

基于概率增益的电路划分算法

胡云 王伶俐* 唐璞山 童家榕

(复旦大学专用集成电路与系统国家重点实验室 上海 201203)

摘要: 该文提出了一种新的划分算法, 算法中引入可变线网权重。由于超图(hypergraph)中的线网连接节点数一般多于两个, 为了充分将线网增加的权重作用到与该线网相连的所有节点上去, 线网增益采用了概率增益模型。该算法与原有算法相比较, 可以有效地让电路的划分跳出局部最小, 结果有较大的改进, 特别是当电路规模比较大的时候, 改进更明显。由于采用概率增益模型, 出现浮点数, 节点增益的存储采用了平衡二叉树(balanced binary tree), 因此算法的速度相对于FM算法有所下降, 但是时间复杂度仍然接近为线性复杂度, 时间复杂度为 $O(P \log_2(n))$ (P 为电路所有逻辑单元的引脚数之和, n 为电路的逻辑单元数)。

关键词: 电路划分; 最小割; 概率增益; NP-完全问题

中图分类号: TN47

文献标识码: A

文章编号: 1009-5896(2007)11-2762-05

A Circuit Partition Algorithm Basing on Probability Gain

Hu Yun Wang Ling-li* Tang Pu-shan Tong Jia-rong

(ASIC & System State-Key-Lab, Fudan University, Shanghai 201203, China)

Abstract: This paper proposes a new partition algorithm. The variable weight of the nets is introduced into the algorithm. Generally, the nodes connected to the nets are more than two in the hypergraph, so the probability gain model is used to enforce the effect of the increased weight of the nets. This algorithm can jump out of local minimal effectively when compared with original algorithm. The improvement is especially obvious for the large-scale circuits. For the gain value of the cell is floating-point, balanced binary tree is used to store the gain value of the cells, so the speed of this algorithm is several times slower than FM algorithm. The time complexity of this algorithm is $O(P \log_2(n))$ (where P is the sum of the pins of all logic cells of the circuit, and n is the number of the circuit logic cells).

Key words: Circuit partition; Min cut; Probability gain; NP complete

1 引言

电路划分是VLSI电路设计中比较重要的一个环节, 同时也是改善EDA软件自动化效果的一个重要因素, 特别是当集成电路的规模越来越大的时候, 将规模比较大的电路划分成比较小的电路, 然后分别处理各个小规模电路, 降低设计的复杂性, 同时划分也可以用在多块FPGA来进行原型验证。电路划分处理的元素一般是逻辑门或标准单元。由于电路划分是一个NP完全问题, 所以如何在比较小的时间复杂度的情况下, 找到比较接近最优结果是我们研究划分问题的目标。文献[1]算法是第一个比较有效的二划分算法, 以后的算法都是在该算法基础上进行改进的。关于多路划分有文献[2-4], 其中文献[3]相对于其它多路划分无论从速度还是效果上都比其它的多路划分算法要好。文献[5], 文献[6]是基于

比率划分, 这些算法可以得到优于FM算法^[7]的结果, 但是并不能有效的跳出局部最小。文献[8]通过对节点编号, 将一些群体(cluster)划分到一个集合, 改善初始划分结果。文献[9, 10]在划分的过程中, 通过发现群体, 使群体迅速收敛于一个集合, 较有效地跳出了局部最小。文献[11, 12]中的作者通过实验发现采用LIFO结构(桶排序的实现方式)得出的结果基本上比FIFO好 43%, 比随机的结果好 36%。一种可能的解释就是, 这种选择方式加强了局部选择的程度, 使一些高度结合的节点划分到一个集合, 所以文献[11]通过改进文献[13]的目标函数来加强这种选择方式, 但是效果并不是很明显。

本文的算法通过增加与被锁定节点相连边的权重, 可以显著地改变与锁定节点相连节点的增益, 提高局部选取的优先级, 使那些高度连接的节点移向一个集合, 有效地驱使划分跳出局部最小。同时还采用线网概率增益函数, 避免在超图中一个节点的移动有时并不能影响其相邻节点增益的情况。

2 问题的定义

2006-03-28 收到, 2006-08-25 改回

国家自然科学基金(60676020), 上海应用材料科技合作共同计划项目(AM0406)和复旦大学青年科学基金(JKH1203001)资助课题

*通信作者

电路网表的连接关系一般采用超图来表示^[14]。其中节点对应电路中的逻辑单元，边对应于线网。定义 $H(V, E)$ 为一个超图，其中节点集合为 $V = \{v_1, v_2, v_3, \dots, v_n\}$ ， n 是指节点数。边集合为 $E = \{e_1, e_2, e_3, \dots, e_m\}$ ， m 是线网数，同时每条边由与它相连接的节点所组成，即超图中每条边是 V 的一个子集。由于描述电路的是超图，所以每条边至少连接 2 个节点，即 $|e| \geq 2$ 。本文算法将电路划分为满足面积平衡条件的 V_1 和 V_2 两块， $V_1 \cup V_2 = V$ ， $V_1 \cap V_2 = \phi$ 。式(1)和式(2)是面积平衡约束条件：

$$\frac{|V|}{2}(1-\alpha) \leq |V_1| \leq \frac{|V|}{2}(1+\alpha) \quad (1)$$

$$\frac{|V|}{2}(1-\alpha) \leq |V_2| \leq \frac{|V|}{2}(1+\alpha) \quad (2)$$

其中 α 是控制面积的平衡因子，一般取 0.1。同时划分目标为

$$\min |\{e | e \in E, e \cap V_1 \neq \phi \text{ 和 } e \cap V_2 \neq \phi\}| \quad (3)$$

3 本文提出的划分算法

3.1 基本定义

定义线网 e 的权重为 W_e ，线网 e 和集合 V 的交集的节点个数为

$$C_V(e) = |V \cap e| \quad (4)$$

同时定义 e 中的节点被锁定在集合 V 中的个数为 $L_V(e)$ ，其中 $Already_Lock$ 表示已经被锁定节点的集合

$$L_V(e) = |\{v | v \in e, v \in V, v \in Already_Lock\}| \quad (5)$$

对于超图，在FM算法^[7]中，假定节点 v 在集合 V_1 中，如果将节点 v 从 V_1 移到 V_2 的增益计算目标函数为

$$G(v) = |\{e | v \in e, C_{V_1}(e) = 1\}| - |\{e | v \in e, C_{V_2}(e) = 0\}| \quad (6)$$

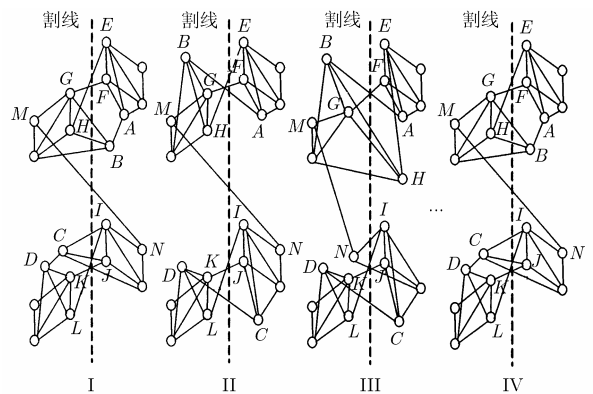
3.2 算法描述

FM算法的主要缺点是，对于有些属于不同群体的节点，它们具有相同的增益，而FM算法在选择这些节点作为下一次移动的目标节点时，随机性很大，所以很可能产生同时在多个群体之间选择节点进行移动的情形，例如将群体中的某个节点从 V_1 移到 V_2 ，另外一个节点从 V_2 移到 V_1 ，那么，这个群体将被跨在两个集合之间，无法移动到一个集合。本文首先增加与锁定节点相连线网的权重，然后将线网增加的权重作用到节点的增益，从而使得与锁定节点相连的节点具有更高的增益，也就具有更高概率被选中，即加强了被选择节点之间的关联度，避免FM算法在多个群体之间同时选择节点进行移动，不致使群体内的某些节点过早地被锁定在不同的集合之间。由于一个节点的增益最多不会超过该节点的度，所以我们在选取线网的权重时，将线网的权重提升(初始权重为 1)，使其达到最大度节点的度的两倍，然后，用该线网的权重乘以它在FM算法中的增益，使被更新节点的增益或在桶(存储增益的数据结构)的最前端，或在最后端，从而有利

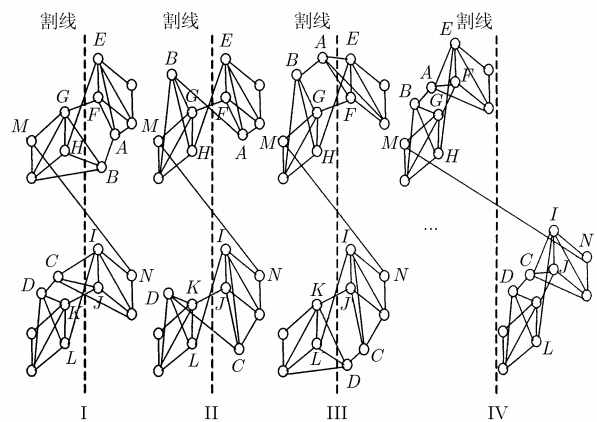
于那些驱使整个群体移动到一个集合的节点首先被选取，此时目标函数变为

$$G(v) = \sum_{v \in e_i, C_{V_1}(e_i)=1} W_{e_i} - \sum_{v \in e_j, C_{V_2}(e_j)=0} W_{e_j} \quad (7)$$

如图 1(a)所示，是我们使用 FM 算法划分的结果，FM 算法很快就收敛到局部最小值 7(如图 1(a)中 II 所示)。如图 1(b)，同样根据 FM 算法，我们得到节点 B, C 的增益最大，首先选择 B 节点移动，然后将节点 B 锁定，更新与节点 B 相连边的权重到节点度最大的度的两倍，此值为 10，再使用目标函数式(7)来更新与 B 节点相连节点的增益，那么节点 A 的增益将由原来 FM 算法中的 -2 变为现在的 7。同时为满足面积平衡，下面选择节点 C 移动，同样更新与节点 C 相连节点的增益，节点 D 的增益由原来 FM 算法中的 -2 变为 7。所以下一次选择移动的节点就会从 A 和 D 之间选择了，而不是像 FM 算法，到此就已经达到局部最小了，接着选择节点 F 和节点 K 来进行交换，再下来选择节点 E 和节点 L 来进行交换，其它节点也跟着交换过来，这样就突破 FM 划分的局部最小，从而达到最优结果。



(a) 使用目标函数式(6)划分的结果



(b) 使用目标函数式(7)划分的结果

图 1 两种不同目标函数划分结果

3.3 超图节点的增益计算模型

由于描述电路采用的是超图，所以实际情况中一条边连接好几个节点，如果采用增益函数式(7)的话，有时在更新节点增益时，节点的增益将保持原有值，因此线网增加的权重不能作用到节点增益。如图 2(a)所示，为了消除该线网处于被分割的状态，首先应该选择 A, B, C 移动到左边，而不是选择 D, H, F, E 来移动到右边，但是如果采用目标函数式(7)计算增益的方法的话，它们之间并没有什么区别，为此我们提出了基于概率增益的模型。如图 2(b)所示，现在假设节点 A 已经被锁定了，如果要消除该线网处于被分割的状态，我们必须将节点 B 和 C 全部移到 A 所在的集合，通过将该线网的权重全部作用到对节点 B 和 C 的引力上，这样就将具有更大的概率来消除该线网的分割状态。

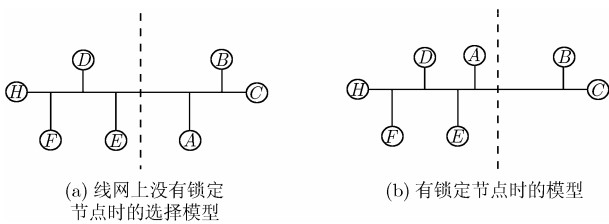


图 2 超图中线网模型

假定线网 e 与节点 v 相连，并且节点 v 在集合 V_1 中，本文将线网 e 对节点 v 增益影响分为两部分。其中线网 e 在 V_1 中的节点阻止节点 v 移动到 V_2 中增益函数为

$$g_{V_1}(e) = \begin{cases} W_e, & L_{V_1}(e) \neq 0 \\ \frac{(C_{V_1}(e)-1)}{|e|} W_e, & L_{V_1}(e) = 0 \end{cases} \quad (8)$$

线网 e 在 V_2 中的节点吸引节点 v 移动到 V_2 中增益函数为

$$g_{V_2}(e) = \begin{cases} W_e, & L_{V_2}(e) \neq 0 \\ \frac{C_{V_2}(e)}{|e|-1} W_e, & L_{V_2}(e) = 0 \end{cases} \quad (9)$$

最后节点 v 的增益函数为

$$G(v) = \sum_{e \cap v \neq \emptyset, e \in E} (g_{V_2}(e) - g_{V_1}(e)) \quad (10)$$

上面目标函数同时还能比较准确反映节点的实际增益。例如当 $C_{V_1}(e)=1$ 时，该线网对节点增益的贡献为 1，由 $g_{V_2}(e) = \frac{C_{V_2}(e)}{|e|-1} W_e = W_e$ ， $g_{V_1}(e) = \frac{(C_{V_1}(e)-1)}{|e|} W_e = 0$ 知该增益结果刚好为 1，但是由于我们的线网存在权重，所以最后得到的增益为 W_e ，与目标函数(7)相吻合。反之，如果我们假设 $C_{V_2}(e)=0$ 的话，那么 $g_{V_2}(e) = \frac{C_{V_2}(e)}{|e|-1} W_e = 0$ ， $g_{V_1}(e) = \frac{(C_{V_1}(e)-1)}{|e|} W_e \approx W_e$ ，最后得到的增益近似为 $-W_e$ ，也和目标函数式(7)比较吻合，所以该目标函数很好地将各种

情况考虑进来，并能比较准确地反映实际情况。

3.4 数据结构

由于我们采用了新的增益函数，所以计算出的增益是浮点数，但 FM 所采用的数据结构要求目标函数计算出来的结果必须是整数，于是 FM 算法的桶数据结构将不再适用于我们的算法。由于保持 FM 算法的基本思想，因此数据结构在进行节点增益的添加和删除需要具有很高效率，为此我们采用了具有添加和删除都很快的数据结构平衡二叉树(BBT)。该形式的数据结构可以适应任何形式的数据类型，并且数据项的添加和删除，都可以在 $O(\log_2(n))$ 的时间复杂度范围内完成(n 表示节点数)。

3.5 算法伪代码描述

Initialize:

Random bipartition G into V_0 and V_1 while satisfying the area balance;

Set the weight of all nets to 1;

Set all cells unlock;

Calculate the gain of the cells and insert them into the BBT(Balanced Binary Tree);

End

Start:

While(existing free cell)

if(area is balance)

then select the biggest gain cell between left_BBT and right BBT;

//left_BBT and right_BBT are used to store the gain of the cells

else

select the biggest gain cell from left_BBT or right_BBT that can keep area balance;

end if

lock this cell and delete it from the BBT;

move this cell to other side;

set the weight of the nets connected to this cell to $2 * \text{Max_Degree}$;

//Max_Degree is the degree of the cells with max degree in all cells.

Update the gain of the cells that connected to this cell;

end while

Find the minimum cut point from the move sequence that satisfies the area balance and then reverse all the cells after this point;

End

3.6 时间复杂度

该算法完成一次划分的条件是所有节点都被锁定，节点在被移动后，我们需要将与该锁定节点相连节点的增益更新一遍，包括把节点原来的增益从平衡二叉树中删除，同时计算一下它的新增益，然后再将该节点增益插入到我们的平衡

二叉树中去。由于节点增益的删除和添加都可以在 $O(\log_2(n))$ 时间复杂度的情况下完成，而这些需要更新的节点数与我们电路的引脚数目有关。实际上，一个节点的增益可能需要更新好几次，但是更新的次数不会超过某个常数(与电路网表信息有关系)，所以该算法的时间复杂度为 $O(P \log_2(n))$ ，其中 P 表示电路所有逻辑单元的引脚数之和， n 表示电路的逻辑单元数。

4 实验结果和分析

我们测试的 16 个电路是来自ISCAS89 的标准测试电路(benchmark)，同时使用C语言分别实现了FM，CLIP^[9]，NWA(Net Weight Algorithm)(本文的算法) 3 个算法，并对它们作了比较。程序是使用VC进行编译的，运行环境是在CPU主频为 2.0G，内存 512MB，操作系统为windows2000的PC机上执行的。每个电路初始划分都是随机的，然后使用各个算法进行优化，每个算法都运行 20 次。实验结果以表格形式给出各种信息，其中表 1 是测试网表信息，表 2 是实验结果，改进比率是NWA相对于FM和NWA相对于CLIP的结果。

表 1 测试电路的信息

电路名	节点数	线网数	引脚数
P1	833	902	2908
S1488	688	667	1977
P2	3104	3029	11219
S15850	10472	10383	24712
S9234	5868	5844	14065
S6669	3459	3402	8658
S4863	2513	2495	6349
Intel	3286	2842	9851
S3271	1730	1714	4366
S3330	2036	1961	4754
S35932	18150	17828	48145
S38417	23951	23843	57613
S38584	20997	20717	55203
S1512	889	866	1966
S1238	556	540	1379
S1196	577	561	1388

表 2 输出结果和数据比较，面积平衡条件(45%—55%)

测试 电路名	20 次运行中最小值					20 次运行的平均值				
	割线数			改进比率(%)		割线数			改进比率(%)	
	FM	CLIP	NWA	$\frac{FM - NWA}{FM}$	$\frac{CLIP - NWA}{CLIP}$	FM	CLIP	NWA	$\frac{FM - NWA}{FM}$	$\frac{CLIP - NWA}{CLIP}$
P1	47	52	47	0.00	9.60	74.9	65.8	56.9	24.0	13.5
S1488	64	73	42	34.3	42.4	73.45	88.05	50.2	31.6	42.9
P2	212	149	138	34.9	7.3	273.9	231.2	167.5	38.8	27.5
S15850	157	67	50	68.1	25.3	241.75	136.2	90.85	62.4	33.2
S9234	57	51	40	29.8	21.5	124.35	94.35	57.85	53.4	38.6
S6669	34	29	27	20.5	6.89	59.4	39.1	32.5	45.2	16.8
S4863	27	24	24	11.1	0.00	47.25	27.6	26	44.9	5.79
Intel	74	39	23	68.9	41.0	157	137	28.15	82.0	79.4
S3271	15	15	15	0.00	0.00	30.9	19.6	16.85	45.4	14.0
S3330	39	34	28	28.2	17.6	56.95	48.05	34.9	38.7	27.3
S35932	39	42	34	12.8	19.0	221.5	51.45	44.15	80.0	14.1
S38417	204	78	54	73.5	30.7	459.55	199.6	104.5	77.2	47.6
S38584	200	47	50	75.0	-6.38	483.5	239.2	105.75	78.1	55.7
S1512	8	9	8	0.00	11.1	16.85	16	10.8	35.9	32.5
S1238	36	36	32	11.1	11.1	48.45	47.05	37.25	23.1	20.8
S1196	37	35	33	10.8	5.71	45.3	46.8	38.25	15.5	18.2

5 结论

本算法在FM算法^[7]和文献^[11,12]结论的基础上,提出了改进的划分算法,本算法通过加大线网的权重和采用改进的增益函数来使高度互连的群体比较有效地收敛于一个集合。从上面的实验数据可以看出,本算法不论是从最优值,还是从平均值的角度来说都比另外两个算法好(见表2)。

参考文献

- [1] Kernighan B W and Lin S. An efficient heuristic procedure for partitioning graphs [J]. *Bell System Tech. Journal*, 1970, 49: 291-307.
- [2] Cong J, Labio W, and Shivakumar N. Multi-way VLSI circuit partitioning based on dual net representation [J]. *IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems*, 1996, 15: 396-409.
- [3] Karypis G and Kumar V. Parallel multilevel k-way partitioning scheme for irregular graphs [A]. Proceedings of the 1996 ACM/IEEE Conference on Supercomputing, Pennsylvania, CA, 1996: 35-42.
- [4] Saab Youssef G. An effective multilevel algorithm for bisecting graphs and hypergraphs [J]. *IEEE Transactions on Computers*, 2004, 53: 641-652.
- [5] Wei Y C and Cheng C K. Towards efficient hierarchical designs by ratio cut partitioning [A]. IEEE/ACM International Conference on Computer Aided Design, Santa Clara, CA, 1989: 298-301.
- [6] Chan P, Schlag M, and Zien J. Spectral k-way ratio-cut partitioning and clustering [J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1994, 13: 1088-1096.
- [7] Fiduccia C M and Mattheyses R M. A linear time heuristic for improving network partitions [A]. Proceedings of the 19th Conference on Design Automation, Piscataway, CA, 1982: 175-181.
- [8] Alpert C J and Kahng A B. A general framework for vertex ordering with application to netlist clustering [J]. *IEEE Transactions on Very Large Scale Integration Systems*, 1996, 4: 240-246.
- [9] Dutt D and Deng W. VLSI circuit partitioning by cluster-removal using iterative improvement techniques [A]. IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, 1996: 194-200.
- [10] Eem C K and Chong J W. An efficient iterative improvement technique for VLSI circuit partitioning using hybrid bucket structures [A]. Proceedings of the ASP-DAC '99 on Design Automation Conference, Wanchai, HK, 1999, 1: 73-76.
- [11] Hagen L W, Huang D J H, and Kahng A B. On implementation choices for iterative improvement partitioning algorithms[A]. Proceedings of the Conference on European Design Automation, Brighton, UK, 1995: 144-149.
- [12] Alpert Arles J and Kahng Andrew B. Recent directions in netlist partitioning [J]. *Integration the VLSI Journal*, 1995, 19:1-81.
- [13] Krishnamurthy B. An improved min-cut algorithm for partitioning VLSI networks [J]. *IEEE Trans. on Computer*, 1984, C-33: 438-446.
- [14] Schweikert D G and Kernighan B W. A proper model for the partitioning of electrical circuits[A]. Proceedings of the 9th Workshop on Design Automation, New York, CA, 1972: 57-62.

胡云: 男, 1982年生, 硕士, 从事多块FPGA的原型验证和CAD软件开发。

王伶俐: 男, 1971年生, 副教授, 博士, 研究方向为IC设计和EDA软件的开发和研究。

唐璞山: 男, 教授, 博士生导师, 长期在VLSI电路设计和EDA软件算法等领域从事科研和教学工作。目前的研究方向是EDA和VLSI的形式验证。

童家榕: 男, 1942年生, 教授, 博士生导师, 主要研究方向为电子线路与系统、集成电路设计及集成电路计算机辅助设计技术。