

DOI: 10.3969/j.issn.1007-5461.2023.04.016

一种二维架构下的量子电路布局与优化方法

张超, 管致锦*, 冯世光, 牛义仁, 朱明强

(南通大学信息科学技术学院, 江苏 南通 226019)

摘要: 为解决将量子电路映射到二维架构并实现量子位近邻问题, 提出了一种二维架构下的量子电路布局与优化方法。首先根据量子门在量子电路中的执行顺序和相互作用, 提出基于量子位权重的深度优先搜索量子位映射次序, 再考虑到映射次序的已放入量子位、待放入量子位和未放入量子位的关系进行量子位的初始布局实现量子位的初始映射; 进而对近邻过程中的相同前瞻量子代价的选择进行了优化, 再根据优化后的代价结果, 插入 SWAP 门, 实现所有双量子门的最近邻。最后利用实验对提出的方法进行了验证, 并与已有的方法进行了比较。结果表明所提出方法在中小规模的基准电路上平均优化率达到 18%, 在中大规模的基准电路上平均优化率达到 17%。

关键词: 量子物理; 量子电路; 量子映射; 最近邻; 二维架构

中图分类号: TP302.2

文献标识码: A

文章编号: 1007-5461(2023)04-00570-12

A quantum circuit layout and optimization method in two-dimensional architecture

ZHANG Chao, GUAN Zhijin*, FENG Shiguang, NIU Yiren, ZHU Mingqiang

(School of Information Science and Technology, Nantong University, Nantong 226019, China)

Abstract: In order to solve the problem of mapping quantum circuits to two-dimensional architecture and realizing qubit nearest neighbor, a quantum circuit layout and optimization method in two-dimensional architecture is proposed. Firstly, according to the execution order and interaction of quantum gates in quantum circuit, a depth-first search qubit mapping order based on the weight of qubits is proposed, then the initial qubit mapping is realized by taking into account the relationship between the put qubits in the mapping order, the qubits to be put in and the unput qubits. Secondly, the selection of the same look-ahead quantum cost in the nearest neighbor process is optimized, then according to the optimized cost results, SWAP gates are inserted to realize the nearest neighbor of all double quantum gates. Finally, the proposed method is verified by experiments and compared with the existing methods, and it is shown that

基金项目: 国家自然科学基金面上项目 (62072259), 江苏省研究生科研与实践创新计划项目 (SJCX20_1151)

作者简介: 张超 (1996 -), 江苏扬州人, 研究生, 主要从事计算机辅助量子逻辑综合方面的研究。E-mail: chao_z_ntu@qq.com

导师简介: 管致锦 (1962 -), 江苏连云港人, 博士, 教授, 博士生导师, 主要从事量子计算、逻辑综合、信息安全、计算机辅助智能制造等方面的研究。E-mail: guan.zj@ntu.edu.cn

收稿日期: 2021-04-12; 修改日期: 2021-05-24

*通信作者。

the average optimization rate of the proposed method reaches 18% on the small and medium-sized Benchmark and 17% on the medium and large-scale Benchmark.

Key words: quantum physics; quantum circuits; quantum mapping; nearest neighbor; two-dimensional architecture

0 引言

量子计算能够比经典计算更快地解决某些重要问题,如整数因子分解^[1]、未排序数据库搜索^[2]等。许多研究工作都致力于发现新的量子算法和量子技术,然而,当前的超导^[3]、量子点^[4]等量子技术具有相当大的局限性。目前的量子技术要求双量子位门之间发生相互作用时必须是近邻的状态且量子技术所支持的物理架构大多是二维的,最新的量子计算平台 IBMQ^[5]上支持的物理架构也都是二维的。

量子电路是量子算法的一种描述方法。而实现量子线路描述的量子算法,需要将量子电路映射在物理架构上。此外,考虑到当前量子技术的局限性,映射过程中还需要考虑双量子位的近邻约束。为了解决上述问题,研究人员提出各种将量子逻辑电路映射在物理架构上的方法^[6-10]。Shen等^[6]提出的和谐算法在小规模量子电路上具有很好的表现,但是随着电路规模的扩大,时间复杂度较高,极易产生超时问题。Hattori等^[7]提出SAT求解器来处理初始映射问题,但是在最近邻过程中采用的是A*算法。A*算法的搜索空间随着电路规模呈指数增长,不适用于映射大规模电路。Zulehner等^[8]提出了基于IBM QX架构的量子电路映射方法,该方法虽然可以处理任意量子位的耦合,但由于是穷尽搜索,运行时间较长,且初始映射缺乏全局优化能力。Zhang等^[9]提出了一种基于量子位活跃度的将一维量子电路映射在二维量子电路上的方法,该方法在大规模电路中的表现较好,但是在中小规模电路上表现不佳。Farghadan等^[10]提出了量子位放置优先级的概念,将映射方法分为三个部分:量子位映射次序搜索;二维映射布局;最近邻优化。该方法具有很好的扩展性,但是仍然存在两个问题:1)量子位次序搜索采用的方法只考虑了两个量子位之间关联的量子门数,没有考虑量子门在量子电路的先后次序;2)最近邻优化过程中没有考虑相同前瞻量子代价调整的问题。

本文提出了一种二维物理架构下的量子位布局和最近邻方法。该方法分为两个部分:量子位初始映射算法、最近邻优化算法。并且在最近邻优化上引入了相同前瞻量子代价的调整方法,该方法能够有效地解决二维物理架构下量子位的映射与最近邻问题。对IBMQ上的多个基准电路进行了实验验证,结果表明所提出的方法在大多数基准电路上均能实现一定程度的优化,其中最大优化率可达69%。

1 基本概念

1.1 量子比特

在经典计算机中,一个经典比特具有0和1两种可能状态。而在量子计算机中,一个量子比特也有两种可能状态: $|0\rangle$ 和 $|1\rangle$ 。不同于经典比特,量子比特的状态可以落在 $|0\rangle$ 和 $|1\rangle$ 之外,量子比特可以是状态的线性组合,称之为叠加态,表示为 $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$,其中 α 、 β 是复数,称为几率幅,且满足归一化条件,即 $|\alpha|^2 + |\beta|^2 = 1$ 。表示处在 $|0\rangle$ 的概率是 $|\alpha|^2$,处在 $|1\rangle$ 的概率是 $|\beta|^2$,这里 $|\alpha|$ 表示复数 α 的模。

1.2 量子电路

量子电路是用于量子计算的模型。量子电路上的量子位通过量子门来实现量子操作。量子门是作用在

量子位上的么正变换,用来修改量子位的状态。常用的量子门门库有 NCV、MCT、Clifford + T 门库等;常用的 Clifford+T 门库里的量子门有 CNOT、NOT、H、T 门。用符号表示量子门时,• 代表控制位,⊕ 代表目标位。一些基本量子门的符号表示如图 1 所示。

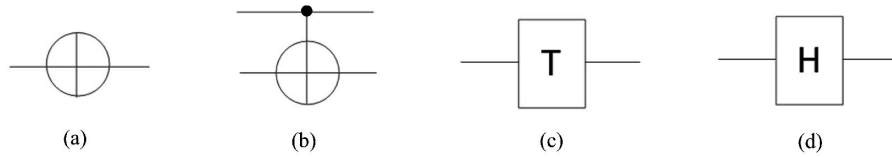


图1 基本量子门。(a) NOT 门;(b) CNOT 门;(c) T 门;(d) H 门

Fig.1 Elementary quantum gates. (a) NOT gate; (b) CNOT gate; (c) T gate; (d) H gate

2 量子位初始映射

量子位初始映射讨论将一维量子电路如何映射在二维网格布局上。本研究将量子位初始映射问题分为两个子问题进行讨论:1) 量子位究竟按照怎样的次序进行映射;2) 量子位应该如何映射在二维网格上。为了解决以上问题提出了量子位映射次序搜索算法和二维映射布局算法。

2.1 量子位映射次序搜索

定义1: 在 N 个量子门、 M 个量子位的一维量子电路 $G = G_1, G_2, G_3, \dots, G_N$ ($0 < i \leq N$) 中, 给每个量子门的位置编号 i , 其值 G_i 代表该量子门的位置权重, 量子门的权重之和 Q_{wj} ($0 < j \leq M$) 代表该量子位的权值。

在一维量子电路中, 某个量子门 g_i 的控制位 c_i 或者目标位 t_i 的位置权重 G_i 可以表示为

$$G_i = \frac{N-i+1}{N}, \quad (1)$$

量子电路中某条量子电路(量子位)上的权重 Q_{wj} 可以表示为

$$Q_{wj} = \sum_{i=1}^N G_i. \quad (2)$$

根据定义可知权值最大的量子位代表与其可以交互的量子门个数最多, 所以在量子位映射的过程中应该优先考虑量子位权重最大的量子位, 且考虑量子门在量子电路中的先后次序。因此, 在二维映射布局前要先确定量子位映射的次序, 此处提出了一种基于量子位权重的深度优先搜索量子位映射次序的方法。

用于初始映射的原始量子电路如图 2 所示, 量子门的序号是 1~11, 量子门个数是 11。根据 (1)、(2) 式可以计算出每条量子电路的权重。选出其中权重最大的量子位 q_3 。因为量子位的权重越大表明与之发生交互的量子位越多, 所以次序中的第一个量子位是权重最大的量子位。考虑到量子门在量子电路中的先后次序, 本研究采用深度优先搜索的方式搜索量子位映射次序。所以从量子位 q_3 开始进行深度优先搜索量子位映射次序。如图 2 所示的一维量子电路的量子位映射顺序是 q_3, q_1, q_2, q_0, q_4 。

量子位映射次序搜索算法如下:

算法 1: 量子位映射次序搜索

输入: N 个量子门, M 个量子位的一维量子电路: $G = G_1, G_2, G_3, \dots, G_N$. ($0 < i \leq N$), ($0 < j \leq M$)

输出: 量子位映射次序队列 queue

- 1: 生成当前量子位上的相关门列表 $gate$
- 2: 计算每个量子位上的权重 Q_{wj} 。
- 3: $queue[j] = \max(Q_{wj})$
- 4: 初始化临时存储栈 $stack[]$, 栈位置索引 s_index , 量子位映射次序队列 $queue[]$
- 5: $stack[0] = queue[j]$, $queue[0]=queue[j]$
- 6: **While** $len(queue) < M$
- 7: 初始化量子位上量子门的临时存储列表 $temp[]$, 临时存储位置索引 g_index
- 8: **For** $qbit \in gate[stack[s_index]]$ **do**
- 9: **If** $w \notin queue$ **do**
- 10: $stack.append(qbit)$, $q.append(qbit)$
- 11: $s_index = s_index + 1$
- 12: **break**
- 13: **If** $w \in queue$ **do**
- 14: $temp.append(w)$
- 15: $g_index = g_index + 1$
- 16: **If** $g_index = len(gate[stack[s_index]])$ **do**
- 17: $stack.pop()$ //回溯至上一个量子位
- 18: $s_index = s_index - 1$
- 19: **End For**

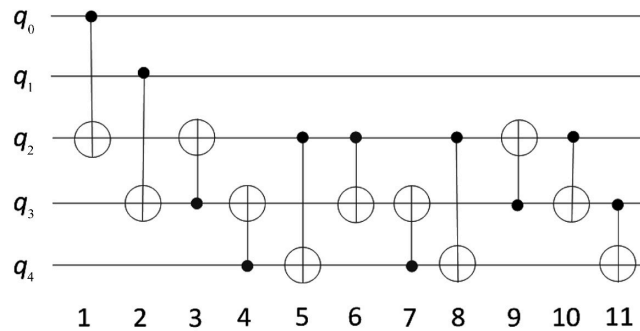


图2 用于初始映射的原始量子电路

Fig.2 The original quantum circuit for the initial mapping

2.2 二维映射布局

二维映射布局的目标是将由一维量子电路表示的量子算法上的量子位近邻地布局在二维网格上, 这样的量子位布局问题已经被证明是 NP 完全问题^[11]。根据 2.1 中的量子位映射次序搜索算法, 可以找到量子位的映射次序。在映射次序中考虑已放入量子位、待放入量子位和未放入量子位之间的关系, 为了衡量每个布局位置的优劣程度, 这里引入代价函数 $f^{[10]}$ 。

定义 2: 在 $m \times n$ ($m \geq 3, n \geq 3$) 的二维网格中, $(\lfloor \frac{m-1}{2} \rfloor, \lfloor \frac{n-1}{2} \rfloor)$ 称为网格的中心位置。如图 3 所示, 是一个 3×3 的二维网格, 中心位置是图中黑色的位置 (1, 1), 中心的位置有更多的位置与之近邻, 图中画圈的位置代价与中心相邻的位置。

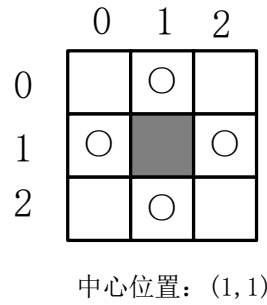


图3 3×3的二维网格中心位置示意图

Fig.3 A 3×3 two-dimensional grid center location diagram

根据2.1节中定义的量子位权重,可知权重越大的量子位表示与其交互的量子位越多。而在二维网格中越是中心的位置越可以与更多近邻位置上的量子位进行交互,因此将量子位映射次序中的第一个量子位映射在二维网格的中心,再根据量子位的映射次序完成量子电路上量子位的二维映射布局。

2.2.1 量子位交互图

量子位交互图可以表示量子位之间的相互关系,也可以描述量子电路中相互作用的量子比特的频率^[12],其中两量子位之间的关系代表两个量子位之间相关的量子门的个数。如图4所示的量子位交互图是基于图2的一维量子电路构建的,可以反映出该量子电路量子位之间的相互关系。其中有5个点,代表一维量子电路上的5个量子位,两个点之间的数代表两个量子位之间相互关联的量子门个数。

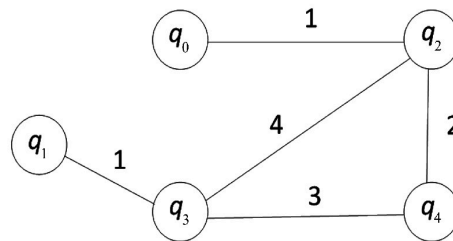


图4 量子位交互图

Fig. 4 Qubit interaction diagram

2.2.2 代价函数

在量子位映射过程中,为了衡量量子位映射位置的优劣,引入代价函数 f 。考虑到已放入量子位与待放入量子位之间的关系,引入 f_1 计算代价;考虑到待放入量子位与未放入量子位之间的关系,引入 f_2 计算代价。其中已放入量子位、待放入量子位和未放入量子位都是根据量子位映射次序确定的。再根据2.2.1节中提出的量子位交互图可以计算出量子位中已放入量子位、待放入量子位和未放入量子位之间的代价,选择代价最小的位置放置量子位。

代价函数 f 可以表示为

$$f = f_1 + f_2, \quad (3)$$

式中: $f_1 = \sum \left(\left(\text{dist} \left(d, \text{loc} \left(\text{nbr}[v][i][0] \right) \right) - 1 \right) \text{nbr}[v][i][1] \right)$,代表当前量子位与已放入量子位 $\text{nbr}[v]$ 之间的代价关系,其中 i 代表与当前量子位相关的已放置量子位索引, $\text{nbr}[v][i][0]$ 代表与当前量子位相关的已放置量

子位, $nbr[v][i][1]$ 代表与当前量子位相关的已放置量子位上的量子门个数, d 代表当前二维网格类型, $loc()$ 代表当前量子位所在位置和已放入量子位位置的欧氏距离; $f_2 = \min\{0, \deg(v) - frd(v)\} \frac{act(v)}{\deg(v)}$, 代表当前量子位与待放置量子位的代价关系, 其中 $\deg(v)$ 代表下一个待放置量子位与未放置量子位在量子位交互图中的度, $act(v)$ 代表下一个待放置量子位与未放置量子位在量子位交互图上的度的权值之和, $frd(v)$ 表示在当前网格周围可放置量子位的空白网格的个数。

根据量子位映射的次序, 将第一个量子位映射在网格的中心。然后根据当前所有位置, 找出所有位置的未放置的近邻位置, 根据代价函数 f 计算每个位置的代价, 选择代价最小的位置映射量子位。

二维映射布局的算法如下:

算法 2: 二维映射布局

输入: N 个量子门, M 个量子位的一维量子电路: $G = G_1, G_2, G_3, G_i \dots G_N$, ($0 < i \leq N$), ($0 < j \leq M$), 量子位映射次序队列 $queue$ 和 $W \times H$ 的二维网格 $grid$

输出: 量子位的二维布局

- 1: 生成量子位交互图 $G(V, E)$, $v \in V$
 - 2: 初始化顶点的待映射量子位与未映射量子位的度 $\deg(v)$
 - 3: 初始化所有度的权值之和表 $act(v)$
 - 4: 初始化当前量子位放置周围可继续放置位置的个数表 $frd(v)$
 - 5: 初始化量子位的邻接数组 $nbr[v]$
 - 6: 将 $q[0]$ 映射在二维网格 $grid$ 的中心位置。更新 $\deg(v)$, $act(v)$, $frd(v)$, $nbr[v]$
 - 7: **For** $temp_q_bit \in queue[1:]$ **do**
 - 8: 生成网格中已放置位置的近邻位置列表 $qbit_mapped_nbr_list$
 - 9: **For** $d \in qbit_mapped_nbr_list$ **do**
 - 10: 计算每个位置的代价 $f(d)$
 - 11: $\min(f(d)) \leftarrow grid(temp_q_bit)$
 - 12: 更新 $\deg(v)$, $act(v)$, $frd(v)$, $nbr[v]$
 - 13: **End For**
 - 14: **End For**
-

3 最近邻优化

初始映射并不能使量子电路上所有的量子门实现近邻, 需要插入 SWAP 门来更新映射, 从而在整个映射过程中量子电路的所有双量子门都能够进行近邻化操作。

3.1 量子代价

量子代价是衡量量子电路综合优化能力的重要指标。一个量子电路的量子代价通常可以通过量子电路的深度、执行时间、基本门数、最近邻过程中添加的 SWAP 门数等来衡量。在映射量子电路过程中,添加的 SWAP 门同时也会增加量子电路的深度和执行时间,因此减少 SWAP 门的数量可以减少量子电路的量子代价^[13]。此处将二维量子电路最近邻过程中插入的 SWAP 门的数量作为考虑的量子代价。

在二维量子电路中,量子代价可表示为目标位和控制位在网格中的欧氏距离减1,即

$$NNC(g_{ij}) = |x_i - x_j| + |y_i - y_j| - 1, \quad (4)$$

式中: g_{ij} 代表以 q_i 为控制位, q_j 为目标位的一个量子门, (x_i, y_i) 、 (x_j, y_j) 分别代表量子位 q_i 、 q_j 在网格中的横坐标和纵坐标。

3.2 基于相同前瞻量子代价调整的最近邻优化

将量子电路映射在二维网格中时,量子位在二维网格中的位置影响了整个量子电路映射的代价,因此需要将控制位移动到目标位近邻的位置,或者将目标位移动到控制位近邻的位置。本研究提出了一种基于相同前瞻量子代价调整的最近邻优化算法,考虑到量子电路最近邻过程中相同前瞻量子代价的情况,对根据量子代价选择插入 SWAP 门的方式进行了优化。

定义3: 对非近邻量子门 $q(q_i, q_j)$ 进行近邻化的过程中,通过添加 SWAP 门使得 q_i 与 q_j 处于近邻位置,将此时量子位 q_i 在网格中的位置称为控制位 q_i 与目标位 q_j 的相遇点。

规则1: 近邻化过程中,将双量子门在二维网格生成的局部网格中除控制位外的所有位置均设置为相遇点候选位置。图5是一个双量子门 CNOT 门在网格中生成的局部网格,其中 q_6 、 q_4 、 q_0 的位置均可以设置为相遇点的候选位置。

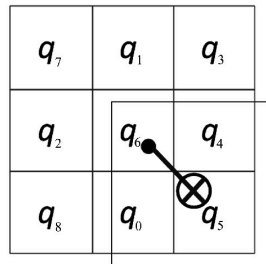


图5 CNOT 门生成的局部网格

Fig.5 A local grid generated by CNOT gate

定义4: 在对双量子门的控制位和目标位可能的相遇点进行计算选择时,需要计算的后续量子门个数称为前瞻窗口。前瞻窗口的选择需要根据实验对比选出比较合适的窗口。由实验可知,量子电路在前瞻窗口10以后的量子代价渐趋收敛,所以对窗口值由1~10进行了计算,选出其中量子代价最小的窗口。如图6所示,分别是量子电路4mod5-v1_22和alu-v0_27在窗口大小在1~10下的量子代价。

在对量子电路进行近邻化操作时,首先遍历其中的每一个双量子门,当双量子门是非近邻状态时,需要对该双量子门的控制位和目标位之间可能的相遇点进行计算选择。相遇点根据规则1生成,对每一个可能的相遇点进行前瞻量子代价计算时,会遇到相同的前瞻量子代价问题。此处考虑用前瞻窗口中最小量子代价的双量子门来选择合适的相遇点,其最近邻优化的具体算法如算法3所示。

例如, 对于如图 7 所示的非近邻量子门, 需要将量子位 q_6 与 q_5 实现近邻, 有 3 个相遇点候选位置。如图 8 所示, 量子电路中的前瞻窗口为 3, 计算每一个前瞻窗口中量子门的量子代价, 结果如图 9 所示, 其中前三个插入方式下的前瞻量子代价最小且都为 1, 但是第一个插入 SWAP 门的方式对量子电路后续优化效果是最好的, 因为当一个量子门执行后对后续量子门的影响越来越小。因而选择前瞻量子代价中对后续双量子门的量子代价最小的相遇点来插入 SWAP 门, 以实现双量子门的近邻操作。如图 9 所示, 方框中前三个插入方式下的数是前瞻窗口中当前执行量子门的后一个量子门的量子代价, 分别是 0, 1, 1。其中插入方式 1 的量子代价是最小的。

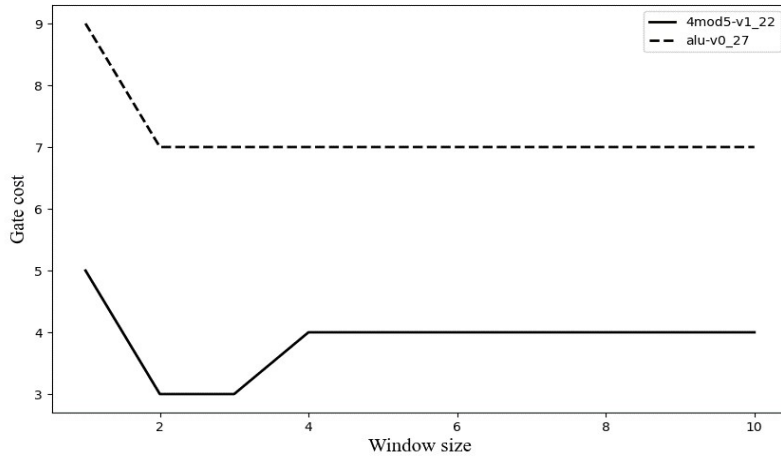


图 6 量子电路 4mod5-v1_22 和 alu-v0_27 在不同窗口下的量子代价

Fig. 6 Quantum cost of 4mod5-v1_22 and alu-v0_27 in different windows

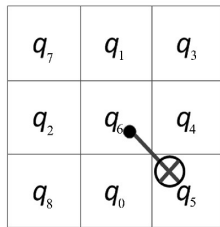


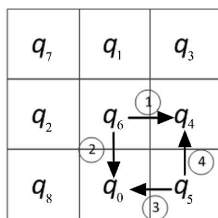
图 7 非近邻量子门

Fig.7 No-adjacent quantum gate

- 0 CNOT q_2, q_6
- 1 CNOT q_0, q_5
- 2 CNOT q_6, q_5
- 3 CNOT q_1, q_4
- 4 CNOT q_3, q_4
- 5 CNOT q_2, q_3
- 6 CNOT q_0, q_4

图 8 量子电路

Fig.8 Quantum circuit



$$dist(x, y) = |x - x_j| + |y - y_j| - 1$$

插入方式 ① $cost(q_6 \rightarrow q_4) = dist(q_6, q_5) + dist(q_1, q_4) + dist(q_3, q_4) = 0 + 0 + 1 = 1$

插入方式 ② $cost(q_6 \rightarrow q_0) = dist(q_6, q_5) + dist(q_1, q_4) + dist(q_3, q_4) = 0 + 1 + 0 = 1$

插入方式 ③ $cost(q_5 \rightarrow q_4) = dist(q_6, q_5) + dist(q_1, q_4) + dist(q_3, q_4) = 0 + 1 + 0 = 1$

插入方式 ④ $cost(q_5 \rightarrow q_0) = dist(q_6, q_5) + dist(q_1, q_4) + dist(q_3, q_4) = 0 + 2 + 1 = 3$

图 9 四种插入 SWAP 门方式下的量子代价

Fig. 9 Quantum cost of four ways of inserting SWAP gate

最近邻优化算法表示如下:

算法3: 最近邻优化

输入: N 个量子门, M 个量子位的一维量子电路: $G = G_1, G_2, G_3, G_i, \dots, G_N, (0 < i \leq N), (0 < j \leq M)$, 一个映射在 $W \times H$ 网格上的初始映射, 前瞻窗口 window

输出: 量子位在二维网格中的最终映射和总的插入 SWAP 门数 SWAP_num

代价函数: $\sum_{i=0}^{w-1} \text{dist}(w[i]c, w[i]t)$

1: For non-adjacent gate $\in G$ do

2: 移动一个前瞻窗口 window

3: 生成相遇点列表 meeting_point_list \leftarrow via local grid(con, tar)

4: For meeting_point \in meeting_point_list do

5: 生成 con 到 meeting_point 的所有最短路径 con_mp_path_list

6: For con_mp_path \in con_mp_list do

7: 计算 con_mp_path 的量子代价 con_mp_qc

8: If qc[i] == qc[j] do // 存在相同前瞻量子代价

9: 计算前瞻窗口 window 中的每一个 gate 量子代价

10: min_con_mp_path \leftarrow min(dist(gate))

11: End For

12: 生成 tar 到 meeting_point 的所有最短路径 tar_mp_list

13: For tar_mp_path \in tar_mp_path_list do

14: 计算 tar_mp_path 的量子代价 tar_mp_qc

15: If qc[i] == qc[j] do // 存在相同前瞻量子代价

16: 计算前瞻窗口 window 中的每一个 gate 量子代价

17: min_tar_mp_path \leftarrow min(dist(gate))

18: End For

19: End For

20: 生成该相遇点路径 m_p_path \leftarrow min_con_mp_path + min_tar_mp_path

21: For m_p_path \in meeting_point_list

22: If qc[i] == qc[j] do // 存在相同前瞻量子代价

23: 计算前瞻窗口 window 中的每一个 gate 量子代价

24: min_m_p_path \leftarrow min(dist(gate))

25: SWAP_num = len(min_m_p_path) - 2 // 计算 SWAP 门数

26: End For

27: End For

4 实验结果与讨论

用 Python 语言实现本研究所提出的方法, 实验环境为 Window 10 操作系统、Intel core i5 处理器和 8 GB 的内存, 量子电路是 IBMQ 上提供的 QASM 基准电路, 网格的大小均为 4×5 。表 1 是各量子电路在不同的前瞻窗口下的量子代价, 选择其中最小的量子代价作为所提出方法中的最后代价, 其中每行的加粗部分是最后选择的量子代价。表 2 是本研究所提出方法与文献 [9] 提出方法的比较, 表中前 4 列分别代表量子电路的类型 (Type)、线路名称 (Name)、量子位数 (N) 和原电路的量子门数 (G), 其中 Small 代表小型量子算法, Sim 代表量子模拟算法, Qft 代表量子傅里叶变换算法, Large 代表大型量子算法。第五列是文献 [9] 提出方法的量子代价 (QC), 第六列是本研究提出方法的量子代价 (QC)。最后一列是本研究所提出方法与文献 [9] 提出的方法在相同二维架构下的优化率 Impr。实验结果表明, 本研究所提出方法在实验的量子电路上能够实现一定程度的优化, 其中最高、最低及平均优化率分别为 69%、2% 与 18%。

表 1 每一个前瞻窗口大小下的量子代价

Table 1 Quantum cost of each look-ahead window size

Benchmarks				Quantum cost in different windows									
Type	Name	N	G	1	2	3	4	5	6	7	8	9	10
Small	4mod5-v1_22	5	21	5	3	3	4	4	4	4	4	4	4
Small	alu-v0_27	5	36	9	7	7	7	7	7	7	7	7	7
Small	decod24-v2_43	4	52	16	10	10	11	10	10	11	10	10	11
Small	4gt13_92	5	66	19	13	17	13	16	14	15	13	14	12
Sim	ising_model_16	16	786	10	16	10	17	15	13	9	9	9	9
Qft	qft_10	10	200	49	49	24	27	25	28	27	28	24	23
Qft	qft_13	13	403	100	91	52	62	51	47	52	49	57	55
Qft	qft_16	16	512	176	167	108	108	95	105	95	106	92	95
Qft	qft_20	20	970	332	260	182	166	153	154	158	178	157	172
Large	rd84_142	15	343	114	89	86	83	85	83	93	92	88	82

表 2 本研究所提出方法与文献 [9] 方法对比的实验结果

Table 2 Experimental results comparison of the proposed method and literature [9]

Benchmarks				QC		
Type	Name	N	G	Literature [9]	Results	Impr/%
Small	4mod5-v1_22	5	21	4	3	25
Small	alu-v0_27	5	36	8	7	13
Small	decod24-v2_43	4	52	10	10	0
Small	4gt13_92	5	66	13	12	8
Sim	ising_model_16	16	786	43	9	69
Qft	qft_10	10	200	29	23	21
Qft	qft_13	13	403	48	47	2
Qft	qft_16	16	512	101	92	9
Qft	qft_20	20	970	107	153	-
Large	rd84_142	15	343	94	82	13

如表3所示,为了公平地将本研究提出的方法与文献[10]提出的方法作比较,此处不考虑前瞻窗口的变化,统一将前瞻窗口设置为4。通过实验相同的量子电路,将本研究的实验结果与文献[10]的实验结果进行比较,结果表明,本研究所提出方法对于大部分量子电路有一定程度的优化,尤其在中小规模量子电路上效果最佳。在 4×5 的二维架构下,本研究提出方法所实验的量子电路中有92%能达到优化效果,其中最高、最低及平均优化率分别为64%、2%和17%。

表3 本研究所提出方法与文献[10]方法对比的实验结果

Table 3 Experimental results comparison of the proposed method and literature [10]

Benchmarks				QC		
Type	Name	N	G	Literature [10]	Results	Impr/%
Small	4mod5-v1_22	5	21	4	4	0
Small	mod5mils_65	5	35	7	6	14
Small	alu-v0_27	5	36	12	7	42
Small	decod24-v2_43	4	52	11	4	64
Small	4gt13_92	5	66	18	13	28
Sim	ising_model_10	10	480	0	10	-
Sim	ising_model_13	13	633	24	11	54
Sim	ising_model_16	16	786	17	12	29
Qft	qft_10	10	200	28	23	18
Qft	qft_13	13	403	71	56	21
Qft	qft_16	16	512	124	105	15
Qft	qft_20	20	970	176	166	6
Large	rd84_142	15	343	105	86	18
Large	adr4_197	13	3439	983	860	13
Large	radd_250	13	3213	884	812	8
Large	z4_268	11	3073	859	810	5
Large	sym6_145	14	3888	1023	902	12
Large	misex1_241	15	4813	1298	1184	9
Large	rd73_252	10	5321	1467	1356	8
Large	cycle10_2_110	12	6050	1681	1531	9
Large	square_root_7	15	7630	1681	1572	6
Large	sqn_258	10	10223	2832	2465	13
Large	rd84_253	12	13658	3884	3533	9
Large	co14_215	15	17936	5238	5112	2
Large	sym9_193	10	34881	9725	8914	8
Large	9symml_195	11	34881	9725	8937	8

5 结 论

二维物理架构下的量子位映射和最近邻实现是目前量子设备实现量子计算的巨大障碍。所提出量子位

初始映射包含基于量子位权重的深度优先搜索量子位映射次序和二维量子位映射布局,该方法可以较好地完成量子位的初始映射和最近邻。实验证明该方法有效地降低了电路的整体量子代价,同时也适用于其他二维架构。

参考文献:

- [1] Shor P W. Algorithms for quantum computation: Discrete logarithms and factoring [C]. *Proceeding of 35th Annual Symposium on Foundations of Computer Science*, 1994: 124-134.
- [2] Grover L K. A fast quantum mechanical algorithm for database search [C]. *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, 1996: 212-219.
- [3] Córcoles A D, Magesan E, Srinivasan S J, *et al.* Demonstration of a quantum error detection code using a square lattice of four superconducting qubits [J]. *Nature Communications*, 2015, 6: 6979.
- [4] Taylor J M, Petta J R, Johnson A C, *et al.* Relaxation, dephasing, and quantum control of electron spins in double quantum dots [J]. *Physical Review B*, 2007, 76(3): 035315.
- [5] IBM Q Experience [OL]. <https://www.research.ibm.com/ibm-q/technology/experience/>.
- [6] Shen M Y, Cheng X Y, Guan Z J, *et al.* Realization method of two-dimensional nearest neighbor for quantum circuit [J]. *Chinese Journal of Quantum Electronics*, 2019, 36(4): 476-482.
沈鸣燕,程学云,管致锦,等.一种量子线路二维近邻实现方法[J].量子电子学报,2019,36(4):476-482.
- [7] Hattori W, Yamashita S. Mapping a quantum circuit to 2D nearest neighbor architecture by changing the gate order [J]. *IEICE Transactions on Information and Systems*, 2019, E102.D(11): 2127-2134.
- [8] Zulehner A, Paler A, Wille R. An efficient methodology for mapping quantum circuits to the IBM QX architectures [J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019, 38(7): 1226-1236.
- [9] Zhang Y X, Guan Z J, Ji L Y, *et al.* A method of mapping and nearest neighbor optimization for 2D quantum circuits [J]. *Quantum Information and Computation*, 2020, 20(3&4): 194-212.
- [10] Farghadan A, Mohammadzadeh N. Quantum circuit physical design flow for 2D nearest-neighbor architectures [J]. *International Journal of Circuit Theory and Applications*, 2017, 45(7): 989-1000.
- [11] Shafaei A, Saedi M, Pedram M. Qubit placement to minimize communication overhead in 2D quantum architectures [C]. *19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2014: 495-500.
- [12] Lin C C, Sur-Kolay S, Jha N K. PAQCS: Physical design-aware fault-tolerant quantum circuit synthesis [J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2015, 23(7): 1221-1234.
- [13] Cheng X Y, Tan Y Y, Guan Z J. *et al.* An optimized simplification algorithm for reversible MCT circuits [J]. *Chinese Journal of Quantum Electronics*, 2017, 34(6): 713-720.
程学云,谈莹莹,管致锦,等.优化的可逆MCT电路化简算法[J].量子电子学报,2017,34(6):713-720.