

量子可逆逻辑电路综合的快速算法研究

李志强^{1),2)} 陈汉武¹⁾ 徐宝文¹⁾ 李文骞¹⁾ 王佳佳¹⁾ 刘文杰^{1),3)}

¹⁾(东南大学计算机科学与工程学院 南京 210096)

²⁾(扬州大学信息工程学院 江苏 扬州 225009)

³⁾(南京信息工程大学计算机科学与技术系 南京 210044)

摘 要 可逆逻辑有许多应用,尤其在量子计算领域,量子可逆逻辑电路是构建量子计算机的基本单元,量子可逆逻辑电路综合就是根据电路功能,以较小的量子代价自动构造量子可逆逻辑电路.文中结合可逆逻辑电路综合的多种算法,提出了一种新颖高效的算法,自动构造正极性 Reed-Muller 展开式(RM),在生成量子可逆逻辑电路的解空间树上,采用总体层次遍历,局部深度搜索,借鉴模板优化技术,构造限界函数快速剪去无解或非最优解的分枝,优先探测 RM 中的因子,以极高的效率生成最优电路.以国际公认的 3 变量可逆函数测试标准,该算法不仅能够生成全部最优电路,而且运行速度远远超过同类算法.

关键词 量子电路优化;Reed Muller;可逆逻辑电路;Toffoli 门;量子计算

中图法分类号 TP38 **DOI 号:** 10.3724/SP.J.1016.2009.01291

A Fast Algorithm for Synthesis of Quantum Reversible Logic Circuits

LI Zhi-Qiang^{1),2)} CHEN Han-Wu¹⁾ XU Bao-Wen¹⁾ LI Wen-Qian¹⁾ WANG Jia-Jia¹⁾ LIU Wen-Jie^{1),3)}

¹⁾(School of Computer Science and Engineering, Southeast University, Nanjing 210096)

²⁾(College of Information Engineering, Yangzhou University, Yangzhou, Jiangsu 225009)

³⁾(Department of Computer Science & Technology, Nanjing University of Information Science & Technology, Nanjing 210044)

Abstract Reversible logic finds many applications, especially in the area of quantum computing. Quantum reversible logic circuits are basic elements in quantum computer construction. Synthesis of quantum reversible logic circuits means to automatically construct desired quantum reversible logic circuit with minimal quantum cost. The authors absorb different ideas of reversible logic circuits synthesis and present a novel and efficient algorithm which can automatically derive the positive polarity Reed-Muller expansion (RM). A solution space tree is constructed to create quantum reversible logic circuits. Firstly, floor traversal is applied globally, and depth-first search is used locally. Secondly, according to the technique of template optimization, the bound function is constructed, which can rapidly prune the branches with no or nonoptimal result. Thirdly, factors of RM are first considered, therefore the algorithm can effectively construct optimal result and saves computational cost significantly. Judging by the internationally recognized reversible functions of three variables, the proposed algorithm not only synthesizes all optimal reversible functions, but also runs extremely faster than others of the same kind.

Keywords quantum circuit optimization; Reed Muller; reversible logic circuit; Toffoli gate; quantum computing

收稿日期:2006-07-05;最终修改稿收到日期:2008-05-10. 本课题得到国家自然科学基金(60572071,60873101)、国家自然科学基金会重大研究计划(90412014)、江苏省自然科学基金(BK2008209, BK2007104)和江苏省高校自然科学研究计划(06KJB520137)资助. 李志强,男,1974年生,博士研究生,讲师,主要研究方向为量子计算、量子电路综合. E-mail: yzqlzq@163.com. 陈汉武,男,1955年生,博士,教授,博士生导师,主要研究领域为量子计算、信息论. 徐宝文,男,1961年生,博士,教授,博士生导师,主要研究领域为程序设计语言、软件智能化. 李文骞,男,1979年生,硕士,主要研究方向为可逆电路综合. 王佳佳,女,1981年生,硕士,主要研究方向为量子计算模拟. 刘文杰,男,1979年生,博士研究生,讲师,主要研究方向为量子通信.

1 引 言

量子计算机可等效一个量子图灵机. 理论上已证明, 量子图灵机可等价一个量子逻辑电路. 量子逻辑门的组合与级联是组成量子计算机的基本元素.

所有量子逻辑门均可表示成复变空间酉矩阵, 其输入与输出的比特数相等, 也可称可逆算子. 量子逻辑门对输入比特进行确定的酉变换, 得到输出比特. Deutsch^[1]最早考虑用量子逻辑门构造量子计算机的问题, 他发现几乎所有的三比特量子逻辑门都是通用逻辑门. 通用逻辑门的含义是指, 通过该逻辑门的级联, 能够以任意精度逼近任意一个么正操作, 么正操作对应操作的数理解析, 逻辑门的级联将生成物理上的量子逻辑电路.

Deutsch 的结果随后得到发展, 最后 Deutsch 等^[2]和 Lloyd^[3]各自独立证明了几乎所有的二比特量子逻辑门都是通用的, 这里“几乎”是指, 二比特通用量子逻辑门的集合是所有二比特逻辑门的集合的一个稠密子集. 实验上通常用一些具体的量子逻辑门构造量子计算机. Barenco^[2]等人证明, 一个二比特的异或门与对一比特进行任意操作的门可构成一个通用量子门集.

相对而言, 单比特逻辑门在实验上比较容易实现, 现在多数实验方案都集中于制造量子异或门. 量子异或门和经典异或门非常相似, 它有两个输入比特: 控制比特和受控比特. 当控制比特处于微粒子上能级(激活态)时, 受控比特状态发生反转. 用记号 C_{12} 代表量子异或操作, 其中 1, 2 分别代表控制和受控比特, 则有 $|n_1\rangle_1 |n_2\rangle_2 \xrightarrow{C_{12}} |n_1\rangle_1 |n_1 \oplus n_2\rangle_2$, 其中 n_1, n_2 取值 0 或 1, \oplus 表示模 2 加(异或)运算.

迄今为止, 虽然世界上还没有真正意义的量子计算机. 但是, 世界主要经济发达国家都在制定战略性规划, 各国具有代表性的实验室正以巨大的热情投入人财物, 期望在新一代计算机的科学与技术上占据领导地位. 正因为实现量子计算机的技术困难重重, 而量子计算机的实现必将为信息科学与通信技术带来革命性的突破, 所以量子可逆逻辑电路的设计、优化与测试等方法的研究作为量子信息与量子计算理论的基础研究越来越受到理论研究者与应用研究者的关注, 也将笔者所在研究小组的研究重点从经典信息领域逐渐转向量子信息与计算领域.

量子可逆逻辑综合源于可逆计算机的研究.

20 世纪中叶, 人们发现计算机芯片的能耗导致芯片发热, 限制芯片集成度, 影响计算机的运行速度. Landauer^[4]发现, 芯片能耗主要源于计算中的不可逆操作. 因此降低能耗的关键是将不可逆操作变为可逆操作. Bennett^[5]对此有严格证明. 经典计算机本质上是一个通用图灵机, 是不可逆的, 但所有不可逆通用图灵机, 都对应一个可逆图灵机, 且两者的计算能力和计算效率完全相同. 由于量子逻辑门都是可逆的, 因此可以用可逆的设计方法综合量子逻辑电路. 量子电路理论上不丢失输入信息, 因此也不存在热耗散, 从而从理论上有效地解决了芯片的热耗问题.

Bennett 证明只要是可逆门构造的网络, 能量零损耗是可能的. 可逆逻辑已广泛应用在量子计算、低功耗 CMOS 电路、纳米技术、光计算、加密技术等许多领域, 因此可逆逻辑的研究将变得越来越重要.

最近 30 年, 人们提出了多种可逆量子门. 如 Feynman 提出的控制非门 (CNOT)^[6]、Toffoli 门^[7]、Fredkin 门^[8]等. 如何使用规定的量子门自动生成量子代价较小的量子电路, 即制造量子电路的成本较低, 通常认为量子代价是指使用量子门的数量, 最优的量子电路是指使用量子门的数量最少. Shende^[9]、Song^[10]等人提出了一些可逆逻辑综合的算法, Shende^[11]等人提出了一种 3 个输入变量的综合方法. Iwama^[12]等人给出了 CNOT 电路的综合规则, 提出 CNOT 门序列顺序变化的规则, 通过将实现么变换的相邻且相同的门消除, 最终实现可逆电路的化简. Miller^[13]应用谱函数实现近似最优的可逆电路化简. 然而目前人们还没有找到通用高效的算法, 特别对多个输入变量的量子电路, 这是量子电路中急需解决的重要问题之一. Shende 等人提出的穷举算法较慢, Miller^[14]、Iwama、Maslov^[15]给出了几个启发式算法, 有些还需要模板优化技术. Mishchenko^[16]等人提出使用 RM 综合可逆逻辑电路, Gupta^[17]给出了基于 RM 的启发式规则, 但这些算法缺乏普遍适用性, 且通常生成的电路不能达到最优.

本文首先给出生成 RM 展开式的通用算法, 并给出详细的证明, 然后根据化简 RM 展开式生成可逆逻辑电路的基本思想, 提出了在生成量子可逆逻辑电路的解空间树上, 层次遍历找到的第一个解必定是最优解, 又吸收深度搜索可以复用前面计算的结果, 在遍历的过程中借鉴模板优化技术, 构造限界函数, 快速剪去无解与非最优解的分枝, 优先探测 RM 中因子对应的量子门, 因此该算法的平均空间、

时间复杂度较小,实验中,能快速生成全部最优量子可逆逻辑电路。

2 量子可逆逻辑电路的基本概念

利用微观粒子状态表示的信息称为量子信息,量子信息的基本单位是量子比特(qubit),与经典信息不同,量子比特能够以叠加态的形式存在,任何量子比特均可由一个二元向量形式表示,形式如 $|\varphi\rangle=\alpha|0\rangle+\beta|1\rangle$,其中 α 和 β 为复数,满足归一化条件 $|\alpha|^2+|\beta|^2=1$ 。

量子逻辑门是处理量子信息的基本单元,量子逻辑门的级联构成量子电路,量子电路必须是可逆的,即量子信息的动态过程在复向量空间上必须保

持正交变换。在量子计算中,一个量子逻辑门对应一个么正变换,根据输入输出的对数,逻辑门可分为单量子比特门与多量子比特门。

定义 1. 通用 Toffoli 量子门记为 $TOF(C;t)$,其中输入变量集合 $In=\{x_1,x_2,\cdots,x_n\}$,控制端集合 $C=\{x_{i_1},x_{i_2},\cdots,x_{i_k}\},k\in\{1,2,\cdots,n-1\}$,受控端集合为 $t=\{x_j\}$,且满足 $C\cap t=\varnothing,C\cup t\subset In$ 。 $TOF(C;t)$ 将输出变量集合映射成: $\{x_1,x_2,\cdots,x_{j-1},x_j\oplus x_{i_1}x_{i_2}\cdots x_{i_k},x_{j+1},\cdots,x_n\}$ 。若 $\exists m\in\{1,2,\cdots,k\},x_{i_m}=0\rightarrow x_{i_1}x_{i_2}\cdots x_{i_k}=0$,受控端 x_j 输出为 $x_j\oplus x_{i_1}x_{i_2}\cdots x_{i_k}=x_j\oplus 0=x_j$;若 $\forall m\in\{1,2,\cdots,k\},x_{i_m}=1\rightarrow x_{i_1}x_{i_2}\cdots x_{i_k}=1$,受控端 x_j 的输出为 $x_j\oplus x_{i_1}x_{i_2}\cdots x_{i_k}=x_j\oplus 1=\bar{x}_j$ 。该 Toffoli 量子门通常表示为 $TOF(x_{i_1},x_{i_2},\cdots,x_{i_k};x_j)$,如图 1 所示。

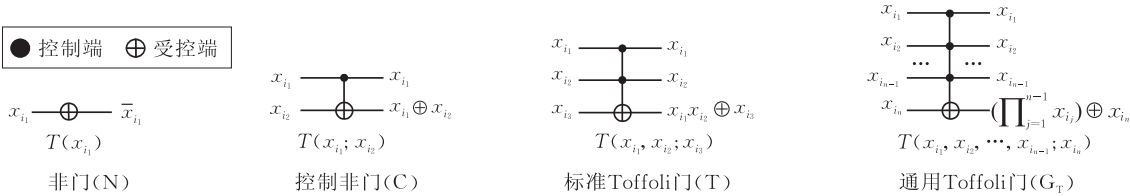


图 1 量子逻辑门

当 $k=0$ 时, $C=\varnothing$, $TOF(x_j)$ 为非门(NOT);
当 $k=1$ 时, $C=\{x_{i_1}\}$, $TOF(x_{i_1};x_j)$ 为控制非门(CNOT);
当 $k=2$ 时, $C=\{x_{i_1},x_{i_2}\}$, $TOF(x_{i_1},x_{i_2};x_j)$ 为标准 Toffoli 门。

这里只有非门为单量子比特门,其它均为多量子比特门。

定义 2. n 个输入与 n 个输出变量的布尔函数 $f(x_1,x_2,\cdots,x_n)=\{y_1,y_2,\cdots,y_n\}$,是可逆函数,当且仅当它是双射,即任意一个输入都对应着唯一的输出,如任意输入值 $(x_n\cdots x_2x_1)_2$ 对应唯一的输出值 $(y_n\cdots y_2y_1)_2$,反之亦然。其中, $x_i,y_i,1\leq i\leq n$ 分别表示第 i 个输入与输出变量, $(X)_2$ 表示将二进制数 X 转换为十进制的值,如 $(101)_2=5$ 。

可逆函数可用真值表表示,也可用整数集合 $\{0,1,\cdots,2^n-1\}$ 的置换表示。图 2 表示一个 3 个变量的量子可逆逻辑电路,用真值表表示见表 1,用置换表示为 $\sigma=\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 6 & 0 & 1 & 7 & 3 & 5 & 4 \end{pmatrix}$ 。

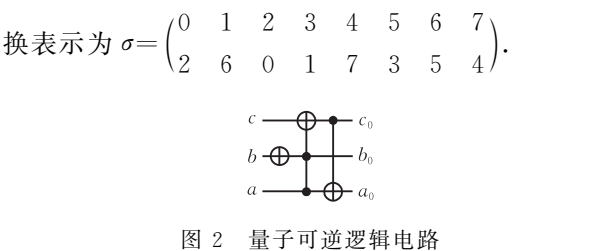


图 2 量子可逆逻辑电路

同样该量子可逆逻辑电路也可表示成
 $f(0)=2, f(1)=6, \cdots, f(7)=4$ (1)

表 1 图 2 量子电路的真值表								
输入				输出				
$(x_3x_2x_1)_2$				$(y_3y_2y_1)_2$				
$(c\ b\ a)_2$	c	b	a	$(c_o\ b_o\ a_o)_2$	c_o	b_o	a_o	
0	0	0	0	2	0	1	0	
1	0	0	1	6	1	1	0	
2	0	1	0	0	0	0	0	
3	0	1	1	1	0	0	1	
4	1	0	0	7	1	1	1	
5	1	0	1	3	0	1	1	
6	1	1	0	5	1	0	1	
7	1	1	1	4	1	0	0	

定义 3. 能用可逆函数描述的电路称为可逆逻辑电路。

如图 2 所示,可逆逻辑电路的特点是:(1) 输入线数与输出线数相等;(2) 没有扇出与扇入;(3) 没有反馈;(4) 电路分层级联,有时为保证电路可逆,需添加一些辅助位,即垃圾位。

定义 4. n 个输入与 n 个输出变量的布尔函数 $f(x_1,x_2,\cdots,x_n)=\{y_1,y_2,\cdots,y_n\}$ 的展开式:

$$f(x_n,x_{n-1},\cdots,x_1)=\bigoplus_{i=0}^{2^n-1}(d_iP_i) \quad (2)$$

称为正极性 Reed Muller 展开式(RM),即任意一个布尔函数均可用若干个输入变量的乘积的异或和的

形式表示. 其中 $P_i = \prod_{k=1}^n (x_k)^{i_k}$, i_k 表示 i 的二进制数的第 k 位数, 即 $i = (i_n i_{n-1} \cdots i_k \cdots i_1)_2$, i_k 是输入变量 x_k 的指数, 决定 x_k 是否出现, $d_i \in \{0, 1\}$, 决定 P_i 是否出现. 当 n 确定后, RM 展开式中唯一可变的是 d_i , 因此生成 RM 展开式的本质就是计算 d_i 的值. 其中, 设 $\forall i \in \{0, 1, \cdots, h\}$, $y_i \in \{0, 1\}$, 则 $\bigoplus_{i=0}^h y_i = y_0 \oplus y_1 \oplus \cdots \oplus y_h = (\sum_{i=0}^h y_i) \bmod 2$, \bmod 是取余运算.

以 3 个变量的量子可逆逻辑电路为例, RM 展开式为

$$f(x_1, x_2, x_3) = d_0 \oplus d_1 x_1 \oplus d_2 x_2 \oplus d_3 x_2 x_1 \oplus d_4 x_3 \oplus d_5 x_3 x_1 \oplus d_6 x_3 x_2 \oplus d_7 x_3 x_2 x_1.$$

如图 2 所示, 令 $a = x_1, b = x_2, c = x_3, a_0 = y_1, b_0 = y_2, c_0 = y_3$, 则量子电路用 RM 展开式表示为

$$\begin{aligned} a_0 &= ba \oplus c, \\ b_0 &= 1 \oplus b, \\ c_0 &= a \oplus ba \oplus c \end{aligned} \tag{3}$$

验证式(3)的 RM 展开式的正确性. 任选式(3)中的等式 q , 从表 1 中任选输入值 x 与对应的输出值 y , 即 $f(x) = y$, 分别代入到等式 q 的右边与左边, 如果等式 q 始终成立, 则式(3)的 RM 展开式为正确; 否则为不正确.

引理 1. 在可逆逻辑电路中, 任意两个相同且相邻的通用 Toffoli 门可以从电路中同时去除.

证明. 设量子电路中存在两个如图 1 所示的相同且相邻的通用 Toffoli 门, 分别为 $TOF_1(C; t)$, $TOF_2(C; t)$. TOF_1 的任意输入值 $In_{TOF_1} = \{x_1, x_2, \cdots, x_n\}$, 经过 TOF_1 后的输出值为 $\{x_1, x_2, \cdots, x_{j-1}, x_j \oplus (x_{i_1} x_{i_2} \cdots x_{i_k}), x_{j+1}, \cdots, x_n\}$, 并成为 TOF_2 的输入值, 再经过 TOF_2 后的输出值为 $\{x_1, x_2, \cdots, x_{j-1}, x_j \oplus (x_{i_1} x_{i_2} \cdots x_{i_k}) \oplus (x_{i_1} x_{i_2} \cdots x_{i_k}), x_{j+1}, \cdots, x_n\} = \{x_1, x_2, \cdots, x_{j-1}, x_j, x_{j+1}, \cdots, x_n\} = In_{TOF_1}$, 即任意输入值与其经过量子门 TOF_1, TOF_2 的输出值恒等, 因此这两个门可以从电路中同时去除.

证毕.

引理 2. 在可逆逻辑电路中, 任意两个相邻的通用 Toffoli 门, 分别为 $TOF_1(C_1; t_1), TOF_2(C_2; t_2)$. 若 $C_1 \cap t_2 = \emptyset, C_2 \cap t_1 = \emptyset$, 则 TOF_1 和 TOF_2 可以交换位置.

证明. 设 $C_1 = \{x_{i_1}, x_{i_2}, \cdots, x_{i_k}\}, t_1 = \{x_j\}, C_2 = \{x_{m_1}, x_{m_2}, \cdots, x_{m_p}\}, t_2 = \{x_h\}$, 设 $j \leq h$. 则任意输入值 $In = \{x_1, x_2, \cdots, x_n\}$ 经过 TOF_1 后的输出值为 $\{x_1, x_2, \cdots, x_{j-1}, x_j \oplus (x_{i_1} x_{i_2} \cdots x_{i_k}), x_{j+1}, \cdots,$

$x_n\}$, 它也是 TOF_2 的输入值, 因为存在 $C_1 \cap t_2 = \emptyset, C_2 \cap t_1 = \emptyset$, 所以唯一变化的 $x_j \notin C_2$, 即 TOF_1 对 TOF_2 的控制端的信息没有影响, 则经过 TOF_2 的输出值为

如果 $j < h$, 则 $\{x_1, x_2, \cdots, x_{j-1}, x_j \oplus (x_{i_1} x_{i_2} \cdots x_{i_k}), x_{j+1}, \cdots, x_{h-1}, x_h \oplus (x_{m_1} x_{m_2} \cdots x_{m_p}), x_{h+1}, \cdots, x_n\}$;
如果 $j = h$, 则 $\{x_1, x_2, \cdots, x_{j-1}, x_j \oplus (x_{i_1} x_{i_2} \cdots x_{i_k}) \oplus (x_{m_1} x_{m_2} \cdots x_{m_p}), x_{j+1}, \cdots, x_n\}$.

同理可得, In 经过 TOF_2 和 TOF_1 后的输出值与上式相同, 即 TOF_1 和 TOF_2 与 TOF_2 和 TOF_1 的功能相同, 所以在可逆逻辑电路中, TOF_1 和 TOF_2 可以交换位置. 证毕.

引理 3. 若可逆逻辑电路有 n 条线, 则可使用图 1 所示的通用 Toffoli 量子门共有 $n2^{n-1}$ 种.

证明. 根据定义 1 可知, 量子门的受控端只有 1 个, 可从 n 条线中任选, 有 n 种选择; 控制端则在余下的 $n-1$ 条线中任选 i 条线, 有 C_{n-1}^i 种选择, 则控制端有 i 条线的通用 Toffoli 门共有 $N_{GT}(i) = nC_{n-1}^i$ 种, 已知 $0 \leq i \leq n-1$, 所以全部 $NCTG_T$ 量子门共有 $N_{GT} = \sum_{i=0}^{n-1} N_{GT}(i) = \sum_{i=0}^{n-1} (nC_{n-1}^i) = n2^{n-1}$ 种.

证毕.

定理 1. 在可逆逻辑电路 $TOF_1, TOF_2, \cdots, TOF_n$ 中, 如果存在相同的量子门 $TOF_m, TOF_k, 1 \leq m < k \leq n$, 且任意 $TOF_i, m < i < k$ 都满足: $C_i \cap t_k = \emptyset, C_k \cap t_i = \emptyset$, 则 TOF_m 和 TOF_k 可同时从电路中去除. 其中 $C_j, t_j, 1 \leq j \leq n$ 分别表示量子门 TOF_j 的控制端集合与受控端集合.

证明. 设可逆逻辑电路中包含 $TOF_1, TOF_2, \cdots, TOF_{m-1}, TOF_m, TOF_{m+1}, \cdots, TOF_{k-1}, TOF_k, TOF_{k+1}, \cdots, TOF_n$, 因为 $C_i \cap t_k = \emptyset, C_k \cap t_i = \emptyset, 1 \leq m < i < k \leq n$, 根据引理 2, TOF_k 可分别与 $TOF_{k-1}, TOF_{k-2}, \cdots, TOF_{m+1}$ 交换位置, 得到新的电路序列为 $TOF_1, TOF_2, \cdots, TOF_{m-1}, TOF_m, TOF_k, TOF_{m+1}, \cdots, TOF_{k-1}, TOF_{k+1}, \cdots, TOF_n$. 由条件可知 $TOF_m = TOF_k$, 根据引理 1, TOF_m 与 TOF_k 可以从电路中同时去除, 可得新的可逆逻辑电路序列为 $TOF_1, TOF_2, \cdots, TOF_{m-1}, TOF_{m+1}, \cdots, TOF_{k-1}, TOF_{k+1}, \cdots, TOF_n$. 证毕.

3 量子可逆逻辑电路综合方法及其比较

量子可逆逻辑综合是以较小的量子代价自动构造所求的量子可逆逻辑电路, 具有很强的实际应用价值.

3.1 量子可逆逻辑电路主要构造方法

(1)真值表法^[15]. 根据如表 1 所示的真值表自动构造可逆逻辑电路. f 为如式(1)的可逆函数. 有 3 种方法:①前向合成. 先按照 i 的升序, 对于每一个 i 寻找 j , 使得 $f(j)=i$, 再通过选取 Toffoli 门, 将 j 转换为 i , 不断重复此过程, 直至满足 $\forall i, f(i)=i$. 将选择的门顺序排列; ②后向合成. 先按照 i 的升序, 对于每个 i , 通过选取 Toffoli 门, 将 $f(i)$ 转换为 i , 不断重复此过程, 直至满足 $\forall i, f(i)=i$. 将选择的门逆序排列; ③双向合成. 综合前面两种方法. 先按照 i 的升序, 对于每个 i , 寻找 j , 使得 $f(j)=i$, 若 $Ham(i, f(i)) \leq Ham(i, j)$, 使用后向合成, 否则使用前向合成. Ham 表示汉明距离. 设 $i = (i_n i_{n-1} \cdots i_1)_2$, $j = (j_n j_{n-1} \cdots j_1)_2$, i 与 j 之间的汉明距离为

$$Ham(i, j) = \sum_{k=1}^n sg(i_k, j_k), \text{ 其中函数}$$

$$sg(i, j) = \begin{cases} 1, & i \neq j \\ 0, & i = j \end{cases}.$$

(2)专用构造方法. 本方法根据一些特定功能的可逆逻辑电路的特点, 用专用构造算法, 快速生成电路. 如用 Bitonic^[18] 方法可快速构造大规模的量子排序电路, 然而这些算法不具有通用性, 但性能很好. 例如构造可逆排序电路, 如果用其它方法, 则算法复杂性较高, 且构造的电路没有规律, 很难由此直接构造更大的排序电路.

(3)RM 法^[17]. 执行下列步骤:①根据可逆逻辑电路的功能构造可逆函数; ②通过可逆函数构造 RM 展开式; ③通过对 RM 展开式逐步化简, 生成可逆逻辑电路. 这是本文使用的方法.

3.2 基于 RM 的可逆逻辑电路的构造方法

该方法共分 3 步:

(1)将可逆逻辑电路的功能用表 2 所示的真值表描述其可逆函数. 设有 n 个输入与输出变量, 分别为 $x_n, x_{n-1}, \cdots, x_1$ 与 $y_n, y_{n-1}, \cdots, y_1$, $x_j \in \{0, 1\}$, $y_j \in \{0, 1\}$, $1 \leq j \leq n$, 则全体输入值分别为 $\{0, 1, \cdots, 2^n - 1\}$, 与之相对应的输出值分别为 $\{Y_0, Y_1, \cdots, Y_{2^n-1}\}$, 其中 $Y_i = (y_{i,n} y_{i,n-1} \cdots y_{i,1})_2$, $0 \leq i \leq 2^n - 1$; 而当输入值 $(x_n x_{n-1} \cdots x_1)_2$ 为 i 时, 则输出变量 y_j 的值记为 $y_{i,j}$.

表 2 n 个输入、输出变量的通用真值表

输入		输出
$i = (x_n \cdots x_2 x_1)_2$	$x_n \cdots x_2 x_1$	$y_n \cdots y_j \cdots y_1$
0	0 \cdots 0 0	$y_{0,n} \cdots y_{0,j} \cdots y_{0,1}$
1	0 \cdots 0 1	$y_{1,n} \cdots y_{1,j} \cdots y_{1,1}$
...
$2^n - 2$	1 \cdots 1 0	$y_{2^n-2,n} \cdots y_{2^n-2,j} \cdots y_{2^n-2,1}$
$2^n - 1$	1 \cdots 1 1	$y_{2^n-1,n} \cdots y_{2^n-1,j} \cdots y_{2^n-1,1}$

(2)生成 y_j 的 RM 展开式, 即表 2 中 $y_{i,j}$ 的通用表达式, 其中 $0 \leq i \leq 2^n - 1$, $1 \leq j \leq n$, 这里给出递归算法.

算法 1. y_j 的 RM 展开式生成算法 $GRM(i, j)$.

输入: i 是可逆函数的输入, $j \in \{1, 2, \cdots, n\}$

输出: RM 展开式

1. if $i=0$ then

2. return $y_{0,j}$

3. else

4. return $GRM(i-1, j) \oplus (\bigoplus_{h=0}^i (eq(h|i, i) y_{h,j})) P_i$ (4)

计算 y_j 的 RM 展开式为 $GRM(2^n - 1, j)$, 即此展开式分别输入 $\{0, 1, \cdots, 2^n - 1\}$ 时, 对应的输出值分别为 $\{y_{0,j}, y_{1,j}, \cdots, y_{2^n-1,j}\}$, 其中

$$GRM(2^n - 1, j) |_{(x_n x_{n-1} \cdots x_1)_2 = k} = GRM(2^n - 1, j) |_{k = y_{k,j}}.$$

式(4)中“ $|$ ”为位或运算, $eq(h|i, i)$ 的含义是若 h 的二进制的 1 全部包含在 i 的二进制中, 为 1, 否则为 0;

算法第 4 步中 $\bigoplus_{h=0}^i (eq(h|i, i) y_{h,j})$ 的含义是选择全部 $y_{h,j}$, $0 \leq h \leq i$ 且 $eq(h|i, i) = 1$, 求它们的异或和, 若为 1, 式(4)返回为 $GRM(i-1, j) \oplus P_i$, 否则为

$$GRM(i-1, j). \text{ 其中函数 } eq(i, j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}.$$

(3)基于 RM 构造量子可逆逻辑电路的通用方法. 设可逆逻辑电路有 n 个输入、输出变量分别为 x_1, x_2, \cdots, x_n 与 y_1, y_2, \cdots, y_n , 分别有 n 个 RM 展开式: $y_j = d_0 \oplus d_1 x_1 \oplus d_2 x_2 \oplus d_3 x_2 x_1 \oplus \cdots \oplus d_{2^n-1} x_n \cdots x_2 x_1$, 其中 $1 \leq j \leq n$, $d_i \in \{0, 1\}$. 依次试探量子门 $TOF(C_k; x_k)$, 根据引理 3 可知, 共有 $n2^{n-1}$ 种量子门, 则将 RM 展开式中全部 n 个等式中所有的 x_k 用 $x_k \oplus C_k$ 代替, 并化简, 不断重复此过程, 直至 RM 展开式变为恒等式. 即 $\forall i \in \{1, 2, \cdots, n\}$, $y_i = x_i$, 将化简过程中使用的量子逻辑门顺序排列, 生成所求量子可逆逻辑电路.

3.3 用归纳法证明构造 RM 展开式的 GRM 算法

求证: $GRM(i, j) |_{k \in \{0, 1, \cdots, i\}} = y_{k,j}$, $0 \leq i \leq 2^n - 1$, 如表 2 所示, 即输入值为 k 时, RM 展开式 $GRM(i, j)$ 的值为 $y_{k,j}$.

证明.

(1)归纳基础. 当 $i=0$ 时, 由 GRM 算法可得 $GRM(0, j) = y_{0,j}$, 因此输入值为 $k \in \{0\}$ 即 $k=0$ 时, $GRM(0, j)$ 的值等于 $y_{0,j}$, 所以, $GRM(0, j) |_{k \in \{0\}} = y_{0,j} = y_{k,j}$ 成立.

(2)归纳假设. 当输入值 $i = m - 1$ 时, $1 \leq m \leq$

$2^n - 1, GRM(m-1, j) \mid_{k \in \{0, 1, \dots, m-1\}} = y_{k,j}$.

(3) 归纳步骤. 当输入值 $i = m$ 时, 由算法 1 可得 $GRM(m, j) = GRM(m-1, j) \oplus (\bigoplus_{h=0}^m (eq(h \mid m, m) y_{h,j})) P_m$. 当输入值为 $k \in \{0, 1, \dots, m-1\}$ 时, 将 k 的二进制值代入到 P_m , 见定义 4. 由 $0 \leq k < m$ 可得 $k \mid m \neq k$, k 的二进制数中不能包括 m 的二进制数中所有的 1, 因此 k 输入到 P_m 时, P_m 至少有一个输入变量为 0, 可得 $P_m \mid_{k \in \{0, 1, \dots, m-1\}} = 0$. 因此:

$$\begin{aligned} &GRM(m, j) \mid_{k \in \{0, 1, \dots, m-1\}} \\ &= GRM(m-1, j) \mid_k \oplus ((\bigoplus_{h=0}^m (eq(h \mid m, m) y_{h,j})) P_m) \mid_k \\ &= GRM(m-1, j) \mid_k \oplus ((\bigoplus_{h=0}^m (eq(h \mid m, m) y_{h,j})) P_m \mid_k) \\ &= GRM(m-1, j) \mid_k \oplus 0 \\ &= GRM(m-1, j) \mid_k \xrightarrow{\text{假设}} GRM(m, j) \mid_{k \in \{0, 1, \dots, m-1\}} = y_{k,j} \end{aligned} \tag{5}$$

当输入值为 $k \in \{m\}$ 即 $k = m$ 时, $P_m \mid_m = 1$, 见定义 4, 条件 $h \mid m = m$, 是指 h 的二进制数中的 1 包含在 m 的二进制数中, 若 m 中有 t 个 1, 满足条件的 h 共有 $u = 2^t$ 个, 分别 $\{h_0, h_1, \dots, h_{u-1}\}$, 得 $h_0 = 0, h_{u-1} = i$. 因此

$$\begin{aligned} &GRM(m, j) \mid_m \\ &= (GRM(m-1, j) \oplus (\bigoplus_{h=0}^m (eq(h \mid m, m) y_{h,j})) P_m) \mid_m \\ &= (GRM(m-1, j)) \mid_m \oplus (\bigoplus_{h=0}^m (eq(h \mid m, m) y_{h,j})) P_m \mid_m \\ &\xrightarrow{P_m \mid_m = 1} GRM(m, j) \mid_m \\ &= (GRM(m-1, j)) \mid_m \oplus (\bigoplus_{h=0}^m (eq(h \mid m, m) y_{h,j})) \\ &= (GRM(m-1, j)) \mid_m \oplus (\bigoplus_{h=0}^{m-1} (eq(h \mid m, m) y_{h,j}) \oplus \end{aligned}$$

$$\begin{aligned} &eq(m \mid m, m) y_{m,j} \xrightarrow{\text{假设}} GRM(m, j) \mid_m \\ &= (GRM(m-1, j)) \mid_m \oplus \\ &\quad \bigoplus_{h=0}^{m-1} (eq(h \mid m, m) GRM(m-1, j) \mid_h) \oplus y_{m,j} \\ &= \bigoplus_{h=0}^m (eq(h \mid m, m) GRM(m-1, j) \mid_h) \oplus y_{m,j} \end{aligned} \tag{6}$$

根据算法可知 $GRM(m-1, j)$ 表达式中是由 $y_{0,j}$ 与若干个 P_r 的异或和组成, 其中 $r \in \{1, 2, \dots, m-1\}$, $\bigoplus_{h=0}^m (eq(h \mid m, m) y_{0,j} \mid_h) = 0$, 因为满足条件的 h 的个数为偶数, 而偶数个常数 $y_{0,j}$ 的异或之和必为 0; $\bigoplus_{h=0}^m (eq(h \mid m, m) P_r \mid_h) = 0$, 因为若 P_r 不是 P_m 的因子, 则 $P_r \mid_h = 0$, 若 P_r 是 P_m 的因子, 设 P_m 是 t 个输入变量相乘, 而 P_r 是 P_m 的 s 个输入变量相乘, $0 \leq s < t$, 则 P_r 输入所有满足条件的 h 的值有 2^{t-s} 个 1 与 $2^t - 2^{t-s}$ 个 0, 将这些偶数个 1 与偶数个 0 异或的和为 0, 所以由式 (6) 可得 $GRM(m, j) \mid_m = \bigoplus_{h=0}^m (eq(h \mid m, m) GRM(m-1, j) \mid_h) \oplus y_{m,j} = 0 \oplus y_{m,j} = y_{m,j}$, 即 $GRM(m, j) \mid_{k \in \{m\}} = y_{k,j}$, 再与式 (5) 合并可得 $GRM(m, j) \mid_{k \in \{0, 1, \dots, m\}} = y_{k,j}$, 即当输入为 $i = m$ 时, 结论也成立. 证毕.

3.4 构造 RM 展开式的计算方法

下面根据式 (1) 可逆函数, 构造 RM 展开式, 有两种方法:

(1) 直接使用 GRM 的递归算法. 计算过程详见表 3. 表中最后一行的 RM 展开式与上一行没有变化, 因为根据定义 3 可知: $\bigoplus_{i=0}^{2^n-1} y_{i,j} = (\sum_{i=0}^{2^n-1} y_{i,j}) \bmod 2 = 2^{n-1} \bmod 2 = 0$, 如表 3 中, 当 $n = 3$ 时, $R_7 = R_6 \oplus 0 = R_6$.

表 3 Reed Muller 展开式的生成过程表

输入			输出			生成 y_i 的 Reed Muller 展开式			
i	cba	P_i	f	$y_{i,3} y_{i,2} y_{i,1}$	R_i	$R_{i-1} \oplus (\bigoplus_{h=0}^i (eq(h \mid i, i) y_{h,j})) P_i$	$j=3$	$j=2$	$j=1$
0	0 0 0	$c^0 b^0 a^0$	1	2 0 1 0	R_0	$y_{0,j}$	0	1	0
1	0 0 1	$c^0 b^0 a^1$	a	6 1 1 0	R_1	$R_0 \oplus (y_{0,j} \oplus y_{1,j}) P_1$	a	1	0
2	0 1 0	$c^0 b^1 a^0$	b	0 0 0 0	R_2	$R_1 \oplus (y_{0,j} \oplus y_{2,j}) P_2$	a	$b \oplus 1$	0
3	0 1 1	$c^0 b^1 a^1$	ba	1 0 0 1	R_3	$R_2 \oplus (y_{0,j} \oplus y_{1,j} \oplus y_{2,j} \oplus y_{3,j}) P_3$	$ba \oplus a$	$b \oplus 1$	ba
4	1 0 0	$c^1 b^0 a^0$	c	7 1 1 1	R_4	$R_3 \oplus (y_{0,j} \oplus y_{4,j}) P_4$	$c \oplus ba \oplus a$	$b \oplus 1$	$ba \oplus c$
5	1 0 1	$c^1 b^0 a^1$	ca	3 0 1 1	R_5	$R_4 \oplus (y_{0,j} \oplus y_{1,j} \oplus y_{4,j} \oplus y_{5,j}) P_5$	$c \oplus ba \oplus a$	$b \oplus 1$	$ba \oplus c$
6	1 1 0	$c^1 b^1 a^0$	cb	5 1 0 1	R_6	$R_5 \oplus (y_{0,j} \oplus y_{2,j} \oplus y_{4,j} \oplus y_{6,j}) P_6$	$c \oplus ba \oplus a$	$b \oplus 1$	$ba \oplus c$
7	1 1 1	$c^1 b^1 a^1$	cba	4 1 0 0	R_7	$R_6 \oplus (\bigoplus_{k=0}^i y_{k,j} P_r) = R_6$	$c \oplus ba \oplus a$	$b \oplus 1$	$ba \oplus c$

(2) 将 GRM 算法转变为矩阵计算. 执行下列步骤:

1. 根据式 (1) 可逆函数的输出值的二进制数构造矩阵

F , 用递归定义 M 方阵.

$$M^0 = [1], M^n = \begin{bmatrix} M^{n-1} & \mathbf{0} \\ \mathbf{M}^{n-1} & M^{n-1} \end{bmatrix} \tag{7}$$

2. 生成 RM 展开式的向量计算公式为

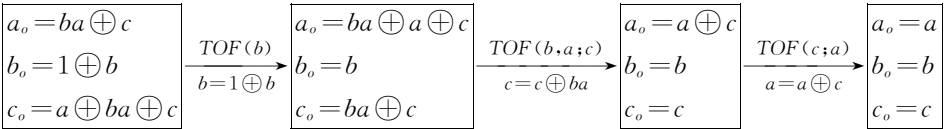
$$(\mathbf{R}^T \cdot \mathbf{P}) \bmod 2 \tag{8}$$

其中 $\mathbf{R} = \mathbf{M}^n \mathbf{F}$, $\mathbf{P} = (P_0, P_1, \dots, P_{2^n-1})^T$, P_i 的定义参见式(2), $0 \leq i \leq 2^n - 1$.

$$\mathbf{R} = \mathbf{M}^n \mathbf{F}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} c \\ b \\ a \end{pmatrix} = (\mathbf{R}^T \cdot \mathbf{P}) \bmod 2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}^T \begin{pmatrix} 1 \\ a \\ b \\ ba \\ c \\ ca \\ cb \\ cba \end{pmatrix} \bmod 2$$



将上面的量子门顺序排列: $TOF(b)$ $TOF(b, a; c)$ $TOF(c; a)$, 生成如图 2 所示的量子可逆逻辑电路.

3.6 综合量子可逆电路方法的比较

综合算法主要有构造电路与优化电路两步操作.

(1) 量子可逆逻辑电路常用构造方法比较, 如表 4 所示.

表 4 本文构造方法与常用方法的比较			
构造方法	构造速度	有无优化	通用性
真值表法	快	无	强
专用方法	很快	无	无
RM 方法	本文方法很快	已优化	强

(2) 量子可逆逻辑电路常用优化方法比较, 如表 5 所示.

表 5 本文优化方法与常用方法的比较				
优化方法	优化层次	通用性	易用性	速度
模板	较优	弱, 要符合模板要求	难, 技巧较多	一般较慢
CNOT 规则	较优	弱, 要符合 CNOT 规则	较繁, 规则较多	一般较慢
RM 方法	可达到最优	较强	较易	本文方法很快

从上面两张表可以得出, 使用 RM 方法构造电路的同时还能优化电路, 具有许多优点. 因此为降低制造量子可逆逻辑电路的成本, 增强量子电路综合

$$= \begin{pmatrix} a+ba+c \\ 1+b \\ ba+c \end{pmatrix} \bmod 2 = \begin{pmatrix} a \oplus ba \oplus c \\ 1 \oplus b \\ ba \oplus c \end{pmatrix}.$$

根据 GRM 算法的特点, 为提高算法性能, 可用非递归方法快速构造 \mathbf{M} 方阵.

以上两种 RM 构造方法本质相同, 结果都为式(3). 但 GRM 算法计算每个可逆函数都使用递归, 而矩阵的方法只要首次构造 \mathbf{M}^n 方阵, 然后将任意 n 变量的可逆函数代入式(8), 并生成 RM 展开式所构成的向量, 避免使用递归. 又因为 \mathbf{M}^n 是稀疏方阵, 笔者设计了高效算法, 避免大量无效计算, 提高了算法整体性能.

3.5 基于 RM 构造量子可逆逻辑电路

根据 3.2 节的第 3 部分提出的基于 RM 构造量子可逆逻辑电路的方法, 生成量子可逆逻辑电路的过程如下:

算法的通用性与效率, 本文提出了基于 RM 的量子可逆逻辑电路综合的快速算法.

4 基于 RM 的量子可逆逻辑电路综合的快速算法

基于 RM 的量子可逆逻辑电路综合的本质是对 RM 展开式进行化简, 可以将 RM 展开式的化简过程表示为一棵如图 3 所示的解空间树. 综合可逆

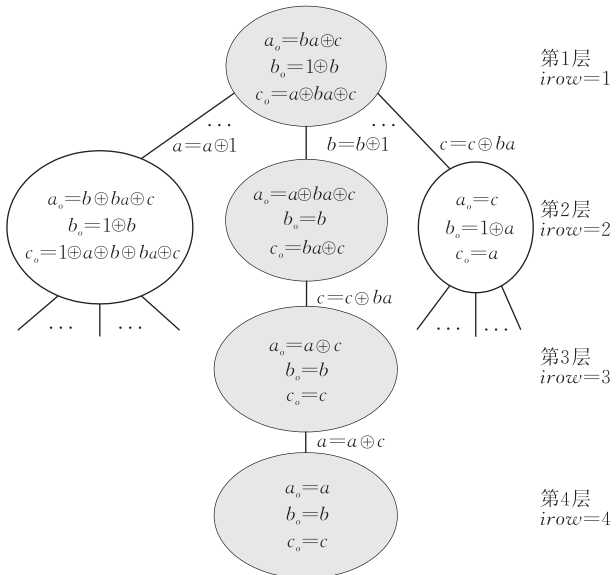


图 3 高度为 4 的量子电路解空间树

逻辑电路的方法主要有 3 种,分别为:(1)启发式规则;(2)回溯法;(3)分枝定界法,本文算法是在这些方法的基础上研制而成.其中“解”是指满足要求的量子可逆逻辑电路(表 6 所示).

表 6 3 种算法的特点

方法	内容	优点	缺点
启发式规则	利用启发式规则,以较高概率快速找到次优解.	大多数情况,能够快速找到次优解.	(1)一般不能获得最优解. (2)启发式规则不具有普遍适用性. (3)文献[17]因采用优先排序队列,常因入队元素太多而内存溢出.
回溯法	穷举可能的解,最后通过比较,可得到最优解.	通常能得到最优解.能复用前面的计算结果.	(1)通用性不强.必须先估计电路的最大长度,而这没有确定的算法. (2)速度较慢,遍历出可能的全部解,最后,通过比较,方能得到最优解.
分枝定界法	逐层遍历,依次试探可能的解,第一个解为最优.	逐层遍历,第一个解必定是最优解,提高效率.	(1)速度较慢.如果最优电路较长,需逐层试探电路较短的情况. (2)如使用队列,内存消耗较大.否则因无法复用计算,速度较慢.

设最优电路的长度为 N ,可选用的量子门数为 M ,生成的量子电路解空间树是高度为 $N+1$ 的满 M 叉树,最坏情况下访问节点总数为 $M^0+M^1+M^2+\cdots+M^N=(M^{N+1}-1)/(M-1)$,若排除相邻且相同的量子门,解空间树变为准满 $M-1$ 叉树,例外的是根节点有 M 个子节点,则访问节点数为 $1+(M-1)^0+(M-1)^1+\cdots+(M-1)^N=((M-1)^{N+1}+M-3)/(M-2)$. 根据引理 3 可得,当电路有 n 条线,全部通用 Toffoli 门有 $M=n2^{n-1}$ 种. 因此当 n 增大时,最坏情况下访问节点总数成指数量级增加,算法若不优化,当 n 较大时,如 $n\geq 3$,量子电路就很难综合.

笔者的方法是用式(8)自动构造 RM 展开式,在生成量子可逆逻辑电路的解空间树上,采用总体层次遍历,局部深度搜索,借鉴模板优化技术,构造限界函数快速删除无解与非最优解的分枝,优先探测 RM 中的因子,以极高的效率生成最优量子可逆逻辑电路. 其中,总体层次遍历的思想来自分枝定界法,局部深度搜索的思想来自回溯法,优先探测 RM 中的因子的思想来自启发式规则. 因此本方法充分吸收了前面三种方法的优点,并巧妙地去除了它们的缺点,其算法的平均空间、时间复杂度都很小.

算法先执行层次遍历算法 WHH,从不需要量子门(图 3 的第 1 层)开始寻找最优解,若找到,立即返回,否则,再从需要 1 个量子门(图 3 的第 2 层)开始寻找解,依次类推,找到的第一个解必为最优解,无需继续找其它解,算法如下:

算法中使用的全局变量有: n 表示量子电路的线数,MAX 为量子电路的最大长度,假设 MAX 足够大,确保算法有解,mgate[MAX]存放量子电路中各层使用的量子门序号,数组 RM[MAX+1][$n\times 2^n$]存放解空间树中每层的 RM 展开式. 其中表示该电路的 RM 展开式最多需 $n\times 2^n$ 个整数,因为根据定义 4 可知,任一条线的 RM 展开式最多有 2^n 个数

据项进行异或和,每个数据项可用一个 n 位的二进制数对应值域为 $0\sim 2^n-1$ 的整数表示,即描述一条线的 RM 展开式最多需 2^n 个整数,因此描述 n 条线的量子电路的 RM 展开式最多需 $n\times 2^n$ 个整数.

图 3 所示的解空间树对应的全局变量值为 $n=3$;MAX=8;mgate[i], $i\in\{1,2,\cdots,8\}$,表示解空间树的某条路径上第 $i,i+1$ 层之间使用的量子门;RM[j], $j\in\{1,2,\cdots,9\}$,可存放第 j 层的某一节点对应的 RM 展开式.

根据引理 3 可得量子门总数为 $n\times 2^{n-1}$,数组 mgateidx[$n\times 2^{n-1}$]存放可使用的全部量子门的序号.

算法 2. 基于 RM 的量子可逆逻辑电路综合的快速算法 WHH(f).

- 输入:所求量子可逆逻辑电路的可逆函数 f .
- 输出:所求量子可逆逻辑电路的量子门序列.
1. 应用式(8)的方法,将可逆函数 f ,自动生成 RM 展开式,存入 RM[1]中

2. if RM[1]为恒等式 then return 空序列

3. 数组 mgateidx 中存放全部量子门的序号,且与 RM 因子相对应的量子门序号排在数组的前列 (9)

4. $i_{high}=2$

5. while not DFS($i_{high},2$) do {

6. $i_{high}=i_{high}+1$ }

7. return mgate[1],mgate[2], \cdots ,mgate[$i_{high}-1$]

算法 2 是本文量子电路综合算法的主程序,是根据电路的功能自动高效地生成最优的量子可逆逻辑电路. 第 1 步是根据可逆电路的功能对应的可逆函数 f 生成 RM 展开式;第 2 步判断 RM 展开式是否为恒等式,如果是,则电路中没有量子门,返回空序列;否则,第 3~7 步是从解空间树的第二层开始,调用如下算法 DFS 逐层寻找解,直至找到解为止,从而快速生成最优的量子电路序列.

假设最优量子电路的门数为 i ,则可调用深度搜索算法 DFS,快速搜索深度为 $i+1$ 的解空间树,

且只要判断第 $i+1$ 层(叶子层)是否有解;深度搜索的优点是不使用队列,在量子电路数一定的情况下,所需存储空间仅与树的高度成正比,还可复用上层计算的结果,快速化简本层 RM 展开式。

算法 3. 深度搜索解空间树的算法 $DFS(ihigh, irow)$.

输入: $ihigh$ 为解空间树的高度, $irow$ 为当前访问的层号

输出: 返回在第 $ihigh$ 层(叶子点)是否找到解,若找到,则可以从全局数组 $mgate$ 中读取量子门序列

```

1. if  $irow = ihigh + 1$  then {
2.   if  $RM[irow - 1]$  为恒等式 then
3.     return true
4.   else
5.     return false}
6. else{
7.   for  $i = 1$  to  $n \times 2^{n-1}$  do {
8.      $mgate[irow - 1].value = mgateidx[i]$ 
9.      $mgate[irow - 1].index = i$ 
10.    if not  $BBF(irow - 1)$  then continue
11.    将  $mgate[irow - 1]$  代入  $RM[irow - 1]$ , 经过化简, 生成新的 RM 展开式, 存入  $RM[irow]$ 
12.    if  $DFS(f, ihigh, irow + 1)$  then return true}
13.   return false}

```

递归算法 $DFS(ihigh, irow)$ 是从第 $irow$ 层开始, 在高度为 $ihigh$ 的解空间树上深度搜索解, 第 1 步是判断当前第 $irow$ 层是否超过树的高度, 若超过, 则第 $irow - 1$ 层为叶子层, 否则为非叶子层; 若为叶子层, 则第 2~5 步判断当前电路是否为解, 因为算法 WHH 是逐层寻找解, 因此非叶子层中一定没有解, 只有叶子层才有可能有解; 第 7~13 步是在已有电路后面, 依次试探所有可能的一个量子门, 生成 RM 展开式, 并递归调用本算法, 进入下一层试探; 若第 2 步成立, 即找到第一个解, 返回真; 若叶子层中不存在解, 则返回假; 第 10 步 $BBF(irow - 1)$ 判断量子门 $mgate[irow - 1]$ 能否追加到当前电路的后面, 若不能, 则从解空间树中剪去此量子门为根的子树, 缩小搜索空间, 提高算法整体性能, 因此设计限界函数 BBF 非常重要。

设量子电路中量子门序列为 $g_1 g_2 \cdots g_{h-1}$, 判断能否追加量子门 g_h , 若出现如下两种情况, 禁止追加。

(1) 如果 g_h 能够根据定理 1 与前面 $h-1$ 个量子门中某个量子门化简, 此为可化简情况。因为笔者是寻求最优解, 而最优解内部是不能化简的。这与模板的思想相反, 模板是用来化简电路, 而本方法是选

取不能化简的量子门, 可利用模板技术快速判断大多数可化简的情况, 但判断算法的复杂度不能太高, 否则虽然访问节点数减少了, 但因判断算法的复杂性增强, 算法整体性能反而变差, 为此设计简单高效的算法, 判断大多数可化简情况, 详见如下算法 BBF 的式(10)。

(2) 如果 g_h 能够根据引理 2 移动到 g_j 位置, 且 g_j 的序号大于 g_h 的序号, 此为无解情况。因为本算法在每个节点上是按照量子门的序号顺序试探, 因此移动后的量子门序列 $g_1 g_2 \cdots g_{j-1} g_h g_j \cdots g_{h-1}$ 在前面一定试探过, 且没有获得解, 否则算法会提前返回, 而不可能运行到当将状态。详见如下算法 BBF 的式(11)。

此判断方法每次最多比较 $h-1$ 次, 因此速度很快; 而使用模板化简电路时, 因为化简的可能性很多, 需要依次试探, 因此运行速度较慢。 BBF 算法如下:

算法 4. 算法 $BBF(h)$, 判断能否在量子门序列 $gate[1], gate[2], \cdots, gate[h-1]$ 的后面追加量子门 $gate[h]$ 。

输入: 量子门 $gate[h]$ 的数组下标 h 。

输出: 如果能追加量子门 $gate[h]$, 则返回 true, 否则返回 false

```

1. for  $i = h - 1$  downto 1 do {
2.   if  $gate[i].t$  in  $gate[h].C$  or  $gate[h].t$  in  $gate[i].C$  then
3.     return true
4.   else if  $gate[h] = gate[i]$  then
5.     return false
6.   else if  $gate[h].index < gate[i].index$  then

```

(11)

```

7.     return false}
8. return true

```

算法 4 将量子门 $gate[h]$ 从后至前依次与量子门 $gate[i]$ 比较; 第 2 步是判断是否符合引理 2 的位置交换条件, 若不能交换, 则放弃化简, 认为可追加; 若能交换, 第 4 步判断这两个门是否相同, 若相同, 则根据定理 1 可知, 这两个门可从电路中去掉, 为可化简的情况, 不可追加; 第 6 步判断追加量子门 $gate[h]$ 的序号是否小于当前量子门 $gate[i]$ 的序号, 若小于, 为已探测过的无解情况, 不可追加; 第 8 步, 当前面都不能确定能否追加, 则认为可追加。

本文的程序使用 C++ 编程, 应用多种编程技术, 如将递归函数转变为非递归函数, 缓存常用的中间计算结果, 减少重复计算相同数据, 利用位操作,

快速化简 RM 展开式和相关数据,增强内存的复用率,避免反复申请与释放内存,借鉴模板技术,快速删除无解与非最优解,优先考虑 RM 的因子,设计布尔稀疏矩阵快速算法等. 如综合式(1)的可逆函数,若应用 WHH 算法,仅需 244 步,而在高度为 9 的解空间树上深度搜索,却要 7180 步,显然本文的算法可以极大提高运行效率.

5 实验结果与分析

笔者采用国际同行认可的 3 变量可逆函数测试标准的实验,共生成 $2^3!=8!=40320$ 个可逆逻辑电路. 本实验的目的是用较短的时间找到全部量子代价尽可能小的电路,因此主要有两个评价指标:(1)运行时间要短;(2)平均使用量子门的数量要小. 实验首先根据回溯算法构造每个可逆函数,如定义 2 所示,即生成 0~7 这 8 个整数的全排列,再运行 WHH 算法,快速找到每个函数对应的全部最优解. 在 DELL P4 3.0GHz 512MB 电脑上,若式(9)~(11)这 3 种优化技术中只考虑去除可化简的情况,即式(10),历时 6h5min10s;加入优先考虑 RM 因子,即式(9)、(10),历时 4h1min20s;应用全部优化技术,共历时仅为 1h58min44s. 这是目前国外文献的同类算法中速度最快、结果最优的算法. 而本领域的权威 Miller 教授的最新文献[19]公布的情况是,在 Sun Blade 1000 750MHz 电脑上,应用迭代算法,平均门数为 6.38,历时 33h;增加模板等优化技术,达到近似最优,历时却有 96h,表 7 实验数据表明,本文的算法与最优解完全相同,且时间极快,因此具有显著的优势.

表 7 3 变量可逆电路综合的实验数据				
量子门的数量	电路数量及门数量均值			
	本文最优	Pallav Guptas	Miller ^[19]	AJ ^[20]
9		36	2	30
8	577	3351	659	3297
7	10253	12476	10367	12488
6	17049	13596	16953	13620
5	8921	7479	8819	7503
4	2780	2642	2780	2642
3	625	625	625	625
2	102	102	102	102
1	12	12	12	12
0	1	1	1	1
门数量均值	5.866	6.10	5.875	6.101

为深入了解算法中各种优化技术对性能的影响,笔者采用 4 种方案分别实验,分别为(A)深度优

先,仅去除相邻且相同的量子门;(B)整体广度优先,局部深度优先,仅去除相邻且相同的量子门;(C)深度优先,应用全部优化技术;(D)整体广度优先,局部深度优先,应用全部优化技术,即本文最优的算法. 它们的共同点是获得全部最优解,不同的是运行速度有差异. 除了上面的优化技术有差异外,其它优化技术都相同,因此整体运行性能都比较高. 其中(A)种算法是对(C)种算法去除相应优化,而(B)种算法是对(D)种算法去除相应优化,上述的(C)种算法详细内容见算法 DFM,DFC.

引用算法 WHH 的全局变量,再加入全局变量 *irowmin*,表示当前最优解在树中的高度,即当前最优量子电路的门数加 1,初始值置为无穷大.

算法 5. 深度优先搜索最优量子电路的算法 DFC(*irow*).

输入:*irow* 为当前搜索的解空间树的层号
输出:返回是否搜索到最优的量子电路,若找到,则可以从全局数组 *mgate* 中读取量子门序列

1. if *irow* ≥ *irowmin* or *irow* > MAX + 1 then return false
2. *myok* = false
3. 引用算法 DFS 第 8~12 行代码
4. if RM[*irow*]是恒等式 then {
5. if *irowmin* > *irow* then {
6. *irowmin* = *irow*
7. return true }}
8. else if *irow* + 1 < *irowmin* and *irow* < MAX + 1 then {
9. if DFC(*irow* + 1) then *myok* = true }}
10. return *myok*

算法 DFC(*irow*)是对算法 DFS 进行了一些修改,它是从第 *irow* 层深度搜索最优解. 第 1 步,如果 *irow* ≥ *irowmin*,表示当前最优解使用的量子门数(*irowmin* - 1)比当前搜索生成的量子电路门数(≥ *irow* - 1)少,即当前搜索生成的量子电路不可能是最优的,则放弃搜索;如果 *irow* > MAX + 1,表示当前访问的层已超出解空间树的范围,则放弃搜索;算法 DFS 中已知解只能出现在叶子层,因此只需在叶子层寻找解,且找到的第一个解一定是最优解;而算法 DFC 不知道最优解在哪一层,因此必须在深度遍历时,通过各个解的层号比较,找出最小层的解,若当前电路是解,则在第 5~7 步记下最小层号 *irowmin*;否则,第 8 步判断第 *irow* + 1 层是否可能有最优解;如有则进入第 9 步,递归调用本算法,

判断能否从 $irow+1$ 层深度搜索到最优解,当本算法递归结束,若存在解, $irowmin$ 必为最优解所在的层号.

算法 6. 调用算法 DFC 生成量子可逆逻辑电路的算法 $DFM(f)$.

输入: 所求量子可逆逻辑电路的可逆函数 f

输出: 所求量子可逆逻辑电路的量子门序列

1. 引用算法 WHH 第 1~3 行代码
2. if not $DFC(2)$ then return 无解
3. return $mgate[1],mgate[2],\cdots,mgate[irowmin-1]$

算法的第 2 步若提前返回,是因为设定的 MAX 太小,无法生成所求的量子可逆逻辑电路.

表 8 各个长度的量子电路访问解空间树的节点总数

电路数量	电路长度	A	B	C	D
1	0	0	0	0	0
12	1	24	24	24	24
102	2	12334	2127	38446	2104
625	3	3646986	193195	2472032	149051
2780	4	177328722	13429360	81416811	7266155
8921	5	5254945610	604425324	1342678137	228599180
17049	6	46429446708	13570260781	9968606306	3671993552
10253	7	97168913824	64326465993	17878744171	13276279639
577	8	19382232051	21320452462	3053423134	3492586889
均值	5.866	4176997.179	2476072.154	801770.314	512819.36
运行总时间		14h59min	8h49min	3h13min	1h59min

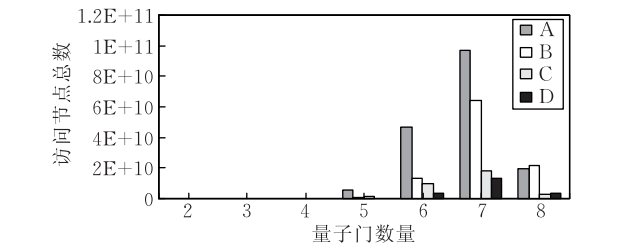


图 4 各个长度的量子电路访问解空间树的节点总数比较

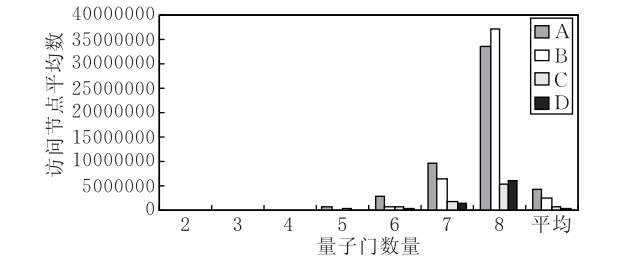


图 5 各个长度的量子电路平均访问解空间树的节点数比较

由表 8 可知,程序运行时间与访问节点的总数量基本成正比,因此优化算法主要考虑如何减少访问节点的总数量.表 8 中使用 1 个量子门时平均都访问了 $24/12=2$ 个节点,这是因为每个程序都优先考虑了 RM 的因子.在使用 2~7 个量子门时,D 算法访问节点数都比 C 算法少,但使用 8 个量子门时,D 算法访问节点数却比 C 算法多.是因为 D 算法总体层次遍历,设量子门数为 N ,D 算法访问节点数为依次深度遍历高度为 $2,3,\cdots,N+1$ 的解空间树的节点数之和,而 C 算法是在高度为 9 的解空间树上深度遍历.当使用量子门较少时, N 较小,D 算法只要依次深度遍历较少且较矮的解空间树,而 C

算法必须先深度遍历到较高层,然后逐步回溯到较低层,因此 D 算法访问节点数要比 C 算法少.当使用量子门较多时, N 较大,D 算法要依次深度遍历较多且较高的解空间树,重复访问了许多节点,而 C 算法只进行一次在高度为 9 的解空间树上深度遍历,在使用 8 个量子门时,即解空间树高度为 9 的遍历中,显然 D 算法访问节点数比 C 算法多.且算法 BBF 对访问节点数也有较大影响.在 3 个变量的量子电路中,图 6 表示的通用 Toffoli 门共有 $M=n2^{n-1}|_{n=3}=12$ 种,假设各种量子门使用的概率均等,则两个相邻的门共有 $12\times 12=144$ 种组合情况,对于非门、控制非门、Toffoli 门可移动的情况分别有 8,6,4 种,每种量子门只有一种可化简情况,即两个相邻的门相同,可移动的概率为 $\alpha=((8+6+6+4)\times 3)/(12\times 12)=50\%$,可化简的概率为 $\beta=12/(12\times 12)=8.33\%$.设电路中已有 $n-1$ 个门,若追加 1 个门,则可化简的概率为 p_n ,计算方法如下: $p_1=0$, $p_n=p_{n-1}+(\alpha(1-\beta))^{n-2}\times \beta$,而 $0<\alpha(1-\beta)<1$,因此当 n 较大时, p_n 收敛,计算可得 n 为 1,2, \cdots ,8 时,可化简的概率分别为 0,8.33%,12.15%,13.9%,14.7%,15.07%,15.24%,15.31%,最终收敛于 15.38%.由此可知,当量子门数增加时,可化简的概率不断增大,但最终收敛于固定值,保持不变.而每

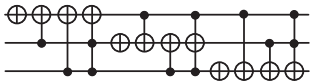


图 6 在 3 变量的实验中使用的全部量子门

次化简都可以减少访问解空间树的节点空间. 设树的高度为 h , 在第 i 层中若有一个节点可化简, 则可以从树中剪去 $(M^{h-i+1} - 1)/(M - 1) \mid_{M=12} = (12^{h-i+1} - 1)/11$ 个节点. 因此当 $h-i$ 越大, 去除的节点数越多.

由图 4 可知, 综合 6, 7, 8 个门的量子可逆逻辑电路时, 在解空间树中访问节点数最多, 对算法性能影响最大. 由图 5 可以知, 当量子门数增加时, 平均访问解空间树的节点数也快速增加, 在 C, D 算法中增加速度远没有解空间树的节点数增加速度快, 这是因为 C, D 算法应用了笔者的优化技术, 快速减少了解空间树可访问节点的空间.

实验数据表明, 笔者的优化技术显著提高了程序运行效率, 在相同的软硬件环境下, 加入全部优化技术的 C, D 算法明显优于没有加入主要优化技术的 A, B 算法, 但 A, B 算法又比其它同类算法快许多, 是因为它们除了没有应用主要优化技术外, 其它与 C, D 算法相同, 因此速度不会很慢, 笔者也实验过用高度为 9 的深度优先搜索, 仅去除相邻且相同的量子门, 历时却有 40 小时. 相比而言, D 的算法更加优化. 主要原因是: (1) 平均访问节点少; (2) 仅需判断解空间树的叶子节点是否为解, 因此判断是否为解的次数就更少了. 平均访问节点少的原因是, D 算法总体采用层次遍历, 在解空间树较矮时访问节点数很少; 当解空间树较高时访问节点数增多, 但采用 *BBF* 限界函数, 以较大的概率去除大量子树.

6 结 论

本文提出了一种基于 RM 的新颖高效的量子可逆逻辑电路综合算法, 以国际同行认可的 3 变量可逆函数测试标准, 该算法不仅能够生成全部最优电路, 而且运行速度远远超过同类算法. 在量子计算机的实现过程中, 量子逻辑门的量子代价与所采用的实现技术相关. 如何基于任意的量子代价标准, 使用任意的量子逻辑门, 高效构造最优的量子电路, 这只要对算法 *DFC* 适当修改便可实现, 且运行效率同样很高. 如何综合大规模的量子电路, 进一步提高效率, 还需要引入新的优化技术, 这是笔者下一步要研究的重要问题.

致 谢 对审稿人提出的有益建议表示感谢!

参 考 文 献

- [1] Deutsch D. Quantum computational networks//Proceedings of the Royal Society, London, 1985, 425: 73-90
- [2] Deutsch D, Barenco A, Ekert A. Universality in quantum computation//Proceedings of the Royal Society, London, 1995, 449: 669-677
- [3] Lloyd S. Almost any quantum logic gate is universal. *Physical Review Letters*, 1995, 75(2): 346
- [4] Landauer R. Irreversibility and heat generation of the computing process. *IBM Journal of Research and Development*, 1961, 5(3): 183-191
- [5] Bennett C H. Logical reversibility of computation. *IBM Journal of Research and Development*, 1973, 17(6): 525-532
- [6] Feynman R. Quantum mechanical computers. *Optic News*, 1985: 11-20
- [7] Toffoli T. Reversible computing//de Bakker J W, Van Leeuwen J eds. *Automata, Languages and Programming*. New York: Springer, 1980
- [8] Fredkin E, Toffoli T. Conservative logic. *International Journal of Theoretical Physics*, 1982, 21: 219-253
- [9] Shende V V, Prasad A K, Markov I L, Hayes J P. Reversible logic circuit synthesis//Proceedings of the International Conference on Computer-Aided Design. California, 2002, 125-132
- [10] Song X Y, Yang G W, Perkowski M, Wang Y. Algebraic characteristics of reversible gates. *Theory of Computing Systems*, 2006, 39(2): 311-319
- [11] Shende V V, Prasad A K, Markov I L, Hayes J P. Synthesis of reversible logic circuits. *IEEE Transactions on Circuits and Systems-I*, 2003, 22(6): 723-729
- [12] Iwama K, Kambayashi Y, Yamashita S. Transformation rules for designing CNOT-based quantum circuits//Proceedings of Design Automation Conference. New Orleans, 2002, 28(4): 419-424
- [13] Miller D M. Spectral and two-place decomposition techniques in reversible logic//Proceedings of the 45th IEEE International Midwest Symposium on Circuits and Systems. Tulsa, 2002: 493-496
- [14] Miller D M, Maslov D, Dueck G W. A transformation based algorithm for reversible logic synthesis//Proceedings of the International Conference on Computer-Aided Design. California, 2003: 318-323
- [15] Maslov D, Dueck G W, Miller D M. Toffoli network synthesis with templates. *IEEE Transactions on Circuits and Systems-I*, 2005, 24(6): 807-817
- [16] Mishchenko A, Perkowski M. Logic synthesis of reversible wave cascades//Proceedings of the 11th IEEE International Workshop on Logic Synthesis. New Orleans, 2002: 197-202
- [17] Gupta P, Agrawal A, Jha N K. An algorithm for synthesis of reversible logic circuits. *IEEE Transactions on Circuits and Systems-I*, 2006, 25(11): 807-817
- [18] Cheng S T, Wang C Y. Quantum switching and quantum merge sorting. *IEEE Transactions on Circuits and Systems-I*, 2006, 53(2): 316-325

[19] Maslov D, Miller D M, Dueck G W. Techniques for the synthesis of reversible toffoli networks. *ACM Transactions on Design Automation of Electronic Systems*, 2007, 12 (4): 42:1-42:28

[20] Agrawal A, Jha N K. Synthesis of reversible logic//*Proceedings*

of the Design, Automation and Test in Europe Conference and Exhibition. Paris, 2004, 2: 1384-1385

[21] Nielsen M A, Chuang I L. *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press, 2000



LI Zhi-Qiang, born in 1974, Ph. D. candidate, lecturer. His current research interests include quantum computing, quantum circuit synthesis.

CHEN Han-Wu, born in 1955, professor, Ph. D. supervisor. His current research interests include quantum computing, information theory.

XU Bao-Wen, born in 1961, professor, Ph. D. supervi-

sor. His current research interests include programming languages, software intelligence.

LI Wen-Qian, born in 1979, M. S. . His current research interests include quantum computing, quantum reversible logic circuit synthesis.

WANG Jia-Jia, born in 1981, M. S. . Her current research interests include quantum computing, quantum computer simulation

LIU Wen-Jie, born in 1979, Ph. D. candidate, lecturer. His current research interests include quantum computing, quantum communication.

Background

This work is supported by the National Natural Science Foundation of China under grant Nos. 60572071, 60873101, 90412014, Natural Science Foundation of Jiangsu Province under grant Nos. BK2008209, BK2007104, and Natural Science Foundation for colleges of Jiangsu Province under grant No. 06KJB520137.

A quantum computer is equivalent to a quantum Turing machine, which is an abstract mathematical model; and the quantum Turing machine is equivalent to a quantum logic circuit, as has been theoretically proven. Therefore, the quantum computer can be constructed by cascading and combining the quantum logical gates. How to automatically construct the desired quantum circuit with relatively small cost using appointed quantum gates? The principles of quantum logical gate cascading, the automatic production of basic quantum circuit and the optimization of quantum circuit are the foun-

dations of the study. An effective method is used to synthesize quantum reversible logic circuits using the positive polarity Reed-Muller (RM) expansion of a reversible function. To which Agrawal and Pallav Gupta et al. have made their own contributions. They tried to improve its performance by adopting a few heuristics algorithms; however, these heuristics algorithms cannot guarantee that a solution will be found when it does exist and can hardly get optimal result.

In this paper, a novel and efficient algorithm is presented, which can automatically derive RM expansion, and a solution space tree is constructed to effectively create the desired quantum reversible logic circuits. In the experiments on 3-qubit synthesis, the algorithm not only synthesizes all optimal reversible circuits, but also runs extremely faster than others of the same kind.