет достигнуть лучших результатов и перейти к решению задач меньшей размерности: глобальному размещению макроэлементов в макрообластях, локальному размещению, глобальной и локальной трассировке. Разрезание также широко применяется при размещении элементов ЭВМ на печатных платах. Например, дихотомическое разрезание схем по критерию минимальной связности позволяет получить вариант размещения, обеспечивающий прокладку минимального количества трасс через заданное сечение платы.

Вместе с этим можно отметить, что многие методы комбинаторной оптимизации не исследованы применительно к проблеме компоновки схем или исследованы недостаточно подробно. В частности, существующие алгоритмы декомпозиции схем реализуют лишь приближенные методы комбинаторной оптимизации и, как правило, не обеспечивают высокое качество решения для разрезания широкого круга схем или же обладают большой вычислительной сложностью. Таким образом, существует потребность в разработке алгоритмов, реализующих точные методы, а также в модификации и сравнении существующих приближенных методов с целью снижения их вычислительной сложности и повышения качества результатов декомпозиции.

Очевидно, что разработка высокопроизводительных алгоритмов невозможна без исследования вариантов их конечной реализации, что

связано с проблемой выбора структур данных. Вычислительная сложность вариантов алгоритма при использовании различных структур может существенно различаться. Данная задача, т.е. оптимизация выбора структур данных, является самостоятельной научной проблемой, актуальной для многих научно-технических направлений. Пути ее формализации частично рассматриваются в многочисленных работах таких ученых, как Кормен, Кнут, Ахо, Вирт, Райли, Трамбле и др.. Однако ее четкой формальной постановки до сих пор не получено. Это во многом связано с разнообразием стоящих перед исследователями задач.

Таким образом, актуальными являются разработка высокопроизводительных реализаций как эвристических так и точных методов, применимых для декомпозиции схем, а также формальная постановка и исследование методов решения задачи выбора структур данных для указанного класса комбинаторно-оптимизационных задач структурного синтеза.

Цель диссертационной работы. Целью данной диссертационной работы являются разработка новых моделей автоматизированной декомпозиции схем ЭВМ и создание высокопроизводительных алгоритмов их реализации, обеспечивающих высокое качество решения, а также расширение области применения существующих алгоритмов.

Задачи исследований. Для достижения поставленных целей потребовалось:

1. Выявить перспективные методы решения задачи разрезания схем.
2. Разработать математические модели процессов разрезания схем выбранными методами.
3. Исследовать способы снижения вычислительной и емкостной сложности реализаций выбранных методов.
4. Разработать аналитические средства оценки алгоритмов декомпозиции схем различной степени интеграции.
5. Создать алгоритмы декомпозиции и экспериментально исследовать области их применения.

Методы исследований. При решении задач данной диссертационной работы были использованы: методы теории графов, методы решения задач комбинаторной оптимизации (метод ветвей и границ, поиска в глубину, двоичной свертки); методы комбинаторного анализа, математического анализа.

Научная новизна работы.

1. Разработаны математические модели процесса декомпозиции по методам двоичной свертки и ветвей и границ, позволившие создать высокопроизводительные алгоритмы их реализации.

2. В результате анализа алгоритмов решения широкого класса комбинаторно-оптимизационных задач была выявлена совокупность операций преобразования исходной модели в модель результата и получена матрица сложности их реализаций для различных структур данных. Впервые получена в матричной форме формальная постановка задачи выбора структур данных для оптимизации комбинаторных алгоритмов по критерию минимума их вычислительной сложности, что позволило создать методику выбора структур данных.
3. Применение методики выбора структур данных и исследование вариантов реализаций последовательного алгоритма разрезания гиперграфа и алгоритмов двоичной свертки позволили существенно снизить их вычислительные сложности.
4. Разработана модификация метода ветвей и границ для решения задачи разрезания гиперграфа, выбраны структуры данных, обеспечивающие минимальную вычислительную сложность построения одной вершины дерева решений. Определены точные размеры дерева решений.

Практическая ценность работы. По результатам проведенных исследований создана исследовательская система, которая включает рабочие модули, реализующие алгоритм двоичной свертки, последовательный алгоритм разрезания и алгоритм разрезания гиперграфа по методу ветвей и границ. Получены экспериментальные зависимости времени их работы от размеров схемы. Данная система позволила, также, исследовать характеристики качества разрезания реализованных алгоритмов.

Апробация работы. Основные положения диссертационной работы обсуждены на Межвузовской юбилейной научно-технической конференции "Современные информационные технологии" (г. Москва 2001), Межвузовской научно-технической конференции "Современные информационные технологии" (г. Москва 2002).

Реализация и внедрения. Теоретические и практические результаты работы, полученные автором, нашли применение при разработке архитектуры и организации цифровой вычислительной системы для обработки сигналов и управления бортовым фурье-спектрометром высокого разрешения, выполняемой в рамках проекта ОКР «Метеор-М-ИКФС-2-МГТУ», а также при анализе предприятия ОАО "АВТОТОР ХОЛДИНГ". Документы, подтверждающие внедрение, приведены в приложении диссертации.

Публикации. По результатам диссертационной работы автором опубликовано 4 работы.

Объем и структура диссертации. Диссертационная работа состоит из введения, пяти глав, заключения, списка литературы и приложения, занимающих 175 страниц текста, в том числе 51 рисунок и 20 таблиц, список литературы из 63 наименований на 7 страницах, приложение на 2-х листах.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертации, сформулированы основные задачи и научные результаты, дано краткое содержание глав.

В первой главе описана проблемная область и определены объекты исследования: проанализированы задачи разрезания и выбраны математические модели исходного описания схемы, получена формальная постановка задачи разрезания, показано большое значение выбора структур данных для реализации комбинаторно-оптимизационных алгоритмов, проанализированы методы решения задачи разрезания.

Показано, что для задачи разрезания адекватной моделью схемы является гиперграф. Задача разрезания схемы сводится к задаче разбиения множества X вершин гиперграфа $H(X, U)$ на L подмножеств X_l , ($l \in 1..L$, L - количество типовых конструкций) таким образом, чтобы целевая функция принимала оптимальное значение при выполнении конструктивно-технологических ограничений. Критерием оптимизации при разрезании выбран минимум суммарного количества цепей, попадающих в разрез между кусками.

Для учета характеристик схемы (размера схемы, связности ее элементов и уровня интеграции элементной базы) в работе использованы показатели: n - количество элементов схемы; m - количество цепей схемы; p - средние количества связей элемента схемы; A - среднее количество элементов, связанных цепью; r - среднее количество элементов, связанных с данным.

Показано, что важной задачей при разработке высокопроизводительных комбинаторных алгоритмов является выбор структур данных, обеспечивающих минимизацию вычислительной сложности при допустимом значении емкостной сложности. Обзор отечественной и иностранной литературы, посвященной структурам данных, показывает, что задача минимизации вычислительной сложности алгоритмов посредством выбора структур данных формально не поставлена и четко не сформулирована, что ограничивает возможность ее целенаправленного решения.

Определена совокупность из известных методов решения задач комбинаторной оптимизации, которые могли бы быть применены для решения задачи разрезания гиперграфа. Изучение литературных источников, рассматривающих методы автоматизированной компоновки схем показывает, что актуальным является исследование метода ветвей и границ и метода свертки.

Во второй главе рассматривается проблема выбора структур данных. Проанализированы различные структуры данных, обобщены методы их использования, получены вычислительные сложности базовых операций для каждой из структур и матрица сложностей операций. Предложены критерии классификации базовых структур данных и создана их многокритериальная классификация. Получена формальная постановка задачи выбора структур данных, создана методика выбора структур данных. На примере последовательного алгоритма разрезания гиперграфа показана эффективность ее применения - получена реализация метода, обладающая линейной вычислительной сложностью.

В первом разделе второй главы рассматривается общая постановка задачи выбора структур данных. Показано, что вычислительная сложность комбинаторных алгоритмов зависит от используемых для их реализации структур. Их состав и характеристики связи для каждого алгоритма определяются по совокупности производимых в алгоритме операций. Сложная структура в общем случае может состоять из иерархически связанных простых, которые будем называть базовыми. Таким образом, для формальной постановки задачи необходимо исследовать совокупность базовых структур и базовых операций над структурами данных.

Исследованы приемы применения и предложены правила условного отображения сложных иерархических структур данных (Рис. 1). Рассмотрены понятия, относимые к структурам данных, неоднозначность которых препятствует дальнейшей формализации. Для выявления сходств и различий структур выделены три уровня абстракции: физический, характеризующий строение памяти для хранения структуры; уровень топологии, отражающий содержательные связи (отношения) элементов; уровень эффективности операций, характеризующий структуры данных по совокупности эффективно выполняемых над ними операций (под эффективностью понимается асимптотическая оценка вычислительной сложности). Для упрощения использования асимптотических оценок предложены пять классов сложности операций над структурами данных: класс I означает оценку сложности $\Theta(1)$ или $O(1)$; класс II - оценку $\Theta(\log n)$ или $O(\log n)$; класс III - $\Theta(n)$ или $O(n)$; класс IV - $\Theta(n \log n)$, $O(n \log n)$; класс V - $\Theta(n^2)$ или $O(n^2)$.

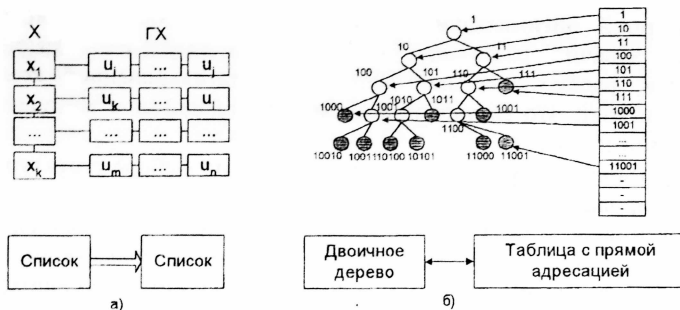


Рис. 1. Пример иерархической организации сложных структур данных

Определена совокупность базовых операций над структурами данных, отражающая простейшие операции над множествами: доступ к элементу структуры; поиск элемента с заданным значением ключа; добавление; удаление; поиск элемента с минимальным и максимальным значением ключа; поиск k-го по порядку; удаление элемента с минимальным и максимальным значениями ключа; изменение ключа; поиск следующего и предыдущего элемента по порядку на ключах; определение размеров структуры - операция "мощность"; конкатенация; объединение; исключающее объединение; пересечение; дополнение и разбиение структур; копирование структуры; создание и уничтожение пустой структуры).

Во втором разделе второй главы исследованы линейные структуры данных: массив, таблица с прямой адресацией, хеш-таблица, список. Для всех структур обобщены приемы их применения, описаны алгоритмы реализации базовых операций и определены оценки их сложностей.

В третьем разделе исследованы древовидные и сетевые структуры данных. Обобщены правила применения различных сбалансированных деревьев: AVL-деревьев, двоичной кучи, 2-3-деревьев, красно-черных деревьев, 2-3-4- и Б-деревьев, α -сбалансированных деревьев, биномиальных деревьев и образованных на их основе биномиальной и фибоначчисвой кучи. Рассмотрены способы поддержания свойства сбалансированности. Для каждой структуры исследованы алгоритмы реализации базовых операций и определены оценки их сложностей.

В четвертом разделе по предложенным критериям создана многокритериальная классификация структур данных, выявляющая сходства и различия базовых структур на физическом уровне представления, их топологическим свойствам и эффективности выполнения операций. Полученные оценки сложностей объединены в матрицу сложностей С операций для базовых структур (Рис. 2).

Структура	Операция															
	Доступ	Поиск	Поиск MIN	Поиск MAX	Поиск k	Добавление	Удаление	Удаление MIN	Удаление MAX	Изменение ключа	Следующий	Предшлющий	Комплекс	Комплексация	Объединение	Объединение (исключения)
Массив	1	3	3	3	3	3	3	3	3	1	3	3	1	3	3	4
Упорядоченный массив	1	2	1	1	1	3	3	3	3	3	1	1	1	3	3	2
Таблица с прямой адресацией	1	1	3	3	3	1	1	1	1	1	3	3	3	3	3	1
Хэш-таблица	1	1	3	3	3	1	1	1	1	1	3	3	3	3	3	3
Неупорядоченный список	1	3	3	3	3	1	1	1	1	1	3	3	3	1	5	3
Упорядоченный список	1	3	1	1	1	3	1	1	1	3	1	1	3	1	3	3
AVL-деревья	1	2	2	2	2	2	2	2	2	2	2	2	3	2	3	2
2-3 - деревья	1	2	2	2	2	2	2	2	2	2	2	2	3	2	3	2
Красно-черные деревья	1	2	2	2	2	2	2	2	2	2	2	2	3	2	3	2
B-деревья	1	2	2	2	2	2	2	2	2	2	2	2	3	2	3	2
α-сбалансированные двоичные деревья	1	2	2	2	2	2	2	2	2	2	2	2	3	2	3	2
Расширяющиеся деревья	1	2	2	2	2	2	2	2	2	2	2	2	3	2	3	2
Двоичные кучи	1	3	1	3	2	2	2	2	2	2	3	3	1	3	3	4
Биномиальные кучи	1	3	2	3	2	2	2	2	2	2	3	3	2	2	3	4
Фибоначчиевы кучи	1	3	1	3	2	1	2	2	2	1	3	3	2	1	3	4
n-мерный массив	1	2	1	1	1	3	3	3	3	3	1	1	1	3	3	2
n-мерный список	1	3	1	1	1	3	1	1	1	3	1	1	3	1	3	3

Рис. 2. Матрица сложностей С операций над структурами данных (I класс - $C_{ij}=1$; II класс - $C_{ij}=2$; III класс - $C_{ij}=3$; IV класс - $C_{ij}=4$; V класс - $C_{ij}=5$)

В пятом разделе второй главы на основе выявленных совокупностей базовых операций и структур данных, а также матрицы сложностей С создана формальная постановка задачи выбора структур данных.

Для формального представления алгоритма все его действия преобразуются в базис операций над структурами данных. Прочие операции, не влияющие на выбор структур данных, не учитываются. Переход к формальному описанию алгоритма продемонстрирован на примере последовательного алгоритма разрезания гиперграфа.

После выполнения данного перехода все операции алгоритма разделяются на независимые группы. При этом определяется состав и характеристики связей различного рода информации внутри создаваемых структур, т.е. данных одной группы.

Далее необходимо определить для всех групп количество выполнения каждой из операций, которые в дальнейшем будем описывать множеством G векторов A: $G = \{A^1, A^2, \dots, A^g\}$, $A^i_{1 \times |A|} = (a^i_1) = (0, \dots, p_1(n), \dots, p_m(n), \dots, 0, \dots)$, имеющих некоторое количество ненулевых элементов, где $a^i_1 = p_1(n)$, если в группе i есть операция 1 и она выполняется $p_1(n)$ раз ($a^i_1 \in A^i \rightarrow p_1(n) \neq 0$) и $a^i_1 = 0$ если операции a_1 в группе i нет ($a^i_1 \notin A^i \rightarrow p_1(n) = 0$).

Задачу выбора оптимальной структуры можно сформулировать как задачу дискретной оптимизации: алгоритм задан в виде g групп множества векторов операций $G = \{A^1, \dots, A^g\}$. Необходимо определить множество G' из g матриц S: $G' = \{S^1, \dots, S^g\}$, вида:

$$S'_{|A| \times |S|} = \begin{pmatrix} 0 & \dots & 1 & 0 \\ 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}_{|A| \times |S|},$$

содержащих лишь нули и единицы, у которых в каждой строке не более одной единицы и единица на пересечении строки i и столбца j означает, что операция i будет реализована с помощью структуры j . Тогда:

$$O^{\Sigma} = O' + \sum_{i=1}^k O' \longrightarrow \min, \quad (O') = A' \times ((S' \times C) \times \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}_{|A| \times 1}),$$

где C - матрица сложностей, O' - "удорожание" из-за применения более чем одной базовой структуры - в данный показатель входят сложности дополнительных действий, вызванных применением комбинации структур.

В шестом разделе предложен локально-оптимальный алгоритм выбора структур данных, на каждом шаге которого определяется структура данных для наиболее трудоемкой операции. Основные пункты алгоритма продемонстрированы на примере последовательного алгоритма разрезания гиперграфа в седьмом разделе второй главы. Полученные структуры позволили создать модификацию данного алгоритма, обладающую линейной вычислительной сложностью. Выявлены мало формализуемые этапы методики выбора структур данных и намечены пути ее дальнейшей формализации.

в третьей главе исследованы алгоритмы разрезания схем по методу двоичной свертки.

В первом разделе третьей главы разработана математическая модель разрезания гиперграфа по методу свертки, учитывающая конструкторские ограничения на состав формируемых кусков:

1. $|B_0(H)| = |X|$;
2. $B_{i-1}(H) \rightarrow B_i(H)$, $|B_i(H)| < |B_{i-1}(H)|$, $i \in J$;
3. $\forall H_i \in B_i(H)$, $H_i \neq \emptyset$, $i \in I_i$;
4. $\forall H_i, H_p \in B_i(H)$, $X_i \cap X_p = \emptyset$, $U_i \cap U_p = U_{i,p}$, $i \neq p$, $i, p \in I_i$;
5. $\cup H_i = \cup H_p = H$, $H_i \in B_i(H)$, $H_p \in B_{i-1}(H)$, $i \in I_i$, $p \in I_{i-1}$;
6. $\exists H_i$, $H_p \in B_{i-1}(H)$, $\exists H_{i,p} \in B_i(H)$, $H_{i,p} = H_i \cup H_p$, $i \neq p$, $i, p \in I_{i-1}$;
7. $|\cup U_{i,p}| = \max$, $i \neq p$, $i, p \in I_{i-1}$;
8. $B_i(H) \in D$;
9. $|B_n(H)| = L$, $n = |J|$,

где $H(X, U)$ -исходный гиперграф; $B_i(H)$ - множество кусков гиперграфа на уровне свертке i ; J - множество уровней свертки; I_i - множество индексов кусков гиперграфа на уровне свертке i ; D - область определения, задаваемая ограничениями; L -количество типовых конструкций.

Во втором, третьем и четвертом разделах исследованы различные варианты алгоритмов свертки: при неуравновешенной свертке на каждом уровне дерева происходит факторизация двух вершин в новую по критерию максимальной связности всех пар вершин; при уравновешенной свертке на

каждом уровне объединяются все пары в порядке уменьшения их связности. Для варианта свертки с предварительным учетом связности рассматриваются только те пары вершин, которые связаны между собой. Для свертки без учета связности рассматриваются все возможные пары вершин. Данные варианты были исследованы для выявления наиболее эффективной реализации метода.

Исследованы и сравнены вычислительные и емкостные сложности вариантов свертки простейших реализаций без предварительного выбора структур данных - исходный гиперграф задается матрицей инцидентности. Для неуравновешенной свертки исследованы два крайних случая формирования дерева свертки. Все полученные оценки учитывают изменение интегральных характеристик исходной схемы (p, r, A) в процессе свертки в соответствии с законом Рента. В таблице 1 показаны оценки вычислительной сложности в наиболее трудоемком случае при значении показателя Рента $R=0.5$.

Таблица 1

Вариант свертки	Вычислительная сложность	Емкостная сложность
Уравновешенная свертка без учета связности	$O(n^3, n^2 p^2)$	n^2
Неуравновешенная свертка без учета связности	$O(n^3(p^2+1), np^2)$	n^2
Уравновешенная свертка с учетом связности	$O(pn^2, pnAp)$	pn
Неуравновешенная свертка с учетом связности	$O(n^2 p p^2, npAp)$	pn

Таким образом, теоретические исследования показали, что учет связности позволяет на порядок сократить вычислительную и емкостную сложности алгоритмов свертки.

В пятом разделе к алгоритмам свертки применена методика выбора структур данных. Проведено преобразование блок-схемы алгоритма неуравновешенной свертки с учетом связности к блок-схеме в базовых операциях над структурами данных (Рис.3).

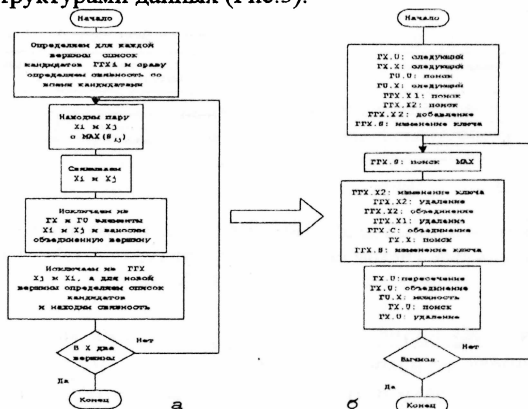


Рис. 3. Общая схема алгоритма неуравновешенной двоичной свертки с учетом связности (а) и схема алгоритма в базисе операций над структурами данных (б)

Операции алгоритма были разделены на группы и для каждой группы применен локально-оптимальный алгоритм выбора структур данных. Так же в данном разделе предложены способы снижения вычислительной сложности посредством коррекции показателей связности и разработан уточненный алгоритм неуравновешенной свертки с учетом связности.

Показано, что выбранные структуры данных (Рис. 4) позволяют сократить вычислительную сложность алгоритмов свертки с учетом связности до оценки $O(n^{R+1}pr, pr^2A \log p)$ в наиболее трудоемком случае (где R - показатель Рента). Также показано, что параметры реальных схем влияют на старшую степень полинома оценки вычислительной сложности полученного алгоритма.

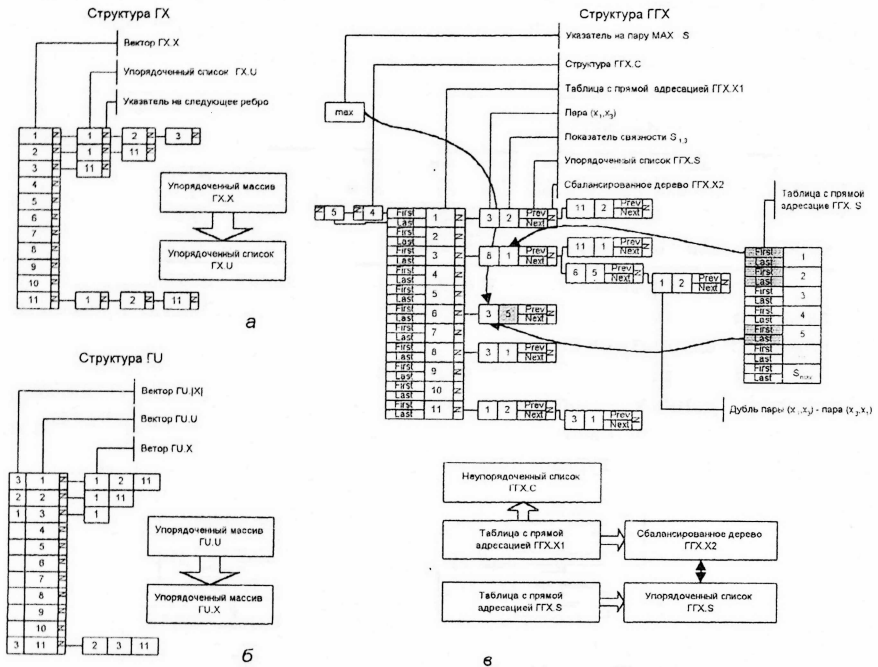


Рис. 4. Структуры данных для алгоритма неуравновешенной двоичной свертки с предварительным учетом связности: структура GX (а); структура GU (б); структура GX (в)

В четвертой главе исследована реализация метода ветвей и границ для дихотомического разрезания схем.

В первом разделе четвертой главы рассмотрены способы построения дерева решений и получена математическая модель дихотомического разрезания гиперграфа по методу ветвей и границ, которая может быть

представлена следующим образом: найти разрезание гиперграфа $\Pi(X, U)$ на два куска $H_1(X_1, U_1)$ и $H_2(X_2, U_2)$, что

1. $H_1^0 = H_2^0 = \emptyset, H_n^0 = H$;
2. $H_1^i = (X_1^i, U_1^i), H_2^i = (X_2^i, U_2^i), H_d^i = (X_d^i, U_d^i) = H \setminus \{H_1^i \cup H_2^i\}$;
3. $X_d^i = X_d^{i-1} \setminus x_j, X_1^i = X_1^{i-1} \cup x_j$ или $X_2^i = X_2^{i-1} \cup x_j$;
4. $X_1^i \cup X_2^i \cup X_d^i = X, |U_1^i \cap U_2^i| = S_{1,2}^i \rightarrow \min$;
5. $|H_1^n| = \lceil |H|/2 \rceil$ или $|H_2^n| = \lceil |H|/2 \rceil$,

где $S_{1,2}$ - количество ребер, попадающих в разрез между кусками H_1 и H_2 . Без учета целевой функции (S) задача имеет множество $T = \{T_k / k=1..K_B\}$ решений.

Данная модель отражает перенос вершин из исходного гиперграфа H_d в один из двух формируемых кусков H_1 или H_2 , что порождает правую или левую ветви дерева решений. Так как $X_2 = X \setminus X_1$ и количество вариантов

разрезания $K_B = \frac{C_n^2}{2} = C_{n-1}^2$, то задача является NP-сложной.

Во втором разделе четвертой главы доказана применимость метода ветвей и границ для разрезания схем: описана целевая функция и доказано, что критерий $S_{1,2}^i = |U_1^i \cap U_2^i|$ является нижней границей целевой функции, т.е. определяет перспективность ветви дерева решений (Рис 5).

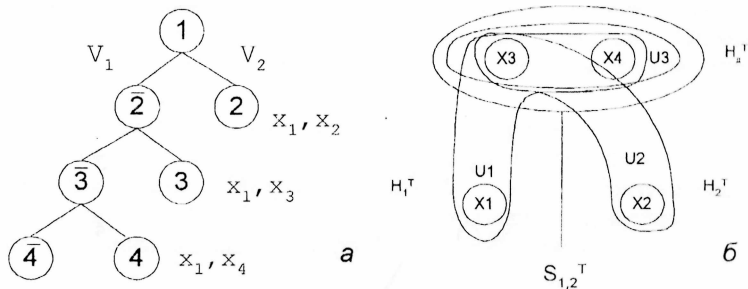


Рис. 5. Дерево решений при разрезании гиперграфа по методу ветвей и границ (а) и текущее состояние гиперграфа, характеризующееся критерием перспективности $S_{1,2}^i$ (б)

Доказано неубывание данного критерия при дальнейшем формировании вариантов, т.е. при описанном способе формирования, $\forall H_1, H_2, \forall H_1', H_2', H_1 \subset H_1', H_2 \subset H_2' \rightarrow S_{1,2} \leq S_{1,2}'$.

В третьем разделе рассмотрены различные схемы реализации метода ветвей и границ: аспекты отсекаания ветвей и выбор вершин ветвления. Исследованы два возможных способа ветвления: ветвление по минимуму нижней границы целевой функции и последовательное построение ветвей. Также показано, что опорное решение $H_1^0(X_1^0, U_1^0)$, полученное последовательным алгоритмом целесообразно использовать в качестве дополнительной отсекающей оценки.

В четвертом разделе рассмотрены способы формирования дерева решений: при заданном заранее порядке включения вершин в X_1 и при определении очередной вершины $x_i \in X$ при каждом ветвлении. Были рассмотрены различные стратегии выбора очередной вершины для переноса в H_1 или H_2 . Так как при полном заполнении одного из кусков, второй автоматически дополняется всеми не перенесенными вершинами из H_d , то критерий перспективности изменяется скачкообразно при получении конечного решения. Это может значительно увеличить количество рассмотренных вершин дерева решений. Поэтому, для сглаживания подобных скачков было предложено предварительное упорядочивание множества переносимых вершин последовательным алгоритмом разрезания.

В пятом разделе четвертой главы рассмотрены способы хранения структуры дерева, кодирования вершин дерева решений и оценена сверху их емкостная сложность. Показано, что структура дерева решений необходима только для определения X_1^T и X_2^T как последовательности номеров определенных вершин маршрутов в висячие или конечные вершины соответственно. В связи с этим нет необходимости хранить информацию в явной форме отражающую структуру дерева решений. Достаточно хранить лишь последовательность его конечных и висячих вершин, т.е. огибающую цепь (Рис. 6).

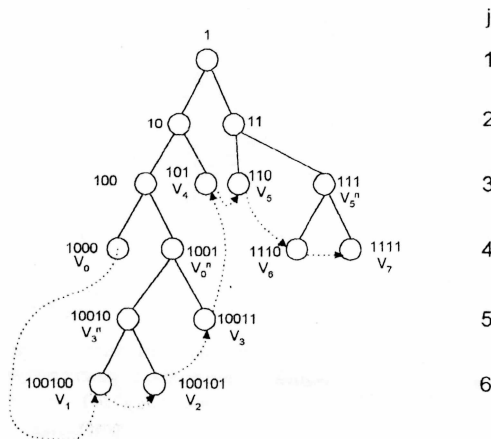


Рис. 6. Огибающая цепь ($V_0, V_1, V_3, V_4, V_5, V_6, V_7$) дерева решений

Исследованы емкостные сложности для различных способов кодирования кусков X_1^T и X_2^T . Показано что битовое кодирование состава кусков позволяет сократить вычислительную и емкостную сложности алгоритма.

В шестом разделе четвертой главы проведено исследование структуры двоичного числового дерева решений для разрезания гиперграфа по методу ветвей и границ и доказано, что для восстановления состава множеств X_1^T и

X_2^T достаточно хранить номера уровней конечных вершин дерева, т.е. огибающую цепь можно представлять таблицей уровней ее вершин. Разработаны правила вычисления состава кусков по таблице уровней. Для

$$K(V_l) = \begin{cases} \{K(V_r) + 1\} \bullet \{0\}_{\Delta, r} & \text{при } l < r, \\ \{K(V_r) + 1\} \setminus \{0 \vee 1\}_{\Delta, r} & \text{при } l \geq r, \end{cases}$$

соседних в огибающей цепи вершин V_l и V_r , находящихся соответственно на l -м и r -м уровнях дерева решений двоичный код $K(V_l)$ определяется:

где " $\bullet \{0\}_{\Delta, r}$ " - операция конкатенации $|l-r|$ нулей справа, " $\setminus \{0 \vee 1\}_{\Delta, r}$ " - операция удаления $|l-r|$ разрядов справа.

Предложен алгоритм нахождения полного состава кусков при двоичном кодировании огибающей с помощью таблицы уровней.

В седьмом разделе четвертой главы на основании разработанной схемы реализации метода ветвей и границ для дихотомического разрезания гиперграфа, найденного критерия перспективности ветвей дерева решений, выбора способа формирования дерева и способов кодирования дерева предложен алгоритм дихотомического разрезания гиперграфа по методу ветвей и границ.

В восьмом разделе четвертой главы проведен выбор структур данных для полученного алгоритма разрезания по методу ветвей и границ. Результатом применения методики выбора структур данных является структура "упорядоченный список" - "таблица с прямой адресацией", примененная ранее для алгоритма двоичной свертки. Определена вычислительная сложность построения одной вершины дерева решений: $O(\pi/|W|)$, где $|W|$ - длина слова в битах.

Для определения области применения алгоритма по методу ветвей и границ были определены точные размеры полного числового дерева решений для данного метода. Цифровое дерево, образующееся при разрезании гиперграфа по методу ветвей и границ значительно меньше полного двоичного и полного цифрового деревьев (Рис. 8).

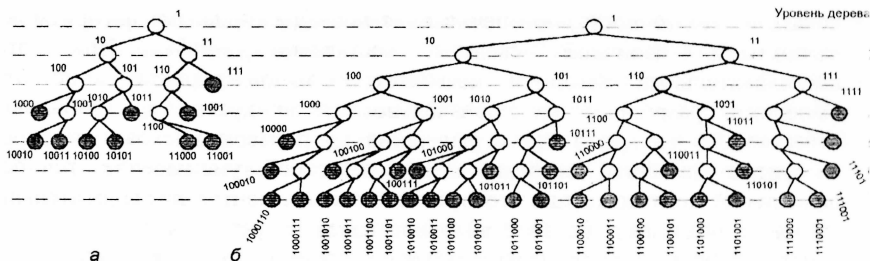


Рис. 8. Цифровое дерево решений при разрезании гиперграфа по методу ветвей и границ для $n=6$ (а) и $n=8$ (б). Рядом с вершинами указан их код. Конечные вершины закрашены

Количество вершин полного дерева решений при разрезании гиперграфа по методу ветвей и границ можно выразить следующим образом:

$$V = \sum_{i=1}^{\frac{n}{2}-1} 2^{i-1} + \sum_{i=\frac{n}{2}}^{n-2} \sum_{r=i-\frac{n}{2}}^{\frac{n}{2}-2} C_{i-1}^r + \sum_{i=n/2}^{n-1} C_{i-1}^{n/2-1}.$$

Была доказана следующая теорема:

Теорема 1. *Количество вершин полного дерева решений равно:*

$$V = C_n^{\frac{n}{2}} - 1.$$

В пятой главе описана система автоматизированной декомпозиции, созданная на основе результатов проведенных исследований. Данная система предназначена для разрезания схем на части с заранее заданным или неизвестным количеством элементов, а также исследования качества декомпозиции и временной сложности алгоритмов двоичной свертки, последовательного алгоритма и алгоритма дихотомического разрезания по методу ветвей и границ. Проведенное с помощью данной системы экспериментальное исследование алгоритмов двоичной свертки подтвердило линейный характер их вычислительной сложности. Исследование возможности применения алгоритма дихотомического разрезания гиперграфа по методу ветвей и границ для декомпозиции схем показало, что предварительное упорядочивание вершин и использование отсекающей оценки сокращает количество рассматриваемых вершин дерева решений на ~30%. Несмотря на значительное сокращение реальных размеров дерева решений по сравнению с наихудшим случаем, вычислительная сложность алгоритма разрезания по методу ветвей и границ близка к экспоненциальной. В связи с этим он может быть применен лишь для схем малой размерности.

Экспериментальные исследования качества решений алгоритмов показало, что точность, обеспечиваемая последовательным алгоритмом и алгоритмом двоичной свертки составляют ~90%. При этом последовательный алгоритм позволяет с малой скрытой константой выполнять разрезание на произвольное число кусков со стабильным качеством, в то время как ограничения на размеры кусков при работе алгоритма двоичной свертки влекут за собой уменьшение точности решения. В подтверждение этого, в ходе экспериментов последовательный алгоритм показывал более стабильное качество результатов. Т.е. можно констатировать большую универсальность последовательного алгоритма по сравнению с двоичной сверткой при практически идентичных характеристиках точности решения.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Разработаны математические модели процесса разрезания гиперграфа по методам двоичной свертки и ветвей и границ, что дало возможность создать алгоритмы их реализации.
2. Проведенный анализ комбинаторно-оптимизационных алгоритмов выявил совокупность операций при преобразовании исходной модели в модель результата и позволил определить сложности их реализаций для различных структур данных.
3. Созданная классификация базовых структур данных, отражающая их топологические свойства и физическую реализацию, а также оценка эффективности выполнения операций позволили выявить сходства и различия между структурами и обобщить приемы их использования.
4. Получена формальная постановка задачи выбора структур данных, как задачи дискретной оптимизации, что дало возможность создать методику выбора структур данных, включающую ряд правил и формальных алгоритмов.
5. Методика выбора структур данных применена для последовательного алгоритма разрезания гиперграфа. Выбранные структуры позволили получить реализацию алгоритма, обладающую линейной вычислительной сложностью.
6. Исследование вариантов реализации метода двоичной свертки для решения задачи разрезания гиперграфа и проведенный выбор структур данных дали возможность значительно сократить вычислительную сложность алгоритма.
7. Разработан рабочий модуль, реализующий алгоритм двоичной свертки, обладающий малой вычислительной сложностью. Экспериментально исследованные характеристики его работы подтвердили правильность теоретических расчетов вычислительной сложности алгоритма.
8. Разработана модификация метода ветвей и границ для решения задачи разрезания гиперграфа, выбраны структуры данных, обеспечивающие минимальную вычислительную сложность построения одной вершины дерева решений. Определены точные размеры дерева решений. Создан алгоритм разрезания гиперграфа по методу ветвей и границ, получены оценки сверху времени работы алгоритма.
9. Разработан рабочий модуль, реализующий алгоритм разрезания гиперграфа по методу ветвей и границ. Экспериментально исследованы характеристики его работы.

Работы по теме диссертации.

1. Овчинников В.А., Иванова Г.С., Попов А.Ю. Применение метода ветвей и границ для решения задачи дихотомического разрезания гиперграфа // Вестник Московского государственного технического университета. Приборостроение. - 1999. - N2. - С.78-91.
2. Овчинников В.А., Николаев К.В., Попов А.Ю. Исследование вычислительной сложности алгоритмов двоичной свертки схем ЭВМ // Вестник Московского государственного технического университета. Приборостроение. - 1997. - N2. - С.113-120.
3. Попов А.Ю. Вопросы применения автоматизированных обучающих систем в технических вузах // Современные информационные технологии, Сборник докладов и сообщений Межвузовской юбилейной научно-технической конференции аспирантов и студентов. - М.: Изд-во МГТУ им Баумана, 2001. - С. 28-35.
4. Попов А.Ю. Проблема выбора структур данных в задачах схемно-топологического проектирования // Современные информационные технологии, Юбилейный сборник трудов кафедры. -М.: Изд-во МГТУ им Баумана, 2002. - С. 57-60.