

一种新的基于真值表分解技术的电路加速演化策略

孟亚峰, 张峻宾, 蔡金燕
(军械工程学院电子与光学工程系, 石家庄 050003)

摘要: 硬件演化技术在电路演化过程中具有自组织、自适应、自修复性能, 具有重要的工程应用价值, 硬件演化技术能够实现的核心就是演化算法。目前中等规模及大规模电路演化的时候存在演化速度慢等缺陷, 直接影响电路演化的时效性。提出了一种新颖的电路加速演化策略, 即基于真值表分解的电路演化策略, 从不同的角度对电路演化算法进行了改进, 有别于常规的改进演化算法。通过对典型电路进行仿真分析, 证明了所提出的加速演化策略的有效性, 并且具有重要的应用价值。

关键词: 硬件演化; 电路演化; 演化算法; 真值表分解技术; 加速演化

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1671-637X(2016)04-0062-04

A Novel Accelerated Circuit Evolution Strategy Based on Truth Table Separate Technique

MENG Ya-feng, ZHANG Jun-bin, CAI Jin-yan

(Department of Electronic and Optical Engineering, Ordnance Engineering College, Shijiazhuang 050003, China)

Abstract: Evolvable Hardware (EHW) technology has self-organizing, self-adaptive, and self-repairing performance in circuit evolution, and has very important engineering application value. The core of carrying out EHW is Evolutionary Algorithm (EA). When the evolved target circuit is medium-sized circuit or large-scale circuit, the evolution is time-consuming. To solve the problem, a novel accelerated circuit evolution strategy was proposed, which was based on truth table separate technique. Different from the normal algorithm improvement, the new strategy made improvements to the circuit evolutionay algorithm from several aspects. Simulation analysis of typical circuit proved the effectiveness of the proposed strategy.

Key words: Evolvable Hardware (EHW); circuit evolution; Evolutionary Algorithm (EA); truth table separate technique; accelerate evolution

0 引言

随着电子技术的不断发展, 为提高电子电路在各种复杂、恶劣环境的自适应性, 近年来兴起了一种新颖技术——硬件演化(Evolvable Hardware, EHW)技术^[1-3]。

硬件演化公式可表示为硬件演化 = 可编程逻辑器件 + 演化算法。硬件演化是以演化算法为组合优化和全局搜索的工具, 通过模拟进化来获得预期的电路或系统结构。

正因如此, 采用 EHW 技术, 能够使电路具有自适

应、自组织、自修复的能力。可编程逻辑器件(Programmable Logic Device, PLD)是 EHW 的硬件基础, 而演化算法(Evolutionary Algorithm, EA)是 EHW 技术能否实现的关键。

目前, EA 存在演化速度慢、容易陷入局部最优等缺陷, 因而对 EA 进行改进是现在的一个研究热点^[3-4]。改进的目标都是为了加速 EA 的收敛、减少迭代次数, 而较多采用遗传算法及其改进算法和粒子群算法及其改进算法, 或者是采用动态变异率演化算法^[4-5]。在针对大规模电路进行演化的时候, 目前主流演化算法的策略为分而治之。

本文针对组合逻辑电路演化过程中真值表所表现出来的特点, 提出了一种基于真值表分解技术加速电路演化的策略, 并进行了仿真分析。

收稿日期: 2014-04-24

修回日期: 2015-12-29

基金项目: 国家自然科学基金(61271153, 61372039)

作者简介: 孟亚峰(1970—), 男, 河北石家庄人, 副教授、硕导, 研究方向为电子装备状态检测、故障诊断。

1 真值表分解技术加速演化电路基本策略

在实现演化算法的过程中,当面临大规模的真值表的时候,演化电路的规模将随真值表输入端个数的增加呈指数增加。但是,演化收敛的速度将随真值表规模的增加呈指数降低,因此提高具有大规模真值表的电路演化速度将显得尤为重要。对真值表进行必要的拆分,能够在很大程度上提高演化算法的收敛速度。

为了解决复杂电路快速演化问题,TORRESEN J将“分而治之”思想引入到 EHW 技术之中,而拆分真值表技术是分而治之(divide-and-conquer)的深化,在演化过程中,结合二元判决图(Binary Decision Diagram, BDD),将真值表实行一层一层的剥离。

在对真值表进行拆分的时候,不能拆分得太细,因为实现同样功能的电路拓扑结构可能不一样,拆分得太细就不能表现出电路结构的多样性。例如有如下的电路布尔函数: $Y1 = AB + AC$, $Y2 = A(B + C)$,如图1所示。

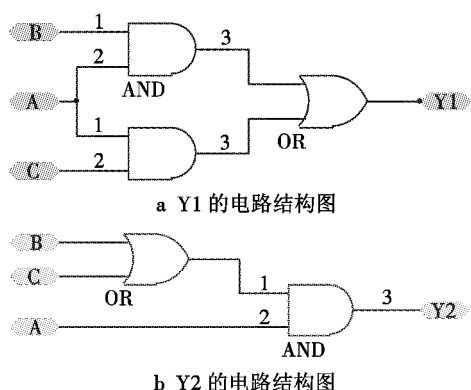


图1 Y1 和 Y2 的电路结构图

Fig. 1 The circuit structures of Y1 and Y2

从 Y1 和 Y2 可以看出,虽然它们有相同的具体功能,但是电路结构完全不一样,所消耗的硬件资源也不一样,各有各的特点。不能直接判定哪一种电路结构好,哪一种电路结构差,需要根据具体的问题环境,找出所需要的电路结构。如果采用真值表分解技术将真值表全部进行分解,将只能得到一种固定不变的电路结构;如果采用真值表分解技术同 EHW 技术相结合,不仅可以演化出相同功能的不同结构的电路,还能加速电路的演化。

为了有利于演化算法的展开,使演化算法的自适应、自组织特点得到有效的发挥,本文采用的方法是将真值表拆分到只剩 4 个输入组合为止,剩余的 4 个输入端再采用演化算法进行演化,得到最终的目标电路。拆分至 4 个输入端的另一个参考依据是目前主流 FP-

GA 的查找表(Look-up Table, LUT)具有 4 个输入端,且目前 FPGA 被作为硬件演化的硬件平台。

虽然 FPGA 的 LUT 相当于一个 ROM,能够直接存储任意 4 输入 1 输出的真值表,本文为了更好地在电路设计过程中对演化算法的自适应自组织特性进行展示,对剩余未拆分电路采用演化算法进行设计。

在基于真值表分解技术的电路演化过程中,根据 BDD 原理,将大规模电路拆解成了关联度较小的小规模电路,将纯粹的电路演化过程变成了电路演化和 BDD 的有效结合过程。而小规模电路在电路演化的时候,约束条件较少,相对于大规模电路的电路演化搜索到目标电路编码将比较容易,演化算法迭代次数将大幅度缩减,同时收敛速度也将加快。

本文中演化算法采用遗传离散粒子群算法(Genetic Algorithm Discrete Particle Swarm Optimization, GADPSO)。GADPSO 演化算法是将遗传算法(Genetic Algorithm, GA)的交叉、变异融合于离散粒子群算法(Discrete Particle Swarm Optimization, DPSO)之中,GA 算法在优化问题方面限制很小,对目标函数及约束要求很低,具有较强的鲁棒性和适应性,虽然收敛速度比较慢,但是能够较好地克服寻优过程中易陷入局部最优的缺点。然而 DPSO 搜索速度较快,因此将 GA 算法和 DPSO 算法相结合能克服相互的缺陷,将 GADPSO 演化算法引入电路演化,能够极大地减少演化目标电路编码所需要消耗的时间^[6-10]。能明显提高粒子群的多样性,有助于改善 DPSO 算法的收敛速度和收敛精度。

2 加速演化策略的步骤

任何一个真值表均能表示成一个布尔函数。首先假设某待演化的真值表具有 M 个输入端($x_1, x_2, x_3, \dots, x_M$), N 个输出端($y_1, y_2, y_3, \dots, y_N$)。而它的布尔函数就是目标函数,此时令目标函数为 $f(x)$,且 $f_1(x)$, $f_2(x)$ 是未知的布尔函数。

演化算法中对真值表的拆分详细步骤如下所述。

1) 判定真值表输入端个数 $input_num = M$, 输出端的个数 $output_num = N$ 。

2) 在真值表中任选一个输入端 x_1 , 并判定输入端 x_1 的取值情况。

3) 如果只存在“0”或者“1”一种取值情况,只将 x_1 输入端从中剥离,剩余的还是一个整体真值表,且 $input_num = M - 1$ 。

4) 如果存在“0”和“1”两种取值情况,则在将 x_1 输入端从真值表剥离的同时,真值表一分为二,此时

$input_num = M - 1$ 。

5) 重复3)和4)的步骤,直到 $input_num = 4$,并得到式(1)。

6) 最终演化电路只针对剩余的4输入真值表进行快速演化。演化出的布尔函数是 $f_1(x), f_2(x)$ 。此时只需对式(1)进行求解,即可得到需要演化的电路布尔函数表达式 $f(x)$,即

$$f(x) = (x_1 + \bar{x}_1)(x_2 + \bar{x}_2) \cdots (x_{M-5} + \bar{x}_{M-5}) \cdot [x_{M-4} \cdot f_1(x) + \bar{x}_{M-4} \cdot f_2(x)] \quad (1)$$

式中:“+”是或运算;“·”是与运算。

需要注意的是,此时式中的 $x_{M-i} + \bar{x}_{M-i}$ ($5 \leq i \leq M-1, i$ 为整数)不一定为0,需要视真值表中 x_{M-i} 取值为0和1的数量不等,则 $x_{M-i} + \bar{x}_{M-i} \neq 1$;相反,如果 x_{M-i} 取值为0和1的数量相等, $x_{M-i} + \bar{x}_{M-i} = 1$ 。

在以上过程中,真值表输入端的分离需要符合约束条件,由演化算法执行实现,且由于 x_{M-i} 的取值0和1的数量不一定相等,此时演化算法的自适应性也将得到体现。

完成真值表拆分之后,紧接着施加 GADPSO 演化算法,如图2所示。

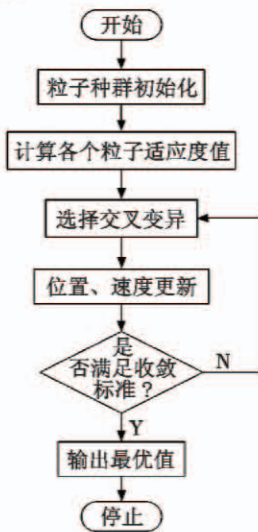


图2 GADPSO 演化策略流程图

Fig. 2 The process of GADPSO evolution strategy

GADPSO 演化算法实现的步骤如下所述。

1) 对粒子种群进行初始化,随机生成符合要求的 M 个电路染色体编码;并设置 GADPSO 演化算法的交叉率 p_c , 变异率 p_d 和惯性权重 w 。

2) 计算每一个粒子的适应度值 $fitness_i$, 其中 $i < M$ 。选取这一代中最佳粒子的位置 g_{best} , 并和全局最优粒子位置 p_{best} 进行比较,保存最优位置为 p_{best} 。

3) 对粒子种群进行遗传算法的基本操作,即进行

选择、交叉、变异操作。

4) 更新各个粒子的位置和速度。

5) 计算每一个粒子的适应度值,并和全局最优值进行比较,对全局最优值进行更新。

6) 根据计算出的适应度值判断算法是否满足收敛标准,如果满足,则输出最终粒子编码;如果不满足,则转入步骤3)。

7) GADPSO 算法结束。

3 仿真验证与分析

为了验证所提出的基于真值表分解技术在电路演化过程中的有效性,选取带进位的两位加法器作为典型电路进行仿真研究,其真值表如表1所示。由于篇幅的关系,表1中只列出了真值表的部分输入输出关系。其输入从00000一直到11111,一共32个组合。

表1 带进位的两位加法器真值表

Table 1 The truth table of two-bit adder with carry

输入		进位		输出
高位	低位	高位	低位	低位
A1	A2	B1	B2	CIN
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
⋮	⋮	⋮	⋮	⋮
1	1	1	0	1
1	1	1	1	0
1	1	1	1	1

在进行真值表分解的时候,由演化算法进行输入端的分离。根据约束条件,演化算法将 A1 输入端选出,A1 有“0”,“1”两种取值,取“0”的时候为 $\bar{A1}$,取“1”的时候为 A1,以此为标准将真值表划分为两部分。对于剩下的输入端仍然采用划分 A1 的标准进行分解,直至未分解的输入端剩余4个即可。由于表1本身只有5个输入端,因而只需要拆分一次真值表即可。

在演化过程中设置适应度函数为

$$V_{fitvalue} = \sum_{i=1}^{2^n} N_{fitnumber_i} \quad (2)$$

式中: n 是待演化电路的输入端个数,有别于输入端的总个数 N ; $N_{fitnumber_i}$ 表示第 i 个输入组合是否满足电路要求,满足记为“1”,不满足记为“0”。因而适应度函数转换为求 $V_{fitvalue}$ 的最大值。

采用 GADPSO 算法进行20次仿真分析,同时设置 GADPSO 算法的相关参数。令初始化样本数 $N = 40$, 变异率 $p_d = 0.05$,交叉率 $p_c = 0.95$,交叉算子采用2点交叉,变异算子采用2点变异。

最后得到仿真数据如图3所示。

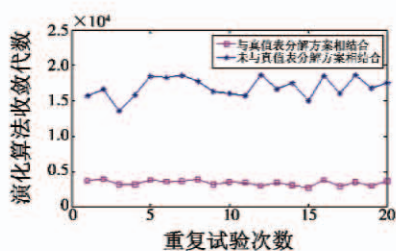


图3 真值表部分分解技术和演化算法迭代次数关系图

Fig. 3 The relationship between partial truth table split technology and the number of EA iterations

从图3中可以清晰地看出,在20次仿真分析中,采用真值表分解技术实现电路演化的平均演化迭代次数都远远小于直接对真值表进行电路演化。

并且数据统计表明:基于真值表部分分解平均演化迭代次数是3407次,然而,没有采用真值表部分分解技术的平均演化迭代次数是16915次。

前者只对真值表进行了一次分解,即待演化的目标电路分解成了两个子电路,并结合式(1),演化出目标电路的布尔函数。通过仿真对比分析,可以看出真值表分解技术能够极大地加速演化算法的收敛速度,在仿真中,基于真值表分解技术在加速电路演化方面的有效性也得到了证明。

4 结论

本文从不同的视角,提出了一种基于真值表分解的加速电路演化策略,有别于常规的改进演化算法。通过仿真分析,证明了所提出的策略在电路加速演化方面的有效性,同时保证了电路的多样性,此技术的优势在组合电路的演化方面得到了很好的体现。

参考文献

[1] 张峻宾,蔡金燕,李丹阳,等. 三进制编码实现硬件演化方法研究[J]. 微电子学与计算机, 2013, 30(3): 1-4. (ZHANG J B, CAI J Y, LI D Y, et al. Research of ternary code implemented in hardware evolution[J]. Microelectronics & Computer, 2013, 30(3): 1-4.)

- [2] LEE Y M, CHOI C S, HWANG S G, et al. Transistor-level evolution of digital circuits using a special circuit simulator [J]. Lecture Notes in Computer Science, 2008, 5216 (1): 320-331.
- [3] ZHANG J B, CAI J Y, MENG Y F, et al. Fault self-repair strategy based on evolvable hardware and reparation balance technology [J]. Chinese Journal of Aeronautics, 2014, 27(5): 1211-1222.
- [4] 张志美. 基于改进的遗传程序设计实现组合电路的进化设计[D]. 长春: 东北师范大学, 2007. (ZHANG Z M. Evolutionary design of combinational circuits based on improved genetic programming [D]. Changchun: Northeast Normal University, 2007.)
- [5] KEYMEULEN D, ZEBULUM R S, JIN Y, et al. Fault tolerant evolvable hardware using field programmable transistor arrays [J]. IEEE Transactions on Reliability, 2000, 49(3): 305-316.
- [6] PIRES E J S, TENREIRO M J A, DE MOURA OLIVEIRA P B. Dynamical modelling of a genetic algorithm [J]. Signal Processing, 2006, 86(10): 2760-2770.
- [7] 周育人, 李元香, 王勇, 等. 基于 $(\mu + 1)$ 演化策略的多目标优化算法[J]. 计算机工程, 2003, 29(18): 1-3. (ZHOU Y R, LI Y X, WANG Y, et al. Multiobjective optimization algorithm based on $(\mu + 1)$ evolutionary strategy [J]. Computer Engineering, 2003, 29(18): 1-3.)
- [8] NIU B, ZHU Y L, HE X X, et al. MCPSO: a multi-swarm cooperative particle swarm optimizer [J]. Applied Mathematics and Computation, 2007, 185(2): 1050-1062.
- [9] CHEN E X, LI J Q, LIU X Y. In search of the essential binary discrete particle swarm [J]. Applied Soft Computing Journal, 2011, 11(3): 3260-3269.
- [10] 张涛, 于雷, 周中良, 等. 基于人工势场启发粒子群算法的空战机动决策[J]. 电光与控制, 2013, 20(1): 77-82. (ZHANG T, YU L, ZHOU Z L, et al. Decision-making of air combat maneuvering based on APF and PSO [J]. Electronics Optics & Control, 2013, 20(1): 77-82.)



请扫描二维码关注我刊