量子增强模式搜索优化 n

Colton Mikes、Ismael R. de Farias Jr.、David Huckleberry Gutman、Victoria E. Howle

德克萨斯理工大学，美国德克萨斯州拉伯克，邮编 91125

arXiv:2305.01703v1 [quant-ph] 2 May 2023

本文介绍了一种用于广义模式搜索（GPS）算法的量子-经典混合算法。我们介绍了一种采用振幅放大的量子搜索步骤算法，可将搜索步骤所需的预言机调用次数从经典调用的 $\mathcal{O}(N)$ 次减少到量子调用的 $\mathcal{O}\left(\sqrt{N}\right)$ 次。本研究解决了在 GPS 中使用量子搜索步骤时的三个基本问题。首先，我们讨论了标记改进的网格点这一需求，这是振幅放大算法的必要条件。其次，我们介绍了振幅放大算法 QSearch 的一个修改版本，该版本保证在有限次迭代内终止。第三，我们将量子算法限制在搜索步骤中，以避免干扰 GPS 算法的收敛性。

1 引言

传统优化算法依赖导数来获得结果。然而，在许多情况下，导数信息可能无法获取。即使某些阶数的导数存在，计算其中任何一个导数也可能不切实际，或成本高得令人难以承受。例如，当目标函数通过黑箱模拟计算时，获取梯度信息可能并不现实。同样，对于计算过程含噪或计算成本高昂的目标函数，有限差分等近似方法的成本也会高得难以承受。即使对于通过复合和算术运算由初等函数递归构造的函数，计算梯度也可能在时间和内存方面产生高昂的开销。在某些情况下，还可以使用自动微分来获取梯度信息，但由于必须能够获得该函数的源代码，这种方法并不总是切实可行。在这些情况下，我们会转而采用无导数优化（DFO）算法。DFO 算法旨在解决给定的优化问题，而不使用任何阶数的微分信息。相反，它们依赖调用给定的预言机来评估目标函数。这些调用通常成本高昂，并会成为算法的主要开销。

在这项工作中，我们关注一类称为广义模式搜索（GPS）的 DFO 算法。这些算法通过系统地选择要评估并纳入下一迭代点考量的点，生成一系列目标函数值不增的迭代点。该过程分为两个步骤：搜索步骤和轮询步骤。两者唯一的区别在于所评估的点。搜索步骤旨在提供很大的灵活性，以便选择要考量的点；而轮询步骤只考量保证算法收敛所必需的点。一次典型的迭代从搜索步骤开始。只有当搜索步骤未能产生目标函数值降低的迭代点时，才会调用轮询步骤。由于轮询步骤并非总是必需的，实现 GPS 算法的成本是

<escape></escape>通常主要由搜索步骤决定。然而，Audet 和 Dennis 提出的各种收敛结果均未对搜索步骤作出任何假设 <escape></escape> [2, <escape></escape>定理 3.7、3.9、3.14 以及推论 3.11、3.12、3.<escape></escape>16]<escape></escape>。<escape></escape>

本文的主要贡献是一种量子算法，该算法能够使用 $\mathcal{O}\left(\sqrt{N}\right)$ 次量子预言机调用来执行搜索步骤，而使用经典预言机则需要 $\mathcal{O}(N)$ 次调用。与 [6],[7], 和 [3], 一样，我们遵循标准做法，假设量子预言机调用与经典预言机调用之间的计算成本差异可以忽略不计。我们的算法可分为两个组成部分：修改后的振幅放大算法 QSearch [4], ，以及一种量子算法，该算法利用量子预言机为修改后的 QSearch 算法准备必要的输入。

Arunachalam 曾在 [1]中提到过使用振幅放大来加速 GPS算法 的想法。然而，在该工作中，他们未能解决以下问题：如何标记能够降低目标函数值的点，如何保证振幅放大在有限步数内终止，以及如何在量子步骤引入误差的情况下确保收敛性。本文解决了这三个问题。我们的算法将量子预言机作为子程序的一部分，用于标记能够降低目标函数值的点；该算法保证在有限步数内终止，并通过在经典计算机上执行轮询步骤来保持原算法的收敛性质。

本文其余部分的组织如下。在第 2 节中，我们回顾 GPS算法、量子计算，以及如何使用量子计算机表示我们的优化问题。在第3 节中，我们介绍振幅放大算法 QSearch 的一个修改版本 [4, 定理 3]。在第 3.1 节中，我们讨论振幅放大和原始的 QSearch算法。在第 3.2 节中，我们介绍为确保 QSearch 使用有限次迭代终止所需的修改。在第4 节中，我们讨论我们的主要成果——量子搜索步骤算法。在第 4.1 节中，我们构建使用修改后的 QSearch执行 GPS 算法搜索步骤所需的组件。在第 4.2 节中，我们讨论使用量子搜索步骤算法相较于经典方法的优势。最后，我们在第 5 节给出一些总结。

2 预备知识

本节回顾理解本文主要发展内容所需的 GPS 算法和量子计算中的全部技术与符号。在第 2.1 节中，我们按照 Audet 和 Dennis 在 [2] 中提出的表述回顾 GPS 算法。在 Torczon 于 [9], 中引入的原始表述里，搜索步骤和轮询步骤并未被视为彼此独立的实体。通过将一次迭代分为搜索步骤和轮询步骤，Audet 和 Dennis 得以证明，搜索步骤可以提供更大的灵活性，而收敛性则由轮询步骤保证。在第 2.2 节中，我们回顾量子计算的一些基本组成部分，并着眼于在量子计算机上表示我们的优化问题所需的要素。

2.1 广义模式搜索算法

本节回顾 GPS 算法的构造。这些算法最初由 Torczon 在 [9] 提出，随后由 Audet 和 Dennis 在 [2] 加以拓展。这里我们采用 Audet 和 Dennis 的表述方式，将迭代分为不同的搜索

步骤和轮询步骤，以便加速搜索步骤，同时保留轮询步骤所提供的收敛性保证。

GPS算法求解优化问题

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

对于某个 f： $\mathbb{R}^n \to \mathbb{R}$，通过生成一系列迭代点 $\{x_k\}_{k=0}^{\infty}$，使得 $\{f(x_k)\}_{k=0}^{\infty}$是一个非增序列。每次迭代包含两个阶段：搜索步骤 和轮询步骤。

在搜索步骤中，我们在称为网格的集合中的有限个点处计算 f。在第 k 次迭代中，网格是由以下方式定义的集合:

$$M_k = \{x_k + \Delta_k Dz : z \in \mathbb{N}^p\}, \tag{2}$$

其中， x_k 是我们的当前迭代点， Δ_k 是一个称为网格尺寸参数的正实数，D 是一个实值 $n\times p$ 矩阵，其列构成一个正生成集，即 $\mathbb{R}^n$ 中的每个向量都可以表示为 D 的列的非负线性组合。此外， D满足以下限制：其列是某个非奇异生成矩阵 $G \in \mathbb{R}^{n\times n}$ 与整数向量 $z_j \in \mathbb{Z}^n$ 的乘积，其中 $j = 1$、2、.....、p。用符号表示，这一限制意味着 D 具有 $D = [Gz_1 \dots Gz_p]$ 的形式。在第 k 次迭代中，如果某个点$y \in M_k$ 满足 $f(x_k) > f(y)$，我们就称其为改进的网格点 。如果在搜索步骤中找到这样的点，我们就结束该次迭代。否则，我们开始轮询步骤。

在第 k 次迭代中，从 D 的列中选取一个子集，记为 D_k，该子集是一个正生成集。利用当前迭代点和 D_k 中的元素构造轮询集 P_k：

$$P_k = \{x_k + \Delta_k d : d \in D_k\}. \tag{3}$$

如果搜索步骤失败，我们将执行轮询步骤，在此步骤中，我们会对 P_k 中的每个元素计算目标函数，以检查是否存在改进的网格点。如果没有找到任何改进的网格点，我们就称当前迭代点 x_k 为 网格局部优化器。

每当搜索步骤或轮询步骤将某个点 y 确定为改进的网格点时，我们立即结束本次迭代，设置 $x_{k+1} = y$，并增大或保持网格尺寸参数不变。或者，如果 x_k 被确定为网格局部优化器，则我们设置 $x_{k+1} = x_k$ 并缩小网格尺寸参数。有关此版本 GPS 算法的详细介绍，请参阅 [2]。

2.2 量子计算

在本节中，我们回顾量子计算的相关基础知识，并讨论如何在量子计算机上表示我们的优化问题 (1)。顾名思义，量子计算机是一种通过操纵量子力学系统来实现计算结果的计算设备。在任意给定时刻，量子计算机的状态都可以用复值内积空间中的一个单位向量来描述，该向量称为状态向量。我们使用量子比特来存储这个向量，量子比特是量子计算的基本单位。任意量子比特的状态可以写成 $|\phi\mathrm{i} = \alpha|0\mathrm{i} + \beta|1\mathrm{i}$，其中 $\alpha, \beta \in \mathbb{C}$，{$|0\mathrm{i}$ $|1\mathrm{i}$} 是 $\mathbb{C}^2$ 的标准基，称为单个量子比特的计算基，此处记为 C (1)。对于更熟悉非物理数学记法的读者，我们注意到 $|0\mathrm{i} = (1,0)^T$ and $|1\mathrm{i} = (0,1)^T$。由于 $|\phi\mathrm{i}$ 必须是单位向量，我们要求 $\mathrm{k}\alpha\mathrm{k}^2 + \mathrm{k}\beta\mathrm{k}^2 = 1$。我们称 α and β 为 $|\phi\mathrm{i}$ 的振幅。

对于包含多个不同量子比特的量子计算机，可以通过对这些量子比特取克罗内克积来获得状态向量。量子计算机的计算基

对于具有 2 个量子比特的量子计算机，$\mathcal{C}(2)$ 是$\mathcal{C}(1)$ 中各元素彼此作克罗内克积得到的。一般而言，具有 N 个量子比特的量子计算机的计算基 $\mathcal{C}(N)$，是 $\mathcal{C}(N-1)$ 和 $\mathcal{C}(1)$ 中所有元素作克罗内克积得到的。对于任意两个量子比特 $|\phi\rangle$ 和 $|\theta\rangle$，我们将复合态写作 $|\phi\rangle|\theta\rangle = |\phi\theta\rangle$。对于一台具有 N 个量子比特、且所有量子比特都处于相同任意态 $|\phi\rangle$的量子计算机，我们将状态向量记作$|\phi\rangle^{\otimes N}$。此外，当一台具有 N 个量子比特的量子计算机中，所有量子比特都处于 $\mathcal{C}(1)$ 中的同一状态时，只要上下文中能明确量子比特的数量，我们通常会省略上标，直接将状态写作 $|0\rangle$ 或 $|1\rangle$。我们将一组给定的量子比特定义为一个量子寄存器。如果一组量子比特的状态向量无法写成单量子比特态的克罗内克积，则称这组量子比特是纠缠的。量子逻辑门用酉矩阵建模，并可作用于状态向量。

量子力学的一个重要方面是，我们无法直接查看状态向量。任何此类尝试都会得到其中一个基向量。这一操作称为测量。测量结果由状态向量的振幅决定。如果多个振幅非零，我们就说状态向量处于相应状态的叠加态中。例如，测量量子比特 $|\phi\mathbf{i} = \alpha|0\mathbf{i} + \beta|1\mathbf{i}$ 时，得到$|0\mathbf{i}$ 的概率为 $\mathbf{k}\alpha\mathbf{k}_2^2$, 同样，得到 $|1\mathbf{i}$ 的概率为 $\mathbf{k}\beta\mathbf{k}_2^2$。如果 α 和 β 都非零，我们就说 $|\phi\mathbf{i}$ 处于 $|0\mathbf{i}$ 和 $|1\mathbf{i}$ 的叠加态中。量子算法定义为作用于状态向量的酉矩阵，之后也可以进行测量。如果不进行测量，则称该量子算法为无测量量子算法。对于包含多个量子比特的量子计算机，每个量子比特都要单独测量。在某些情况下，只测量选定的几个量子比特更为有利。例如，交换测试等算法 [5] 依赖于测量单个量子比特，并利用测量结果获取有关最终状态向量的信息。

现在我们讨论如何在量子计算机上表示优化问题。我们使用定点补码二进制表示法来表示任意实数，精度最高可达 d 位。补码表示是表示有符号二进制数最常见的方法之一。在这种表示法中，最高有效位表示给定二进制数的符号。如果该位为 0，则该数为正数；如果该位为 1，则该数为负数。对给定数取负的方法是翻转其表示中的每一位，然后加 1。

对于 $x \in \mathbb{R}^n$，我们将 x 表示为 dn 位二进制字符串，其中每个元素的精度最高可达 d 位，而字符串中 d 位的 kth 集合是 x 的 kth 个元素的 d 位定点表示。因此，x 被编码为 dn 个量子比特，其中第 k 个量子比特的状态由 x 的第 k 位决定。例如，如果 $x = 0110$，那么 $|x\rangle = |0110\rangle$。为计算目标函数 f，我们假设可以访问量子预言机 F，使得 $F|x\rangle|0\rangle^{\otimes n\cdot d} \to |x\rangle|f(x)\rangle$。我们可以调用预言机 F 一次，计算一组点 $\{x_j\}_{j=0}^{m-1}$ 上的目标函数。具体做法是，先将量子寄存器初始化为这些点的叠加态，然后应用 F，

$$F\sum_{j=0}^{m-1}|x_j\rangle|0\rangle \to \sum_{j=0}^{m-1}|x_j\rangle|f(x_j)\rangle.$$

然而，由于函数求值以叠加态的形式存储，我们无法直接查看它们。在接下来的章节中，我们将介绍使用 F识别约化目标函数值所需的工具。

3 修改后的QSearch

本节介绍一种经过修改的振幅放大 方法$QSearch$ 算法，该算法由 [4, 定理 3]提出。如果量子计算机处于目标态和非目标态的叠加态中，则可使用 QSearch算法在测量时获得一个目标态。我们之所以需要进行修改，是因为如果叠加态中不存在目标态，QSearch 就无法保证终止。修改后的QSearch算法会在叠加态中包含目标态的概率足够低时终止。在第 3.1 节中，我们将回顾振幅放大和原始 QSearch算法的核心概念。在第3.2节中，我们将介绍一种方法，用于确定叠加态中包含目标结果的概率何时已低于任意给定的容差值。

3.1 振幅放大

本节将说明如何使用振幅放大算法 QSearch [4, 定理3] 执行 GPS 算法的搜索步骤。首先简要回顾该算法本身。

假设我们有一个无测量量子算法 $\mathcal{A}$。此外，假设将 $\mathcal{A}$ 作用于初始态 $|0\rangle^{\otimes N}$ 后，得到的是计算基态在 $\mathcal{C}(N)$ 中的叠加，其中一些称为目标态（$\mathcal{D}$），另一些称为非目标态（$\mathcal{U}$）。更形式化地说，我们假设

$$\mathcal{A}|0\rangle = \sum_{|j\rangle\in\mathcal{D}} \alpha_j |j\rangle + \sum_{|k\rangle\in\mathcal{U}} \alpha_k |k\rangle \tag{4}$$

其中 $\mathcal{D},\ \mathcal{U} \subseteq \mathcal{C}(N)$，且 $\mathcal{D}\cap\mathcal{U} = \emptyset$，并且

$$\sum_{|j\rangle\in\mathcal{D}} |\alpha_j|^2 + \sum_{|k\rangle\in\mathcal{U}} |\alpha_k|^2 = 1.$$

振幅放大方法是一类量子算法，旨在提高测量时获得目标态的概率。文献 [4]，中介绍了相关方法；作者提出了多种振幅放大算法，每种算法对目标态数量以及用户对该数量的了解程度都作出了不同假设。

就我们的目的而言，我们希望能够在目标态数量任意（可能为 0），且用户不知道该数量的情况下应用振幅放大。QSearch 所作的假设与满足这些要求最为接近。QSearch 假设至少存在一个目标态，但目标态的实际数量未知。它通过构造无测量量子算法 Q 来实现，该算法可以重复应用，以提高测量时得到目标态的概率。我们将 Q 定义为 $Q = -\mathcal{A}S_0\mathcal{A}^{-1}S_\chi$，其中

- $\mathcal{A}^{-1}$ 是通过对 $\mathcal{A}$ 取逆运算得到的量子算法。
- S_0 是无测量量子算法，对于除 $|0\rangle^{\otimes N}$ 以外的所有计算基态，它的作用都是恒等变换；而对于 $S_0|0\rangle^{\otimes N} = -|0\rangle^{\otimes N}$。因此，对于$|x\rangle \in \mathcal{C}(N)$

$$S_0|x\rangle = \begin{cases} -|x\rangle & \text{if } |x\rangle = |0\rangle^{\otimes N} \\ |x\rangle & \text{if } |x\rangle \neq |0\rangle^{\otimes N} \end{cases}$$

- S_χ 是无测量量子算法，它会翻转所需计算基态的符号。因此，对于 $|x\rangle \in \mathcal{C}(N)$

$$S_\chi |x\rangle = \begin{cases} -|x\rangle & \text{if } |x\rangle \in \mathcal{D} \\ |x\rangle & \text{if } |x\rangle \in \mathcal{C}(N) \setminus \mathcal{D} \end{cases}$$

有了我们的核心要素——量子算法 Q，我们就可以正式陈述 QSearch 算法了。

算法 1 QSearch

输入： 一个作用于 N 个量子比特上的无测量量子算法 $\mathcal{A}$， S_0 和S_χ 如上所定义；以及一个由 n 个量子比特组成、初始化为状态$|0\rangle^{\otimes N}$的量子态。

1：设置 $l = 0$，并令 c 为满足 $1 < c < 2$的任意常数。2：将 A 应用于 $|0\mathbf{i}^{\otimes N}$ 并进行测量。如果测量结果为目标态，则返回该状态。3：当尚未返回目标态时，循环执行以下步骤：4：将 l 增加 1，并设置 $M = \mathbf{d}c^l\mathbf{e}$。5：将适当大小的量子寄存器初始化为状态 A $|0\mathbf{i}^{\otimes N}$。6：令 j 为从 $[1, M]$ 中均匀随机选取的整数。7：将算法 Q 应用于状态 A $|0\mathbf{i}^{\otimes N}$ j 次。8：测量该状态。如果结果为目标态，则返回该状态。9：结束循环

输出： 一个目标态。

以下定理出自 [4, 定理 3] 阐述了 QSearch 所需的 $\mathcal{A}$ 和 $\mathcal{A}^{-1}$ 应用次数。为简洁起见，其证明请参阅 [4] 。

定理 1. 设 A 是一个无测量量子算法，使得 $A|0i^{\otimes N}$ 可写为 N 个态的叠加，其中 t 个是目标态。根据 t 的值，以下情况恰有一种成立：

- $t > 0$: *QSearch* 将通过 $\Theta\left(\sqrt{\frac{N}{t}}\right)$ 次调用 A 和 A^{-1}返回一个目标态。

- $t = 0$: *QSearch* 将无法终止。

定理 1保证，如果被搜索的状态中存在成功状态，就能通过使用 $\Theta\left(\sqrt{N}\right)$ 次 $\mathcal{A}$的应用找到该状态。然而，如果不存在这样的状态，该算法就不会终止。为了在我们的设定中应用 QSearch，我们必须确保它能够终止，无论是否存在目标态。

3.2 为 QSearch 添加停止准则

在本节中，我们将介绍一种用于修改后的QSearch的停止准则。当目标态存在时，该准则能够使算法终止的概率任意低。

假设我们可以使用一个无测量量子算法 A，使得 $A|0\rangle^{\otimes N}$可以表示为由 N 个状态构成的叠加态，其中 t 个是目标态。正如定理 1 所述，如果 $t = 0$，那么 QSearch 将无法终止。除非我们测量每个 N 个状态，否则不可能确定 $t = 0$。不过，利用我们对 N 以及 QSearch 已执行的迭代次数的了解，我们可以界定 $t > 0$ 的概率。假设 $0 < \frac{t}{N} < \frac{3}{4}$，且 M是一个满足 $M > \sqrt{N}$的整数。那么，正如定理 3], 的证明所示，如果我们应用 Q 共 j 次，其中整数 j 从 $[1, M]$ 中均匀随机选取，则获得目标态的概率下界为

$$\frac{1}{2}\left(1 - \frac{1}{2M\sqrt{\frac{t}{N}}}\right). \tag{5}$$

我们利用这一结论证明以下引理：

引理 2. 假设我们可以访问一个无测量量子算法 A，使得 $A|0\rangle^{\otimes N}$ 可以表示为 N 个状态的叠加态，其中 t 个是目标态。令$M \in \mathbb{Z}$、$M > \sqrt{N}$，并且 $j \in \mathbb{Z}$是从 $[1, M]$ 中均匀随机选取的一个整数。如果$0 < \frac{t}{N} < \frac{3}{4}$，且我们已经应用了 *QSearch* 的 u 次迭代，每次都未能产生目标态，那么在 *QSearch* 的下一次迭代中未能找到目标态的概率上界为 $\left(\frac{3}{4}\right)^u$。

证明。 在证明定理 1 ([4, 定理 3]）时， [4] 表明，如果应用 Q， j 次，则获得目标态的概率下界为$\frac{1}{2}\left(1 - \frac{1}{2M\sqrt{\frac{t}{N}}}\right)$。由于我们事先不知道 t 的值，我们假设 $t = 1$ 的最坏情况，并将该界限降低，从而得到我们的结果。

$$\begin{aligned}\frac{1}{2}\left(1 - \frac{1}{2M\sqrt{\frac{t}{N}}}\right) &= \frac{1}{2}\left(1 - \frac{\sqrt{N}}{2M\sqrt{t}}\right) \\ &\geq \frac{1}{2}\left(1 - \frac{\sqrt{N}}{2M}\right).\end{aligned}$$

对于 $M > \sqrt{N}$，我们有

$$\frac{1}{2}\left(1 - \frac{\sqrt{N}}{2M}\right) > \frac{1}{4}.$$

因此，未能找到目标态的概率最多为$3\ 4$。由于QSearch的每次迭代相互独立，在u次QSearch迭代中未能找到目标态的概率最多为 $\left(\frac{3}{4}\right)^u$。 □

利用引理 2，我们可以确定漏掉现有目标解的概率何时已低于任意给定的容差值。现在，我们具备了介绍修改后的QSearch算法所需的工具。

算法 2 修改后的QSearch

输入： 一个作用于 N 个量子比特上的无测量量子算法 $\mathcal{A}$， S_0 以及S_χ （如上所定义）、容差值 $\tau > 0$，以及一个由 n 个量子比特组成并初始化为状态 $|0\rangle^{\otimes N}$ 的量子态。

1: 设置 $l = 0$ ，并令 c 为满足 $1 < c < 2$的任意常数。2: 设置 $u = 0$。3: 将 A 应用于 $|0\mathrm{i}^{\otimes N}$ 并进行测量。如果测量结果是目标态，则返回该态。4: 当尚未返回目标态且 $u <$ln(τ) ln($\frac{3}{4}$) 时，执行 5: 将 l 增加 1，并设置 $M =$dc^le。6: 如果 $M > \sqrt{N}$ ，则执行 7: 设置 $u = u + 1$。8: 结束 if。9: 将适当大小的量子寄存器初始化为状态 A $|0\mathrm{i}^{\otimes N}$。10: 将 j 设为从 $[1, M]$ 中均匀随机选取的整数。11: 将算法 $\bar{}Q$ 应用于状态 A $|0\mathrm{i}^{\otimes N}$ j 次。12: 测量该状态。如果测得的是目标态，则返回该目标态。13: 结束 while。14: 如果未找到目标态，则返回未能找到目标态的信息。

输出： 目标态，或未能找到目标态。

介绍了我们修改后的QSearch算法后，现在我们来展示如何使用它执行给定GPS算法的搜索步骤。

4 QuantumSearchStep

在本节中，我们将介绍我们的主要产品——量子搜索步骤算法。该算法使用 $\underline{\mathcal{O}}\left(\sqrt{N}\right)$ 次量子预言机调用，完成给定 GPS 算法的搜索步骤；相比之下，需要 $\mathcal{O}(N)$ 次经典预言机调用。在这里，我们遵循标准假设，即量子预言机调用与经典预言机调用之间的计算成本可以忽略不计 [6],[7],[3]。令 $X = \{x_j\}_{j=0}^{N-1}$ 。回顾一下，我们将 $x \in X$ 表示为一个 dn 二进制字符串，其中第 k 组 d 比特表示该点的第 k 个元素。对于给定的目标函数 $f\colon \mathbb{R}^n \to \mathbb{R}$，我们假设可以访问量子预言机 F，该预言机能够实现以下变换

$$F \sum_{j=0}^{N-1} |x_j\rangle |0\rangle \to \sum_{j=0}^{N-1} |x_j\rangle |f(x_j)\rangle .$$

在第4.1 节中，我们展示了如何实现算子 Q ，该算子是在正式陈述量子搜索步骤算法之前使用修改后的QSearch算法所必需的。在第4.2节中，我们讨论了量子搜索步骤算法相较于经典方法的优势、量子搜索步骤算法如何保留给定 GPS 算法原有的收敛性质，以及为何我们选择不使用修改后的QSearch算法来执行轮询步骤。

4.1 准备Q算子

回顾一下，我们的目标是使用修改后的QSearch算法，从网格中的一组点中搜索一个改进的网格点，如第 2.1 节所述。此应用要求我们构造算子 $Q = \mathcal{A}S_0\mathcal{A}^{-1}S_\chi$，这等价于实现算子 $\mathcal{A}$、 S_0、$\mathcal{A}^{-1}$ 和 S_χ。本节将展示如何实现以下 Q 的组成部分：

- $\mathcal{A}$，使得$\mathcal{A}\left|0\right\rangle^{\otimes N}$ 产生我们希望搜索的点的叠加态。
- S_0 是无测量量子算法，对于除 $\left|0\right\rangle^{\otimes N}$之外的所有计算基态均执行恒等运算，而 $S_0\left|0\right\rangle^{\otimes N} = -\left|0\right\rangle^{\otimes N}$。因此，对于$\left|x\right\rangle \in \mathcal{C}(N)$

$$S_0\left|x\right\rangle = \begin{cases} -\left|x\right\rangle & \text{if } \left|x\right\rangle = \left|0\right\rangle^{\otimes N} \\ \left|x\right\rangle & \text{if } \left|x\right\rangle \neq \left|0\right\rangle^{\otimes N} \end{cases}$$

- $\mathcal{A}^{-1}$ 是通过对$\mathcal{A}$取逆运算得到的量子算法。
- S_χ 是对对应于改进网格点的计算基态翻转符号的无测量量子算法。因此，对于 $\left|x\right\rangle \in \mathcal{C}(N)$

$$S_\chi\left|x\right\rangle = \begin{cases} -\left|x\right\rangle & \text{if } x \text{ is an improved mesh point} \\ \left|x\right\rangle & \text{if } x \text{ is not an improved mesh point} \end{cases}$$

S_χ 是构造 Q 时面临的最大挑战，因为这要求我们能够标记改进网格点。为此，我们设计 $\mathcal{A}$，使其作用于三个量子寄存器。前两个量子寄存器与预言机 F所使用的量子寄存器相同。第三个量子寄存器将用于比较当前迭代点处的目标函数值与我们希望搜索的各点处的目标函数值。

设 $X = \{x_j\}_{j=0}^{N-1}$ 是从网格中选取的 N 个点， $f(x_j) = f_j$ 表示在点 x_j 处计算的目标函数， x_k 表示当前迭代点。我们将设计$\mathcal{A}$使得

$$\mathcal{A}\left|0\right\rangle = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\left|x_j\right\rangle\left|f_j\right\rangle\left|f_j - f_k\right\rangle. \tag{6}$$

$\left|f_j - f_k\right\rangle$ 的符号量子比特当且仅当 $\left|x_j\right\rangle\left|f_j\right\rangle\left|f_j - f_k\right\rangle$ 是目标态时，处于状态 $\left|1\right\rangle$。在构造 S_χ 时，我们可以将该量子比特用作控制量子比特。于是我们的算法 $\mathcal{A}$ 如下：

算法 3 $\mathcal{A}$

输入： 设 $f(x_k) = f_k$ 表示在当前迭代点 x_k 处计算的目标函数，并设 $X = \{x_j\}_{j=0}^{N-1}$ 是从网格中选取的 N 个点。

1: 将第三个寄存器置于状态 $\left|-f_k\right\rangle$

$$\left|\phi_0\right\rangle = \left|0\right\rangle\left|0\right\rangle\left|-f_k\right\rangle.$$

2: 将第一个寄存器置于 N 点的均匀叠加态。

$$\left|\phi_1\right\rangle = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\left|x_j\right\rangle\left|0\right\rangle\left|-f_k\right\rangle.$$

3: 将预言机 F 应用于前两个寄存器

$$|\phi_2\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |x_j\rangle |f_j\rangle |f_k\rangle .$$

4: 使用原位有符号量子傅里叶变换加法算法，将第二个量子寄存器加到第三个量子寄存器上，该算法在 [8, 第 3 节。1]

$$|\phi_3\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |x_j\rangle |f_j\rangle |f_j - f_k\rangle .$$

输出： $|\phi_3\rangle$

在此情境下，目标态和非目标态的集合自然定义为

$$\mathcal{D} := \{|x_j\rangle |f_j\rangle |f_j - f_k\rangle : f_j - f_k < 0\}$$
$$\mathcal{U} := \{|x_j\rangle |f_j\rangle |f_j - f_k\rangle : f_j - f_k \geq 0\}.$$

定义了$\mathcal{A}$ 之后，$\mathcal{A}^{-1}$ 可轻易通过对$\mathcal{A}$取逆运算得到。为了实现 S_0，我们先考虑由酉矩阵定义的单量子比特门

$$U = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

该门在计算基上的作用为 $U|0\rangle = -|0\rangle$ 和 $U|1\rangle = |1\rangle$。因此，U 对单个量子比特的作用是 S_0。然而，我们的算法始终需要多个量子比特。对于任意数量的量子比特，我们可以使用 U 来实现 S_0：对第一个量子比特应用 U，并以其余量子比特处于状态 $|0\rangle$为控制条件。

与 S_0一样，我们可以将 S_χ 实现为单量子比特门的受控版本。回想一下，应用$\mathcal{A}$之后，如果比较寄存器表示的是改进的网格点，其符号量子比特便处于状态 $|1\rangle$；否则，它处于状态 $|0\rangle$。考虑由下列矩阵表示的单量子比特门

$$V = \begin{bmatrix} e^{i\pi} & 0 \\ 0 & e^{i\pi} \end{bmatrix}.$$

我们可以通过将 V 应用于输入寄存器的第一个量子比特来实现 S_χ，并以比较寄存器的符号量子比特处于状态 $|1\rangle$ 为控制条件。现在我们已经掌握了制备 Q 所需全部要素的方法，可以介绍量子搜索步骤了。

算法 4 量子搜索步骤

输入： $\mathcal{A}$、S_0、$\mathcal{A}^{-1}$ 和 S_χ（定义如上）、待搜索的点集 $\{x_j\}_{j=0}^{N-1}$、当前迭代点 x_k、容差值 $\tau_0 > 0$，以及处于状态$|\psi\rangle = |0\rangle |0\rangle |0\rangle$的三个量子寄存器。

1: 将修改后的QSearch应用于 $|\psi\rangle$.2: **如果**修改后的QSearch返回一个改进的网格点，则 x' **那么**3: 设置 $x_{k+1} = x'$ 并更新网格尺寸参数。4: **否则**终止算法并返回未能找到改进的网格点。

5: **结束 if**

输出： 下一个迭代点 x_{k+1}，或未能找到改进的网格点。

有了量子搜索步骤，我们现在来讨论它对收敛性的影响，以及它与经典方法的比较。

4.2 收敛性与量子优势

在本节中，我们讨论修改后的 QSearch 可能对 GPS 算法的收敛性质产生的影响，以及量子搜索步骤相比经典方法的优势。

值得注意的是，搜索步骤与轮询步骤之间唯一的区别在于所考虑的点集。将一次迭代划分为这些步骤的动机，在于它们各自在给定的 GPS 算法中承担着独特的作用。搜索步骤允许从网格中选择范围更广、种类更多的点。这是因为它对所考虑的点几乎没有限制：每个点都必须来自给定的网格，且点的总数必须有限。这使得搜索步骤在实现方式上具有令人惊叹的灵活性。在实践中，这种灵活性是 GPS 算法有效性的关键 [2]。然而，这种灵活性也使搜索步骤不足以保证算法收敛。这正是轮询步骤的作用。在 [2, 定理 3.7、3.9、3.14 以及推论 3.11、3.12、3.16], 中，Audet 和 Dennis 给出了多种不同的收敛结果，它们对目标函数的假设各不相同。不过，尽管这些结果都假设轮询步骤按照第2.1节所述的方式执行，但并未对搜索步骤作出任何假设。因此，我们可以将轮询步骤视为确保算法收敛的附加步骤。这就是我们将修改后的 QSearch 限定用于搜索步骤的原因。虽然我们可以确保漏掉改进网格点的概率任意低，但若不将预言机调用次数增加到0 ，就无法确保该概率为$\mathcal{O}(N)$。

假设我们希望考察网格中的 N 个点。经典方法需要调用预言机 $\mathcal{O}(N)$ 次才能完成搜索步骤。然而，量子搜索步骤可以通过调用预言机 $\mathcal{O}\left(\sqrt{N}\right)$ 次来搜索相同的 N 个点。现在，我们以以下定理总结量子搜索步骤的优势，作为本节的结尾：

定理 3. 设 X 是从网格中选取的 N 个点的集合，其中 t 个是改进网格点。以下情况恰有一种成立

- 量子搜索步骤将通过调用量子预言机 $\Theta\left(\sqrt{\frac{N}{t}}\right)$ 次，返回一个改进的网格点。
- 量子搜索步骤将终止，且未找到下一个迭代点。

如果 X 中包含一个改进的网格点，则可以将量子搜索步骤在未找到下一个迭代点时终止的概率任意降低。

证明。 如果量子搜索步骤返回一个改进的网格点，定理1 保证它将通过 $\Theta\left(\sqrt{\frac{N}{t}}\right)$ 次对量子预言机的调用来实现。

假设算法在未生成下一个迭代点的情况下终止，且 X 包含一个改进的网格点。根据引理 2，下一次迭代中未能找到改进网格点的概率上界为 $\left(\frac{3}{4}\right)^u$，其中 u是在 $M > \sqrt{N}$ 之后执行修改后的QSearch的迭代次数。因此，对于 $\tau > 0$，如果 $u < \frac{\ln(\tau)}{\ln\left(\frac{3}{4}\right)}$，那么

$$
\begin{aligned}
u &< \frac{\ln(\tau)}{\ln\left(\frac{3}{4}\right)} \\
uln\left(\frac{3}{4}\right) &< ln(\tau) \\
ln\left(\left(\frac{3}{4}\right)^u\right) &< ln(\tau) \\
\left(\frac{3}{4}\right)^u &< \tau
\end{aligned}
$$

并且，量子搜索步骤在下一次迭代中未能找到改进网格点的概率小于 τ。

□

5 总结

在这项工作中，我们介绍了振幅放大算法 QSearch 的一个修改版本 [4, 定理 3]。修改后的 QSearch 可以保证算法在有限步数内终止；若目标态存在，则未能生成目标态的概率可以任意低。随后，我们展示了如何使用修改后的 QSearch 来实现量子搜索步骤，这是一种用于执行给定 GPS 算法的搜索步骤的量子算法。在搜索 N 个点时，其中 t 个是改进网格点，我们的算法能够通过$\mathcal{O}\left(\sqrt{N}\right)$次量子预言机调用完成搜索步骤，而传统方法则需要$\mathcal{O}(N)$次经典预言机调用。如果 $t > 0$，我们的量子搜索步骤未能识别出改进网格点的概率可以任意低。

参考文献

[1] Srinivasan Arunachalam. 布尔可满足性与无导数优化的量子加速。硕士论文，滑铁卢大学，2014 年。[2] Charles Audet 和 J. E. Dennis. 广义模式搜索分析。《*SIAM* 优化期刊》，13(3)：889–903，2002 年。[3] Ethan Bernstein 和 Umesh Vazirani. 量子复杂性理论。《*SIAM* 计算期刊》，26(5)：1411–1473，1997 年。[4] Gilles Brassard、Peter Høyer、Michele Mosca 和 Alain Tapp. 量子振幅放大与估计。《量子计算与信息》，第 53–74 页，2002 年。[5] Harry Buhrman、Richard Cleve、John Watrous 和 Ronald de Wolf. 量子指纹。《物理评论快报》，87(16)：167902，2001 年。[6] András Gilyén、Srinivasan Arunachalam 和 Nathan Wiebe. 通过更快的量子梯度计算优化量子优化算法。*ArXiv*，abs/1711.00465，2017 年。

[7] Stephen P. Jordan. 数值梯度估计的快速量子算法。《物理评论快报》，95：050501，2005 年。[8] Engin Sahin. 基于有符号整数上的量子傅里叶变换的量子算术运算。《国际量子信息杂志》，18(06)：2050035，2020 年。[9] Virginia Torczon. 模式搜索算法的收敛性。《*SIAM* 优化期刊》，7(1)：1–25，1997 年。