



基于置换群的多粒子环上量子行走的反馈搜索算法

姜瑶瑶 张文彬 初鹏程 马鸿洋

Feedback search algorithm for multi-particle quantum walks over a ring based on permutation groups

Jiang Yao-Yao Zhang Wen-Bin Chu Peng-Cheng Ma Hong-Yang

引用信息 Citation: *Acta Physica Sinica*, 71, 030201 (2022) DOI: 10.7498/aps.71.20211000

在线阅读 View online: <https://doi.org/10.7498/aps.71.20211000>

当期内容 View table of contents: <http://wulixb.iphy.ac.cn>

您可能感兴趣的其他文章

Articles you may be interested in

基于DNA编码与交替量子随机行走的彩色图像加密算法

Color image encryption algorithm based on DNA code and alternating quantum random walk

物理学报. 2021, 70(23): 230302 <https://doi.org/10.7498/aps.70.20211255>

基于引力搜索算法的湍流相位屏生成方法

Generation of turbulence phase screen based on gravitational search algorithm

物理学报. 2019, 68(13): 134205 <https://doi.org/10.7498/aps.68.20190081>

机器学习辅助绝热量子算法设计

Machine learning assisted quantum adiabatic algorithm design

物理学报. 2021, 70(14): 140306 <https://doi.org/10.7498/aps.70.20210831>

基于Cayley图上量子漫步的匿名通信方案

Anonymous communication scheme based on quantum walk on Cayley graph

物理学报. 2020, 69(16): 160301 <https://doi.org/10.7498/aps.69.20200333>

蚁群元胞优化算法在人群疏散路径规划中的应用

Application research of ant colony cellular optimization algorithm in population evacuation path planning

物理学报. 2020, 69(8): 080504 <https://doi.org/10.7498/aps.69.20191774>

基于量子游走的仲裁量子签名方案

Arbitrated quantum signature scheme based on quantum walks

物理学报. 2019, 68(12): 120302 <https://doi.org/10.7498/aps.68.20190274>

基于置换群的多粒子环上量子行走的 反馈搜索算法*

姜瑶瑶¹⁾ 张文彬²⁾ 初鹏程¹⁾ 马鸿洋^{1)†}¹⁾ (青岛理工大学理学院, 青岛 266033)²⁾ (青岛理工大学信息与控制工程学院, 青岛 266033)

(2021年5月27日收到; 2021年9月27日收到修改稿)

在量子计算科学中, 如何更好地构建量子搜索算法一直以来受到学者们的广泛关注, 并且基于量子行走寻找新的搜索算法也仍吸引着学者们不断深入研究与探索. 本文从减少搜索过程中的时间消耗、增加算法搜索的准确性和可控性等多方面进行考虑, 提出了一种基于置换群的多粒子量子行走搜索算法. 首先分析得到置换群在空间中可看成是一个闭环, 定义了置换集合, 并且通过同构映射将数据点所在数据集映射到定义的置换集, 使得置换集中元素数据点形成一一对应的关系. 其次, 根据给定初始态和硬币算符, 在数据点集与置换集合张成的搜索空间中利用多粒子的量子行走在环上进行目标数据搜索. 最后, 根据函数 $\phi(w) = 1$ 找到目标数据, 并用量子态存储数值, 用于形成搜索算法的反馈控制; 同时通过控制硬币算符从而控制量子行走在环上的行走方向, 增加搜索的可操作性与准确性. 本文利用多粒子的量子行走进行搜索, 分析得到粒子数量参数 j 与时间复杂度呈非线性负相关; 提出的量子行走搜索算法符合零点条件与下确界条件, 且不受变量数 j 的影响; 通过数值分析得到量子行走搜索算法的时间复杂度等价于 $O(\sqrt[3]{N})$, 相比于 Grover 搜索算法提高了搜索效率.

关键词: 搜索算法, 量子行走, 置换群, 同构映射**PACS:** 02.10.Ab, 03.67.Ac**DOI:** 10.7498/aps.71.20211000

1 引言

量子行走作为经典行走的推广, 是量子计算的基础, 被视为实现量子计算的一种主要工具. 量子行走利用量子态的叠加性, 能够实现同时行走在不同线路上的可能性, 因此与经典行走相比, 量子行走的效率有着指数级的增长. 并且行走者位置的概率分布也与经典行走有着截然不同的形式, 这些性质都有利于量子算法的实现. 量子行走最初是由

Aharonov 等^[1] 及 Farhi 和 Gutmann^[2] 提出的. 其中 Aharonov 等^[1] 提出了离散时间状态下的量子行走算法; Farhi 和 Gutmann^[2] 提出了连续时间状态下的量子行走算法. 与经典行走相比, 这两种行走方法都提供了加速效果. Godsil 等^[3] 和 Bose^[4] 展示了如何将图论的思想应用到量子行走中来实现更好的状态传输; Childs^[5] 证明了量子行走可以实现普适的量子计算. 这些成果都体现了量子行走在实际应用中的重要性. 此外, 基于量子行走的许多搜索算法都体现了量子行走在算法领域具有独

* 国家自然科学基金 (批准号: 11975132, 61772295)、山东省自然科学基金 (批准号: ZR2019YQ01) 与山东省高等教育科技计划 (批准号: J18KZ012) 资助课题.

† 通信作者. E-mail: hongyang_ma@aliyun.com

特的优势^[6-12].

在搜索算法中, Grover 算法是第二个量子革命的一个里程碑^[13]. 除了数据库搜索, 它还可以用于求解特征值问题, 这是目前最热门的课题之一. 2010 年 Childs^[14] 提出基于连续时间量子漫步的算法, 在黑箱问题中量子行走成功实现了指数级的加速. Shenvi 等^[15] 也在超立方体的拓扑知识的基础上提出了基于离散时间量子漫步的搜索算法, 实现了多项式加速效果. 由于搜索算法以及可转化为搜索问题的算法具有广泛适用性的特点, 在这些开创性的工作之后, 学者们又进行了深入的研究^[14,16-23]. 在国内, Long 等^[24-26] 不断完善优化 Grover 搜索算法, 他们研究的算法是所有优化的 Grover 算法中最优的, 又称龙算法^[26]; Zhou 等^[27-30] 和 Sheng 等^[31,32] 也分别在量子搜索算法和量子计算领域做出了贡献. 考虑到量子计算的优势, 学者们期待大量新的量子算法的出现, 但事实证明这项任务很难, 值得继续研究和探索.

基于量子行走寻找新的算法仍然是持续努力的方向, 本文提出了一种基于置换群的空间量子行走搜索算法, 将置换群作为搜索算法和量子行走相结合的桥梁, 目的是减少搜索时间, 提高目标搜索的精度, 提高空间内的搜索效果. 根据置换群中的元素在几何中可以形成环的性质, 将量子行走应用到置换群中, 从而实现在置换群中的量子搜索. 首先通过设置同构映射, 将数据点集合映射到由多个置换群组成的置换集合中的元素, 为了能够实现环上的量子行走, 在同构映射的作用下使数据点集合和置换集合具有一对一的对应关系. 然后, 构造 Hilbert 空间, 通过控制硬币算子, 给定初始状态, 将搜索空间中的元素作为节点, 进行量子行走. 接着根据函数 $\varphi(w) = 1$ 得到目标点, 并用量子态存储函数值. 最后根据函数的数值形成算法的反馈控制, 以此来判断量子行走的方向以及确定行走的状态. 本文有 3 个创新点: 1) 将数据集同构到置换集, 实现了置换群上的量子行走; 2) 多粒子受控的量子行走; 3) 增加了反馈控制, 增强算法的可操作性

与可控性.

本文结构如下: 第 2 节简要介绍置换群 S_3 以及环上的量子行走; 第 3 节分析并实现量子行走搜索算法, 其中包括 10 个步骤; 第 4 节对量子行走搜索算法进行时间复杂度分析; 第 5 节根据时间复杂度进行数值仿真, 直观明了地体现数值结果; 第 6 节对算法进行总结.

2 相关工作

2.1 置换群 S_3

定义 2.1 (置换群): 有限集合到自身的一一映射称为一个置换. 有限集合 S 上的一些置换组成的集合, 在置换的乘法下所组成的群, 称为置换群^[33]. 任何一个有限群都同构于一个置换群. 因此, 可以把一切有限群都看成置换群. 任一置换可表示成若干不相交循环的乘积, 如 $(a_1, a_2, \dots, a_n) = \begin{pmatrix} \alpha_1 \alpha_2 \cdots \alpha_{n-1} \alpha_n \\ \alpha_2 \alpha_3 \cdots \alpha_n \alpha_1 \end{pmatrix}$ 称为置换的循环表示. 由于每个循环首尾相连, 因此可以看成是一个闭环.

本文置换群的选择是根据搜索空间中目标态的维数决定的. 由于本文提出的搜索算法是在三维空间中进行的, 因此只对置换群 S_3 进行研究, 若搜索空间是 n 维, 可以换成置换群 S_n , 置换群的选择并不会影响算法的时间复杂度. 其中

$$S_3 = \{(e), (ab), (ac), (bc), (abc), (acb)\}.$$

2.2 环上行走

假设群 G 是有限群, S 是该群的生成集合, 环 A 和群 G 存在一一对应关系, 若节点 g 和 g' 满足 $g' = gh$, 则存在一条边 (g, g') , 其中 $g \in G, h \in S$. 将环 A 中元素量子化:

$$h \in S \rightarrow |h\rangle \in \mathcal{H}_S, g \in G \rightarrow |g\rangle \in \mathcal{H}_G, \quad (1)$$

其中, \mathcal{H}_S 为硬币算符所在的 Hilbert 空间, \mathcal{H}_G 为量子行走所处的位置空间. 环上量子行走的演化算符为 $U = T(C \otimes I)$, I 为位置空间的单位算符, C 为硬币算符, T 为转移算符, 具体定义如下:

$$C = \sum_{h_1 h_2 \in H_S} |h_1\rangle \langle h_2|, \quad (2)$$

$$T = \left[|h_1\rangle \langle h_1| \otimes \sum_{g \in G} |gh_1\rangle \langle g| + |h_2\rangle \langle h_2| \otimes \sum_{g \in G} |gh_2\rangle \langle g| \right]. \quad (3)$$

3 量子行走搜索算法

3.1 初始化阶段

步骤1 应用置换群的对称运算

在介绍量子行走搜索算法之前首先利用对称运算证明置换群 S_3 的每个子群在空间中形成一个闭环, 并且根据旋转角度, 规定群内元素的次序. 为了便于下文解释说明, 将群中元素 a, b, c 替换为 $1, 2, 3$.

对于置换群 S_3 , 有如下保持三角形不变的对称运算: (e)是一个恒等变化; (123), (132)分别为绕中心点逆时针旋转 $\frac{2\pi}{3}$ 和 $\frac{4\pi}{3}$; (12), (13), (23)分别为围绕 x, y, z 轴逆时针旋转 π . 每个元素旋转情况如图1所示. 同时可以得到置换群 S_3 的4个子群:

$$H_1 = \{e, (12)\}, \quad H_2 = \{e, (13)\},$$

$$H_3 = \{e, (123), (132)\}, \quad H_4 = \{e, (23)\}.$$

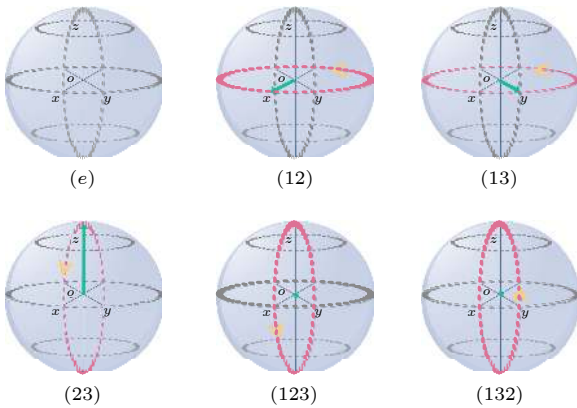


图1 置换群 S_3 中每个元素的旋转方位图
Fig. 1. Diagram of the rotation position of each element in the permutation group S_3 .

按照元素的旋转角度, 以单位元为起点, 规定每个子群的元素排列顺序. 例如在子群 H_1 中, 第一个元素是 e , 第二个元素是 (12). 同理可得其他子群元素之间的排列顺序. 置换群 S_3 的每个子群内, 各元素在三维空间中的旋转关系如图2所示. 在图2中, 对于子群 H_1 , 元素 e 绕 X 轴逆时针旋转 π 到达 (12) 的位置, 同样地, 元素 (12) 绕 X 轴逆时针旋转 π 到达 e 的位置; 同理可得其他子群元素之间位置的旋转关系. 从图2还可看出, 置换群 S_3 中每个子群元素之间通过置换群特有的对称运算, 形成了一个闭环.

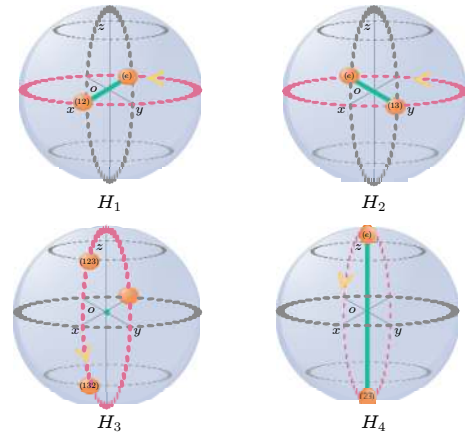


图2 在每个子群中各个元素之间的旋转关系图
Fig. 2. Diagram of the rotation relationship between the elements in each subgroup.

3.2 构造行走搜索空间

本节将数据点集与置换群元素通过同构映射实现一一对应, 从而构建量子行走的搜索空间.

步骤2 构建置换群元素新集合

定义 3.1 (数据集): 由数据点 d_1, d_2, \dots, d_N 生成的集合称为数据集 D , 其中 n 表示量子态数量, $N = 2^n$.

定义 3.2 (置换集): 集合 $P\{p(i, j, k)\}, i, j, k \in Z$ 是由元素 $p(i, j, k)$ 生成的, 称为置换集. 其中元素 $p(i, j, k)$ 通过旋转一定角度变成元素 $p(i, j, k + 1)$, 并且集合 P 中的元素数量大于等于数据集 D 中的元素数量.

构建置换集的具体过程: 由置换群的子群 H_i 衍生出子群族 H_i^j . 对于固定的 i, j , 子群 H_i^j 由 $p(i, j, k)$ 元素生成. 其中 $i = 1, 2, 3, j \in Z$. 对于子群 H_i^j , j 表示第 j 个子群 H_i , $k \in Z$ 表示子群 H_i^j 第 k 个元素. 对于同一个 i 的值, 只要 $j_1 \neq j_2$, 则 $H_i^{j_1} \neq H_i^{j_2}$. 同理, 只要 $k_1 \neq k_2$, 则 $p(i, j, k_1) \neq p(i, j, k_2)$.

对于集合 P 和每个子群 H_i^j , 即使包含的元素数与元素的性质相同, 我们也规定, 只要 j 是不同的, 就被认为是不同的子群. 此外, 对于同一个元素 (如 e), 只要子群不同, 就被认为是不同的元素. 并且每个子群的元素顺序是逆时针方向. 如对于子群 H_1^j , 指定第一个元素为 e , 第二个元素为 (12). 对于 H_1^j 的子群, 元素 e 标记为 $p(1, j, 1)$, (12) 标记为 $p(1, j, 2)$. 集合 $P = p(i, j, k)$ 中的每个元素都可以唯一地表示. 元素在置换集中的分布如表1所列.

表 1 置换集合元素分布情况

Table 1. Distribution of the elements in permutation set.

子群	子群族	群内所在元素
H_1	H_1^1	$p(1, 1, 1), p(1, 1, 2)$
	\vdots	\vdots
	$H_1^{j_1}$	$p(1, j_1, 1), p(1, j_1, 2)$
H_i	H_i^1	$p(i, 1, 1), p(i, 1, 2)$
	\vdots	\vdots
	$H_i^{j_i}$	$p(i, j_i, 1), p(i, j_i, 2)$

又因为

$$P = \bigcup_{j=1}^{j_i} \bigcup_{i=1}^4 H_i^j, \quad (4)$$

因此得到置换集中的元素数满足 $N = \sum_{i=1}^4 j_i k_i = 2^n$, 其中 j_i 表示 H_i 的子群数, k_i 表示元素数.

步骤 3 建立置换集与数据集同构

假设映射 \mathcal{F}

$$\mathcal{F}(d_m) = p^m(i, j, k), \quad \mathcal{F} : D \rightarrow P, \quad (5)$$

$p^m(i, j, k)$ 上标中的 m 无意义, 只是为了方便区分说明, 其中 $m = 1, 2, \dots, N$. 接下来证明, 映射 \mathcal{F} 是同构的.

证明 令 $\mathcal{F}(d_a) = p^a(i, j, c), \mathcal{F}(d_b) = p^b(i, j, d)$.

由于 $p^m(i, j, d)$ 的位置只对应于数据点 d^m , 因此当 $a \neq b$ 时, $p(i, j, c) \neq p(i, j, d) \Rightarrow d_a \neq d_b$. 证得 \mathcal{F} 是单射.

又因为 $d_a \neq d_b (a \neq b)$, 由元素的唯一性得到 $p(i, j, c) \neq p(i, j, d)$, 所以映射 \mathcal{F} 是同构映射.

步骤 4 建立量子行走搜索空间

定义 3.3 (搜索空间): 空间 \mathcal{W} 是 $W = D \times P$ 集合生成, 其元素定义为 $w(d_m, p^m(i, j, k)) \in \mathcal{W}$, 被称为搜索空间 \mathcal{W} . 每个元素 $w(d_m, p^m(i, j, k))$ 包含排列集中的数据点 d_m 和相应的位置 $p^m(i, j, k)$.

重新构造搜索函数 $\Phi(w) = \Phi(d) = \{0, 1\}$. 由于 $p^m(i, j, k)$ 只充当 $w(d_m, p^m(i, j, k))$ 中的位置坐标, 该函数对 $p^m(i, j, k)$ 不作用. 这样, 通过函数 $\Phi(w) = 1$ 得到 $w_{\text{tar}}(d_{\text{tar}}, p^{\text{tar}}(i, j, k))$, 从而得到目标数据 d_{tar} .

定义 3.4 (集合 W_{-1}): 搜索空间 \mathcal{W} 中定义了一个集合 $W_{-1} = \{w_{-1}(-1, p(i, j, k))\}$. 该集合与 W 的集合相比, 在相同位置 $p(i, j, k)$ 上, 集合 W_{-1} 中的数据 $d = -1$. 集合 W 与 W_{-1} 的元素对应关系如图 3 所示.

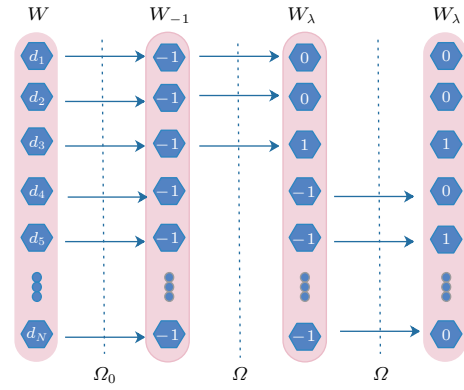


图 3 集合 W, W_{-1} 和 W_{λ} 元素之间的对应关系

Fig. 3. Corresponding relationship between elements of the sets W, W_{-1} and W_{λ} .

集合 W 中的元素和集合 W_{-1} 中的元素满足下列关系:

$$w_{-1} = \Omega_0(w) = \varphi_0 w, \quad (6)$$

其中 Ω_0 是转化函数; φ_0 是矩阵并满足

$$\varphi_0 = \begin{pmatrix} -1/\|d_1\| & 0 & \cdots & 0 & 0 \\ 0 & -1/\|d_2\| & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & -1/\|d_N\| \end{pmatrix}.$$

3.3 确定行走粒子数量

步骤 5 确定量子行走粒子数量

根据置换集合对应的子集数目, 确定量子行走过程中粒子的数量. 再根据前面的假设, 得到子集数 $j_1 + j_2 + j_3 + j_4$. 为了方便计算, 选择 $j_1 + j_2 + j_3 + j_4$ 的最大值 $4 \times \max\{j_1, j_2, j_3, j_4\}$. 在不失一般性的情况下, 令 $j = \max\{j_1, j_2, j_3, j_4\}$. 得到 $j_1 + j_2 + j_3 + j_4 \leq 4 \times \max\{j_1, j_2, j_3, j_4\} = 4 \times j$.

因此, 得到子集的数目是 $4 \times j$, 即量子行走过程中粒子的数目是 $4 \times j$, 其中 $0 \leq j \leq N/4$.

3.4 行走过程

前文得出置换群 S_3 的每个子群在空间中形成一个闭环, 并且置换群也与 Z_n 的加法群同构. 根据已经提出的基于 Cayley 图的量子行走算法 [34], 可以得到置换群 S_3 上的量子行走.

步骤 6 置换群上量子行走

将置换群看成是闭环, 则它的数学描述如下: 假设 G 是一个有限群, S 是该群的生成集合, 置换群 S_3 的元素和群 G 的元素之间存在一一对应关系.

并且如果两个节点 g 和 g' 满足 $g' = gh$, 其中 $g \in G$ 和 $h \in S$, 则这两个节点之间存在一条边 (g, gh) . 将置换群的元素量子化, 则置换群上的量子行走可以有如下定义: 假设 \mathcal{H}_S 是硬币算子所在的 Hilbert 空间, 它是由态 $|h\rangle$, $h \in S$ 生成的; \mathcal{H}_G 是行走者的位置空间, 它是由态 $|g\rangle$, $g \in G$ 生成, 则演化算符 $U = T(C \otimes I)$ 的硬币算子 C 和转移算子 T 分别定义为

$$C = \sum_{h_1 h_2 \in S} |h_1\rangle\langle h_2|, \quad (7)$$

$$T = \sum_{i=1,2} \left(|h_i\rangle\langle h_i| \otimes \sum_{g \in G} |gh_i\rangle\langle g| \right). \quad (8)$$

其中, 转移算子的作用是

$$T|h_1\rangle|g\rangle = |h_1\rangle|gh_1\rangle, \quad (9)$$

从 (9) 式可以得到, 在置换群上进行量子行走时, 以 g 为起点位置, 当硬币算符为 h_1 时, 行走者会由节点位置 g 转移到相邻节点 gh_1 , 其中 $gh_1 = g'$. 根据相邻节点的关系, 可以得到:

$$g \oplus h_1 = g', \quad (10)$$

其中 \oplus 是模 2 加运算或模 3 加运算 (在 H_3 中).

对群元素进行傅里叶变换^[35], 算子的形式如下:

$$F = \frac{1}{\sqrt{G}} \sum_{k,g \in G} \chi_g(k) |g\rangle\langle k|, \quad (11)$$

其中, χ_g 为群的特征标, $\chi_g = \prod_{j=1}^n \omega_{N_j}^{\varepsilon, k_j}$, $\omega_{N_j} = e^{\frac{2\pi}{N_j}}$.

将位置空间的基态转换为傅里叶基态: $|\tilde{\chi}_k\rangle = \frac{1}{\sqrt{G}} \sum_{g \in G} \chi_g(k^{-1}) |g\rangle$, 则在 t 时刻量子行走的状态用傅里叶变换后的基态表示为

$$|\Psi(t)\rangle = \sum_{h \in S} \sum_{k, g \in G} \tilde{\psi}_{h,g}(t) |\tilde{\chi}_k\rangle, \quad (12)$$

其中, 转移算符对傅里叶基态作用后的形式为

$$T|h\rangle|\tilde{\chi}_g\rangle = \chi_h(g) |h\rangle|\tilde{\chi}_g\rangle. \quad (13)$$

可以证明傅里叶基态下, 转移算符只改变基态的振幅.

最后, 得到傅里叶基态下 t 时刻的振幅为 $\tilde{\psi}_{h,g}(t)$, 通过逆傅里叶变换求解离散时间的振幅:

$$\Theta = \begin{pmatrix} 1 & -(2)^{\Phi(w_{m3})} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \Rightarrow \Theta^\dagger = \begin{pmatrix} 1 & 0 & 0 \\ -(2)^{\Phi(w_{m3})} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (18)$$

得到

$$\psi_{h,g}(t) = \sum_{g \in G} \frac{\chi_g(g^{-1})}{\sqrt{|G|}} \tilde{\psi}_{h,g}(t). \quad (14)$$

3.4.1 构造 Hilbert 空间

步骤 7 构造 Hilbert 空间并检测合理性

构造 Hilbert 空间 $\mathcal{H} = \mathcal{H}_S^{4j} \otimes \mathcal{H}_G^{2^n} \otimes \mathcal{W}$. 其中 \mathcal{H}_S 是硬币算子所在的 Hilbert 空间. 两个正交向量 $|1\rangle$ 和 $|-1\rangle$ 可以用来表示硬币算子的状态; ($|1\rangle$ 表示顺时针方向, $|-1\rangle$ 表示逆时针方向). \mathcal{H}_G 是量子行走的位置空间; $4j$ 表示量子态的数量; $N = 2^n$; \mathcal{W} 是搜索空间, 用来存储函数值 $\Phi(w) = \{0, 1\}$.

确定硬币算符 C , 通过硬币算符控制每一步行走. 这一步的目标是保证量子行走的方向一致, 不会往返. 然后通过迭代算子 $U = T(C \otimes I) \otimes \Theta$, 进而通过迭代的方法得出迭代算子的数值解. 其中当算子 Θ 作用到元素 $w(d_m, p^m(i, j, k))$ 时, 可以得到数值 $\delta = \{1, 0\}$. 根据硬币算子 C 在本文中的作用, 其被定义为

$$C = \sum_h |-1\rangle\langle h|, h \in S, \quad (15)$$

其中 $S = \{-1, 1\}$. 并且令:

$$\Theta = \begin{pmatrix} 1 & -(2)^{\Phi(w_{m3})} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (16)$$

根据构建的 Hilbert 空间, 检验转移算子 T 、转换算子 Θ 和迭代算子 U_i 是酉算子, 其中 $i = 1, 2, 3$. 推导结果如下:

$$\begin{aligned} T^\dagger &= \sum_{i=1,2} \left(|h'_i\rangle\langle h'_i| \otimes \sum_{g' \in G} |g'\rangle\langle g'h'_i| \right), \\ T^\dagger T &= \left[\sum_{i=1,2} \left(|h'_i\rangle\langle h'_i| \otimes \sum_{g' \in G} |g'\rangle\langle g'h'_i| \right) \right] \\ &\quad \times \left[\sum_{i=1,2} \left(|h_i\rangle\langle h_i| \otimes \sum_{g \in G} |gh_i\rangle\langle g| \right) \right] \\ &= I. \end{aligned} \quad (17)$$

因为转换算子

$$\Theta^\dagger \Theta = \begin{pmatrix} 1 & 0 & 0 \\ -(2)^{\Phi(w_{m3})} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -(2)^{\Phi(w_{m3})} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{I}. \quad (19)$$

又因为

$$\begin{aligned} & [(C \otimes \mathbf{I}) \otimes \Theta]^\dagger \cdot [(C \otimes \mathbf{I}) \otimes \Theta] \\ &= \left[\sum_{h \in S} |-1\rangle \langle h| \otimes \mathbf{I} \otimes \begin{pmatrix} 1 & -(2)^{\Phi(w_{m3})} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right]^\dagger \cdot \left[\sum_{h \in S} |-1\rangle \langle h| \otimes \mathbf{I} \otimes \begin{pmatrix} 1 & -(2)^{\Phi(w_{m3})} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right] \\ &= \left[\sum_{h \in S} |h\rangle \langle -1| \otimes \mathbf{I} \otimes \begin{pmatrix} 1 & 0 & 0 \\ -(2)^{\Phi(w_{m3})} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right] \cdot \left[\sum_{h \in S} |-1\rangle \langle h| \otimes \mathbf{I} \otimes \begin{pmatrix} 1 & -(2)^{\Phi(w_{m3})} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right] = \mathbf{I}, \quad (20) \end{aligned}$$

根据 $U = T(C \otimes \mathbf{I}) \otimes \Theta$, 得到:

$$\begin{aligned} U^\dagger U &= [T(C \otimes \mathbf{I}) \otimes \Theta]^\dagger [T(C \otimes \mathbf{I}) \otimes \Theta] \\ &= [(C \otimes \mathbf{I}) \otimes \Theta]^\dagger \cdot T^\dagger T \cdot [(C \otimes \mathbf{I}) \otimes \Theta] = \mathbf{I}, \quad (21) \end{aligned}$$

即 $U_i^\dagger U_i = \mathbf{I}$.

因此通过上述验证过程得出转移算子 T 、转换算子 Θ 和迭代算子 U_i 是酉算子.

3.4.2 执行行走过程

步骤 8 分析量子行走路径

给定初始态 $|\psi_0\rangle = \frac{1}{\sqrt{N/(4j)}} \sum_1^{[N/(4j)]} |-1\rangle \otimes |0\rangle \otimes |\delta\rangle$, 其中 $\delta = 1, 0$, 符号 $[\]$ 表示取整函数. 通过迭代算子 $U = T(C \otimes \mathbf{I}) \otimes \Theta$ 可以得到:

$$U_i = \psi_{t=1} = |-1\rangle \otimes |-i\rangle \otimes |\delta\rangle,$$

从而控制了置换群中的量子行走方向. 整个量子行走过程如图 4 所示.

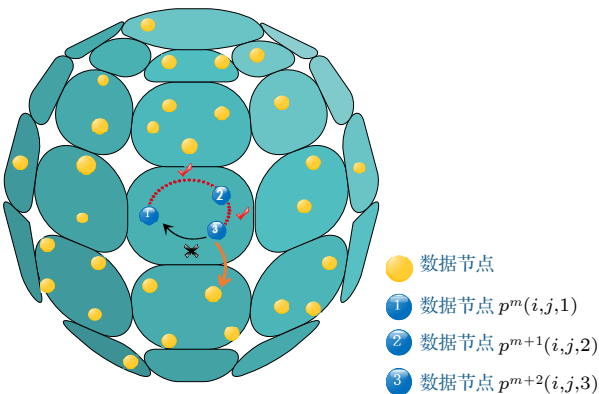


图 4 量子行走的过程

Fig. 4. Process of quantum walk.

图 4 中黄色的圆点表示数据点, 绿色的球体表示空间, 标有数字 1, 2, 3 的蓝色圆圈表示数据点 $p^m(i, j, 1)$, $p^{m+1}(i, j, 2)$, $p^{m+2}(i, j, 3)$, 位置在某个子群 H_i^j 中. 红色的虚线表示运动轨迹, 从 $1 \rightarrow 2 \rightarrow 3$ 都是利用反馈控制继续前进的. 当 $3 \rightarrow 1$ 时, 运动轨迹就是黑色箭头所表示的方向, 通过反馈控制, 发现此方向是被禁止的, 于是设定硬币算符为 $C^* = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, 随机选择行走方向 (橙色箭头), 进行下一个子群中进行的量子行走.

3.5 储存数值结果

步骤 9 存储函数结果形成反馈控制

当 Φ 作用于元素 $w(d_m, p^m(i, j, k))$ 时, 得到函数的值 $\delta = 1$ 或 0 , 并将数值存储在量子态中. 其中对于量子态 $|1\rangle$, 表示所对应的数据 d 正是目标数据 d_{tar} . 为了形成含有数值结果的量子态集合, 让 δ 替换 W_{-1} 集合中的 -1 .

更换过程如下:

$$\begin{aligned} \delta_{m1}^{H_1} &= \Omega_1(w_{-1}) = \phi_1 w_{-1} = \Theta_1(w_{m1}^{H_1}) = \varphi_1 w_{m1}^{H_1}, \\ \delta_{m2}^{H_2} &= \Omega_2(w_{-1}) = \phi_2 w_{-1} = \Theta_2(w_{m2}^{H_2}) = \varphi_2 w_{m2}^{H_2}, \\ \delta_{m3}^{H_3} &= \Omega_3(w_{-1}) = \phi_3 w_{-1} = \Theta_3(w_{m3}^{H_3}) = \varphi_3 w_{m3}^{H_3}, \\ \delta_{m4}^{H_4} &= \Omega_4(w_{-1}) = \phi_4 w_{-1} = \Theta_4(w_{m4}^{H_4}) = \varphi_4 w_{m4}^{H_4}, \end{aligned} \quad (22)$$

其中, $\Omega_i(\Theta_i)$ 表示转换函数, 使得数据 $d = -1(d)$ 成为 δ , $w_{mi}^{H_i}$ 表示 H_i 中的一个元素, $\delta_{mi}^{H_i}$ 表示 w_{mi} 在函数 Φ 的作用下的函数值, $i = 1, 2, 3, 4$. ϕ_i 是矩阵

并满足

$$\begin{aligned} \phi_1 &= \begin{pmatrix} 1 & -(2)^{\Phi(w_{m1})} \\ 0 & 1 \end{pmatrix}, \quad \phi_2 = \begin{pmatrix} 1 & -(2)^{\Phi(w_{m2})} \\ 0 & 1 \end{pmatrix}, \\ \phi_3 &= \begin{pmatrix} 1 & -(2)^{\Phi(w_{m3})} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \phi_4 = \begin{pmatrix} 1 & -(2)^{\Phi(w_{m4})} \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

φ_i 是矩阵并满足

$$\begin{aligned} \varphi_1 &= \begin{pmatrix} \frac{-1}{\|d_{m1}\|} & \frac{(2)^{\Phi(w_{m1})}}{\|d_{m1+1}\|} \\ 0 & \frac{-1}{\|d_{m1+1}\|} \end{pmatrix}, \quad \varphi_2 = \begin{pmatrix} \frac{-1}{\|d_{m2}\|} & \frac{(2)^{\Phi(w_{m2})}}{\|d_{m2+1}\|} \\ 0 & \frac{-1}{\|d_{m2+1}\|} \end{pmatrix}, \\ \varphi_3 &= \begin{pmatrix} \frac{-1}{\|d_{m3}\|} & \frac{(2)^{\Phi(w_{m3})}}{\|d_{m3+1}\|} & 0 \\ 0 & \frac{-1}{\|d_{m3+1}\|} & 0 \\ 0 & 0 & \frac{-1}{\|d_{m3+2}\|} \end{pmatrix}, \quad \varphi_4 = \begin{pmatrix} \frac{-1}{\|d_{m4}\|} & \frac{(24)^{\Phi(w_{m24})}}{\|d_{m4+1}\|} \\ 0 & \frac{-1}{\|d_{m4+1}\|} \end{pmatrix}. \end{aligned}$$

当函数 Ω_i (图 3 中简称为 Ω) 在进行数据更新时, 存储函数值的量子态会出现 3 个数值, 分别是 $|1\rangle$, $|0\rangle$ 和 $|-1\rangle$. 为了方便说明, 用量子态 $|\lambda\rangle$ 表示, 并且建立集合 $W_\lambda = \{(\lambda, p(i, j, k))\}$ ($W_{-1} \rightarrow W_\lambda$), 其中 $\lambda = 0, 1, -1$. 集合 W , W_{-1} 和 W_λ 三者之间的关系变化如图 3 所示.

3.6 反馈结果判定方向

步骤 10 根据反馈结果判定后续方向

当量子行走进行到某步时, 设此时的迭代次数是 t . 当迭代次数是 $t+1$ 时, 判断量子行走过程是否继续或者停止. 判断过程如下:

当行走到位置 $p^m(i, j, k)$ 时, 判断相应集合 W_λ 对应的数据 λ , 如果 $\lambda < 0$, 量子行走在此环 (置换群) 中继续进行; 如果 $\lambda \geq 0$, 量子行走将在此置换群中停止, 设定硬币算符为 $C^* = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, 随机行走到其他置换群的位置.

4 复杂度分析

本文采取多粒子的量子行走, 这样时间复杂度取决于数据点的数量 N 和量子行走的数量 j , 即时间复杂度 $t = t_{in} + t_{out}$ 最终取决于参数 j 和 N .

对于量子态 $|\chi\rangle = \frac{1}{\sqrt{N}} \sum_{x=1}^N |x\rangle \otimes |y\rangle \otimes |\delta\rangle$, 其

中 $y = 1, 2, 3$, $\delta = 0, 1$, 正交归一化后得到:

$$\begin{aligned} |\chi\rangle &= \sqrt{\frac{M}{N}} \left(\frac{1}{\sqrt{M}} \sum_{x \in \Phi^{-1}(1)} |x\rangle \otimes |y\rangle \otimes |\delta\rangle \right) \\ &+ \sqrt{\frac{N-M}{N}} \left(\frac{1}{\sqrt{N-M}} \sum_{x \in \Phi^{-1}(0)} |x\rangle \otimes |y\rangle \otimes |\delta\rangle \right), \end{aligned} \quad (23)$$

其中, M 表示目标点的数量.

分析时间复杂度 t_{in} , 根据 $C = \sum_h |-1\rangle \langle h|$, $h \in S$ 可以得到每一步的硬币算符是

$$C_1 = |-1\rangle \langle 1|, \quad C_2 = |-1\rangle \langle -1|, \quad C_3 = |-1\rangle \langle -1|,$$

其中, 两个正交向量 $|1\rangle$ 和 $|-1\rangle$ 可以用来表示硬币的状态. $|1\rangle$ 表示顺时针方向, $|-1\rangle$ 表示逆时针方向.

通过迭代方法, 可以得到:

$$\begin{cases} U_1 = \psi_{t=1} = |-1\rangle \langle -1| \otimes |\delta\rangle, \\ U_2 = \psi_{t=2} = |-1\rangle \langle -2| \otimes |\delta\rangle, \\ U_3 = \psi_{t=3} = |-1\rangle \langle -3| \otimes |\delta\rangle. \end{cases}$$

由于本文选择的是置换群 S_3 , 因此完成一次置换群上行走次数为 $t \leq 3$. 又因为在置换群中完成一次量子行走时, 才进行一次硬币算符为 C^* 的量子行走. 即完成一次更换需要的时间 $t = t_{in} + t_{out} \leq 3 + 1 = 4$, 因此完成所有行走过程时间复杂度为 $t = t_{in} + t_{out} \leq 4t_{out}$, 得到 $t = 4t_{out}$. 所以只需

要计算时间复杂度 t_{out} 即可.

令

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{x \in \Phi^{-1}(0)} |x\rangle \otimes |y\rangle \otimes |\delta\rangle,$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x \in \Phi^{-1}(1)} |x\rangle \otimes |y\rangle \otimes |\delta\rangle,$$

$$\sin \theta = \sqrt{M/N} = \sqrt{\kappa},$$

于是得到

$$|\chi\rangle = \sqrt{M/N}|\alpha\rangle + \sqrt{(N-M)/N}|\beta\rangle. \quad (24)$$

令迭代算符 $U = \mathbf{T}(\mathbf{C}^* \otimes \mathbf{I}) \otimes \Theta$ 作用到量子态 $|\chi\rangle$, 通过计算离散量子行走时刻连续叠加态的振幅得到时间复杂度 t_{out} 和参数 N 的关系.

$$\begin{aligned} U_t|\chi\rangle &= [\mathbf{T}(\mathbf{C}^* \otimes \mathbf{I}) \otimes \Theta]^t |\chi\rangle \\ &= [\mathbf{T}(\mathbf{C}^* \otimes \mathbf{I}) \otimes \Theta]^t (\cos \theta |\alpha\rangle + \sin \theta |\beta\rangle) \\ &= \frac{1}{\sqrt{N}} \sum_k X(t) |\alpha\rangle + \frac{1}{\sqrt{N}} \sum_k Y(t) |\beta\rangle, \end{aligned} \quad (25)$$

其中, 当 t_{out} 为偶数时:

$$\begin{aligned} X(t) &= \cos(\theta_k t) - \frac{i \cos \frac{2\pi k}{N} \sin(\theta_k t)}{\sqrt{1 + \cos^2(2\pi k/N)}}, \\ Y(t) &= -\frac{ie^{i\frac{2\pi}{N}} \sin(\theta_k t)}{\sqrt{1 + \cos^2(2\pi k/N)}}; \end{aligned} \quad (26)$$

当 t_{out} 为奇数时:

$$\begin{aligned} X(t) &= -i \sin(\theta_k t) - \frac{i \cos(2\pi k/N) \sin(\theta_k t)}{\sqrt{1 + \cos^2(2\pi k/N)}}, \\ Y(t) &= -\frac{ie^{i\frac{2\pi k}{N}} \sin(\theta_k t)}{\sqrt{1 + \cos^2(2\pi k/N)}}. \end{aligned} \quad (27)$$

θ_k 满足 $\sin \theta_k = \frac{1}{\sqrt{2}} \sin \frac{2\pi k}{N}$.

搜索目的是为了找遍空间中所有子群, 即搜索到目标数据点的概率等于 M/N .

得到:

$$\begin{aligned} &\frac{1}{N^2} \left| \sum_{j=1}^{\frac{N}{4}} \sum_{k=0}^{4j-1} Y_k(t) e^{ik} \right|^2 \\ &= \frac{M}{N} \Rightarrow \left| \sum_{j=1}^{\frac{N}{4}} \sum_{k=0}^{4j-1} Y_k(t) e^{ik} \right|^2 = MN. \end{aligned} \quad (28)$$

由于参数 j 与 N 有关, 若取 $j = \sqrt[3]{N}$. 求解方程 (28) 可得

$$\begin{aligned} &\left| \sum_{j=1}^{N/4} \sum_{k=0}^{4j-1} Y_k(t) e^{ik} \right|^2 = MN \\ &\Rightarrow \sum_{j=1}^{\frac{N}{4}} \sum_{k=0}^{4j-1} |Y_k(t) e^{ik}|^2 \leq MN \\ &\Rightarrow \sum_{j=1}^{N/4} \sum_{k=0}^{4j-1} \left| \frac{ie^{i\frac{2\pi k}{N}} \sin(\theta_k t)}{\sqrt{1 + \cos^2 \frac{2\pi k}{N}}} e^{ik} \right|^2 \leq MN \\ &\Rightarrow t^2 \sum_j \left\{ \frac{1}{2} [j^2(j+1)] + \left(\frac{\pi^2}{6N^2} - \frac{\pi^4}{27N^4} \right) \right. \\ &\quad \left. \times (j+1)^2 j^3 + \frac{2\pi^4}{27N^4} (j+1)^3 j^4 \right\} \leq MN \\ &\Rightarrow t^2 \left(\frac{14\pi^4}{27N^2} N^{7/3} \right) \leq MN \Rightarrow t \leq \frac{27M}{14\pi^4} \sqrt[3]{N}, \end{aligned} \quad (29)$$

得出关于变量 N 的数值表达式 $t_{\text{out}}(N) = O(\sqrt[3]{N})$, 即得到关于变量 N 的数值表达式 $t(N) = O(\sqrt[3]{N})$.

5 数值仿真

将本文提出的量子行走搜索算法与 Grover 搜索算法关于时间复杂度进行对比, 设定参数 $M = 10$, $N = 200$, 得到图 5 所示的对比曲线. 从图 5(a) 得到量子行走搜索算法所用时间要比 Grover 搜索算法所用时间短, 即量子行走搜索算法的速率更高; 从图 5(b) 得到量子行走搜索算法是满足原点条件和下确界条件的, 即满足当 $N = 0$ 时, $t = 0$; $N = 1$ 时, $t = 1$.

为了直观地体现算法的高效性, 下面用两种不同算法进行举例比较: 一个采用标准的 Grover-Long 算法 [36], 一个采用本文提出的量子行走搜索算法. $N = 64$ 时, Grover-Long 算法需要 $O(\sqrt{N}) = 8$ 次搜索, 采用本文的方法, 最多需要 $O(\sqrt[3]{N}) = 4$ 次搜索.

进而又对参数 j 对搜索的时间复杂度的影响进行了分析. 选取了参数 $M = 10$, $N = 200$ 分别对参数 $j = 1$, $j = 2$, $j = 3$ 三条曲线进行分析, 得到图 6 所示曲线. 从图 6(a) 得到参数 j 与时间复杂度呈负相关, 即参数 j 的取值越大, 所用时间越短, 搜索速率越快; 从图 6(b) 得到参数 j 不影响量子行走搜索算法, 且满足原点条件和下确界条件.

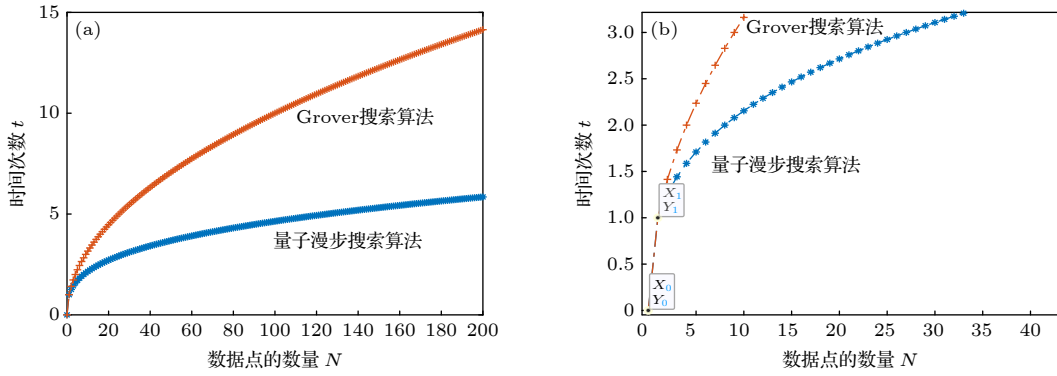


图 5 两种搜索算法时间按复杂度的对比

Fig. 5. Comparison of the time complexity of two search algorithms.

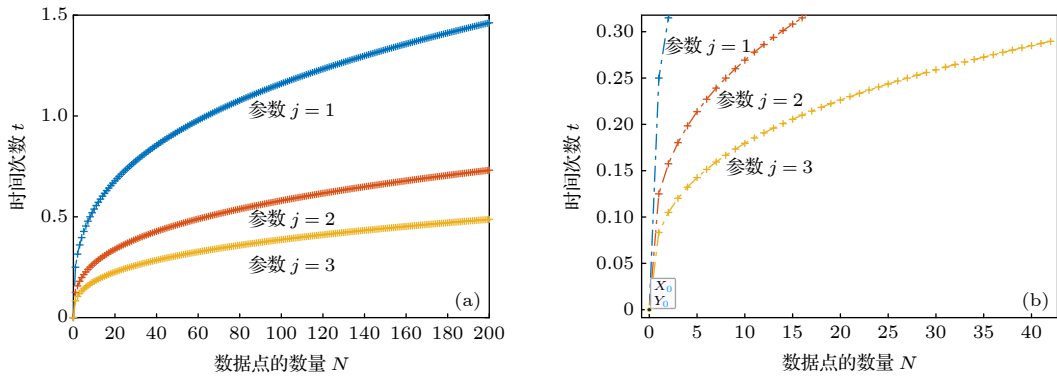


图 6 参数 j 对搜索算法时间复杂度的影响

Fig. 6. Influence of parameter j on the time complexity of search algorithm.

最后分析了变量参数 j , N 与时间复杂度 t 的关系, 得到数值仿真结果如表 2 所列. 从表中数据进一步得出, 参数与时间复杂度不是呈负线性关系.

表 2 数据仿真结果

Table 2. Numerical simulation results.

数据点数量 N	$j = 1$	$j = 2$	$j = 3$
1024	3	2	1
2048	4	2	2
4096	4	2	2
8192	6	3	2
16384	7	4	3
32768	8	4	3

6 总结

本文基于量子行走提出的搜索算法, 将时间复杂度减少到 $O(\sqrt[3]{N})$. 相对于 Grover 算法中时间复杂度等价于 $O(\sqrt{N})$, 降低了时间消耗; 通过对硬币算符进行控制, 从而控制了整个行走过程, 增加了算法的可控性; 通过增加反馈控制, 使得算法更具

有灵活性与可操作性. 最后通过模型仿真, 分析变量参数 j 与时间复杂度的关系得到, 参数 j 与时间复杂度呈非线性负相关; 提出的量子行走搜索算法符合零点条件与下确界条件, 即满足当 $N = 0$ 时, $t = 0$; $N = 1$ 时, $t = 1$; 且不受参数 j 的影响.

参考文献

- [1] Aharonov Y, Luiz D, Naim Z 1993 *Phys. Rev. A* **48** 1687
- [2] Farhi E, Gutmann S 1998 *Phys. Rev. A* **58** 915
- [3] Godsil C, Kirkland S, Severini S, Smith J 2012 *Phys. Rev. Lett.* **109** 050502
- [4] Bose S 2003 *Phys. Rev. Lett.* **91** 207901
- [5] Childs A M 2009 *Phys. Rev. Lett.* **102** 180501
- [6] Berry S D, Wang J B 2011 *Phys. Rev. A* **83** 042317
- [7] Zähringer F, Kirchmair G, Gerritsma R, Solano E, Blatt R, Roos C F 2010 *Phys. Rev. Lett.* **104** 100503
- [8] Ma H Y, Zhang X, Xu P A, Liu F 2020 *Wirel. Pers. Commun.* **113** 2203
- [9] Wojan P, Abeyesinghe A 2008 *Phys. Rev. A* **78** 042336
- [10] Orsucci D, Briegel H J, Dunjko V 2018 *Quantum* **2** 105
- [11] Chakraborty S, Novo L, Ambainis A, Omar Y 2016 *Phys. Rev. Lett.* **116** 100501
- [12] Chakraborty S, Novo L, Di Giorgio S, Omar Y 2017 *Phys. Rev. Lett.* **119** 220503
- [13] Chang C R, Lin Y C, Chiu K L, Huang T W 2020 *AAPPS*

Bull. **30** 9

- [14] Childs A M 2010 *Commun. Math. Phys.* **294** 581
- [15] Shenvi N, Kempe J, Whaley K B 2003 *Phys. Rev. A* **67** 052307
- [16] Sansoni L, Sciarrino F, Vallone G, Mataloni P, Crespi A, Ramponi R, Osellame R 2012 *Phys. Rev. Lett.* **108** 010502
- [17] Caruso F 2014 *New J. Phys.* **16** 055015
- [18] Liu F, Zhang X, Xu P A, He Z X, Ma H Y 2020 *Int. J. Theor. Phys.* **59** 3491
- [19] Dunjko V, Briegel H J 2015 *New J. Phys.* **17** 073004
- [20] Glos A, Krawiec A, Kukulski R, PuchaPuchała Z 2018 *Quantum Inf. Process.* **17** 1
- [21] Mülken O, Blumen A 2011 *Phys. Rep.* **502** 37
- [22] Long G L, Wang C, Deng F G, Zheng C 2013 *Conference on Coherence and Quantum Optics* Rochester, New York, USA, June 17–20, 2013 ppM6–42
- [23] Kempf A, Portugal R 2009 *Phys. Rev. A* **79** 052317
- [24] Zhou L, Sheng Y B, Long G L 2020 *Sci. Bull.* **65** 12
- [25] Zhou Z, Sheng Y, Niu P, Yin L, Long G L, Hanzo L 2020 *Sci. China: Phys. Mech. Astron.* **63** 1
- [26] Long G L 2001 *Phys. Rev. A* **64** 022307
- [27] Zhou N R, Huang L X, Gong L H, Zeng Q W 2020 *Quantum Inf. Process.* **19** 1
- [28] Zhou N R, Zhu K N, Zou X F 2019 *Ann. Phys.* **531** 1800520
- [29] Zhou N R, Zhu K N, Bi W, Gong L H 2019 *Quantum Inf. Process.* **18** 1
- [30] Li H H, Gong L H, Zhou N R 2020 *Chin. Phys. B* **29** 110304
- [31] Sheng Y B, Zhou L 2018 *Phys. Rev. A* **98** 052343
- [32] Sheng Y B, Zhou L 2017 *Sci. Bull.* **62** 1025
- [33] Zhang H R 1997 *Base of Recent Generations* (Vol. 19) (Beijing: Higher Education Press) p50 (in Chinese) [张禾瑞 1997 近世代数基础 (修订本) (北京: 高等教育出版社) 第50页]
- [34] Schreiber A, Cassemiro K N, Potoček V, Gábris A, Mosley P J, Andersson E, Jex I, Silberhorn Ch 2010 *Phys. Rev. Lett.* **104** 050502
- [35] Diaconis P, Rockmore D 1990 *J. Am. Math. Soc.* **3** 297
- [36] Toyama F M, Van Dijk W, Nogami Y 2013 *Quantum Inf. Process.* **12** p1897

Feedback search algorithm for multi-particle quantum walks over a ring based on permutation groups^{*}

Jiang Yao-Yao¹⁾ Zhang Wen-Bin²⁾ Chu Peng-Cheng¹⁾ Ma Hong-Yang^{1)†}

1) (*School of Science, Qingdao University of Technology, Qingdao 266033, China*)

2) (*School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266033, China*)

(Received 27 May 2021; revised manuscript received 27 September 2021)

Abstract

In quantum computing science, much attention has been paid to how to construct quantum search algorithms better, moreover, searching for new search algorithms based on quantum walk is still attracting scholars' continuous in-depth research and exploration. In this paper, a multi-particle quantum walk search algorithm based on permutation group is proposed by considering many aspects, such as reducing time consumption and increasing the accuracy and controllability of algorithm search. Firstly, the permutation group can be regarded as a closed loop in space, and the permutation set is defined. The data set of data points is mapped to the defined permutation set by isomorphism mapping, which makes the element data points in the permutation set form a one-to-one correspondence. Secondly, according to the given initial state and coin operator, the target data search is carried out on the ring by using the quantum walk of multiple particles in the search space of the data point set and the permutation set. Finally, the target data is found according to the function $\Phi(w) = 1$, and the value is stored by the quantum state, which is used to form the feedback control of the search algorithm. At the same time, the direction of quantum walking on the ring is controlled by controlling the coin operator, thus increasing the operability and accuracy of the search. In this paper, the quantum walk of multiple particles is used to search, and the analysis shows that the particle number parameter j is negatively correlated with the time complexity, but not negatively linear. The proposed quantum walk search algorithm conforms to the zero condition and the lower bound condition, and is not affected by the variable parameter j . Through numerical analysis, it is found that the time complexity of the quantum walk search algorithm is equivalent to $O(\sqrt[3]{N})$, which improves the search efficiency compared with the Grover search algorithm.

Keywords: search algorithm, quantum random walk, permutation group, isomorphic mapping

PACS: 02.10.Ab, 03.67.Ac

DOI: [10.7498/aps.71.20211000](https://doi.org/10.7498/aps.71.20211000)

^{*} Project supported by the National Natural Science Foundation of China (Grant Nos. 11975132, 61772295), the Natural Science Foundation of Shandong Province, China (Grant No. ZR2019YQ01), and the Project of Higher Educational Science and Technology Program of Shandong Province, China (Grant No. J18KZ012).

[†] Corresponding author. E-mail: hongyang_ma@aliyun.com