

一种基于 GEP 的演化硬件复杂电路优化算法

李康顺^{1,2}, 梁九生¹, 张文生², 李元香³

LI Kang-shun^{1,2}, LIANG Jiu-sheng¹, ZHANG Wen-sheng², LI Yuan-xiang³

1.江西理工大学 信息工程学院,江西 赣州 341000

2.中国科学院 自动化研究所,北京 100083

3.武汉大学 软件工程国家重点实验室,武汉 430072

1.School of Information Engineering, Jiangxi University of Science & Technology, Ganzhou, Jiangxi 341000, China

2.Institute of Automation, Chinese Academy of Sciences, Beijing 100083, China

3.State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China

LI Kang-shun, LIANG Jiu-sheng, ZHANG Wen-sheng, et al. Optimization algorithm for complicated circuit based on GEP. *Computer Engineering and Applications*, 2008, 44(18): 83-86.

Abstract: Evolutionary Hardware (EHW) is a new focus in recent research work. The new method of design hardware is to combine evolution algorithm with programmable logic device. Optimization gate circuit is a main research domain of EHW. A new algorithm to optimize the complicated circuit by using Gene Expression Programming (GEP) is proposed in this paper. The experiments demonstrate that this algorithm has not only fast convergent speed but also it can be used to optimize large scale of circuit structure efficiently and conquers the slow convergent speed even no convergence compared with traditional methods.

Key words: Gene Expression Programming (GEP); Evolutionary Hardware (EHW); optimization circuit

摘要:演化硬件是近年来新兴的研究热点,它是演化算法和可编程逻辑器件相结合而形成的硬件设计新方法。在演化硬件中门电路的优化设计是一个重要的研究领域。提出一种新的基于基因表达式程序设计(GEP)的算法来进行复杂优化电路的设计,通过仿真实验表明,该算法不仅收敛速度快,而且还能利用该算法优化大规模的门电路,克服了传统优化方法的求解速度慢甚至不收敛等缺点。该算法较传统的电路优化方法更简单、更高效。

关键词:基因表达式程序设计;演化硬件;优化电路

DOI:10.3778/j.issn.1002-8331.2008.18.026 **文章编号:**1002-8331(2008)18-0083-04 **文献标识码:**A **中图分类号:**TP302;TN702

演化硬件^[1,2](Evolvable Hardware, EHW)是在 20 世纪 90 年代提出的一种新的硬件设计方法。演化硬件可以用下面的公式来形象地定义演化硬件^[3]: $EAs+PLDs=EHW$,即:演化算法+可编程逻辑器件=演化硬件。

电路的演化设计基本思想是将电路的结构和参数等作为演化算法的染色体位串来进行编码,将电路要实现的功能作为个体的适应度,通过演化遗传操作来得到满足特定逻辑功能的电路。数字电路是演化硬件的一个重要方面,演化算法能有效地对复杂电路进行优化。

逻辑电路是集成电路的主要组成部分,它可以用真值表或逻辑函数来描述。在实际工作中,常用代数法或卡诺图法进行优化,但当功能复杂,变量较多时它们的优化就比较困难。本文用遗传基因表达式来实现对复杂电路进行优化。

1 逻辑函数优化的方法^[4]

一个电路各个门电路之间的关系可以利用逻辑代数抽象为数学表达式,得出相应的逻辑函数,根据逻辑函数的不同特点,可以用多种方法表示逻辑函数:

(1)函数表达式,对于逻辑关系比较复杂、逻辑变量比较多的逻辑问题,是一种简洁的表示方法,便于利用公式进行逻辑运算和化简,如例 1, $F=A\bar{C}+ABC+ACD+CD$ 。

(2)真值表,它可以较容易根据逻辑功能直接列出来,抽象成为数学问题,直观、明了。

(3)逻辑图,逻辑图中的逻辑符号和实际的电路器件有明显的对应关系,比较接近工程实际,可比较方便地把它转化为实际的电路。

(4)卡诺图,则用几何位置的相邻性来形象地表示了组成

基金项目:国家重点基础研究发展规划(973)(the National Grand Fundamental Research 973 Program of China under Grant No.2004CB318103); 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60133010); 江西省研究生创新基金项目(No. YC07A073)。

作者简介:李康顺(1962-),男,CCF 高级会员,中科院自动化所在站博士后,博士,教授,硕士生导师,研究方向为演化计算、神经网络;梁九生(1971-),硕士研究生,研究方向为演化计算;张文生(1966-),男,研究员,博导,研究方向为机器学习理论与算法;李元香(1962-),男,教授,博导,研究方向为并行计算、演化计算。

收稿日期:2007-09-25 **修回日期:**2007-12-10

逻辑函数的各个最小项之间在逻辑上的相邻性,从而可以很容易地求出函数的最简与或式。这几种方法在本质上是相通的。

实际工作中,为了使得到的电路更简单和紧凑。常用公式化和卡诺图化对逻辑函数进行化简来得到比较简单的逻辑函数。传统的方法常用代数法或卡诺图法进行化简。用代数法化简例 1 如下:

$$F = A\bar{C} + ABC + A\bar{C}D + CD = A(\bar{C} + BC) + C(\bar{A}D + D) = A(\bar{C} + B)(\bar{C} + C) + C(A + D)(\bar{D} + D) = A\bar{C} + AB + AC + CD = A(\bar{C} + C) + AB + CD = A(1 + B) + CD = A + CD$$

例 1 用卡诺图方法化简如图 1。

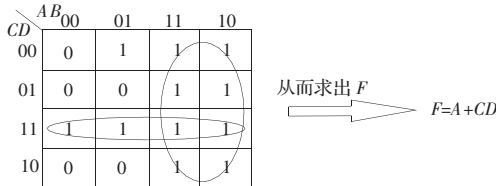


图 1 卡诺图化简

公式化是利用逻辑代数的公式和定理对逻辑函数化简来得到最简的表达式,它没有任何的局限性,但在化简一些复杂的逻辑表达式时不仅需要熟练的运用公式和定理,而且需要有一定的技巧,没有一定的规律和步骤,也比较难以判断化简结果是否最简。卡诺图化简的方法比较简单和直观易于避免差错。但这些方法有一个共同的缺陷就是在逻辑变量多于 4 个时进行化简时变得非常困难。

2 演化硬件复杂电路的优化设计

2.1 演化硬件的设计方法

在演化硬件中,以可编程逻辑器件为物质基础,它的基本流程与演化算法相似。它仅以编码和对适应值函数的评估来得到所需功能的电路结构,而不需要先验知识和人工干预。因此可获得更大的设计空间,更好的设计结构和更低的设计成本。电路演化设计时,它将结构位串看作演化算法中的染色体,将要实现的功能当作演化算法的评价函数,利用演化算法实现电路的自动设计。如图 2 所示。

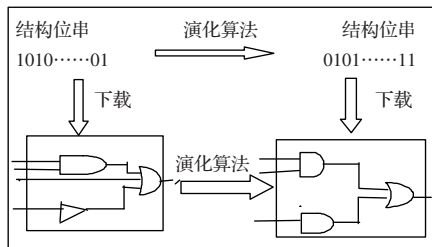


图 2 电路演化设计

在算法中,个体的适应度定义为电路的输入和输出特性与预期结果的符合情况。适应度评估就是对通过真值表进行计算,通过软件仿真来获得对应的输出序列,从而求得个体的适应值。因此,在用演化算法技术逻辑电路设计时,主要关键的技术就是编码的设计和适应度评估函数的设计,本文采用与真值表功能的符合程度来实现逻辑门电路功能的评估。

2.2 用基因表达式编程^[5](Gene Expression Programming, GEP)方法实现复杂电路的优化设计

遗传算法(GA)包括遗传程序设计方法(GP)^[6]和基因表达

式编程算法(GEP)等,GP 在建模^[7]等方面取得很多重要的成果,但在对基因操作时会产生无效的基因,要进行些剪裁等操作而比较复杂。而基因表达式编程,在进行遗传变异等操作时都能得出语法正确的程序,而不需要复杂的编辑过程。从而在实现时更为简单和高效。它实现的技术主要包括编码方式、K 表达式、各种算子及适应度函数的选择。

2.2.1 编码方式

GEP 的基因由头部和尾部构成。头部既可以含有函数符号,也可以含有终点符号,而在尾部只能是终结点符号。表达式树这种编码方法,可以避免产生无效表达式树,Zuo 等^[8]对此给出了理论证明。基因 A 组成如表 1。对于每个个体的编码假设头部的长度设定为 h ,而且这个 h 的长度是根据问题需要而选择的。尾部的长度则是关于头部长度 h 和 n (所需变量数最多的函数的参数个数)的一个函数,可以由以下公式来表达: $t = h(n-1) + 1$ 。GEP 染色体(Chromosome)通常由等长的多个基因构成。对于每个程序或者每次运行,基因的数目以及头部的长度都是固定的。每个基因可以由表达式树构成一个子表达式树(sub-ET)而且子表达式树相互作用构成一个更复杂的多子树(Multi-subunit)的表达式树,子表达式可以根据问题的具体情况采用一般的四则运算或者用逻辑运算符来连接。由于 GEP 的染色体可以由多个子表达式树来组成,所以 GEP 能有效地解决非常复杂的问题。本文的基因由函数符号{&,!,!,}和终结点符号{a,b,c,d,...}组成。

表 1 基因 A 的基因组成

基因序号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
基因位值	+	Q	-	/	b	*	a	a	Q	b	a	a	b	a	a	b	b	a	a	a	b
基因分布	头部基因										尾部基因										

2.2.2 K 表达式树

按照从左到右的顺序逐个读取基因中的字符,并按照层次顺序,从上到下,从左到右构成表达式树(Expression Tree, ET)。按照此规则,则基因 A 可以转化成如图 3 所示的表达式树。可以发现,表达式树的最终结点与基因的最终结点不一定是重叠的,它往往要靠前。称起点到表达式树的终点的长度为这个基因的有效长度。显然,在这个例子中,表达式树的终点是 a,在最后一个结点之前。从起点到终点 a 的长度为 10,即有效长度为 10。把它按从左到右,从上到下的顺序写出的表达式为 +Q-/b*aaQba,就称为 K-表达式。

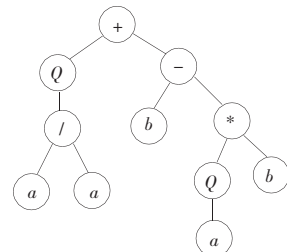


图 3 表达式树(1)

在计算 K-表达式的值时,Ferreira 采用 ET 树与 K-表达式之间的相互转化计算表达式值,这种方法的计算量和复杂性比较大。本文在对 K-表达式值计算时采用 GRMC^[9](Gene Read & Compute Machine)方法,该方法的特点是不需要把染色体转换为表达式树,而是直接对染色体进行操作得到适应值,从而

在计算适应值时更为快速和方便。

2.2.3 演化算子的设计

算子的设计与所采用的表示方法有关,主要有变异算子等,假设在位置 9 发生了变异,由“b”变成了“+”,基因 A 变异后的组成如表 2。将此时的基因表示成 ET,如图 4 所示。

表 2 基因 A 在位置 9 变异后基因组成

基因序号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
基因位值	+	Q	-	/	b	*	a	a	Q	+	a	a	b	a	a	b	b	a	a	a	b
基因分布	头部基因										尾部基因										

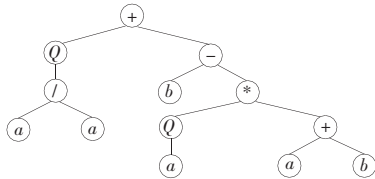


图 4 表达式树(2)

在这个例子中,终点向右移了两位(位置 12)。在遗传操作中,只要尾部变异不出现函数符号,则该染色体的 K-表达式一定是一个有效的表达式。当然,运算中出现的被 0 除、负数开平方等不认为是表达式无效。因此,在进行任何算子操作时要保证都将得到有效的 ET,基因结构化的组织结构必须被维持,即必须使头部和尾部分开,且不允许函数集中的符号出现在尾部。

2.2.4 适应度评价函数的设计

在 GEP 中,适应度函数设计非常重要。对于许多数学的应用来说,为了发现一个好的解,使用小的相对或者绝对误差是有益的。但是,假如选择带过分狭窄,种群的进化非常缓慢甚至不能找到正确的解。另一方面,放宽选择带将会出现众多与优良解相去甚远的解。为了解决这个问题,设计出了进化策略(ES)允许各个个体的适应度直接取自它所对应的目标函数,每次从群体中选取最好的几个个体,将它们保存到下一代的种群中。经典 GEP 算法有 3 种适应度计算函数:

基于绝对误差的适应度函数

$$f_i = \sum_{j=1}^{c_i} (M - |C_{(i,j)} - T_{(j)}|) \quad (1)$$

基于相对误差的适应度函数

$$f_i = \sum_{j=1}^{c_i} \left(M - \left| \frac{C_{(i,j)} - T_{(j)}}{T_{(j)}} \cdot 100 \right| \right) \quad (2)$$

基于逻辑合成问题

$$\text{If } n \geq \frac{1}{2} C_{(i)} \text{ then } f_i = n; \text{ else } f_i = 1 \quad (3)$$

其中, M 为一常量,是控制适应度 f_i 的取值范围; $C_{(i,j)}$ 表示第 i 个基因对应的。

函数表达式中利用第 j 个样本变量数据求得的函数值; $T_{(j)}$ 表示第 j 个样本中包含的实际测得的该目标函数的真实值; C_i 是测试样本数据总数, n 是正确适例的个数。式(1)和式(2)可以解决任何一个符号回归问题。式(3)主要解决逻辑合成问题。

在本文中,主要是讨论逻辑复杂电路的优化,因此,将图 3 中的函数符号和终结点符号转换成逻辑表达式中的运算符和

操作符后,就代表一种单输出的逻辑电路。在给定真值表后,就可以将实际电路值计算输出,实际输出与真值表相同的数目定义为适应度。

2.3 算法步骤

(1)初始化种群,随机产生 100 组(过大会增大算法运行时间,过小则很难收敛)初始染色体。

(2)按照 GCRM 方法求解初始种群的各染色体的适应值,本文以得到的逻辑函数满足真值表的个数越多和染色体的有效长度越小适应值函数越大。保存适应值最高的个体。

(3)执行变异,以轮盘赌原理来选择变异的基因,然后,随机产生一个 0 到 1 之间的数 $randompm$ 判断此数是否大于变参数的变异率 p_m ,若 $randompm > p_m$,则又随机产生一个数 $randommc$ 为当前基因变异的位置,在基因头部,任何符号都可以变异成函数符号或终结点。在基因尾部,终结点只能变异成终结点,在本文中仅变异一个基因位。

(4)随机产生一个 0 到 1 之间的数 $randomps$,当此数大于插串的概率时,执行 IS 插串、RIS 插串、Gene 插串。

(5)随机产生一个 0 到 1 之间的数 $randompr$ 当此数大于重组的概率时,执行单点重组、两点重组和基因重组。

(6)如果运行达到预先设定的最大代数或者获得了最大适应值且基因的有效长度短,则停止运行,输出结果并将结果保存到记录文件中,否则转到(2)继续运行。

因此,用 GEP 实现门电路的优化设计与传统的方法相比,有很大的不同,主要区别是:用 GEP 实现复杂电路的优化设计不需要事先知道电路知识(如:电子学、电工学等知识),只需要根据问题的特征确定一些基本组成单元,把能实现所需功能的电路看作为“黑箱”,只要能实现所需要的功能,不必了解电路的细节;其次,用 GEP 实现复杂电路的优化设计不是采用经典的数学方法求解问题,而是采用演化算法搜索问题的最优解,这与传统的优化方法在思想上、方法上都有本质的区别。

3 实例仿真与分析

本文就以上算法设计思想,对所需要实现的逻辑功能的复杂电路,用 GEP 方法优化进行了软件仿真实验,并取得了良好的结果。在仿真实验中,所用的参数定义如表 3 所示。

实验 1 对例 1 用 GEP 优化。基因由 {&,!,a,b,c,d} 构成。

在仿真实验中得到数据如表 4,程序在第 49 代找到最优个体 $!a!c!d!d!b!a!b!c!b!d!c!b!c!a!b$,其变量个数为 3,门数为 2,即得到逻辑函数 $F=A+CD$,与用代数化简所得到的最简式相同,程序找到最佳个体计算完成的时间只用了 0.34 s。说明这种方法是可行的,而且也是高效的。

表 4 实验 1 的实验结果

个体	满足真值表的个数	门个数	变量个数
$!a!c!d!d!b!a!b!c!b!d!c!b!c!a!b$	15	8	10
$!a!c!d!c!a!b!b!d!d!c!c!b!a!b$	16	3	4
$!a!c!d!d!d!b!a!b!c!b!d!c!b!c!a!b$	16	2	3

实验 2 利用随机函数生成一个 4 变量的随机真值表,用 GEP 方法优化,仿真实验的数据如表 5。

表 3 参数定义

参数名	演化代数	种群数目	函数符号集	基因头部长度	变异率	单点重组率	双点重组率	IS 插串率	IS 插串长度	RIS 插串率	RIS 插串长度
参数值	2 000	100	'&','!'	10	0.04	0.3	0.3	0.1	1,2,3	0.1	1,2,3

表5 实验2的实验结果

个体	满足真值表的个数	门个数	变量个数
&! lca! dbdecdbdbdecdbb	14	3	3
&! laab!&acccb!ddcbdc	16	3	3
! ! ab! dbdadcbdbdecda	16	3	2

这个实验说明了用基因表达式编程化简随机生成的真值表,同样可获得不同满足真值表的逻辑表达式,而且在满足所有真值表时也可以获得具有不同的表达形式的逻辑表达式。所得到符合真值表的逻辑函数表达式也比较简单。

实验3 在逻辑函数的变量大于5时,一般的代数化和卡诺图化在优化时就很难,。对随机生成的5个变量逻辑函数用GEP方法求解时,在用基因表达式编程优化仿真实验中,在1143代时找到最优个体,仿真实验的结果如表6。

表6 实验3的实验结果

个体	满足真值表的个数	门个数	变量个数
! ! a&acc! eededeceacbbd	28	3	3
! &b&a&&acblacdbaebce	32	7	7
! &&! bc! aed!alldedceac	32	5	3

上述实验结果表明,用GEP确实能够有效对复杂电路进行优化,得到所需要的功能,而且得到的电路比较简单、紧凑。在实验中可以得到多个满足功能要求的逻辑门电路,对于人们在电路设计中可能具有一定的启发性。也为容错系统的设计提供了新的方法。从输出的实验统计可以发现,在演化的过程中其最小的适应值都比较小,甚至为0,这意味着利用GEP方法在优化演化逻辑门电路时,它的搜索空间是比较大,因此可以避免早熟,有利于找到最优电路。因此,在用一般的代数方法和卡诺图的方法在优化逻辑复杂电路比较困难的情况下,利用GEP的方法来优化是一种可行和有效的新方法。

(上接53页)

3 总结及展望

精确的碰撞检测对提高虚拟环境的真实性、增强用户的沉浸感有着至关重要的作用,而提高速度是虚拟环境中碰撞检测的核心问题。围绕提高碰撞检测的速度,提出了对碰撞检测进行预处理的方法,该方法将整个碰撞检测过程分成预处理、估计检测、精确计算三个阶段,排除了许多明显不发生的碰撞,避免了大量不必要的检测计算,提高了实时性能,并提出了在碰撞检测中针对具有不同几何特征的物体选择不同类型包围盒的混合包围盒算法,给出了不同包围盒之间的求交算法,实现了层次包围盒碰撞检测算法的优化,进一步提高了碰撞检测速度。本文优化算法可应用于形状复杂的刚体碰撞检测,并且,当场景规模越大时,算法的改进效果越明显。

本文算法在实际应用中需要注意的是:

(1)时间片的长度和虚拟场景中运动物体的速度上限的选取。场景中运动物体的速度越快,则时间片的长度越短;反之,越长。

(2)将物体视为质点时,距离域的选取。比如当两个物体之

4 结论

本文提出一种新的基于GEP的方法来实现演化硬件复杂电路的优化,根据大量实验的结果表明,在缺少先验知识和人工干预的情形下,对于给定具有一定功能的逻辑电路,特别是对于规模较大的复杂电路用GEP的方法来实现演化硬件复杂电路的优化是一种可行、高效的方法。同时,这种新的方法对于实现电路的自动设计来说也是具有重要的意义。

参考文献:

- [1] 潘正君,康立山,陈毓屏.演化计算[M].北京:清华大学出版社,1998.
- [2] de Garis H.Evolvable hardware in principles and practice,communications of the association for computer machinery[J].CACM Journal,1997(8).
- [3] Virtual Computer Corporation.Hardware object technology development system user's guide[EB/OL].(1998-03).http://www.vcc.com.
- [4] 余孟尝.数字电子技术基础简明教程[M].北京:高等教育出版社,2004.
- [5] Ferreira C.Gene expression programming:a new adaptive algorithm for solving problems[J].Complex System(S0219-5259),2001.
- [6] Cao H,Kang L,Chen Yu J.Evolutionary modeling of systems of ordinary differential equations with genetic programming[J].Genetic Programming and Evolvable Machines,2000,1:309-337.
- [7] 李康顺,李元香,汤铭端,等.遗传程序设计在统计建模中的应用[J].系统仿真学报,2005(7).
- [8] Zuo Jie,Tang Chang-jie,Zhang Tian-qing. Mining predicate association rule by gene expression programming[C]/LNCS 2419:International Conference for Web Information Age 2002,WAIM02. Berlin:Springer-Verlag,2002:92-103.
- [9] 姜大志,吴志健,康立山,等.基因表达式程序设计的GRCM方法[J].系统仿真学报,2006(6).

间的距离是物体本身大小的10倍时,该物体可被视为质点。该值的选取要根据系统的精度要求、应用范围的不同等来确定。

参考文献:

- [1] 石教英.虚拟现实基础及使用算法[J].计算机世界,2002(2).
- [2] 李芙玲,张瑾.碰撞检测技术研究[J].华北科技学院学报,2004,6:71-73.
- [3] 王志强,洪嘉振,杨辉.碰撞检测问题研究综述[J].软件学报,1999,10(5):545-551.
- [4] 马登武,叶文,李瑛.基于包围盒的碰撞检测算法综述[J].系统仿真学报,2006(4):1058-1064.
- [5] 崔汉国,陈军,王大宇.虚拟环境中优化的OBB碰撞检测算法研究[J].计算机工程与设计,2007(11):2524-2526.
- [6] 冯善达,刘怡昕.虚拟场景中碰撞检测实用算法研究[J].计算机仿真,2004(6):100-112.
- [7] 涂超.虚拟空间中的碰撞检测[J].武汉理工大学学报,2001(11):84-87.
- [8] 蔡文军,陈虎.基于混合模型的碰撞检测优化算法研究[J].计算机与现代化,2006(7):49-52.
- [9] 刘涛,王增波,李占利.碰撞检测过程中的包围盒技术及应用研究[J].西安科技大学学报,2006(9):395-399.