

基于快速模拟退火算法的可切割布图规划方法

罗 佐, 杜世民, 戚利侠, 夏银水

(宁波大学 电路与系统研究所, 浙江 宁波 315211)

摘 要: 为了提高基于经典模拟退火算法的可切割布图规划方法的效率和效果, 提出了一种动态改变温度下降比例来实现快速模拟退火算法的可切割布图规划方法, 并将提出的方法应用于 MCNC 基准电路进行测试. 实验结果表明该算法可以提高搜索优化解的效率和效果.

关键词: VLSI 设计; 可切割布图规划; 快速模拟退火算法; 波兰正则表达式

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2012)09-0109-04

Slicing Floorplanning Based on Fast Simulated Annealing

LUO Zuo, DU Shi-min, QI Li-xia, XIA Yin-shui

(Institute of Circuits and Systems, Ningbo University, Ningbo 315211, China)

Abstract: In order to improve the efficiency and effectiveness of slicing floorplanning based on classical simulated annealing. A new slicing floorplanning based on fast simulated annealing which can change the ratio of temperature descending dynamically are proposed. The proposed method is applied to MCNC benchmark circuits, and the experimental results show that this algorithm can improve the efficiency and effectiveness of searching optimization solution.

Key words: VLSI design; slicing floorplanning; fast simulated annealing; normalized polish expression

1 引言

随着芯片规模的增大、频率的增高、尺寸的减小, 获得快速布图规划已成为 VLSI 设计中最大的挑战之一. 布图规划表示方法可分为两类: 可切割布图规划和不可切割布图规划. 可切割布图规划一般用 slicing trees 和 NPE (normalized polish expression) 表示^[1]; 不可切割布图规划表示方法有 sequence-pair, bounded slicing grid, O-tree, B*-tree, TCG-S^[2-6] 等. 因为可切割布图规划速度快, 可以实现分层设计, 而且能够充分利用模块的形状和转向的灵活性, 所以这种布图规划方法仍然得到广泛的研究. 在布图规划中硬模块相对软模块有易操作, 易收敛的特点, 所以本文主要研究基于硬模块的布图规划.

布图规划实际上是 NP-hard 组合优化问题^[7], 经典模拟退火算法等启发式算法被广泛地应用在这一领域. 模拟退火算法因为在初始阶段接受了很多不必要的不良解, 所以它在速度和效果方面尚需得到进一步的提高. 文献[7]采用了先使温度变化因子升高, 然后减小的退火策略来提高退火效率的方法, 文献[8]提出了动态改变权重的自适应模拟退火算法来优化 B*-tree 布图规划的方法.

本文通过改进模拟退火策略来加快退火速度和效果, 并应用于可切割布图规划, 实验中使用 MCNC 基准电路进行测试. 实验结果表明相对于经典模拟退火算法, 快速模拟退火算法可以提高搜索优化解的效率和效果, 而相对于基于模拟退火算法的 B*-tree, 在模块数少时可以提高搜索优化解的效率和效果.

收稿日期: 2012-02-20; **修回日期:** 2012-03-18

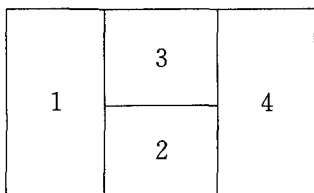
基金项目: 国家自然科学基金重点项目(61131001); 浙江省自然科学基金重点项目(Z1090622); 教育部博士点基金(20113305110001)

2 可切割布图规划的表示方法和快速模拟退火算法

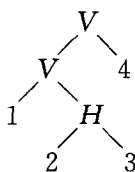
2.1 可切割布图规划表示

可切割布图规划通常用二叉树来表示. 波兰表达式可以用来标记这个二叉树, 波兰表达式 $E=e_1e_2\cdots e_{2n-1}$, 其中 $e_i \in \{1, 2, \cdots, H, V\}$, 每一个数字表示一个模块, $H(V)$ 表示水平切割(垂直切割). 当模块的数目为 n 时, E 的长度为 $2n-1$. 因为波兰表达式是二叉树的后序遍历, 所以它有 balloting 性质: 即对于每一个子式 $E_i=e_1e_2\cdots e_i, 1 \leq i \leq 2n-1$, 操作数的数目要比操作数的数目多.

同一个可切割布图规划可能产生多个可切割二叉树, 这种冗余会增加解空间, 从而使得优化解的搜索难度增大. 为了除去这种冗余, 定义了一种倾斜可切割二叉树(skewed slicing tree), 这种可切割二叉树可以用波兰正则表达式 NPE 表示, 如图 1 所示. 波兰正则表达式是指不包含连续相同的操作如“VV”或“HH”的波兰表达式, 每一种可切割二叉树对应着一种波兰正则表达式, 即可切割布图规划和波兰正则表达式是一一对应的关系.



(a) 可切割布图规划



(b) 倾斜树(skewed)

图 1 可切割布图规划及其倾斜二叉树

在处理组合优化问题的过程中, 模拟退火算法不能保证能搜索到最优解, 但是可以在一个合理的时间内得到优化解, 由于这种特性它被广泛的应用在模块的布图规划中.

2.2 快速模拟退火算法

经典模拟退火算法的策略是给予一个较高的初始温度, 随着模拟退火过程的进行, 温度以一个固定的比例下降. 在每一个温度, 扰动常数当前解来找到一系列相邻解, 这个过程进行到“冷却”(frozen)

状态或是预先设定的终止状态为止, 此时得到的解就是搜索空间里最好的解. 在模拟退火的过程中, 当成本函数的增量比较小或是当前温度比较高时接受不良解的概率较大.

快速模拟退火算法通过改变经典模拟退火算法的退火策略来提高搜索解空间的速度和效果. 快速模拟退火算法的退火过程包括三个部分: (1) 高温随机搜索阶段; (2) 伪贪心局部搜索阶段; (3) 爬坡(hill-climbing)搜索阶段^[8]. 在初始阶段, 温度 $T \rightarrow \infty$, 接受差解的概率接近于 1, 使得初始解被接受; 在第二个阶段, 温度 $T \rightarrow 0$, 由于温度很低, 接受差解的概率很低, 相当于局部贪心搜索; 第三个阶段, 温度上升充分利用爬坡代技术.

为了实现上面的三个过程, 文献[8]定义了快速模拟退火算法中温度的退火策略如下:

$$T_n = \begin{cases} \Delta_{\text{avg}} / \ln P & l=1 \\ T_1 < \Delta_{\text{cost}} > / (l * c) & 2 \leq l \leq j \\ T_1 < \Delta_{\text{cost}} > / l & l \geq j \end{cases} \quad (1)$$

其中, l 代表迭代的次数, Δ_{avg} 代表成本函数增加的平均值, P 是指接受的成本函数增加时解的原始概率, P 要尽可能接近 1, 但不能等于 1, cost 是指成本函数的值, $<\Delta_{\text{cost}}>$ 是指在当前温度下平均成本函数改变的值(新的平均成本函数-原来的平均成本函数), c, j 是常数. 在第一次迭代中, 根据给定的 P 值来设定初始温度, 因为 P 通常设置为非常接近 1 的常数, 所以温度 $T \rightarrow \infty$, 这个阶段解的接受概率 $\exp(-\Delta C/T)$ 接近于 1, 从而实现了了解的随机搜索. 从第二次迭代到第 j 次迭代, 常数 c 控制温度下降的比例, 通常设置较大的 c 值使温度 $T \rightarrow 0$, 接受概率 $\exp(-\Delta C/T)$ 很小, 所以接受不良解的概率很小, 从而实现了伪贪心算法, 而且这个阶段温度下降比例比经典模拟退火的快, 总的迭代次数减小, 从而时间有所减小, 在 j 次迭代之后, 温度上升从而提高解的质量, $<\Delta_{\text{cost}}>$ 的值影响温度下降的比例, 当相邻解的成本函数变化较大时, $<\Delta_{\text{cost}}>$ 较大, 温度值下降的较慢, 迭代次数较多; 当 $<\Delta_{\text{cost}}>$ 较小时, 温度下降较快, 迭代次数较少, 但是总的下降比例比经典模拟退火的快, 总的迭代次数减小, 从而时间有所减小.

3 基于快速模拟退火算法的可切割树的实现

3.1 相邻结构的定义

在一个波兰正则表达式中, 如果两个操作数(代表模块的数字)之间没有其它操作数则称这两个操

作数相邻,如果两个操作(V或H)之间没有其它操作数或操作则称这两个操作相邻,如果一个操作和一个操作数紧挨着对方则称这个操作和这个操作数相邻.操作链是指一系列相邻的操作,因为波兰正则表达式中没有连续相同类型的操作,所以在波兰正则表达式只可能有两种操作链,即VV...或HH...

一个波兰正则表达式通过三种相邻扰动来产生另一个波兰正则表达式则称这两个波兰正则表达式相邻.

三种相邻扰动定义如下:

(1) 交换两个相邻的操作数;

(2) 反转波兰正则表达式中的操作链,将V反转H,将H反转成V;

(3) 交换两个相邻的操作数和操作.

3.2 成本函数的定义

在布图规划设计中,面积和线长是构建成本函数中最常考虑的两个因素.本文采用如下的成本函数进行模拟退火:

$$\text{cost} = \alpha \frac{A}{A_{\text{norm}}} + (1 - \alpha) \frac{W}{W_{\text{norm}}} \quad (2)$$

其中, cost 代表成本函数, A 是布图规划的面积, A_{norm} 是布图规划的平均面积, W 是总线长, W_{norm} 是平均线长, α 是权重因子.为了计算 A_{norm} 和 W_{norm} , 模拟退火过程开始之前, 通过以上三种操作扰动原始波兰正则表达式 m 次, 然后求面积和线长的平均值 A_{norm} 和 W_{norm} , m 值与模块的数目成正比.

基于快速模拟退火算法的可切割树的伪代码如算法1所示.

算法1 The process of Slicing Floorplanning

```

1.  $E = 12V3V4V \dots nV$ ; //initial solution
2.  $E_{\text{best}} = E$ ;  $T = -\Delta \text{avg} / \ln P$ 
3. do
4.   reject = 0;
5.   process Fast SA Temperatrue Scheme
6.   for  $ite = 0$  to  $k$  do
7.     SelectOpertion(Op)
8.     Case Op
9.     Op1: Select two adjacent operands  $e_i$  and  $e_j$ ;  $E' = \text{Swap}(E, e_i, e_j)$ 
10.    Op2: Slect a nonzero length chain C;  $E' = \text{Complement}(E, C)$ 
11.    Op3: done = FALSE
12.    while not(done) do
13.      Conice 1: Select two adjacent oper and  $e_i$  and operator  $e_{i+1}$ 
14.      if  $(e_{i-1} \neq e_{i+1})$  and  $(2N_{i+1} < i)$  then done

```

= TRUE

```

15.   Choice 2: Select two adjacent operator  $e_i$  and operand  $e_{i+1}$ 
16.     if  $(e_i \neq e_{i+2})$  then done = TRUE
17.   end while
18.    $E' = \text{Swap}(E, e_i, e_{i+1})$ 
19. end case
20.  $\Delta \text{cost} = \text{cost}(E') - \text{cost}(E)$ 
21. if  $(\Delta \text{cost} \leq 0)$  or  $(\text{Random}) < e^{-\Delta C/T}$  then
22.    $E = E'$ 
23.   if  $\text{cost}(E) < \text{cost}(E_{\text{best}})$  then  $E_{\text{best}} = E$ 
24. else
25.   reject = reject + 1
26. end if
27. end for
28. until  $(\text{reject}/k > 0.95)$  or  $(T < )$  or (Out of Time)

```

4 实验结果

文中提出的基于快速模拟退火算法的可切割布图规划方法, 采用C语言进行了实现. 算法中相关参数设置如下: 面积和线长权重分别为1和0, 即 $\alpha = 1$, 只考虑面积, 迭代次数 $k = 20 * n$ (n 为模块数), $c = 150$, $j = 10$, $P = 0.98$, 运行在 2.2GHz Intel Pentium M 处理器和 1GB 内存的 PC 机上, 并对 MCNC 基准电路进行了测试, 采用硬模块, 实验结果如表1所示.

表1中 Classical SA Slicing 代表的是文献[1]中采用经典模拟退火算法实现的可切割布图规划, B*-tree 代表的是文献[5]中采用基于模拟退火算法的 B*-tree 实现不可切割布图规划的结果, Fast-SA Slicing 代表的是采用快速模拟退火算法实现可切割布图规划的结果. 从表1的 Classical SA Slicing 和 Fast-SA Slicing 的实验结果中可以看出 apte 电路的面积没有改进, 但是速度提高了 1.42 倍, 这是由于 apte 电路特性决定的, 因为在 apte 电路的 9 个模块中 8 个模块比较大, 1 个模块比较小, 最后布图规划的结果始终是一样的, 所以面积没有优化, 而其他 4 种电路在面积和时间方面都有所改进, 面积平均减小了 1.77%, Deadspace 平均小了 29.60%, 时间平均缩短了 26.36%; 从表2的 B*-tree 和 Fast-SA Slicing 的结果中可以看出, 当模块数较少时, 如 xerox, hp 电路, 本文提出的方法在面积和时间上都有所改进, 面积平均减小了 1.14%. Deadspace 平均小了 22.49%, 速度平均提高了 1.43 倍; 在模块数较多时, 如 ami33, ami49 电路, 速

表 1 Classical SA Slicing, B*-tree 和 Fast-SA Slicing 的对比

Circuits	Classical SA Slicing ^[1]			B*-tree ^[5]			Fast-SA Slicing		
	Area (mm ²)	Dead space (%)	Time (sec)	Area (mm ²)	Dead space (%)	Time (sec)	Area (mm ²)	Dead space (%)	Time (sec)
apte	46.92	0.78	1.43	46.92	0.78	2.65	46.92	0.78	0.59
xerox	20.49	5.89	3.13	20.45	5.68	5.39	20.23	4.54	2.13
hp	9.33	5.66	3.67	9.27	4.98	5.74	9.16	3.74	2.46
ami33	1.23	6.03	15.37	1.20	3.45	29.62	1.21	4.31	11.21
ami49	38.20	7.75	39.53	36.97	4.29	46.47	37.29	5.19	34.22

度平均提高了 1 倍。

5 结束语

本文提出了基于快速模拟退火算法的可切割布图规划方法。实验结果表明相对于经典模拟退火算法,快速模拟退火算法可以提高搜索优化的效率和效果,相对于基于模拟退火算法的 B*-tree,在模块少时可以提高搜索优化的效率和效果,在优化模块较多的大电路时可提高优化速度。

参考文献:

- [1] Wong D F, Liu C L. A new algorithm for floorplan design[C]//Proceedings of the 23rd ACM/IEEE Design Automation Conference. New York: IEEE Press Piscataway, 1986: 101-107.
- [2] Murata H, Fujiyoshi K, Nakatake S, Kajitani Y. VLSI module placement based on rectangle-packing by the sequence-pair[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1996, 15(12): 1518-1524.
- [3] Nakatake S, Fujiyoshi K, Murata H, et al. Module placement on BSG-structure and IC layout applications [C]//Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design. Washington D C: IEEE Computer Society Press, 1996: 484-491.
- [4] Guo P N, Cheng C K, Yoshimura T. An o-tree representation of non-slicing floorplan and its applications [C]//Proceedings of the 36th annual ACM/IEEE Design Automation Conference. New York: IEEE Computer Society Press, 1999: 268-273.
- [5] Chang Y C, Chang Y W, Wu G M, et al. B*-trees: a new representation for non-slicing floorplans [C]//Proceedings of the 37th Annual Design Automation Conference. New York: IEEE Computer Society Press, 2000: 458-463.
- [6] Lin J M, Chang Y W. TCG-S: orthogonal coupling of P*-admissible representations for general floorplans [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002, 23(6): 842-847.
- [7] Sechen C, Sangiovanni-Vincentelli A. The Timber-Wolf placement and routing package[J]. IEEE Journal of Solid-State Circuits, 1985, 20(2): 510-522.
- [8] Chen T C, Chang Y W. Modern floorplanning based on B*-tree and fast simulated annealing[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005, 25(4): 637 - 650.

作者简介:



罗佐男, (1988—), 硕士研究生。研究方向为低功耗与高信息密度集成电路设计。

杜世民男, (1976—), 博士研究生, 讲师。研究方向为低功耗与高信息密度集成电路设计。

戚利侠男, (1986—), 硕士研究生。研究方向为低功耗与高信息密度集成电路设计。

夏银水男, (1963—), 博士, 研究员。研究方向为集成电路综合和优化、高信息密度集成电路设计。