Grover Search for Portfolio Selection

A. Ege Yilmaz[1], Stefan Stettler[2], Thomas Ankenbrand[3] and Urs Rhyner[4]

July 26, 2023

***Abstract*—We present explicit oracles designed to be used in Grover's algorithm to match investor preferences. Specifically, the oracles select portfolios with returns and standard deviations exceeding and falling below certain thresholds, respectively. One potential use case for the oracles is selecting portfolios with the best Sharpe ratios. We have implemented these algorithms using quantum simulators.**

I. Introduction

Consider that the efficient frontier of a portfolio universe is calculated and presented to an investor, who would like to choose portfolios from it. The efficient frontier would consist of two lists of numbers - one containing the expected returns and the other containing the corresponding standard deviations of the optimal portfolios. Our implementation employs quantum algorithms to enable selection from optimal portfolios with specified risks and returns.

The goal of portfolio optimization is to find the asset allocations, which result in optimal portfolios with maximum expected returns for given risk levels or minimal risk for a given level of expected return. Portfolios could be derived using the mean-variance method [15], where variances of asset prices are used as a measure for portfolio risk. The objective constructed in this scheme yields a nondominated set, which is called "the efficient frontier". It allows the investors to choose among the optimal (efficient) portfolios based on their preferences. To identify portfolios with preference (utility function) maximizing risk-return pairs, we utilize two quantum algorithms - the Quantum Exponential Search (QES) algorithm [2] and the Grover Adaptive Search (GAS) algorithm [4]. Both algorithms leverage the quantum search algorithm called Grover's algorithm [7] to locate desired items in a list of items. A general application of quantum computing in finance is discussed in [1].

This paper shows how to construct oracles (see Section III-A) to select items in a list, based on their values and applies the resulting algorithms to a financial use case. Here, the items are varying portfolio allocations in a given asset universe with their corresponding risks and returns. In Section II, we name the sources, which are related to our work. Although this study concentrates on the implementation of a conditional search algorithm in the context of investment portfolios, we also outline the literature regarding quantum portfolio optimization. A background on the relevant quantum algorithms are given in Section III. Our hypothesis is presented in Section IV, along with the discussion of the implemented algorithms. Lastly, our experimental results are presented in Section V.

[1]Hochschule Luzern, Institut für Finanzdienstleistungen Zug IFZ, Suurstoffi 1, 6343 Rotkreuz, Email: <ahmetege.yilmaz@hslu.ch>

[2]Abraxas Informatik AG, The Circle 68, 8058 Zürich, Phone: +41 58 660 16 85, Email: <stefan.stettler@abraxas.ch>

[3]Hochschule Luzern, Institut für Finanzdienstleistungen Zug IFZ, Suurstoffi 1, 6343 Rotkreuz, Phone: +41 41 757 67 23, Email: <thomas.ankenbrand@hslu.ch>

[4]Inventx AG, The Circle 37, 8058 Zürich, Phone: +41 81 287 19 79, Email: <urs.rhyner@inventx.ch>

II. Literature Review

Grover's algorithm is based on amplitude amplification [3, 8], where the probability of measuring the desired quantum states are amplified. Using this, QES returns one of the desired states after an expected number of $\mathcal{O}(\sqrt{N/M})$ calls to the oracle, where M and N are the number of desired states and the total number of possible states, respectively. Another example of Grover-based algorithms is the GAS algorithms. These are optimization algorithms, which utilize Grover search in an iterative scheme to find the optimum value of an objective function. The relevant GAS algorithm that we use in our work is [4]. Including QES as a subroutine, GAS finds the minimum among the items in an unsorted table of size N, also after $\mathcal{O}(\sqrt{N})$ calls to the oracle. An extension of GAS to the k-minima problem is investigated in [16]. The algorithms QES and GAS allow us to select portfolios from a list of optimal portfolios. We would like to stress that this approach differs from the following literature, where the calculation of efficient portfolios are studied.

An instance of GAS in the context of portfolio optimization is [5], where oracles for Constrained Polynomial Binary Optimization (CPBO) problems using GAS are constructed and tested on IBM's gate-based hardware. A subclass of CPBO are Quadratic Unconstrained Binary Optimization (QUBO) problems, which the portfolio optimization problem falls into. Since QUBO is the natural input of quantum annealers, a significant portion of quantum portfolio optimization implementations use annealers [18, 6, 14, 19, 22].

In [13] a quantum version of the portfolio optimization algorithm by [9] is given. Their quantum algorithm achieves a quadratic speed-up in the time complexity with respect to the number of assets in the portfolio due to quantum state preparation and norm estimation with the assumptions of no short selling and quantum query access to asset returns. The latter describes having access to the desired state vectors by means of a certain quantum operation. Experimental results with 14 assets from SP500 using Honeywell's trapped-ion System Model H1 are shown in [23]. Their algorithm is based on the hybrid HHL algorithm [12], where the phase estimation is enhanced by mid-circuit measurement, quantum conditional logic, and qubit reset and reuse. In a follow-up paper, the authors carry out the constrained portfolio optimization on the same hardware, where they utilize Quantum Zeno Effect in Quantum Approximate Optimization Algorithm and Layer Variational Quantum Eigensolver. By means of repeated projective measurements leading to Zeno dynamics, they achieve

a higher in-constraint probability compared to the state-of-the-art technique of enforcing constraints by introducing a penalty into the objective. Details on some of the theoretical results regarding quantum portfolio optimization [11, 21] can be found in [1].

III. Preliminaries

In this section, the relevant concepts are introduced and the notation is set.

A. Grover's search algorithm

Given an N item unstructured search problem with M solutions, Grover's algorithm finds the solutions with $\mathcal{O}(\sqrt{N})$ calls to the oracle [17], where $M \in \mathbb{N}$ and for the sake of convenience, we set $N = 2^n$ for an $n \in \mathbb{N}^+$. We also assume that $M \leq N/2$, since this can always be ensured by introducing an additional qubit to the system that doubles the size of the search space. At each Grover iteration, the initial state is rotated by an angle θ towards the solution state $|\beta\rangle$ by means of the Grover operator

$$G = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{1}$$

on the plane spanned by the basis states $|\beta\rangle$ and $|\alpha\rangle \coloneqq |\beta\rangle^{\perp}$. Here, the angle between $|\psi\rangle \coloneqq |+\rangle^n$ and $|\alpha\rangle$ is given by

$$\frac{\theta}{2} = \arcsin\sqrt{\frac{M}{N}}. \tag{2}$$

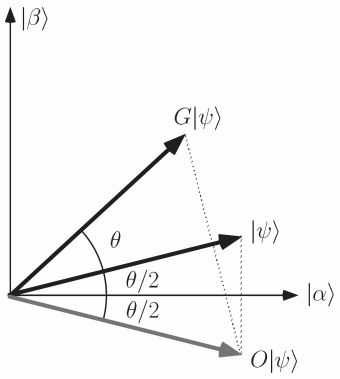


Figure 1: Geometrical illustration of the first Grover iteration. Source of image: [17].

The quantum search algorithm is formulated in terms of an *oracle*. An oracle is a 'black-box', which is able to 'recognize' the solutions of a given problem. Upon an input, the oracle f outputs 1 if the input is a solution to the problem we want to solve and 0 otherwise. In quantum computing, an oracle is implemented as a unitary operator O acting on the computational basis as

$$|x\rangle\,|q\rangle \xrightarrow{O} |x\rangle\,|q \oplus f(x)\rangle\,, \tag{3}$$

where $\oplus$ denotes addition modulo 2, $|q\rangle$ is the oracle qubit constituting the decision of the oracle and the input is registered in $|x\rangle$. If $|q\rangle$ is prepared in the $|-\rangle$ state, (3) becomes

$$|x\rangle\,|-\rangle \xrightarrow{O} (-1)^{f(x)}\,|x\rangle\,|-\rangle\,,$$

where the solutions are marked by a phase kickback. This corresponds to reflecting the initial state about $|\alpha\rangle$. The complete Grover rotation (1) can be achieved by applying a second reflection about $|\psi\rangle$ (see Figure 1). The number of Grover rotations required to reach β within an angle $\theta/2 \leq \pi/4$ is given by

$$\mathrm{CI}\left(\frac{\pi/2}{\theta} - \frac{1}{2}\right),$$

where $\mathrm{CI}(x)$ denotes the integer closest to the real number x. By convention, halves are rounded down. Measurement on the state then yields a solution to the search problem with probability at least one-half.

B. Quantum phase estimation

If the Grover search involves non-integer values, the decimal parts need to be encoded in the quantum circuits. One way to achieve this is to use quantum phase estimation. It estimates the strength φ of a quantum phase $e^{i2\pi\varphi}$ in units of 2π to a specified accuracy, where $\varphi \in [0, 1)$ and $e^{i2\pi\varphi}$ is an eigenvalue of a unitary operator U. In order to have an accuracy of 2^{-m} and a success probability of $1 - \epsilon$, the number of phase estimating qubits must be

$$t = m + \log(2 + 1/2\epsilon).$$

The estimated phase strength $\widetilde{\varphi}$ is then represented as an integer b in t qubits following the relation

$$b = 2^t\widetilde{\varphi}. \tag{4}$$

The runtime of quantum phase estimation is $\mathcal{O}(t^2)$ [17].

C. Quantum counting

Grover's algorithm serves to find the solutions of a search problem with fewer consultations to the oracle. Classically, we would need $\mathcal{O}(N)$ consultations to estimate M. Utilizing Grover operators and quantum phase estimation, quantum counting can speed this process up. Moreover, it can be used to estimate whether there is a solution to the problem or not.

The number of solutions M and the Grover angle θ are related by (2). Hence, choosing the unitary operator U mentioned in Section III-B as the Grover operator (1) with the eigenvalues $e^{\pm i\theta}$, quantum phase estimation yields an estimate of M. The error related to the estimation of M is given by [17]

$$\Delta M < 2^{-m}\left(\sqrt{NM} + \frac{N}{4}\cdot 2^{-m}\right). \tag{5}$$

D. Quantum Exponential Search

QES is a generalization of Grover's algorithm, allowing us to use Grover's algorithm without prior knowledge of the number of solutions. After an expected number of $\mathcal{O}(\sqrt{N/M})$ calls to the oracle, a solution is returned almost certainly. Basically, it employs Grover's algorithm in a loop, where each time the upper bound of the range is increased, which the number of Grover iterations is uniformly randomly chosen from. The program is exited upon measuring a solution and the case of no solution is handled by an appropriate time-out. We state QES in Algorithm 1 for the reader's convenience.

Algorithm 1 QES

repeat
  $m \leftarrow 1$
  $\lambda \leftarrow 8/7$
  Choose integer j uniformly at random with $0 \leq j < m$
  Apply j Grover iterations
  Measure i
  if $f(i) = 1$ **then**
    Exit
  else
    $m \leftarrow \min(\lambda m, \sqrt{N})$
  end if
until Program is exited

E. Grover Adaptive Search

As mentioned in Section II, GAS is a Grover-based algorithm to find the minimum value in a list by iteratively applying Grover search and using the best-known value as a threshold to flag all values smaller than the threshold. It uses QES as a subroutine and returns the index of the minimum with a success probability of at least one-half after making $\mathcal{O}(\sqrt{N})$ calls to the oracle. We state GAS in Algorithm 2 for the reader's convenience.

Algorithm 2 GAS

Choose integer j uniformly at random with $0 \leq j < N$
repeat
  Initialize memory as $\sum_i \frac{1}{\sqrt{N}} |i\rangle |j\rangle$
  Mark items whose values are smaller than the j^{th} value
  Apply QES with outcome j'
  $j \leftarrow j'$
until Number of calls to the oracle has exceeded $22.5\sqrt{N} + 1.4 \log^2(N)$

IV. Implementation

The research question initially aimed to investigate the feasibility of implementing portfolio optimization with continuous asset allocations and positivity constraints on gate-based quantum hardware or simulators. However, to the best of our knowledge, the only available literature on this topic proposes an algorithm that primarily focuses on theoretical runtime rather than practical implementation [11]. After an analysis that demonstrates the infeasibility of implementing the complete portfolio optimization problem with these criteria on existing quantum hardware or simulators, the research objective has been redefined to focus on selecting portfolios from the set of efficient portfolios that maximize investor preferences. In this context, the hypothesis posits that by utilizing Grover-based algorithms with suitable oracles, it is possible to identify portfolios that maximize investor preferences, assuming the risk-return pairs of the optimal portfolios are available as input. In the following, we assume that the investor preferences are maximized by selecting portfolios with returns and standard deviations higher and lower than desired threshold values, respectively.

In this section, the gates and circuits required to select the preferred portfolios are presented. Once the efficient portfolios are generated by classical mean-variance optimization with no-shortselling condition, we select portfolios with returns and standard deviations higher and lower than desired threshold values, respectively. This is a conditional slicing problem with two lists and conditions. It is solved by employing QES with the oracles given in Section IV-A2. The second part of our implementation deals with searching for the portfolio with the maximum Sharpe ratio, which is described in Section IV-B. Lastly, the scaling of the number of required qubits is analyzed in Section IV-C.

A. Conditional Slicing

We explore the quantum version of selecting list items by a given condition, also known as (list) slicing by condition. Specifically, we are interested in conditions involving comparisons between non-negative values.

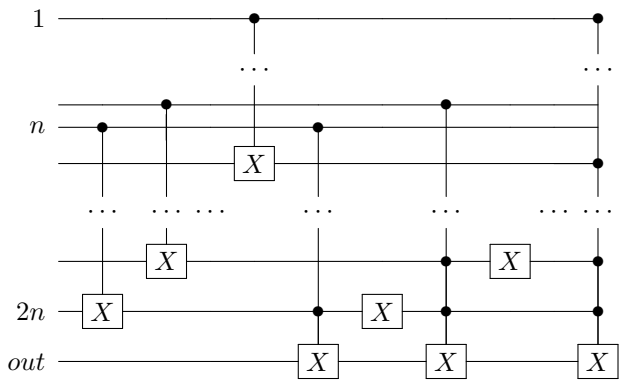
1) GT-gate

In order to carry out the comparisons between values, a greater-than operation (GT-gate) is necessary. Our implementation of a GT-gate for the comparison $a > b$ is given by

$$\begin{aligned} &CX(1, n+1) \cdots CX(n, 2n) \\ &\quad MCX(n, 2n, 2n+1) X(2n) \\ &\quad MCX(n-1, 2n-1, 2n, 2n+1) X(2n-1) \cdots \\ &\quad MCX(1, n+1, \ldots, 2n, 2n+1) X(n+1) \\ &\quad X(n+1) \cdots X(2n) \\ &\quad CX(1, n+1) \cdots CX(n, 2n) \, |0\rangle^{2n+1}, \end{aligned} \tag{6}$$

where $a, b \in \mathbb{N}$. The last qubit carries the outcome of the comparison. The non-negative integers a and b are encoded bitwise in the first n qubits and the remaining qubits, respectively. Note that the leading order bits have higher indices, i.e. they are located at lower positions in the circuit shown below.

The last two lines of (6) are for uncomputation.



Circuit 1: GT-gate without uncomputation

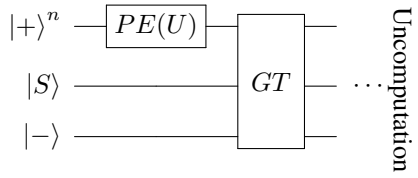
The circuit consists of NOT (X), controlled-NOT (CX) and multi-controlled-NOT (MCX) gates. The number of all gates scales as $\mathcal{O}(n)$.

2) Constructing Oracles

The quantum list slicing algorithm is based on Grover search. For a list with items s_k and a threshold value S, the oracle must check each item for $s_k > S$, $k \in \{1, \ldots, N\}$. In the case of non-integer values, the GT-gate must be applied to the integer part and the decimal part, separately. Since the comparison of integers with GT-gate is straightforward, we first concentrate on the decimal parts comparison. Assuming that we have quantum access to the decimal parts in the form of

$$U = diag(e^{i2\pi s_1}, \ldots, e^{i2\pi s_N}), \tag{7}$$

they can be phase-encoded with the associated unitary operator U, resulting in integer representations of their t-bit approximations in the register, given by (4). This allows us to apply GT-gate on them. In a similar manner, GT-gate can be applied to the integer parts by having quantum access to their phase representations in the form of (7). For simplicity, we assume that all values are in the range $[0, 1)$. A diagram of the initial state and the oracle is shown below:



Circuit 2: Initial state and oracle. Single list.

PE stands for phase estimation. The initial state (Grover eigenstate) is $\psi_0 = |+\rangle^n |S\rangle |-\rangle$. This way, the first register carries the superposition of the list items. The second register has S. The comparison $s_k > S$ is realized by the GT-gate. Finally, the decision of the oracle is registered at the third register.

Slicing by multiple conditions, e.g.

$$S_2 > s_k > S_1$$

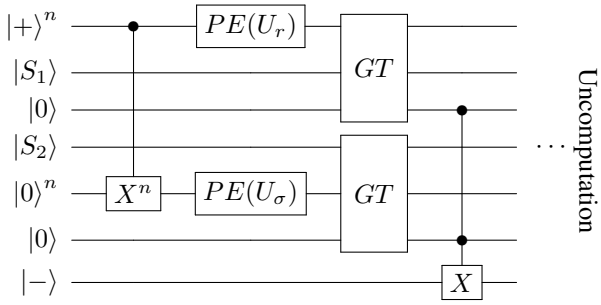
or

$$S_2 > s_k \text{ AND } s_k = S_1,$$

can be done in an analogous way, where S_1 and S_2 are threshold values. An interesting case is when multiple conditions are based on different lists, such as

$$S_2 > \sigma_k \text{ AND } r_k > S_1.$$

A diagram of the oracle for such a slicing is shown below:



Circuit 3: Initial state and oracle for two lists.

The unitary operators U_r and U_σ contain the list items r_k and σ_k in the previously given form (7), respectively.

A point to elaborate on is the resolving power of the GT-gate, when comparing phase-encoded values. The resolving power quantifies how close two values can be, while the GT-gate can successfully compare them. If two distinct values $\varphi_1, \varphi_2 \in [0, 1)$ are represented by integers $b_1, b_2 \in \mathbb{N}$ according to (4), we would like to satisfy the separation condition $|b_1 - b_2| \geq 1$, which implies

$$t \geq \log |\varphi_1 - \varphi_2|^{-1}. \tag{8}$$

For a desired resolving power d, (8) implies that we can choose

$$t = \left\lceil \log d^{-1} \right\rceil. \tag{9}$$

3) The Algorithm

The quantum conditional slicing is realized by inserting the initial states and the oracles presented in Section IV-A2 into QES. One of the solutions is returned with equal probability after an expected number of $\mathcal{O}(\sqrt{N/M})$ calls to the oracle. In case there are multiple solutions, one needs to run the algorithm multiple times to obtain them. By Theorem A.2, it can be checked whether there is no solution, a single solution or multiple solutions with quantum counting after $\mathcal{O}(\sqrt{N})$ calls to the oracle by choosing

$$m = \left\lceil \frac{\log N}{2} + 1.58 \right\rceil \tag{10}$$

and rounding the measured number of solutions to the closest integer. According to Theorem A.1, the exact number of solutions can be obtained after $\mathcal{O}(N)$ calls to the oracle by choosing

$$m = \left\lceil \log N + \frac{1}{2} \right\rceil. \tag{11}$$

Then, the regular Grover's algorithm with the correct number of Grover iterations can also be used instead of QES, since the number of solutions is known.

B. Finding the maximum

We can use the oracle in Circuit 2 from Section IV-A2 in GAS to get the maximum value in a list. At the first iteration, the threshold value in the initial state is prepared as the list value with the index that is chosen uniformly at random. At the later iterations, it is substituted by the outcome of QES. Implementation of a minimum function is analogous. After running GAS for c times, the probability of success is at least $1 - 1/2^c$.

C. Total number of qubits

The number of qubits required to encode the list elements scales logarithmically with the number of elements in the list. Additionally, the choices (10) and (11) for m show that the number of counting qubits scales as $\mathcal{O}(\log N)$. From (9) we see that the number of phase estimating qubits is independent of N, but dependent on the resolving power d. Hence, the total number of qubits required by both algorithms scales as $\mathcal{O}\left(\log(Nd^{-1})\right)$.

V. Results

The circuits that are described in Section IV are implemented using the simulators from IBM's Qiskit SDK [20]. Our results are accessible at [10], where the corresponding repository includes other gates, such as Less Than, Equals and OR gate, which enable different selection conditions.

The first part of our implementation shows that the desired portfolios with specified risks and returns are found by employing our oracles in QES. Note that we take here a subsample of the optimal portfolios resulting in eight values for returns and standard deviations each. This way we reduce the number of qubits required to hold the values, which is too high in terms of the required memory, otherwise. In the second part of our implementation, Sharpe ratios of the portfolios are calculated, where a risk-free rate of 0% is chosen for simplicity. Then, the index of the portfolio with the highest Sharpe ratio is successfully obtained by using Circuit 2 in GAS. Note that all values in both experiments are in $[0, 1)$ and we do not need to compare integer parts. A resolving power $d = 0.01$ is chosen for both experiments, resulting in fifteen qubits for GT-gate. Three qubits are required for encoding the indices of eight values. Hence, Circuit 2 and 3 have eighteen and thirty-seven qubits, respectively. For the case $M = 0$, a time-out of $\mathcal{O}(\sqrt{N})$ is set in QES. Quantum counting with (10) can be used as a substitute to the handling of the case $M = 0$ with a time-out. This could be implemented as a termination condition in GAS.

VI. Conclusion

The described financial use case for quantum computing regarding portfolio selection by investor preferences has been successfully implemented by employing the Grover-based algorithms QES and GAS with the oracles, whose explicit forms are given in this paper. To advance this research, one can implement the proposed method on physical quantum hardware to assess its performance in the presence of noise. This approach would also enable the inclusion of larger portfolios by leveraging devices with a higher number of qubits than those currently available in simulators. Nevertheless, the small-scale examples in [10] are conceptually viable.

Appendix

A. Theorems

Theorem A.1. *The error ΔM of the quantum counting algorithm is less than $1/2$, if the list size N is large, the number of solutions M is at most $N/2$ and the number of bit accuracy qubits is chosen as*

$$m = \left\lceil \log N + \frac{1}{2} \right\rceil.$$

Proof. Suppose we want to bound ΔM by ε, i.e. $\Delta M < \varepsilon$. Then, (5) implies

$$2^{-m}\sqrt{MN} + \frac{N}{4}2^{-2m} \leq \varepsilon. \tag{12}$$

After defining

$$f(m, N) \coloneqq 2^{-m}\sqrt{N}, \tag{13}$$

from (12) we get

$$\begin{aligned} & 2^{-m}\sqrt{MN} + \frac{N}{4}2^{-2m} \leq \varepsilon \\ \Leftrightarrow \quad & 4\sqrt{M}f + f^2 \leq 4\varepsilon \\ \Leftrightarrow \quad & (f + 2\sqrt{M})^2 \leq 4(\varepsilon + M) \\ \Rightarrow \quad & f + 2\sqrt{M} \leq 2\sqrt{\varepsilon + M} \\ \Rightarrow & 2^{m+1} \geq \sqrt{N}\frac{\sqrt{M+\varepsilon} + \sqrt{M}}{\varepsilon} \end{aligned} \tag{14}$$

where in the last line we have used

$$2\sqrt{\varepsilon + M} - 2\sqrt{M} = \frac{2\varepsilon}{\sqrt{M+\varepsilon} + \sqrt{M}}$$

and plugged (13) back in. Plugging $\varepsilon = 1/2$ into (14) gives

$$\begin{aligned} & 2^{m+1} \geq \sqrt{2N}(\sqrt{2M+1} + \sqrt{2M}) \\ \Rightarrow \quad & 2^{m+1} \geq \sqrt{2N}(\sqrt{N+1} + \sqrt{N}) \\ \Leftrightarrow \quad & 2^{m+1} \geq \frac{\sqrt{2}}{\sqrt{1 + 1/N} - 1} \\ \Rightarrow \; & m \geq -\frac{1}{2} - \log\Big(\sqrt{1 + 1/N} - 1\Big), \end{aligned} \tag{15}$$

where in the second line we have used $M \leq N/2$. Assuming $N \gg 1$, we have

$$\log\Big(\sqrt{1 + 1/N} - 1\Big) \approx \log(1/2N). \tag{16}$$

Combining (15) with (16) gives

$$m \geq \log N + \frac{1}{2},$$

which is the desired result. □

Theorem A.2. *Quantum counting algorithm checks whether there is no solution, a single solution or multiple solutions, if the number of bit accuracy qubits is chosen as*

$$m = \left\lceil \frac{\log N}{2} + 1.58 \right\rceil.$$

Proof. Choosing

$$\varepsilon = M - 3/2$$

for all $M \in \{2, 3, \ldots, N/2\}$ and plugging ε into (14) gives

$$2^{m+1} \geq \sqrt{N} \frac{\sqrt{2M - 3/2} + \sqrt{M}}{M - 3/2}.$$

The right hand side is maximized at $M = 2$ yielding

$$2^{m+1} \geq \sqrt{N} \frac{\sqrt{5/2} + \sqrt{2}}{1/2}$$

$$\Rightarrow m \geq -\frac{1}{2} + \frac{1}{2} \log N + \log\Big(2 + \sqrt{5}\Big).$$

Since $\log\big(2 + \sqrt{5}\big) - 1/2 \approx 1.58$, we can choose

$$m = \left\lceil \frac{\log N}{2} + 1.58 \right\rceil. \tag{17}$$

For the cases $M \in \{0, 1\}$, we need to choose $\varepsilon = 1/2$, but those cases are already covered by (17). □

References

[1] Franco D. Albareti et al. *A Structured Survey of Quantum Computing for the Financial Industry*. 2022. DOI: 10.48550/ARXIV.2204.10026. URL: https://arxiv.org/abs/2204.10026.

[2] Michel Boyer et al. "Tight bounds on quantum searching". In: *Fortschritte der Physik: Progress of Physics* 46.4-5 (1998), pp. 493–505.

[3] G. Brassard and P. Hoyer. "An exact quantum polynomial-time algorithm for Simon's problem". In: *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*. IEEE Comput. Soc, 1997. DOI: 10.1109/istcs.1997.595153. URL: https://doi.org/10.1109%2Fistcs.1997.595153.

[4] Christoph Durr and Peter Hoyer. "A quantum algorithm for finding the minimum". In: *arXiv preprint quant-ph/9607014* (1996).

[5] Austin Gilliam, Stefan Woerner, and Constantin Gonciulea. "Grover Adaptive Search for Constrained Polynomial Binary Optimization". In: *Quantum* 5 (Apr. 2021), p. 428. DOI: 10.22331/q-2021-04-08-428. URL: https://doi.org/10.22331%2Fq-2021-04-08-428.

[6] Cláudio Gomes et al. "An Empirical Study on the Use of Quantum Computing for Financial Portfolio Optimization". In: *SN Computer Science* 3.5 (2022), p. 335.

[7] Lov K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 212–219. ISBN: 0897917855. DOI: 10.1145/237814.237866. URL: https://doi.org/10.1145/237814.237866.

[8] Lov K. Grover. "Quantum Computers Can Search Rapidly by Using Almost Any Transformation". In: *Physical Review Letters* 80.19 (May 1998), pp. 4329–4332. DOI: 10.1103/physrevlett.80.4329. URL: https://doi.org/10.1103%2Fphysrevlett.80.4329.

[9] David P Helmbold et al. "On-line portfolio selection using multiplicative updates". In: *Mathematical Finance* 8.4 (1998), pp. 325–347.

[10] HSLU IFZ Competence Center Investments. *Grover Search for Portfolio Selection*. available at: https://github.com/HSLU-IFZ-Competence-Center-Investments/Quantum-Portfolio-Selection/blob/main/QAIF_PortfOpt.ipynb (Jul. 2023). 2023.

[11] Iordanis Kerenidis, Anupam Prakash, and Dániel Szilágyi. *Quantum Algorithms for Portfolio Optimization*. 2019. DOI: 10.48550/ARXIV.1908.08040. URL: https://arxiv.org/abs/1908.08040.

[12] Yonghae Lee, Jaewoo Joo, and Soojoon Lee. "Hybrid quantum linear equation algorithm and its experimental test on IBM Quantum Experience". In: *Scientific reports* 9.1 (2019), p. 4778.

[13] Debbie Lim and Patrick Rebentrost. "A Quantum Online Portfolio Optimization Algorithm". In: *arXiv preprint arXiv:2208.14749* (2022).

[14] Marcos Lopez de Prado. "Multi-Period Integer Portfolio Optimization Using a Quantum Annealer". In: *Available at SSRN 2670033* (2015).

[15] Harry Markowitz. "PORTFOLIO SELECTION*". In: *The Journal of Finance* 7.1 (1952), pp. 77–91. DOI: https://doi.org/10.1111/j.1540-6261.1952.tb01525.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1952.tb01525.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1952.tb01525.x.

[16] Kohei Miyamoto, Masakazu Iwamura, and Koichi Kise. *A Quantum Algorithm for Finding k-Minima*. 2019. DOI: 10.48550/ARXIV.1907.03315. URL: https://arxiv.org/abs/1907.03315.

[17] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000. ISBN: 521635039.

[18] Samuel Palmer et al. "Quantum portfolio optimization with investment bands and target volatility". In: *arXiv preprint arXiv:2106.06735* (2021).

[19] Frank Phillipson and Harshil Singh Bhatia. *Portfolio Optimisation Using the D-Wave Quantum Annealer*. 2020. DOI: 10.48550/ARXIV.2012.01121. URL: https://arxiv.org/abs/2012.01121.

[20] Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing*. 2023. DOI: 10.5281/zenodo.2573505.

[21] Patrick Rebentrost and Seth Lloyd. *Quantum computational finance: quantum algorithm for portfolio optimization*. 2018. DOI: 10.48550/ARXIV.1811.03975. URL: https://arxiv.org/abs/1811.03975.

[22] Davide Venturelli and Alexei Kondratyev. "Reverse quantum annealing approach to portfolio optimization problems". In: *Quantum Machine Intelligence* 1.1-2 (Apr. 2019), pp. 17–30. DOI: 10.1007/s42484-019-00001-w. URL: https://doi.org/10.1007%2Fs42484-019-00001-w.

[23] Romina Yalovetzky et al. "NISQ-HHL: Portfolio optimization for near-term quantum hardware". In: *arXiv preprint arXiv:2110.15958* (2021).