



## Article

# $\ell_0$ Optimization with Robust Non-Oracular Quantum Search

Tianyi Zhang and Yuan Ke \*

Department of Statistics, The University of Georgia, Athens, GA 30605, USA; tianyi.zhang@uga.edu

\* Correspondence: yuan.ke@uga.edu

**Abstract:** In this article, we introduce an innovative hybrid quantum search algorithm, the Robust Non-oracle Quantum Search (RNQS), which is specifically designed to efficiently identify the minimum value within a large set of random numbers. Distinct from the Grover's algorithm, the proposed RNQS algorithm circumvents the need for an oracle function that describes the true solution state, a feature often impractical for data science applications. Building on existing non-oracular quantum search algorithms, RNQS enhances robustness while substantially reducing running time. The superior properties of RNQS have been demonstrated through careful analysis and extensive empirical experiments. Our findings underscore the potential of the RNQS algorithm as an effective and efficient solution to combinatorial optimization problems in the realm of quantum computing.

**Keywords:** combinatorial optimization; Grover's algorithm; minimum voting; quantum computing; robustness

## 1. Introduction

Suppose we have a large unsorted set  $\mathcal{S}$  of real numbers randomly generated from a bounded interval. The objective of interest is to accurately and efficiently find the minimum element(s) in  $\mathcal{D}$ . Let  $S$  be the cardinality of  $\mathcal{S}$  and  $q$  be a positive integer such that  $2^{q-1} < S \leq 2^q$ . To keep the presentation focused, we assume the minimum is unique and we can always enlarge  $\mathcal{S}$  to  $\mathcal{D}$  whose cardinality is exactly  $D = 2^q$  by adding some arbitrarily large real numbers to  $\mathcal{S}$ . Then,  $\mathcal{S}$  and  $\mathcal{D}$  share the same minimum element(s). In addition, searching the minimum over  $\mathcal{D}$  is computationally more challenging than searching the minimum over  $\mathcal{S}$ . When the minimum element of  $\mathcal{D}$  is unique, any algorithm implemented on an electronic computer will take at least  $D/2$  "moves" to find the minimum with a success probability greater than 50%. Therefore, unfortunately, this simple minimum searching task, as a snapshot of many  $\ell_0$  optimization problems, suffers from NP-hard computational bottlenecks when  $q$  is moderate or large.

Quantum computing represents a groundbreaking approach to computation, leveraging the principles of quantum mechanics to process information. In contrast to classical computers, which utilize binary bits, quantum computers employ quantum bits (qubits) that admit superpositions on a unit sphere. The capacity of quantum states to characterize information often expands exponentially as the system's size increases. Specifically, a qubit-based quantum system with  $q$  qubits has the potential to represent a myriad of superpositions of  $2^q$  distinct orthonormal states concurrently. In contrast, classical systems are limited to representing a single state at any given moment [1]. Such a revolutionary shift in understanding has sparked substantial advancements in crafting scalable quantum algorithms. Recently, hybrid quantum computing algorithms have drawn huge attention. We refer to Refs. [2–4] as representative of the literature, among many others.

However, existing quantum search algorithms, e.g., Refs. [5–9], are not tailored for the minimum searching problem stated at the beginning of the section as they require some oracular knowledge of the solutions. For instance, the renowned Grover's algorithm [5] necessitates an oracle function capable of associating all solutions with a specific state and all non-solutions with the other state. Yet, in data science contexts, obtaining this

**Citation:** Zhang, T.; Ke, Y. $\ell_0$  Optimization with Robust Non-Oracular Quantum Search.*Technologies* **2023**, *11*, 148.[https://doi.org/10.3390/](https://doi.org/10.3390/technologies11050148)[technologies11050148](https://doi.org/10.3390/technologies11050148)

Academic Editor: Bernard Gil

Received: 20 September 2023

Revised: 7 October 2023

Accepted: 16 October 2023

Published: 19 October 2023

**Copyright:** © 2023 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license ([https://](https://creativecommons.org/licenses/by/4.0/)[creativecommons.org/licenses/by/](https://creativecommons.org/licenses/by/4.0/)

4.0/).

precise oracle insight is often challenging, primarily because the solution typically emerges from randomly observed datasets. In the works by Biham [10,11], Grover's algorithm underwent modifications to cater to both a generic initial superposition and a mixed initial state, ensuring the algorithm's robustness even when minuscule errors occur in quantum computers. Furthermore, Ref. [12] introduced the notion of amplitude amplification, broadening the applicability of Grover's algorithm to an extensive array of quantum search challenges. Nevertheless, the requirement of the oracle information about the solution states is not waived.

Recently, the authors of Ref. [13] showed that Grover's algorithms could perform as badly as random guesses when the oracle function is inaccurate or missing. To overcome the problem, a novel algorithm called Non-oracle Quantum Search (NQS) was developed to search the minimum without any oracle function. NQS works like a "quantum elevator" that iteratively descends an initial superposition to minimize the loss function. During each iteration, the updated superposition is juxtaposed against the previous version using a localized evaluation function that operates independently of any oracle-based knowledge of the correct solution. When searching for the smallest element in  $\mathcal{D}$ , NQS boasts a computational complexity by a superpolynomial rate. Notably, this rate is only greater by a logarithm factor when compared to the foundational limits set for oracular quantum search algorithms [14].

Despite its superior theoretical and empirical properties, NQS suffers from a numerical instability issue; the "quantum elevator" may become stuck at a quantum state and fail to descend to a better state over a huge number of iterations. To tackle this critical issue, we propose a robust extension of NQS named Robust Non-oracle Quantum Search (RNQS). RNQS utilizes multiple quantum nodes to improve the robustness of quantum search. We redesigned the step sizes and introduced a minimum voting scheme to overcome the numerical instability issue in NQS. We use extensive numerical experiments to demonstrate the advantages of RNQS over NQS. Our contributions can be summarized in the following three points: (1) we identify the limitations of NQS; (2) we propose a novel RNQS algorithm to address the numerical instability issues; and (3) RNQS is numerically accurate and much more efficient than existing quantum search algorithms.

## 2. Review of Quantum Search Algorithms and Their Limitations

### 2.1. Notations

The foundation of quantum computing is rooted in the state-space postulate, signifying that every system state aligns with a unit vector within a Hilbert space. As an illustration, a  $q$  qubit quantum computer aligns with the superposition of  $D = 2^q$  states, allowing its representation through vectors in a  $D$  dimensional Hilbert space  $\mathcal{H}$ . For example, denote every state  $|\psi\rangle \in \mathcal{H}$  admits a decomposition  $|\psi\rangle = \sum_{i=0}^{D-1} \phi_i |i\rangle$ , where  $\{|i\rangle\}_{i=0}^{D-1}$  is an orthonormal basis of  $\mathcal{H}$ . Another distinctive characteristic of quantum computing emerges in the measurement phase: assessing a quantum state leads to a probabilistic result rather than a fixed one. Upon measuring  $|\psi\rangle$ , the system undergoes a collapse to a stochastic state in  $\mathcal{D}$ , with the probability of landing on  $|i\rangle$  being  $|\phi_i|^2$  for indices  $i = 0, \dots, D-1$ .

### 2.2. Grover's Algorithm and Its Limitations

For the minimum searching task stated in the Introduction section, we can represent the elements of  $\mathcal{D}$  by  $\{|i\rangle\}_{i=0}^{D-1}$ . Then, the minimum element corresponds to a quantum state, say  $|k\rangle$ . Grover's algorithm operates under the premise that there is an oracle function, denoted as  $S(\cdot)$ . For this function,  $S(|k\rangle) = 1$  while  $S(|i\rangle) = 0$  for all  $i$  where  $i \neq k$ . The algorithm commences with a uniformly distributed superposition across the orthonormal basis. More precisely, this starting superposition is articulated as

$$|\psi_0\rangle = \frac{1}{\sqrt{D}} \sum_{i=0}^{D-1} |i\rangle \equiv c_0 |k\rangle + d_0 \sum_{i \neq k} |i\rangle,$$

where  $c_0 = d_0 = \frac{1}{\sqrt{D}}$ . Let  $\theta$  be the angle that satisfies  $\sin^2 \theta = \frac{1}{D}$ . After the  $j$ -th iteration, the coefficients are updated to  $c_j$  and  $d_j$  using Grover's operations, which admit a closed form [1]:

$$\begin{cases} c_j = \sin((2j+1)\theta), \\ d_j = \frac{1}{\sqrt{D-1}} \cos((2j+1)\theta). \end{cases}$$

Consider the vector  $|\zeta\rangle = \frac{1}{\sqrt{D-1}} \sum_{i \neq k} |i\rangle$  which represents the mean of all states that are not solutions and hence are orthogonal to the desired solution state. From a geometrical perspective, every iteration within Grover's algorithm can be visualized as a rotation of the superposition  $|\psi_j\rangle$  in the direction of the solution state  $|k\rangle$  by an angle of  $2\theta$ . Grover's method typically concludes once  $c_j$  nears 1. Consequently, an intuitive selection for the iteration count  $\tau$  would be such that  $(2\tau+1)\theta \approx (2\tau+1)/\sqrt{D} = \pi/2$ . This approximation stands true especially when  $D$  is large, leading to a smaller  $\theta$  and where  $\theta$  closely mirrors  $\sin \theta = \frac{1}{\sqrt{D}}$ . As a result,  $\tau$  is roughly  $\lceil \sqrt{D}\pi/4 \rceil$ , where  $\lceil \cdot \rceil$  represents the ceiling function. The steps of Grover's algorithm can be delineated in Algorithm 1.

---

**Algorithm 1** Grover's algorithm [15]

---

**Input:** A set  $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$  with  $D = 2^q$ ; a binary evaluation function  $S$  associated with the oracle state  $|k\rangle$ , such that  $S(|k\rangle) = 1$  and  $S(|i\rangle) = 0$  for  $i \neq k$ ; number of iterations  $\tau = \lceil \pi\sqrt{D}/4 \rceil$ .

**Initialization:** Prepare a superposition  $|\psi_0\rangle = \frac{1}{\sqrt{D}} \sum_{i=0}^{D-1} |i\rangle$  on a quantum register of  $q$ -qubits.

**for**  $j = 1, \dots, \tau$ ; **do**

Grover's operation: Let  $|\psi_j\rangle = GF|\psi_{j-1}\rangle$ , where  $F|i\rangle = (1 - 2S(|i\rangle))|i\rangle$ ,  $G = 2|\psi_0\rangle\langle\psi_0| - \mathbf{I}_D$ , and  $\mathbf{I}_D$  is a  $D \times D$  identity matrix.

**end for**

**Output:** Measure the latest superposition  $|\psi_\tau\rangle$  on the quantum register.

---

The efficacy of Grover's algorithm is deeply intertwined with the precision of the oracle data. These data furnish a function that flags the solution state as 1 and all other states as 0. But, in fields like statistics and machine learning, these exact oracle data are hard to come by, given that states are often discerned from randomly drawn samples. As highlighted in Ref. [13], experiments depicted the compromised performance of Grover's method when equipped with only partial or imprecise details about the solution state. When the solution state can be pinpointed only to a subset of states, represented as  $|k\rangle \in \mathcal{M} \subset \{|0\rangle, \dots, |D-1\rangle\}$ , Grover's approach nudges the superposition in the direction of the hyperplane shaped by the states within  $\mathcal{M}$ . This maneuvering introduces a skew, obstructing the algorithm's convergence. In the absence of any oracle knowledge about the solution state, the function  $S(\cdot)$  is crafted using a solution state picked haphazardly. This configuration amplifies the chances of Grover's algorithm skewing the starting superposition towards a misaligned state, making its efficacy on par with mere chance-based selections.

### 2.3. Non-Oracular Quantum Search and Its limitations

To circumvent the need for oracle data in quantum searches, a data-responsive algorithm termed Non-oracular Quantum Search (NQS) was introduced in Ref. [13], detailed in Algorithm 2. In brief, NQS begins with a random state in  $\mathcal{D}$  as the benchmark state; then, it iteratively updates the benchmark state to reduce a pre-defined state loss function  $g(\cdot)$ .

**Algorithm 2** Non-oracular Quantum Search (NQS) [13]

**Input:** An orthonormal basis  $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$  of size  $D = 2^q$ , a state loss function  $g(\cdot)$  that maps a state in  $\mathcal{D}$  to a real number, and a learning rate  $\lambda \in (0, 1)$ .

**Initialization** Set  $m = 1$ . Randomly select a state in  $\mathcal{D}$  as the initial benchmark state  $|w\rangle$ . Define a local evaluation function  $S(\cdot, |w\rangle, g)$  such that  $S(|i\rangle, |w\rangle, g) = 1$  if  $g(|i\rangle) \leq g(|w\rangle)$  and  $S(|i\rangle, |w\rangle, g) = 0$  if  $g(|i\rangle) > g(|w\rangle)$ .

**repeat**

(1) Run Algorithm 1 by inputting  $\mathcal{D}$ ,  $S(\cdot, |w\rangle, g)$  and  $\tau(m) = \lceil \pi \lambda^{-m/2} / 4 \rceil$ .

(2) Measure the quantum register and denote the readout by  $|w^{new}\rangle$ .

(3) If  $g(|w^{new}\rangle) < g(|w\rangle)$ , set  $|w\rangle = |w^{new}\rangle$  and update  $S(\cdot, |w\rangle, g)$  accordingly.

(4)  $m = m + 1$ .

**until**  $m > C(\lambda) \ln D$ , where  $C(\lambda)$  is a positive constant depends on  $\lambda$ .

**Output:** The latest benchmark state  $|w\rangle$ .

Without loss of generality, states within  $\mathcal{D}$  are labeled in an increasing sequence based on their respective state loss function values as

$$g(|0\rangle) < g(|1\rangle) \leq g(|2\rangle) \leq \dots \leq g(|D-1\rangle).$$

Here,  $|0\rangle$  represents the unique solution state, while  $|w\rangle$  serves as the benchmark state at the  $m$ -th iteration of the process. According to Ref. [13], after  $\tau(m) = \lceil \pi \lambda^{-m/2} / 4 \rceil$  Grover's operations, NQS updates the state from an equally weighted superposition  $|\psi_0\rangle$  to  $|\psi_{\tau(m)}\rangle$ . Specifically, the update of coefficients can be summarized by

$$\begin{cases} \alpha_{\tau(m)} = \frac{1}{\sqrt{w+1}} \sin((2\tau(m) + 1)\theta), \\ \beta_{\tau(m)} = \frac{1}{\sqrt{D-w-1}} \cos((2\tau(m) + 1)\theta), \end{cases}$$

where  $\sin^2 \theta = (w+1)/D$ ,  $\alpha_{\tau(m)}$  is a coefficient for all solution states, and  $\beta_{\tau(m)}$  is a coefficient for all non-solution states in the  $m$ -th iteration. Consequently, the output of NQS, denoted by  $|w_{new}\rangle \in \mathcal{D}$ , satisfies a probability mass function

$$P(|w_{new}\rangle = |i\rangle) = \begin{cases} \alpha_{\tau(m)}^2, & \text{if } i \leq w, \\ \beta_{\tau(m)}^2, & \text{if } i > w. \end{cases}$$

This suggests that NQS augments the likelihood of states exhibiting a loss lesser than that of  $|w\rangle$  while diminishing the probability of states with a loss exceeding that of  $|w\rangle$ . The probability of updating in the  $m$ -th iteration is given by

$$P_m(g(|w_{new}\rangle) \leq g(|w\rangle)) = (w+1)\alpha_{\tau(m)}^2.$$

However, NQS has two limitations. The first limitation is that NQS may fail to converge for large-scale problems. The main reason is the step size  $\tau(m) = \lceil \pi \lambda^{-m/2} / 4 \rceil$  increases approximately exponentially regardless of the rank of the current state  $|w\rangle$ . A problem will arise when the following two conditions are met simultaneously in practice: (i) the benchmark state is not updated in the  $m$ -th iteration, and (ii)  $P_{m+1}(g(|w_{new}\rangle) \leq g(|w\rangle)) < P_m(g(|w_{new}\rangle) \leq g(|w\rangle))$ . Under these conditions, NQS will add more Grover's operations than needed to the  $(m+1)$ -th iteration, causing the algorithm to over-rotate and eventually cross the solution states. This naturally induces a convergence problem for NQS, as it fails to increase the probability to update the current state to the solution. Unfortunately, this issue is not negligible when NQS is applied to solve large-scale search problems that require a large number of iterations. For example, with  $\lceil C(\lambda) \ln(D) \rceil = \lceil -6 \log_{\lambda} 10 \ln(D) \rceil$  iterations and  $\lambda = 0.5$  as suggested in Ref. [13], there would be  $6.56 \times 10^{20}$  Grover's operations in the 139th iteration (which is the last iteration) for the  $q = 10$  case, where only 1024 states are compared.

The second limitation of NQS is its lack of robustness. While NQS can quickly find the correct solution in the majority of attempts, its performance varies significantly among replications. For example, when we repeatedly apply NQS to a dataset, it can find the correct solution state within a few hundred Grover's operations in some attempts but may fail to find the solution even after millions of Grover's operations. The inconsistency in NQS's robustness is demonstrated in Section 4.

### 3. Robust Non-Oracular Quantum Search

#### 3.1. Methodology

A natural idea to overcome the limitations of NQS is to avoid increasing  $\tau(m)$  immediately after finding that the state has not been updated. This idea motivates the algorithm we are introducing, which is articulated through the subsequent three pivotal stages:

- (i) Initialization: We randomly select an initial benchmark state  $|w\rangle$  from  $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$ . Additionally, we pre-specify a learning rate  $\lambda \in (0, 1)$ .
- (ii) Updating: We treat  $|w\rangle$  as an initial guess of the solution state and run Algorithm 1 over  $\mathcal{D}$  with  $\tau = \lceil \pi \lambda^{-m/2} / 4 \rceil$  iterations on  $q$  quantum nodes, where  $m$  is a positive integer. We denote the  $q$  independent outputs as  $|w_j^{new}\rangle$ , which are states in  $\mathcal{D}$ , with  $j = 1, \dots, q$ . Next, we choose the state with the smallest loss, denoted as  $|w^{new}\rangle$ , among the  $q$  candidates and compare it with  $|w\rangle$  in terms of their state loss  $g(\cdot)$ . If  $g(|w^{new}\rangle) < g(|w\rangle)$ , we update the benchmark state  $|w\rangle$  to  $|w^{new}\rangle$ ; if not,  $|w\rangle$  remains unchanged.
- (iii) Iteration and output: Initiating with  $m = 1$ , we continue with the step of updating. Post every iteration,  $m$  is incremented by one unit. The RNQS procedure halts when the condition  $m > C_1(\lambda)(\ln q)^\alpha + C_2$  is met, where  $C_1(\lambda)$  and  $C_2$  are positive constants and  $C_1(\lambda)$  may depend on the learning rate  $\lambda$ . Based on our experiments, the rule-of-thumb recommendations are  $C_1(\lambda) = -0.02 \log_\lambda 10$ ,  $C_2 = 4$ , and  $\alpha = 5$ . After the final iteration, we assess the quantum register on  $q$  occasions using the most recent  $q$  superpositions. The result is the state exhibiting the least loss among the  $q$  potential states.

RNQS differs from NQS in each iteration, as it measures the superpositions  $q$  times and votes for the minimum. By performing multiple measurements, RNQS effectively slows down the increment of  $\tau$ . Additionally, RNQS not only attempts to find a state that updates the oracle in each iteration but also seeks a state with a significantly smaller loss compared to NQS. This approach is inspired by the advantage of minimum voting. The algorithm involves comparisons among the  $q$  measured states in each iteration, for which it is acceptable to use any classical algorithm with a complexity of  $O(q) = O(\log D)$ . We summarize RNQS in Algorithm 3 below. In Figure 1, we provide an illustrative example to demonstrate the mechanism of RNQS with  $q = 5$ , i.e.,  $D = 32$  states. It is evident that RNQS adaptively increases the probability of finding a state with a smaller loss. Moreover, with  $q$  measurements and minimum voting, the probability of updating in RNQS is significantly higher than that in NQS, as illustrated in the following Section 3.2.

**Algorithm 3** Robust Non-oracular Quantum Search (RNQS)

**Input:** An orthonormal basis  $\mathcal{D} = \{|i\rangle\}_{i=0}^{D-1}$  of size  $D = 2^q$ , a state loss function  $g(\cdot)$  that maps a state in  $\mathcal{D}$  to a real number, and a learning rate  $\lambda \in (0, 1)$ .

**Initialization** Set  $m = 1$ . Randomly select a state in  $\mathcal{D}$  as the initial benchmark state  $|w\rangle$ . Define a local evaluation function  $S(\cdot, |w\rangle, g)$  such that  $S(|i\rangle, |w\rangle, g) = 1$  if  $g(|i\rangle) \leq g(|w\rangle)$  and  $S(|i\rangle, |w\rangle, g) = 0$  if  $g(|i\rangle) > g(|w\rangle)$ .

**repeat**

(a) Set  $j = 1$ .

**repeat**

(1) Run Algorithm 1 by inputting  $\mathcal{D}$ ,  $S(\cdot, |w\rangle, g)$  and  $\tau(m) = \lceil \pi \lambda^{-m/2} / 4 \rceil$ .

(2) Measure the quantum register and denote the readout by  $|w_j^{new}\rangle$ .

(3)  $j = j + 1$ .

**until**  $j > q$ .

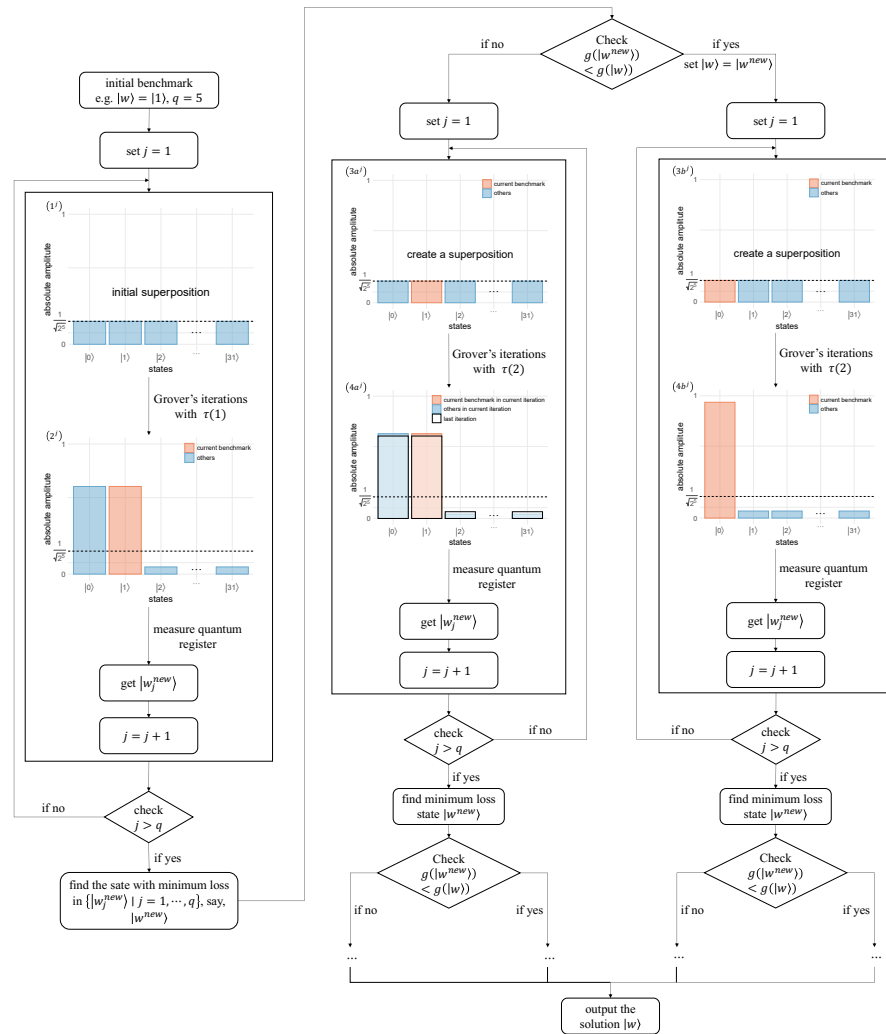
(b) Set  $|w^{new}\rangle = |w_k^{new}\rangle$ , where  $k = \arg \min_{1 \leq j \leq q} \{g(|w_j^{new}\rangle)\}$ .

(c) If  $g(|w^{new}\rangle) < g(|w\rangle)$ , set  $|w\rangle = |w^{new}\rangle$  and update  $S(\cdot, |w\rangle, g)$  accordingly.

(d)  $m = m + 1$ .

**until**  $m > C_1(\lambda)(\ln q)^\alpha + C_2$ , where  $C_1(\lambda)$  is a positive constant depending on  $\lambda$ ,  $C_2$  is a different constant, and  $\alpha$  is a positive integer.

**Output:** The latest benchmark state  $|w\rangle$ .



**Figure 1.** An illustrative example ( $D = 2^5 = 32$ ) for Robust Non-oracular Quantum Search. The subfigures represent the absolute amplitude of states in various superpositions within the algorithm.



### 3.2. Inspirations of Minimum Voting

The idea of minimum voting originates from the hybrid NQS in Ref. [13], where NQS was implemented on  $k$  quantum nodes and the output state was chosen as the majority voting among all  $k$  states. However, when  $k$  is small, the diversified outcomes and ties can hamper the voting to reach a majority. Furthermore, the probability of an event in which the majority voting result selects the correct subset varies depending on the probability of selecting not only the solution states but also other possible states in  $\mathcal{D}$ . In other words, with a fixed probability  $p$  of correct selection in each measurement, if a few non-solution states have a comparably high probability of being selected as the solution states, the probability of correct selection in majority voting would be much smaller than in the case where all non-solution states have approximately the same probability of being selected. In fact, majority voting does not perform well enough in the NQS mechanism, since all states with relatively small loss have a considerable probability of being selected unless the number of iterations is quite large. Consequently, the corresponding states would confuse the majority voting results. Estimating the likelihood of the event where the majority voting chooses the appropriate subset, represented as  $\mathcal{E}$ , is a complex task. The subsequent theorem offers a lower limit on the successful probability associated with majority voting.

**Theorem 1.** Assume every node operates independently in a quantum-classical network that comprises  $k = 2s + 1$  quantum nodes with a probability  $p > \frac{s+1}{2s+1}$  of choosing the accurate solution state. Let  $\mathcal{A}$  represent the event where the majority vote of the output leans towards the right solution state. The probability of  $\mathcal{A}$  to occur is then bounded below by

$$\mathbb{P}(\mathcal{A}) \geq \Phi\left(\sqrt{2(s+1)D_{KL}\left(p, \frac{s+1}{2s+1}\right)}\right),$$

where  $\Phi(\cdot)$  is the standard Gaussian cumulative distribution function and

$$D_{KL}(u, v) = u \ln \frac{u}{v} + (1 - u) \ln \frac{(1 - u)}{(1 - v)}$$

is the Kullback–Leibler divergence between two Bernoulli distributions with parameters  $u$  and  $v$ .

Compared with majority voting, choosing the state with the smallest loss function value over the  $k$  states undoubtedly increases the accuracy, with an extra small computational cost of order  $O(k)$ . With the same setup, i.e.,  $k$  nodes and each node has a success probability  $p$ , the success probability of minimum voting admits

$$\mathbb{P}(\mathcal{A}^*) = 1 - (1 - p)^k = 1 - (1 - p)^{(2\xi+1)}.$$

As we can see,  $\mathbb{P}(\mathcal{A}^*) \geq \mathbb{P}(\mathcal{A})$  since the occurrence of  $\mathcal{A}$  indicates that the solution state has been selected in the  $k$  measurements, and, consequently,  $\mathcal{A}^*$  occurs, i.e.,  $\mathcal{A} \subset \mathcal{A}^*$ .

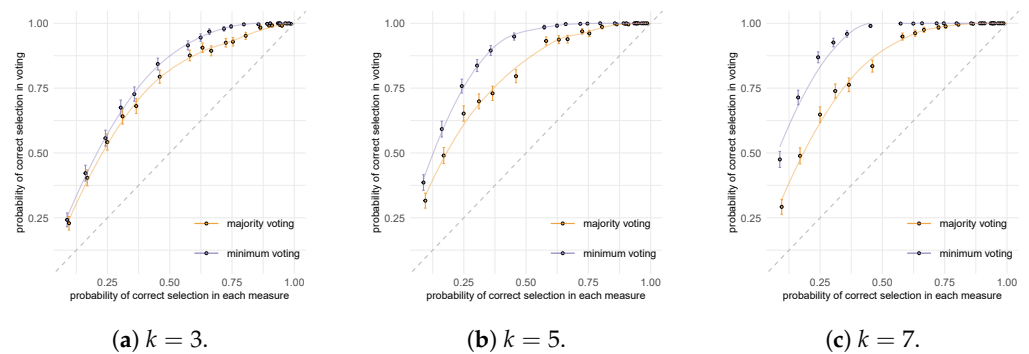
The aforementioned modification merely considers replacing the one-time majority voting in hybrid NQS with minimum voting. In contrast, RNQS incorporates minimum voting in every iteration, which not only outperforms hybrid NQS in terms of accuracy but also greatly accelerates the search process. The empirical advantage of minimum voting over majority voting is demonstrated through an experiment in Section 4.1.

## 4. Experiments

In this section, we conduct numerical experiments to validate some key ideas in this paper. We first compare the performances of minimum voting and majority voting in a quantum search mechanism. Second, we compare the empirical performance of NQS and RNQS in terms of their accuracy, computational cost, and robustness.

#### 4.1. Minimum Voting versus Majority Voting

To obtain a clear comparison between minimum voting and majority voting in the quantum search regime, we evaluate the performance of hybrid NQS (i.e., NQS combined with majority voting or minimum voting) and NQS with minimum voting. For both methods, we set the number of quantum nodes to be  $k = 3, 5$ , and  $7$ . The dataset is generated as a random permutation of the integers from  $0$  to  $31$ . Thus, we know that  $0$  is the minimum among the  $32$  numbers, but its location within the set is unknown. For both methods, we run  $1000$  replications. We calculate the probability of selecting  $0$  and the corresponding  $95\%$  confidence interval after each iteration for both methods. Furthermore, we use  $10,000$  replications of NQS to approximate the baseline: the probability distribution of  $32$  states after each iteration without applying either majority or minimum voting. The results are reported in Figure 2.



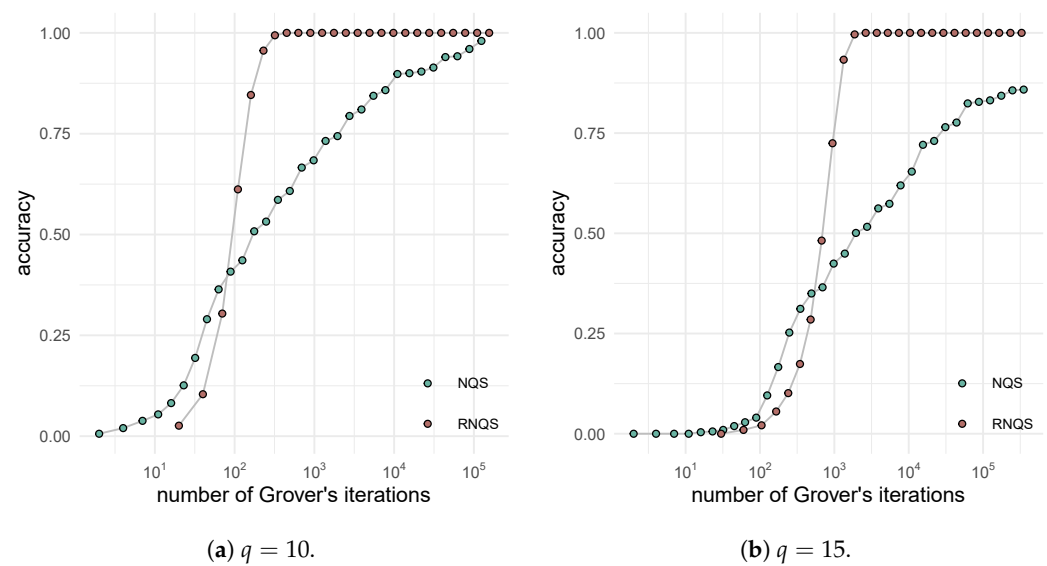
**Figure 2.** Probability of correct selection in majority/minimum voting versus the probability of correct selection in each measure for quantum search with 5 qubits. Three patterns are plotted for different  $k$  values. Over  $1000$  replications, the calculated probabilities are illustrated as dots, while their associated  $95\%$  confidence intervals are depicted using vertical error bars. The unbroken lines present the smoothed trajectories. The dashed lines on the diagonal are references for probability of correct selection in each measure.

The results demonstrate that both majority voting and minimum voting can enhance the success probability compared to the baseline. Furthermore, minimum voting consistently outperforms majority voting, which aligns with our discussion in Section 3.2. The advantage of minimum voting over majority voting is more pronounced when the success probability of a single node (i.e., the baseline) is low. This finding validates our conjecture that minimum voting is a superior strategy, as it not only improves accuracy but also strengthens robustness for quantum search.

#### 4.2. RNQS versus NQS: Accuracy and Computational Cost

In the second experiment, we compared the accuracy and computational cost of RNQS and NQS by applying them to search for the minimum in a random set of size  $D = 2^q$  for  $q = 10$  and  $15$ . The elements in the random set were drawn as a random permutation of the integers  $0, 1, \dots, 2^q - 1$ . For each method, we ran  $500$  replications and reported their accuracy probability versus the computational cost, measured by the total number of Grover's operations used. The results are visualized in Figure 3. The comparison clearly shows that the accuracy probability of RNQS converges quickly to 1 within a few hundred Grover's operations, while the accuracy probability of NQS requires more than  $10^5$  Grover's operations to approach 1. Additionally, we observed that the computational cost of RNQS increases much more slowly with  $q$  compared to NQS. This experiment clearly justifies our conjecture that RNQS is a preferred alternative to NQS, as it significantly increases accuracy and reduces the computational cost.





**Figure 3.** Increment of accuracy in number of Grover's operations for NQS and RNQS.

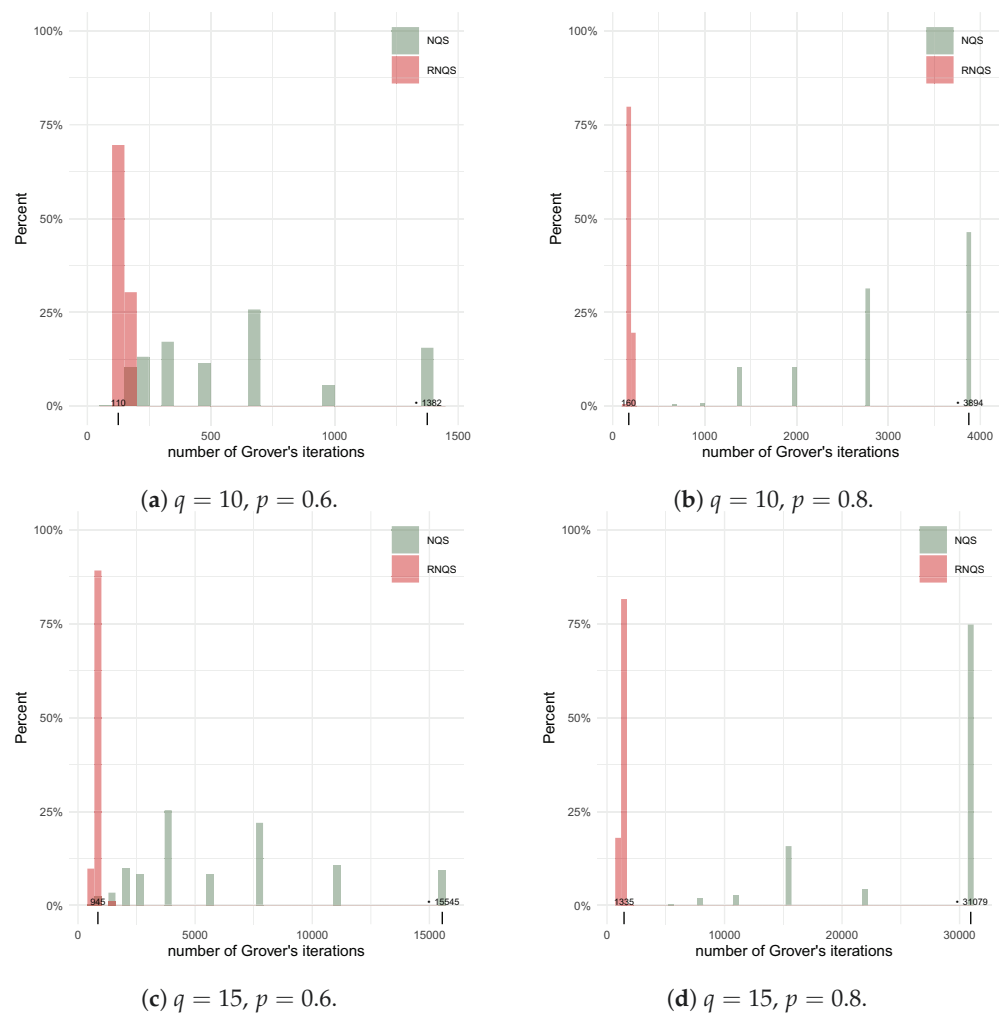
#### 4.3. RNQS versus NQS: Robustness Analysis

The third experiment was designed to compare the robustness of RNQS and NQS. We set  $q = 10, 15$ , and  $20$ . For each  $q$ , we established two desired accuracy probabilities,  $p = 0.6$  and  $p = 0.8$ . We repeated 500 replications for each scenario. In each replication, we recorded the number of Grover's iterations needed by each method to achieve the desired accuracy probabilities. The empirical distribution of the recorded numbers provided us with key information to understand the robustness of RNQS and NQS. To that end, we reported the empirical quantiles of the recorded numbers in Table 1 and depicted histograms in Figure 4. As NQS converges too slowly, we only reported the results of RNQS for  $q = 20$ .

**Table 1.** Simulation results over 500 replications showing quantiles of  $N_G$ .

| Accuracy  | Algorithm | NQS      |        |        |        |               |               |               | RNQS |      |      |      |      |      |      |
|-----------|-----------|----------|--------|--------|--------|---------------|---------------|---------------|------|------|------|------|------|------|------|
|           | Quantile  | 5%       | 10%    | 25%    | 50%    | 75%           | 90%           | 95%           | 5%   | 10%  | 25%  | 50%  | 75%  | 90%  | 95%  |
| $p = 0.6$ | $q = 10$  | 176      | 176    | 349    | 492    | 694           | 1382          | 1382          | 110  | 110  | 110  | 110  | 160  | 160  | 160  |
|           | $q = 15$  | 1382     | 1951   | 3894   | 5503   | 7778          | 10,995        | 15,545        | 675  | 945  | 945  | 945  | 945  | 945  | 945  |
|           | $q = 20$  | -        | -      | -      | -      | -             | -             | -             | 4960 | 4960 | 4960 | 6980 | 6980 | 6980 | 6980 |
| $p = 0.8$ | $q = 10$  | 1382     | 1382   | 2756   | 2756   | 5503          | 7778          | 10,995        | 160  | 160  | 160  | 160  | 160  | 230  | 230  |
|           | $q = 15$  | 15,317.5 | 15,545 | 21,979 | 43,947 | $\geq 62,146$ | $\geq 62,146$ | $\geq 62,146$ | 945  | 945  | 1335 | 1335 | 1335 | 1335 | 1335 |
|           | $q = 20$  | -        | -      | -      | -      | -             | -             | -             | 6980 | 6980 | 6980 | 6980 | 9840 | 9840 | 9840 |

The distributions of NQS are right-skewed and fat-tailed in all four cases. As a result, there is a non-negligible probability that NQS may fail to find the solution even with a huge number of Grover's operations. This indicates that, if the user is not fortunate, NQS cannot successfully find the solution within a reasonable computational budget. This provides empirical evidence of the robustness concerns of NQS, as discussed in Section 2.3. On the other hand, the distributions of RNQS are tightly concentrated towards a very small sample median in all scenarios. Even for  $q = 20$  (i.e.,  $D = 2^{20} = 1,048,576$ ), the 95% quantile of RNQS is less than 10 thousand Grover's operations and only less than 50% larger than its 5% quantile. The histograms in Figure 4 also provide a clear visual comparison between the empirical distributions of RNQS and NQS. The experimental results show strong evidence that RNQS is empirically robust and much more efficient than NQS.



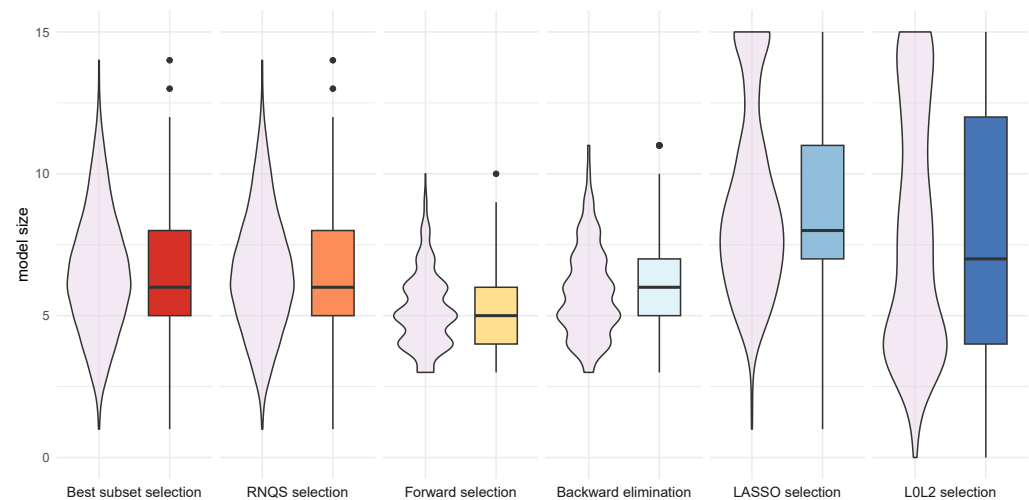
**Figure 4.** Histograms of number of Grover's operations to achieve specific accuracy  $p$ .

#### 4.4. Application to Best Subset Selection for High-Dimension Linear Regression

Over the past few decades, numerous techniques have been proposed to address feature selection problems. Best subset selection, with a rich history dating back to Refs. [16,17], is notable. However, this approach is NP-hard, which led to the advent of Forward selection and Backward elimination techniques as proposed in Ref. [18] to counteract this complexity. While these methods can be effective, they may also be sensitive to factors like sample size and correlation among variables. Such sensitivities can make them unreliable or inconsistent in certain scenarios. The LASSO method, introduced in Ref. [19], emerged as a promising contender for producing sparse solutions. Nevertheless, LASSO has its shortcomings, particularly when managing highly correlated variables. It tends to retain just one variable and sets coefficients of other correlated variables to zero. This can pose challenges in high-dimensional datasets with prevalent correlations, even if the correlations are not pronounced at the population level. More recently, the methods proposed in Ref. [20] harness the power of coordinate descent and local combinatorial optimization for feature selection. As quantum computing continues to make strides, the RNQS algorithm offers a tantalizing opportunity to enhance Best subset selection for such problems.

In this application, we delve into a feature selection problem in the context of a linear additive model. Given a response vector  $Y$ , we aim to select features among  $p = 15$  variables  $X_1, \dots, X_{15}$  within a linear regression framework. With a sample size set at 100, the true model is represented by  $Y = 2 + 5X_3 + 3X_6 + 4X_9 + 6X_{12} + 10\epsilon$ , wherein the covariate vector  $\mathbf{X} = (X_1, \dots, X_{15})$  originate from a multivariate normal distribution with a mean of  $\mathbf{0}$ . The power decay covariance matrix is given by  $\Sigma = (\sigma_{ij})_{15 \times 15}$ , where  $\sigma_{ii} = 1$

and  $\sigma_{ij} = \rho^{|i-j|}$  for  $i \neq j$ . In this scenario,  $\epsilon \sim N(0, 1)$  is independent of  $\mathbf{X}$ , and  $\rho$  is set at 0.8. Figure 5 showcases selected model sizes via box plots and violin plots for six distinct methods: Best subset selection, RNQS selection, Forward selection, Backward elimination, LASSO, and the  $\ell_0\ell_2$  selection as presented in Ref. [20]. Despite each method's intrinsic strengths and limitations, RNQS exemplifies the most faithful replication of Best subset selection. By capitalizing on the computational prowess of the RNQS algorithm, it becomes clear that the domain of  $\ell_0$  optimization in statistics and machine learning stands to benefit immensely.



**Figure 5.** Violin plots and Box plots of the selected model size. Six patterns are plotted for different methods. In each box plot, the bold line represents the median; two hinges represent the 25% and 75% quantile; the length of the whiskers is 1.5 of IQR.

## 5. Conclusions

The practical application of Grover's Algorithm [15] is constrained, in part, by the necessity of oracle information, specifically the function  $S(\cdot)$  in Algorithm 1. While NQS [13] was introduced to address this issue, it struggles to converge for large-scale problems and lacks robustness. In this paper, we propose a novel quantum algorithm that leverages multiple quantum nodes and incorporates the concept of minimum voting to enhance the precision and robustness of quantum search. Three simulation studies underscore the algorithm's superior practical performance compared to its predecessor, NQS. Another simulation investigates the algorithm's suitability for Best subset selection. Future research will explore the application of this algorithm in various additional statistical and machine learning problems involving  $\ell_0$  optimization, including sparse principal component analysis and Best subset selection.

**Author Contributions:** Conceptualization, Y.K.; methodology, T.Z. and Y.K.; software, T.Z.; validation, T.Z. and Y.K.; formal analysis, T.Z. and Y.K.; investigation, T.Z. and Y.K.; resources, Y.K.; data curation, T.Z.; writing—original draft preparation, T.Z.; writing—review and editing, Y.K.; visualization, T.Z.; supervision, Y.K.; project administration, Y.K.; funding acquisition, Y.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is supported by the National Science Foundation grant number NSF-2210468, NSF-2324389, and NSF-2243044, and the National Institute of Health grant number NIH-1R01HL172291.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

### Abbreviations

The following abbreviations are used in this manuscript:

|      |                                    |
|------|------------------------------------|
| NQS  | Non-oracular Quantum Search        |
| RNQS | Robust Non-oracular Quantum Search |

### References

1. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2010.
2. Jin, C.; Jin, S.W. Prediction approach of software fault-proneness based on hybrid artificial neural network and quantum particle swarm optimization. *Appl. Soft Comput.* **2015**, *35*, 717–725. [\[CrossRef\]](#)
3. Zhou, N.R.; Xia, S.H.; Ma, Y.; Zhang, Y. Quantum particle swarm optimization algorithm with the truncated mean stabilization strategy. *Quantum Inf. Process.* **2022**, *21*, 42. [\[CrossRef\]](#)
4. Zhou, N.R.; Zhang, T.F.; Xie, X.W.; Wu, J.Y. Hybrid quantum–classical generative adversarial networks for image generation via learning discrete distribution. *Signal Process. Image Commun.* **2023**, *110*, 116891. [\[CrossRef\]](#)
5. Grover, L.K. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.* **1997**, *79*, 325. [\[CrossRef\]](#)
6. Boyer, M.; Brassard, G.; Høyer, P.; Tapp, A. Tight bounds on quantum searching. *Fortschritte Der Phys. Prog. Phys.* **1998**, *46*, 493–505. [\[CrossRef\]](#)
7. Kwiat, P.; Mitchell, J.; Schwindt, P.; White, A. Grover’s search algorithm: An optical approach. *J. Mod. Opt.* **2000**, *47*, 257–266. [\[CrossRef\]](#)
8. Long, G.L. Grover algorithm with zero theoretical failure rate. *Phys. Rev. A* **2001**, *64*, 022307. [\[CrossRef\]](#)
9. Høyer, P.; Neerbek, J.; Shi, Y. Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica* **2002**, *34*, 429–448.
10. Biham, E.; Biham, O.; Biron, D.; Grassl, M.; Lidar, D.A. Grover’s quantum search algorithm for an arbitrary initial amplitude distribution. *Phys. Rev. A* **1999**, *60*, 2742–2745. [\[CrossRef\]](#)
11. Biham, E.; Kenigsberg, D. Grover’s quantum search algorithm for an arbitrary initial mixed state. *Phys. Rev. A* **2002**, *66*, 062301. [\[CrossRef\]](#)
12. Brassard, G.; Høyer, P.; Mosca, M.; Tapp, A. Quantum Amplitude Amplification and Estimation. *arXiv* **2000**, arXiv:quant-ph/0005055.
13. Chen, J.; Park, C.; Ke, Y. Learning High Dimensional Multi-response Linear Models with Non-oracular Quantum Search. In Proceedings of the 2022 IEEE International Conference on Quantum Computing and Engineering (QCE), Broomfield, CO, USA, 18–23 September 2022; pp. 1–12. [\[CrossRef\]](#)
14. Bennett, C.H.; Bernstein, E.; Brassard, G.; Vazirani, U. Strengths and weaknesses of quantum computing. *SIAM J. Comput.* **1997**, *26*, 1510–1523. [\[CrossRef\]](#)
15. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; ACM: New York, NY, USA, 1996; pp. 212–219.
16. Beale, E.; Kendall, M.; Mann, D. The discarding of variables in multivariate analysis. *Biometrika* **1967**, *54*, 357–366. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Hocking, R.; Leslie, R. Selection of the best subset in regression analysis. *Technometrics* **1967**, *9*, 531–540. [\[CrossRef\]](#)
18. Efroymson, M. Stepwise regression—A backward and forward look. In Proceedings of the Eastern Regional Meetings of the Institute of Mathematical Statistics, Long Island, NY, USA, 27–29 April 1966; pp. 27–29.
19. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B* **1996**, *58*, 267–288. [\[CrossRef\]](#)
20. Hazimeh, H.; Mazumder, R. Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. *Oper. Res.* **2020**, *68*, 1517–1537. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.