

基于交换门的前瞻启发式量子线路映射算法



张辰逸, 尚涛*, 刘建伟

(北京航空航天大学网络空间安全学院 北京 海淀区 100083)

【摘要】含噪声中规模量子硬件的耦合约束使得大多数量子算法通过插入附加量子门改变量子位映射, 令量子算法直接在硬件上运行。为了降低量子线路的运行时间及提高量子线路的保真度, 设计了一种基于交换门的前瞻双向启发式映射算法。首先, 利用前瞻机制考虑前端层信息, 提高了附加门数结果的稳定性。其次, 设计搜索策略评估物理上近邻的候选交换门, 降低交换门搜索空间的复杂度。最后, 采用双向遍历全局考虑量子线路的门信息, 得到更高质量的初始映射。此外, 该算法适用于任意耦合量子硬件架构, 同时具有线路深度和附加门数的选择能力。实验结果表明, 相较于主流算法 A*-based 算法和 SABRE 算法, 该文提出的 SPBHA 算法可减少约 68% 与 34% 的附加门数, 线路执行时间缩短, 保证了量子程序结果的可靠性。

关键词 耦合约束; 映射; 前瞻双向; 量子计算; 量子线路

中图分类号 TP38; O413

文献标志码 A

doi:10.12178/1001-0548.20222339

SWAP-Based Prospective Heuristic Quantum Circuit Mapping Algorithm

ZHANG Chenyi, SHANG Tao*, and LIU Jianwei

(School of Cyber Science and Technology, Beihang University Haidian Beijing 100083)

Abstract The coupling constraint of noisy intermediate-scale quantum hardware makes most quantum algorithms change quantum bit mapping by inserting additional quantum gates, so that quantum algorithms run directly on the hardware. In order to reduce the quantum circuit running time and improve the quantum circuit fidelity, this paper designs a SWAP-based prospective heuristic quantum circuit mapping algorithm. First, the prospective mechanism is used to consider the front layer information, which improves the stability of the additional gate count results. Secondly, the search strategy is designed to evaluate the physically close candidate SWAP gates to reduce the search space complexity of SWAP gates. Finally, a bidirectional traversal is used to globally consider the gate information of quantum circuits to obtain higher quality initial mappings. The proposed algorithm is suitable for arbitrary coupled quantum hardware architecture, and has the ability to select circuit depth and additional gate number. The experimental results show that compared with the mainstream algorithms A*-based algorithm (Algorithm based on A* search) and SABRE algorithm (SWAP-based Bidirectional heuristic search algorithm), the SPBHA (SWAP-based Prospective Bidirectional Heuristic) algorithm proposed in this paper can reduce the number of additional gates by about 68% and 34%, shorten the circuit execution time, and ensure the reliability of the quantum program results.

Key words coupling constraint; mapping; prospective bidirectional; quantum computing; quantum circuit

量子计算强大的并行计算加速能力使其在量子模拟、整数分解、数据库搜索以及机器学习等领域具有巨大潜力^[1-4]。未来量子计算将进入有噪声的中规模量子计算机 (noisy intermediate-scale quantum computer, NISQ) 时代^[5]。谷歌、IBM、华为等公司

相继提供了 72、50、42 个量子位的量子计算模拟器^[6-7], 量子计算的加速潜力正在逐步实现。

由于 NISQ 时代技术限制, 只有在量子硬件耦合架构上允许相连的量子位上可以发生交互。通过量子线路映射算法, 量子程序才能高效地编译并运

收稿日期: 2022-09-30; 修回日期: 2022-10-15

基金项目: 国家自然科学基金 (61971021); 河北省重点研发计划 (22340701D); 航空科学基金 (2018ZC51016)

作者简介: 张辰逸 (2000-), 博士生, 主要从事网络安全、量子计算方面的研究。

*通信作者: 尚涛, E-mail: shangtao@buaa.edu.cn

行在量子硬件上。在量子程序编译的过程中如何高效地进行量子线路映射至关重要。研究成果表明,优秀的量子线路映射算法^[8]可以将量子线路中的附加量子门数降低 80%,甚至无需附加量子门,缩短量子线路运行时间,保证运行结果的可靠性。

量子线路映射问题已经被证明是 NP 完全问题^[9],在量子线路规模日益增大、架构日益复杂的今天,更为灵活、扩展性更佳的启发式算法是量子线路映射算法的主要研究方向^[10]。目前,国内外学者对量子线路启发式映射算法已开展了许多研究。文献 [11] 令初始映射优先满足物理上相邻的 CNOT 门,每次量子位移动,只解析一个双量子位门,根据它们之间的双量子位门数量决定是否移动量子位。虽然算法速度很快,但过于简单,效果一般。文献 [12] 尝试将 A*-based (algorithm based on A* search) 算法引入启发式成本函数,将双量子位门划分为独立的层,最小化层中耦合量子位之间的距离之和,同时减少最终输出线路的深度。不足在于搜索所有可能的并发 SWAP 门组合仍然需要指数时间,初始映射只由线路开始时的两个量子位门决定,不考虑全局性。文献 [8] 提出了 SABRE (SWAP-based Bidirectional heuristic search algorithm) 算法,对量子映射进行更加精准的建模,通过优化每次搜索尝试,将指数复杂度的搜索空间缩小至线性复杂度,但存在附加门数不稳定的问题。文献 [13] 使用动态分配并回收量子位来对占用的计算资源进行优化。

现有的量子线路映射算法相关研究主要关注量子位映射的中间变换策略,对量子位映射的初始化方式关注较少。研究表明初始量子位映射对量子线路映射算法的结果有很大影响^[8],在中间映射变换之前利用已知量子线路信息提前对量子位映射初始化是一种行之有效的优化方式。本文致力于设计高效的量子线路映射算法,利用前端层信息初始化量子位映射,采用耦合图节点近邻化代替全局搜索,有助于降低量子程序执行时间,提高量子线路保真度。

1 量子线路映射问题

由于量子位具有连接脆弱、串扰和状态不稳定等特性^[14],因而量子线路运行在实际量子硬件中可能会遇到以下 3 种错误类型:相干时间^[15]、量子位保真度^[16]与量子位耦合约束^[17]。相干时间与量子位保真度需要量子硬件的技术进步来改善,量子位的耦合约束则可以通过软件层面的量子线路映射算法

解决。

结合一个小规模量子线路示例进一步解释量子线路映射问题,如图 1 和图 2 所示。4 量子位硬件模的硬件架构采用图 2a 所示的硬件耦合图。该硬件模型允许在以下量子位对上使用双量子位门: $\{Q_1, Q_2\}, \{Q_2, Q_4\}, \{Q_4, Q_3\}, \{Q_3, Q_1\}$, 不允许在 $\{Q_1, Q_4\}$ 和 $\{Q_2, Q_3\}$ 上使用双量子位门。假设有一个小规模量子线路要在图 2 所示的 4 量子位设备上执行。该量子线路由 6 个 CNOT 门组成(如图 2b 所示),假设初始的逻辑量子位到物理量子线路映射为 $\{q_1 \rightarrow Q_1, q_2 \rightarrow Q_2, q_3 \rightarrow Q_3, q_4 \rightarrow Q_4\}$ 。

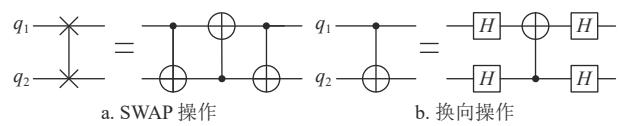


图 1 更改映射关系的量子操作

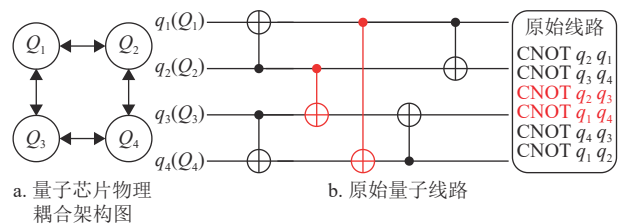


图 2 更新前的量子线路及其硬件架构

由图 2a 可见,6 个 CNOT 门中有 4 个可以直接执行,第 3 个和第 4 个 CNOT 门无法执行,因为相应的量子位对没有连接到硬件上。图 2 的量子线路示例中不存在满足所有双量子位门依赖关系的完美初始映射,需要在程序执行期间更改量子位映射,使所有 CNOT 门都可执行。

主流的更改映射方法为使用 SWAP 门(即交换门),通过在两个量子位之间交换状态来改变量子线路映射(如图 1a 所示)。SWAP 门由 3 个 CNOT 门组成。如果两个量子位在量子器件上并不相邻,可以使用多个 SWAP 门将 1 个逻辑量子位移动到任意物理量子位的位置,然后在线路中应用双量子位门。如图 3 所示,在第 2 个 CNOT 门之后的 q_1 和 q_2 之间插入 1 个 SWAP 门,更新后量子线路是可以执行的。前两个 CNOT 门可以在初始映射下执行。插入 SWAP 门后,映射更新为 $\{q_1 \rightarrow Q_2, q_2 \rightarrow Q_1, q_3 \rightarrow Q_3, q_4 \rightarrow Q_4\}$,所有剩余的 4 个 CNOT 门都可在此映射下执行。

当量子线路应用在非对称硬件模型时,即只允许运行方向符合连通关系的双量子位门时,可使用

基变换技术, 添加 4 个 Hadamard 门, 将 $|01\rangle$ 基转换为 $|+-\rangle$, 从而实现 CNOT 门换向操作, 如图 1b 所示。

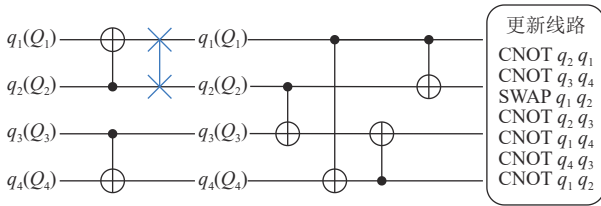


图 3 更新硬件映射后的量子线路

在量子线路中引入额外的 SWAP 门可以解决所有双量子位门的依赖性, 并不改变原有功能的情况下生成硬件兼容线路。由于 NISQ 硬件设备的限制, 在量子线路中插入 SWAP 门会导致以下问题。1) 量子线路中的操作次数增加, 由于操作不完美, 会引入噪声, 导致总体错误率会增加。2) 量子线路深度会增加, 导致执行时间将增加, 且量子线路的退相干可能会累积许多错误。

比较图 2b 和图 3 中的原始线路和更新线路, 量子门的数量从 6 增加到 9, 线路深度从 5 增加到 8。额外的 SWAP 门将在保真度和执行时间方面带来巨大的开销, 需要量子线路映射算法来尽量减少额外 SWAP 门的数量, 降低总体错误率和总执行时间。

2 SPBHA 算法

本文在量子线路映射算法 SABRE 算法^[8]的基础上, 保留 SABRE 算法双向遍历和就近搜索候选 SWAP 门空间的特点, 使用前端层信息初始化量子线路映射代替随机初始化量子位映射, 并对遍历流程和启发式代价函数进行改进, 设计了基于交换门的前瞻双向启发式算法 (SWAP-based prospective bidirectional heuristic algorithm, SPBHA)。SPBHA 算法包含 4 个部分: 预处理、单次遍历、双向遍历更新映射与启发式代价函数, 表 1 定义了 SPBHA 算法使用的数学符号。

2.1 预处理

SPBHA 算法预处理的具体步骤如下。

计算距离矩阵: 给定量子器件的耦合图, 通过 Floyd-Warshall^[18] 算法计算所有点对最短路径 (all pairs shortest path, APSP), 以获得距离矩阵 D , 耦合图每条边距离为 1。 $D_{[i][j]}$ 表示将逻辑量子位从物理量子位 Q_i 移动到 Q_j 所需的最小 SWAP 数。

表 1 SPBHA 算法数学符号定义

数学符号	定义
n	逻辑量子位数
$q_{(1,2,\dots,n)}$	量子线路中的逻辑量子位
$Q_{(1,2,\dots,n)}$	量子设备中的物理量子位
d	量子线路的深度
g	量子线路的量子门序号
N	物理量子位数
$G(V,E)$	量子硬件的耦合图
D	物理量子位的距离矩阵, $D_{[i][j]}$ 表示 Q_i 到 Q_j 的距离
π	$q_{(1,2,\dots,n)}$ 到 $Q_{(1,2,\dots,n)}$ 的映射关系
π^{-1}	$Q_{(1,2,\dots,n)}$ 到 $q_{(1,2,\dots,n)}$ 的映射关系
F	前端层
E	扩展集

生成量子线路有向无环图 (directed acyclic graph, DAG): 使用 DAG 表示量子线路中受控非门 CNOT 门的依赖顺序。单量子位门只在一个量子位上执行, 不会对其他量子位产生依赖, 因而量子线路映射算法不考虑单量子位门。当 CNOT 门的两个量子位 q_i 或 q_j 上所有之前的 CNOT 门都已执行时, 才能执行 $CNOT(q_i, q_j)$ 。遍历整个量子线路, 并构造一个 DAG。图 4b 的 DAG 是由图 4a 的量子线路生成的。如由于量子位 q_2 同时存在于量子门 g_1 和量子门 g_3 中, 而在 g_1 之前不能执行 g_3 , 所以量子门 g_3 依赖于量子门 g_1 。以此类推, 根据量子门之间的依赖关系生成 DAG。

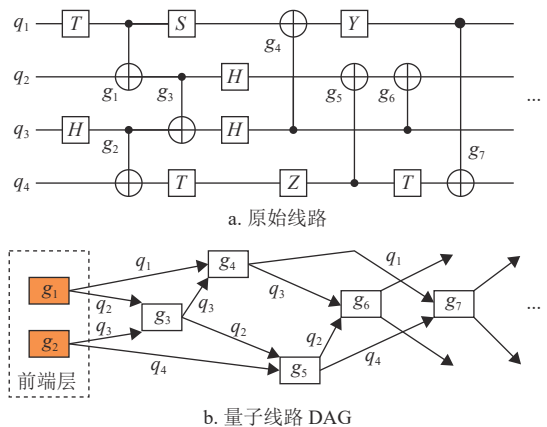


图 4 原始线路生成 DAG 的示例以及前端层初始化

初始化前端层: 前端层 (F) 被定义为 DAG 中没有未执行前导的所有 CNOT 门的集合, 意味着对应的双量子位门没有依赖性, 可以立即在硬件上执行。对于 $CNOT(q_i, q_j)$, 当 CNOT 门的两个量子位 q_i 或 q_j 上所有之前的量子门都被执行时, 它可以被放置在集合 F 中。遍历第二步得到 DAG, 并找

到图中入度为 0 的所有顶点 (量子门), 将这些量子门放入 F , 以初始化 F 。图 4 中, 初始化的前端层包含 g_1 和 g_2 , 因为它们没有前导。

利用前瞻机制初始化量子位映射: SPBHA 算法使用前端层量子门的信息, 在进行单次遍历算法之前利用前瞻机制对量子位映射进行初始化。图 5c 线路使用图 5a 量子硬件架构, 并根据定义 (见表 1) 划分出前端层、前端层的后继门组成的短期层和远离前端层的长期层。在前端层的 CNOT(q_1, q_3) 中, 将 q_1 映射到 Q_1 , 并根据 Q_1 相邻的物理量子位情况, 将 q_3 映射到 Q_2 , 使得 q_1 和 q_3 的物理量子位相邻, 该 CNOT 门无需插入任何 SWAP 门可以直接在硬件上运行。同理, CNOT(q_2, q_6) 中, q_6 映射到 Q_6 , q_2 映射到 Q_3 。在满足前端层的量子门均可以直接在硬件上运行后, 剩余未映射的量子位进行随机分配。图 5b 中虚线框的映射为利用前瞻信息初始化的映射, 实线框映射为随机初始化映射。由于量子程序具有与经典程序类似的局部性原理, 作用在相同量子位对上的量子门可能在短时间内再次运行。利用局部性原理, SPBHA 算法可以进一步减少附加门的数量, 提高算法效率。由于前端层的量子门之间没有依赖性, 因此必然可以找到一个使得满足前端层量子门均可以直接在硬件上运行的初始量子位映射。

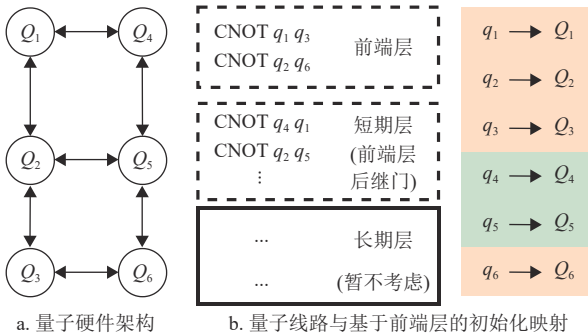


图 5 利用前瞻机制初始化量子位映射

2.2 单次遍历

算法 1 单次遍历算法

Input: 距离矩阵 D 、线路 DAG、初始 F 和初始映射 π

Output: 更新的量子映射 π_u , 插入 SWAP 门的信息

while $F \neq \emptyset$ do

 Execute_gate_list $\leftarrow \emptyset$ /*初始化可执行门集合*/;

```

for gate  $\in F$  do
    if 量子门可直接在硬件上运行 then
        Execute_gate_list.add(gate) /*添加可执行门*/;
    end
end
if Execute_gate_list  $\neq \emptyset$  then
    for gate  $\in$  Execute_gate_list do
        F.remove(gate);
        从 DAG 中获得 F 的后继门 successor_gate;
        F.add(successor_gate); /*添加后继门*/;
    end
end
else
    score  $\leftarrow \emptyset$  /*初始化评分数组*/;
    SWAP_Candidate_list  $\leftarrow$  GetSWAP(F,G)
    /*根据前端层和耦合图得到候选插入的 SWAP 门*/;
    for SWAP  $\in$  SWAP_Candidate_list do
         $\pi_{temp} \leftarrow \pi$ .update(SWAP) /*用当前 SWAP 门暂时更新映射*/;
        score[SWAP]  $\leftarrow H(\pi_{temp}, F, D, DAG, SWAP)$ ; /*根据启发式代价函数获得 SWAP 门的评分*/;
        F.add(successor_gate);
    end
    找到最佳评分的 SWAP 门;
     $\pi \leftarrow \pi$ .update(SWAP) /*用当前 SWAP 门更新映射*/;
end
end

```

算法 1 描述了 SPBHA 的单次遍历算法流程, 算法扫描整个 DAG 并插入 SWAP 门, 以使所有 CNOT 门都可执行。

在单次遍历算法的每次迭代中, 检查 F 中是否有任何可以直接在硬件上执行的量子门。若存在可直接执行的量子门, 算法将执行这些量子门并将其从 F 中移除, 然后向 F 添加新的后继门。如果 F 中所有量子门都不能直接在硬件上执行, 算法将尝试搜索可能插入的 SWAP 门集合并进行评估, 在量子线路中插入当前最适合的 SWAP 门更新映射。该算法将不断迭代, 当前端层 F 为空时, 说明量子线路中的所有量子门都已执行, 算法停止。

2.3 双向遍历更新映射

双向遍历的过程如图6所示, SPBHA 算法进行预处理初始化量子位映射, 并使用单次遍历算法正向遍历原始线路。从正向遍历中获得的量子位映射将作为反向遍历中的初始映射。利用量子计算的可逆性, 原始线路可生成对应的反向线路。反向线路中的两个量子位门完全相同, 只有顺序相反。之后, 再次使用单次遍历算法遍历反向量子线路, 最终将经过反向遍历得到的映射作为最终映射。

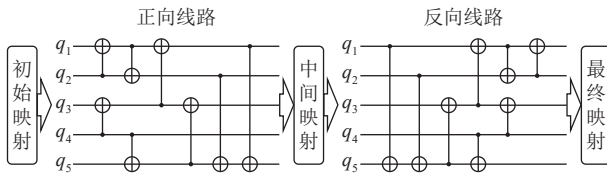


图6 双向遍历更新量子位映射

使用双向遍历的方式获得量子位映射可以全局考虑量子线路上所有量子位门的信息。由于在同一量子硬件模型上正反向运行同一线路, 考虑了所有的量子位门信息, 更新后的初始映射较单次遍历的初始映射使用的附加门更少。尽管位于量子线路中间的量子门对映射的影响小于两端的量子门, 双向遍历对减少附加门数的效果依然明显。

2.4 启发式代价函数

启发式成本函数是 SPBHA 算法的核心, 用来评估每个候选插入的 SWAP 门的分数, 从而帮助 SPBHA 算法找到当前线路最适合插入的 SWAP 门。

基于最近邻代价 (nearest neighbor cost, NNC) 的启发式代价函数是最普遍与基本的启发式代价函数^[19]。评估时暂时改变量子位映射 π , 然后对 F 中所有量子位对之间距离求和 (量子位间的距离可通过预处理阶段计算的矩阵 D 求得) 得到启发式代价函数的函数值。若函数值很小, 意味着 F 中两个量子位对之间的距离很短, 此 SWAP 门更可能使 F 中的量子位可执行, 最终将选择具有最小函数值的 SWAP 门改变映射。基于 NNC 的启发式代价函数的形式为:

$$H_{\text{basic}} = \sum_{\text{gate} \in F} D[\pi(\text{gate}.q_1)][\pi(\text{gate}.q_2)] \quad (1)$$

SABRE 算法的启发式代价函数在式 (1) 的基础上考虑后续的量子门, 引入了扩展集 E , 它包含了 F 中部分门的后继门。SABRE 算法用 F 和 E 的大小来规范化 F 与 E 的总和, 添加权重参数 $W(0 \leq W < 1)$ 以减少 E 的影响。为了选择可以并行执行的 SWAP

门, 在函数中引入了衰减效应。如果一个量子位 q_i 最近参与了 SWAP 门, 那么它的衰变参数将增加 $\delta(\text{decay}(q_i) = 1 + \delta)$ 。该衰减参数将使启发式搜索倾向于选择不重叠的 SWAP 门, 增加生成线路中的并行性。SABRE 算法的启发式代价函数形式为:

$$H_{\text{SABRE}} = \max(\text{decay}(\text{SWAP}.q_1), \text{decay}(\text{SWAP}.q_2)) \times \left\{ \frac{1}{|F|} \sum_{\text{gate} \in F} D[\pi(\text{gate}.q_1)][\pi(\text{gate}.q_2)] + W \frac{1}{|E|} \sum_{\text{gate} \in E} D[\pi(\text{gate}.q_1)][\pi(\text{gate}.q_2)] \right\} \quad (2)$$

然而, 式 (2) 有可以继续改进的地方。式 (2) 只将部分 F 中量子门的后继门放入扩展集 E , 以体现前瞻能力的灵活性, 实际上 E 设置为灵活大小并无必要, 通常来说考虑 F 的全部后继门将更充分地考虑后续量子门的位置信息, 从而插入更高效的 SWAP 门。因此 SPBHA 算法的启发式函数改为将 F 中所有量子门的后继门放入 E 中, 得到具有最大前瞻能力的扩展集 E_{max} 。

在 SPBHA 算法的启发式代价函数中, 衰减函数定义为:

$$\text{decay}(q_i) = \begin{cases} 1 + n\delta - m\varepsilon & \text{decay} \geq 1 \\ 1 & \text{decay} < 1 \end{cases} \quad (3)$$

式中, n 为 q_i 参与的 SWAP 门数; m 为当前已执行门数; δ 为衰变参数; ε 为执行参数, 且 $\delta = 4\varepsilon$ 。式 (3) 的设计目的是将衰变函数变得更加连续, 相较于式 (2) 采用每 5 个搜索步骤或在执行 CNOT 门后重置函数, 式 (3) 不重置函数, 转而全局考虑 q_i 参与过的所有 SWAP 门, 结合已执行量子门数确定衰变函数值。当衰变函数值小于 1 时, 说明最近插入的 SWAP 门距离当前量子门过远, 因此衰变函数值设为 1, 不影响当前的并行选择, 而且能够综合考虑量子位参与过的全部 SWAP 门以及到当前门的距离。

此外, 将取衰变函数中最大值改为求和, 从而同时考虑 q_i 和 q_j 的情况。最终 SPBHA 算法的启发式代价函数形式为:

$$H = (\text{decay}(\text{SWAP}.q_1) + \text{decay}(\text{SWAP}.q_2)) \times \left\{ \frac{1}{|F|} \sum_{\text{gate} \in F} D[\pi(\text{gate}.q_1)][\pi(\text{gate}.q_2)] + W \frac{1}{|E_{\text{max}}|} \sum_{\text{gate} \in E_{\text{max}}} D[\pi(\text{gate}.q_1)][\pi(\text{gate}.q_2)] \right\} \quad (4)$$

2.5 算法分析

在量子线路映射问题中, 量子位映射 π 具有指数级可能性: $\frac{N!}{(N-n)!}$ 。因此, 精确求解最优解并不可行, 往往使用启发式方法和其他优化方式降低计算复杂度并得到次优解。现有启发式算法^[12]使用穷举搜索, 每个 CNOT 的候选 SWAP 门搜索空间与时间复杂度是 $O(e^N)$, 其中包括冗余的搜索空间, 可以继续改进。设 q_i 是 F 中 CNOT 门的一个逻辑量子位, SPBHA 算法不搜索全部可能的逻辑量子位, 而是通过量子位映射 π 和耦合图 G 得到 j 个近邻的物理量子位:

$$Q_{i_1}, Q_{i_2}, \dots, Q_{i_j} = G(\pi(q_i)) \quad (5)$$

通过逆映射 π^{-1} 获得 q_i 在硬件上近邻的全部逻辑量子位:

$$q_{i_1}, q_{i_2}, \dots, q_{i_j} = \pi^{-1}(Q_{i_1}), \pi^{-1}(Q_{i_2}), \dots, \pi^{-1}(Q_{i_j})$$

将 $q_{i_1}, q_{i_2}, \dots, q_{i_j}$ 与 q_i 构成候选 SWAP 门的量子位对, 通过启发式代价函数评估候选 SWAP 门, 得到当前最优的 SWAP 门并更改映射:

$$\min\{H[\text{SWAP}(q_i, q_{i_1}), \text{SWAP}(q_i, q_{i_2}), \dots, \text{SWAP}(q_i, q_{i_j})]\}$$

式 (5) 不考虑远离前端层的长期层中的量子门 (见图 4b), 因为 π 在执行期间可能显著变化, 在无法穷举搜索的情况下很难准确地估计当前映射对长期层的影响。由于 SPBHA 算法只搜索能帮助解决前端层中依赖关系的候选 SWAP 门 (即物理上近邻的逻辑量子位), 因此搜索候选 SWAP 门的时空复杂度由 $O(e^N)$ 降至 $O(N)$, 大幅提升了算法执行速度。

SPBHA 算法在最坏情况下, 每个双量子位门都需要改变映射满足耦合约束。为方便分析, 设 CNOT 门数量为 m ; CNOT 门的搜索空间中评估一个候选 SWAP 门的时间为 t ; 最大可能的 SWAP 门搜索空间大小为 s ; 每个 CNOT 门的最大搜索步数为 k 。

SPBHA 算法预处理阶段的 4 个步骤中, Floyd 算法时间复杂度是 $O(N^3)$, 生成量子线路 DAG 的时间复杂度为 $O(m)$, 初始化前端层即在量子线路 DAG 中搜索入度为零的顶点, 时间复杂度是 $O(m)$ 。利用前瞻机制初始化量子位映射的时间复杂度为 $O(N)$ (最坏情况下前端层涉及所有物理量子位), 因此预处理阶段的时间复杂度可表示为 $T_1(N, m) = O(N^3 + m)$ 。

SPBHA 算法中的单次遍历算法的时间复杂度

可表示为 $t \times s \times k \times m$, 根据式 (4), 最坏情况下所有量子位都出现在前端层和扩展集, 则 t 的时间复杂度为 $O(N)$ 。最坏情况下, CNOT 门的两个逻辑量子位在物理上与其余全部的量子位近邻, 故最大可能 SWAP 门搜索空间大小 s 为 $O(N)$ 。单次遍历算法至多需要量子芯片耦合图直径数量的 SWAP 门为每个 CNOT 门移动两个量子位, 对于 2D 网格布局来说最大搜索步数 k 为 $O(\sqrt{N})$ 。因此, 单次遍历算法的时间复杂度 $T_2(N, m) = O(N^{2.5}m)$ 。

SPBHA 算法的总时间复杂度为:

$$T(N, m) = T_1(N, m) + T_2(N, m) = \max(O(N^3 + m), O(N^{2.5}m)) = O(N^{2.5}m)$$

主要计算开销来自 3 次运行单次遍历算法, 而对于运行在 NISQ 设备上的量子程序, 当量子位数处于 100 以内, 量子线路中 CNOT 门数在 10 000 以内时, 时间复杂度是可以接受的。

3 实验分析

量子线路运行平台使用 IBM Quantum Experience 平台, 测试数据选自 IBM 的 QisKit 中的量子线路、RevLib^[20] 中的函数以及 Quipper^[21] 和 Scaffold^[22] 中编译的算法。实验在英特尔 Core i7-8750H 2.2 GHz 和 16 GB 内存的计算机上进行, 操作系统为 Windows 10。

所有实验均使用 IBM Q20 Tokyo 芯片的耦合图。该芯片为对称量子硬件模型, 每对相连的物理量子位之间的两个方向上都允许使用 CNOT 门。实验进行了 A*-based 算法、SABRE 算法和 SPBHA 算法的对比分析。由于 SPBHA 算法和 SABRE 均存在随机初始化量子线路映射过程, 所以结果存在波动。实验将每种算法运行 5 次, 取平均结果。

SPBHA 算法经过多次测试, 最终选取效果最佳的一组参数, 具体如下: 权重参数 $W = 0.5$ 固定不变, 执行参数 $\varepsilon = 0.0005$, 衰变参数 $\delta = 0.002$, 每次增加 0.001。SABRE 算法参数选择如下: 权重参数 $W = 0.5$ 固定不变, 衰减参数 $\delta = 0.001$ 。

3.1 附加门数

表 2 列出了 A*-based 算法、SABRE 算法和 SPBHA 算法在不同规模量子线路下的附加门添加数量。由表 2 可知 SPBHA 算法减少附加门数的效果远好于 A*-based 算法。小规模量子线路 (量子规模小于 1 000) 的附加门数降低百分比可以达到

76%, 中规模量子线路为 71%, 大规模量子线路则为 57%。部分测试中 SPBHA 算法可通过双向遍历

找到完美初始映射, 无需额外附加门 (如 `ising_model_10` 等)。

表 2 在不同规模量子线路下的附加量子门数实验结果

量子线路名称	量子位数/bit	原始量子门数	A*-based算法量子门数	SABRE算法量子门数	SPBHA算法量子门数
4mod5-v1_22	5	21	3	3	0
mod5mils_65	5	35	18	6	0
alu-v0_27	5	36	27	0	3
decod24-v2_43	4	52	18	3	3
4gt13_92	5	66	30	3	3
mini_alu_305	10	173	69	42	36
qft_10	10	200	60	54	42
rd84_142	15	343	93	105	51
qft_13	13	412	156	93	57
ising_model_10	10	480	30	0	0
cnt3-5_180	16	485	220	166	90
qft_16	16	512	171	186	45
ising_model_13	13	633	30	0	0
ising_model_16	16	786	108	0	0
wim_266	11	986	390	237	136
cm152a_212	12	1 221	462	232	0
cm42a_207	14	1 776	870	453	300
adr4_197	13	3 439	1 212	780	771
misex1_241	15	4 813	1 869	1 521	69
cycle10_2_110	12	6 050	1 911	912	6
square_root_7	15	7 630	1 953	1 701	1 419
sqn_258	10	10 223	3 990	2 799	2 079
rd84_253	12	13 658	4 656	1 614	96
co14_215	15	17 936	6 357	892	231
max46_240	10	27 126	9 597	7 938	7 305
9symml_195	11	34 881	10 053	8 892	8 037

A*-based 算法的附加门数多于 SPBHA 算法的原因是 A*-based 算法只考虑量子线路开头的两个量子位门, 没有继续改进初始映射。对于 `ising_model` 基准, 最优解的意义不大, 因为量子力学中的 `ising_model` 只考虑附近的耦合能量。SPBHA 算法和 SABRE 算法都可以找到完美初始映射, 而 A*-based 不行, 这从一个侧面反映了算法的性能。

与 SABRE 算法相比, SPBHA 算法在小规模量子线路的附加门数降低百分比可以达到 38%, 中规模量子线路为 48%, 大规模量子线路则为 16%。上述两个算法都有随机因素, 因此进行 5 次实验取平均结果可以一定程度上反映算法的真实情况。如表 2 所示, 在大多数测试中, SPBHA 算法都有更低的附加门数量, 仅在少数测试时略高于 SABRE 算法, 其中一些测试结果显著低于 SABRE 算法。SPBHA 算法的附加门数少于 SABRE 算法的原因

是表 2 的部分测试中存在大量运行在相同量子位上的 CNOT 门。根据局部性原理, SPBHA 算法具有满足前端层 CNOT 门耦合的初始映射, 在部分测试上 (如 `cycle10_2_110`) 可大幅减少附加门数, 符合算法分析。

3.2 遍历方式

SPBHA 算法使用了双向遍历, 即正向遍历线路得到初步的量子位初始映射, 利用量子线路的可逆性反转线路并将正向遍历得到的映射作为输入反向遍历线路, 最终得到更新的初始映射。

表 3 列出了使用不同遍历方式的 SPBHA 算法附加门数的影响。实验结果显示双向遍历得到的量子位映射相较单次遍历普遍具有更好的性能, 更新的初始映射使得算法添加了更少的附加门。随着量子线路规模增大, 双向遍历与单次遍历的附加门数差距逐渐减小, 最差情况可以减少 7% 的 CNOT 门数。原因是量子线路规模增大, 两端量子门的情况

不足以代表全部量子门的情况，因此对附加门数的影响减小，较单次遍历的降幅不够明显。使用双向

遍历方式的算法在附加门数上总体少于单次遍历方式的附加门数，与算法分析是一致的。

表 3 使用不同遍历方式的 SPBHA 算法实验结果

量子线路名称	量子位数/bit	原始量子门数	单次遍历量子门数	双向遍历量子门数
4mod5-v1_22	5	21	3	0
mod5mils_65	5	35	12	0
alu-v0_27	5	36	16	3
decod24-v2_43	4	52	9	3
4gt13_92	5	66	15	3
mini_alu_305	10	173	63	36
qft_10	10	200	60	42
rd84_142	15	343	66	51
qft_13	13	412	102	57
ising_model_10	10	480	9	0
cnt3-5_180	16	485	123	90
qft_16	16	512	83	45
ising_model_13	13	633	6	0
ising_model_16	16	786	21	0
wim_266	11	986	198	136
cm152a_212	12	1 221	6	0
cm42a_207	14	1 776	354	300
adr4_197	13	3 439	886	771
misex1_241	15	4 813	93	69
cycle10_2_110	12	6 050	15	6
square_root_7	15	7 630	1 731	1 419
sqn_258	10	10 223	2 814	2 079
rd84_253	12	13 658	126	96
co14_215	15	17 936	261	231
max46_240	10	27 126	8 240	7 305

3.3 量子线路深度

图 7 展示了 6 组测试在不同衰变参数 δ 值下产生的量子线路深度变化。每组测试的数据分别在 $\delta = 0.002$, $\delta = 0.004$, $\delta = 0.006$ 的情况得到。

由于映射算法的改变，SABRE 算法和 SPBHA 算法的选择能力难以比较。如图 7 所示，SPBHA 算法可通过改变附加门的数量使得生成线路深度变化平均达到 10.6%。根据文献 [8] 的实验，SABRE 算法可提供约 8% 的生成量子线路深度变化，从量子线路深度变化的指标对比，可以反映出 SPBHA 算法可以根据量子位相干时间和门保真度数据来改变 δ 的值，因而具有相对更好的深度与量子门数的选择能力。

实验结果表明，如果继续增大 δ 值，量子线路深度和门的数量都可能增加，原因是 δ 过大导致启发式代价函数将过多地考虑未移动的量子位，而不是去满足当前与后继 CNOT 门的依赖性，这将带来冗余的附加门数，提高算法执行时间。

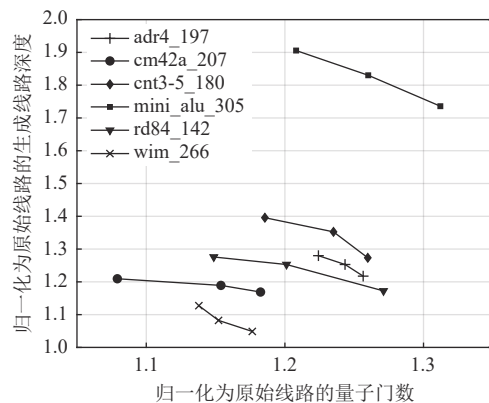


图 7 量子线路深度与附加门数的选择

4 结束语

本文提出了基于交换门的前瞻双向启发式算法，实现了量子线路的高效映射。为减少附加门的使用，降低算法复杂度，SPBHA 算法利用前瞻的量子门信息进行量子位映射的初始化，采用就近寻

找候选 SWAP 门的原则大幅缩小了交换门的搜索空间。为进一步优化量子位映射, SPBHA 算法考虑全局量子门信息, 利用量子计算的可逆性进行双向遍历。通过实验证明了 SPBHA 算法的可行性, 为量子线路映射算法的设计提供了新思路。在量子编译过程中, 量子线路映射与调度往往紧密相关, 从时序与跨平台调度的角度优化用户的量子程序是未来量子线路映射算法一个可行的研究方向。此外, 量子线路映射问题可进行更精准地建模, 考虑量子位错误率及量子程序保真度, 将在后续工作中进行深入地探索。

参 考 文 献

- [1] SHOR P W. Polynomial-Time algorithms for prime factorization and discrete logarithms on a quantum computer[J]. *SIAM Review*, 1999, 41(2): 303-332.
- [2] GROVER L K. A fast quantum mechanical algorithm for database search[C]//Proceedings of the 28th annual ACM symposium on Theory of computing. New York: ACM, 1996: 212-219.
- [3] 黄一鸣, 雷航, 李晓瑜. 量子机器学习算法综述[J]. *计算机学报*, 2018, 41(1): 145-163.
HUANG Y M, LEI H, LI X Y. A survey on quantum machine learning[J]. *Chinese Journal of Computer*, 2018, 41(1): 145-163.
- [4] 张焕国, 毛少武, 吴万青, 等. 量子计算复杂性理论综述[J]. *计算机学报*, 2016, 39(12): 2403-2428.
ZHANG H G, MAO S W, WU W Q, et al. Overview of quantum computation complexity theory[J]. *Chinese Journal of Computer*, 2016, 39(12): 2403-2428.
- [5] PRESKILL J. Quantum computing in the NISQ era and beyond[J]. *Quantum*, 2018, 2: 79.
- [6] HUAWEL. HiQ quantum computing cloud platform [EB/OL]. (2022-09-30). <https://hiq.huaweicloud.com/en/>.
- [7] IBM. IBM Q experience device[EB/OL]. (2022-09-30). <https://quantum-computing.ibm.com/qx/devices>.
- [8] LI G, DING Y, XIE Y. Tackling the qubit mapping problem for NISQ-era quantum devices[C]//Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2019: 1001-1014.
- [9] CHAKRABARTI A, SUR-KOLAY S, CHAUDHURY A. Linear nearest neighbor synthesis of reversible circuits by graph partitioning[EB/OL]. (2012-12-27). <https://arxiv.org/pdf/1112.0564.pdf>.
- [10] 窦星磊, 刘磊, 陈岳涛. 面向超导量子计算机的程序映射技术研究[J]. *计算机研究与发展*, 2021, 58(9): 1856-1874.
DOU X L, LIU L, CHEN Y T. An investigation into quantum program mapping on superconducting quantum computers[J]. *Journal of Computer Research and Development*, 2021, 58(9): 1856-1874.
- [11] SIRAICHI M Y, SANTOS V F, COLLANGE C, et al. Qubit allocation[C]//Proceedings of the 2018 International Symposium on Code Generation and Optimization. New York: ACM, 2018: 113-125.
- [12] 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*, 2018, 38(7): 1226-1236.
- [13] DING Y, WU X C, HOLMES A, et al. Square: Strategic quantum ancilla reuse for modular quantum programs via cost-effective uncomputation[C]//2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA). [S.l.]: IEEE, 2020: 570-583.
- [14] NIELSEN M A, CHUANG I. Quantum computation and quantum information[J]. *American Journal of Physics*, 2002, 70: 558-559.
- [15] 王吉林, 刘建设, 陈培毅. 量子计算与超导量子计算机[J]. *微纳电子技术*, 2009, 46(6): 321-326.
WANG J L, LIU J S, CHEN P Y. Quantum computation and superconducting quantum computer[J]. *Micronano-Electronic Technology*, 2009, 46(6): 321-326.
- [16] 喻志超, 李扬中, 刘磊, 等. 量子计算模拟及优化方法综述[J]. *计算机工程*, 2022, 48(1): 1-11.
YU Z C, LI Y Z, LIU L, et al. Survey of quantum computing simulation and optimization methods[J]. *Computer Engineering*, 2022, 48(1): 1-11.
- [17] 付祥, 郑宇真, 苏醒, 等. 一种面向含噪中尺度量子技术的量子-经典异构计算系统[J]. *计算机研究与发展*, 2021, 58(9): 1875-1896.
FU X, ZHENG Y Z, SU X, et al. A heterogeneous quantum-classical computing system targeting noisy intermediate-scale quantum technology[J]. *Journal of Computer Research and Development*, 2021, 58(9): 1875-1896.
- [18] FLOYD R W. Algorithm 97: Shortest path[J]. *Communications of the ACM*, 1962, 5(6): 345.
- [19] 徐海. 线性最近邻量子线路构建与优化研究[D]. 南通: 南通大学信息科学技术学院, 2016
XU H. Construction and optimization for linear nearest neighbor quantum circuits[D]. Nantong: School of Information Science and Technology Nantong University, 2016.
- [20] WILLE R, GROBE D, TEUBER L, et al. RevLib: An online resource for reversible functions and reversible circuits[C]//38th International Symposium on Multiple Valued Logic. Washington DC: IEEE, 2008: 220-225.
- [21] GREEN A S, LUMSDAINE P L F, ROSS N J, et al. Quipper: A scalable quantum programming language [C]//Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation. New York: ACM, 2013: 333-342.
- [22] ABHARI A J, FARUQUE A, DOUSTI M J, et al. Scaffold: Quantum programming language[R]. [S.l.]: Princeton Univ NJ Dept of Computer Science, 2012.