Раздел II. Автоматизация проектирования

УДК 658.512

Б.К. Лебедев, О.Б. Лебедев

РАЗБИЕНИЕ НА ОСНОВЕ ГИБРИДНОЙ МНОГОУРОВНЕВОЙ АДАПТАЦИИ*

Введение. Весьма популярными алгоритмами разбиения графа $G(X, U)$ на части являются итеративные алгоритмы, требующие для своей работы некоторого начального разбиения [1-3]. Итеративные алгоритмы делятся на детерминированные и вероятностные. В детерминированных алгоритмах изменение разбиения (решения) реализуется на основе четкой, детерминированной зависимости от изменяемого решения. Недостатком является частое попадание в локальный оптимум («локальную яму»). В вероятностных алгоритмах переход к новому решению осуществляется случайным образом. Недостатком алгоритмов, реализующих чисто случайный поиск, является значительная трудоемкость. Один из путей преодоления этой трудности состоит в группировке сильно связанных подсхем в кластеры и дальнейшем объединении этих кластеров в узлы (вершины) перед выполнением алгоритма. Размерность данной задачи, таким образом, значительно уменьшается, что ведет к снижению времени поиска [4].

Недавно был разработан новый класс алгоритмов разбиения гиперграфа, которые основаны на многоуровневой парадигме [5-7]. В этих алгоритмах создан упорядоченный набор последовательно уменьшающихся гиперграфов. Находится разбиение самого маленького гиперграфа, затем это разбиение последовательно проектируется на следующий уровень, и на каждом уровне используется итерационный алгоритм для улучшения качества решения. Эксперименты представленные в [5-7] показывают, что многоуровневая парадигма позволяет существенно улучшить решения за приемлемое время.

Повышение эффективности итерационных алгоритмов связано с разработкой методов, основанных на моделировании естественных процессов. К ним относятся методы моделирования отжига, метод эволюционного моделирования, эволюционной адаптации [8-11]. Особый интерес представляет поисковая адаптация, основанная на использовании обучающихся автоматов, моделирующих поведение объекта адаптации в среде. Трудности использования такого подхода связаны, в первую очередь, с проблемой представления исходной формулировки задачи в виде адаптивной системы. Достоинством является повышенная целенаправленность и сходимость алгоритма.

В работе используется представление задачи разбиения в виде адаптивной системы, на основе комбинирования эволюционного, многоуровневого и параллельного подходов к поиску решения

Адаптивная система разбиения верхнего уровня. Задача разбиения гиперграфа формулируется следующим образом.

* Работа выполнена при поддержке РФФИ (гранты № 07-01-00174, № 08-01-00473).

Дан гиперграф $H=(X,E)$, где $X=\{x_i \mid i=1,2,\ldots,n\}$ − множество вершин, а $E=\{e_j \mid e_j \subset X, j=1,2,\ldots,m\}$ – множество ребер (каждое ребро – подмножество связываемых им вершин). Вес ребер задается множеством $\Psi=\{\psi_i \mid i=1,2,\ldots,n\}$. Необходимо сформировать κ-узлов, т.е. множество X разбить на κ непустых и непересекающихся подмножеств X_v: $X=\cup X_v$, $(\forall i,j)\ [X_i \cap X_j = \varnothing]$, $X_v \neq \varnothing$. На формируемые узлы накладываются ограничения. С помощью вектора $N=\{n_v \mid v=1,2,\ldots,k\}$ задается максимально допустимое число вершин назначенных в v-й узел. Ограничения на назначение в узел имеют вид:

$$|X_v| \le n_v, \quad v=1,2,\ldots,k.$$

Основным критерием является F_1 – суммарная стоимость ребер в разрезе:

$$F_1 = \sum \psi_j,$$

где $j \in J = \{j \mid e_j \in C\}$, $C = \{e_j \mid (\forall v)\ [e_j \cap X_v \neq e_j]\}$ – множество ребер в разрезе.

Вторым, часто используемым, критерием является F_2 – суммарное число выводов:

$$F_2 = \sum \gamma_v,\ v=1,2,\ldots,k,$$

где γ_v – число выводов узла X_v.

Возможно использование критерия F, являющегося аддитивной сверткой критериев F_1 и F_2:

$$F=k_1 \cdot F_1 + k_2 \cdot F_2,$$

где k_1 и k_2 – коэффициенты значимости критериев F_1, F_2.

Пусть имеется начальное решение задачи разбиения. Другими словами задано разбиение множество X на κ непустых и непересекающихся подмножеств X_v. На каждом шаге адаптивной поисковой процедуры осуществляются групповые парные перестановки вершин между узлами. Формирование непересекающегося множества пар вершин таких, что вершины каждой пары принадлежат разным узлам, и каждая вершина принадлежит только одной паре, осуществляется на двух уровнях. На верхнем уровне формируется множество непересекающихся пар узлов (МПУ) – $P=\{p_l \mid p_l=<X_i,X_j>;\ (\forall l,r)\ [p_l \cap p_r=\varnothing];\ \cup p_l=X\}$. На нижнем уровне для каждой пары узлов $p_l=<X_i,X_j>$ формируется множество непересекающихся пар вершин (МПВ) – $V_l=\{v_{ln} \mid v_{ln}=<x_s,x_t>;\ x_s \in X_i;\ x_t \in X_j;\ X_i \in p_l;\ X_j \in p_l\}$.

Исходное формирование множества непересекающихся пар узлов выполняется следующим способом. Если число узлов нечетное, то к ним добавляется пустой узел X_0, не содержащий вершин, для того чтобы число узлов было четным. Не теряя общности, будем считать, что имеется чётное число узлов. Множество узлов X случайным образом разбиваем на два равных по мощности подмножества $X1$ и $X2$. Для образования множества пар узлов формируются две опорные линейки позиций – верхняя и нижняя, с числом позиций в каждой – $\kappa/2$ (рис. 1).

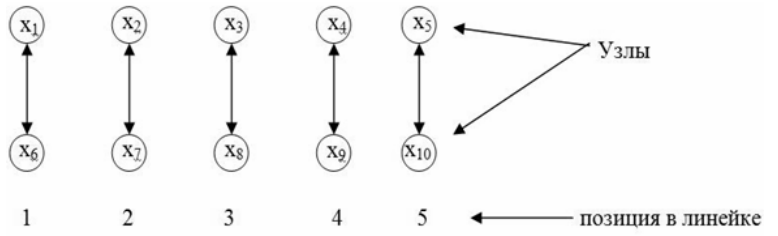


Рис. 1. Две опорные линейки позиций для образования множества пар узлов

Случайным образом подмножество узлов *X1* назначается в позиции верхней опорной линейки, и подмножество *X2* назначается в позиции нижней опорной линейки. Между узлами, занимающими одну и ту же позицию в верхней и нижней опорной линейке, образуются пары (рис. 1).

Множество пар узлов и множество пар вершин эволюционируют в процессе работы адаптивных поисковых процедур. Для модификации множества пар узлов используются два оператора D_1 и D_2. При выполнении оператора D_1 в выбранной паре узлов p_l производится взаимный обмен узлами между одноименными позициями в опорных линейках. Оператор D_2 осуществляет парный обмен узлами между соседними позициями в нижней опорной линейке. Оператор D_2 реализован и двух модификациях $D^1{}_2$ и $D^2{}_2$. При выполнении $D^1{}_2$ рассматриваются такие пары соседних позиций в нижней опорной линейке, у которых первый элемент пары – позиция с нечетным номером, а второй элемент пары – позиция с четным номером. При выполнении $D^2{}_2$ рассматриваются такие пары соседних позиций в нижней опорной линейке, у которых первый элемент пары – позиция с четным номером, а второй элемент пары – позиция с нечетным номером. Операторы $D^1{}_2$ и $D^2{}_2$ выполняются поочередно по одному на каждой итерации. Эволюционные изменения МПУ производятся под управлением автоматов адаптации [10]. Принципы работы адаптивной системы верхнего уровня рассмотрены ниже.

Рассмотрим механизмы управления процессом эволюционного изменения множества непересекающихся пар узлов, выполняющимся на верхнем уровне разбиения. Каждому узлу X_i ставится в соответствие автомат адаптации (АА) $aa3_i \in AA3$, представленный на рис. 2. Если $aa3_i$ находится в одном из состояний группы $C^1{}_i$, то для узла X_i предпочтительным будет расположение в верхней опорной линейке позиций. Если $aa3_i$ находится в одном из состояний группы $C^2{}_i$, то для узла X_i предпочтительным будет расположение в нижней опорной линейке позиций.

Работа группы автоматов $AA3=\{aa3_i \mid i=1,2,\ldots,k\}$ осуществляется под действием управляющих сигналов вырабатываемых после окончания очередной итерации перераспределения вершин между узлами – “поощрения” и “наказания”. На рис. 2 переходы под действием сигнала “поощрение” помечены знаком (+), а переходы под действием сигнала “наказание” помечены знаком (-).

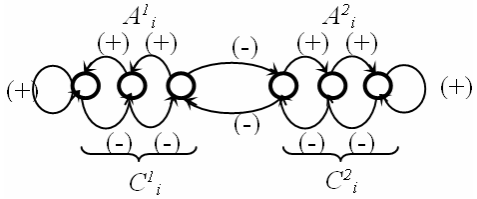


Рис. 2. Граф-схема переходов автомата адаптации

На первом такте работы группы автоматов адаптации *AA3* для каждого узла X_i рассчитывается число t_i его связей с узлами верхней опорной линейки и число b_i связей с узлами нижней опорной линейки.

На втором такте работы группы автоматов адаптации *AA3* вырабатываются управляющие сигналы – “поощрения ” и “наказания”.

Введем **Правило предпочтения.** Будем считать расположение узла X_i в рассматриваемой линейке l_1 предпочтительным, если число связей узла X_i с узлами, расположенными в этой же линейке, меньше числа связей узла X_i с узлами, расположенными в другой линейке l_2.

Если по выше рассмотренному правилу и в соответствии с состоянием автомата адаптации $aaЗ_i$ предпочтительной является одна и та же линейка, то вырабатывается сигнал “поощрение”.

Если по выше рассмотренному правилу предпочтительной является одна линейка, а в соответствии с состоянием автомата адаптации $aaЗ_i$ предпочтительной является другая линейка, то вырабатывается сигнал“ наказание”.

На третьем такте работы группы автоматов адаптации *ААЗ* во всех автоматах адаптации $aaЗ_i$ осуществляются переходы под действием управляющих сигналов.

На четвертом такте работы группы автоматов адаптации *ААЗ* для каждого узла X_i определяется значение индикатора соответствия Q_i.

Если узел X_i расположен в линейке, являющейся предпочтительной в соответствии с состоянием $aaЗ_i$, то Q_i=1. В противном случае Q_i=0.

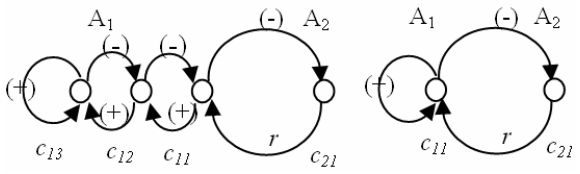
Процедура обмена узлами между линейками осуществляется на основе анализа индикаторов соответствия и будет рассмотрена ниже. Адаптивная система верхнего уровня представлена на рис. 3.

Рис. 3. Адаптивная система верхнего уровня

В соответствии с выше рассмотренной методикой на основании имеющегося начального решения задачи разбиения формируется исходное множество непересекающихся пар узлов, а для каждой пары узлов – исходное множество непересекающихся пар вершин. Все автоматы адаптации устанавливаются в начальное состояние.

Процесс поиска состоит из повторяющихся этапов, каждый из которых представляет собой переход от одного решения к другому, лучшему, что и образует процедуру последовательного улучшения решения.

На каждой итерации эволюционного процесса осуществляется модификация МПУ для которого выполняется процедура “Перераспределение вершин для заданного МПУ”, которая описана ниже. Динамика изменения общей оценки решения отслеживается с помощью автомата адаптации *А4*. Если на некотором интервале итерационного процесса модификации МПУ не приводят к улучшению оценки решения, то процесс поиска прекращается. Структура автомата адаптации *А4* приведена на рис. 4.



а б

Рис. 4. Граф-схема переходов А4

Автомат адаптации *A4* обладает двумя группами состояний:

$$C_1 = \{c_{1i} | \ i=1,2,\ldots,m\} \text{ и } C_2 = \{c_{2i} \mid i=1,2,\ldots,n\},$$

соответствующим двум альтернативам A_1 и A_2.

На рис. 4. а,б приведены два варианта граф-схем переходов АА. Здесь альтернатива A_1 – модификация МПУ; A_2 – прекращение работы.

Число состояний в группе задаётся параметром m – глубина памяти (степень доверия). На каждом шаге работы адаптивной системы процесс коллективной адаптации осуществляется за четыре такта.

На первом такте определяется – есть улучшение оценки решения после смены МПУ или нет.

На втором такте вырабатывается сигнал “поощрения” (+), если есть улучшение оценки решения, или сигнал ”наказание” (-), если улучшения оценки решения нет.

На третьем такте под действием сигналов “поощрения” и ”наказания” *A4* переходит в новое состояние.

На четвёртом такте реализуются альтернативы, в соответствии с состояниями автомата адаптации *A4*. При этом, если автомат адаптации *A4* находится в состоянии, соответствующем альтернативе A_1, то управляющему сигналу *U4* присваивается значение “да”, а если в состоянии, соответствующем альтернативе A_2 , то – значение “нет”. После реализации альтернативы A_2 в АА осуществляется постпереход. Вырабатывается сигнал *r*, под действием которого АА, находящийся в состоянии c_{21}, переходит в c_{11}. Таким образом, в начале каждого шага адаптивной системы АА находится в группе C_1.

Как уже рассматривалось выше, модификация МПУ производится с помощью операторов D_1 , D^1_2 и D^2_2

Взаимный обмен узлами между одноименными позициями в опорных линейках в выбранной паре узлов $p_l=<X_i,X_j>$ производится оператором D_1 на основе анализа значений индикаторов соответствия Q_i и Q_j.

Если (Q_i =0 и Q_j =0), то производится обмен.

Если (Q_i =1 и Q_j =0) или (Q_i =0 и Q_j =1), то обмен с вероятностью P_1.

Если (Q_i =1 и Q_j =1), то обмен производится с вероятностью P_2.

Наилучшие результаты получаются при значениях P_1=0.5, P_2 = 0.1.

Эволюционная процедура перераспределения вершин между парами узлов. Рассмотрим теперь эволюционную процедуру перераспределения вершин между парами узлов выполняемую на втором (нижнем уровне).

Рассмотрим пару узлов $p_l=<X_i,X_j>$, между которыми осуществляется перераспределение вершин. Для образования множества пар вершин используются две опорные линейки позиций (рис. 5).

Мощности множеств X_i и X_j могут быть разными. Пусть $|X_i|=n_i$, а $|X_j|=n_j$, $n_i < n_j$. Число позиций в опорных линейках равно большему значению из n_i и n_j. В позиции верхней линейки, начиная с первой, помещаются вершины меньшего по мощности узла X_i. В позиции нижней линейки, начиная с первой, помещаются

вершины большего по мощности узла X_j. Тогда между вершинами, занимающими одну и ту же позицию в верхней и нижней опорной линейке, образуются n_i пар $V_l=\{v_{ln}|v_{ln}=<x_s,x_t>;\ x_s\in X_i;\ x_t\in X_j;\ X_i\in p_l;\ X_j\in p_l\}$.

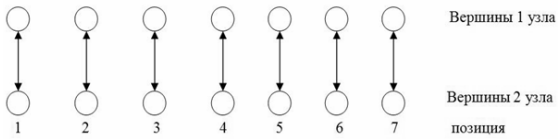


Рис. 5. Две опорные линейки позиций для образования множества пар вершин

Структурная схема процедуры “Перераспределение вершин для заданного МПУ” приведена на рис. 6.

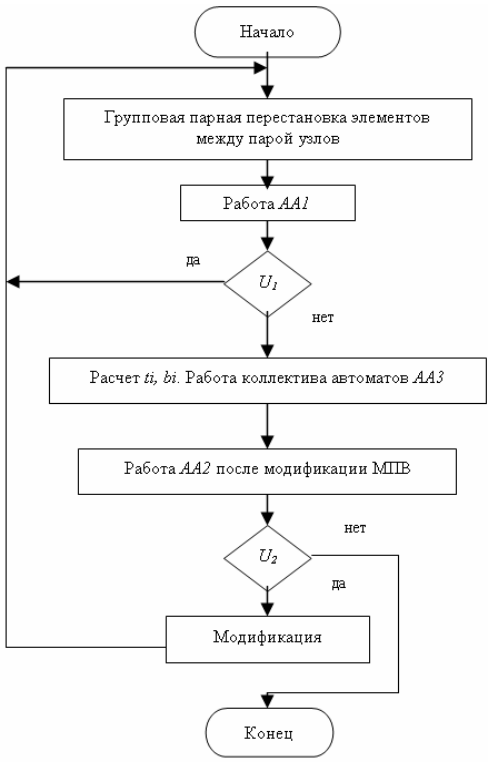


Рис. 6. Структурная схема процедуры “Перераспределение для заданного МПУ”

Процедура включает два цикла внутренний и внешний. На каждой итерации внешнего цикла осуществляется модификация множества пар вершин V_l и перераспределение вершин для заданного множества пар вершин V_l . На каждой итерации внутреннего цикла осуществляется групповая парная перестановка вершин для заданного множества пар вершин V_l. Для каждой пары вершин $v_{ln}\in V_l$, сформированной для пары узлов $p_l=<X_i,X_j>$, определяется приращение δ числа связей, если произвести обмен этими вершинами между узлами X_i и X_j. Затем производится обмен вершинами во всех парах с отрицательным значением приращения δ числа связей.

Итерационные процессы на внутреннем и внешнем циклах управляются с помощью автоматов адаптации *AA1* и *AA2* соответственно. Структура и механизмы поведения автоматов адаптации *AA1* и *AA2* аналогичны автомату адаптации *AA4* (см. рис. 4).

Описанная процедура выполняется параллельно во всех парах узлов.

AA1 отслеживает динамику изменения числа связей между парой узлов $p_l=<X_i,X_j>$ для заданного множества пар вершин V_l. На втором такте работы *AA1* вырабатывается сигнал “поощрения” (+), если есть уменьшение числа связей s_{ij} между парой узлов $p_l=<X_i,X_j>$, или сигнал ”наказание” (-), если уменьшение числа связей нет. Если на некотором интервале итерационного процесса по внутреннему циклу групповые парные перестановки вершин для заданного множества пар вершин V_l не приводят к уменьшение числа связей s_{ij}, то запоминается достигнутое (лучшее) значение числа связей $(s_{ij})_o$, управляющему сигналу U_1 присваивается значение “нет” и осуществляется выход из внутреннего цикла после чего в составе уже внешнего цикла производится модификация множества пар вершин V_l.

AA2 отслеживает динамику изменения числа связей $(s_{ij})_o$ между парой узлов $p_l=<X_i,X_j>$ в процессе эволюционной модификации множества пар вершин V_l. На втором такте работы *AA2* вырабатывается сигнал “поощрения” (+), если есть уменьшение числа связей $(s_{ij})_o$ между парой узлов $p_l=<X_i,X_j>$, или сигнал ”наказание” (-), если уменьшение числа связей нет. Если на некотором интервале итерационного процесса по внешнему циклу модификации множества пар вершин V_l не приводят к уменьшению числа связей $(s_{ij})_o$, то управляющему сигналу U_2 присваивается значение “нет” и осуществляется выход из процедуры “перераспределение вершин для заданного МПУ”.

Экспериментальные исследования. Сравнительный анализ с другими алгоритмами разбиения производился на стандартных тестовых примерах и схемах (бенчмарках). Данные примеры представляют собой стандартные промышленные цепи (блоки, схемы, ИС, БИС, СБИС). Сравнение производилось по параметру **Cut** – число цепей, попавших в разрез. В табл. 1 приведены сравнительные оценки по результатам тестирования разработанного алгоритма ГМА со стандартным многоуровневым алгоритмом hMetis (hM) [7], FM-алгоритмом, и алгоритмом, основанным на эвристике Kernighan-Lin (KL) [5]. В табл. 1 в колонке **cut** представлено число соединений в разрезе полученное алгоритмом ГМА. Результаты двух других алгоритмов описаны в процентах к разбиению, найденному с помощью ГМА. Как видно из табл. 1, ГМА. значительно улучшил результаты FM. Ощутимо улучшение результатов МПЭА по сравнению к hMetis, который является очень сильным алгоритмом. Анализируя данные (см. табл. 1) можно сделать вывод, что разработанный алгоритм находит решения, не уступающие по качеству, а иногда и превосходящие своих аналогов в среднем на 5-10%.

Таблица 1

Test	Cut	МПЭА	FM	hM	KL
ibmO1	180		6.1	0.0	0.0
ibmO2	262		1.5	0.0	0.0
ibmO3	931		23.1	2.6	2.0
ibmO4	516		16.5	4.8	1.1
ibmO5	1640		14.1	4.6	2.1
ibmO6	876		10.9	1.3	1.0
ibmO7	815		27.3	4.9	1.2
ibmO8	1112		14.2	1.7	1.6
ibmO9	620		47.1	0.6	0.0
ibmlO	1249		20.3	1.6	1.1
ibml1	941		53.7	1.6	1.2
ibml2	1835		22.8	4.8	2.3
ibml3	823		43.0	1.9	0.8
ibml4	1758		67.3	4.5	2.1
ibml5	2587		99.8	4.7	3.2
ibml6	1680		41.0	5.4	2.8
ibml7	2126		43.1	5.9	3.5
ibml8	1490		14.2	3.3	2.0

Определены теоретические оценки пространственной и временной сложности разработанного алгоритма. ПСА и ВСА алгоритма – $O(N^2)$.

Заключение. На основе сравнительного анализа существующих подходов и методов решения задачи разбиения, в качестве перспективного выбран гибридный подход нахождения решений, позволяющий
работать с задачами разбиения большой размерности и получать качественные результаты за приемлемое время.

Организация поисковой процедуры произведена на основе интеграции многоуровневого, параллельного и эволюционного подходов.

Для усиления сходимости алгоритма и способности выхода из локальных оптимумов при работе алгоритма использованы механизмы альтернативной поисковой адаптации, разработаны методики выработки откликов среды, организации переходов в автомате адаптации и реализации альтернатив, что позволило разработать представление поиска решения задачи разбиения в виде адаптивного поискового процесса.

Повышение эффективности можно добиться путем интеграции предложенного подхода с механизмами генетического поиска. В этом случае формируется популяция, каждый член которой соответствует исходному состоянию системы (начальные МПВ и МПУ). В процессе эволюции отыскивается такой индивидуум, на базе которого предложенный алгоритм дает наилучшее решение. Рассмотренный алгоритм может быть использован как базовая процедура при многоуровневой парадигме разбиения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Naveed Sherwani.* Algorithms for VLSI Physical Design Automation. Kluwer academic publishers. Boston /Dordrecht/ London. 1995.
2. *M. Sarrafzadeh and C. K. Wong.* An Introduction to VLSI Physical Design. New York: McGraw Hill. 1996.
3. *Деньдобренко Б.П., Малика А.С.* Автоматизация проектирования радиоэлектронной аппаратуры. М., Высш. шк., 2002
4. *J. Cong, C. Wu,* 'Global Clustering-Based Performance-Driven Circuit Partitioning', Proc. ISPD, 2002.
5. *G. Karypis.* Multilevel hypergraph partitioning. In J. Cong and J. Shinnerl, editors, Multilevel Optimization Methods for VLSI, chapter 6. Kluwer Academic Publishers, Boston, MA, 2002.
6. *Yongseok Cheon, Seokjin Lee, Martin D. F. Wong,* "Stable Multiway Circuit Partitioning for ECO", 2003
7. *Navaratnasothie Selvakkumaran and George Karypis,* "Multi-Objective Hypergraph Partitioning Algorithms for Cut and Maximum Subdomain Degree Minimization", ICCAD 2003.
8. *Курейчик В.М., Курейчик В.В.* Генетический алгоритм разбиения графа // Известия Академии наук. Теория и системы управления, №4, 1999.
9. *C. Alpert and A. Kahng.* A hybrid multilevel/genetic approach for circuit partitioning. In Proceedings of the Fifth ACM/SIGDA Physical Design Workshop, pages 100–105, 2002.
10. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Поисковая адаптация: теория и практика. – М.: Физматлит, 2006.
11. *Mazumder P., Rudnick E.* Genetic Algorithm For VLSI Design, Layout & Test Automation. India, Pearson Education, 2003.