На правах рукописи

ФИЛИМОНОВ АНДРЕЙ ВИКТОРОВИЧ

РАЗРАБОТКА И РЕАЛИЗАЦИЯ МНОГОУРОВНЕВЫХ АЛГОРИТМОВ ДЕКОМПОЗИЦИИ ГИПЕРГРАФОВЫХ МОДЕЛЕЙ

Специальность 05.13.18 — Математическое моделирование, численные методы и комплексы программ

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Н.Новгород
2008

Работа выполнена в Федеральном Агентстве по Образованию «Нижегородский Государственный Университет им. Н.И.Лобачевского»

Научный руководитель — доктор технических наук, профессор Батищев Дмитрий Иванович

Официальные оппоненты — доктор технических наук, профессор Гергель Виктор Павлович,

доктор технических наук, доцент Хранилов Валерий Павлович

Ведущая организация — ФГУП ФНПЦ «НИИИС им. Ю.Е. Седакова»

Защита состоится «___» _____ 2008 г. в ___ часов на заседании Диссертационного Совета Д 212.166.13 при Нижегородском Государственном Университете им. Н.И.Лобачевского по адресу: 603950, Н.Новгород, пр.Гагарина, д. 23, .

С диссертацией можно ознакомиться в библиотеке Нижегородского Государственного Университета им. Н.И.Лобачевского.

Автореферат разослан «___»_____2008 г.

Ученый секретарь
диссертационного совета

кандидат
физико-математических наук_____ Савельев Владимир Петрович

Актуальность темы исследований

Проектирование современной радиоэлектронной аппаратуры (РЭА), программных систем и вычислительных комплексов предполагает решение большого количества различных математических задач. Естественно, способность решать такие задачи предполагает наличие соответствующих алгоритмов решения. Однако, зачастую размерности и сложность таких задач настолько велики, что делают невозможным их решение, даже с использованием мощнейшей вычислительной техники. В связи с этим возникла проблема практической разрешимости задач: найти реализуемый, эффективный или хотя бы достаточно простой в практически важных случаях алгоритм ее решения. Подходы, относящиеся к этой категории применимы исключительно к оптимизационным задачам (однако это обстоятельство не является сильным ограничением, поскольку огромное количество прикладных задач естественно формулируются как оптимизационные) и включают прием, который заключается в отказе от поиска оптимального решения и нахождении вместо этого “приемлемого” решения за обозримое время. Методы, применяемые для построения алгоритмов такого типа, сильно зависят от специфики задачи, и хотя на практике оказываются весьма неплохими, для получения удовлетворительных характеристик алгоритма обычно требуется довольно большая работа по его “доводке”. В результате удается только в очень редких случаях предсказать и оценить поведение таких алгоритмов.

Конечно, термин “приемлемые” не строг в математическом смысле. Под “приемлемыми” решениями нами понимаются решения, которые удовлетворяют исследователя. Ведь в большинстве реальных задач нет особой необходимости находить именно глобальный оптимум. Чаще всего целью поисков являются решения, удовлетворяющих определенным ограничениям. Например, электрические цепи принципиальной схемы некоторого электронного устройства требуется распределить в непересекающихся слоях, а число таких слоев не должно превышать заданной величины. В этом смысле достаточно найти именно “приемлемые”, т.е. разумные решения.

Традиционно, решения задач проектирования РЭА, программных систем и вычислительных комплексов связывают с решением графовых задач. Так, например, при производстве современного электронного оборудования особое место занимает разработка печатных плат и микроэлектронных схем большой плотности. Одной из задач, которая требует решения при их конструировании, является задача разбиения принципиальных схем устройств на составные части (подсхемы, функциональные блоки). Типовые элементы принципиальной схемы должны таким образом быть распределены по отдельным подсхемам, чтобы число внешних соединений было как можно меньше. Это необходимо с целью повышения надежности схемы, уменьшения влияния наводок, повышения технологичности и простоты конструктивного оформления.

При разработке топологии многослойных схем возникает задача распределения “конфликтующих” соединений по отдельным слоям (задача расслоения цепей). Расслоение до трассировки основано на анализе схемы

соединений для выявления тех соединений или их групп, которые не могут быть расположены на одной плоскости из-за неизбежности “сложных” ситуаций трассировки. В частности, усложняется форма соединений, увеличивается их длина, растет время поиска трасс при реализации алгоритма трассировки на ЭВМ.

Использование графов при решении такого рода задач часто налагает существенные ограничения, как на математические модели, так и на решающие алгоритмы, что, в конечном итоге, отрицательно сказывается на качестве получаемых решений. Часто более предпочтительным с точки зрения точности моделирования является использование обобщения графов – гиперграфов, поскольку они не только более естественно описывают названные выше инженерные задачи, но и позволяют решать некоторые задачи линейной алгебры и дискретной математики, сведение которых к классическим графам либо невозможно, либо порождает очень сложные модели. Однако, несмотря на значительно более широкие возможности, предоставляемые гиперграфовыми моделями, сложность машинной реализации самих моделей и алгоритмов, работающих над ними, также выше. В этой ситуации особенно остро встает проблема построения алгоритмов, осуществляющий поиск приближенных решений.

Можно выделить два класса подходов решения задач высокой сложности – это упрощение алгоритмов: снижение их математической и инженерной сложности, применение различного вида эвристик, и упрощение решаемых задач. Часто для решения задачи создается метод, который сочетает в себе два этих подхода. К таким методам можно отнести целый класс алгоритмов, получающий широкое распространение в последнее время – многоуровневые алгоритмы. Ключевая идея многоуровневого алгоритма состоит следующем: размерность задачи редуцируется путем удаления наименее существенной информации, затем производится поиск решения редуцированной задачи, после этого производится ряд улучшений полученного решения на основе добавления в модель задачи ранее удаленной информации.

Многоуровневые алгоритмы предполагают широкие возможности по настройке алгоритма, поскольку позволяют достаточно легко интегрировать в структуру алгоритма различные эвристики, направленные на улучшение получаемого решения. Не смотря на очевидные достоинства многоуровневых методов, разработка и исследование этого класса алгоритмов применительно к графовым и гиперграфовым моделям у нас в стране – редкое явление. За рубежом такого рода методами занимаются G.Karypis, V.Kumar, B.Hendrickson, R.Leland, S. Barnard, H. Simon и др.

Еще одним примером способа сокращения вычислительной сложности решающих алгоритмов может служить введение элементов рандомизации. Данное направление дало толчок к развитию методов эволюционных вычислений (подкласс методов случайного поиска), что позволило построить универсальные и достаточно мощные алгоритмы для широкого класса задач. У нас в стране разработкой такого типа алгоритмов занимались Д.И. Батищев, Л.С. Бернштейн,

И.Л. Букатова, Я.Г. Дорфман, Б.П. Коробков, В.М. Курейчик, А.М. Мелихов, Ю.И. Неймарк, Г.И. Орлова, Л.А. Растригин, А.П. Рыжков, Д.Б. Юдин и другие. Методы эволюционных вычислений не гарантируют обнаружения оптимального решения. Однако практический интерес к ним не ослабевает, а наоборот усиливается. Объяснить это можно тем обстоятельством, что эти методы позволяют исследовать и находить приемлемые решения таких задач, решение которых при помощи традиционных методов оказывается затруднительным, а в некоторых случаях и просто невозможным.

Синтез многоуровневых и эволюционно-генетических алгоритмов позволяет получать гибкие, легко настраиваемые под каждую конкретную задачу методы, которые позволяют контролировать баланс между качеством решения и вычислительным ресурсом, потраченным на его получение.

Цель работы

Цель работы заключается в:

1) разработке и реализации многоуровневого алгоритма поиска решения задачи k-декомпозиции гиперграфов большой размерности.
2) разработке и реализации ряда эвристик, входящих в состав многоуровневого алгоритма.
3) разработке и реализации алгоритма улучшения k-разбиения гиперграфа, основанного на принципах эволюционно-генетического поиска.
4) разработке методологии работы с атрибутированными гиперграфами и ее реализации в виде программной среды.
5) разработке способа формализации задачи компоновки радиоэлектронной схемы на основе БМК и решении конкретной задачи компоновки
6) проведении ряда вычислительных экспериментов с целью выяснения эффективности работы алгоритма.

Научная новизна

Разработана *концепция «фильтр-вид»*, предполагающая построение специального вида иерархий гиперграфов на основе атрибутивной информации, содержащейся в них, правила изменения и передачи этой атрибутивной информации, а также методы построения и управления такими иерархиями на основе специальных операторов – фильтров.

Разработана гибридная схема решения задачи k-декомпозиции графа на основе многоуровневого и генетического алгоритма. Предложена методика работы и принципы построения улучшающего ГА в структуре многоуровневого алгоритма. Использование ГА в качестве улучшающего алгоритма, позволяет значительно повысить качество получаемого разбиения вследствие значительно более детального исследования области поиска на каждой из стадий многоуровневого алгоритма.

Предложен метод реализации многоуровневого алгоритма декомпозиции гиперграфа на основе концепции «фильтр-вид», позволяющей автоматизировать общие для многоуровневой парадигмы процессы и сосредоточиться

непосредственно на логике решающего алгоритма. Алгоритм позволяет решать задачи достаточно большой размерности за счет ее редуцирования, исследовать обширные области поиска и получать приемлемые результаты за счет применения ГА в сочетании с различными эвристиками, направленными на улучшение качества решения.

Разработана и реализована система машинного представления гиперграфовых структур, а также логика наиболее употребимых при работе с гиперграфами операций. Это позволяет значительно сэкономить время при разработке алгоритмов. Система оформлена в виде библиотеки классов, что делает возможным использовать ее при решении практических и академических задач сторонними исследователями. Исследованы и реализованы различные методики ускорения работы системы представления гиперграфов. Проведены вычислительные эксперименты, подтверждающие эффективность разработанных методов.

Предложен метод сведения экстремальной задачи компоновки БИС на основе БМК к задаче k-декомпозиции гиперграфа. Выполнена компоновка конкретного радиоэлектронного устройства.

Теоретическая и практическая ценность работы

В качестве технических приложений класса задач декомпозиции гиперграфов могут выступать реальные технические задачи, возникающие на этапе проектирования сложных электронных систем. Подобные задачи также имеют место при проектировании и управлении вычислительными системами, при сегментации машинных программ, при распределении программ между машинами в вычислительных системах, на транспортных сетях, при автоматизации конструкторского синтеза дискретных устройств и т.д.

Программные системы, реализованные в ходе данной работы, не имеют четкой направленности на перечисленные задачи и могут быть использованы при решении целого ряда различных проблем, связанных с гиперграфами и многоуровневыми алгоритмами, как практического, так и академического характера.

Апробация результатов

Результаты диссертации докладывались и обсуждались на 6-ом международном конгрессе по математическому моделированию (Н.Новгород, 2004г.), Всероссийской научно-технической конференции «Информационные системы и технологии» ИСТ-2005 (Н.Новгород, 2005г.), конференции «Технологии Microsoft в теории и практике программирования» (Н.Новгород, 2006), Всероссийской научно-технической конференции «Информационные системы и технологии» ИСТ-2007 (Н.Новгород, 2007г.), на семинарах кафедры информатики и автоматизации научных исследований факультета ВМК ННГУ.

Кроме того, результаты диссертационной работы прошли апробацию при выполнении хоздоговорных работ между ННГУ и НИИИС, в которых автор был исполнителем-разработчиком алгоритмов и автором программных реализаций

функциональных блоков систем. Это хоздоговорные работы: «Оптимальная трассировка кабельных соединений в монтажном шкафе» (2005г.), «Исследование методов и разработка системы интеллектуальной поддержки этапов конструкторского проектирования РЭА» (2006г.), «Разработка и реализация программного средства трассировки БИС с одним слоем металлизации»(2007г.), «Разработка и реализация диалоговых программных средств решения задач размещения элементов БИС» (2008г.).

Структура и объем работы

Работа состоит из введения, пяти глав, заключения, списка литературы и приложений. Общий объем работы составляет 124 страницы. Список литературы составляет 145 наименований. Основные результаты излагаются в главах 2, 3 и 5.

Краткое содержание работы

Во **введении** обосновывается актуальность графовых и гиперграфовых проблем, указывается место многоуровневых и генетических алгоритмов в общем классе методов оптимального проектирования. Обсуждаются преимущества и недостатки, а также основные сферы применимости этих алгоритмов в сравнении с традиционными "классическими" методами оптимизации.

В **главе 1** ставится задача k-декомпозиции гиперграфа и приводится обзор точных и эвристических методов и подходов к решению данной задачи, существующих в данный момент, с кратким анализом их характеристик.

Проблема k-разбиения гиперграфа H=(V,E), где V-множество вершин, E – множество гиперребер, состоит в разбиении множества вершин V по k подмножествам $V_1,\dots V_k$, таким образом, чтобы удовлетворялся набор ограничений, называемых декомпозиционными ограничениями, а оптимизируемая функция (функция цели, критерий оптимальности, функционал), определенная над разбиением, принимала экстремальное значение. Разбиение предполагает, что множества $V_1,\dots V_k$ попарно не пересекаются и при объединении составляют исходное множество V:

$$V_i \cap V_j = \varnothing,\ \ i,j = \overline{1,k},\ \ i \neq j,$$
$$\bigcup_{i=1}^{k} V_i = V. \qquad (1)$$

Подмножества разбиения определяют подграфы разбиения. Значение k фиксировано и задается как параметр задачи.

Определим декомпозиционные ограничения так, чтобы вес каждого подграфа разбиения $w(V_i) = \sum_{v_q \in V_i} w(v_q)$, лежал в следующих пределах:

$$L_i \le w(V_i) \le U_i, i = \overline{1,k},$$
L_i и U_i – минимальный и максимальный вершинные веса подграфов разбиения. (2)

В ограничениях (2) константы L_i и U_i определяют пределы, в которых могут варьироваться веса подграфов разбиения.

Критерии оптимальности, как и декомпозиционные ограничения, определяются спецификой конкретной задачи. Один из наиболее часто используемых критериев – минимизация веса сечения. Вес сечения определяется суммой весов ребер, которые целиком не принадлежат ни одному подграфу разбиения:

$$W_c = \sum_{e_c \in E_c} w(e_c) \to \min,$$

$$E_c = \left\{ e \in E \mid \exists v_i, v_j \in e : v_i \in V_a, v_j \in V_b, a \neq b, \quad a = \overline{1,k}, b = \overline{1,k} \right\} \qquad (3)$$

Задачу (1)-(3) будем называть задачей k-разбиения гиперграфа. Возможны различные расширения проблемы (1)-(3). Например, задача может иметь несколько функций цели (многокритериальный случай), тогда решение состоит в нахождении Парето-области решений.

Глава 2 посвящена обзору теоретических основ гипеграфовых структур. В данной главе вводится ряд понятий, необходимых описания манипуляций с гиперграфами. В этот ряд входят как базовые понятия, такие как вершина, ребро, гиперграф, так и более сложные, такие как части и куски гиперграфа, подграфы, суграфы и т.д. Также описывается ряд базовых операций над гиперграфом и элементами гиперграфа, такие как введение, удаление и стягивание вершин и гиперребер, расщепление вершин и ребер, удаление частей гиперграфа и различные бинарные операции с гипеграфами.

Отдельное место в главе занимает проблема визуализации гиперграфа. Визуализация сложных концептуальных структур, отражающая особенности их внутренней организации, занимает ключевое место во многих приложениях в науке и технике. Гиперграф – это абстрактная структура, которая используется для моделирования информации. Системы из объектов и групповых связей между ними естественно моделируются гиперграфами, а их грамотная визуализация обеспечивает дополнительной информацией исследователя в понимании принципов организации объекта исследования. Поэтому многие прикладные системы, например, в сфере проектирование радиоэлектронной аппаратуры, требуют понятного изображения объекта исследования. В главе рассматривается несколько методов визуализации гипеграфа, это: кенигово представление, диаграмма Венна и группа шинно-блочных методов. Все эти методы обладают достоинствами и недостатками, описанными в виде сравнительной таблицы. Выбор того или иного метода зависит от преследуемых при визуализации целей.

Кроме набора базовых понятий и операций в главе описана так называемая концепция «фильтр-вид» и лежащие в ее основе понятия атрибута, фильтра, вида и основания. Эта концепция расширяет классическое определения веса элемента гиперграфа, вводя понятие атрибута, и предлагает общий аппарат для построения многоуровневых алгоритмов над гипеграфами. Это связано с тем, что в реальных задачах в гиперграфовые модели приходится добавлять разнообразную дополнительную информацию, причем часто нечислового характера. В силу этого

в действительности исследователям приходится работать с помеченными гиперграфами. Задавая на вершинах и гиперребрах гиперграфа $H=(V,E)$ отображения $\alpha_H: V\cup E\to M$, где M – некоторое семейство непустых (необязательно различных) подмножеств множества A, и $\omega_H:(V\cup E)\times A\to\mathfrak{R}$ получим *помеченный* (*взвешенный*) *гиперграф* (H,α,ω). Элементы множества A называются *метками или атрибутами*. В дальнейшем четверкой (V,E,α,ω) будем обозначать помеченный гиперграф $H=(V,E,\alpha_H,\omega_H)$.

Если отображение α_H ставит в соответствие каждому элементу гиперграфа действительное число (вектор из действительных чисел), то (H,α,ω) будем называть *взвешенным гиперграфом*.

Рассмотрим понятие атрибута подробнее. В общем случае α_H отображает вершину $v\in V$ в набор атрибутов $\alpha_H(v)\subseteq A$, приписанных этой вершине, и гиперребро $e\in E$ в набор атрибутов $\alpha_H(e)\subseteq A$, приписанных этому гиперребру.

Выше было введено отображение $\omega_H:(V\cup E)\times A\to\mathfrak{R}$, которое каждой паре значений: элементу гиперграфа и его атрибуту ставит в соответствие *значение атрибута*. Здесь $\mathfrak{R}$ – это множество всевозможных значений атрибутов. Допустим, каждый элемент схемы характеризуется собственными габаритами: ширина и высота. В гиперграфовой модели этой схемы каждой вершине $v\in V$ будет приписан набор $\alpha_H(v)$, состоящий из двух атрибутов «ширина» и «высота», а отображение $\omega_H(v,\alpha_i)$, где $\alpha_i\in\alpha_H(v)$, укажет численное значение габарита.

Будем говорить, что элемент c гиперграфа H *имеет атрибут* α_i *со значением* β_j, если $\alpha_i\in\alpha_H(c)$ и $\beta_j=\omega_H(c,\alpha_i)$. Соответственно, будем говорить, что элемент c гиперграфа H *имеет атрибут* α_i *без значения*, если $\alpha_i\in\alpha_H(c)$ и отображение $\omega_H(c,\alpha_i)$ на указанном наборе значений не определено. Набор атрибутов $\alpha_H(c)$ и отображение $\omega_H(c)$: $\alpha_H(c)\to\mathfrak{R}$ будем называть *атрибутивной информацией* элемента $c\in H$.

Можно привести массу алгоритмов, которые в процессе работы над графом не меняют его структуру, но вводят в графовую модель дополнительную информацию. Подобная информация укладывается в графовую модель в виде специальных меток, которые приписываются элементам графа. Примером подобных алгоритмов могут служить алгоритмы раскраски графа. В этом случае информация принадлежности цветовому классу элемента графа носит глобальный характер и фактически в этой информации кодируется решение задачи. В других алгоритмах подобная дополнительная информация носит временный или вспомогательный характер. Так, например, в волновых алгоритмах схема распространения волны строится на приписывании специальных пометок для уже рассмотренных элементов графа.

Очевидно, что операции, позволяющие менять атрибутивную информацию гиперграфа, являются не менее важным компонентом в общей модели управления гиперграфом. Введем операции над атрибутами элементов гиперграфа.

Пусть имеется помеченный гиперграф $H=(V,E,\alpha,\omega)$. *Добавление атрибута* $\alpha_0 \in A$ без значения к элементу $c \in V \cup E$ предполагает переход к новому помеченному гиперграфу $H'=(V,E,\alpha',\omega)$, где $\alpha'_H(c)=\alpha_H(c) \cup \{\alpha_0\}$. *Добавление атрибута* $\alpha_0 \in A$ со значением $\beta_0 \in \mathfrak{R}$ к элементу $c \in V \cup E$ предполагает переход к новому помеченному гиперграфу $H'=(V,E,\alpha',\omega')$, где $\alpha'_H(c)=\alpha_H(c) \cup \{\alpha_0\}$, $\omega'_H(c,\alpha_0)=\beta_0$.

Противоположной операцией добавлению атрибута является операция удаления атрибута. *Удаление атрибута* $\alpha_0 \in A$ элемента $c \in V \cup E$ предполагает переход к новому помеченному гиперграфу $H'=(V,E,\alpha',\omega)$, где $\alpha'_H(c)=\alpha_H(c)/\{\alpha_0\}$.

Важно не только добавлять и удалять атрибуты, но и менять значение атрибута с одного на другое. *Изменение значения атрибута* $\alpha_0 \in A$ на значение $\beta_0 \in \mathfrak{R}$ для элемента $c \in V \cup E$ предполагает переход к новому помеченному гиперграфу $H'=(V,E,\alpha,\omega')$, где $\omega'_H(c,\alpha_0)=\beta_0$. Удобно ввести специальное значение атрибута – null. Будем считать, что если атрибут имеет значение null, то этот атрибут не имеет значения. Т.е. если $\omega_H(c,\alpha_0)$=null, то для элемента c гиперграфа H атрибут задан без значения. В этом случае, операция удаления значения атрибута формально сводится к операции изменения значения атрибута.

Структура многоуровневых алгоритмов (описана в главе 3) предполагает наличие нескольких достаточно общих с точки зрения описания и реализации операций. В частности, любой многоуровневый алгоритм над гиперграфом подразумевает стадию редукции размерности задачи, в ходе которой выполняется ряд действий по удалению избыточной информации из гиперграфовой модели. Было решено унифицировать такого рода операции, что привело к появлению аппарата фильтров, использование которых, однако, не ограничено конструированием многоуровневых алгоритмов, т.к. фильтры представляют собой достаточно универсальный механизм выявления тех или иных структурных особенностей гиперграфа. Итак, под *фильтром* понимается оператор Ф, на вход которого подается гиперграф H, а на выходе получается новый гиперграф ФH. Исходный гиперграф назовем гиперграфом-прообразом. Кроме исходного гиперграфа H фильтр на вход получает *условие фильтрации*, которое можно представить в виде отображения $\varphi: A \times \mathfrak{R} \to \{0,1\}$. Отображение φ каждой паре («атрибут», «значение») ставит в соответствие либо 0, либо 1. Значение 1 подразумевает, что элемент исходного гиперграфа, имеющий данный атрибут с данным значением, будет профильтрован – т.е. подвержен работе оператора фильтрации. Таким образом, отображение $\varphi: A \times \mathfrak{R} \to \{0,1\}$ для гиперграфа H однозначно определяет множество элементов φH гиперграфа, для которых будет выполняться условие фильтрации:

$$\varphi H = \{c \in V \cup E \mid \exists\, \alpha_i \in \alpha_H(c),\ \varphi(\alpha_i,\ \omega_H(c,\alpha_i))=1\}.$$

Будем обозначать через φHV – множество вершин гиперграфа H, для которых выполняется условие фильтрации φ, φHE – множество гиперребер гиперграфа H, для которых выполняется условие фильтрации φ. Таким образом, $\varphi H = \varphi HV \cup \varphi HE$.

Элементы, для которых НЕ выполняется условие фильтрации, “переходят” в новый гиперграф φH, причем набор атрибутов элемента и их значений в гиперграфе φH будет таким же, как и в исходном гиперграфе H:

$$\alpha_H(c)=\alpha_{\Phi H}(c) \text{ и } \omega_H(c,\alpha_i)=\omega_{\Phi H}(c,\alpha_i) \text{ для } \forall\alpha_i\in\alpha_H(c), \text{ где } c\in H/\varphi H^1.$$

Предлагается *два типа фильтров*: скрывающие фильтры и объединяющие фильтры. Для обозначения фильтров примем следующую *форму записи*:

ФH=***filter***{ **множество_элементов** [| **разбиение_на_классы**] }H.

Здесь ФH – результат работы фильтра; вместо ***filter*** указывается либо ***hide*** – скрывающий фильтр, либо ***join*** – объединяющий фильтр; **множество_элементов** описывает множество элементов, удовлетворяющих условию фильтрации; **разбиение_на_классы** (может быть опущено) описывает признак, по которому будет происходить разбиение множества элементов на классы, каждый класс будет подвержен действию, указанному в ***filter***.

В общем случае *скрывающий фильтр* на входе принимает гиперграф H, а на выходе получает новый гиперграф ФH = H/φH. Таким образом, результатом работы скрывающего фильтра есть исходный гиперграф без элементов, удовлетворяющих условию фильтрации.

Другая группа фильтров – объединяющие фильтры. Основная идея *объединяющих фильтров* заключается в применении операций стягивания элементов исходного гиперграфа H, удовлетворяющих условию фильтрации φ.

В работе описаны подклассы данных классов фильтров, их формальное описание и принципы действия, работа фильтров показана на примерах.

Аппарат фильтров подразумевает создание из исходного гиперграфа другого, который обладает теми или иными структурными особенностями, необходимыми исследователю. Исходный гиперграф и гиперграф, полученный в результате работы фильтра – два несвязанных между собой гиперграфа. Однако, часто, особенно при использовании фильтров в составе многоуровневых алгоритмов, необходимо передавать информацию из одного гиперграфа в другой; под информацией здесь подразумевается атрибутивная информация. Будем говорить, что два гиперграфа, способные передавать друг другу атрибутивную информацию имеют атрибутивную связь. Пусть имеется помеченный гиперграф H=(V,E,α,ω), задано условие фильтрации φ для некоторого фильтра Ф, осуществляющего перенос атрибутивной информации от исходного графа H к результирующему ФH. Представим работу фильтров на уровне отдельных элементов. Очевидно, что после работы фильтра Ф для любого элемента c гиперграфа ФH можно указать непустое множество элементов $\Phi^{-1}(c)$ исходного гиперграфа H, которые оказали влияние на формирование атрибутивной информации $\alpha_H(c)$ и $\omega_H(c)$.

---

[1] Под операцией вычитания гиперграфов здесь понимается операция, состоящая из двух частей: слабого удаления вершин ФHV=HV\φV и слабого удаления гиперребер ФHE=HE\φE, таким образом ФH=(HV\φ, HE\φE)

Результат работы фильтра $\Phi H=(\Phi V,\Phi E,\alpha_\Phi,\omega_\Phi)$ назовем *видом гиперграфа* H, если любая операция по изменению атрибутивной информации для любого элемента из ΦH приводит к изменению атрибутивной информации соответствующих элементов из H по следующему правилу: добавление (изменение) атрибута $\alpha_0\in A$ со значением $\beta_0\in\mathfrak{R}$ для элемента $c\in\Phi H$ приводит к добавлению (изменению) атрибута $\alpha_0\in A$ со значением $\beta_0\in\mathfrak{R}$ для всех элементов из $\Phi^{-1}(c)\subseteq H$, где $\Phi^{-1}(c)$ - множество элементов исходного гиперграфа H, которые оказали влияние на формирование атрибутивной информации $\alpha_\Phi(c)$ и $\omega_\Phi(c)$

В тоже время, помеченный гиперграф $H=(V,E,\alpha,\omega)$ будем называть *основанием вида* $\Phi H=(\Phi V,\Phi E,\alpha_\Phi,\omega_\Phi)$, если любая операция по изменению атрибутивной информации для любого элемента из H приводит к изменению атрибутивной информации соответствующих элементов из ΦH по следующему правилу: добавление (изменение) атрибута $\alpha_0\in A$ со значением $\beta_0\in\mathfrak{R}$ для элемента $c\in H$ приводит к добавлению (изменению) атрибута $\alpha_0\in A$ со значением $\beta_0\in\mathfrak{R}$ для всех элементов $\Phi(c)\subseteq\Phi H$.

Пусть имеется пара помеченных гиперграфов H_1 и H_2. Будем говорить, что H_2 *атрибутивно зависим* от H_1 и обозначать $H_1\rightarrow H_2$ (либо $H_2\leftarrow H_1$), если либо H_2 является видом H_1, либо H_1 является основанием H_2.

Будем говорить, что H_1 и H_2 *атрибутивно связаны* и обозначать $H_1\leftrightarrow H_2$, если $H_1\rightarrow H_2$ и $H_1\leftarrow H_2$.

Очевидны следующие утверждения.

- Суперпозиция фильтров Ф есть фильтр, т.е. если Φ_1 и Φ_2 – фильтры, тогда и $\Phi_1\Phi_2$ – тоже фильтр.
- Если $H_1\rightarrow H_2$ и $H_2\rightarrow H_3$, тогда $H_1\rightarrow H_3$.
- Если $H_1\leftrightarrow H_2$ и $H_2\leftrightarrow H_3$, тогда $H_1\leftrightarrow H_3$.
- Отношение $\leftrightarrow$ является отношением эквивалентности.

Число суперпозиций фильтров Ф, примененных к гиперграфу H назовем порядком вида гиперграфа H, если между ними имеется отношение гиперграф–вид. Таким образом, H – это вид нулевого порядка гиперграфа H; ΦH – вид первого порядка гиперграфа H, если $H\leftarrow\Phi H$; $\Phi\Phi H$ – вид второго порядка гиперграфа H, если $H\leftarrow\Phi\Phi H$.

Глава 3 описывает основные принципы построения и функционирования многоуровневого алгоритма декомпозиции гиперграфа, обосновывает необходимость применения многоуровневых схем при решении задач большой размерности, а также описывает способ организации многоуровневых методов с применением концепции «фильтр-вид».

Применение эвристик, таких как многоуровневые алгоритмы связано с тем, что исследуемые задачи декомпозиции гиперграфа NP-трудны и на практике, как правило, имеют большие размерности. Алгоритмы, получающие точные решения таких задач, оказываются неприменимыми в силу большой вычислительной сложности. Известно множество попыток сведения подобных задач к менее

сложным. Условно методы решения задач декомпозиции больших порядков можно разбить на две группы. В первую группу отнесем методы, которые осуществляют переход к более простым задачам, не используя понижение размерности задачи. Например, бисекционные алгоритмы предполагают рекурсивную бисекцию графа до тех пор, пока не будет получено необходимое количество подграфов разбиения. Вторую группу составляют методы, которые сокращают вычислительную сложность за счет уменьшения размерности исходной задачи.

Методы второй группы, осуществляющие непосредственное k-разбиение, в целом получают более качественное решение. Например, показано, что решение, полученное рекурсивной бисекцией, может быть вплоть до O(log n), где n – размерность графа, хуже, чем решение алгоритма, осуществляющего непосредственное k-разбиение. Такое положение вещей объясняется тем, что рекурсивный метод не позволяет напрямую оптимизировать функционалы, значение которых зависит от того, каким способом ребра распределены по подграфам разбиения. Второе существенное преимущество – в ходе работы алгоритм второго типа способен более жестко контролировать декомпозиционные ограничения, в то время как бисекционные алгоритмы должны иметь специальные фазы коррекции решения для удовлетворения этим ограничениям, что существенно ограничивает область поиска решения.

В последнее время популярность приобрели так называемые многоуровневые алгоритмы. Ключевая идея многоуровневого алгоритма декомпозиции[1] состоит следующем: вместо того, чтобы строить k-разбиение непосредственно для исходного гиперграфа, сначала строится ряд его приближений (загрублений). Каждое загрубление уменьшает (редуцирует) размерность исходной задачи. Процесс редукции продолжается до тех пор, пока порядок гиперграфа не снизится до сотен или даже десятков. На гиперграфе таких размерностей и отыскивается так называемое начальное разбиение. Полученное решение используется для построения решения для исходной задачи. Это достигается путем выполнения серии переносов решения задачи меньшей размерности на задачу большей размерности с последующим улучшением перенесенного решения. Таким образом, работу многоуровневого алгоритма можно разделить на три фазы: первая - фаза загрубления, когда производится ряд последовательных редукций размерности задачи, вторая - фаза поиска начального разбиения и третья - фаза восстановления решения, когда производится серия последовательных переносов решения задачи меньшей размерности на менее редуцированную задачу с последующим улучшением спроецированного решения. Работа алгоритма для задачи 3-декомпозиции схематично показана на рис.1.

[1] В дальнейшем будем употреблять термин «декомпозиция» как синоним термина «разбиение», за исключением специально оговоренных случаев.

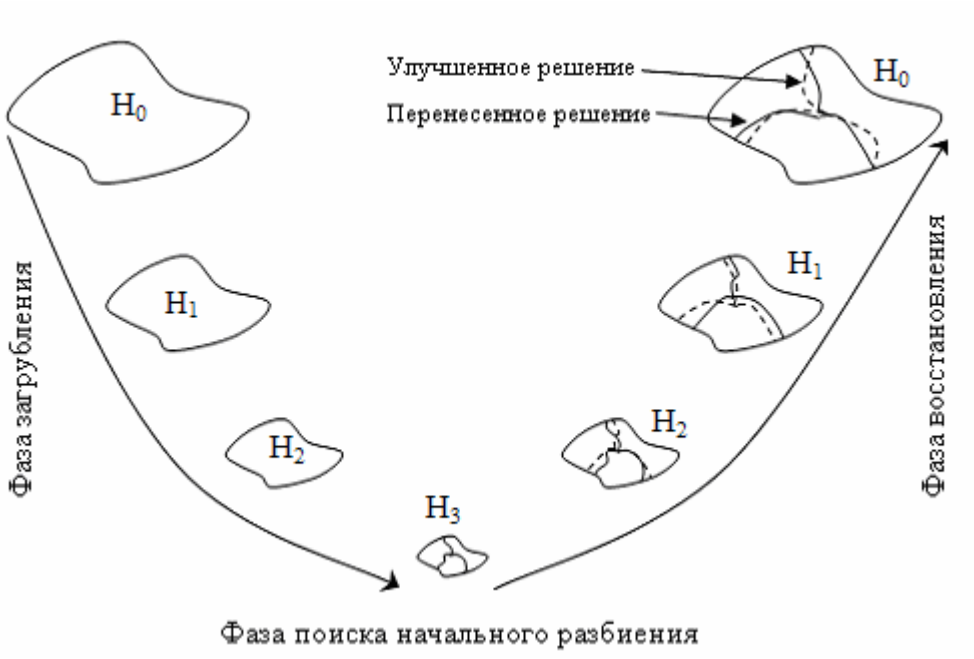

Рисунок 1. Многоуровневый алгоритм 3-разбиения гиперграфа

Загрубление гиперграфа – это удаление избыточной информации, которая, с одной стороны, не является необходимой для поиска решения на данном этапе, а с другой стороны ухудшает характеристики решающего алгоритма.

Рассмотрим фазу загрубления на примере (рис. 1). Исходная задача 3-разбиения поставлена над гиперграфом H_0. В ходе фазы загрубления строится редуцированный гиперграф H_1, получаемый из H_0 удалением избыточной информации[1], и над полученным гиперграфом ставится новая задача 3-декомпозиции. Удаление избыточной информации из гиперграфа и постановка новой задачи должно производиться таким образом, чтобы по решению новой задачи могло быть построено решение исходной задачи. Если это условие выполнено, то процесс удаления избыточной информации из гиперграфа назовем загрублением гиперграфа, а процесс построения загрубленного гиперграфа и постановки над ним новой декомпозиционной задачи назовем редуцированием задачи; построение решения исходной задачи по решению вновь построенной будем называть переносом (проекцией) решения; исходную задачу будем называть задачей-прообразом, а вновь полученную – задачей-редуктантом. Таким образом, на первом этапе фазы загрубления мы получили из задачи-прообраза, определенной над гиперграфом H_0, задачу-редуктант, определенную над загрубленным гиперграфом H_1, обладающую меньшей размерностью. В дальнейшем для простоты будем обозначать декомпозиционную задачу, определенную над гиперграфом H_m, «задача H_m».

На втором этапе повторим процесс редуцирования, используя задачу H_1 в качестве задачи-прообраза, и получим задачу H_2. Следующий этап позволит

[1] Критерий определения избыточности специфичен как для каждой конкретной задачи, так и для конкретного многоуровневого алгоритма, поэтому здесь не уточняется.

получить задачу H_3, обладающую достаточно малой размерностью, чтобы для нее можно было отыскать разбиение.

Описанное выше условие редуцирования задачи, при котором решение, полученное для задачи-редуктанта, может быть спроецировано на задачу-прообраз, позволяет утверждать, что решение задачи H_m может быть использовано для построения решения задачи H_n, n < m. Действительно, решение H_m можно спроецировать на H_{m-1}, затем, полученное решение H_{m-1} спроецировать на H_{m-2}, и так далее до H_n. Таким образом, задача H_m, по сути, является задачей-редуктантом H_n, а H_n в свою очередь – задачей-прообразом H_m. Число m-n назовем глубиной редукции. Задачу H_m будем называть задачей-редуктантом задачи H_n глубины m-n. Очевидно, задача H_m является задачей-редуктантом исходной задачи глубины m. Так, к примеру, на рис.1 задача H_2 является задачей-редуктантом задачи H_0 глубины 2 и задачей-редуктантом глубины 1 для H_1. Возможность построения решения исходной задачи по решению любой задачи-редуктанта позволяет говорить о решении задачи-редуктанта как о решении исходной задачи. Т.е. можно говорить о процессе построения начального разбиения (для примера на рис.1 это решение задачи H_3) как об отыскании некоего начального решения исходной задачи.

Для рассматриваемых задач декомпозиции гиперграфа процесс загрубления строится на выделении и объединении групп вершин, которые целесообразно всегда определять целиком в одну из компонент разбиения. Естественно, что в такой ситуации на группы должны быть наложены некоторые ограничения, например, вес группы не должен превышать вес компоненты разбиения, иначе алгоритм поиска начального разбиения не сможет найти решение.

Глава описывает ряд схем загрубления гипеграфа, которые обладают различными характеристиками и применимы для различных видов гиперграфов.

Поскольку размерность задачи-редуктанта самой большой глубины невелика, то ряд алгоритмов, способных получить начальное разбиение в многоуровневом алгоритме достаточно широк. В случае, когда самый загрубленный гиперграф содержит десяток вершин возможно применение переборных методов. Однако, в силу ограничений схем загрубления, редукция гиперграфов больших размеров (десятки и сотни тысяч) до размерностей в несколько десятков часто невозможна. Поэтому можно использовать менее ресурсоемкие алгоритмы, например, т.н. region-growing алгоритмы или алгоритмы, строящие случайные разбиения с последующим их улучшением с помощью эвристик типа алгоритма Кернигана-Лина (Kernigan and Lin, KL), Фидуччиа-Мэтьюза (Fiduccia and Mattheyses, FM), спектральных методов, табу-поиска или методов simulated annealing, модифицированных для работы с гиперграфами или описанного ниже генетического улучшающего алгоритма.

Решение задачи H_3 может быть спроецировано на задачу H_0, в этом состоит суть фазы восстановления решения. Поскольку вычислительная сложность алгоритмов загрубления, как правило, несравнимо меньше вычислительной сложности декомпозиционных алгоритмов, а размерность задачи-редуктанта

самого высокого порядка на несколько порядков ниже размерности исходной задачи, такой подход дает существенный выигрыш по вычислительной сложности по сравнению с «классическими» декомпозиционными алгоритмами.

Однако, приведенная схема решения, когда решение, полученное для задачи-редуктанта самого высокого порядка, проецируется на исходную задачу, обладает одним существенным недостатком. Потеря информации в процессе редуцирования задачи не может не сказаться на качестве решения, построенного на этапе построения начального разбиения. Поэтому имеет смысл попытаться улучшить это решение в ходе фазы восстановления. Действительно, при переходе от задачи-редуктанта на задачу-прообраз появляется информация, отброшенная при редукции, которая может быть использована для улучшения спроецированного решения. Таким образом, каждый этап фазы восстановления решения состоит из двух подэтапов – проецирования решения задачи-редуктанта на задачу-прообраз и улучшение спроецированного решения. Описанный в работе алгоритм использует жадную методику улучшения разбиения, основанную на т.н. процессе локализации гиперребер, когда «тяжелые» ребра, попавшие в сечение гиперграфа целиком переносятся в одну из компонент разбиения.

Кроме того, в **главе 3** описан принцип построения улучшающего генетического алгоритма. Алгоритмы улучшения решения задачи декомпозиции гиперграфа, как правило, основаны на жадном принципе и работают достаточно быстро. Однако, они не содержат никаких средств преодоления локальных экстремумов, что сказывается на качестве находимых ими решений.

Для того чтобы значительно расширить исследуемое пространство области поиска задачи предлагается сконструировать гибридный алгоритм улучшения решения, основанный на предложенной жадной методике и эксплуатирующий эволюционно-генетический подход.

Классический генетический алгоритм (ГА), в том числе работающий с графовыми или гиперграфовыми моделями, подразумевает манипулирование так называемыми кодировками (генотипами). Каждая кодировка представляет собой некоторое представление решения задачи в форме, удобной для применения различных операторов ГА. Каждая кодировка имеет оценку, называемую приспособленностью, которая характеризует качество решения, соответствующего данной кодировке. В результате процесса, моделирующего эволюцию набора кодировок, определяется «кодировка-победитель», обладающая наилучшей приспособленностью, которой соответствует решение задачи с наивысшим качеством.

Поскольку классический ГА – это решающий алгоритм, а цель – построить улучшающий, то, очевидно, необходимо изменить состав информации, закодированный в генотипе. Предлагается кодировать не само решение, а некий сценарий получения нового решения из существующего.

Описанный выше жадный алгоритм улучшения решения, основанный на локализации ребер, на каждом шаге принимает решение, руководствуясь наибольшим сокращением веса сечения, т.е. выбирает одно из нескольких

возможных направлений движения по ОП, оставляя остальные направления неисследованными. Внедрение генетического алгоритма в качестве управляющей схемы, принимающей решение, в каком направлении двигаться по ОП на каждом шаге, позволяет существенно расширить исследуемую область ОП.

Для реализации предложенного подхода модифицируем жадный алгоритм таким образом, чтобы решение о том, какой шаг сделать в ОП принимал не он, а ГА в соответствии с маршрутом ОП, закодированном в генотипе. Фактически, модифицированный жадный алгоритм формирует набор направлений, в которых можно двигаться по ОП с целью улучшения решения, а ГА принимает решение в каком именно направлении сделать шаг. Такой синтез ГА и жадных техник позволяет построить достаточно быстрый улучшающий алгоритм, лишенный недостатка жадных алгоритмов скатываться в локальный экстремум.

На рис.2 схематично изображен процесс движения по ОП; точки обозначают решения, через которые проходит маршрут, стрелки – возможные направления движения на каждом шаге; на каждом шаге выбирается одно из направлений определяемое соответствующим геном.

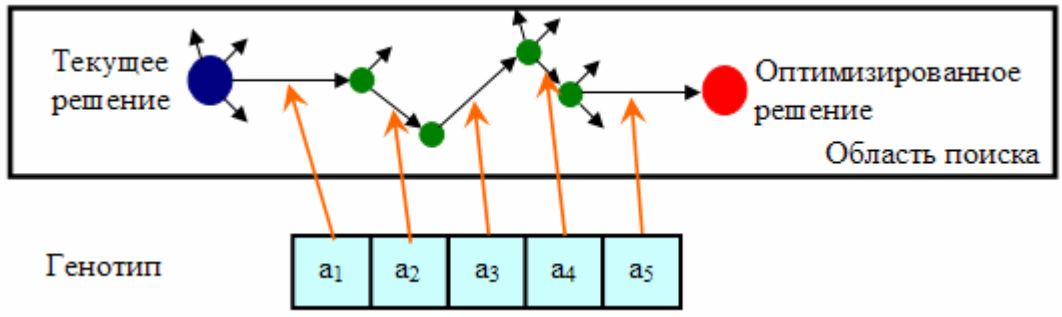

Рис 2. Маршрут поиска закодирован в генотипе

В главе описана адаптация этой общей схемы построения улучшающего ГА к жадному алгоритму улучшения разбиения, основанному на переносах групп вершин. Для этого предложены символьная модели и оператор декодирования генотипа в решение задачи.

В **главе 4** содержится описание программной реализации принципов и понятий, изложенных в предыдущих главах. Условно глава может быть поделена на четыре части: описание реализации среды работы с гипеграфами, описание реализации концепции «фильтр-вид», описание реализации многоуровневого алгоритма декомпозиции гиперграфа с элементами эволюционно-генетического поиска и вычислительный эксперимент.

Первая часть главы кроме непосредственного описания структур хранения и работы с гиперграфами содержит изложение приемов решения технических проблем, возникающих при реализации среды работы с гиперграфами, таких как когерентность (целостность) хранимых данных и производительность.

Вторая часть четвертой главы содержит описание механизмов обработки гиперграфов, составляющих реализацию концепции «фильтр-вид». Описываются сущности, обеспечивающие данную функциональность, их обязанности, контракты

и взаимоотношения. Рассматриваются проблемы, возникающие при работе фильтров и видов, в частности проблема бесполезных вычислений в многоуровневых иерархиях видов. Основные и наиболее сложные для понимания аспекты функционирования рассмотрены на примерах.

Третья часть главы описывает абстракции и конкретные реализации сущностей, обеспечивающих конструирование и функционирование многоуровневого алгоритма декомпозиции. Указаны проблемы, возникающие при реализации и работе многоуровневого алгоритма, описаны способы решения этих проблем. Кроме того, описана абстрактная реализация улучшающего генетического алгоритма и конкретная адаптация этого ГА для нужд многоуровневого алгоритма декомпозиции гиперграфа.

Для реализации гиперграфов и алгоритмов работы с ними был выбран объектно-ориентированный подход. Такой выбор обусловлен двумя факторами: большой инженерной сложностью реализуемой системы и тем, что большинство современных средств программирования, обладающих достаточной для написания данной системы функциональностью, поддерживают объектно-ориентированный подход, и было бы неразумно не воспользоваться предлагаемыми возможностями. Большая инженерная сложность означает, что код, представляющий систему, во-первых, имеет значительный объем, т.к. реализует обширную функциональность, во-вторых, достаточно сложен по структуре. Описание такой системы с помощью, например, структурного подхода, сулило бы разрастание и так достаточно большого объема кода и, как следствие, увеличение вероятности появления различного рода ошибок. Кроме того, объекты предметной области достаточно легко идентифицируются и абстрагируются в классы, что значительно упрощает проектирование с помощью объектно-ориентированной парадигмы. Почти все современные среды разработки и языки программирования поддерживают объектно-ориентированную парадигму. Более того, выбранный для реализации системы язык C# полностью объектно-ориентирован, и, хотя и допускает некоторые формы структурного программирования, все же значительно более удобен при использовании ООП.

Последняя часть главы содержит описание результатов вычислительного эксперимента решения задачи декомпозиции гиперграфа для набора тестовых задач и различных конфигураций многоуровневого алгоритма.

В **главе 5** изложены основные принципы и виды процессов конструирования больших и сверхбольших интегральных схем (БИС и СБИС). Более подробно рассматривается одна из задач проектирования БИС на основе БМК. Классический вариант конструкции БМК представлен на рис. 3. Такая БИС содержит центральную часть с матрицей регулярно расположенных базовых ячеек (БЯ) и каналы трассировки связей между нескоммутированными элементами БМК. По периферии расположены ячейки ввода-вывода и контактные площадки, предназначенные для организации ввода-вывода сигналов. Каждая БЯ может представлять собой как функционально законченный узел, выполняющий конкретную операцию, например И-НЕ, так и набор нескоммутированных

элементов – транзисторов и резисторов, из которых формируется тот или иной библиотечный элемент.

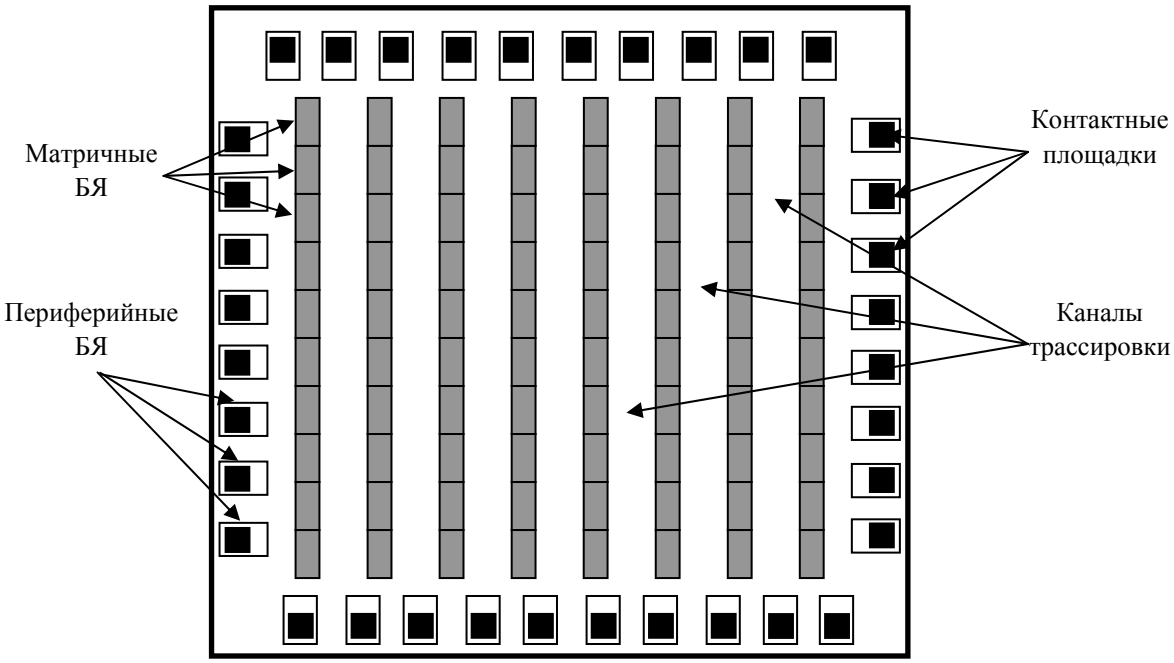

Рисунок 3. Типовая структура БМК

Будем рассматривать процесс проектирования БМК, который подразумевает то, что БЯ – это набор нескоммутированных элементов. Каждая БЯ или группа БЯ путем коммутации внутренних элементов может быть превращена в один из библиотечных элементов, из которых состоит функциональная схема заказчика. Таким образом, проектирование БМК состоит из двух этапов – определения какие именно БЯ будет занимать тот или иной элемент функциональной схемы и трассировки соединений между этими элементами на кристалле. Эти два этапа формализуются в задачи размещения и трассировки соединений.

Суть задачи трассировки – в проведении всех трасс между элементами функциональной схемы на кристалле. Трассы при этом могут проходить только в специально отведенных зонах БМК, называемых каналами трассировки. Очевидно, что, поскольку место для трассировки ограничено, то с увеличением количества связей между элементами вероятность успешной прокладки всех трасс уменьшается. Кроме того, успех решения задачи трассировки напрямую зависит от того, каким именно образом элементы функциональной схемы (ФС) расположены на кристалле. Поэтому основная цель решения задачи размещения – расположить элементы ФС так, чтобы максимизировать вероятность успешного решения задачи трассировки.

Нетрудно заметить, что при размещении элементов ФС существуют ситуации, возникновение которых резко снижает вероятность успешной трассировки. К таким ситуациям можно отнести возникновение так называемых «горизонтальных связей». Рассмотрим фрагмент БМК с размещенными

элементами ФС (рис 4). Пусть между элементами 5 и 15 существует связь. Это означает, что трассировщик должен будет проложить горизонтальную трассу через весь канал трассировки. Такое расположение трассы 5-15 приведет к тому, что возможности прокладки других трасс в этом канале станут намного меньше. В частности, в этом канале не смогут быть проложены любые трассы, соединяющие элемент выше горизонтальной трассы и ниже, например трассы 4-9 и 3-18.

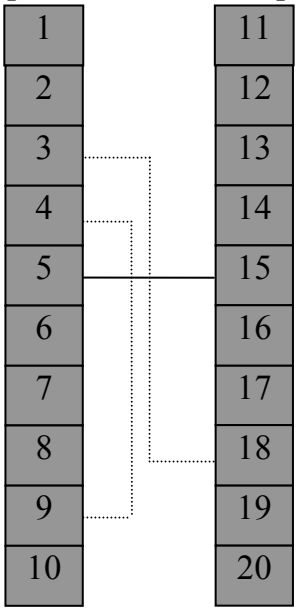

Рисунок 4. Пример нежелательного размещения

Очевидно, что для предотвращения подобных ситуаций необходимо размещать элементы ФС, имеющие связи в пределах одного столбца матричных БЯ, т.е. необходимо максимально сократить число связей между элементами, расположенными на разных столбцах матричных БЯ. Таким образом, одним из этапов решения задачи размещения может являться решение задачи компоновки элементов ФС в ряды с минимизацией связей между рядами.

Как уже было сказано, гиперграф является удобным средством моделирования электронных схем, в том числе и интегральных. Это связано с тем, что поскольку гиперграф является обобщением классического графа, то значительная часть методов и приемов теории графов может быть перенесена на гиперграфы, очень часто с сохранением семантики той или иной операции. С другой стороны, гиперграфы – более общий аппарат и предоставляют несравненно большие возможности моделирования электронных схем в различных задачах проектирования. В частности, предложенная выше задача компоновки элементов подразумевает достаточно простую формализацию в терминах гиперграфовой модели.

Формализация электронной схемы может быть проведена следующим образом: элементам ФС ставятся в соответствие вершины гиперграфа, а цепям, соединяющим элементы ФС – гиперребра. Рассмотрим небольшую схему (рис. 5) и ее гиперграфовую модель (рис. 6). Пусть мы имеем элементы L_5 и L_6 и соответствующие им вершины гиперграфа v_5 и v_6, тогда в гиперграфе будет существовать гиперребро $\{v_5, v_6\}$. Цепи, естественно, могут соединять более двух

элементов, как например цепь c_2, соединяющая элементы L_1, L_2, L_3 и L_4, в этом случае соответствующее гиперребро будет содержать вершины v_1,v_2,v_3,v_4.

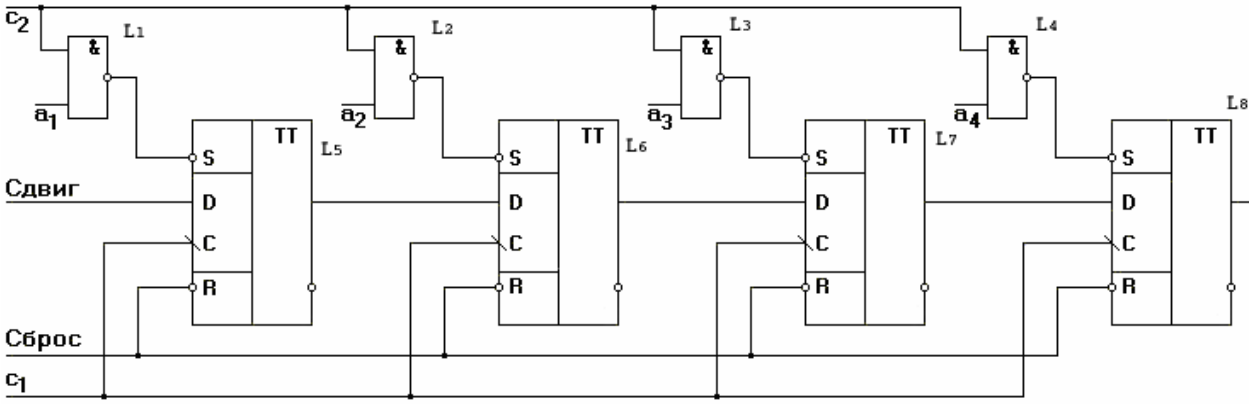

Рисунок 5. Пример функциональной схемы

Теперь каждой вершине гиперграфа припишем вес равный количеству занимаемых соответствующим элементом ФС матричных базовых ячеек БМК. Так, для простых элементов типа И – это число 1, для триггеров – 2.

Предложенная модель позволяет формализовать задачу компоновки, описанную выше, в задачу декомпозиции гиперграфа. При этом:

- подграфы разбиения соответствуют вертикальным рядам матричных БЯ.
- Для каждого подграфа разбиения определяется минимальный суммарный вес, равный нулю, и максимальный суммарный вес, равный количеству БЯ в ряду.
- Вес каждой вершины равен количеству БЯ, занимаемому соответствующим элементом ФС.
- Вес каждого гиперребра равен 1.
- Значение критерия задачи декомпозиции соответствует количеству межрядных связей в решении задачи компоновки.

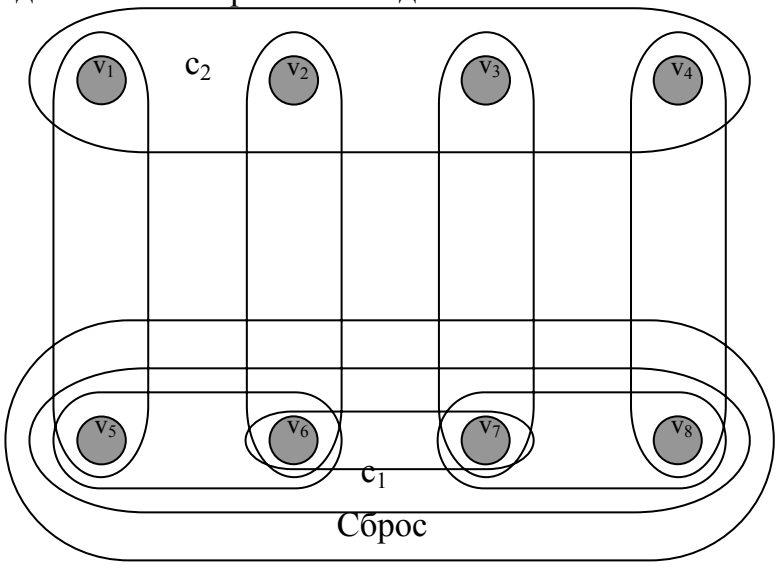

Рисунок 6. Гиперграфовая модель схемы

Решение задачи декомпозиции гиперграфа в терминах исходной задачи компоновки следует интерпретировать следующим образом: попадание какой-либо вершины в подграф разбиения говорит о том, что соответствующий элемент ФС должен быть определен в вертикальный ряд БЯ БМК, который соответствует этому подграфу. Дальнейшее размещение элементов ФС, т.е. определение какие именно БЯ будет занимать тот или иной элемент, выполняется алгоритмом размещения, частью входной информации которого являются данные о том, какому ряду принадлежит тот или иной элемент. Таким образом, задача алгоритма размещения – определение местоположения элемента ФС в априорно определенном алгоритмом компоновки ряду БЯ, что является несравненно более простой задачей, чем полное размещение, т.е. размещение элемента ФС в пределах всего кристалла.

Заключительная часть данной главы содержит описание вычислительного эксперимента по компоновке реальной ИС, основанной на БМК. Данная ИС используется в технологическом процессе ФГУП «ФНПЦ НИИИС им. Ю.Е. Седакова». Схема содержит 55 элементов, организованных в 6 рядов по 76 БЯ, каждый из которых занимает от одной до двадцати двух БЯ и 86 цепей, соединяющих от двух до девяти элементов. Схема имеет решение задачи компоновки с количество горизонтальных связей равным 40, построенное вручную. Для схемы была построена математическая модель в виде атрибутированного гиперграфа и решена задача компоновки с применением многоуровневого алгоритма декомпозиции с элементами эволюционно-генетического поиска. Решение, полученное многоуровневым алгоритмом, содержит 14 горизонтальных связей.

Основные результаты:

1. Описан и реализован в виде среды управления гиперграфами ряд понятий и операций над гиперграфами.
2. Представлены различные способы визуализации гиперграфов, выявлены преимущества и недостатки каждого из способов.
3. Предложена и реализована концепция «фильтр-вид», представляющая собой инструмент анализа структурных особенностей гиперграфа и средство упрощения конструирования алгоритмов над гиперграфами.
4. Поставлена задача k-разбиения гиперграфа с минимизацией веса сечения. Для решения задачи разработан и реализован многоуровневый алгоритм декомпозиции гиперграфа.
5. Предложено несколько схем загрубления гиперграфа и алгоритмов улучшения решения в ходе фазы восстановления.
6. Описаны и реализованы общая концепция улучшающего генетического алгоритма, принципы построения и функционирования.
7. Предложена и реализована адаптация улучшающего ГА для оптимизации k-разбиения гиперграфа.
8. Построен многоуровневый алгоритм декомпозиции гиперграфа с использованием улучшающего ГА.

9. Проанализированы некоторые проблемы, возникающие при работе с гиперграфами и использовании концепции «фильтр-вид». Предложены и реализованы способы преодоления таких проблем.
10. Проведен ряд вычислительных экспериментов на тестовых задачах с априорно известными решениями, а так же выполнена компоновка реальной ИС, основанной на БМК, принимающей участие в технологическом процессе ФГУП «ФНПЦ НИИИС им. Ю.Е. Седакова». Во всех сериях экспериментов многоуровневый алгоритм декомпозиции гиперграфов показал способность получать приемлемые решения за разумное время, что позволяет сделать вывод о пригодности данного алгоритма для решения реальных задач проектирования ИС.
11. Разработан способ формализации задачи компоновки ИС в задачу декомпозиции гиперграфа. Задача компоновки элементов функциональной схемы по k рядам БЯ БМК сводится к задаче k-разбиения гиперграфа.

Публикации

Публикации в изданиях, рекомендованных ВАК РФ

1) Батищев, Д.И. Оптимальная трехмерная трассировка кабелей по кабельным каналам монтажного шкафа. / Д.И. Батищев, С.Е. Власов, Н.В. Старостин, А.В. Филимонов. //Известия СПбГЭТУ "ЛЭТИ". Серия "Информатика управление и компьютерные технологии", Вып. 1, 2004 г., стр. 34-42.
2) Батищев, Д.И. Компоновка радиоэлектронного оборудования по блокам. /Д.И. Батищев, С.Е. Власов, Н.В. Старостин, А.В. Филимонов. //Вестник Нижегородского университета им. Н.И. Лобачевского. Н. Новгород. Изд-во ННГУ им. Н.И. Лобачевского, 2007, стр. 183-189
3) Батищев, Д.И. Многоуровневый генетический алгоритм решения задачи декомпозиции гиперграфа. /Д.И. Батищев, Н.В. Старостин, А.В. Филимонов. //Известия СПбГЭТУ "ЛЭТИ". Серия "Информатика управление и компьютерные технологии", выпуск 2 2007, стр. 3 – 14.
4) Батищев, Д.И. Многоуровневый алгоритм решения задачи компоновки интегральных схем. /Д.И. Батищев, Н.В. Старостин, А.В. Филимонов. //Научно-технический журнал «Системы управления и информационные технологии», г.Воронеж: Изд-во «Научная книга», 2007, с.48-53
5) Батищев, Д.И. Многоуровневая декомпозиция гиперграфовых структур. /Д.И. Батищев, Н.В. Старостин, А.В. Филимонов. //Прилож. к журналу «Информационные технологии» №5(141) 2008, стр.1 - 32

Публикации в прочих изданиях

6) Батищев, Д.И. Эволюционно-генетический подход к автоматическому распараллеливанию программ. / Д.И. Батищев, Н.В. Старостин, А.В. Филимонов, А.В. Лобанов. //Сборник научных статей юбилейной научно-практической конференции «Математика и кибернетика 2003» факультета

ВМК ННГУ и НИИ ПМК, Нижний Новгород: Изд-во ННГУ, 2003 г., стр. 43-48.

7) Filimonov, A.V. Hybrid solving algorithm for uniform graph k-decomposition problem. /A.V. Filimonov, N.V. Starostin. //VI International congress on mathematical modeling/ book of abstract/ September 20-26, 2004, Nizhny Novgorod, University of Nizhny Novgorod, page 353.
8) Старостин, Н.В. Многоуровневый подход к решению задач декомпозиции гиперграфов большой размерности. /Н.В. Старостин, А.В. Филимонов. //Тезисы докладов Всероссийской научно-технической конференции «Информационные системы и технологии» ИСТ-2005, Нижний Новгород, 2005 г., стр. 118-119.
9) Батищев, Д.И. Новый подход к представлению гиперграфовых структур. /Д.И Батищев, С.Е. Власов, Н.В. Старостин, А.В. Филимонов. //Вестник ВГАВТ. Межвузовская серия Моделирование и оптимизация сложных систем. Вып. 14, 2005 г., стр. 67-78
10) Филимонов, А.В. О решении задачи трехмерной трассировки кабелей по кабельным каналам монтажного шкафа. /А.В. Филимонов. //Труды НГТУ. Том 64. Радиоэлектронные и телекоммуникационные системы и устройства. Выпуск 11. С.136-140
11) Старостин, Н.В. Аспекты программной реализации гиперграфов. /Н.В. Старостин, А.В. Филимонов.//Сборник научных статей НГТУ 2005 “Информационные технологии”, том 56, с. 80-89
12) Старостин, Н.В. Применение генетического алгоритма в многоуровневых схемах. /Н.В. Старостин, А.В. Филимонов. //Материалы конференции «Технологии Microsoft в теории и практике программирования», г.Н.Новгород: Изд-во ННГУ, 2006г, с. 282-284
13) Филимонов, А.В. Эволюционно-генетический подход к построению улучшающих алгоритмов. /А.В. Филимонов. //Информационные системы и технологии ИСТ-2007. Материалы международной научно-технической конференции, посвященной 90-летию Нижегородского государственного технического университета. Изд-во НГТУ, 2007, стр. 209-211
14) Старостин, Н.В. Место задачи компоновки в процессе проектирования БМК. /Н.В. Старостин, А.В. Филимонов. //Материалы конференции «Технологии Microsoft в теории и практике программирования», г.Н.Новгород: Изд-во ННГУ, 2007г, с. 271-274

Подписано в печать 3.07.2008. Формат 60×84 1/16.
Бумага офсетная. Печать офсетная. Гарнитура «Таймс».
Усл. п. л. 1. Заказ № 513. Тираж 100 экз.

Отпечатано с готового оригинал-макета в типографии
Нижегородского госуниверситета им. Н.И. Лобачевского.
603000, г. Н. Новгород, ул. Б. Покровская, 37