VLSI电路划分问题的分散搜索算法

朱文兴,程 泓
(福州大学数学与计算机科学学院,福建福州 350108)

摘 要: 电路划分是超大规模集成电路(VLSI)设计自动化中的一个关键阶段,是NP困难的组合优化问题.本文把基于顶点移动的Fiduccia-Mattheyses(FM)算法结合到分散搜索算法框架中,提出了电路划分的分散搜索算法.算法利用FM算法进行局部搜索,利用分散搜索的策略进行全局搜索.为满足该方法对初始解的质量和多样性的要求,采用贪心随机自适应搜索过程(GRASP)和聚类相结合的方法产生初始解.实验结果表明,算法可以求解较大规模的电路划分实例,且与基于多级框架的划分算法hMetis相比,划分的质量有明显的提高.

关键词: 分散搜索; GRASP; FM算法; 电路划分
中图分类号: TP391.72 **文献标识码:** A **文章编号:** 0372-2112 (2012) 06-1207-06
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.06.023

Scatter Search Algorithm for VLSI Circuit Partitioning

ZHU Wen-xing, CHENG Hong
(*College of Mathematics and Computer Science, Fuzhou University, Fuzhou, Fujian* 350108, *China*)

Abstract: Circuit partitioning is an important stage in the very large scale integration (VLSI) physical design automation, which influences further circuit design. The VLSI circuit partitioning problem is an NP-hard combinatorial optimization problem. In this paper, we propose a scatter search method for the problem, which incorporates the single-vertex-move based Fiduccia-Mattheyses algorithm (FM) within the scatter search framework. The FM algorithm is used for local exploitation, while the scatter search strategy is used for global exploration. To meet the quality and diversity of initial solutions required by the scatter search method, we incorporate the greedy randomized adaptive search procedure (GRASP) with the clustering method to generate initial solutions. Experimental results show that the proposed scatter search algorithm is capable of partitioning large benchmark circuits, and yields results better than those of the well-known multilevel partitioning package hMetis.

Key words: scatter search; GRASP; FM algorithm; circuit partitioning

1 引言

VLSI电路划分将单元器件分成两个或多个子集,降低超大规模集成电路设计的复杂性,增强划分电路的可读性.电路划分在VLSI设计中占有重要的地位,作为超大规模集成电路物理设计中的一个关键阶段,电路划分的结果影响后续的布图、布局和布线等过程,而且电路划分在超大规模集成电路可靠度计算[1]等领域中也有应用.以最小割为目标,以子集大小平衡为约束的划分问题是NP完全问题[2],因此要提高电路划分的质量是一项困难的工作,特别是当电路规模增大时,如何保证划分的质量是有意义的研究课题.

解决VLSI电路划分问题的主要方法可以分为三类:其一是基于移动迭代的启发式算法,如文献[3]提出的KL算法,文献[4]提出的FM算法等.随着问题规模的增大,单一的基于顶点移动的迭代方法容易陷入"平均"质量的局部最优解,因此比较适合规模不大的划分问题.第二种是智能计算方法,如模拟退火算法[5]和遗传算法[6,7]等,该类算法很难求解较大规模的VLSI电路划分问题.第三种是基于聚类的多级划分方法.多级划分的算法框架首先在解决图划分问题时被提出,Karypis等人将这个算法框架推广到电路划分问题[8],当前主要的多级划分工具有hMetis[9]和MLPart[10]等.

分散搜索(Scatter Search,简称SS)是Glover在20世纪70年代提出的一种启发式算法[11].从提出至今,SS算法已被用于解决各个领域的复杂问题,如车辆路径规

收稿日期:2011-07-18;修回日期:2012-04-06
基金项目:国家重点基础研究发展规划(973计划)项目(No.2011CB808000);国家自然科学基金(No.61170308,No.61070020,No.10931003);教育部高校博士点专项科研基金(No.20093514110004)

划、弧路径规划、神经网络、图形绘制、二次规划、混合整数规划和多目标优化问题等.大量实验结果表明 SS 算法是一种行之有效的优化方法[12].本文试图改进分散搜索算法使适合求解超大规模集成电路划分问题,主要思想是把基于顶点移动的迭代 FM 算法结合到分散搜索算法框架中,利用 FM 算法搜索局部最优解,而利用分散搜索算法搜索全局最优解.

分散搜索是基于种群进化的高级启发式算法,它利用解的组合来跳出局部最优,并利用参考集(Reference Set)这一种群的概念,通过不断更新参考集,保证种群不断向好的方向进化.作为演化类算法,它与遗传算法的不同之处在于:遗传算法的种群规模一般比较大,往往要大于 100,且随机地从种群中选择两个个体杂交产生一个或多个个体;而分散搜索的种群规模则小得多,参考集中解的个数一般不超过 20,且用确定性的规则从参考集中选择两个或者多个解,通过组合规则产生一个或者多个新解.SS 算法的框架主要包含五个步骤[13],分别是:多样性产生方法、解的改进方法、子集产生方法、解的组合方法、参考集更新方法.

2 基于分散搜索的电路划分算法

超大规模集成电路通常用超图来描述[14].超图中的顶点表示电路中的单元模块,超边表示电路中的线网.给定一个超图 $H(V,E)$,顶点集 $V=\{v_1,v_2,v_3,\cdots,v_n\}$,$n$ 表示电路中单元模块的数目,$|v_i|$ 表示模块 i 的面积.超边集合 $E=\{e_1,e_2,e_3,\cdots,e_m\}$,$m$ 表示电路中的线网数.本文研究集成电路的二划分问题,即将超图 H 的顶点集 V 划分成两个非空子集 V_1 和 V_2,使得 $V_1\cup V_2=V$,$V_1\cap V_2=\varnothing$,且满足面积约束条件:

$$\frac{\sum|v_i|}{2}(1-\beta)\leqslant|V_1|\leqslant\frac{\sum|v_i|}{2}(1+\beta) \quad (1)$$

$$\frac{\sum|v_i|}{2}(1-\beta)\leqslant|V_2|\leqslant\frac{\sum|v_i|}{2}(1+\beta) \quad (2)$$

其中 β 是控制子集面积的平衡因子,也叫做子集容量的偏差比,β 一般取 10% 或 2%.同时,划分的目标为:

$$\min|E(V_1,V_2)| \quad (3)$$

其中,$E(V_1,V_2)$表示被子集 V_1 和 V_2 切割的线网的集合.

2.1 算法总体设计

本文的基于分散搜索的电路划分算法框架如算法 1 所描述,接下来的 2.2 节到 2.5 节,分别对算法 1 中涉及到的几个算法做详细的阐述.

2.2 初始解生成方法

从算法的角度看,分散搜索是对一个由优良的解组成的参考集(Reference Set)迭代操作的过程,包括解的组合、改进,以及更新.需要注意的是,这里的“优良”不仅指参考集中的解对应的目标函数值要好(在本文中指割集大小),也要求能将多样性引入参考集.因此在构造初始解的时候,要求设计的算法同时具备随机的性质和贪心的性质.本文利用贪心随机自适应过程[15](GRASP)的初始解构造方法和聚类方法来产生初始解.算法 2 描述了具有多样性的初始解的产生过程.

算法 1 基于分散搜索的电路划分算法

输入:超图 $H(V,E)$

输出:超图 $H(V,E)$的划分

1. 读取超图 H 的信息;
2. 构造初始解集 P(见 2.2 节),其中$|P|\geqslant 100$;
3. 利用局部搜索算法改进 P 中的初始解(见 2.3 节);
4. 从 P 中选取 b 个解构造初始参考解集(见 2.5.1 节);
5. Repeat
6. 在参考解集中,通过对解的组合产生新的解(见 2.4 节);
7. 利用局部搜索算法改进新解(见 2.3 节);
8. 更新参考解集(见 2.5.2 节);
9. Until 终止条件满足;
10. Return 参考解集中最好的解.

算法 2 基于 GRASP 的单点聚类算法

输入:超图 $H(V,E)$

输出:超图 $H(V,E)$的一个初始二划分 $P(V_1,V_2)$

1. 读取超图 H 的信息;
2. 将所有顶点标记为未分配;
3. Repeat
4. 从标记为未分配的顶点中找到最大和最小顶点度数 $\mathrm{Deg}_{\max}$ 和 $\mathrm{Deg}_{\min}$;
5. 构造 RCL = $\{v\mid v$ 未分配 & $\mathrm{Deg}(v)\geqslant \mathrm{Deg}_{\min}+\alpha*(\mathrm{Deg}_{\max}-\mathrm{Deg}_{\min})\}$;
6. 从 RCL 中随机选择一个顶点 v;
7. 将 v 及与 v 相邻的未分配顶点随机分配到同一个子集中,如果达到该子集的容量上限,则分配到另一个子集中,并将它们标记为已分配;
8. Until 所有的顶点都已分配.

文献[16]指出超大规模集成电路对应的超图一般具有高度聚类的特性.如果随机构造初始划分的话,强关联的顶点经常被分在不同的集合里,而 FM 算法经常不能把这些顶点移到相同的子集,导致最后的划分结果往往不理想.因此为了提高初始解的质量,本文采用了单点聚类的策略,每次挑选一个度数比较大的顶点,将与它相邻的顶点聚到同一个划分集合中(步骤 7).

分散搜索不仅对初始解有质量的要求,还有多样性的要求.本文在算法 2 中用 GRASP 构造阶段的方法对初始解的质量和多样性可以进行控制,其核心在于构造限制候选列表 RCL.首先从还未分配的顶点中找到最大和最小顶点度数 $\mathrm{Deg}_{\max}$和 $\mathrm{Deg}_{\min}$(步骤 4);然后将度数大于等于 $\mathrm{Deg}_{\min}+\alpha*(\mathrm{Deg}_{\max}-\mathrm{Deg}_{\min})$的未分配顶点加

入 RCL(步骤 5);其次,每次从 RCL 中随机选取一个顶点 v 进行聚类,其中,参数 $\alpha\in[0,1]$是算法 2 随机性质和贪心性质的控制因子.若 $\alpha=0$,那么该算法就是纯随机的,即每次随机地选取一个未分配的顶点 v 进行聚类,这样得到初始解的质量最差,但多样性最好;若 $\alpha=1$,那么该算法是纯贪心的,因为每次总是选取未分配的顶点中度数最大的顶点 v 进行聚类,这样得到的初始解质量最好,但多样性最差;为了平衡初始解的质量和多样性,一般来说,取 $\alpha\in\{0.25,0.5,0.75\}$.

2.3 局部搜索阶段—FM 算法

本文采用 FM 算法[4]作为分散搜索框架中解的改进方法(Improvement Method),用它对分散搜索过程中的初始解、以及种群进化阶段产生的新解进行局部搜索,提高解的质量.

定义一个顶点的增益为移动该顶点到另一个子集目标函数值的减少量.FM 算法每次通过移动增益最大的顶点来改进当前的划分,且按 pass 来组织.在一个 pass 中,每个顶点只能被移动一次,称为一次 move.当顶点被移动之后,在当前 pass 中,它被锁定(不能再被移动),直到下一次 pass 开始时才被重新释放.在一个 pass 结束时,所有顶点都被锁定,此时从所有的 move 中找到前 k 个 move,使得 $G=\sum_{i-1}^{k} g_i$ 最大,其中 g_i 是所移动的顶点的增益,保留这 k 个顶点的移动,并撤销之后 $|V|-k$ 次的移动.注意,某个 $g_j, j\in\{1,\cdots,k\}$,可能是负值,这意味着移动 v_j 会使得当前划分的结果变差,但借此来跳出局部最优解,使得划分的质量进一步提高.整个算法一直重复,直到某次 pass 之后,划分的质量不再提高,算法终止.

2.4 子集产生方法和解的组合方法

2.4.1 子集产生方法

子集产生方法(Subset Generation Method)是解的组合方法(Combination Method)产生新解的基础,它根据某种策略选取参考集中的解构成若干个子集,然后解的组合方法将每个子集中的解合并,产生新的解,因此新解的数量与构造的子集数相等.子集的构造策略有以下 4 种(假设参考集中有 b 个解):

策略 1　选取参考集中两两不同的解构成一个子集,共产生 C_b^2 个子集;

策略 2　在策略 1 两个解组成的子集的基础上,通过从参考集中选择不在此子集中且目标值最好的解加入其中,将每个子集扩展成由三个解组成的子集,子集的数目仍是 C_b^2 个;

策略 3　在策略 2 三个解组成的子集的基础上,通过从参考集中选择不在此子集中且目标值最好的解加入其中,将每个子集扩展成由四个解组成的子集,子集的数目仍是 C_b^2 个;

策略 4　随机选取目标函数值较好的 i 个解构成若干个子集,i 从 5 到 b.

2.4.2 解的组合方法

本文采用策略 1 进行子集的生成,因此设计的组合方法是对两个解进行组合.本文采用长度为 n 的二进制串来表示划分问题的解 x,其中 n 等于超图的顶点数,0 和 1 表示划分的两个集合,x_i 表示第 i 个顶点所在的集合.假设有两个解 j 和 k,cut(j)表示解 j 对应的割集大小,x_i^j 表示解 j 中第 i 个顶点所在的集合,cut(k)和 x_i^k 同理;算法 3 描述了新解的组合算法.

算法 3　解的组合算法

输入:两个解 j 和 k

输出:新解 m

1. 计算 $r=\text{cut}(j)/(\text{cut}(j)+\text{cut}(k))$;
2. If cut(j) < cut(k) DO
3. For $i=0$ to n
4. Begin
5. If(random(0,1) > r && 满足平衡约束式(1)和式(2)) $x_i^m=x_i^j$
6. Else $x_i^m=x_i^k$;
7. END
8. If cut(j) > cut(k) DO
9. For $i=0$ to n
10. Begin
11. If(random(0,1) > r && 满足平衡约束式(1)和式(2)) $x_i^m=x_i^k$
12. Else $x_i^m=x_i^j$;
13. END
14. Return m.

算法 3 在合并解 j 和解 k 之前,先对比二者的目标函数值,如果 j 的目标函数值优于 k(步骤 1),那么新解 m 的每一位以较大的概率取 j 对应位的值;相反地,如果 k 的目标函数值优于 j(步骤 8),那么新解 m 的每一位以较大的概率取 k 对应位的值.并且在确定新解 m 每一位的值的时候,都要判定子集面积约束(步骤 5 和 11),这样就能保证所产生的新解一定是可行解.

2.5 参考解集的产生和更新

参考解集(Reference Set)是整个分散搜索算法的核心,是一个类似于种群的概念.通过不断更新参考解集,保证种群不断地向好的方向进化.算法 1 中的步骤 4 和步骤 8 分别对应了初始参考解集的产生和分散搜索过程中参考解集的更新,虽然在其中将二者都称作参考集更新方法(Reference Set Update Method),但从算法角度看,二者还是不同的.

2.5.1 构造初始参考解集

本文通过以下三步来构造初始的参考解集(RefSet):

(1)利用算法 2 得到初始解集 P,并用 FM 算法对 P 中所有的初始解进行优化,得到优化后的解集 P^*;

(2)从 P^* 中选择 $b/2$ 个质量最好的解,加入 RefSet;

(3)再从 $P^* \setminus \text{RefSet}$ 中选择 $b/2$ 个与 RefSet 中已有的解差异最大的解加入其中.

步骤 2 中所谓质量最好的解是指目标函数值最小的解,这 $b/2$ 个解加入 RefSet 后就将它们从 P^* 中删除.在步骤 3 中,从剩余的 P^* 中再选择 $b/2$ 个与 RefSet 中的解差异性最大的解是为了保证初始的参考解集具有多样性.为进行该多样性选择,本文定义了距离的概念来衡量解之间的差异.

假设超图 $H(V,E)$,以最小割为目标的两个解 P_1 和 P_2,对应的割集为 C_1 和 C_2,那么 P_1 和 P_2 之间的距离定义如式(4):

$$\text{Distance}(P_1,P_2)=|(C_1\cap(E-C_2))\cup(C_2\cap(E-C_1))| \tag{4}$$

其中 $E-C_1$ 和 $E-C_2$ 表示未被割的超边的集合.

2.5.2 参考解集的更新

当初始参考解集构造完成后,整个分散搜索的过程其实就是对参考解集不断更新的过程,主要的步骤如下:

(1)将所产生的新解都放入集合 NewSet;

(2)利用 FM 算法对 NewSet 中所有新解进行优化;

(3)将 NewSet 和 RefSet 合并,然后清空 RefSet;

(4)从 NewSet 中选择 $b/2$ 个质量最好的解加入 RefSet;

(5)再从 NewSet \ RefSet 中,选择 $b/2$ 个与 RefSet 中已有的解差异最大的解加入其中,清空 NewSet.

3 实验结果与分析

3.1 测试用例

本文用 18 个 IBM 电路实例作为测试用例[17],其模块数从 12752 到 210613 不等,具体属性见表 1.在计算中假设每个模块的面积相同.算法采用C++实现,实验环境为 PC Intel Pentium4 3.0GHz,2GB 内存.

3.2 实验结果与分析

在第一组实验中,我们以电路 IBM01 为例对算法 2 进行测试.以"纯随机"方法产生的初始解作为基准,对比当参数 α 的值分别取 0.25、0.5、0.75 时产生的初始解的质量和多样性.在实验中,共产生 100 个初始解,表 2 中的"质量"是指初始解的平均割集大小,"多样性"是指两两不同的初始解之间距离的平均值,距离定义如式(4).

从表 2 可以看出,纯随机方法产生的初始解的质量最差,但是多样性最好.当 $\alpha=0.75$ 时解的质量与多样性取得比较好的平衡,因此本文 α 取 0.75.

表 1 ISPD98 测试电路属性

电路名称	模块数	线网数	引脚数
IBM01	12752	14111	50566
IBM02	19601	19584	81199
IBM03	23136	27401	93573
IBM04	27507	31970	105859
IBM05	29347	28446	126308
IBM06	32498	34826	128182
IBM07	45926	48117	175639
IBM08	51309	50513	204890
IBM09	53395	60902	222088
IBM10	69429	75196	297567
IBM11	70558	81454	280786
IBM12	71076	77240	317760
IBM13	84199	99666	357075
IBM14	147605	152772	546816
IBM15	161570	186608	715823
IBM16	183484	190048	778823
IBM17	185495	189581	860036
IBM18	210613	201920	819697

表 2 初始解质量与多样性的对比

	质量(Quality)	多样性(Diversity)
Random	9233.6	6823.3
$\alpha=0.25$	6632.5	5988.7
$\alpha=0.5$	6575.7	5924.6
$\alpha=0.75$	6323.7	5924.4

在第二组实验中,我们对比本文的基于分散搜索的划分算法 SS 和最著名的多级划分算法 hMetis 的计算结果.hMetis 是基于多级框架的划分算法,其粗化和细化过程可以选择不同的策略,本文采用文献[18]中的默认设置.文献[12]指出,为了保证 SS 算法的效率,初始解集 P 的大小一般取 100,参考解集 RefSet 大小不超过 20.因此,在第二组实验中,取参数 $|P|=100$,b 分别取 10 和 20.表 3 和表 4 分别列出 hMetis 算法和 SS 算法在子集容量偏差比 β 为 10% 和 2% 的划分结果,算法对每个测试用例运行 20 次,其中 Min 表示计算 20 次所找到的目标函数最小值,Avg 表示计算 20 次所得到的目标函数值的平均值,CPU 表示计算 20 次所耗费时间的平均值(s).

从表 3 和表 4 可以看出,无论 β 取 10% 或是 2%,本文提出的 SS 算法对 18 个测试用例的平均划分结果都要优于 hMetis 算法,只是在个别的测试用例上的最小划分结果不如 hMetis.而且从二者最小值和平均值的对比还能看出,SS 算法的最小值与平均值的差距较小,因此其稳定性要优于 hMetis 算法.

表 3　当 $\beta=10\%$ 时 hMetis 和 SS 计算结果对比

Circuit	hMetis			SS($b=10$)			SS($b=20$)		
	Min	Avg	CPU	Min	Avg	CPU	Min	Avg	CPU
IBM01	181	188	3.4	180	182	48.1	180	181	67.3
IBM02	262	292	6.8	262	262	59.2	261	262	82.9
IBM03	959	1068	7.8	954	972	62.1	922	963	86.9
IBM04	542	588	7.9	538	551	79.6	511	523	111.4
IBM05	1690	1715	9.1	1682	1696	103.4	1596	1633	144.8
IBM06	851	888	17.7	852	881	226.3	831	877	316.8
IBM07	848	930	18.8	849	859	523.8	848	855	733.3
IBM08	1112	1142	25.7	1123	1137	644.8	1078	1126	902.7
IBM09	624	685	13.9	623	632	588.6	619	635	824.0
IBM10	1265	1573	32.8	1275	1356	852.3	1278	1354	1193.2
IBM11	963	1146	25.3	974	1048	842.8	963	1012	1179.9
IBM12	1899	2123	31.8	1932	1998	1056.6	1871	1986	1479.2
IBM13	841	979	35.5	849	870	1522.6	833	846	2131.6
IBM14	1789	1837	77.8	1782	1811	3622.2	1758	1800	5071.1
IBM15	2450	2796	99.0	2496	2580	3822.1	2336	2333	5350.9
IBM16	1758	2339	123.3	1748	1789	4522.3	1726	1755	6331.2
IBM17	2341	2430	150.6	2212	2288	4822.3	2237	2253	6751.2
IBM18	1528	1669	99.2	1521	1528	5212.6	1518	1524	7297.6

表 4　当 $\beta=2\%$ 时 hMetis 和 SS 计算结果对比

Circuit	hMetis			SS($b=10$)			SS($b=20$)		
	Min	Avg	CPU	Min	Avg	CPU	Min	Avg	CPU
IBM01	203	274	2.5	203	222	53.4	203	219	70.5
IBM02	353	384	6.4	349	356	65.7	349	351	86.7
IBM03	957	1048	7.2	980	1008	68.9	960	993	91.0
IBM04	595	660	7.3	608	628	88.4	588	619	116.6
IBM05	1733	3467	9.1	1733	1748	114.8	1722	1736	151.5
IBM06	978	1116	10.3	982	1011	251.2	974	996	331.6
IBM07	951	1043	16.3	925	966	581.4	932	968	767.5
IBM08	1141	1217	26.1	1142	1151	715.7	1128	1134	944.8
IBM09	629	670	15.5	657	701	653.3	644	690	862.4
IBM10	1333	1549	30.7	1389	1432	946.1	1352	1411	1248.8
IBM11	1075	1307	30.4	1138	1384	935.5	1064	1363	1234.9
IBM12	2014	2297	33.8	2098	2169	1172.8	2056	2136	1548.1
IBM13	860	1100	34.3	854	902	1690.1	854	888	2230.9
IBM14	1897	2185	82.3	1886	1976	4020.6	1882	1946	5307.2
IBM15	3007	3520	106.3	2876	3300	4242.5	2848	3251	5600.1
IBM16	2309	2571	122.3	2136	2178	5019.8	2136	2145	6626.1
IBM17	2479	2719	162.5	2463	2706	5352.8	2456	2665	7065.6
IBM18	1603	1818	156.1	1602	1653	5786.0	1602	1628	7637.5

SS 算法的计算结果与参考解集的大小 b 是成正相关的,b 越大,算法的计算结果就越好,然而所耗费的时间也会相应的增加,因此 b 的取值不应过大.

从对比中我们可很明显看出,SS 算法在效率上不及 hMetis 算法,这是由分散搜索算法的框架和多级算法框架的差异造成的.前者的设计目标是以追求更好计算结果为导向的,而后者则是追求效率.但是,在目前的一些大规模集成电路设计中,需要对电路进行并行仿真,这一工作的周期往往长达数月,而且前期对电路划分的质量很大程度上影响了后期并行仿真的相率,因此不断地提高划分算法的质量仍是十分有意义的.

4　结论

本文将分散搜索方法应用于电路划分问题的研究,在分散搜索算法的框架中,嵌入了基于顶点移动的迭代方法—FM 算法,并且提出了采用 GRASP 和聚类相结合的方法,以满足分散搜索方法对初始解的质量和多样性的要求.从实验结果的对比可以看出,本文提出的方法能取得较好的结果.因此用分散搜索方法来解决电路划分问题是行之有效的.在今后的研究中,可以从两方面进行改进:第一,针对划分问题设计更有效的聚类方法,如用文献[19]的量子进化聚类算法等,改进整体初始解的质量,从而提高整个算法的结果.第二,在参考解集的更新操作中,对于合并操作产生的新解,引入适当的选择策略,没有必要对全部的新解都进行改进,以此提高算法的整体效率.

参考文献

[1] 王真,江建慧.基于概率转移矩阵的串行电路可靠度计算方法[J].电子学报,2009,37(2):241 – 247.
WANG Z, JIANG J H. A serial method of circuit reliability calculation based on probabilistic transfer matrix[J]. Acta Electronica Sinica, 2009, 37(2): 241 – 247. (in Chinese)

[2] GAREY M, JOHNSON D. Computers and Intractability: a Guide to the Theory of NP-completeness[M]. New York: W. H. Freeman Company, 1979. 209 – 209.

[3] KERNIGHAN B W, LIN S. An efficient heuristic procedure for partitioning graphs[J]. Bell System Technical Journal, 1970, 49: 291 – 307.

[4] FIDUCCIA C M, MATTHEYSES R M. A linear time heuristic for improving network partitions[A]. Proceedings of the 19th ACM/IEEE Design Automation Conference[C]. New York: The Institute of Electrical and Electronics Engineers Press, 1982. 175 – 181.

[5] ALPERT C J, HUANG J H, KAHNG A B. Recent directions in netlist partitioning[J]. Integration, the VLSI Journal, 1995, 19(1): 1 – 81.

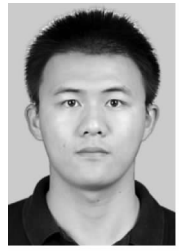
[6] MORAGLIO A, KIM Y H, YOON Y, et al. Geometric crossovers for multiway graph partitioning[J]. Evolutionary Computation, 2007, 15(4): 445 – 474.

[7] 罗胜钦,马萧萧,陆忆.基于改进的 NSGA 遗传算法的 SOC 软硬件划分方法[J].电子学报,2009,37(11):2595 –

2599.
LUO S Q, MA X X, LU Y. An advanced non-dominated sorting genetic algorithm based SOC hardware/software partitioning [J]. Acta Electronica Sinica, 2009, 37(11): 2595 – 2599. (in Chinese)
[8] KARYPIS G, AGGARWAL R, KUMAR V, SHEKHAR S. Multilevel circuit partitioning[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1998, 17(8): 655 – 667.
[9] hMetis[DB/OL]. http://glaros.dtc.umn.edu/gkhome/metis/hmetis/download, 2012.
[10] MLPart [DB/OL]. http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/Partitioning/MLPart/, 2012.
[11] GLOVER F. Heuristics for integer programming using surrogate constraints [J]. Decision Sciences, 1977, 8(1): 156 – 166.
[12] LAGUNA M, MARTI R. Scatter Search: Methodology and Implementations in C[M]. Massachusetts: Kluwer Academic Publishers, 2003. 185 – 218.
[13] MARTI R, DUARTE A, LAGUNA M. Advanced scatter search for the max-cut problem [J]. INFORMS Journal on Computing, 2009, 21(1): 26 – 38.
[14] ALPERT C J, MEHTA D P, SAPATNEKAR S S. Handbook of Algorithms for Physical Design Automation[M]. Florida, USA: Auerbach Publications, 2008. 109 – 134.
[15] FESTA P, RESENDE M G C. An annotated bibliography of GRASP Part I: Algorithms [J]. International Transactions in Operational Research, 2009, 16(1): 1 – 24.
[16] DUTT S, DENG W. Cluster-aware iterative improvement techniques for partitioning large VLSI circuits [J]. ACM Transactions on Design Automation of Electronic Systems, 2002, 7(1): 91 – 121.
[17] ALPERT C J. The ISPD98 circuit benchmark suite[A]. Proceedings of the International Symposium of Physical Design [C]. New York: ACM Press, 1998. 80 – 85.
[18] KARYPIS G, AGGARWAL R, KUMAR V, SHEKHAR S. hMetis: A Hypergraph Partitioning Package, Version 1.0[R]. Minneapolis, USA: Department of Computer Science and Engineering, University of Minnesota, 1997.
[19] 李阳阳,石洪竺,焦李成,马文萍.基于流形距离的量子进化聚类算法[J].电子学报,2011,39(10):2343 – 2347.
LI Y Y, SHI H Z, JIAO L C, MA W P. Quantum inspired evolutionary clustering algorithm based on manifold distance [J]. Acta Electronica Sinica, 2011, 39(10): 2343 – 2347. (in Chinese)

作者简介

朱文兴 男,1968 年 5 月生,博士,教授,博士生导师.主要从事 NP 困难离散与连续全局优化问题的算法设计与分析,以及超大规模集成电路计算机辅助设计中的算法等方面的研究.已在 IEEE T CAD, IEEE T SMCC, INFORMS J Computing 等国内外重要学术刊物发表论文 40 多篇.
E-mail: wxzhu@fzu.edu.cn



程 泓 男,1987 年 8 月生,福州大学数学与计算机科学学院计算机软件与理论专业硕士生,研究方向为智能计算与智能软件.
E-mail: muredheart@qq.com