计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
[doi: 10.15888/j.cnki.csa.010094] [CSTR: 32024.14.csa.010094]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

多策略量子麻雀搜索算法在 NISQ 量子比特映射中的应用①

刘述娟[1], 韩雪娜[1], 李　晖[1,2], 王杰鹏[1], 姬迎松[1]

[1](哈尔滨商业大学 计算机与信息工程学院, 哈尔滨 150028)
[2](黑龙江省电子商务与信息处理重点实验室, 哈尔滨 150028)
通信作者: 李　晖, E-mail: hrbcu_lh@163.com

摘　要: 在噪声中等规模量子 (noisy intermediate-scale quantum, NISQ) 时代, 由于硬件耦合约束, CNOT 门往往无法直接执行, 需要引入额外的 SWAP 门将逻辑量子比特映射至合适的物理位置, 以确保电路的可执行性. 为了减少传统量子比特映射过程中由于 SWAP 操作引起的额外开销, 提出多策略量子麻雀搜索算法 (multi-strategy quantum sparrow search algorithm, MQSSA), 并应用于量子比特映射中. 基于作用同一量子比特对的非近邻 (non-nearest neighbour, non-NN) CNOT 门的个数, 定义量子比特联动次数; 结合 CNOT 门物理间距, 定义联动量子门集; 根据量子比特联动次数以及 SWAP 门数量构建适应度函数; 同时将适应度最优的个体定义为发现者, 通过引入量子叠加态机制, 使发现者具备并行搜索能力, 可以同时探索多个位置, 扩大搜索空间; 此外, 为了避免陷入局部最优, MQSSA 引入高斯噪声作为跟随者位置更新扰动机制, 增强跳出局部最优能力; 设置警戒者机制维持搜索多样性. 实验结果表明, 在 t|ket⟩和 Qiskit 编译器中, MQSSA 的 SWAP 门数量分别平均减少 37.5% 和 46.6%, 硬件开销分别平均减少 13.3% 和 13.2%. 表明算法在量子比特映射中的表现更加高效, 提升了优化结果质量.

关键词: 耦合约束; 量子比特映射; 多策略量子麻雀搜索算法; 量子比特联动次数; 联动量子门集; 适应度函数; 高斯噪声; 扰动机制

引用格式: 刘述娟,韩雪娜,李晖,王杰鹏,姬迎松.多策略量子麻雀搜索算法在 NISQ 量子比特映射中的应用.计算机系统应用. http://www.c-s-a.org.cn/1003-3254/10094.html

Application of Multi-strategy Quantum Sparrow Search Algorithm in NISQ Qubit Mapping

LIU Shu-Juan[1], HAN Xue-Na[1], LI Hui[1,2], WANG Jie-Peng[1], JI Ying-Song[1]

[1](School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China)
[2](Heilongjiang Provincial Key Laboratory of Electronic Commerce and Information Processing, Harbin 150028, China)

Abstract: In the noisy intermediate-scale quantum (NISQ) era, due to hardware coupling constraints, CNOT gates often cannot be executed directly. Additional SWAP gates need to be introduced to map logical qubits to suitable physical locations for ensuring circuit executability. To reduce the additional overhead caused by SWAP operations during traditional qubit mapping, this study proposes the multi-strategy quantum sparrow search algorithm (MQSSA) and applies it to qubit mapping. The qubit linkage count is defined based on the number of non-nearest neighbour (non-NN) CNOT gates acting on the same qubit pair. Combined with the physical spacing of CNOT gates, a linked quantum gate set is defined, and a fitness function is constructed based on the qubit linkage count and SWAP gate number. Simultaneously, the individual with the optimal fitness is defined as the discoverer. A quantum superposition state mechanism is introduced to equip the discoverer with parallel search capabilities, enabling the simultaneous exploration of multiple

① 基金项目: 黑龙江省自然科学基金 (LH2024F042); 黑龙江省普通本科高等学校青年创新人才培养计划 (UNPYSCT-2020212); 哈尔滨商业大学“青年科研创新人才”培育计划 (2023-KYYWF-0983)
收稿时间: 2025-07-30; 修改时间: 2025-09-15; 采用时间: 2025-10-14; csa 在线出版时间: 2025-12-29

positions to expand the search space. Furthermore, to avoid falling into local optima, MQSSA introduces Gaussian noise as a follower position update perturbation mechanism, enhancing the ability to jump out of local optima. Additionally, the sentinel mechanism is implemented to maintain search diversity. Experimental results demonstrate that within the t|ket⟩and Qiskit compilers, MQSSA achieves average reductions of 37.5% and 46.6% in SWAP gate counts respectively, alongside average hardware overhead reductions of 13.3% and 13.2% respectively. This indicates the proposed algorithm has more efficient performance in qubit mapping, thereby improving the quality of optimization outcomes.

Key words: coupling constraint; qubit mapping; multi-strategy quantum sparrow search algorithm (MQSSA); qubit linkage count; linked quantum gate set; fitness function; Gaussian noise; perturbation mechanism

随着量子计算的发展, 这一领域正逐渐从理论探索走向实际应用. 在过去的几十年中, 量子计算的基本原理和算法得到了深入研究, 如 Shor 算法[1]、Grover 算法[2]以及量子模拟[3]等具有革命性的理论突破. 随着量子比特技术的不断迭代, 超导、离子阱和光学量子比特等多种硬件架构正在快速发展. 尽管如此, 当前量子计算机仍处于噪声中等规模量子 (noisy intermediate-scale quantum, NISQ) 时代, 受限于较高噪声水平、有限的量子比特数量以及量子比特间的耦合约束, 实际计算能力尚未完全达到预期.

在 NISQ 时代, 由于量子硬件与软件技术发展的不平衡, 量子硬件和软件之间的差距成为制约量子计算发展的重要因素[4]. 量子硬件在过去几年取得了显著进展, 但仍面临着量子比特数量、噪声水平、量子比特之间连接性等方面的限制. 量子软件方面, 尽管量子算法和编程语言的开发相对较为活跃, 许多重要的量子算法理论已经提出, 并逐步转向对硬件资源的优化利用, 但由于硬件的限制, 这些算法尚未完全发挥其潜力.

量子硬件与软件技术之间的不匹配导致量子算法和程序在实际执行中的效率受限. 在量子比特映射过程中, 由于约束限制逻辑量子比特通常不能直接映射到硬件的物理量子比特上, 如 CNOT 门在物理架构中经常无法直接执行, 为了使逻辑量子比特适应硬件拓扑结构的限制, 目前主要采用两种方法: 一是全局调整量子比特的位置, 二是在映射过程中插入额外的 SWAP 门操作[5–8]. 然而, 这两种方法都会引入额外的噪声和计算开销[9,10]. 其中, 全局调整量子比特位置的计算复杂度较高, 对硬件特性依赖性强, 且通用性不足; 相比之下, 引入附加 SWAP 门的计算复杂度相对较低, 并能灵活适应硬件的拓扑约束. 然而, 过多的 SWAP 门不仅增加电路深度和执行时间, 还会引入额外噪声, 降低量子电路的保真度. 因此, 最小化额外 SWAP 门数量已成为众多研究人员关注的核心问题[11–13]. 因此, 量子比特映射问题已成为连接量子算法与硬件的核心环节, 其优化效果直接决定量子程序的执行性能.

量子比特映射问题已被证明是 NP 完全问题[14,15]. 随着量子电路规模的逐渐增大, 灵活性强、扩展性佳的启发式算法成为研究的主流方向. 目前, 国内外学者对量子比特映射算法进行了大量探索. Siraichi 等人[16]提出一种初始映射方法, 优先考虑物理上相邻的 CNOT 门, 依据双量子比特门的数量决定是否移动; 该算法虽然具有较快的运行速度, 但设计较为简单, 整体效果不佳. Sünkel 等人[17]提出 GA4QCO 框架, 在研究拓扑结构对启发式量子比特分配算法的影响方面提供了有价值的分析, 但实验规模相对较小, 难以评估算法在更大规模问题上的计算开销. Steinberg 等人[18]提出一种启发式量子比特分配算法 HQAA, 用于在 NISQ 设备上为量子算法分配初始量子比特位置, 但在大规模量子线路中不够高效. Liu 等人[19]提出一种基于动态前瞻策略的量子比特映射方法 QM-DLA, 利用动态规划和前瞻技术来优化量子电路映射过程中的决策, 以减少所需的 SWAP 操作和其他量子门操作, 其计算复杂度高、动态调整困难、硬件适应性相对较差. 韩子傲等人[20]提出一种基于遗传算法的量子电路映射策略 GQCMS, 通过交叉、突变和邻近门初始化有效减少了 SWAP 门和 CNOT 门, 但其适应度函数仅考虑 SWAP 和 CNOT 数量, 过于单一. Lao 等人[21]介绍一种针对量子模拟和量子近似优化算法 (quantum approximate optimization algorithm, QAOA) 的应用特定量子编译器 2QAN, 并提出一种酉统一技术, 以减少硬件中两量子位门的数量.

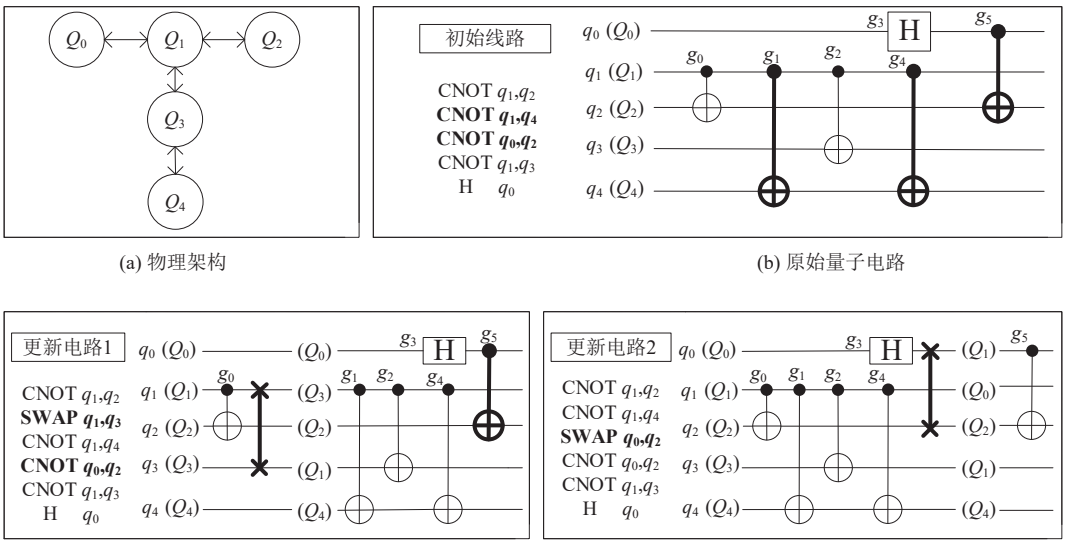
现有研究提出了多种量子比特映射方法, 包括基

于启发式规则的初始映射以及遗传算法等智能优化方法. 尽管这些方法在一定程度上缓解了映射带来的额外开销, 但它们对量子比特间的“联动关系”考虑不足, 忽视了量子比特联动次数对映射结果的影响.

本文重点关注受硬件耦合约束影响的量子比特映射问题. 量子比特映射问题本质上是一个复杂的组合优化问题[14]. 麻雀搜索算法 (sparrow search algorithm, SSA) 作为一种群体智能优化方法, 兼具全局探索与局部开发的特性, 适合用于解决此类复杂优化任务. 然而, 传统 SSA 在复杂问题中易陷入局部最优. 因此, 本文提出了多策略量子麻雀搜索算法 (multi-strategy quantum sparrow search algorithm, MQSSA), 并将其应用于量子比特映射中. MQSSA 优先处理最大量子比特联动次数的非近邻 (non-nearest neighbor, non-NN) CNOT 门; 引入量子叠加态机制, 使发现者具备并行搜索能力, 扩大搜索空间; 为了避免陷入局部最优, 引入高斯噪声对跟随者进行扰动, 使得跟随者的位置更新具有一定的随机性, 避免局部收敛. MQSSA 考虑量子硬件的耦合约束, 以及量子电路的逻辑结构, 实现电路的有效映射.

1 量子比特映射

量子比特的映射过程主要包括两个方面: 一是逻辑量子位映射到物理量子位; 二是满足耦合约束, 交互路由. 为了更直观地理解量子比特映射问题, 通过一个小规模的实例进一步阐释其具体过程. 如图 1(a) 所示, 以 4 量子比特物理架构作为硬件平台. 在物理量子比特对$\{Q_0, Q_1\}$, $\{Q_1, Q_2\}$, $\{Q_1, Q_3\}$和$\{Q_3, Q_4\}$上可以直接执行双量子比特门, 而在$\{Q_0, Q_2\}$, $\{Q_0, Q_3\}$, $\{Q_0, Q_4\}$, $\{Q_1, Q_4\}$, $\{Q_2, Q_3\}$和$\{Q_2, Q_4\}$上不能直接执行. 现假设图 1(b) 所示的量子线路要在此硬件架构上执行. 设初始映射为$\{q_0fQ_0, q_1fQ_1, q_2fQ_2, q_3fQ_3, q_4fQ_4\}$. 由图 1(a) 知, 图 1(b) 中的 5 个 CNOT 门有 2 个可以直接执行, 即 g_0 和 g_2, 但 g_1、g_4 和 g_5 无法执行, 因为这 3 个 CNOT 门对应的量子比特对在量子硬件架构上并未直接相连. 因此, 在执行 g_1、g_4 和 g_5 前需更改量子映射, 以使所有门均可执行, 在 g_0 后插入 SWAP (q_1, q_3), 映射更新为$\{q_0fQ_0, q_1fQ_3, q_2fQ_2, q_3fQ_1, q_4fQ_4\}$, 使 g_1 和 g_4 更新为可执行门, 如图 1(c) 所示; 更新映射后, 在 g_4 后插入 SWAP (q_0, q_2), g_5 更新为近邻 (nearest neighbor, NN) 门, 使量子电路可执行, 如图 1(d) 所示.



(a) 物理架构 (b) 原始量子电路

(c) 更新量子电路1 (d) 更新量子电路2

图 1 量子比特映射示意图

本文通过插入附加的 SWAP 门, 交换相邻量子位的位置以更改量子比特映射[22,23]. 一个 SWAP 门由 3 个 CNOT 门组成, 如图 2(c) 所示. 然而, 引入 SWAP 操作会带来额外的噪声, 影响量子电路保真度; 其次, 量子电路深度的增加会影响电路整体执行时间, 这不仅对计算速度产生负面影响, 还会导致量子比特的相干时间不足以支持完整的计算过程, 增加出错的可能性. 对比图 1(b)、图 1(c) 和图 1(d), 量子门数量和电路深度均增加, CNOT 门数量由 5 个增加到 11 个; 电路深度由 5 增加到 11, 这些变化均会对电路整体性能

产生负面影响, 如造成噪声积累、量子纠错成本上升和延长总体执行时间等. 因此, 有效减少 SWAP 门数量、降低电路深度, 成为量子比特映射中的核心问题. 针对这一挑战, 本文引入了群体智能优化思想, 旨在在复杂搜索空间中实现更高效、更优质的量子比特映射.

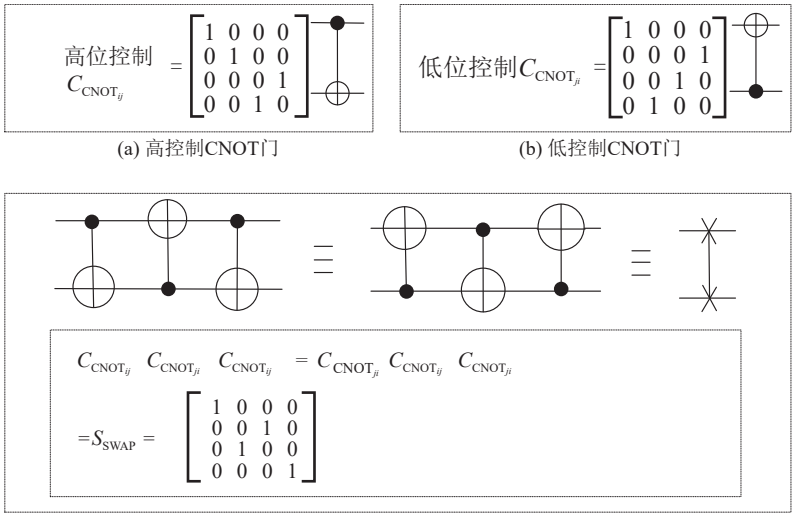


图 2 基本量子门

2 基于多策略量子麻雀搜索算法的量子比特映射

优化量子比特映射过程是量子计算中的重要挑战, 其核心在于高效地将逻辑量子比特映射到物理比特. 随着量子计算机规模的增大和硬件架构的复杂化, 传统方法往往难以有效解决这一问题[17,18,21]. 麻雀搜索算法凭借其全局搜索与局部优化的优势, 在复杂优化问题中展现出良好的适应性, 因此可用于优化量子比特映射. 然而, 传统麻雀搜索算法易陷入局部最优. 为此, 本文结合量子力学的特性, 提出多策略量子麻雀搜索算法, 该算法通过量子叠加态机制扩大搜索空间以提升全局搜索能力, 并结合发现者、跟随者、高斯扰动与警戒者等多策略协同机制, 增强搜索的灵活性与鲁棒性, 从而在优化量子比特映射的同时, 有效减少 SWAP 门的数量并提升量子电路执行效率.

2.1 麻雀搜索算法 (SSA)

麻雀搜索算法是一种群体智能优化算法, 其灵感来源于麻雀群体的觅食行为和防御策略, 通过模拟麻雀觅食时的群体协作机制, 在解空间中进行全局探索和局部开发, 从而寻求最优解, 其过程主要包括个体初始化、发现者和跟随者的角色划分、位置更新机制、适应度评估、全局最优解更新以及随机扰动机制等几个核心部分.

麻雀搜索算法通过随机生成初始个体的位置 I_i 来初始化种群, 其中 i 为个体位置的编号, 这些位置代表问题的潜在解. 在群体 $P(t)$ 中, 个体被分为发现者和跟随者, 其中 t 表示第 t 代种群. 种群的个体代表一组候选解, 通过算法的迭代与不断进化, 以寻求最优解. 具体而言, 发现者负责全局搜索, 探索广泛的区域以发现潜在的优质解; 跟随者则围绕发现者进行局部搜索, 进一步优化当前区域的解. 这种角色分工, 使麻雀搜索算法能够在全局范围内进行广泛探索, 同时在局部区域进行精细优化, 提升整体求解效果.

个体位置更新是麻雀搜索算法的核心部分. 发现者通过随机跳跃或大范围探索来更新位置. 跟随者根据发现者的位置信息和自身适应度, 在局部区域内进行精细搜索. 每次迭代中, 发现者和跟随者的位置更新公式分别如式 (1)、式 (2) 所示:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(\dfrac{-i}{\alpha \cdot iter_{\max}}\right), & R_2 < ST \\ X_{i,j}^{t} + Q \cdot L, & R_2 \geqslant ST \end{cases} \tag{1}$$

其中, $iter_{\max}$ 是一个常数, 表示最大的迭代次数; $X_{i,j}$ 表示第 i 个麻雀在第 j 维中的位置信息; $\alpha \in (0,1]$ 是一个随机数; $R_2\,(R_2 \in (0,1])$ 和 $ST\,(ST \in (0.5,1])$ 分别表示预

警值和安全值; Q 是服从正态分布的随机数; L 表示一个 $1\times d$ 的矩阵, 其中该矩阵内每个元素全部为 1.

$$X_{i,j}^{t+1}=\begin{cases}Q\cdot\exp\left(\dfrac{X_{\text{worst}}^{t}-X_{i,j}^{t}}{i^2}\right), & i>\dfrac{n}{2}\\ X_P^{t+1}+\left|X_{i,j}^{t}-X_P^{t+1}\right|\cdot A^{+}\cdot L, & i\leqslant\dfrac{n}{2}\end{cases}\tag{2}$$

其中, X_P 是目前发现者所占据的最优位置; X_{worst} 则表示当前全局最差的位置; A 表示一个 $1\times d$ 的矩阵, 其中每个元素随机赋值为 1 或−1, 并且 $A^{+}=A^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1}$.

在麻雀搜索算法中, 个体的优劣通过适应度函数 $F(I_i)$ 进行评估, 适应度函数根据问题的评价标准对个体的解进行量化, 反映个体的质量和优化方向. 适应度高的个体有更高概率成为发现者, 引导群体的搜索方向. 通过计算个体的适应度值, 算法能够找到当前种群中表现最优的个体, 即当前代的最优解. 此外, 历史最优解也会被记录下来, 作为搜索过程中的参考, 指导后续探索. 每次迭代后, 算法会更新全局最优解, 并在达到最大迭代次数或其他终止条件时停止.

2.2 量子比特映射中的 MQSSA 优化算法

传统麻雀搜索算法, 发现者通过随机跳跃进行全局搜索, 尽管这种搜索方式能够探索广泛的区域, 但在复杂问题中, 其搜索能力存在局限性. 当发现者在全局搜索过程中找到较优解时, 大量跟随者会迅速向其靠近, 使部分区域的种群密度过高, 导致算法易陷入局部最优. 为了克服这些问题, 引入量子表示方式, 利用量子力学中的叠加原理使发现者处于叠加态, 实现其对多个位置同时探索. 这种“同时性”允许发现者在更广阔的搜索空间内并行搜索, 增加搜索的多样性和全面性. 在跟随者的位置更新过程中, 引入高斯噪声对个体进行扰动, 增加位置更新的随机性, 避免局部收敛.

2.2.1 联动量子门集

定义 1. 量子比特联动次数: 在量子电路 C 中, 设 CNOT(q_i, q_j) 为任一非近邻 CNOT 门, 其作用于同一量子比特对 (q_i, q_j) 的次数 $R_{q_iq_j}$ 为 CNOT(q_i, q_j) 的量子比特联动次数, 其中 $i\neq j$.

定义 2. 联动量子门集: 在量子电路 C 中, 设 $G_n=\{g_1,g_2,\cdots,g_n\}$ 为所有 non-NN 门集合, $G_m=\{g_1,g_2,\cdots,g_m\}$ 为 $\max(R_{q_iq_j})$ 的 non-NN 门集合, G_m 中各量子门对应量子比特的物理间距集合为 $D_m=\{D(g_1),D(g_2),\cdots,D(g_m)\}$, 若 $D(g_1)=D(g_2)=\cdots=D(g_k)=\min D_m$, 则称 $G_k=\{g_1,g_2,\cdots,g_k\}$ 为联动量子门集, 其中, $0<k\leqslant m\leqslant n$.

在初始映射中, 对于 non-NN 门, 需要引入 SWAP 操作, 将其转换为 NN 门. 对于 non-NN 门, 传统方法通常基于某种固定规则或进行随机处理[21], 忽视了量子比特联动次数对映射过程的影响, 导致 SWAP 门过多. 以图 1(b) 为例, CNOT 门 g_1、g_4 和 g_5 量子比特物理间距均为 2. 根据定义 1 和定义 2 可知, $\max(R_{q_1q_4})=2$; 联动量子门集 $G_2=\{g_1,g_4\}$. 若优先处理 G_2 中的量子门, 只需插入 2 个 SWAP 门即可使量子电路顺利执行, 如图 1(c) 和图 1(d). 传统映射方法通常对物理间距相等的 non-NN 门进行随机处理, 然而, 若优先对图 1(b) 中的 g_5 执行交换操作, 需在 g_5 前插入 SWAP(q_0, q_1), 更新电路后, g_1 和 g_4 物理间距由 2 延长至 3, 要使整个电路可执行, 至少额外引入 2 个 SWAP 门 (SWAP(q_0, q_1) 和 SWAP(q_3, q_4)), 相比图 1 所示方法, 增加一个 SWAP 操作. 因此, 在执行多策略量子麻雀搜索算法前, 首先, 确定联动量子门集, 并优先对其处理, 可以更好地利用现有硬件拓扑结构, 提高映射效率; 同时, 减少额外迁移需求, 降低整体操作延迟, 更有效调度其他操作, 使得更多量子操作能够并行执行.

2.2.2 多策略量子麻雀搜索算法

在映射过程中, 若对 non-NN 门的处理顺序不当, 可能会导致量子比特间的有效耦合路径变长, 引入过多 SWAP 门, 增大电路的执行复杂度. 优先识别联动量子门集, 可以合理安排 CNOT 门的执行顺序, 针对性地减少冗余 SWAP 操作, 降低门数开销. 多策略量子麻雀搜索算法 (MQSSA) 基于联动量子门集, 通过模拟麻雀觅食时的群体协作和逃逸机制, 优化量子比特映射. MQSSA 主要包含 4 方面核心机制: 即在初始化阶段引入“联动量子门集”的构建, 使算法在搜索早期即可针对高频作用的量子比特对进行优化, 减少冗余 SWAP 的可能性; 利用量子叠加态描述发现者位置, 提升全局搜索能力, 增强算法全局性; 设计多策略协作机制, 包括发现者的全局探索、跟随者的局部开发、高斯扰动抑制早熟收敛以及警戒者维持种群多样性, 以提升搜索过程的灵活性; 构建结合 SWAP 门数量和量子比特联动次数的适应度函数, 基于该函数, MQSSA 能够区分优劣解, 引导搜索过程朝向更优映射方案收敛, 使优化目标更契合量子比特映射问题的实际需求. 通过这 4 个方面的设计, MQSSA 在量子比特映射任务中兼具全局性与灵活性, 能够在减少 SWAP 门和 CNOT 门数

量的同时有效避免陷入局部最优.

在量子电路执行过程中, SWAP 门用于交换两个比特的状态. 然而, 过多的 SWAP 操作导致额外硬件开销. 因此, 最小化 SWAP 门数量是提高量子电路性能的关键目标之一. 此外, 量子比特联动次数反映量子比特之间的相互作用和信息传递的复杂性, 影响电路的可执行性. 基于此, 考虑 SWAP 门数量和量子比特联动次数, 设计适应度函数如式 (3):

$$F(I_{\mathrm{i}}) = \frac{1}{\omega_1 \times \sum_{i=1}^{n} N_{\mathrm{SWAP}} + \omega_2 \times \sum_{j=1}^{n} R_{q_i q_j}} \tag{3}$$

其中, $F(I_i)$为个体的适应值, 值越大越好; N_{SWAP}为 SWAP 门的数量, $R_{q_i q_j}$为量子比特联动次数; ω_1和ω_2为调整权重, 分别赋值为 0.67 和 0.33, 用于平衡各指标对整体适应度的贡献.

在量子比特映射过程中, 合理设置 SWAP 门数量和量子比特联动次数之间的权重十分重要. SWAP 门数量直接影响电路的执行时间、硬件开销和保真度. 量子比特联动次数反映作用于同一量子比特对的次数. 较高的联动次数意味着量子比特在执行操作时, 可能会增加量子比特间物理间距的交互, 会引入更多 SWAP 操作. 然而, 这种影响通常是间接的, 它通过 SWAP 门的数量影响最终的优化效果. 若 SWAP 门数量的权重过低, 优化过程中可能会优先减少量子比特联动次数, 而忽略对 SWAP 门数量的控制, 导致最终的映射方案间接增加交换操作. 若量子比特联动次数的权重过高, 可能会导致优化过程过于注重联动关系, 加大映射复杂性, 影响算法的全局搜索能力.

在量子力学中, 粒子通过波函数表示, 表现出概率性. 量子系统的状态随时间变化, 薛定谔方程正是描述这一变化的数学公式. 其中, 时间依赖的薛定谔方程用于描述量子系统在外界作用下的时间演化, 适用于动态过程. 因此, 利用其来更新波函数的演化, 具体形式如式 (4):

$$\mathrm{i}\hbar \frac{\partial \psi(r,t)}{\partial t} = \hat{H}\psi(r,t) \tag{4}$$

其中, $\psi(r,t)$为波函数, 描述粒子在位置 r 和时间 t 的概率幅; i 为虚数单位; ħ为约化普朗克常数 ($\hbar = \frac{h}{2\pi}$); $\hat{H}$为哈密顿量.

在 MQSSA 中, 哈密顿量的引入可以描述搜索过程中发现者所处状态的“能量”, 引导其位置演化. 系统的能量通过哈密顿量表示, 如式 (5). 其中, 在动态系统中, 动能项反映个体的“活动性”或“能量”传递, 动能本身不直接反映解的优劣, 而更多与搜索过程中的状态变化相关; 势能描述一个系统在某种状态下的稳定性, 低势能通常对应着系统处于较为稳定的状态. 适应度函数用于评估个体解的优劣, 而势能项代表系统的“潜在能量”, 可以反映解的优劣.

$$\hat{H} = \hat{T} + \hat{V} = -\frac{\hbar^2}{2m}\nabla^2 + V(r) \tag{5}$$

其中, $\hat{T}$为动能算符; $\hat{V}$为势能算符; m 为粒子的质量; ∇^2为拉普拉斯算符, 描述波函数的空间二阶导数.

传统麻雀搜索算法中, 发现者负责全局搜索, 在搜索空间中寻找最优解. 量子麻雀搜索算法中, 引入势能概念, 将适应度函数和麻雀搜索算法中发现者的位置更新公式映射到势能项中, 模拟量子系统的演化, 使搜索过程更符合物理优化机制. 这种方式可以确保优质解具有较低的“能量”, 使质量较好的解对应较低的能量状态. 势场$V(r)$定义为:

$$V(r) = -F(I_i)\frac{1}{\left|x_{i,j}^{t} - x_{i,j}^{t+1}\right|} \tag{6}$$

其中, $F(I_i)$为适应度函数; $x_{i,j}$表示第 i 个个体在第 j 维中的位置信息; $x_{i,j}^{t}$为第 i 个个体的位置, t 表示当前迭代次数; $x_{i,j}^{t+1}$为第 i 个个体更新后的位置; $\left|x_{i,j}^{t} - x_{i,j}^{t+1}\right|$为个体当前位置与最优解之间的欧几里得距离.

物理学中, 势能描述系统中粒子之间的相互作用. 对于引力, 势能与粒子之间的距离相关, 且具有负值. 符合常见的物理规律: 物体在靠近吸引源时, 势能减小, 远离时势能增大. 在 MQSSA 中, 负号的引入是为了模拟发现者个体朝向最优解靠近的“吸引力”, 当个体接近最优解时, 势能减小, 表示它受到吸引, 能够继续向最优方向靠近.

在量子比特映射过程中, 跟随者的主要作用是对全局搜索策略进行补充, 通过局部探索提升整体优化效果. 为了避免跟随者陷入局部最优, 在更新过程中引入高斯噪声, 对个体位置更新施加扰动, 增加位置更新的随机性, 降低算法陷入局部收敛的风险. 跟随者位置更新公式如式 (7):

$$x_{\mathrm{new}} = x_{\mathrm{old}} + \alpha \cdot (x_{\mathrm{leader}} - x_{\mathrm{old}}) + \eta \cdot N\left(0, \sigma^2\right) \tag{7}$$

其中, x_{new}为更新后的个体位置; x_{old} 为个体当前的位置; α为调整因子, 控制个体向发现者位置移动的程度; x_{leader}为当前发现者的位置; $N(0,\sigma^2)$为一个从高斯分布中采样的噪声, 通常是随机生成的值, 表示扰动; η 为高斯噪声扰动中的调节因子, 决定噪声对个体位置更新的贡献大小, 其计算公式如式 (8):

$$\eta = k\cdot\left(1-\frac{F(I_{\text{current}})}{F(I_{\text{best}})}\right) \tag{8}$$

其中, k为调节系数, $F(I_{\text{current}})$为当前解的适应度, $F(I_{\text{best}})$为最优解的适应度. 当前解与最优解的差距较大时, 增加噪声强度; 差距较小时, 减小噪声强度.

警戒者在整个搜索过程中承担监控角色, 旨在增加搜索多样性, 防止算法陷入过度局部优化, 同时有效避免无效的映射方案和错误, 以增强算法的鲁棒性. 通过适时的干预和引导, 警戒者在优化过程中维持平衡, 确保在复杂的量子硬件约束下找到更优的映射方案. 警戒者约占种群的 10%, 在每次迭代时动态随机选取, 其位置更新如式 (9):

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^{t}+\beta\cdot\left|X_{i,j}^{t}-X_{\text{best}}^{t}\right|, & f_i > f_g \\ X_{i,j}^{t}+K\cdot\left(\dfrac{\left|X_{i,j}^{t}-X_{\text{worst}}^{t}\right|}{(f_i-f_w)+\varepsilon}\right), & f_i = f_g \end{cases} \tag{9}$$

其中, X_{best}为当前全局最优位置; β作为步长控制参数, 服从均值为 0, 方差为 1 的正态分布的随机数; $K\in[-1,1]$为一个随机数; f_i则是当前麻雀个体的适应度值; f_g和f_w分别是当前全局最佳和最差的适应度值; ε是最小的常数, 以避免分母为零; K表示步长控制参数.

2.2.3 MQSSA 算法流程

多策略量子麻雀搜索算法是一种融合量子计算特性的智能优化算法, 旨在优化 NISQ 设备上的量子比特映射问题. MQSSA 结合量子计算原理, 通过量子比特交互关系的优化、并行搜索、高斯噪声扰动等策略, 提供了一种高效的量子比特映射方法. 该算法基于量子比特联动次数, 筛选出联动量子门集; 依据最优适应度值确定发现者, 结合量子特性进行解的干涉与更新, 使其具备并行搜索的能力, 可以动态调整个体搜索方向, 使搜索过程更加灵活; 对跟随者施加高斯噪声扰动机制, 增强搜索随机性, 规避局部最优; 选取种群 P 约 10% 的个体作为警戒者, 用于监测种群环境, 提高搜索多样性. 随着 MQSSA 的迭代更新, 每一代个体都会根据其适应度值进行筛选和更新, 多次迭代后, MQSSA 逐步收敛至最优映射方案. 为优化量子电路、降低硬件开销提供有效解决方法. 算法整体流程如图 3 所示.

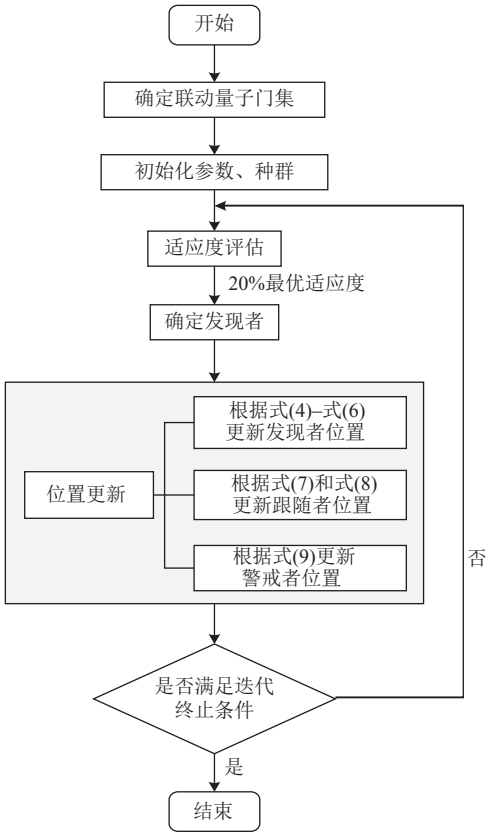


图 3 MQSSA 整体流程图

具体操作步骤如下.

步骤 1: 计算最大量子比特联动次数, 确定联动量子门集;

步骤 2: 基于适应度函数筛选个体, 选取值最优的前 20% 个体作为发现者, 其余 80% 个体作为跟随者, 并随机选取种群中约 10% 个体作为警戒者;

步骤 3: 引入量子表示方式, 使发现者处于叠加态, 进行并行探索, 扩大搜索空间;

步骤 4: 对跟随者施加高斯噪声扰动, 增强搜索随机性, 规避局部最优;

步骤 5: 更新发现者、跟随者、警戒者位置, 调整搜索方向, 提高优化效果;

步骤 6: 判断是否满足终止条件, 若满足, 则输出最优解, 否则, 转至步骤 2 继续迭代.

多策略量子麻雀搜索算法伪代码如算法 1.

算法 1. 多策略量子麻雀搜索算法

输入: 未路由电路, 初始映射 π, 电路拓扑结构, 迭代次数.
输出: 路由电路, 一系列映射 $\{\pi_i\}$, 一系列 NN 门 $\{G_{\pi i}\}$.

1. Begin
2. Initialize the set of qubit maps $T = \{\pi\}$
3. Initialize the set of NN gates for each map, $G = \{G_\pi\}$, $G_\pi \leftarrow$ all NN two-qubit gates for map π
4. Initialize $G_{no} \leftarrow$ all non-NN two-qubit gates
5. Initialize population P with N individuals
6. while $G_{no} = \varnothing$ do
7. Select the coupled quantum gate set $G_k \in G_{no}$ in π_i
8. for iteration = 1 to Max_iterations do
9. Discoverers←Introduce quantum representation to enable parallel search
10. Followers←Apply Gaussian perturbation to enhance search diversity
11. Scouters←10% of the population monitors the population environment to enhance search diversity
12. end
13. Update positions of discoverers, followers, and scouts
14. $Z \leftarrow$Find all SWAP gates on qubits in G_k
15. Select the best SWAP from Z and add it to π_i
16. Update qubit map from π_i to π_{i+1}
17. $G_{\pi i+1} \leftarrow$Find NN gates in G_{no} for map π_{i+1}
18. Remove all NN gates from G_{no}
19. end while
20. end

3 实验分析

为了验证 MQSSA 的有效性, 本文采用文献[21]中的量子电路进行测试, 该电路可针对不同的量子比特拓扑和硬件门集进行编译, 适用于多种量子计算架构. 为了保证 MQSSA 在映射优化中的收敛性和计算效率, 将种群规模设置为 50, 迭代次数设置为 100, 以避免因种群规模过小或过大而导致早熟收敛, 同时确保在合理计算时间内获得较优解.

本文以附加 SWAP 和 CNOT 门的数量作为评估各编译器性能的关键指标, 这两个数值越小, 说明编译器的优化效果越好, 映射性能越优. 基准测试程序为 IBM 的 Qiskit 量子程序, 对于 CX/CNOT 门集, 使用 Qiskit 编译器进行分解和优化, 并采用文献[24]的基准测试方法. MQSSA 在 Python 3.11 下实现, 并在一台搭载 Intel Core i5 处理器 (2.50 GHz, 8GB RAM) 的笔记本电脑上进行所有编译. 使用 t|ket⟩编译器[25] (版本 0.11.0, 推荐 “FullPass”) 和 Qiskit 编译器[26] (版本 0.26.2, 优化级别为 3) 进行测试, 实验采用不同规模的量子电路 (4–22 量子比特) 进行测试, 并针对每个规模的电路运行 5 次, 最终选取最佳结果作为映射性能的评估依据.

将 MQSSA 与 t|ket⟩和 Qiskit 的编译开销进行对比. 图 4 和图 5 展示了 MQSSA 在 IBM 上的运行结果. 正如图 4 和图 5 所示, 相较于 t|ket⟩和 Qiskit, MQSSA 在额外的 SWAP 门和双量子门硬件开销方面均有所优化, 且随着量子电路规模的增大, 优化效果越明显. 具体而言, 对于 22 量子比特规模的电路, SWAP 门和 CNOT 门的数量比 t|ket⟩分别减少 43.1% 和 16.7%; 相比 Qiskit 则分别减少 68.3% 和 13.3%.

在 t|ket⟩编译器中, 对于 4–10 规模的量子电路, 附加 SWAP 门的数量平均分别减少 26.4%, CNOT 门的数量分别减少 13.8%; 对于 12–22 规模的量子电路, SWAP 门和 CNOT 门的数量平均分别减少 42.9% 和 13%, 比仅使用原编译器 2QAN, SWAP 门和 CNOT 门的数量平均分别减少 5.8% 和 6.1%, 如图 4 所示.

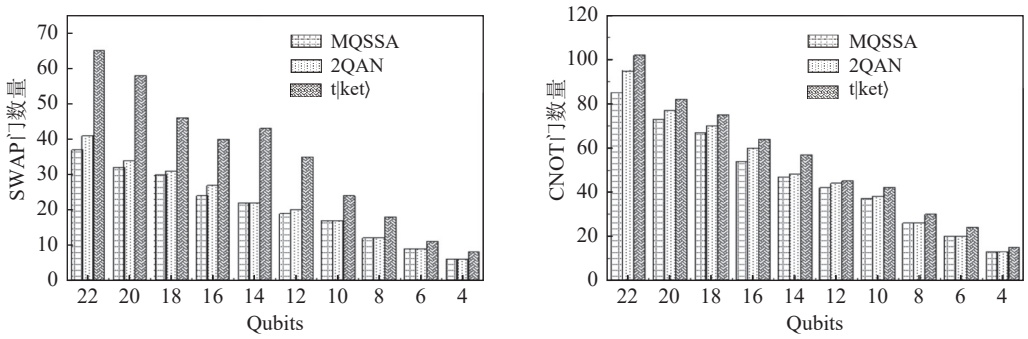


图 4 t|ket⟩编译器 SWAP 门和 CNOT 门开销对比

在 Qiskit 编译器中, 对于 4–10 规模的量子电路, 附加 SWAP 门的数量平均分别减少 26.9%, CNOT 门的数量分别减少 13.2%; 对于 12–22 规模的量子电路, SWAP 门和 CNOT 门的数量平均分别下降 61.8% 和 13.3%, 与

原编译器 2QAN 相比, SWAP 门和 CNOT 门的数量平均分别下降 8.8% 和 9.7%, 如图 5 所示.

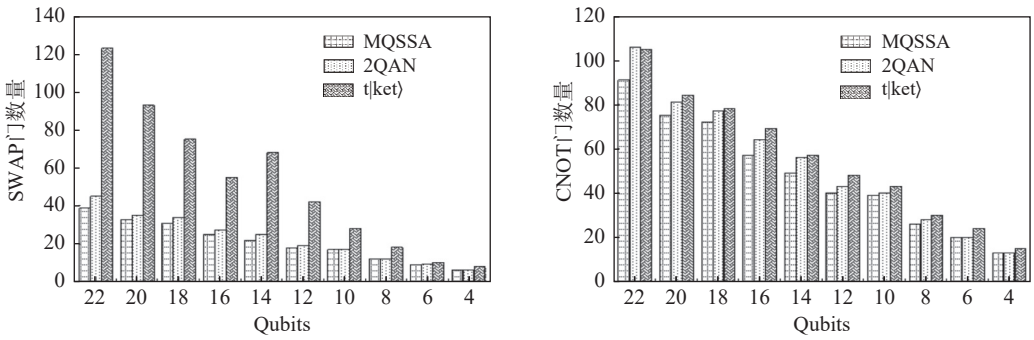


图 5 Qiskit 编译器 SWAP 门和 CNOT 门开销对比

在 t|ket⟩和 Qiskit 编译器中, 2QAN 附加 SWAP 门数量比 MQSSA 多, 主要是因为 2QAN 在处理映射问题时, 未充分考虑量子比特间联动性的优化. 对于物理间距相等的 non-NN 门, 2QAN 更倾向于进行随机选择处理, 这导致后续部分量子门的物理距离增大, 从而需要引入更多 SWAP 门来确保电路的可执行性.

为了进一步验证 MQSSA 的先进性, 本文将其与当前流行的 CSOS[13]、GA4QCO[17]和 HQAA[18]方法进行对比分析. 实验结果表明, MQSSA 在量子比特映射优化方面具有一定的有效性. 在 t|ket⟩编译器中, 对于 12–22 量子比特规模的电路, MQSSA 相较 HQAA 算法, SWAP 门和 CNOT 门的数量分别减少 7% 和 10.2%. 在 Qiskit 编译器中, 对于 10–22 量子比特规模的电路, MQSSA 相较 CSOS、GA4QCO 和 HQAA 这 3 种方法, SWAP 门的数量平均减少 5%、6.2% 和 12%, CNOT 门的数量平均减少 4%、7% 和 9.4%, 如图 6 和图 7 所示, 表 1 具体列出了 4 种映射方法执行后所产生的附加 SWAP 门数量, 清晰地反映出 MQSSA 在量子电路优化中的优势.

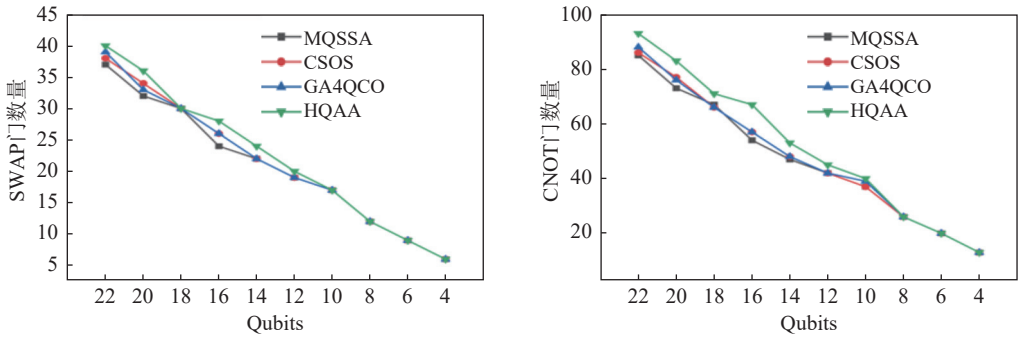


图 6 t|ket⟩编译器中各算法 SWAP 门和 CNOT 门开销对比

MQSSA 基于联动量子门集优化了 CNOT 门的执行顺序, 并结合量子叠加态原理, 在多个解之间进行并行探索, 提高了搜索效率. 相比之下, GA4QCO 在 SWAP 插入策略上更倾向于逐步优化, 无法动态适应搜索空间的变化; 此外, MQSSA 的适应度函数综合考虑 SWAP 数量与量子比特联动次数, 使搜索过程朝着最优 SWAP 规划方向收敛. 而 CSOS 主要从 SWAP 组合的角度优化, 虽然能减少局部 SWAP 开销, 但未充分考虑量子比特联动次数对电路整体优化的影响; HQAA 主要采用基于拓扑图的量子比特分配策略, 在优化过程中更关注局部拓扑信息, 而非全局优化, 容易因局部贪心策略导致过多 SWAP 门开销. 而 MQSSA 采用“全局探索+局部开发”的机制, 能够在优化过程中同时评估多个候选映射方案, 有效避免局部最优, 从而最小化 SWAP 门数量. 因此, 在量子比特映射优化过程中, MQSSA 比 GA4QCO、CSOS 和 HQAA 引入的附加 SWAP 操作更少.

MQSSA 在减少 SWAP 门数量的同时, 也关注量

子电路执行时间的变化, 确保提升整体计算效率. 表 2 展示了 2QAN 与 MQSSA 在 t|ket⟩和 Qiskit 编译器中的平均运行时间对比结果. 随着量子比特数量的增加, 算法的时间复杂度逐步上升. 对于小规模的量子电路 (如 4 量子比特模型), 在 t|ket⟩和 Qiskit 编译器中的运行时间分别约为 0.09 s 和 0.08 s. 然而, 随着电路规模的扩大, 运行时间显著增加. 例如, 在 22 量子比特电路中, t|ket⟩和 Qiskit 运行时间分别约为 27.30 s 和 0.29 s.

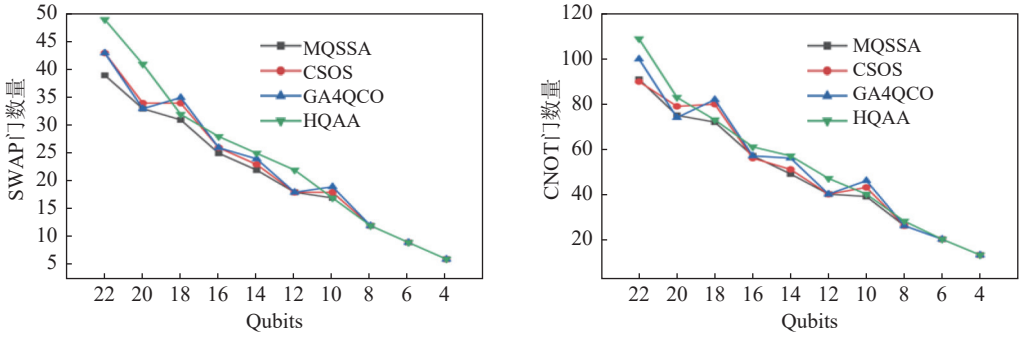


图 7 Qiskit 编译器中各算法 SWAP 门和 CNOT 门开销对比

表 1 不同映射方法附加 SWAP 门数对比

编译器名称	量子比特数 (qubit)	初始门数	CSOS	GA4QCO	HQAA	MQSSA
t|ket⟩	22	65	38	39	40	37
	20	58	34	33	36	32
	18	46	30	30	30	30
	16	40	27	26	28	24
	14	43	22	22	24	22
Qiskit	22	123	43	43	49	39
	20	93	34	33	41	33
	18	75	34	35	32	31
	16	55	26	26	28	25
	14	68	23	24	25	22

表 2 比较 2QAN 和 MQSSA 在 t|ket⟩和 Qiskit 编译器中平均运行时间 (s)

量子比特	编译器	运行时间	
		2QAN	MQSSA
4	t|ket⟩	0.10	0.09
	Qiskit	0.09	0.08
6	t|ket⟩	0.30	0.29
	Qiskit	0.09	0.09
20	t|ket⟩	11.79	11.28
	Qiskit	0.24	0.22
22	t|ket⟩	28.23	27.30
	Qiskit	0.30	0.29

4 结论与展望

在 NISQ 时代, 优化量子比特映射对于提高电路整体性能至关重要. 基于传统麻雀搜索算法, 本文提出多策略量子麻雀搜索算法, 将其应用于量子比特映射优化. MQSSA 通过增强搜索的全局性, 有效克服了传统算法在复杂问题中易陷入局部最优解的局限性. 为验证算法的有效性, 进行多组实验, 结果表明, 在 t|ket⟩和 Qiskit 编译器中, SWAP 门数量平均分别减少 37.5% 和 46.6%, CNOT 门数量平均分别减少 13.3% 和 13.2%. 相较于现有的量子比特映射方法, MQSSA 在 SWAP 门和 CNOT 门数量上均有一定减少, 进一步体现了算法的优越性, 为量子计算领域的持续发展提供了有力支持.

量子计算的快速发展对优化算法提出了更高的要求, 尤其在量子比特映射与量子电路优化等关键问题上. 未来的研究聚焦于在更大规模的量子电路中提升 MQSSA 的收敛速度和计算效率. 此外, 随着量子硬件的不断进步, 算法的灵活性和鲁棒性需进一步增强, 以适应不同量子计算架构的需求, 从而推动更多创新性解决方案的涌现, 助力量子计算领域持续发展.

参考文献

1 Xiao LG, Qiu D, Luo L, *et al*. Distributed Shor's algorithm. Quantum Information and Computation, 2023, 23(1-2): 27–44.

2 Mandviwalla A, Ohshiro K, Ji B. Implementing Grover's algorithm on the IBM quantum computers. Proceedings of the 2018 IEEE International Conference on Big Data (Big Data). Seattle: IEEE, 2018. 2531–2537.

3 Brown KL, Munro WJ, Kendon VM. Using quantum computers for quantum simulation. Entropy, 2010, 12(11): 2268–2307. [doi: 10.3390/e12112268]

4 De Luca G. A survey of NISQ era hybrid quantum-classical machine learning research. Journal of Artificial Intelligence and Technology, 2022, 2(1): 9–15.

5 Matsuo A, Yamashita S, Egger DJ. A SAT approach to the initial mapping problem in SWAP gate insertion for commuting gates. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2023, E106. A(11): 1424–1431.

6 Zhu PC, Feng SG, Guan ZJ. An iterated local search methodology for the qubit mapping problem. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, 2022, 41(8): 2587–2597. [doi: 10.1109/TCAD.2021.3112143]

7 Wille R, Lye A, Drechsler R. Optimal SWAP gate insertion for nearest neighbor quantum circuits. Proceedings of the 19th Asia and South Pacific Design Automation Conference (ASP-DAC). Singapore: IEEE, 2014. 489–494.

8 黄泓凯, 张雪松. 含噪中型量子计算机的量子比特映射算法. 计算机工程与应用, 2024, 60(24): 110–118.

9 Ash-Saki A, Alam M, Ghosh S. QURE: Qubit re-allocation in noisy intermediate-scale quantum computers. Proceedings of the 56th ACM/IEEE Design Automation Conference. Las Vegas: IEEE, 2019. 1–6.

10 Li H, Lu K, Qin H, *et al*. Research on qubit mapping technique based on batch SWAP optimization. International Journal of Advanced Computer Science and Applications, 2023, 14(12): 172–180.

11 Wille R, Burgholzer L. MQT QMAP: Efficient quantum circuit mapping. Proceedings of the 2023 International Symposium on Physical Design. New York: ACM, 2023. 198–204.

12 Li GS, Ding YF, Xie Y. Tackling the qubit mapping problem for NISQ-era quantum devices. Proceedings of the 34th International Conference on Architectural Support for Programming Languages and Operating Systems. Providence: ACM, 2019. 1001–1014.

13 李晖, 韩子傲, 卢凯, 等. 改进量子位初始映射的综合 SWAP 优化策略. 计算机工程与应用, 2024, 60(14): 66–73.

14 Ruan Y, Marsh S, Xue XL, *et al*. Quantum approximate algorithm for NP optimization problems with constraints. arXiv:2002.00943, 2020.

15 朱鹏程, 卫丽华, 冯世光, 等. 面向分布式超导量子计算架构的量子线路映射. 软件学报, 2025, 36(5): 2381–2400. [doi: 10.13328/j.cnki.jos.007167]

16 Siraichi MY, dos Santos VF, Collange C, *et al*. Qubit allocation. Proceedings of the 2018 International Symposium on Code Generation and Optimization. Vienna: ACM, 2018. 113–125.

17 Sünkel L, Martyniuk D, Mattern D, *et al*. GA4QCO: Genetic algorithm for quantum circuit optimization. arXiv:2302.01303, 2023.

18 Steinberg MA, Feld S, Almudever CG, *et al*. Topological-graph dependencies and scaling properties of a heuristic qubit-assignment algorithm. IEEE Transactions on Quantum Engineering, 2022, 3: 3101114.

19 Liu H, Zhang BJ, Zhu Y, *et al*. QM-DLA: An efficient qubit mapping method based on dynamic look-ahead strategy. Scientific Reports, 2024, 14(1): 13118. [doi: 10.1038/s41598-024-64061-0]

20 韩子傲, 李晖, 卢凯, 等. 遗传启发映射策略在量子电路优化中的应用. 计算机工程与应用, 2025, 61(5): 94–103.

21 Lao LL, Browne DE. 2QAN: A quantum compiler for 2-local qubit hamiltonian simulation algorithms. Proceedings of the 49th Annual International Symposium on Computer Architecture. New York: ACM, 2022. 351–365.

22 Liu L, Dou XL. QuCloud: A new qubit mapping mechanism for multi-programming quantum computing in cloud environment. Proceedings of the 2021 IEEE International Symposium on High-performance Computer Architecture (HPCA). Seoul: IEEE, 2021. 167–178.

23 张辰逸, 尚涛, 刘建伟. 基于交换门的前瞻启发式量子线路映射算法. 电子科技大学学报, 2023, 52(4): 489–497.

24 Murali P, Baker JM, Javadi-Abhari A, *et al*. Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems. Providence: ACM, 2019. 1015–1029.

25 Sivarajah S, Dilkes S, Cowtan A, *et al*. t|ket⟩: A retargetable compiler for NISQ devices. Quantum Science and Technology, 2020, 6(1): 014003.

26 Paltenghi M, Pradel M. Analyzing quantum programs with lintq: A static analysis framework for qiskit. Proceedings of the ACM on Software Engineering, 2024, 1(FSE): 2144–2166. [doi: 10.1145/3660802]

(校对责编: 李慧鑫)