

文章编号: 2095-4980(2024)01-0096-09

一种提高量子线路保真度的映射方法

牛义仁, 管致锦*, 马天宏, 金圣翔

(南通大学 信息科学与技术学院, 江苏 南通 226019)

摘要: 由噪声导致的双量子比特门操作错误是可逆的量子线路执行结果保真度下降的重要原因。提出一种将量子线路映射至含噪声量子计算设备的策略, 提高量子线路在量子计算设备上的执行成功率。该策略给出一种寻找近邻路径方法, 用于在量子比特不是全连通的量子计算设备上找出 2 个量子比特的近邻路径; 由于在近邻路径上任意 2 个近邻量子比特的噪声一般不同, 基于近邻路径上的双量子比特门执行成功率构建启发式函数, 估算前瞻窗口内的线路保真度, 并根据该代价函数选择保真度最高的近邻方式。通过测试多个 Benchmarks, 实验结果表明, 与 Qiskit 工具包中 SabreSwap 和 BasicSwap 算法相比, 本文所提策略的量子线路保真度平均提高 65.67% 和 71.60%。该方法可以提高量子线路保真度。

关键词: 量子线路; 映射; 量子计算设备; 保真度; 可逆

中图分类号: TN02; TP391.72

文献标志码: A

doi: 10.11805/TKYDA2021406

A mapping method to improve fidelity of quantum circuits

NIU Yiren, GUAN Zhijin*, MA Tianhong, JIN Shengxiang

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

Abstract: The error of the double-qubit gate operation caused by the noise is an important reason for the reduction of the fidelity of the execution result in the reversible quantum circuit. A strategy for mapping quantum circuits to noisy quantum computing devices is proposed, so as to improve the success rate of quantum circuits on quantum computing devices. This strategy first provides a method for finding the neighbor path of two qubits on a quantum computing device in which the qubits are not fully connected; since the noise of any two neighbor qubits on the neighbor path is generally different, a heuristic function is constructed based on the execution success rate of the double-qubit gate on the nearest neighbor path to estimate the fidelity of the line in the look-ahead window, and the nearest neighbor method with the highest fidelity is selected according to the cost function. After testing multiple Benchmarks, the experimental results show that the proposed strategy improves the fidelity of the quantum circuit by an average of 65.67% and 71.60% respectively compared with the SabreSwap and BasicSwap methods in Qiskit. The proposed method can improve the fidelity of quantum circuits

Keywords: quantum circuit; mapping; quantum computing device; fidelity; reversible

不同于传统的太赫兹组件, 基于硅基的太赫兹系统在大规模使用情况下具有成本低, 尺寸小, 集成度高, 操作性强, 更容易实现大阵列等特点^[1]。据 cnBeta 2021 年 9 月 7 日消息, 昆士兰大学的物理学家和工程师们找到了一种新方法, 用来识别和解决商用量子计算中的材料缺陷。超导量子线路的应用受退相干现象的阻碍, 会导致信息丢失。退相干现象主要是由超导电路和硅芯片之间的相互作用以及在制造过程中引入的材料缺陷共同导致的。研究团队使用太赫兹扫描近场光学显微镜, 实现了在纳米尺度而非宏观尺度的探测, 减少了退相干现象, 提供了一条改进超导设备用于量子计算的途径^[2]。

量子计算在整数因子分解、量子模拟、数据库搜索等问题上具备超越经典计算的潜在优势, 因而受到广泛关注。目前量子计算设备仅是一种量子计算原型机, 存在易出错和不支持量子容错计算等问题, 但已为未来的

收稿日期: 2021-11-30; 修回日期: 2022-02-09

基金项目: 国家自然科学基金资助项目(62072259); 福建省科技厅引导性资助项目(2021H0029), 江苏省自然科学基金面上项目资助(BK20221411)

*通信作者: 管致锦 email:guan.zj@ntu.edu.cn

通用量子计算设备发展打下了良好的基础。

在超导量子计算设备上,通常量子比特之间并不是全连通的,只有少数近邻量子比特能直接相互作用,量子比特之间的这种有限连通性也称为连通约束。量子算法一般用量子线路的形式表示,在量子计算设备上运行量子线路时,需要将量子线路中的逻辑量子比特映射至量子计算设备上的物理量子比特。一般情况下很难保证量子线路中的所有双量子比特操作均满足连通约束,因此通过引入交换(SWAP)门对线路进行变换,将双量子比特操作相关的逻辑比特移动到近邻的物理量子比特上。量子线路变换一般分为 2 个步骤: a) 初始化映射: 将逻辑量子比特分配给物理量子比特; b) 量子比特移动: 将逻辑量子比特从一个位置移动到另一个位置^[3]。

除连通约束外,目前的量子计算设备还易受各种噪声影响,如退相干,从而降低量子线路执行时的保真度。因此,使用保真度高的量子操作对提高计算成功率至关重要。当前,含噪量子计算设备上的量子操作容易出错,虽然使用量子纠错码(Quantum Error Correction codes, QEC)可以保护量子比特免受错误,但 QEC 需要大量的物理资源开销,通常需要 10~100 个物理量子比特编码一个容错量子比特^[4]。由于现有的量子计算设备的量子比特数有限, QEC 无法使用。

已有的研究提出了以最小化交换(SWAP)门数为优化目标的量子比特分配策略^[5-7],这些研究假设所有量子比特和量子操作的质量相同,但在实际量子计算中,量子比特和量子操作表现出不同的质量特性,这些研究没有解决好保真度问题。文献[3]表明,在线路变化时考虑这种量子操作质量的变化对总体保真度有显著影响。为了提高保真度,应该尽量将量子门映射到高保真度的量子比特上。Murali 等^[8]使用可满足性模理(Satisfiability Modulo Theories, SMT)求解器生成量子线路时考虑了量子计算机的连通约束以及多种错误参数,实验结果表明其优于国际商业机器公司(International Business Machines, IBM)的 Qiskit 方案,但该方法只适合小规模量子线路。随着量子线路规模扩大,量子线路映射问题成为非确定性多项式(Non-deterministic Polynomial, NP)完全问题^[6]。受控非门(Controlled-NOT, CNOT)错误指可以翻转 1 个或 2 个量子位的状态值或相位,线路中的量子门对量子比特进行修改时,会以一定的概率引入错误修改,错误率越低,得出正确结果的概率越高。文献[9]采用估计成功率(Estimated Success Probability, ESP)作为保真度衡量指标,但其只考虑了最短路径上的近邻方式对保真度的影响,并未考虑其他路径的近邻方式。当前门的近邻影响着后续门,选择最短路径上的近邻方式,具有局限性。对后续门的影响是因为当前门选择的近邻方式较少,没有考虑更多的近邻方式。事实上,对于线路总体保真度,最短路径的近邻方式未必能带来全局范围的改进。

本文以提高量子线路执行保真度为目标,提出一种量子线路变换策略,通过感知量子门保真度的差异,选择最有利提高线路保真度的交换(SWAP)门。

1 背景

1.1 量子比特与量子线路

量子计算机的基本单元是量子比特,量子比特有 2 种基本状态 $|0\rangle$ 和 $|1\rangle$ 。与经典比特不同,量子比特能够同时处于 $|0\rangle$ 和 $|1\rangle$ 的叠加态,表示为 $|\varphi\rangle=\alpha|0\rangle+\beta|1\rangle$ 。其中, α 和 β 为复数,且满足 $|\alpha|^2+|\beta|^2=1$, α 和 β 分别为 $|0\rangle$ 和 $|1\rangle$ 两个基态对应的概率幅,当对该量子态进行测量时,量子态塌缩到 $|0\rangle$ 和 $|1\rangle$ 的概率分别为 $|\alpha|^2$ 和 $|\beta|^2$ 。

量子线路由若干量子比特和量子门构成,量子线路中的每一条横线表示一个量子比特^[10],如图 1 所示。这是由 2 个受控非门(CNOT)组成的量子线路,线路中的 CNOT 门是一种常见的双量子比特门,用符号 CNOT(C,T)表示,其中符号 C 和 T 分别代表控制位和目标位。图 1 中的 2 个 CNOT 门,分别为 CNOT(q_0,q_2)、CNOT(q_2,q_1)。

1.2 量子计算设备连通约束与保真度

图 2 给出一个量子计算设备 ibmq_quito 拓扑结构,图中 Q_0 、 Q_1 、 Q_2 、 Q_3 、 Q_4 分别代表物理量子比特,双向边表示物理量子比特之间的近邻关系。只有近邻的物理量子比特可以相互作用,即双量子比特门只可以相互作用在近邻的物理量子比特上。逻辑量子线路的量子比特 q_i 到量子计算设备的物理量子比特 Q_j 的初始化指按照 $\{q_i \rightarrow Q_j\}$ 映射, $i \in \{1,2,\dots,n\}$, $j \in \{1,2,\dots,m\}$, $n \leq m$,逻辑量子比特数不能超过物理量子比特数。默认初始化指按照 $\{q_i \rightarrow$

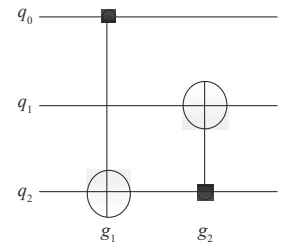


Fig.1 Quantum circuit
图 1 量子线路

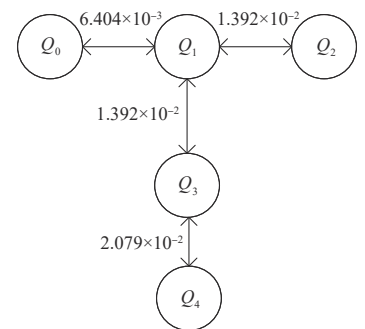


Fig.2 ibmq_quito topology
图 2 ibmq_quito 拓扑结构

Q_j 进行映射, 假设图 1 中的线路按默认初始化映射 $\{q_0 \rightarrow Q_0, q_1 \rightarrow Q_1, q_2 \rightarrow Q_2\}$ 映射到图 2 所示的拓扑结构, 即 g_1 的量子比特分别映射到物理量子比特 Q_0 和 Q_2 , 由于违反连通约束, 此时该量子门不能直接执行, 需引入额外的辅助门将该门相关的量子比特移动到近邻位置; g_2 的 2 个逻辑量子比特分别映射到物理量子比特 Q_2 和 Q_1 上, 此时满足连通约束, 该门可直接执行。图 2 中的数字代表 CNOT 门的错误率。

可使用 SWAP 门将双量子门的量子比特移动到量子计算设备的任意近邻物理量子比特。由于 SWAP 门不是量子计算设备直接支持的基本门, 因此将其分解成基本门, 分解方式如图 3 所示。

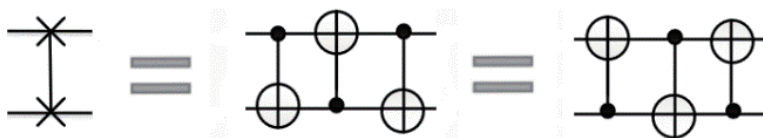


Fig.3 SWAP decomposition

图3 SWAP门分解

保真度指量子线路在量子计算设备上运行并输出正确结果的概率。本文主要基于 IBM 量子计算架构讨论保真度。从图 2 可见, 作用在不同量子比特上的 CNOT 门错误率不一样, 最高错误率与最低错误率相差近 2 倍^[1]; 双量子门的错误率比单量子门高一个数量级。以图 2 为例, 当同一个 CNOT 门映射到不同的 2 对近邻量子比特上时, 执行正确率一般存在较大差异。因此, 在进行线路变化时, 除了考虑量子设备的连通约束, 还需考虑量子操作作用在不同量子比特上表现出的保真度差异。

2 提高保真度的映射策略

当在量子计算设备上执行逻辑量子线路时, 需满足量子计算设备的物理约束。以 IBMQ 为例, 单量子比特门可以直接执行, 不近邻的双量子比特门即 CNOT 门需要通过线路变换满足设备的物理约束。CNOT 门在变换中合理地插入 SWAP 门, 在满足物理约束的同时, 提高了量子线路的保真度。

2.1 近邻路径构建算法

在量子计算设备上, 近邻方式取决于 2 个量子比特之间的路径。2 个量子比特之间存在 1 条或多条路径, 其不同的近邻方式, 执行后的保真度一般不同。

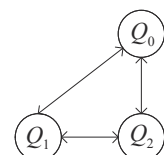
定义 1: 2 个物理量子比特之间的路径 $a_1-a_2-\dots-a_n$ 称为近邻路径 w , 其中 $a_i \neq a_j$, $1 \leq i < j \leq n$, 路径 w 中无重复的量子比特, 并且每对 (a_i, a_{i+1}) 是 2 个可以直接相互作用的物理量子比特, a_1 代表控制位 C , a_n 代表目标位 T 。

图 4 所示的拓扑结构上, 物理量子比特 Q_0 和 Q_2 之间的路径 $Q_0-Q_1-Q_2$ 和 Q_0-Q_2 为近邻路径, 该路径不包含重复的量子比特。对于拓扑结构图上存在 1 条或多条近邻路径, 本文首先提出一种在 2 个量子比特之间寻找全部的近邻路径方法。

在图 5(a)所示的量子计算设备的拓扑结构上, Q_0 、 Q_1 、 Q_2 、 Q_3 代表物理量子比特。因为通过插入 SWAP 门可以将逻辑位移动到任意的物理量子比特上, 因此可将量子计算设备的拓扑结构图看作一个连通图, 寻找量子拓扑结构近邻路径问题就可看作寻找连通图的近邻路径问题。本文所提寻找全部近邻路径基本思想为: 若起点为 Q_0 (即控制位 C), 终点为 Q_3 (即目标位 T), 则将 Q_0 记为已访问过, 如图 5(a)所示, 然后随机选取与 Q_0 邻接且未被访问的顶点 Q_1 , 并将 Q_1 记为已访问过; 如图 5(b)所示, 再随机选取与 Q_1 邻接且未被访问的顶点 Q_2 , 并将 Q_2 记为已访问过; 以此重复进行, 直至图中顶点的邻接点全部被访问或访问到终点 Q_3 为止。当访问完终点或该顶点的全部邻接点时进行回退。当顶点 Q_3 的唯一邻接顶点 Q_2 被访问过时 (如图 5(c)所示), 则根据访问的顺序回退到上一个访问过的顶点 Q_2 , 记顶点 Q_3 为未访问的顶点; 当顶点 Q_2 所有的邻接顶点 Q_0 、 Q_1 和 Q_3 都被访问过时 (如图 5(d)所示), 则根据访问的顺序回退到上一个访问过的顶点 Q_1 , 记顶点 Q_2 为未访问的顶点; 若回退到顶点 Q_0 时, Q_0 还有其他未被访问的邻接点 (如图 5(e)所示), 则从未被访问的顶点中选取 Q_2 并重复上述访问过程。按序记录访问过的顶点 Q_0 、 Q_1 、 Q_2 、 Q_3 和 Q_0 、 Q_2 、 Q_3 , 即为近邻路径。

算法 1 中, 变量 $path$ 表示控制位与目标位之间的 1 条近邻路径; $Paths$ 表示控制位与目标位之间的所有近邻路径集合; $AdjList$ 表示可以与当前量子比特之间交换的量子比特集合; 集合 $records$ 表示其中量子比特是否被访问, 若出现在 $path$ 集合中, 表示被访问; $G(V, E)$ 表示量子计算设备拓扑结构图。

算法 1 中第 2 行表示寻找出可与当前量子门的控制位直接相互作用的量子比特, 放入集合 $AdjList$ 中。第 4 行

Fig.4 The topology
图4 拓扑结构

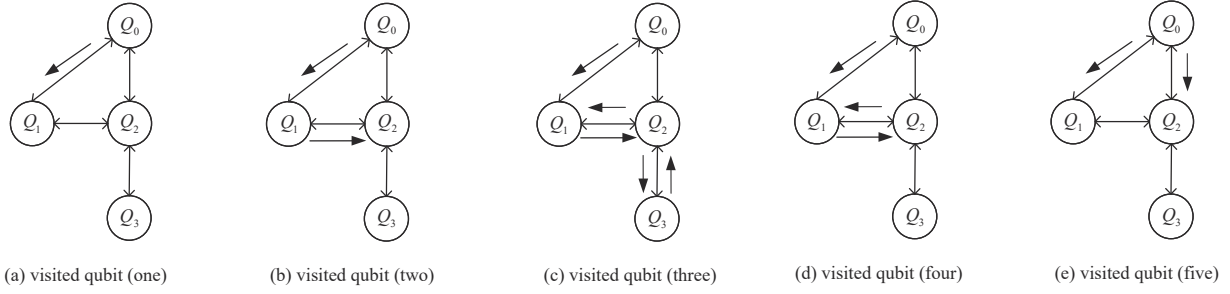


Fig.5 Find the nearest neighbor path on the connected graph

图 5 连通图上寻找近邻路径

algorithm 1 find the nearest neighbor path

input: $C, T, path, Paths, records, G(V, E)$

output: $Paths$

```

1  function GET_PATH( $G, C, T$ )
2  AdjList  $\leftarrow$  graph[C]; /* get a qubit that can interact with the current qubit */
3  If records[AdjList[i]] = 0
4  for AdjList do
5      If records[AdjList[i]] = 0
6          path.add(AdjList[i]) /* add the qubit to a nearby path */
7          records[AdjList[i]]  $\leftarrow$  1 /* mark that the current qubit has been visited */
8          if AdjList[i] = T then /* If the current visited qubit is the target bit */
9              Paths.add(path)
10             GET_PATH( $G, C, T$ ); /* call the current algorithm recursively */
11             path.pop(); /* retrace the last visited qubit */
12             records[AdjList[i]]  $\leftarrow$  0 /* mark that the current qubit has not been visited */

```

表示遍历 $AdjList$ 集合，取出其中的一个量子比特作为当前量子比特，放入 $path$ 中，并设置当前量子比特已经访问过，如果当前访问的量子比特为目标位，则将 $path$ 添加到 $Paths$ 中。第 10 行表示调用当前算法。第 11 行和第 12 行表示当按照遍历访问完量子比特时，返回上一个量子比特，设置当前量子比特为未访问过，以上一个量子比特的其他邻接的量子比特为出发点，继续进行访问。

图 6 为在图 5(a)所示的量子计算设备拓扑结构上以 Q_0 为控制位、 Q_3 为目标位时的近邻路径的具体形成过程：

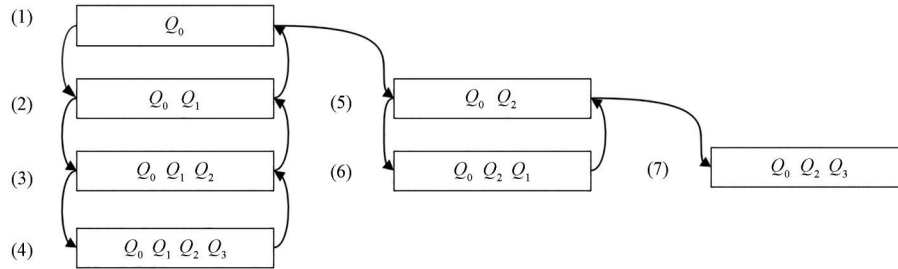


Fig.6 Formation process of neighboring paths

图 6 近邻路径的形成过程

第 1 步：将当前量子比特 Q_0 放入集合 $path$ 中，并标记 Q_0 被访问过，记为状态(1)；

第 2 步：能与 Q_0 直接相互作用的量子比特集合 $AdjList$ 有 Q_1 和 Q_2 ，随机选择 Q_1 作为当前量子比特，将量子比特 Q_1 放入集合 $path$ 中，并标记 Q_1 被访问过，记为状态(2)；

第 3 步：能与 Q_1 直接相互作用的量子比特为 Q_0 和 Q_2 ，因为 Q_0 被访问过，因此将量子比特 Q_2 放入集合 $path$ 中，并标记 Q_2 被访问过，记为状态(3)；

第 4 步：能与 Q_2 直接相互作用的量子比特集合 $AdjList$ 为 Q_0 、 Q_1 、 Q_3 ，因为 Q_0 和 Q_1 被访问过，因此将量子比特 Q_3 放入集合 $path$ 中，并标记 Q_3 被访问过，记为状态(4)；

第 5 步：因为 Q_3 为目标位，则将 $path$ 中的量子比特添加到 $Paths$ 中；

第 6 步：因为 Q_3 邻接的量子比特集合 $AdjList$ 包含的 Q_0 被访问过，按后进先出的原则，从 $Paths$ 中取出量子比特 Q_3 回到状态(3)，设置量子比特 Q_3 为未访问过状态；因为与 Q_2 邻接的量子比特 Q_0 和 Q_1 被访问过，因此继续回

退, 退到状态(2), 设置量子比特 Q_2 为未访问过状态; 再继续回退, 因与 Q_1 邻接的量子比特 Q_0 被访问过, 退到状态(1), 设置量子比特 Q_1 为未访问过状态; 因为 Q_0 的邻接量子比特 Q_2 之前未被访问过, 因此将 Q_2 放入集合 $path$ 中, 设置 Q_2 被访问过。以此类推, 得出控制位为 Q_0 、目标位为 Q_2 时, 近邻路径为 $Q_0-Q_1-Q_2-Q_3$ 和 $Q_0-Q_2-Q_3$ 。

2.2 基于估计成功概率(ESP)的保真度估算

本文使用 $ESP^{[9]}$ 作为 CNOT 门近邻方式的保真度衡量指标, 如式(1)所示。该式表示 CNOT(C, T) 门在图 7 所示的近邻路径 $a_1-a_2-\cdots-a_n$ 上的近邻代价, ESP 越大, 表示该 CNOT 门执行成功的概率越高, 因此尽可能选择使 ESP 值最大的近邻方式。

$$ESP[\pi(C)][\pi(T)] = \prod_{j \in (1, 2, \dots, n-1)} p_j^3 p_i \quad (1)$$

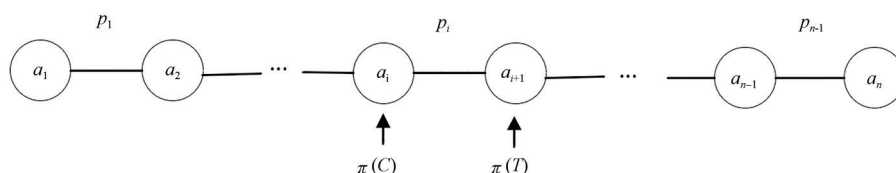


Fig.7 A neighbor path
图 7 近邻路径

图 7 中权值 p_i 表示 CNOT(a_i, a_{i+1}) 的执行成功概率, 因为线路中的 1 个 SWAP 可用 3 个 CNOT 替换, 因此用 p_{i-1}^3 表示插入 SWAP 的执行成功概率, 即 CNOT 的执行成功概率三次方; g 表示量子线路的 CNOT 门; $\pi(C)$ 表示将目标位移至物理量子比特 a_i 上; $\pi(T)$ 表示将控制位移至物理量子比特 a_{i+1} 上。

从整体量子线路角度看, 当前门的量子比特移动会影响后续每个门的近邻方式, 导致后续门近邻的 ESP 不一样。式(1)只考虑了当前门近邻后的 ESP , 并未考虑后续门。根据式(1), 选择使当前 CNOT 门 ESP 值最高的近邻方式, 不一定能提高后续 CNOT 门的 ESP , 因此整体量子线路的保真度不是太高。为解决上述问题, 本文使用滑动窗口, 该窗口内包含多个 CNOT 门。滑动窗口内的所有门存在多种可能的近邻方式, 为判断哪个近邻方式更有利于提高线路总体保真度, 根据式(2)的值选择最优的近邻方式。因此, 选择使式(2)最大的近邻方式将有利于提高窗口内所有门的计算成功率。

$$H_{\text{basic}} = \prod_{g \in W} ESP[\pi(C)][\pi(T)] \quad (2)$$

式中: W 代表滑动窗口; $ESP[\cdot][\cdot]$ 代表单个 CNOT 门的计算成功率, 如式(1)所示。

对滑动窗口内 CNOT 门进行近邻化变换时, 多个 CNOT 门按照时序依次进行处理, 且前面 CNOT 门的近邻化(插入 SWAP 门)方式不同, 会对后续 CNOT 门的近邻化造成影响。将这种连锁反应表达在树形结构上, 将该树称为近邻代价树。

通过一个示例解释式(2)计算过程。

图 8 为滑动窗口为 3 时生成的树形结构, 根结点代表初始状态, 根结点以下每一层中的结点代表滑动窗口中某个 CNOT 门的一种近邻方式。如, 层 Layer 1 中的结点 A 和 B 分别代表门 g_1 的 2 种近邻方式; 在 g_1 的特定近邻方式下, g_2 同样存在多种可能近邻方式, 这些近邻方式对应层 Layer 2 中的结点; 类似地, 层 Layer 3 中的结点表示 g_3 多种近邻方式。从根结点到每一个叶子结点的所有可能路径分别表示滑动窗口内所有门的一种近邻方式。按照从上往下的方式计算每条路径对应的式(2)值, 即先计算 g_1 的 ESP , 再计算 g_2 的 ESP , 最后计算 g_3 的 ESP 。显然, g_1 的多种可能近邻实现方式(对应 Layer 1 中的若干结点), 位于使式(2)值最大那条路径上的结点所代表的近邻方式更有利于提高滑动窗口所包含量子线路的整体保真度。

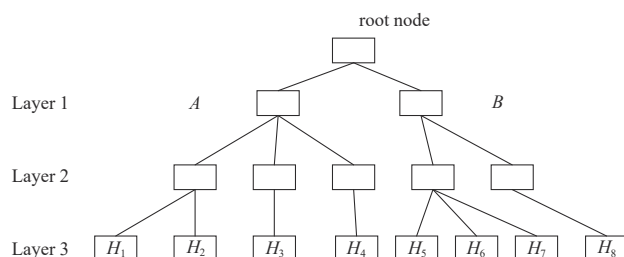


Fig.8 A nearest neighbor cost tree
图 8 近邻代价树

2.3 高保真度的量子线路变换策略

基于式(2)，本文提出一种量子线路变换策略，其主要思想为：首先，滑动窗口从量子逻辑线路的左边开始扫描，每次扫描固定数量的量子门，对于进入滑动窗口的所有量子门，进行以下操作：

第1步：取出滑动窗口的第1个CNOT门，基于近邻路径构建算法找到该CNOT门对应的所有近邻路径，进而得到所有的可能近邻实现方式，每个实现方式对应近邻代价树Layer 1中的一个结点，计算Layer 1中所有结点的ESP；

第2步：取出滑动窗口的第2个CNOT门，对Layer 1中所有结点进行扩展，得到Layer 2中的各结点，层Layer 2的各结点对应滑动窗口中第3个CNOT门的所有可能近邻方式，计算Layer 2中的所有结点的ESP；

第3步：类似第2步操作，一直取到滑动窗口的第K个门；

第4步：对层Layer k 中的所有结点计算ESP，并按照ESP进行排序，选择层Layer k 中ESP最高的前 B 个，使用 F_W 为滑动窗口中第1至第 K 个门的集合；

第5步：取出滑动窗口的第 $k+1$ 个CNOT门，对Layer k 中 B 个结点进行扩展，得到Layer($k+1$)中的各结点，Layer($k+1$)的各结点对应滑动窗口中第 $k+1$ 个CNOT门的所有可能近邻方式；

第6步：类似第2步操作，一直取到滑动窗口中第 $2K$ 个门；

第7步：类似第4步操作，选择层Layer $2k$ 中ESP最高的前 B 个，使用 E_W 为滑动窗口中第 k 至第 $2K$ 个门的集合；

第8步：取出滑动窗口的第 $2k+1$ 个CNOT门，对Layer $2k$ 中 B 个结点进行扩展，得到Layer($2k+1$)中的各结点，以此类推第5步至第7步，直到窗口中所有的CNOT门都实现近邻变换；

第9步：最终在窗口第一个门的多种可能实现方式中，选择能带来窗口内子线路整体ESP最高的那个近邻方式。

然后，从左向右移动窗口，将原来窗里中第1个门删除，若原窗口外最右边存在1个门，则将该门添加到窗口中，对于进入滑动窗口的所有量子门进行上述步骤操作。继续移动该窗口，直到量子逻辑线路中所有的门执行完，最后生成满足连通约束的量子线路。

```

algorithm2
input:the sliding window  $W$ , the path set between two qubits  $pathall$ 
output:the nearest neighbor method of the first gate in the sliding window  $NNCNOT$ 

1  Initialize the root  $RNode$ 
2   $CREATE\_TREE(F\_W, Node, Pathall)$ 
3   $LNodes \leftarrow FLN(RNode)$  /* find the leaf nodes */
4   $SNodes \leftarrow Sort(LNodes, B)$  /* sort the leaf nodes */
5   $depthcount \leftarrow C.size / W.size$ ;
6  for  $i \leftarrow 1$  to  $depthcount$  do
7       $LNodes \leftarrow \emptyset$  /* empty the leaf nodes */
8      for  $j \leftarrow 1$  to  $SNodes.size$  do
9           $Node \leftarrow SNodes[j]$ ;
10          $E\_W \leftarrow Node.leftqc$  /* retrieves the unexecuted gate stored in the leaf node */
11          $CREATE\_TREE(E\_W, Node, Pathall)$ ;
12          $LNodes \leftarrow FLN(Node)$ ;
13          $SNodes \leftarrow Sort(LNodes, B)$  /* sort the leaf nodes */
14   $NNCNOT \leftarrow Uplter(SNodes)$  /* get the nearest neighbor method of the first gate in the sliding window */

```

算法2中，变量 W 表示滑动窗口中所有门的集合， W 包含子集 F_W 和 E_W ；变量 $Node$ 表示近邻代价树上的结点；变量 $Layer$ 表示执行到 W 中的第几个门；函数 $CREATE_TREE$ 表示构建叶子结点；函数 FLN 表示寻找近邻路径树上的叶子结点；函数 $Sort$ 表示根据叶子结点中的ESP的值从大到小进行排序后，选出ESP最高的前 B 个结点；变量 $LNodes$ 表示树的叶子结点；变量 $SNodes$ 表示对应 B 个叶子结点的集合。第1行表示初始化根结点，根结点中ESP的值设置为1；第3行表示将最新层的叶子结点放入集合 $LNodes$ 中；第4行表示使用冒泡排序根据ESP的值从大到小重新排列集合 $LNodes$ 中的结点，选出若干较优叶子结点 $SNodes$ ；第8行表示以最新层结点 $SNodes$ 中ESP最高的前 B 个叶子结点为扩展，继续生成近邻代价树的结点。第14行表示 W 里面的门执行结束，从近邻代价树中最新层的叶子结点中选择ESP最大的结点，向上寻找到第1层的结点，得到滑动窗口第1个门的近邻方式。以下通过一个例子说明算法2的过程。

图9(a)是一个含有 g_1 、 g_2 、 g_3 、 g_4 、 g_5 、 g_6 共6个门的量子线路，图9(b)表示 W 的大小为5， F_W 为 g_1 、 g_2 、

g_3 , E_W 为 g_4 、 g_5 , 图9(c)表示滑动窗口为图9(b)所示时生成的树形结构, 根结点代表初始状态。图9(c)中, 设 B 为2, $H_2 > H_5 > H_1 > H_4 > H_5 > H_6 > H_7 > H_8 > H_3$ 时, 则选择结点 C 和 D 。在该两结点基础上, g_4 同样存在多种可能近邻方式, 这些近邻方式对应层 Layer 4 中的结点; 类似地, 层 Layer 5 中的结点表示 g_5 多种近邻方式。若 $G_4 > G_1 > G_2 > G_3 > G_5 > G_6$, 则从 E 结点向上查找得出结点 B , 即滑动窗口中第一个门的近邻方式。

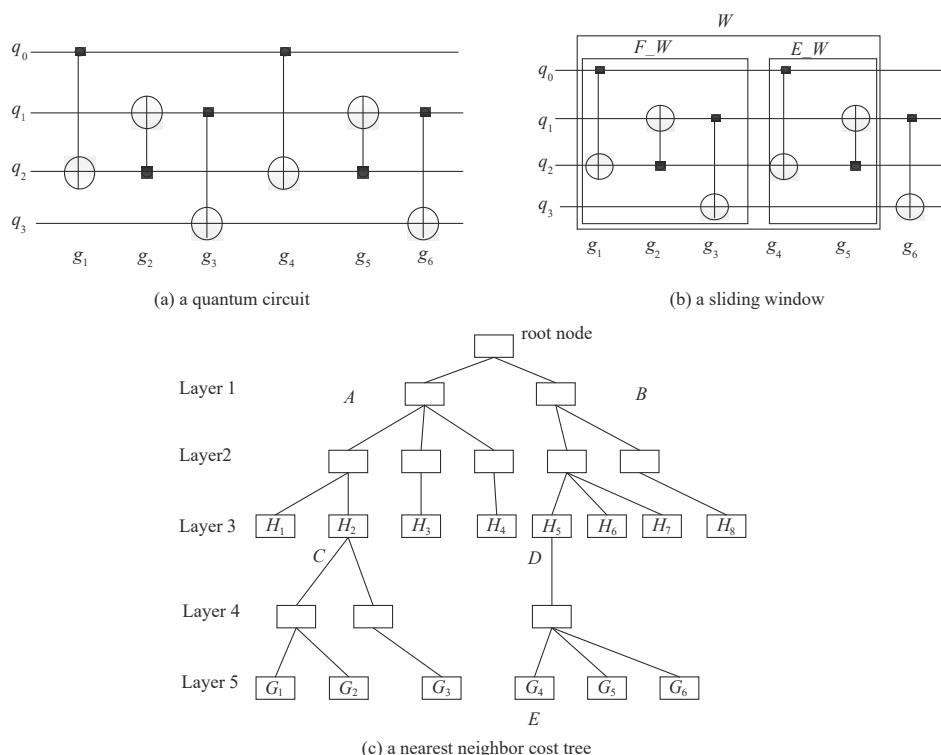


Fig.9 Selection of neighbor method of the first gate in the sliding window

图9 滑动窗口中第一个门近邻方式的选择

算法3中, 第2、3、4行表示获得2个量子比特之间的近邻路径, 第6行表示在窗口大小固定情况下, 根据逻辑量子线路中门的顺序, 从左向右移动窗口, 第7行表示获得滑动窗口中第1个门的近邻方式, 第8行表示获得满足连通约束的量子线路 C' 。以下通过一个示例描述算法3过程。

algorithm 3 quantum circuit transformation strategy

input: quantum circuit C , the sliding window W , topology diagram of quantum computing device $G(V, E)$

output: quantum circuit C' satisfying the connectivity constraint

```

1   $C' = \{\}, Pathall = \{\}$ 
2  for  $i \leftarrow 1$  to  $N$  do
3      for  $i \leftarrow 1$  to  $N$  do
4           $Pathall \leftarrow Pathall \cup GET\_PATH(G, N, N)$ 
5  for  $i \leftarrow 1$  to  $C.size$  do
6       $W \leftarrow C[i : i + W.size]$ 
7       $NNCNOT \leftarrow CREATE\_TREE(W, Node, Layer, Pathall)$ 
8       $C' \leftarrow C \cup NNCNOT$ 

```

图10为量子线路变换算法过程。图10(a)表示当滑动窗口中 W 为3, 第1个门为 g_2 时, W 为 g_2 、 g_3 、 g_4 , 门 g_2 的近邻方式根据对应滑动窗口内的ESP值决定; 当选好 g_2 门的近邻方式后, 继续移动窗口。当滑动窗口中第一个门为 g_3 , 类似得出门 g_3 的近邻方式, 如图10(b)所示。当量子逻辑线路中全部的门依此执行完毕, 得出满足连通约束的量子线路。

3 测试结果

使用基准测试用例(Benchmarks)^[11], 将本文所提策略与Qiskit工具包中SabreSwap和BasicSwap算法进行对比^[12], 所有实验均在IBMQ上量子计算设备的同一个校准周期内运行。

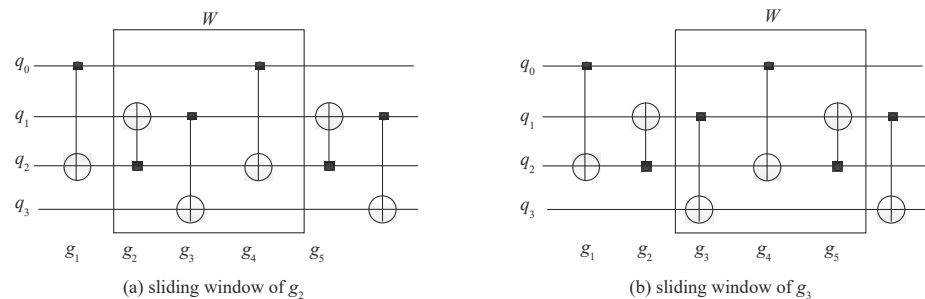


Fig.10 Sliding window
图 10 滑动窗口

评价方法：在五位量子计算设备 `ibmq_quito`、`ibmq_santiago` 上执行 5 组，在每组中将转换后的量子线路执行 1 000 次。评价指标：转换后的量子线路在 IBMQ 平台上的执行成功率，执行成功率是指转换后线路的正确结果次数在 1 000 次中所占的百分比。算法参数配置：所提策略 F_W 为 10， B 为 7， W 为 15，SabreSwap 和 BasicSwap 算法均采用默认参数。环境运行配置：`i7_9750 CPU`、`32 GB RAM`。

测试结果如表 1 所示。表 1 中， S_{ave} 、 S^l_{ave} 和 S^2_{ave} 分别表示本文所提策略、SabreSwap 和 BasicSwap 算法执行 5 组后的平均执行成功率，`avg` 表示本文所提策略与 SabreSwap 和 BasicSwap 算法相比，在量子线路保真度方面平均提高 65.67% 和 71.60%。由于噪声导致双量子比特门操作产生的错误是保真度下降的重要原因之一，本文所提策略在考虑噪声以及后续门的情况下移动量子比特，使用高保真操作，提高了量子线路的保真度。表 1 中有几个案例的保真度低于 SabreSwap 和 BasicSwap 算法，其主要原因是本文所提策略主要针对双量子比特门错误率，其他类型噪声因素，如单量子比特门、测量等也会对计算的保真度产生一定的影响。

表 1 保真度对比(%)
Table1 Fidelity comparison(%)

name	benchmarks	S_{ave}	S^l_{ave}	S^2_{ave}	$(S_{ave}-S^l_{ave})/S^l_{ave}$	$(S_{ave}-S^2_{ave})/S^2_{ave}$
ibmq_santiago	alu-v0_27.qasm	6.17	4.87	2.30	26.71	168.12
	alu-v1_28.qasm	5.50	3.03	1.67	81.32	230.00
	3_17_13.qasm	20.47	12.97	19.27	57.84	6.23
	4gt13_92.qasm	3.43	3.00	1.17	14.44	194.29
	decod24-v2_43.qasm	7.70	9.20	10.13	-16.30	-24.01
	mod5d2_64.qasm	4.83	3.90	4.47	23.93	8.21
	mod5mils_65.qasm	4.90	8.07	4.80	-39.26	2.08
ibmq_qutio	alu-v0_27.qasm	50.20	27.50	28.47	82.55	76.35
	alu-v1_28.qasm	45.77	17.53	24.40	161.03	87.57
	3_17_13.qasm	32.43	22.87	25.57	41.84	26.86
	4gt13_92.qasm	16.63	8.03	12.57	107.05	32.36
	decod24-v2_43.qasm	29.10	6.10	14.70	377.05	97.96
	mod5d2_64.qasm	16.97	18.15	13.03	-6.52	30.18
	mod5mils_65.qasm	14.47	13.43	8.70	7.69	66.28
avg					65.67	71.60

4 结论

量子计算设备上存在各种错误，一个典型的量子线路中包含大量的 CNOT 操作，并且 CNOT 操作的错误率比单量子比特操作的错误率高一个数量级，CNOT 错误率占据重要因素^[3]。本文基于量子比特之间链路在可靠性方面存在的差异，将更多的 CNOT 门映射到 CNOT 错误率低的量子比特上，从而提高量子线路的保真度。量子比特移动依赖控制位与目标位之间的路径，本文提出的近邻路径算法插入 SWAP 门。量子线路映射是 NP 完全问题，本文采取启发式方法；量子比特的移动会对后续门造成影响，在考虑后续门的情况下，选择当前门的近邻方式。实验结果显示，所提方法与 Qiskit 工具包中 BasicSwap 和 SabreSwap 算法相比，量子线路保真度有明显提高。IBMQ 作为典型的带噪声量子计算设备，为进一步提高保真度，还可以考虑其他噪声因素，如执行时间、单量子门的错误等；线路变换过程中除了插入 SWAP 门外，还可以考虑桥接门。虽然含噪声量子计算设备能够执行的量子线路门数和位数是有限的，但能够有效使用这些设备也是有意义的。太赫兹信息技术与量子信息技术同为信息技术的重要发展方向，两者同时又和新型半导体材料、光子器件、微纳系统等研究领域有大量的交叉。太赫兹技术的发展可与量子线路保真度提升技术相互促进，推动量子计算的发展。

参考文献:

- [1] 张蕾,傅海鹏,孟凡易,等. 硅基太赫兹技术及未来趋势[J]. 太赫兹科学与电子信息学报, 2021,19(5):753-768. (ZHANG Lei, FU Haipeng, MENG Fanyi, et al. Silicon-based terahertz technologies and future trends[J]. Journal of Terahertz Science and Electronic Information Technology, 2021,19(5):753-768.) doi:10.11805/TKYDA2021252.
- [2] 科普中国. 解决量子计算中退相干问题的新方法[EB/OL]. (2021-09-08)[2021-11-30]. <https://www.shkp.org.cn/articles/2021/09/if128394.html>. (China science communication. A new method for solving decoherence problems in quantum computing [EB/OL]. (2021-09-08)[2021-11-30]. <https://www.shkp.org.cn/articles/2021/09/if128394.html>.)
- [3] TANNU S S, QURESHI M K. Not all qubits are created equal: a case for variability-aware policies for NISQ-era quantum computers[C]// Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. Providence, RI, USA: Association for Computing Machinery, 2019: 987-999. doi: 10.1145/3297858.3304007.
- [4] HORSMAN D, FOWLER A G, DEVITT S, et al. Surface code quantum computing by lattice surgery[J]. New Journal of Physics, 2012,14(12):123011. doi:10.1088/1367-2630/14/12/123011.
- [5] SHAF AEI A, SAEEDI M, PEDRAM M. Optimization of quantum circuits for interaction distance in linear nearest neighbor architectures[C]// 2013 the 50th ACM/EDAC/IEEE Design Automation Conference(DAC). Austin, TX, USA: IEEE, 2013:1-6. doi: 10.1145/2463209.2488785.
- [6] SIRAICHI M Y, DOS SANTOS V F, COLLANGE C, et al. Qubit allocation[C]// Proceedings of the 2018 International Symposium on Code Generation and Optimization. Vienna, Austria: Association for Computing Machinery, 2018:113-125. doi:10.1145/3168822.
- [7] 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. doi:10.1109/TCAD.2018.2846658.
- [8] MURALI P, BAKER J M, JAVADI-ABHARI A, et al. Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers[C]// Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. Providence, RI, USA: Association for Computing Machinery, 2019:1015-1029. doi:10.1145/3297858.3304075.
- [9] NISHIO S, PAN Yulu, SATOH T, et al. Extracting success from IBM's 20-Qubit machines using error-aware compilation[J]. ACM Journal on Emerging Technologies in Computing Systems, 2020,16(3):1-25. doi:10.1145/3386162.
- [10] ZHU Pengcheng, GUAN Zhijin, CHENG Xueyun. A dynamic look-ahead heuristic for the Qubit mapping problem of NISQ computers[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020,39(12):4721-4735. doi: 10.1109/TCAD.2020.2970594.
- [11] IIC-JKU. ibm_qx_mapping-examples[EB/OL]. [2021-11-30]. https://github.com/iic-jku/ibm_qx_mapping/tree/master/examples.
- [12] ANON. IBMQ[EB/OL]. [2021-11-30]. <https://quantum-computing.ibm.com/>.

作者简介:

牛义仁(1996-), 男, 在读硕士研究生, 主要研究方向为计算机辅助量子逻辑综合. email: 1930320023@stmail.ntu.

管致锦(1962-), 男, 博士, 教授, 博士生导师, 主要研究方向为量子逻辑设计、安全计算和集成电路和系统的计算机辅助设计.

马天宏(2000-), 男, 在读本科生, 主要研究方向为计算机辅助量子逻辑综合.

金圣翔(1999-), 男, 在读本科生, 主要研究方向为计算机辅助量子逻辑综合.