Алгоритм генерации начальных разбиений для решения задачи сбалансированного разбиения гиперграфа

М.В. Шеблаев

eASIC Corporation, sheblaev@gmail.com

Аннотация — В данной работе предлагается подход к оценке качества начальных данных и алгоритм нахождения начальных данных для алгоритма Фидуччи-Матейсиса, позволяющий получить качественное приближенное решение исходной задачи сбалансированного разбиения гиперграфа, часто используемой для различных стадий маршрутов проектирования в САПР СБИС.

Ключевые слова — сбалансированное разбиение графа, алгоритм Фидуччи-Матейсиса, FM-алгоритм, mincut, balanced cut.

I. ВВЕДЕНИЕ

Декомпозиция электронной схемы с учетом ограничений - одна из достаточно часто встречающихся задач при разработке САПР СБИС. В частности, декомпозиция логического или физического представления СБИС широко применяется при представлении схемы графом или гиперграфом. Такие стадии традиционных маршрутов проектирования как планирование кристалла (floorplan), размещение (placement), логический синтез могут использовать метод поиска сбалансированного разбиения гиперграфа на подграфы, например [1], [2]. Так для размещения элементов СБИС может быть использован метод последовательных размещений (mincut), в котором по электронной схеме строится ее гиперграф. Этот гиперграф последовательно разбивается на приблизительно равные части так, чтобы оптимизировать число гиперребер, принадлежащих нескольким частям разбиения. Таким образом, можно разместить интегральную схему без нарушения плотности и минимизируя длину проводов

Задача сбалансированного разбиения (гипер) графа лежит в классе NP-полных задач [3], поэтому для ее решения зачастую используются эвристические алгоритмы. На практике алгоритм Фидуччи-Матейсиса [4] вычисляет приближенные решения задачи, близкие к оптимальным, за время, линейное от размера графа (или гиперграфа).

Алгоритм Фидуччи-Матейсиса является алгоритмом пошагового итерационного улучшения. Как и для многих алгоритмов локального поиска существует сильная зависимость результата работы алгоритма от начальных данных. В данной работе мы предлагаем подход к оценке качества начальных данных и алгоритм нахождения начальных данных для

алгоритма Фидуччи-Матейсиса, позволяющий получить качественное приближенное решение исходной задачи сбалансированного разбиения гиперграфа.

II. ПОСТАНОВКА ЗАДАЧИ

A. Алгоритм Фидуччи-Матейсиса

Пусть $H(V, E)$ – гиперграф с множеством вершин V и множеством гиперребер E . Пусть для каждого ребра $e \in E$ определена весовая функция с целыми значениями $w(e) > 0$, а для каждой вершины гиперграфа $v \in V$ определена весовая функция $w(v)$. Задача минимизации разбиения гиперграфа, удовлетворяющего условию баланса, состоит в построении таких непересекающихся подмножеств $A \in V, B \in V, A \cup B = V, A \cap B = \emptyset$, что для ребер e , принадлежащих разрезу $\Psi = ((u, v) \in E: u \in A, v \in B)$, минимизируется функция стоимости.

$$CutCost = \sum_{e \in \Psi} w(e)$$

при условии выполнения балансовых ограничений:

$$1 - \varepsilon \leq \frac{\sum_{u \in A} w(u)}{\sum_{v \in B} w(v)} \leq 1 + \varepsilon$$

В другой постановке, часто используемой, например, при решении практических задач размещения электронных схем, условие баланса может быть записано следующим образом:

$$\sum w(u) \leq S_A$$

$$\sum_B w(u) \leq S_B$$

где S_A, S_B – допустимые суммы весов вершин в подобластях. Также может быть полезным введение терминальных вершин гиперграфа t , фиксированных в подобласти и имеющих нулевой вес. Гиперребра могут содержать не больше одной терминальной вершины.

Такая формулировка позволяет решать широкий круг задач, возникающих при логическом или физическом синтезе схем. Если решается задача размещения, то данные постановки позволяют решать не только задачу оптимизации длины проводов, но и задачи минимизации временных задержек и обеспечения трассируемости схемы на кристалле (см.,

например, [5]). При использовании алгоритма сбалансированного разбиения гиперграфа в методе последовательных разбиений веса вершин гиперграфа моделируют площадь одной ячейки схемы или, в случае иерархических подходов, площадь кластера из нескольких ячеек. Гиперребра моделируют связи между ячейками, а их вес определяется вкладом ребра в длину межсоединений и критичностью путей, проходящих через гиперребро. Существуют разные варианты методов размещения, в которых используется подход последовательного разбиения графа схемы, но решение NP-полной задачи определяет качество полученного размещения по выбранной метрике и быстродействие программной реализации.

Классический алгоритм Фидуччи-Матейсиса на входе имеет первоначальное разбиение и состоит из последовательных итераций. Каждая итерация начинается с присвоения всем вершинам гиперграфа стоимости - значения, на которое уменьшится размер сечения при передвижении этой вершины в другую подобласть. Далее на каждом шаге узел графа с максимальной стоимостью перемещается из одной подобласти в другую, перемещенный узел фиксируется на данной итерации, и стоимости всех соседних узлов пересчитываются. В конце итерации выбирается последовательность шагов, максимизирующая суммарную стоимость или, что то же самое, минимизирующая размер разреза. Итерации повторяются заданное число раз или до тех пор, пока не наступает насыщение алгоритма.

```

function FM PASS (graph  $G = (V, E)$ , max balance  $maxbal$ )
   $P_{saved} \leftarrow A, B$ 
   $CutCost_{saved} \leftarrow CutCost(P_{saved})$ 
  Calculate  $\Delta(v_i)$  for each  $v_i \in V$ 
  repeat
     $v = GETBESTMOVE$ 
    if  $v \in A$  then
       $A \leftarrow A - v; B \leftarrow B + v$ 
    else
       $B \leftarrow B - v; A \leftarrow A + v$ 
    end if
    for  $v'$ , connected by edge with  $v$  do
      if  $v'$  is unfixed then
        Calculate  $\Delta(v')$ 
        UPDATEGAINCELL( $\Delta(v')$ )
      end if
    end for
    Fix  $v$ 
     $P \leftarrow A, B$ 
    if  $CutCost_{saved} > CutCost(A, B)$  then
       $CutCost_{saved} \leftarrow CutCost(A, B)$ 
       $P_{saved} \leftarrow A, B$ 
    end if
  until free base vertices are available
  return  $P_{saved}$ 
end function

```

Рис. 1. Псевдокод алгоритма Фидуччи-Матейсиса

Результат применения эвристики существенно зависит от входного разбиения.

В. Генерация начальных разбиений

В литературе известны различные способы генерации начальных разбиений. Были предложены методы, использующие градиентный спуск [6],

эволюционные алгоритмы [7], построение остовных деревьев [8], жадные алгоритмы [9], модификации алгоритма PageRank [10] и др. Мы предлагаем рассмотреть свойства задачи изнутри и использовать локальные топологические свойства задачи для генерации начальных разбиений.

Пусть разбиение вершин описывается вектором V размерности $|V(G)|$, состоящим из 0 и 1. Здесь i компонента вектора равна 0, если $v_i \in A$ или равна 1, если $v_i \in B$. Обозначим через $FM(V)$ вектор, показывающий финальное распределение вершин - результат применения алгоритма Фидуччи-Матейсиса к начальному разбиению V . Введем функцию D_{FM} на множестве векторов I , представляющих возможные начальные разбиения.

$$D_{FM}(V_0, V_1) = \sum_i |FM(V_0)_i - FM(V_1)_i|.$$

Иными словами, D_{FM} - расстояние Хэмминга для $FM(V_0), FM(V_1)$. Можно показать, что эта функция будет метрикой и задает структуру метрического пространства на множестве I .

Исходя из свойства локальности алгоритма Фидуччи-Матейсиса можно предположить, что близкие в смысле метрики D_{FM} начальные разбиения будут приводить к близким итоговым разбиениям при применении алгоритма Фидуччи-Матейсиса. Тогда для генерации k начальных разбиений мы можем разбить метрическое пространство I на k подмножеств, элементы которых будут иметь попарное расстояние меньше, чем расстояние между другими элементами множества I , и взять представителя каждого подмножества как начальное разбиение для алгоритма Фидуччи-Матейсиса.

III. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ

Для практического применения изложенный выше подход имеет слабые стороны. Во-первых, вычисление метрики D_{FM} имеет сложность $O(|V|)$, а вычисление попарных значений расстояния для всех элементов включает в себя вычисление сбалансированных разбиений для всех элементов пространства начальных состояний - решение интересующей нас задачи. Во-вторых, элементы пространства I есть вектора размерности $|V|$, при этом у каждого вектора $|V|/2$ компонент равны 0, а остальные компоненты равны 1. Количество таких векторов в случае $w(v) = 1$ является биномиальным коэффициентом $\binom{|V|}{|V|/2} \sim \frac{2^{|V|}}{\sqrt{\pi|V|/2}}$. В-третьих, существующие алгоритмы кластеризации не приспособлены для работы с данными большой размерности.

Мы предлагаем использовать вместо D_{FM} вычислительно более простую метрику $D_{FM_n}(V_0, V_1) = \sum_i |FM_n(V_1)_i - FM_n(V_0)_i|$, где $FM_n(V)$ есть результат применения n первых шагов алгоритма Фидуччи-Матейсиса к исходному элементу $V \in I$.

Для практического анализа структуры множества I мы будем использовать выборку из N элементов из I , генерируемых случайно. Наконец, для анализа результатов мы будем снижать размерность получаемых данных, используя метод t-SNE, предложенный в [13] и модифицированный для нашей задачи. Этот метод сохраняет структуру близости между ближайшими объектами. Преимуществом метода t-SNE является склонность к локализации изолированных плотных пространственных структур произвольной геометрии.

Метод t-SNE использует преобразование многомерного расстояния (в нашем случае D_{FM_n}) между точками в условные вероятности, отражающие сходство точек. В работе [13] предлагается физическая аналогия, соединяющая пружинами точки в пространстве отображения, причем жесткость пружин зависит от разности между сходством в исходном пространстве и пространстве отображения. Равновесное состояние такой системы будет искомым распределением точек.

Предлагаемый алгоритм генерации множества начальных значений для алгоритма Фидуччи-Матейсиса выглядит следующим образом:

Шаг 1. Зафиксировать параметры алгоритма, исходя из практической задачи САПР СБИС:

- n - число шагов, используемых для приближенной метрики D_{FM_n} .
- N - число элементов множества I , генерируемых как случайная выборка. Обозначим эту выборку как I_N .

2. Вычислить $FM_n(I_N)$ - образ I_N при применении FM_n к элементам I_N .

3. Применить алгоритм t-SNE к $FM_n(I_N)$, используя вместо евклидовой метрики расстояние Хэмминга. Обозначим результат как $T(I_N)$. Это множество состоит из N векторов из R^2 .

4. Выделить кластеры в $T(I_N)$.

5. Выбрать представителей $\{V_1, \dots, V_k\}$ из каждого кластера, запустить алгоритм Фидуччи-Матейсиса, используя $\{V_1, \dots, V_k\}$ как начальные разбиения.

6. Выбрать лучшее решение.

Шаги 2, 3 и 5 могут выполняться параллельно.

IV. ОЦЕНКА РЕЗУЛЬТАТОВ

Для оценки предложенного алгоритма использовался набор данных ISPD98 [12], широко применяемый для задач САПР СБИС, а также реализация FM алгоритма из [11], ориентированная на задачи размещения ячеек СБИС. Реализация алгоритма t-SNE взята из [14]. Константа баланса $\varepsilon = 0.02$

На рисунках 2-5 представлены примеры распределения вершин гиперграфа после применения алгоритма понижения размерности для тестового примера ibm01 – ibm10 из набора ISPD98.

Экспериментальное сравнение результатов работы алгоритма проводилось для одноуровневого варианта алгоритма Фидуччи-Матейсиса. Мы сравнивали результаты с выборкой из 30 случайно порожденных начальных разбиений и для стратегии, основанной на основном дереве [8], реализованной в [11]. Полученные результаты приведены в таблице 1.

Таблица 1

Результаты сравнения разбиений для различных стратегий начального разбиения

Тест	Размер разбиения для 30 случайно порожденных вариантов	Размер разбиения для стратегии с использованием остовных деревьев	Размер итогового разбиения при использовании нового метода, $n=5, N=100$	Количество вершин гиперграфа
Ibm01	303	393	288	12752
Ibm02	446	373	293	19601
Ibm03	1734	1558	885	23136
Ibm04	1042	1093	720	27507
Ibm05	2010	1878	1971	29347
Ibm06	1076	1173	775	34498
Ibm07	1505	1332	1074	45926
Ibm08	1937	2491	1531	51309
Ibm09	1246	1457	992	53395
Ibm10	2638	2230	1615	69429

Рис. 2. Распределение образов вершин гиперграфа в пространстве $T(I_N)$, тест ibm01

Рис. 3. Распределение образов вершин гиперграфа в пространстве $T(I_N)$, тест ibm02

Рис. 4. Распределение образов вершин гиперграфа в пространстве $T(I_N)$, тест ibm03

Рис. 5. Распределение образов вершин гиперграфа в пространстве $T(I_N)$, тест ibm04

V. ВЫВОДЫ

Предложенный алгоритм генерации начального разбиения для метода Фидуччи-Матейсиса является практически применимым. Качество полученных финальных разбиений может быть существенно улучшено за счет генерации начальных разбиений предложенным методом. Учет топологии пространства решений и ландшафта оптимизируемой функции позволяет открыть перспективное направление для изучения. В дальнейших работах предполагается изучить топологию пространства решений более подробно, а также оценить оптимальный размер

начальной выборки и возможность применения других алгоритмов кластеризации для предложенного подхода.

ЛИТЕРАТУРА

- [1] Kahng A.B., Lienig J., Markov I., Hu J. VLSI Physical Design: From Graph Partitioning to Timing Closure. Dordrecht: Springer Netherlands, 2011.
- [2] Русаков С.Г., Святский А.Б. Метод автоматического разбиения на подсхемы для машинного анализа больших интегральных схем // Управляющие системы и машины. №3 Киев «Наукова думка», 1975. С. 116 – 119.

- [3] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.:Мир, 1982
- [4] Fiduccia C.M., Mattheyses R.M. A linear time heuristic for improving network partitions // Proc. ACM/IEEE Design Automation Conf. 1982. P. 175 – 181.
- [5] Андреев А.Е., Пависич И., Русаков А.С. Алгоритм глобального размещения структурированных заказных схем // V Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем-2012» Сб. трудов / под общ. Ред. Академика РАН А.Л.Стемпковского. М.:ИППИМ РАН. С. 231-236.
- [6] Liu L.-T. A gradient method on the initial partition of Fiduccia-Mattheyses algorithm // Proceedings of the 1995 IEEE/ACM international conference on CAD / IEEE Computer Society, 1995. P. 229–234.
- [7] Кочетов Ю.А., Плясунов А.В. Генетический локальный поиск для задачи о разбиении графа на доли ограниченной мощности // Журнал вычислительной математики и математической физики. 2012. № 1 (52). С. 164–176.
- [8] Курейчик В.В. Алгоритмы разбиения графа на основе генетического поиска // Известия Южного федерального университета. Технические науки. 1999. № 3 (13).
- [9] Caldwell A.E., Kahng A.B., Markov I. L. Improved algorithms for hypergraph bipartitioning // Proc. Asia and South Pacific Design Automation Conference, Yokohama, Japan, Jan. 2000, P. 661–666.
- [10] Andersen R., Chung F., Lang K. Local Graph Partitioning using PageRank Vectors. // 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06) 2006. P.475–486.
- [11] Русаков А.С., Шеблаев М.В. Алгоритм декомпозиции электронных схем с ячейками произвольной площади // Проблемы разработки перспективных микро- и наноэлектронных систем - 2014. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. М.: ИППИМ РАН, 2014. Часть 1. С. 147-152.
- [12] Alpert C.J. The ISPD-98 Circuit Benchmark Suite 1998 // Proc. ACM/IEEE Int'l Symp. Physical Design (ISPD 99), ACM.
- [13] L.J.P. van der Maaten Learning a Parametric Embedding by Preserving Local Structure. // Proceedings of the Twelfth International Conference on Artificial Intelligence & Statistics (AI-STATS), 2009 P. 384-391.
- [14] Pedregosa E. Scikit-learn: Machine Learning in Python. // Journal of Machine Learning Research №12, 2011. P. 2825-2830.

Generation of Initial Partitions for Hypergraph Balanced Cut Problem

M.V. Sheblaev

eASIC Corporation, sheblaev@gmail.com

Abstract - The mincut problem is extensively used in the EDA industry at the different stages of design flow. It is used for synthesis, floorplanning, placement stages as part of netlist manipulation procedures. The problem is NP-complete and can be approximately solved with heuristic algorithms. The most used approach to solve this problem for EDA is Fiduccia-Mattheyses algorithm (FM algorithm). The algorithm is iterative and its quality strongly depends on initial partition.

This research involves study of metric space and local topologies for deeper understanding the optimal way of initial partitions generation to achieve better cut results with smaller runtime cost. We study metric space generated by partitions of hypergraph's vertices with the distance function generated from Fiduccia-Mattheyses algorithm. To reduce complex calculation and to simplify the approach for engineering purposes, we approximate the distance function by few first steps of FM algorithm and use well-known dimension reduction algorithm t-SNE to map the metric space to Euclidian plane. Unsupervised clustering algorithm is used for detection of communities at the image of the initial metric space. These communities are considered as clusters from which initial partitions should be selected. We use items from each cluster to run Fiduccia-Mattheyses algorithm. As result, we significantly increase quality of the flat FM algorithm without paying much cost. Flat FM can be used as part of multilevel hierarchical method. Also

we limit number of initial partitions for FM algorithm and do not perform extra iterations.

We developed an algorithm of initial data generation for Fiduccia-Mattheyses algorithm, which reduces initial dataset to a sparced one. It allows finding balanced graph or hypergraph partitions with high quality. This work contains comparison analysis of the developed algorithm vs. state-of-art solvers.

Keywords — balanced cut, mincut, placement, Fiduccia-Mattheyses.

REFERENCES

- [1] Kahng A.B., Lienig J., Markov I., Hu J. VLSI Physical Design: From Graph Partitioning to Timing Closure. Dordrecht: Springer Netherlands, 2011.
- [2] Rusakov S.G., Svyatskiy A.B. Metod avtomaticheskogo razbieniya na podshemy dly mashinnogo analiza bol'shih integralnyh shem (A Method of Automated Partitioning into Subschemes for the Computerized Analysis of Integrated Circuits). Upravl. Syst. Mashiny №3, P. 116–119 (1975).
- [3] Garey M., Johnson D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco, 1979
- [4] Fiduccia C.M., Mattheyses R.M. A linear time heuristic for improving network partitions // Proc ACM/IEEE Design Automation Conf. 1982. P. 175 – 181

- [5] A. E. Andreev, I. Pavisich, and A. S. Rusakov, Algoritm globalnogo razmescheniya strukturirovanyh zakaznyh shem (Global Placement Algorithm for Structured ASIC,) // Proc. of conference on Problems of Perspective Micro- and Nanoelectronic Systems Development, 2012, pp. 231–236
- [6] Liu L.-T. A gradient method on the initial partition of Fiduccia-Mattheyses algorithm // Proceedings of the 1995 IEEE/ACM international conference on CAD / IEEE Computer Society, 1995. P. 229–234.
- [7] Kochetov Yu.A. and Plyasunov A.V., Geneticheskiy lokalniy poisk dlya zadachi o razbieni grafa na doli ogranichennoi moschnosti (Local genetic search for problem of balanced graph cut with restricted parts size) // Computational Mathematics and Mathematical Physics. 2012. № 1 (52). C. 164–176.
- [8] Kureichik V.V. Algoritmy razbieni grafa na osnove geneticheskogo poiska (Algorithms of graph cut based on genetic search) // Izvestiya of South Federal university. Technical sciences. 1999. № 3 (13).
- [9] Caldwell A.E., Kahng A.B., Markov I. L. Improved algorithms for hypergraph bipartitioning // Proc. Asia and South Pacific Design Automation Conference, Yokohama, Japan, Jan. 2000, P. 661–666.
- [10] Andersen R., Chung F., Lang K. Local Graph Partitioning using PageRank Vectors. // 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06) 2006. P.475–486.
- [11] Rusakov A.S., Sheblaev M.V. Algoritm dekompozitsii elektronnyh shem s yacheikamo proizvolnoi ploschadi (Algorithm for IC decomposition with cells of arbitrary area) // Proc of Conference for Problems of Perspective Micro- and Nanoelectronic Systems Development-2014 Moscow, 2014, pp 147-152
- [12] Alpert C.J. The ISPD-98 Circuit Benchmark Suite 1998 // Proc. ACM/IEEE Int'l Symp. Physical Design (ISPD 99), ACM.
- [13] L.J.P. van der Maaten Learning a Parametric Embedding by Preserving Local Structure. // Proceedings of the Twelfth International Conference on Artificial Intelligence & Statistics (AI-STATS), 2009 P. 384-391.
- [14] Pedregosa E. Scikit-learn: Machine Learning in Python. // Journal of Machine Learning Research №12, 2011. P. 2825-2830.