DOI: 10.3969/j.issn.1007-5461. 2022.03.015

基于 Cirq 的 Grover 搜索算法的电路实现

吴 希，李志强*

(扬州大学信息工程学院， 江苏 扬州 225100)

摘 要: Grover 量子算法能对传统的搜索算法起到平方级加速的效果, 因此自提出以来一直受到人们的广泛关注。首先将基于 Python 的 Cirq 框架与 Grover 搜索算法进行结合, 并对其进行模拟实现, 可以直观地看到算法的电路实现细节, 同时, 实验验证了该算法的特点与存在的不足。进而针对搜索成功率存在的不足, 从理论上介绍一种基于相位角旋转的精准 Grover 改进算法, 并通过 Cirq 框架对其进行模拟实现, 验证了该算法成功率始终为 1 的有效性。Cirq 框架的引入为量子算法的研究以及量子电路的优化提供了强大的工具支持。

关 键 词: 量子信息; 量子电路; Grover 算法; Cirq 框架

中 图 分 类 号 : O431.2 **文 献 标 识 码 :** A **文章编号:** 1007-5461(2022)03-00431-08

Circuit realization of Grover algorithm based on Cirq

*WU Xi, LI Zhiqiang**

(College of Information Engineering, Yangzhou University, Yangzhou 225100, China)

Abstract: Grover quantum search algorithm can speed up the traditional search algorithm by square order, so it has been widely concerned since it was put forward. In this paper, the Cirq framework based on Python is introduced into the traditional Grover algorithm and the circuit details of the algorithm is realized by simulation at first, and at the same time, the characteristics and defects of the algorithm are verified intuitively through experimental results. Then, in view of the shortcomings of the search success rate of the traditional Grover algorithm, a precise improved Grover algorithm based on phase angle rotation is theoretically introduced, and it is simulated by Cirq framework, which verifies the effectiveness of the algorithm with a success rate of 1. It is believed that the introduction of the Cirq framework can provide a powerful tool support for the research of quantum algorithms and the optimization of quantum circuits.

Key words: quantum information; quantum circuit; Grover algorithm; Cirq framework

基金项目: Supported by National Natural Science Foundation of China (国家自然科学基金, 61070240), Natural Science Foundation of Universities in Jiangsu Province (江苏省高校基金, 10KJB520021)

作者简介: 吴 希 (1997 -), 女, 江苏无锡人, 研究生, 主要从事量子电路综合方面的研究。 **E-mail:** wxyzdd@163.com

导师简介: 李志强 (1974 -), 江苏扬州人, 博士, 教授, 硕士生导师, 主要从事量子信息、量子可逆电路综合方面的研究。

E-mail: yzqqLzq@163.com

收稿日期: 2020–08–26; **修改日期:** 2020–11–02

*通信作者。

0 引 言

Grover 量子搜索算法[1] 是经典的量子算法之一, 体现了量子计算的优势和广阔的应用前景。它利用量子的并行性, 将传统计算机的搜索算法从 $O(N)$ 的复杂度降为 $O(\sqrt{N})$, 很大程度上提高了搜索效率。然而, Grover 算法也存在局限性, 当搜索的目标数超过搜索总数的 1/4 时, 搜索的成功率会迅速下降, 当目标数超过 1/2 时, 算法就会失效[2]。为了让 Grover 算法有更多的应用前景, 研究人员在算法的改进与优化方面做了大量研究, 主要的改进方式有以下三种[3−5]: (1) 改变一些步骤或变换算子; (2) 改变运算算子的相位旋转角度; (3) 寻找更一般的初始态幅值来解决寻找中值、最小值等特定问题。其中, 从相位角度的改进效果最好。

当前还没有研制出成熟的量子计算机, 所以对量子算法进行模拟实验具有重要意义。Cirq[6] 是谷歌发布的一款基于 Python 的开源框架, 用于模拟 NISQ (Noisy Intermediate-Scale Quantum) 计算机上的电路, 使研究人员能在特定的处理器上编写量子算法, 解决实际的计算问题, 对复杂的细节进行研究。本文通过 Cirq 框架实现 Grover 算法的量子电路模拟, 根据实验模拟结果验证了 Grover 算法的正确性, 同时也证明了该算法存在的不足。并且介绍了基于改变相位旋转角度的精准算法[7], 对算法进行量子电路模拟实现, 验证了精准算法的有效性。由此可见, Cirq 为量子电路的模拟提供了技术支持, 使人们对理论的研究更具现实意义。

1 Grover 基本算法实现

1.1 算法步骤和几何描述

假设要在 $N = 2^n$ 个无结构的搜索元素中寻找 M 个目标解, $1 \le M \le N$, 找到 M 中的任意一个解都算成功。于是元素指标可以存储在 n 个量子比特中[8]。

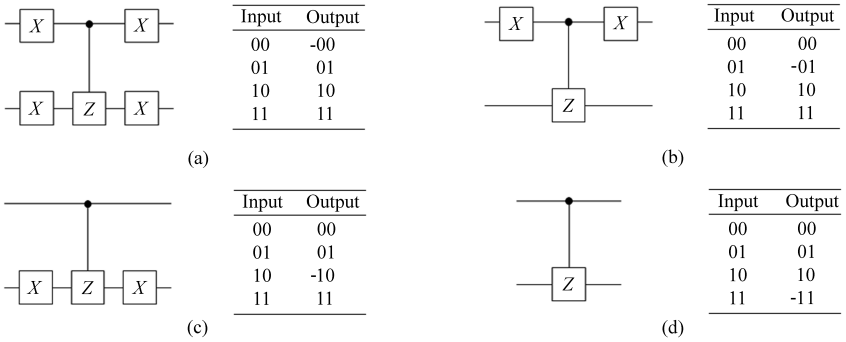
首先介绍 Grover 算法中的一个重要操作: Oracle 操作。定义一个 $f(x)$, x 是搜索的元素, 当满足 $f(x) = 1$, x 是搜索问题的解, 反之则不是解。在搜索的过程中将注意力放在元素的索引上, 搜索范围就是 $0 \sim 2^n - 1$。Oracle 是一个能够识别出搜索问题目标解的酉算子, 被定义为 $|x, q\rangle \to |x, q \oplus f(x)\rangle$, 其中: $|x\rangle$ 存储搜索元素的索引, $|q\rangle$ 是一个单量子比特, $\oplus$ 表示模 2 加。把 $|q\rangle$ 初始化为 $|-\rangle$, 当 $f(x) = 1$ 时, 状态的概率幅进行翻转变成负的; 反之则不变。由于在 Oracle 作用前后 $|q\rangle$ 并没有发生变化, 所以该操作可以化简为 $O : |x\rangle \to (-1)^{f(x)}|x\rangle$。

Grover 算法需要 n 个量子比特存储搜索元素, 以及一些额外的 Oracle 辅助位。电路输入的初始值是应用 Hadamard 变换构成的 N 个均匀叠加态, 将这个状态表示为 $|\varphi\rangle = \dfrac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$, 也就是将搜索元素的索引映射为量子叠加态的基态, 0 对应基态 $|00\cdots0\rangle$, $2^n - 1$ 对应基态 $|11\cdots1\rangle$。搜索算法是由许多个被称为 G 的 Grover 迭代构成的。一次 Grover 迭代包含了四个步骤, 分别表示为: (1) O: Oracle 操作; (2) $W1$: 对 n 个量子位应用 Hadamard 门变换; (3) R: 条件相移操作。应用条件相移算子 $2|0\rangle\langle 0| - I$, 使得与 $|0\rangle$ 正交的基态的概率幅进行翻转, 也就是当 $|x\rangle \ne |0\rangle$ 时, $|x\rangle = -|x\rangle$, $|x\rangle \to -(-)^{\delta x_0}|x\rangle$; (4) $W2$: 再次应用 Hadamard 门。步骤 (2)~(4) 的组合效果为 $H^{\otimes n}(2|0\rangle\langle 0| - I)\mathrm{H}^{\otimes n} = 2|\varphi\rangle\langle\varphi| - I$, 其中 $|\varphi\rangle$ 是初始的均匀叠加态。那么, 一次 Grover 迭代就可以写成 $G = (2|\varphi\rangle\langle\varphi| - I)O$。

为了能够更清晰地了解算法的过程, 用几何方式来描述。假设 X_0 是非解集合, X_1 是解的集合。那么非解与解集合的状态可以表示为: $|\alpha\rangle = \frac{1}{\sqrt{N-M}}\sum_{x\in X_0}|x\rangle$, $|\beta\rangle = \frac{1}{\sqrt{M}}\sum_{x\in X_1}|x\rangle$。初始状态就可以写成: $|\varphi\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle$。假设 $\cos\theta/2 = \sqrt{(N-M)/N}$, $\sin\theta/2 = \sqrt{M/N}$。那么 $|\varphi\rangle = \cos\frac{\theta}{2}|\alpha\rangle + \sin\frac{\theta}{2}|\beta\rangle$。进行一次 Grover 迭代后, $G|\varphi\rangle = \cos\frac{3\theta}{2}|\alpha\rangle + \sin\frac{3\theta}{2}|\beta\rangle$, 初始状态 $|\varphi\rangle$ 的角度从 $\theta/2$ 变为 $3\theta/2$, 使得向量向 $|\beta\rangle$ 方向旋转 θ 角度, 也就是增大了解向量的概率幅。若经过 k 次迭代, 则有 $G^k|\varphi\rangle = \cos\frac{(2k+1)\theta}{2}|\alpha\rangle + \sin\frac{(2k+1)\theta}{2}|\beta\rangle$。若要使 $|\varphi\rangle$ 不断靠近 $|\beta\rangle$, 则需要迭代的次数为 $R = \left\lceil\frac{\arccos(\sqrt{M/N})}{\theta}\right\rceil$, 由于这只能针对 $\theta/2 \le \pi/4$ 的情况, 当 $M \ll N$ 时, $\theta \approx \sin\theta \approx 2\sqrt{M/N}$, 又因为 $\arccos(\sqrt{M/N}) \le \pi/2$, 所以 $R \le \left\lceil\frac{\pi}{4}\sqrt{\frac{N}{M}}\right\rceil = O(\sqrt{N/M})$。

1.2　电路实现

用一个示例来说明 Grover 算法在 2-qubit 搜索空间中的具体电路。其中 Oracle 的相位翻转操作用受控 Z 门来实现。如图 1 所示, 满足 $f(x_0) = 1$ 的四个电路由上至下分别对应 $x_0 = 0, 1, 2, 3$ 的情况。每个电路的右边给出了对应的真值表, 由表可以看出, 电路能够实现 $O: |x\rangle \to (-1)^{f(x)}|x\rangle$ 的操作。



Input	Output
00	-00
01	01
10	10
11	11

Input	Output
00	00
01	-01
10	10
11	11

Input	Output
00	00
01	01
10	-10
11	11

Input	Output
00	00
01	01
10	10
11	-11

图 1　Oracle 电路以及目标位 (a) 00, (b) 01, (c) 10, (d) 11 的真值表

Fig. 1　Oracle circuit and truth table of target bits (a) 00, (b) 01, (c) 10, (d) 11

假设目标解是 |11⟩, 那么 2-qubit 的 Grover 电路如图 2 所示, 实线框中是 |11⟩ 对应的 Oracle 操作, 虚线框中是扩散操作, 只需进行一次 Grover 迭代。

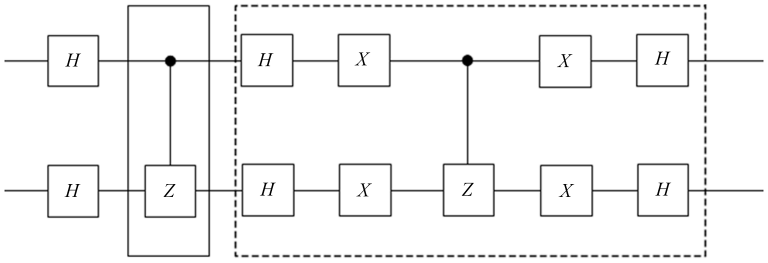


图 2　2-qubit 的 Grover 算法电路

Fig. 2　Circuit of 2-qubit Grover algorithm

将电路扩展至 n-qubit, 通过基于 python3.7 的 Cirq 框架, 在联想 V3000 上实现 n-qubit 的 Grover 量子电路, 实现的部分代码如下:

```
import cirq
def make_Oracle(q, qn_1, x_bits):                    # Oracle操作
yield(cirq.X(q) for (q, bit) in zip(q, x_bits) if not bit)
yield cirq.Z(q[len(q)-1]).controlled_by(*qn_1)
yield(cirq.X(q) for (q, bit) in zip(q, x_bits) if not bit)
def make_Grover(q, qn_1):                            # Grover算子
yield cirq.H.on_each(*q)
yield cirq.X.on_each(*q)
yield cirq.Z(q[len(q)-1]).controlled_by(*qn_1)
yield cirq.X.on_each(*q)
yield cirq.H.on_each(*q)
def main():
q = [cirq.GridQubit(i, 0) for i in range(n)]         # n位量子比特
qn_1 = [cirq.GridQubit(i, 0) for i in range(n-1)]    # 前n-1位量子比特
x = random.sample(range(0, 2 ** n), m)               # 随机生成m个目标数
count = int((math.pi / 4) * (2**n / m) ** 0.5)       # 迭代次数
circuit = cirq.Circuit(cirq.H.on_each(*q))           # 准备叠加态的电路
for _ in range(0, count):                            # Grover 迭代电路
for i in x_bitsM:
circuit.append(make_Oracle(q, qn_1, i))
circuit.append(make_Grover(q, qn_1))
circuit.append(cirq.measure(*q, key='result'))       # 测量
simulator = cirq.Simulator()                         # 模拟器
result = simulator.run(circuit, repetitions=100000)  # 测量结果
```

1.3 实验结果及分析

对电路进行测试, 输入 3 位量子位数和 1 个目标元素, 对电路进行 100000 次测量, 对测量结果进行采样。运行结果如图 3 所示, 电路图中第一个虚线框是 Oracle 操作, 第二个框是扩散操作, 进行了两次 Grover 迭代。随机产生的一个目标元素是 5, 采样结果根据次数由高到低显示, 测量得到 5 的次数最多, 为 94519 次; 测量到其他数的次数在 800 左右。根据结果可知算法正确的概率大约为 94.5%。

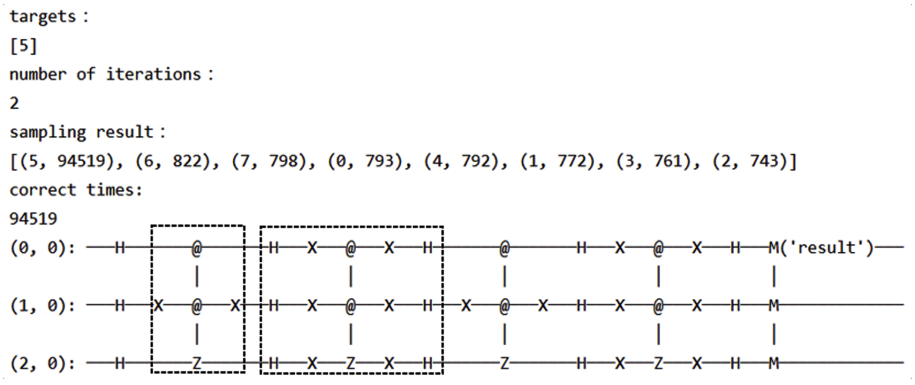


图 3 3-qubit 的实验结果

Fig. 3 Experimental results of 3-qubit

对目标数占搜索总数的 1/2 这一情况进行实验, 测试结果如表 1 所示。时刻表示所有在同一抽象时

间段内执行的操作的集合。例如图 3 的三个用于初始化的 H 门表示一个时刻数。从表 1 可以看出, 当目标元素为总搜索数的 1/2 时, 不论数量有多大, 搜索的成功率都为 50%。当运行到 13 个量子位时, 输出的电路会出现错误, 原因是电路的复杂度达到上限, 但测量得到的结果是准确的。若将 13 个量子位的目标数改为 2048, 则可以输出正确的电路, 说明 Oracle 查询的复杂度对电路复杂度的影响很大。目标数成倍增加伴随着 Oracle 查询越复杂, 所需时刻数与门数也趋于成倍增加, 运行时间也就越长。

表 1　目标数为 1/2 时的实验结果

Table 1 Experimental results when the number of targets accounted for 1/2 of the total

Qubits	Targets	Probability/%	Time/s	Moments	Gates
1	1	50	0.315	10	10
2	2	50	0.399	12	18
3	4	50	0.495	18	35
4	8	50	0.566	28	54
5	16	50	0.665	54	131
6	32	50	0.762	96	244
7	64	50	0.890	190	519
8	128	50	1.118	375	1180
9	256	50	1.476	751	2611
10	512	50	2.122	1508	5664
11	1024	50	3.439	3030	12311
12	2048	50	6.383	6078	26718
13	4096	50	12.518	12199	57409

以固定的 1024 个搜索总数为例验证算法的性能, 实验结果记录在表 2 中。随机生成无序的搜索目标并且不断增大目标元素的数量, 分析目标元素与搜索总数的比值 λ 对 Grover 迭代次数 R 以及搜索成功概率 P 的影响。

表 2　Grover 算法成功率实验结果

Table 2 Experimental results of the success probability rate of Grover algorithm

Parameter	Value								
λ	0.05	0.1	0.2	0.25	0.3	0.35	0.4	0.5	0.6
R	3	2	1	1	1	1	1	1	1
P	99.99%	99.86%	96.66%	100%	97.28%	89.62%	78.30%	50%	21.65%

从表 2 可以看出, 当 $0.3 < \lambda < 0.5$ 时, 算法的成功率迅速下降, 迭代的次数始终是 1; 当 $\lambda > 0.5$ 时, 算法的成功率甚至低于失败率。传统的搜索方法是目标元素越多, 成功率则越高, 而量子 Grover 算法却是在搜索数较少时有较高的成功率。除此以外, 传统的搜索方法总能得到百分百的正确率, 基本 Grover 算法由于其测量特性, 可能需要进行多次搜索才能达到预期效果。

2　精准 Grover 算法的实现

上文介绍了 Grover 量子搜索算法的基本步骤、几何描述以及公式推导。也通过在 Cirq 框架上对电路的模拟分析, 直观地了解到 Grover 算法存在的局限性。已有许多学者针对这些不足进行了深入研究。Long 等[9] 最初发现 Grover 算法中的两个相移算子可以被相位匹配关系的相位角代替, 基本的 Grover 算

法中存在两次相位翻转, 翻转的角度都是固定的 π。目前, 基于两个旋转算子中旋转相位匹配条件的改进, 提出了许多可以提高搜索成功概率的改进算法[10−12]。通过 Cirq 框架可以很容易地模拟这些改进算法。

Long 等[9] 首次提出了精准的量子搜索算法, 使搜索的成功率达到 100%。但该算法存在两个缺陷, 一是只适用于搜索一个目标元素, 二是无法解决目标数占总搜索数 1/2 的情况。所以 Xia 等[8] 在此基础上进行改进, 先将相位取反, 再将搜索总数这一变量引入到相位旋转角度中, 使得算法具有实时性, 成功概率一直为 100%。这一算法非常适合需要高精度的应用场景中的搜索。该算法将相位旋转角改为

$$\phi=\theta=2\arcsin\left[\sin\left(\frac{\pi}{4J+2}\right)/\sqrt{\frac{M}{N}}\right],$$

式中: $J=\left\lceil\frac{\pi-\theta}{2\theta}\right\rceil$ 表示迭代次数, $\theta=2\arcsin\sqrt{M/N}$。

通过基于 google 的 Cirq 框架模拟实现精准算法, 只需要在基本的 Grover 算法中添加旋转角度这一变量, 部分代码如下:

```
♯ 根据公式算出相位旋转角度angle和迭代次数count
a = (m / 2 ** n) ** 0.5
bl = 2 * math.asin(a)
J = (math.pi-bl)/(2*bl)
count = math.ceil(J)
t = math.sin(math.pi/(4*count+2))/a
angle = 2*math.asin(t)/math.pi
♯ 为旋转Z门添加旋转角度
yield Cirq.Z(q[len(q) - 1]).controlled_by(*qn_1)**angle
```

对 3-qubit 精准算法进行测试, 随机产生三个目标元素, 单次运行结果如图 4 所示, 测量正确的概率为 100%, 并给出了旋转门所需旋转的角度为 0.608。电路图中的第一个虚线框对应三个目标元素的 Oracle 操作。

经过多次实验, 模拟实现的结果与理论计算结果一致,算法的成功率始终是 100%, 验证了算法的有效性。用 Cirq 框架可以轻易地模拟出基于相位旋转匹配改进算法的电路。相比于文献 [13] 中使用量子程序设计语言 QCL 模拟实现, 使用 Cirq 框架实现量子算法, 不仅能够看到概率性以及复数矩阵的输出, 还能看到整个算法实现的完整电路, 这对电路的优化起到了很大的帮助。

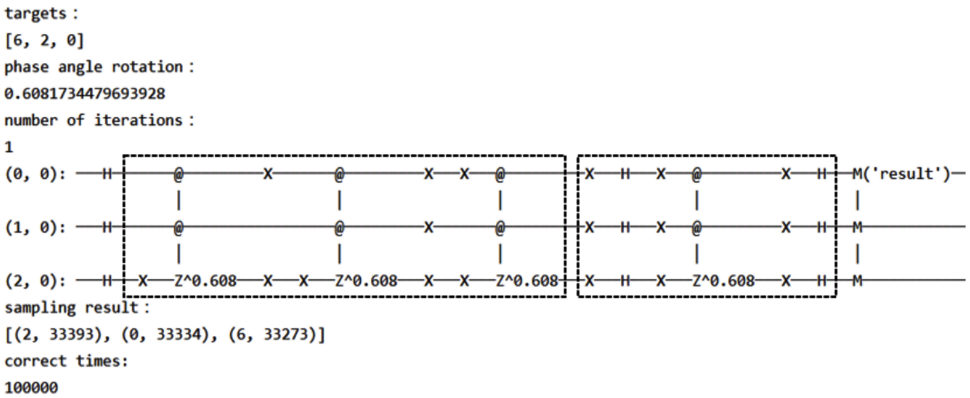


图 4 3-qubit 精准 Grover 算法的结果

Fig. 4 Results of 3-qubit accurate Grover algorithm

3　总结与展望

借助 Cirq 框架, 对基本 Grover 算法及其改进的精准算法进行了电路模拟, 对测试结果进行分析, 验证各算法的准确性以及各自存在的特点, 方便应用于不同的情景。搜索算法的相位旋转通过改变旋转 Z 门的角度来实现, 但目前只是对其进行模拟, 还未在特定的处理器上实现, 而 Cirq 也提供了在实际硬件上运行的功能, 以便未来开展进一步的优化工作。本研究中基于相位旋转匹配的改进算法都是在目标分量已知的情况下, 算法的迭代次数都依赖于目标分量的数量。当目标分量的数量未知时, 往往需要很高的 Oracle 查询复杂度, 而且如何在最优迭代次数时停止也是需要研究的问题。已有一些改进的算法来解决这一问题[14,15], 但在效率与概率方面都存在进步空间, 希望未来能够在目标分量未知的情况下, 通过 Cirq 框架对算法进行优化、验证与分析。此外, 通过对 Cirq 框架源码的深入理解, 还可以更多地扩展其功能, 为电路的模拟提供更强大的技术支持。

参考文献:

[1] Grover L K. A fast quantum mechanical algorithm for database search [C]. *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, 1996.

[2] Li P C, Song K P. Adaptive phase matching in Grover's algorithm [J]. *Journal of Quantum Information Science*, 2011, 1(2): 43-49.

[3] Grover L K. Quantum computers can search rapidly by using almost any transformation [J]. *Physical Review Letters*, 1998, 80(19): 4329-4332.

[4] Biron D, Biham O, Biham E, *et al*. *Generalized Grover Search Algorithm for Arbitrary Initial Amplitude Distribution* [M].// Quantum Computing and Quantum Communications. Berlin, Heidelberg: Springer, 1999: 140-147.

[5] Biham E, Kenigsberg D. Grover's quantum search algorithm for an arbitrary initial mixed state [J]. *Physical Review A*, 2002, 66(6): 062301.

[6] Cirq Developers. Cirq [OL]. https://github.com/quantumlib/Cirq/graphs/contributors, https://github.com/quantumlib/Cirq.

[7] Long G L, Li Y S, Xiao L, *et al*. Phase matching in quantum searching and the improved Grover algorithm [J]. *Nuclear Physics Review*, 2004, 21(2): 114-116.
龙桂鲁, 李岩松, 肖 丽, 等. Grover 量子搜索算法及改进[J]. 原子核物理评论, 2004, 21(2): 114-116.

[8] Xia K W, Su C, Shen J Y, *et al*. Improved Grover's quantum searching algorithm [J]. *Journal of Xi'an Jiaotong University*, 2007, 41(10): 1127-1131.
夏克文, 苏 昶, 沈钧毅, 等. 一种改进的 Grover 量子搜索算法[J]. 西安交通大学学报, 2007, 41(10): 1127-1131.

[9] Long G L, Li Y S, Zhang W L, *et al*. Phase matching in quantum searching [J]. *Physics Letters A*, 1999, 262(1): 27-34.

[10] Younes A. Towards more reliable fixed phase quantum search algorithm [J]. *Applied Mathematics & Information Sciences*, 2013, 7(1): 93-98.

[11] Li P C, Song K P. Adaptive phase matching in Grover's algorithm [J]. *Journal of Quantum Information Science*, 2011, 1(2): 43-49.

[12] Toyama F M, van Dijk W, Nogami Y, *et al*. Multiphase matching in the Grover algorithm [J]. *Physical Review A*, 2008, 77(4): 042324.

[13] Zhang H T, Dai Y T, Tu L Y, *et al*. The simulation of Grover quantum search algorithm [J]. *Journal of Shaanxi Normal University (Natural Science Edition)*, 2016, 44(3): 7-10.
张洪涛, 代永涛, 涂玲英, 等. Grover 量子搜索算法的模拟实现[J]. 陕西师范大学学报 (自然科学版), 2016, 44(3): 7-10.

[14] Zhu W N, Chen H W. Grover auto-control searching algorithm [J]. *Acta Electronica Sinica*, 2016, 44(12): 2975-2980.
朱皖宁, 陈汉武. 迭代次数自适应的 Grover 算法[J]. 电子学报, 2016, 44(12): 2975-2980.

[15] Xie X M, Duan L Z, Qiu T R, *et al*. Improved quantum search algorithm and its application on computation of core [J]. *Computer Engineering and Applications*, 2020, 56(14): 57-61.
谢旭明, 段隆振, 邱桃荣, 等. 改进量子搜索算法及其在核属性求解上的应用[J]. 计算机工程与应用, 2020, 56(14): 57-61.