Quantum algorithm for large-scale market equilibrium computation

Po-Wei Huang[1] and Patrick Rebentrost[1, 2]

[1]Centre for Quantum Technologies, National University of Singapore, Singapore 117543
[2]Department of Computer Science, National University of Singapore, Singapore 117417
huangpowei22@u.nus.edu, patrick@comp.nus.edu.sg

Abstract

Classical algorithms for market equilibrium computation such as proportional response dynamics face scalability issues with Internet-based applications such as auctions, recommender systems, and fair division, despite having an almost linear runtime in terms of the product of buyers and goods. In this work, we provide the first quantum algorithm for market equilibrium computation with sub-linear performance. Our algorithm provides a polynomial runtime speedup in terms of the product of the number of buyers and goods while reaching the same optimization objective value as the classical algorithm. Numerical simulations of a system with 16384 buyers and goods support our theoretical results that our quantum algorithm provides a significant speedup.

1 Introduction

The balance of supply and demand is a fundamental and well-known law that determines the price of goods in a market. In a market with a set of n buyers and m goods, the *competitive equilibrium* [1, 2] determines the optimal price and allocation of goods such that the supply equals the demand in the given market. The computation of the competitive equilibrium is known as the *market equilibrium computation problem*, whose unique solution was shown to exist under a general model of the economics in the seminal work of Arrow and Debreu [3]. The relevance of such problems in algorithmic game theory [4, 5] is substantiated by the first welfare theorem, which implies that the competitive equilibria are *Pareto-efficient* [6], where no allocation is available that makes one agent better without making another one worse. In *competitive equilibrium from equal income* (CEEI) scenarios, such equilibria are further known to by *envy-free* [7, 8], where no agent would prefer an allocation received by another agent over their own.

The market equilibrium computation problem has, in recent years, been extended to various large-scale Internet-based markets [9], including auction markets [10], fair item allocation/fair division [11–13], scheduling problems [14] and recommender systems [15]. Such developments call for the need to further develop algorithmic theories for markets and the computation of market equilibria. We focus on a particular type of market known as the Fisher market [16, 17], where there is a set of n buyers interested in buying m infinitely-divisible goods, and where each buyer has their monetary budget that has no intrinsic value apart from being used to purchase goods. We mainly consider Fisher markets with linear utilities, where the total utility gained by purchasing goods is strictly linear to the value and proportion of the goods obtained.

For combinatorial formulations of the market equilibrium problem, algorithms have been discovered that can obtain exact and approximate solutions [18–21], these algorithms scale poorly against the growing number of buyers and goods. One can also formulate the market equilibrium computation problem as an optimization problem that maximizes a convex objective function known as the Eisenberg-Gale (EG) convex program [22, 23]. For such optimization problems, approximate

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Algorithm	Iterations	Runtime	Memory	Result Prep.
PR dynamics [24]	$\dfrac{\log m}{\varepsilon}$	$\tilde{\mathcal{O}}\left(\dfrac{mn}{\varepsilon}\right)$	$\mathcal{O}(mn)$	N/A, in RAM
Our work	$\dfrac{2\log m}{\varepsilon}$	$\tilde{\mathcal{O}}\left(\dfrac{\sqrt{mn\max(m,n)}}{\varepsilon^2}\right)$	$\mathcal{O}(m+n)^*$	QA: $\mathcal{O}(\text{poly}\log\frac{mn}{\varepsilon})$ SA: $\tilde{\mathcal{O}}(\sqrt{mn})$

Table 1: *Main results.* In this work, n is the number of buyers, m is the number of goods, and ε indicates the additive error of the computed values to the minimally-achievable EG objective value. The memory complexity for the quantum algorithm (annotated with *) refers to the use of quantum query access to classical memory, achievable by QRAM (see Definition 3), instead of classical RAM. As the computed competitive equilibrium consumes $\mathcal{O}(mn)$ memory, our quantum algorithm does not provide the entire bid matrix, but instead provides quantum query access (QA) and sample access (SA) to the competitive equilibrium. The result preparation column refers to the additional runtime cost of preparing QA and SA.

solutions can be found much faster. One such algorithm of the market equilibrium problem is the *proportional response* (PR) dynamics [24, 25]. The PR dynamics is an iterative algorithm that converges with a rate of $\frac{1}{T}$ where T is the number of iterations. Each iteration has a cost of $\mathcal{O}(mn)$ from proportionally updating individual bids of each buyer for different goods.

Given the high number of buyers and goods that can exist in Internet-based markets, the problem of further algorithmic speedups to the computation continues to be an active field of research. Gao and Kroer [26] discovered that by using projected gradient descent instead of PR dynamics, the market equilibrium can be found with linear convergence. Other attempts that aim to reduce the cost per iteration, such as using clustering to reduce the problem size [15], have also been made. However, it is unclear whether advantages beyond a constant-factor speedup can be provided.

While it is unclear whether additional classical strategies can provide further algorithmic speedup to the market equilibrium computation problem, one can utilize algorithmic developments in quantum computation to achieve such goals. Quantum computation [27] is an emerging technology that has been utilized for algorithmic speedups in various optimization problems [28] such as linear programming [29] and semidefinite programming [30–33].

In this work, we consider a Fisher market with n buyers and m goods, where the objective is to find an approximate market equilibrium whose EG objective function is within an additive error ε of the optimal EG objective value. We provide a method to reduce the cost per iteration by utilizing quantum norm estimation and quantum inner product estimation [34, 35] and provide the first quantum algorithm to achieve sublinear performance in terms of the product of the buyers and goods in market equilibrium computation. To arrive at the quantum algorithm, we show an alternate version of the PR dynamics with erroneous updates, which we term the faulty proportional response (FPR) dynamics. We then provide a quantum algorithm that provides a quadratic speedup in terms of the smaller dimension between buyers and goods, as well as less memory consumption, albeit being based on QRAM instead of classical RAM. We summarize our results in Table 1.

Unlike the classical PR algorithm, which provides an entire matrix corresponding to the competitive equilibrium with storage $\mathcal{O}(mn)$ space,[1] we provide quantum query and sample access to the values of the competitive equilibrium, which allows access to the individual values by index querying as well as ℓ_1 sample access to the values. This access format has previously been used as the output model for quantum recommendation systems [36], quantum linear system solvers [37], but has been pointed out to have significant caveats [38]. While the quantum algorithm does not exactly "solve" the market equilibrium problem in the sense of outputting the entire matrix, quantum query and sample access provide a preparation of a quantum state encoding the solution that can be used for further computation, for extracting only a small set of values within the matrix, or for extracting certain properties of the matrix, which may indeed be the use case for large-scale distributed systems.

---

[1] Given that it takes $\mathcal{O}(mn)$ to even output the entire matrix, a sublinear algorithm would require a different output model in the first place, since the sublinearity would be undone by the solution generation.

2

2 Preliminaries

Notations. Let $[n] := \{0, 1, \ldots, n-1\}$. We use $\odot$ to represent element-wise multiplication, as well as $\oplus$ for bit-wise XOR operation and $\otimes$ for tensor products. For vectors $v \in \mathbb{R}^N$, we denote a vector's ℓ_p norm by $\|v\|_p := \sqrt[p]{\sum_{i=1}^{N} |v_i|^p}$. Let $\mathcal{M}_{M \times N}(\mathbb{R})$ indicate the space of square matrices of size $M \times N$ over $\mathbb{R}$. We denote the i-th row vector of A by $A_{i,*}$ and the j-th column vector of A by $A_{*,j}$. We further define $\mathbb{I}$ as $[0, 1]$, and the n-unit simplex as $\mathbb{S}^n$, i.e. $\mathbb{S}^n = \{v \in \mathbb{I}^n, \|v\|_1 = 1\}$. For sets of numbers, we add the subscript $\cdot_+$ to indicate a constraint on positivity for elements in the set. We use $|k\rangle$ to denote a binary encoding of a real number k up to arbitrary precision into a quantum state, and $|\bar{0}\rangle$ to denote a multi-qubit zero state whose number of qubits can be inferred from the context. Lastly, we use $\tilde{\mathcal{O}}(\cdot)$ to omit polylogarithmic factors in asymptotic runtime/memory analysis.

Quantum computation. Quantum algorithms are shown to be able to provide asymptotic speedups over classical counterparts [28, 39, 40] by utilizing characteristics of quantum mechanics such as preparing superpositions of different computational paths.

This property of quantum algorithms allows one to simulate the probability distribution of classical randomized algorithms directly as amplitudes of the quantum state, with the accepted results of the classical algorithms labelled as "good" states, and the rejected results as "bad" states. The *quantum amplitude amplification* (QAA) [41] technique amplifies the amplitudes "good" states such that the success probability increases to a sufficiently high occurrence from its original p upon measurement at a runtime cost of $\mathcal{O}(1/\sqrt{p})$. This provides a quadratic speedup compared to classical Monte Carlo methods, which take on average $\mathcal{O}(1/p)$ samples to achieve success. The QAA technique gives way to another technique known as *quantum amplitude estimation* (QAE) [41], which combines QAA with eigenvalue estimation via quantum phase estimation (QPE) [42] to directly estimate the probability value of the occurrence of a certain event with a quadratic speedup.

Theorem 1 (Quantum amplitude estimation; Theorem 12, [41], Formulation of [43]). *Let $t \in \mathbb{N}$. We are given one copy of a quantum state $|\psi\rangle$ as input, as well as a unitary transformation $U = I - 2|\psi\rangle\langle\psi|$, and a unitary transformation $V = I - 2P$ for some projector P. There exists a quantum algorithm that outputs $\tilde{a}$, an estimate of $a = \|P|\psi\rangle\|^2$, such that*

$$|\tilde{a} - a| \leq 2\pi \frac{\sqrt{a(1-a)}}{M} + \frac{\pi^2}{M^2}$$

with probability at least $8/\pi^2$, using M applications of U and V each.

In this paper, we use QAE to estimate ℓ_1 norms and inner products of vectors $v \in \mathbb{I}^N$ up to a multiplicative error in $M \in \mathcal{O}(\frac{\sqrt{N}}{\varepsilon} \log(\frac{1}{\delta}))$ runtime with probability $1 - \delta$ [35], invoking a quadratic speedup in both the dimension and the error rate. We defer the formulation and details to Appendix A.

Apart from quantum subroutines that provide speedups, we also require the usage of arithmetic operations such as addition, subtraction, multiplication, and division on quantum computers. We assume the arithmetic model, which would allow us to ignore issues arising from the fixed point representation of numbers.[2] We further assume that we have access to quantum arithmetic circuits [44, 45] that can perform such arithmetic operations in $\mathcal{O}(1)$ gates, and that by using such circuits, computation of the n-th power of a number, where $n \in \mathbb{N}$, can be achieved in $\mathcal{O}(\text{poly}\log n)$ gates, using methods like binary exponentiation [46]. We note that quantum arithmetic circuits can be used to execute the same operation on multiple numbers in parallel if the numbers are held in superposition.

In regards to the access of data encoding in quantum states, there are two main models – quantum query access and quantum sample access. For clarity, we highlight quantum states that indicate the index in such access models in bold font throughout the manuscript.

Definition 1 (Quantum query access). Let $n \in \mathbb{N}$ and $c \in \mathcal{O}(1)$. Let a vector of bit strings w be such that $\forall j \in [n], w_j \in \{0, 1\}^c$, and let an arbitrary bit string be $x \in \{0, 1\}^c$. We define quantum query access to w as the access to individual bit strings in w for $j \in [n]$ in the format of

$$|\boldsymbol{j}\rangle |x\rangle \rightarrow |\boldsymbol{j}\rangle |x \oplus w_j\rangle$$

operating on $\mathcal{O}(\log n)$ qubits.

---

[2]If the fixed point representation with an additive error of μ is considered, the additional multiplicative cost required for operations is then $\mathcal{O}(\text{poly}\log \frac{1}{\mu})$. Considering $\mu \in \Omega(1/\text{poly}(m, n))$, the additional cost is $\mathcal{O}(\text{poly}\log(m, n))$, which are polylogarithmic factors that we already omit in this paper.

3

Note 1.1. Note that when $x = 0$, the quantum register stores $|w_j\rangle$ after the query. When $x = w_j$, the quantum register stores $|0\rangle$ after the query. Hence, two consecutive queries onto a quantum register is the identity operation.

Definition 2 (Quantum sample access). Let $n \in \mathbb{N}$ and $w \in \mathbb{R}^n$. We define quantum sample access to w as the access to the index $j \in [n]$ by probability $w_j/\|w\|_1$ in the format of

$$|\bar{0}\rangle \to \sum_{j=0}^{n-1} \sqrt{\frac{w_j}{\|w\|_1}} |\boldsymbol{j}\rangle$$

operating on $\mathcal{O}(\log n)$ qubits.

Note 2.1. In quantum mechanics, Born's rule [47] postulates that the probability of measurement outcome corresponding to a state $|\boldsymbol{j}\rangle$ is proportional to the square of its amplitude under superposition. Hence, the square root in the amplitude regarding the format of quantum sample access.

Lastly, we need to access the input matrices and intermediate vectors as a superposition of encoded quantum states. Such quantum query access to classical data in memory can be achieved by *quantum random access memory* (QRAM) as follows.[3] We refer the reader to [49] for a detailed survey.

Definition 3 (Quantum random access memory; [52, 53]). Let $n \in \mathbb{N}$ and $c \in \mathcal{O}(1)$. Also let w be a vector of bit strings such that $\forall i \in [n], w_i \in \{0,1\}^c$. A quantum RAM provides quantum query access to w_i in superposition after a one-time construction cost of $\tilde{\mathcal{O}}(n)$, where each access costs $\mathcal{O}(\text{poly} \log n)$.

Note 3.1. Quantum sample access can also be provided by QRAM via Grover-Rudolph procedure [54] during construction with logarithmic overhead but is not used in this work.

Fisher market equilibrium. In the Fisher market model [16, 17], we are given a market of m infinitely divisible goods to be divided among n buyers. Without loss of generality, we assume a unit supply for each good. Each buyer $i \in [n]$ has a budget of $B_i > 0$ that has no intrinsic value apart from being used to purchase goods where, again without loss of generality, we assume that the corresponding vector $B \in \mathbb{S}^n$. Each buyer also has a utility function $u_i : \mathbb{R}^m \to \mathbb{R}_+$ that maps an allocation of portions of m items to a utility value. We can then define the allocation matrix $x \in \mathcal{M}_{n \times m}(\mathbb{I})$ such that x_{ij} is the portion of item j allocated to buyer i, where $x_i \in \mathbb{I}^m$ is the bundle of products allocated to buyer i. In this paper, we consider linear utility functions such that $u_i(x_i) = \sum_{j \in [m]} v_{ij} x_{ij}$, where $v_{ij} > 0$ is the value for a unit of item j for buyer i.

Given the Fisher market, we want to compute its *competitive equilibrium*, which consists of the price vector $p \in \mathbb{R}_+^m$ for each item j and allocation matrix x such that each buyer i exhausts their entire budget B_i to acquire a bundle of items x_i that maximizes each of their utility $u_i(x_i)$.

The market equilibrium of Fisher markets can be captured by solving the Eisenberg-Gale (EG) convex program [22, 23]. The program is derived from maximizing the budget-weighted geometric mean of the buyers' utilities i.e. the Nash social welfare, satisfying natural properties such as invariance of the optimal solution to rescaling and splitting [55], and balances the efficiency and fairness regarding the allocation of goods. Applying the $\log$ on the geometric mean, the EG program is as follows:

$$\max_{x \geq 0} \sum_{i \in [n]} B_i \log u_i(x_i) \text{ s.t. } \sum_{i \in [n]} x_{ij} = 1, \forall j \in [m]. \tag{1}$$

where the price p_j is the dual variable of the constraint on x_{ij}. Such convex programs (maximization of a concave function subject to constraints) can be solved by interior point methods [56], but may not scale to large markets. We discuss this further in Section 6.

For the linear Fisher market, an alternative convex program that obtains the same market equilibrium was shown by Shmyrev [57]. Supposing that each buyer i submits a bid b_{ij} for item j such that the sum of the bid of the buyer matches their budget B_i such that each buyer i is allocated $x_{ij} = b_{ij}/p_j$ of item j, we have the following convex program:

$$\max_{b \geq 0} \sum_{ij} b_{ij} \log \frac{v_{ij}}{p_j} \text{ s.t. } \sum_{i \in [n]} b_{ij} = p_j, \forall j \in [m]; \sum_{j \in [m]} b_{ij} = B_i, \forall i \in [n]. \tag{2}$$

As the allocation matrix and price vector can be directly computed from and conversely, be used to compute the bid matrix, the bid matrix can be used as a direct representation of the market equilibrium itself, and hence, is the output of the algorithm we discuss in our paper.

---

[3]Our memory unit can be more precisely termed QRACM [48, 49], including a classical memory, or quantum read-only memory (QROM) [50] as opposed to QRAQM [48, 49] or the quantum random access gate (QRAG) [51], whose memory registers store quantum states instead of classical numbers. However, both are more commonly and jointly referred to as QRAM in the literature.

4

Proportional response dynamics. The proportional response (PR) dynamics is an iterative algorithm [24, 25, 58] that obtains the Fisher market equilibrium computation by updating the bids b_{ij} submitted by buyer i for item j. For each time step t, the elements of the price vector $p_j^{(t)}$ are computed by summing the bids for item j such that $p_j^{(t)} = \sum_i b_{ij}^{(t)}$. The allocation $x_{ij}^{(t)}$ is then obtained by taking $x_{ij}^{(t)} = b_{ij}^{(t)}/p_j^{(t)}$. The buyers then update the bids such that the new bid is proportional to the utility $u_i^{(t)} = \sum_j v_{ij} x_{ij}^{(t)}$ gained in the current time step such that $b_{ij}^{(t+1)} = B_i v_{ij} x_{ij}^{(t)}/u_i^{(t)}$. It was shown by Birnbaum *et al.* [59] that the PR dynamics is equivalent to mirror descent [60, 61] with respect to a Bregman divergence [62] of the Shmyrev convex program.

For ease of discussion, we write the objective function of the EG and Shmyrev convex programs as functions of the bid matrix b, obtaining the EG objective function $\Phi(b) = -\sum_{i\in[n]} B_i \log u_i$ and Shmyrev objective function $\Psi(b) = \sum_{i\in[n],j\in[m]} b_{ij} \log \frac{p_j}{v_{ij}}$. We denote the optimal bid $b^* = \arg\min_{b\in\mathcal{S}} \Phi(b)$, where $\mathcal{S} = \big\{b \in \mathcal{M}_{n\times m}(\mathbb{I}) : \sum_{j\in[m]} b_{ij} = B_i\big\}$.

The convergence bounds of the PR dynamics regarding the EG and Shmyrev objective functions for linear Fisher markets were found as follows:

Theorem 2 (Convergence of PR dynamics; [59]). *Considering a linear Fisher market, for $b_{ij}^{(t)}$ as iteratively defined by the proportional response dynamics where $b_{ij}^{(0)} = \frac{B_i}{m}$, we have*

$$\Psi(b^{(T)}) - \Psi(b^*) \leq \frac{\log m}{T}, \quad \Phi(b^{(T-1)}) - \Phi(b^*) \leq \frac{\log m}{T}. \tag{3}$$

We provide an alternate end-to-end proof of the convergence of both convex programs in Appendix B that varies from Birnbaum *et al.* [59]'s approach and, unlike the latter, is centered around the EG function instead of the Shmyrev function. Elements of this proof are used in the proof of theorems in later sections. Two notable results that we prove and utilize are: 1) $\Psi(b^{(t+1)}) \leq \Phi(b^{(t)}) + \sum_{i\in[n]} B_i \log B_i \leq \Psi(b^{(t)})$, and 2) the telescoping sum of the difference of the KL divergence [63] of the optimal bid and the iterating bids can be lower bounded by the difference of the current EG objective function and the optimal EG function.

3 Faulty proportional response dynamics

Before moving on to our quantum algorithm, we propose the faulty proportional response (FPR) dynamics, which computes an erroneous update to compute a sequence of bids $\hat{b}_{ij}^{(t)}$. Such updates still retain a convergence guarantee, and serve as a counterpart to Theorem 2. We first define a faulty update we use for the FPR dynamics:

Definition 4 (Faulty proportional response update). Let $t \geq 0$ and $\hat{b}^{(t)} \in \mathcal{M}_{n\times m}(\mathbb{I}_+)$. Let us be given $\varepsilon_p \in (0, 0.5)$ and $\tilde{p}^{(t)}$ such that $\forall j, t, |\tilde{p}_j^{(t)} - \hat{p}_j^{(t)}| \leq \hat{p}_j^{(t)} \varepsilon_p$ where $\hat{p}_j^{(t)} = \sum_{i\in[n]} \hat{b}_{ij}^{(t)}$. Further, let us be given $\varepsilon_\nu \in (0, 0.5)$ and $\tilde{\nu}^{(t)}$ such that $\forall i, t, |\tilde{\nu}_i^{(t)} - \hat{\nu}_i^{(t)}| \leq \hat{\nu}_j^{(t)} \varepsilon_\nu$ where $\hat{\nu}_i^{(t)} = \sum_{j\in[m]} v_{ij} \hat{b}_{ij}^{(t)}/\tilde{p}_j^{(t)}$. A faulty proportional response update of the bids from timestep t to $t+1$ is then expressed as follows:

$$\hat{x}_{ij}^{(t)} = \frac{\hat{b}_{ij}^{(t)}}{\tilde{p}_j^{(t)}}, \quad \hat{b}_{ij}^{(t+1)} = B_i \frac{v_{ij} \hat{x}_{ij}^{(t)}}{\tilde{\nu}_i^{(t)}}.$$

Note that while $\tilde{p}_j$ provides an estimation to the price $\hat{p}_j = \sum_{i\in[n]} \hat{b}_{ij}$, $\tilde{\nu}_i$ does not provide an estimation to the exact utility $\hat{u}_i = \sum_{j\in[m]} v_{ij} \hat{b}_{ij}/\hat{p}_j$. Instead, $\tilde{\nu}_i$ estimates $\hat{\nu}_i$, which replaces $\hat{p}_j$ in the computation of u_i with $\tilde{p}_j$.

We find the convergence bounds of the FPR dynamics regarding the EG objective function for linear Fisher markets are as follows:

Theorem 3 (Convergence of the FPR dynamics). *Considering a linear Fisher market, for $b_{ij}^{(t)}$ as iteratively defined by the faulty proportional response dynamics where $\hat{b}_{ij}^{(0)} = \frac{B_i}{m}$, we have*

$$\min_{t\in[T]} \Phi(\hat{b}^{(t)}) - \Phi(b^*) \leq \frac{2\log m}{T}$$

when $\varepsilon_\nu \leq \frac{\log m}{8T}$ and $\varepsilon_p \leq \frac{\log m}{6T}$.

5

A high-level idea of the proof follows from the telescoping sum trick to upper bound the EG objective functions with KL divergence from our proof of PR dynamics but with the consideration of error. We show an end-to-end proof of the convergence of the EG objective function in Appendix C.

Notice that in the FPR dynamics, we do not enforce the monotonicity of the iterations, but instead simply take the minimum value over all iterations. The error terms ε_p and ε_ν in the FPR dynamics are only upper bounded such that the total sum of objective values over T iterations (plus the original iteration) can be upper bounded by $\log m$ plus an accumulated error over T iterations also within $\log m$. If we enforce the monotonicity of the iterations to take the last iteration, the error would require $\mathcal{O}(1/T^2)$ precision and would incur a further multiplicative overhead of T in our algorithm.

However, given the formulation of a faulty update, a problem that comes into question is whether the computation of the exact value of the function $\Phi(b)$ is supported, as we do not compute u_i in the process of updating. Without computation of $\Phi(b)$, one can not be sure which iteration of $\hat{b}^{(t)}$ is the minimum. On the other hand, we can use the computed value of $\tilde{\nu}_i^{(t)}$ as an estimator for the function $\Phi(b)$. The following result is then obtained.

Theorem 4. *Considering a linear Fisher market, for $b_{ij}^{(t)}$ as iteratively defined by the faulty proportional response dynamics where $\hat{b}_{ij}^{(0)} = \frac{B_i}{m}$. Let $t^* = \arg\max_{t \in [T]} \sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)}$. Then*

$$\Phi(\hat{b}^{(t^*)}) - \Phi(b^*) \leq \frac{2 \log m}{T}$$

when $\varepsilon_\nu \leq \frac{\log m}{8T}$ and $\varepsilon_p \leq \frac{\log m}{6T}$.

The proof of this theorem can similarly be found in Appendix C, which has the same proof idea as Theorem 3 apart from some slight differences in error handling.

4 Quantum algorithm

We present our quantum algorithm for solving linear Fisher market equilibrium computation based on the FPR dynamics. Our quantum algorithm does not aim to provide speedups in terms of the number of iterations but provides speedups on the iteration cost of the PR dynamics algorithm. Our algorithm, while reducing the runtime in terms of the number of buyers n or goods m, increases runtime in terms of the number of iterations T but as the T is logarithmically dependent on m, there is an overall quadratic speedup provided in the smaller of the two dimensions.

In this section, we further assume that $v \in \mathcal{M}_{n \times m}(\mathbb{I}_+)$. We note that the multiplicative scaling of v_{ij} does not affect the bid matrix b generated in the FPR dynamics as errors are multiplicative. Hence if the values are larger than 1, we scale down the values by dividing the queried v_{ij} by a number that is larger than $\max_{ij} v_{ij}$.

To compute the market equilibrium for the Fisher market by the FPR dynamics in the quantum setting, we require the data input of both the budget vector $B \in \mathbb{S}^n$ and the value matrix $v \in \mathcal{M}_{n \times m}(\mathbb{I}_+)$. We assume quantum query access to the budget and vector and value matrix by the index is readily given to us as part of the problem input without having to load classical data into a quantum system. That is, given an index state and ancilla quantum registers we can store the value of the budget and value according to the index in the ancilla register. Note that these operations can be performed in superposition, such that $\frac{1}{\sqrt{mn}} \sum_{i,j} |i\rangle|j\rangle |\bar{0}\rangle \rightarrow \frac{1}{\sqrt{mn}} \sum_{i,j} |i\rangle|j\rangle |v_{ij}\rangle$.

We do not explicitly state how the data input of the budget and value entries are generated; they could be extracted from entries of a matrix already preloaded in QRAM, or generated/reconstructed from a low-rank approximation of the matrix [15], which takes $\mathcal{O}(k \operatorname{poly} \log mn)$ cost to access k-rank approximations using quantum arithmetic circuits, but with much lower memory consumption.[4] We note that the low-rank approximation assumption of the value matrix has not yet been utilized to produce reductions in resource consumption in classical methods as the PR dynamics and other methods to compute market equilibrium [26] require all mn entries of the full value matrix as input.

Storing the results of the computed bids $\hat{b}^{(t)} \in \mathcal{M}_{n \times m}(\mathbb{I}_+)$ in QRAM would require a cost of $\tilde{\mathcal{O}}(mn)$ which would remove all possibility of potential speedups. The same applies to the allocation matrix

---

[4]With low-rank approximations, loading classical data into QRAM would only take $\tilde{\mathcal{O}}(k(m + n))$ runtime.

6

$\hat{x}^{(t)}$. Hence, every time we require the usage of $\hat{b}^{(t)}$ or $\hat{x}^{(t)}$, we compute them on-the-fly as follows:

$$\hat{b}_{ij}^{(t)} = \frac{B_i^{t+1} v_{ij}^t}{m \prod_{k=0}^{t-1} \tilde{p}_j^{(k)} \prod_{k=0}^{t-1} \tilde{\nu}_i^{(k)}}, \quad \hat{x}_{ij}^{(t)} = \frac{B_i^{t+1} v_{ij}^t}{m \prod_{k=0}^{t} \tilde{p}_j^{(k)} \prod_{k=0}^{t-1} \tilde{\nu}_i^{(k)}} \tag{4}$$

Given quantum query access to the values of $(\Pi_p^{(t)})_j := \prod_{k=0}^{t} \tilde{p}_j^{(k)}$ and $(\Pi_\nu^{(t)})_i := \prod_{k=0}^{t} \tilde{\nu}_i^{(k)}$, one can encode the values of $\hat{b}^{(t+1)}$ and $\hat{x}^{(t)}$ into a quantum state in superposition via quantum arithmetic circuits in runtime of $\mathcal{O}(\text{poly} \log tmn)$. In our algorithm, we compute and store the vectors $\Pi_p^{(t)}$ and $\Pi_\nu^{(t)}$ in QRAM, after which, query access to such values cost $\mathcal{O}(\text{poly} \log mn)$. Taking the t-th power of the budget and value cost $\mathcal{O}(\text{poly} \log t)$ by using CNOT gates and binary exponentiation [46].

The remaining steps are to compute the price vector $\tilde{p}$ and utility vector $\tilde{u}$ in each iteration. Each entry $\tilde{p}_j$ is the estimation of the ℓ_1 norm of $\hat{b}_{*,j}$ and each entry $\tilde{\nu}_i$ is the estimation of the inner product between $\hat{x}_{i,*}$ and $v_{i,*}$, which can both be obtained using QAE. $\Pi_p^{(t)}$ and $\Pi_\nu^{(t)}$ can then be iteratively updated by multiplying by the values of $\tilde{p}$ and $\tilde{\nu}$ each iteration. The full algorithm is shown in Algorithm 1, and the algorithmic guarantee is shown in Theorem 5.

Algorithm 1: Quantum algorithm for faulty proportional response dynamics

---

Input: Quantum query access to B and v, Timestep limit T, Price error ε_p, Utility error ε_ν

1   maxEGVal $= -\infty$, $b_{ij}^{(0)} = \frac{B_i}{m}$
2   **for** $t = 0$ *to* T **do**
3   **for** $j = 0$ *to* m **do**
4       $\tilde{p}_j^{(t)} = \|\hat{b}_{*,j}^{(t)}\|_1 (1 \pm \varepsilon_p)$ via Q norm est. (Lemma A.1) with success prob. $1 - \frac{\delta}{2mT}$
5   Store vector $\Pi_p^{(t)} = \tilde{p}^{(t)} \odot \Pi_p^{(t-1)}$ into QRAM
6   Gain access to $\hat{x}_{ij}^{(t)}$ via $\Pi_p^{(t)}$ and $\Pi_\nu^{(t-1)}$ in QRAM
7   **for** $i = 0$ *to* n **do**
8       $\tilde{\nu}_i^{(t)} = \langle x_{i,*}^{(t)}, v_{i,*} \rangle (1 \pm \varepsilon_\nu)$ via Q inner prod. est. (Lemma A.3) with success prob. $1 - \frac{\delta}{2nT}$
9   Store vector $\Pi_\nu^{(t)} = \tilde{\nu}^{(t)} \odot \Pi_\nu^{(t-1)}$ into QRAM
10   Gain access to $\hat{b}_{ij}^{(t+1)}$ via $\Pi_p^{(t)}$ and $\Pi_\nu^{(t)}$ in QRAM
11   Classically compute $\tilde{\Phi}^{(t)} = \sum_{i \in [n]} B_i \log(\tilde{\nu}_i^{(t)})$
12   **if** $\tilde{\Phi}^{(t)} > maxEGVal$ **then**
13   maxEGVal $= \tilde{\Phi}^{(t)}$, bestPiP $= \Pi_p^{(t-1)}$, bestPiNu $= \Pi_\nu^{(t-1)}$

14 **return** bestPiP and bestPiNu in QRAM

Theorem 5 (Quantum algorithm for the faulty proportional response dynamics). *Let* $\delta \in (0, 0.5), n, m, T \in \mathbb{N}$, $\varepsilon_p = \frac{\log m}{8T}$, *and* $\varepsilon_\nu = \frac{\log m}{6T}$. *Given quantum query access to B and v, and access to QRAM, with success probability $1 - \delta$, Algorithm 1 produces values stored in QRAM such that query and sample access to the values of $\hat{b}^{(t^*)}$ can be constructed, where*

$$\Phi(\hat{b}^{(t^*)}) - \Phi(b^*) \leq \frac{2 \log m}{T},$$

with $\tilde{\mathcal{O}}(T^2 \sqrt{mn \max(m, n)} \log \frac{1}{\delta})$ runtime and $\tilde{\mathcal{O}}(m + n)$ QRAM space. To provide query access to $\hat{b}^{(t^*)}$, an additional cost of $\mathcal{O}(\text{poly} \log Tmn)$ is incurred from accessing $\Pi_p^{(t^*-1)}$ and $\Pi_\nu^{(t^*-1)}$ in QRAM. Providing sample access to $\hat{b}^{(t^*)}$ requires additional cost of $\mathcal{O}(\sqrt{mn} \, \text{poly} \log T)$ from QAA.

Proof. Per union bound [64], we find that the total success probability is at least $1 - \delta$. The output of Algorithm 1 of bestPiP and bestPiNu corresponds to the values of $\Pi_p^{(t^*-1)}$ and $\Pi_\nu^{(t^*-1)}$. This gives us the guarantee of convergence shown in Theorem 4.

For the runtime analysis, the quantum norm estimation subroutine (Lemma A.1) takes $\mathcal{O}(\frac{\sqrt{n}}{\varepsilon_p} \log \frac{mT}{\delta})$ for mT iterations, while quantum inner product (Lemma A.3) estimation takes $\mathcal{O}(\frac{\sqrt{m}}{\varepsilon_\nu} \log \frac{nT}{\delta})$ for

7

nT iterations, resulting in a total runtime of $\tilde{\mathcal{O}}(T^2\sqrt{mn\max(m,n)}\log\frac{1}{\delta})$. For uses of QRAM, the construction on Lines 5 and 9 is a one-time cost of $\tilde{\mathcal{O}}(n)$ and $\tilde{\mathcal{O}}(m)$, respectively, with a total runtime of $\tilde{\mathcal{O}}(T(m+n))$. The classical computation of the EG value in Line 11 costs $\mathcal{O}(Tn)$. We note that the quantum norm and inner product estimation subroutine is the main bottleneck of the algorithm, and hence the total runtime is then $\tilde{\mathcal{O}}(T^2\sqrt{mn\max(m,n)}\log\frac{1}{\delta})$.

For the memory complexity, for the t-th iteration, we require 6 vectors in QRAM: the current iteration $\Pi_p^{(t)}$ and $\Pi_\nu^{(t)}$, the best iteration `bestPiP` and `bestPiNu` and the previous iteration $\Pi_p^{(t-1)}$ and $\Pi_\nu^{(t-1)}$, in case we need to update `bestPiP` and `bestPiNu`. Note that to update the best iteration, we simply reroute the register of the previous iteration to being the best iteration. There is no need to copy data or reconstruct a new QRAM as the data from the previous iteration is no longer needed in the next iteration. Therefore, the memory is $\mathcal{O}(m+n)$ for storing the 6 vectors.

The values `bestPiP` and `bestPiNu` can be used to construct query access $\hat{b}^{(t^*)}$ per Equation (4). To prepare quantum sample access to the matrix $\hat{b}^{(t^*)}$, using quantum query access to $\hat{b}^{(t^*)}$ from Algorithm 1 and Equation (4) in addition to conditioned rotation gates, we can encode the values of $\hat{b}^{(t^*)}$ onto the amplitudes of an ancilla qubit via Lemma A.1 such that we obtain

$$\frac{1}{\sqrt{mn}}\sum_{i,j}|\boldsymbol{i}\rangle|\boldsymbol{j}\rangle|0\rangle \rightarrow \frac{1}{\sqrt{mn}}\sum_{i,j}|\boldsymbol{i}\rangle|\boldsymbol{j}\rangle\left(\sqrt{\hat{b}_{ij}^{(t^*)}}|0\rangle + \sqrt{1-\hat{b}_{ij}^{(t^*)}}|1\rangle\right) \tag{5}$$

Note that states where the last qubit is $|0\rangle$ are the "good" states that we want, while the states where the last qubit is $|1\rangle$ are "bad" states. To get rid of the "bad" states, we use QAA to amplify the amplitudes of the "good" states. Rewriting the above terms and ignoring the ancilla qubits, we obtain

$$\sqrt{\frac{\|\hat{b}^{(t^*)}\|_1}{mn}}\sum_{i,j}|\boldsymbol{i}\rangle|\boldsymbol{j}\rangle\sqrt{\frac{\hat{b}_{ij}^{(t^*)}}{\|\hat{b}^{(t^*)}\|_1}}|0\rangle + \sqrt{1-\frac{\|\hat{b}^{(t^*)}\|_1}{mn}}\sum_{i,j}|\boldsymbol{i}\rangle|\boldsymbol{j}\rangle\sqrt{\frac{1-\hat{b}_{ij}^{(t^*)}}{mn-\|\hat{b}^{(t^*)}\|_1}}|1\rangle \tag{6}$$

We see from the above that the success probability of obtaining "good" states is $\frac{\|\hat{b}^{(t^*)}\|_1}{mn}$. Hence, QAA costs $\mathcal{O}\left(\sqrt{\frac{mn}{\|\hat{b}^{(t^*)}\|_1}}\right)$ query accesses to $b^{(t^*)}$, and while the exact value of $\|\hat{b}^{(t^*)}\|_1$ is not known, by Definition 4, $\frac{1}{1+\varepsilon_\nu}\leq\|\hat{b}^{(t^*)}\|_1\leq\frac{1}{1-\varepsilon_\nu}$. Hence, the query complexity of $b^{(t^*)}$ can be further upper bounded as $\mathcal{O}(\sqrt{mn(1+\varepsilon_\nu)})\subset\mathcal{O}(\sqrt{mn})$. On the other hand, the cost of query access to $b^{(t^*)}$ has $\mathcal{O}(\text{poly}\log Tmn)$ cost, therefore, the total cost for sample access to $b^{(t^*)}$ is $\tilde{\mathcal{O}}(\sqrt{mn}\,\text{poly}\log T)$.

$\square$

5 Numerical simulations

We simulate the market equilibrium computation under PR dynamics and our quantum algorithm.[5] To showcase the effects of quantum speedups, we fixed the number of queries to all bid matrices $b^{(t)}$ and observed the reduction of the objective value over the number of queries.

As an actual simulation of QAE using quantum gates over multiple qubits is costly, we directly compute the probability vector of $\Pr[Z=z]$ for $z\in[M]$ that one would obtain by QAE [41] for a target value a,

$$\Pr[Z=z] = \frac{\sin^2(M\Delta_z\pi)}{M^2\sin^2(\Delta_z\pi)} \tag{7}$$

where $\Delta_z = \min(|z-\sin^{-1}(\sqrt{a})/\pi|, |1-z+\sin^{-1}(\sqrt{a})/\pi|)$. M is the number of times that call the unitaries U and V in QAE (see Theorem 1), and is linearly correlated to the runtime. We then sample the output according to the computed probabilities to obtain an estimator $\tilde{a}=\sin^2(\pi\frac{z}{M})$.

For our experiments, we generate the input data v where the value v is sampled from a uniform distribution with range $(0,1]$ and a normal distribution $\mathcal{N}(0.5,0.25)$, where we resample values that fall outside the range of $(0,1]$. For the budget B, we either sample from the same distribution as the value matrix or set the same budget for all buyers to simulate competitive equilibrium from equal income (CEEI) applications. Our simulation includes $n=16384$ buyers, $m=16384$ goods, and

---

[5]The codebase can be found at `https://github.com/georgepwhuang/q-market-equilibrium`.

8
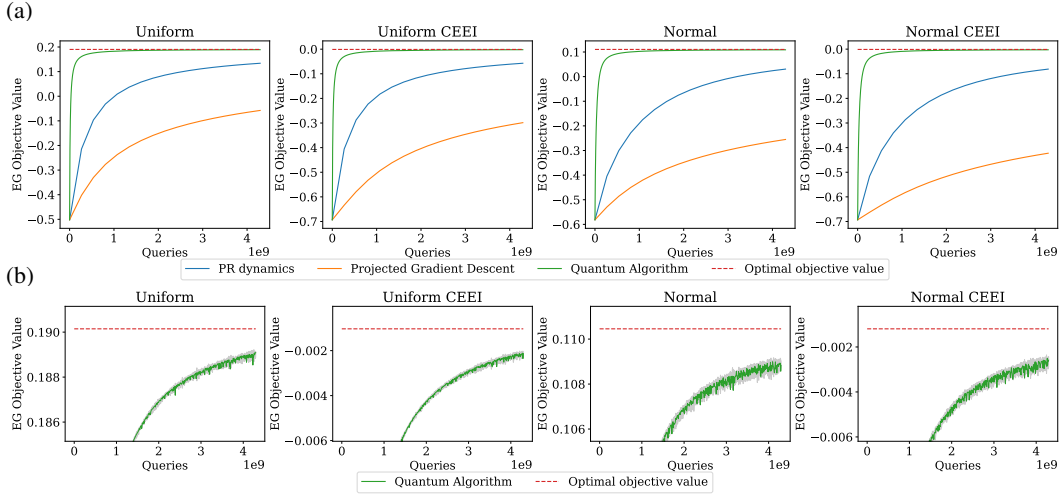
Figure 1: *Experimental results.* We perform our experiments on $n = 16384$ buyers and $m = 16384$ goods given the same amount of queries for all algorithms. We observe in Figure 1a that over different distributions, our quantum algorithm (green) significantly outperforms the PR dynamics (blue), which aligns with our theoretical results. Furthermore, our results also show that both our quantum algorithm and the PR dynamics outperform projected gradient descent (orange) in the mid-accuracy regime. Figure 1b shows the convergence of a single run of the quantum algorithm despite its instability from faulty updates, as well as the variance over the multiple runs (shaded in gray).

iterate for $T = 16$ iterations for the PR dynamics. For the quantum algorithm, note that the queries per iteration would be reduced by $\sqrt{n}$ if we use an actual quantum computer, hence increasing the number of iterations to fix the number of queries. For QAE, we run for $\sqrt{T\sqrt{n}} = 512$ iterations and set $M \in \mathcal{O}(\sqrt{T\sqrt{n}})$. As the classical algorithms are deterministic, we rerun our quantum algorithm over 15 times with the same sample of B and v to observe the variance of convergence progress. Results are shown in Figure 1. Details on implementation and setup are found in Appendix D.

From the plots of Figure 1, we note that the results fit our theoretical results in that the quantum algorithm converges much faster than that of the PR dynamics [24]. Further, we also compare against the convergence of projected gradient descent, which supports empirical results by Gao and Kroer [26] that in the regime of mid-level accuracy and low iterations, PR dynamics-related algorithms, both classical and quantum, converge faster than projected gradient descent.

6 Discussion

Quasi-linear utilities. For the bulk of our paper, we focus on the setting of linear utilities for Fisher markets. However, applications of market equilibrium computation in large-scale Fisher markets involve mostly quasi-linear utilities [9]. An approach for using PR dynamics for quasi-linear utilities proposed by Gao and Kroer [26][6] includes the usage of slack variables $\delta = (\delta_1, \cdots, \delta_m)$ that represent the buyers' leftover budgets. The PR updates are then modified as follows:

$$b_{ij}^{(t+1)} = B_i \frac{v_{ij} x_{ij}^{(t)}}{\sum_{j'} v_{ij'} x_{ij'}^{(t)} + \delta_i^{(t)}}, \ \delta_i^{(t+1)} = B_i \frac{\delta_i^{(t)}}{\sum_{j'} v_{ij'} x_{ij'}^{(t)} + \delta_i^{(t)}}. \tag{8}$$

Further, PR dynamics for quasi-linear utilities exhibit a convergence rate of $\mathcal{O}\left(\frac{\log(m+1)}{T}\right)$. Using the methods discussed in previous sections, the quasi-linear version of PR dynamics can then be readily adapted to its quantum version by employing the same techniques of computing and storing in QRAM the values $\Pi_p^{(t)}$ and $\Pi_\nu^{(t)}$ with on-the-fly computation of $\hat{b}^{(t+1)}$, $\hat{x}^{(t)}$ and $\hat{\delta}^{(t+1)}$.

Constant number of buyers. Notice that our quantum algorithm provides a quadratic speedup on the smaller value in regards to the number of buyers n and number of goods m. Therefore given extreme cases where the number of buyers $n \in \mathcal{O}(1)$, our algorithm does not provide a speedup.

---

[6]There is another method proposed by Cheung *et al.* [65], which we find difficult to convert to quantum due to its use of thresholding, which would cause problems with faulty updates from the FPR dynamics.

9

However, in such cases, quantum speedups may still be obtained simply by removing the QAE step for estimating the price for each item and replacing it with using quantum arithmetic circuits to compute the exact sum. We use a total of $\mathcal{O}(nT \operatorname{poly} \log(T, m, n))$ qubits to compute the values of $b_{i,*}$ separately on the i-th set of qubits, and only conduct QAE when estimating the utility value for each buyer. Given that in this setting, $n \in \mathcal{O}(1)$, the total runtime would then be $\mathcal{O}\left(T^2 \sqrt{m} \log \frac{1}{\delta}\right)$, gaining a quadratic speedup over the number of goods m.

Dequantization. Given the work in recent years towards the development of quantum-inspired classical algorithms [66–69] that achieve similar performances as quantum algorithms using sampling-based techniques, a natural question that arises is whether our algorithm can be "de-quantized". The main speedup in our algorithm stems from the usage of estimation of ℓ_1 norms and inner products. While the use of sampling techniques can indeed provide inner product estimations, they retain the same $\mathcal{O}(1/\varepsilon^2)$ dependency instead of the $\mathcal{O}(1/\varepsilon)$ dependency of QAE. Hence, our algorithm performance may be hard to replicate in classical settings.

On the other hand, while it has been suggested that the computation of market equilibrium may benefit from low-rank approximations [15], methods of using such properties to accelerate the computation of gradients have not been proposed, given that the update of the PR dynamics rely on element-wise multiplication of matrices instead matrix multiplication. This would suggest that using sampling techniques to accelerate updates would be similarly difficult.

Practicality. Our quantum algorithm relies mostly on the QAE subroutine to achieve quantum speedups. In its original formulation, QAE involves the usage of QPE as a subroutine, which requires multiple controlled unitaries and the quantum Fourier transform [70]. QAE is thus regarded as a fault-tolerant quantum subroutine, whose execution may be beyond the capabilities of current quantum hardware. However, many subtle improvements to the QAE algorithm have since been made after its discovery, such as simplifying subroutines [71–76] or trading circuit depth with speedup factors [75, 77, 78]. Such improvements pave the way for potential implementation of QAE, and by extension, this paper, on next-generation quantum hardware in the early fault-tolerant regime [79, 80].

Potential and limitations for further quantum speedups. Our quantum algorithm shares similarities to other quantum algorithms that are based on the *multiplicative weight update* (MWU) method [81, 82]. Such methods have found success in obtaining quantum speedups for LPs [29] and SDPs [30–33], which have been extended to applications such as zero-sum games [29, 83], quadratic binary optimization [84], and financial applications [35, 85]. Apart from the MWU-esque PR dynamics, various other methods for computing market equilibrium have also been proposed. Can quantum speedups obtained from these methods exceed those of our quantum algorithm?

Tracing back to the roots of the EG convex program [22, 23] and Shmyrev convex program [57], it is well known that such programs can be solved in polynomial time with interior-point methods (IPM) [56]. However, as IPMs require using linear solvers as subroutines, and as there is no guarantee of well-conditioned systems, the quantum linear systems solver [37, 86] may not provide significant speedup. Therefore, it may be unlikely that quantum IPMs [87] can provide significant speedups.

First-order methods such as the Frank-Wolfe (FW) algorithm [88] and projected gradient descent (PGD) have also been discussed as candidates for solving market equilibrium [26], with PGD achieving linear convergence classically. While PGD obtains a superior asymptotic convergence rate in terms of the error ε compared to PR dynamics, as our quantum speedups stem from faster computations of results within a single iteration, it may be harder to find such speedups for PGD as there has been no evidence for quantum speedups in projections onto a simplex [89, 90] as required.

On the other hand, the FW algorithm has been shown to provide quantum speedups for regression [91, 92]. However, convergence results of FW [93, 94] show that $\Phi(b^{(T)}) - \Phi(b^*) \leq \frac{C_\Phi}{(T+2)}$, where C_Φ can be shown to be $\mathcal{O}(n)$ by computing relevant values. The number of iterations T required for convergence to additive error ε is then $\mathcal{O}(\frac{n}{\varepsilon})$ as compared to $\mathcal{O}(\frac{\log m}{\varepsilon})$ of PR dynamics. This matches the results of Gao and Kroer [26], which show that FW has slow convergence empirically for market equilibrium computation. Prior no-go results suggest that quantum algorithms cannot provide speedups for T when T is independent of the problem dimension [95, 96]. Assuming no quantum speedups in T, given the $\mathcal{O}(n)$ upper bound in the FW algorithm, the quantum algorithm based on FW can potentially have a higher dependency on n than the classical PR dynamics.

10

Acknowledgments and Disclosure of Funding

The authors thank Gregory Kang Ruey Lau, Jinge Bao, and Warut Suksompong for discussions. This work is supported by the National Research Foundation, Singapore, and A*STAR under its CQT Bridging Grant and its Quantum Engineering Programme under grant NRF2021-QEP2-02-P05.

References

[1] K. J. Arrow, An extension of the basic theorems of classical welfare economics, in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley Symp. on Math. Statist. and Prob (University of California Press, Berkeley, California, USA, 1951) pp. 507–532.

[2] G. Debreu, The coefficient of resource utilization, Econometrica **19**, 273 (1951).

[3] K. J. Arrow and G. Debreu, Existence of an equilibrium for a competitive economy, Econometrica **22**, 265 (1954).

[4] V. V. Vazirani, Combinatorial algorithms for market equilibria, in *Algorithmic Game Theory*, edited by N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani (Cambridge University Press, 2007) p. 103–134.

[5] B. Codenotti and K. Varadarajan, Computation of market equilibria by convex programming, in *Algorithmic Game Theory*, edited by N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani (Cambridge University Press, 2007) p. 135–158.

[6] A. Mas-Colell, M. D. Whinston, and J. R. Green, Equilibrium and its basic welfare properties, in *Microeconomic theory* (Oxford Univ. Press, 1995) Chap. 16.

[7] D. K. Foley, *Resource allocation and the public sector*, Ph.D. thesis, Yale University (1967).

[8] H. R. Varian, Equity, envy, and efficiency, J. Econ. Theory **9**, 63–91 (1974).

[9] C. Kroer and N. E. Stier-Moses, Market equilibrium models in large-scale internet markets, in *Springer Series in Supply Chain Management* (Springer International Publishing, 2021) p. 147–189.

[10] V. Conitzer, C. Kroer, D. Panigrahi, O. Schrijvers, N. E. Stier-Moses, E. Sodomka, and C. A. Wilkens, Pacing equilibrium in first price auction markets, Manag. Sci. **68**, 8515–8535 (2022).

[11] A. Othman, T. Sandholm, and E. Budish, Finding approximate competitive equilibria: efficient and fair course allocation, in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10 (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2010) p. 873–880.

[12] E. Budish, G. P. Cachon, J. B. Kessler, and A. Othman, Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation, Oper. Res. **65**, 314–336 (2017).

[13] M. Babaioff, N. Nisan, and I. Talgam-Cohen, Fair allocation through competitive equilibrium from generic incomes, in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19 (Association for Computing Machinery, New York, NY, USA, 2019) p. 180.

[14] S. Im, J. Kulkarni, and K. Munagala, Competitive algorithms from competitive equilibria: Non-clairvoyant scheduling under polyhedral constraints, J. ACM **65** (2017).

[15] C. Kroer, A. Peysakhovich, E. Sodomka, and N. E. Stier-Moses, Computing large market equilibria using abstractions, Oper. Res. **70**, 329–351 (2022).

[16] I. Fisher, *Mathematical investigations in the theory of value and prices*, Ph.D. thesis, Yale University (1891).

[17] W. C. Brainard and H. E. Scarf, How to compute equilibrium prices in 1891, Am. J. Econ. Sociol. **64**, 57–83 (2005).

[18] H. E. Scarf, The core of an n person game, Econometrica **35**, 50 (1967).

[19] N. R. Devanur, C. H. Papadimitriou, A. Saberi, and V. V. Vazirani, Market equilibrium via a primal-dual algorithm for a convex program, J. ACM **55** (2008).

11

[20] J. B. Orlin, Improved algorithms for computing Fisher's market clearing prices, in *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10 (Association for Computing Machinery, New York, NY, USA, 2010) p. 291–300.

[21] L. A. Végh, Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives, in *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '12 (Association for Computing Machinery, New York, NY, USA, 2012) p. 27–40.

[22] E. Eisenberg and D. Gale, Consensus of subjective probabilities: The pari-mutuel method, Ann. Math. Stat. **30**, 165–168 (1959).

[23] E. Eisenberg, Aggregation of utility functions, Manag. Sci. **7**, 337–350 (1961).

[24] F. Wu and L. Zhang, Proportional response dynamics leads to market equilibrium, in *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07 (Association for Computing Machinery, New York, NY, USA, 2007) p. 354–363.

[25] L. Zhang, Proportional response dynamics in the Fisher market, Theor. Comput. Sci. **412**, 2691–2698 (2011).

[26] Y. Gao and C. Kroer, First-order methods for large-scale market equilibrium computation, in *Advances in Neural Information Processing Systems*, Vol. 33, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, Inc., 2020) pp. 21738–21750.

[27] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).

[28] A. Abbas, A. Ambainis, B. Augustino, A. Bärtschi, H. Buhrman, C. Coffrin, G. Cortiana, V. Dunjko, D. J. Egger, B. G. Elmegreen, N. Franco, F. Fratini, B. Fuller, J. Gacon, C. Gonciulea, S. Gribling, S. Gupta, S. Hadfield, R. Heese, G. Kircher, T. Kleinert, T. Koch, G. Korpas, S. Lenk, J. Marecek, V. Markov, G. Mazzola, S. Mensa, N. Mohseni, G. Nannicini, C. O'Meara, E. P. Tapia, S. Pokutta, M. Proissl, P. Rebentrost, E. Sahin, B. C. B. Symons, S. Tornow, V. Valls, S. Woerner, M. L. Wolf-Bauwens, J. Yard, S. Yarkoni, D. Zechiel, S. Zhuk, and C. Zoufal, Challenges and opportunities in quantum optimization, Nat. Rev. Phys. (2024).

[29] J. van Apeldoorn and A. Gilyén, Quantum algorithms for zero-sum games (2019), arXiv:1904.03180 [quant-ph] .

[30] F. G. S. L. Brandão and K. M. Svore, Quantum speed-ups for solving semidefinite programs, in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE Computer Society, Los Alamitos, CA, USA, 2017) pp. 415–426.

[31] F. G. S. L. Brandão, A. Kalev, T. Li, C. Y.-Y. Lin, K. M. Svore, and X. Wu, Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning, in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 132, edited by C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2019) pp. 27:1–27:14.

[32] J. van Apeldoorn and A. Gilyén, Improvements in quantum SDP-solving with applications, in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 132, edited by C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2019) pp. 99:1–99:15.

[33] J. van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf, Quantum SDP-solvers: Better upper and lower bounds, Quantum **4**, 230 (2020).

[34] T. Li, S. Chakrabarti, and X. Wu, Sublinear quantum algorithms for training linear and kernel-based classifiers, in *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 97, edited by K. Chaudhuri and R. Salakhutdinov (PMLR, 2019) pp. 3815–3824.

[35] P. Rebentrost, Y. Hamoudi, M. Ray, X. Wang, S. Yang, and M. Santha, Quantum algorithms for hedging and the learning of Ising models, Phys. Rev. A **103**, 012418 (2021).

[36] I. Kerenidis and A. Prakash, Quantum recommendation systems, in *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 67, edited by C. H. Papadimitriou (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2017) pp. 49:1–49:21.

12

[37] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. **103**, 150502 (2009).

[38] S. Aaronson, Read the fine print, Nat. Phys. **11**, 291–293 (2015).

[39] A. Montanaro, Quantum algorithms: an overview, npj Quantum Inf. **2** (2016).

[40] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, G. Salton, S. Wang, and F. G. S. L. Brandão, Quantum algorithms: A survey of applications and end-to-end complexities (2023), arXiv:2310.03011 [quant-ph] .

[41] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, in *Quantum computation and information*, Contemporary Mathematics, Vol. 305 (American Mathematical Society, Providence, RI, USA, 2002) pp. 53–74.

[42] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem (1995), arXiv:quant-ph/9511026 [quant-ph] .

[43] A. Montanaro, Quantum speedup of Monte Carlo methods, Proc. R. Soc. A: Math. Phys. Eng. Sci. **471**, 20150301 (2015).

[44] V. Vedral, A. Barenco, and A. Ekert, Quantum networks for elementary arithmetic operations, Phys. Rev. A **54**, 147–153 (1996).

[45] Y. Takahashi, Quantum arithmetic circuits: A survey, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E92-A**, 1276–1283 (2009).

[46] P. L. Montgomery, Speeding the Pollard and elliptic curve methods of factorization, Math. Comput. **48**, 243–264 (1987).

[47] M. Born, Zur Quantenmechanik der Stoßvorgänge, Z. Physik **37**, 863–867 (1926).

[48] G. Kuperberg, Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem, in *8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 22 (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2013) pp. 20–34.

[49] S. Jaques and A. G. Rattew, QRAM: A survey and critique (2023), arXiv:2305.10310 [quant-ph] .

[50] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding electronic spectra in quantum circuits with linear T complexity, Phys. Rev. X **8**, 041015 (2018).

[51] A. Ambainis, Quantum walk algorithm for element distinctness, SIAM J. Comput. **37**, 210–239 (2007).

[52] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum random access memory, Phys. Rev. Lett. **100**, 160501 (2008).

[53] V. Giovannetti, S. Lloyd, and L. Maccone, Architectures for a quantum random access memory, Phys. Rev. A **78**, 052310 (2008).

[54] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions (2002), arXiv:quant-ph/0208112 [quant-ph] .

[55] K. Jain and V. V. Vazirani, Eisenberg–Gale markets: Algorithms and game-theoretic properties, Games Econ. Behav. **70**, 84–106 (2010).

[56] S. Boyd and L. Vandenberghe, Interior-point methods, in *Convex Optimization* (Cambridge University Press, 2004) p. 561–630.

[57] V. I. Shmyrev, An algorithm for finding equilibrium in the linear exchange model with fixed budgets, J. Appl. Ind. Math. **3**, 505–518 (2009).

[58] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee, Bittorrent is an auction: Analyzing and improving bittorrent's incentives, SIGCOMM Comput. Commun. Rev. **38**, 243–254 (2008).

[59] B. Birnbaum, N. R. Devanur, and L. Xiao, Distributed algorithms via gradient descent for Fisher markets, in *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11 (Association for Computing Machinery, New York, NY, USA, 2011) p. 127–136.

13

[60] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization* (Wiley, 1983).

[61] A. Beck and M. Teboulle, Mirror descent and nonlinear projected subgradient methods for convex optimization, Oper. Res. Lett. **31**, 167–175 (2003).

[62] L. Bregman, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, USSR Comput. Math. Math. Phys. **7**, 200–217 (1967).

[63] S. Kullback and R. A. Leibler, On information and sufficiency, Ann. Math. Stat. **22**, 79–86 (1951).

[64] G. Boole, *The Mathematical Analysis of Logic* (Cambridge University Press, 1847).

[65] Y. K. Cheung, S. Leonardos, and G. Piliouras, Learning in markets: Greed leads to chaos but following the price is right, in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, edited by Z.-H. Zhou (International Joint Conferences on Artificial Intelligence Organization, 2021) pp. 111–117.

[66] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019 (Association for Computing Machinery, New York, NY, USA, 2019) p. 217–228.

[67] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd, Quantum-inspired algorithms in practice, Quantum **4**, 307 (2020).

[68] E. Tang, Quantum principal component analysis only achieves an exponential speedup because of its state preparation assumptions, Phys. Rev. Lett. **127**, 060503 (2021).

[69] N.-H. Chia, A. P. Gilyén, T. Li, H.-H. Lin, E. Tang, and C. Wang, Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning, J. ACM **69** (2022).

[70] D. Coppersmith, *An approximate Fourier transform useful in quantum factoring*, Tech. Rep. (IBM Research Division, 1994).

[71] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, Amplitude estimation without phase estimation, Quantum Inf. Process. **19** (2020).

[72] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, Iterative quantum amplitude estimation, npj Quantum Inf. **7** (2021).

[73] K. Nakaji, Faster amplitude estimation, Quantum Inf. Comput. **20**, 1109–1123 (2020).

[74] S. Aaronson and P. Rall, Quantum approximate counting, simplified, in *Symposium on Simplicity in Algorithms* (Society for Industrial and Applied Mathematics, 2020) p. 24–32.

[75] P. Rall and B. Fuller, Amplitude estimation from quantum signal processing, Quantum **7**, 937 (2023).

[76] F. Labib, B. D. Clader, N. Stamatopoulos, and W. J. Zeng, Quantum amplitude estimation from classical signal processing (2024), arXiv:2405.14697 [quant-ph] .

[77] T. Giurgica-Tiron, I. Kerenidis, F. Labib, A. Prakash, and W. Zeng, Low depth algorithms for quantum amplitude estimation, Quantum **6**, 745 (2022).

[78] D.-L. Vu, B. Cheng, and P. Rebentrost, Low depth amplitude estimation without really trying (2024), arXiv:2410.01173 [quant-ph] .

[79] E. T. Campbell, Early fault-tolerant simulations of the Hubbard model, Quantum Sci. Technol. **7**, 015007 (2021).

[80] A. Katabarwa, K. Gratsea, A. Caesura, and P. D. Johnson, Early fault-tolerant quantum computing, PRX Quantum **5**, 020101 (2024).

[81] S. Arora, E. Hazan, and S. Kale, Fast algorithms for approximate semidefinite programming using the multiplicative weights update method, in *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)* (2005) pp. 339–348.

[82] S. Arora, E. Hazan, and S. Kale, The multiplicative weights update method: a meta-algorithm and applications, Theory Comput. **8**, 121–164 (2012).

14

[83] R. Jain, G. Piliouras, and R. Sim, Matrix multiplicative weights updates in quantum zero-sum games: Conservation laws & recurrence, in *Advances in Neural Information Processing Systems*, Vol. 35, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Curran Associates, Inc., 2022) pp. 4123–4135.

[84] F. G.S L. Brandão, R. Kueng, and D. Stilck França, Faster quantum and classical SDP approximations for quadratic binary optimization, Quantum **6**, 625 (2022).

[85] D. Lim and P. Rebentrost, A quantum online portfolio optimization algorithm, Quantum Inf. Process. **23** (2024).

[86] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM J. Comput. **46**, 1920–1950 (2017).

[87] I. Kerenidis and A. Prakash, A quantum interior point method for LPs and SDPs, ACM Trans. Quantum Comput. **1** (2020).

[88] M. Frank and P. Wolfe, An algorithm for quadratic programming, Nav. Res. Logist. Q. **3**, 95–110 (1956).

[89] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, Efficient projections onto the ℓ_1-ball for learning in high dimensions, in *Proceedings of the 25th International Conference on Machine Learning*, ICML '08 (Association for Computing Machinery, New York, NY, USA, 2008) p. 272–279.

[90] L. Condat, Fast projection onto the simplex and the ℓ_1 ball, Math. Program. **158**, 575–585 (2015).

[91] Y. Du, M.-H. Hsieh, T. Liu, S. You, and D. Tao, Quantum differentially private sparse regression learning, IEEE Trans. Inf. Theory **68**, 5217–5233 (2022).

[92] Y. Chen and R. de Wolf, Quantum algorithms and lower bounds for linear regression with norm constraints, in *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 261, edited by K. Etessami, U. Feige, and G. Puppis (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2023) pp. 38:1–38:21.

[93] K. L. Clarkson, Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm, ACM Trans. Algorithms **6** (2010).

[94] M. Jaggi, Revisiting Frank-Wolfe: Projection-free sparse convex optimization, in *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 28, edited by S. Dasgupta and D. McAllester (PMLR, Atlanta, Georgia, USA, 2013) pp. 427–435.

[95] A. Garg, R. Kothari, P. Netrapalli, and S. Sherif, No quantum speedup over gradient descent for non-smooth convex optimization, in *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 185, edited by J. R. Lee (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2021) pp. 53:1–53:20.

[96] A. Garg, R. Kothari, P. Netrapalli, and S. Sherif, Near-optimal lower bounds for convex optimization for all orders of smoothness, in *Advances in Neural Information Processing Systems*, Vol. 34, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Curran Associates, Inc., 2021) pp. 29874–29884.

[97] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani, Random generation of combinatorial structures from a uniform distribution, Theor. Comput. Sci. **43**, 169–188 (1986).

[98] C. Durr and P. Høyer, A quantum algorithm for finding the minimum (1996), arXiv:quant-ph/9607014 [quant-ph] .

[99] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems*, Vol. 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019).

15

A Quantum subroutines

In this section, we show prior results that obtain ℓ_1 norms and inner products with quadratic speedups with QAE.

Lemma A.1 (Quantum state preparation and norm estimation; Lemma 5, [35]). *Let $n \in \mathbb{N}$. We are given quantum query access to non-zero vector $w \in \mathbb{I}^n$, with $\max_j w_j = 1$.*

1. *There exists a quantum circuit that prepares the state $\frac{1}{\sqrt{n}} \sum_{j=1}^n |j\rangle \left(\sqrt{w_j} |0\rangle + \sqrt{1 - w_j} |1\rangle \right)$ with two queries and $\mathcal{O}(\log n)$ gates.*

2. *Let $\varepsilon > 0$ and $\delta \in (0, 1)$. There exists a quantum algorithm that provides an estimate Γ_w of the ℓ_1-norm $\|w\|_1$ such that $|\|w\|_1 - \Gamma_w| \leq \varepsilon \|w\|_1$, with probability at least $1 - \delta$. The algorithm requires $\mathcal{O}\left(\frac{\sqrt{n}}{\varepsilon} \log \frac{1}{\delta} \right)$ queries and $\tilde{\mathcal{O}}\left(\frac{\sqrt{n}}{\varepsilon} \log \frac{1}{\delta} \right)$ quantum gates.*

Proof. We reiterate the proof of Lemma 5 in [35] for the convenience of the reader.

1. First, using $\mathcal{O}(\log n)$ Hadamard gates, prepare the state $\frac{1}{\sqrt{n}} \sum_{j=1}^n |j\rangle$. Then, by quantum query access to w, obtain $\frac{1}{\sqrt{n}} \sum_{j=1}^n |j\rangle |w_j\rangle$. By controlled rotation gates, we can then obtain $\frac{1}{\sqrt{n}} \sum_{j=1}^n |j\rangle |w_j\rangle \left(\sqrt{w_j} |0\rangle + \sqrt{1 - w_j} |1\rangle \right)$. By another quantum query access to w, we can uncompute the intermediate registers and obtain $\frac{1}{\sqrt{n}} \sum_{j=1}^n |j\rangle \left(\sqrt{w_j} |0\rangle + \sqrt{1 - w_j} |1\rangle \right)$.

2. First observe that that with projector $P = I_n \otimes |0\rangle\langle 0|$ and $|\psi\rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^n |j\rangle (\sqrt{w_j} |0\rangle + \sqrt{1 - w_j} |1\rangle)$, one can obtain $a = \|P |\psi\rangle\|_2^2 = \frac{\|w\|_1}{n}$. Setting $M \geq \frac{6\pi}{\varepsilon} \sqrt{n}$, we obtain an estimate

$$|\tilde{a}_{\text{est}} - a| \leq 2\pi \frac{\sqrt{a(1-a)}}{M} + \frac{\pi^2}{M^2} \leq \frac{\varepsilon}{6\sqrt{n}} \left(2\sqrt{a} + \frac{\varepsilon}{12} \right) \leq \frac{3\sqrt{a}\varepsilon}{6\sqrt{n}} \leq \frac{\sqrt{\|w\|_1} \cdot \varepsilon}{2n} \leq \frac{a}{2} \cdot \varepsilon \quad \text{(A.1)}$$

with probability at least $\frac{8}{\pi^2}$. Using the powering lemma [97], we can boost the success probability to $1 - \delta$ by taking the median of $\mathcal{O}(\log \frac{1}{\delta})$ runs of the QAE algorithm.

$\square$

Remark A.1.1. Note that Lemma A.1 has the requirement that $\max_j w_j = 1$. For cases where this is not the case, we can use a maximum finding algorithm to divide all entries by the largest value. Such can be achieved by the following quantum minimum/maximum finding algorithm in $\mathcal{O}(\sqrt{n})$ runtime, which we introduce below. Recall that division takes $\mathcal{O}(1)$ runtime with quantum arithmetic circuits.

Lemma A.2 (Quantum minimum finding; Theorem 1, [98]). *Let $n \in \mathbb{N}$. Given quantum query access to non-zero vector $w \in \mathbb{I}^n$, we can find the minimum $w_{\min} = \min_{j \in [n]} w_j$ with success probability $1 - \delta$ with $\mathcal{O}\left(\sqrt{n} \log \frac{1}{\delta} \right)$ queries and $\tilde{\mathcal{O}}\left(\sqrt{n} \log \frac{1}{\delta} \right)$ quantum gates.*

Corollary A.2.1 (Quantum maximum finding). *Let $n \in \mathbb{N}$. Given quantum query access to non-zero vector $w \in \mathbb{I}^n$, we can find the maximum $w_{\max} = \max_{j \in [n]} w_j$ with success probability $1 - \delta$ with $\mathcal{O}\left(\sqrt{n} \log \frac{1}{\delta} \right)$ queries and $\tilde{\mathcal{O}}\left(\sqrt{n} \log \frac{1}{\delta} \right)$ quantum gates.*

Below we present a quantum inner product estimation algorithm simplified from Lemma 6 of [35].

Lemma A.3 (Quantum inner product estimation with relative accuracy). *Let $n \in \mathbb{N}$, $\varepsilon < 0$ and $\delta \in (0, 1)$. We are given quantum query access to two vectors $u, v \in \mathbb{I}^n$. An estimate $\Gamma_{u,v}$ for the inner product can be provided such that $|\Gamma_{u,v} - u \cdot v| \leq \varepsilon \, u \cdot v$ with success probability $1 - \delta$. This estimate is obtained with $\mathcal{O}\left(\frac{\sqrt{n}}{\varepsilon} \log \frac{1}{\delta} \right)$ queries and $\tilde{\mathcal{O}}\left(\frac{\sqrt{n}}{\varepsilon} \log \frac{1}{\delta} \right)$ quantum gates.*

Proof. Using quantum arithmetic circuits, we can obtain $z_j = u_j v_j$, i.e., $z = u \odot v$, by the following:

$$|j\rangle \to |j\rangle |u_j\rangle |v_j\rangle \to |j\rangle |z_j\rangle |v_j\rangle \to |j\rangle |z_j\rangle |\bar{0}\rangle \quad \text{(A.2)}$$

Using quantum maximum finding in Corollary A.2.1 to find $z_{\max}$ up to probability $1 - \delta/2$, we can then obtain $z_j / z_{\max}$. Lastly, using Lemma A.1, we can obtain $\Gamma_{u,v} = \Gamma_z$ such that $|\Gamma_{u,v} - u \cdot v| = |\Gamma_z - \|z\|_1| \leq \varepsilon u \cdot v$ up to probability $1 - \delta/2$. Using a union bound [64], we find the total success probability of the entire process is $1 - \delta$. $\square$

16

B Convergence guarantees for the PR dynamics

We show the convergence guarantee of the proportional response (PR) dynamics in regards to the Eisenberg-Gale convex program by Zhang [25] and improved upon by Birnbaum *et al.* [59], and the convergence in regards to the Shmyrev convex program, first shown also by Birnbaum *et al.* [59]. Recall that the negative target function from the Eisenberg-Gale convex program is

$$\Phi(b) = -\sum_{i\in[n]} B_i \log u_i, \tag{B.1}$$

and the negative target function from the Shmyrev convex program is

$$\Psi(b) = -\sum_{i\in[n],j\in[m]} b_{ij} \log \frac{v_{ij}}{p_j}. \tag{B.2}$$

We first set up the following convex set:

$$\mathcal{B} = \left\{ b \in \mathcal{M}_{n\times m}(\mathbb{R}_+) : \sum_j b_{ij} = B_i \right\} \tag{B.3}$$

To show the convergence of the PR dynamics, we first need the following inequalities:

Lemma B.1. *Let* $b^* = \arg\min_{b\in\mathcal{B}} \Phi(b)$. *Then* $\Phi(b^*) = \Psi(b^*) - \sum_{i\in[n]} B_i \log B_i$.

Proof. Recall that the price p_j is the dual variable of the constraint on x_{ij} for the Eisenberg-Gale convex program, and that $u_i = \sum_j x_{ij} v_{ij}$. By the KKT stationarity constraint, we see that

$$\frac{\partial}{\partial x_{ij}} \left(-\sum_i B_i \log u_i + \sum_j p_j \left(\sum_i x_{ij} - 1 \right) \right) = -\frac{B_i v_{ij}}{u_i} + p_j = 0 \tag{B.4}$$

from which we can infer that $\frac{B_i v_{ij}}{u_i^*} = p_j^*$ and show $\Phi(b^*) = \Psi(b^*) - \sum_{i\in[n]} B_i \log B_i$. We have that,

$$\Phi(b^*) = -\sum_{i\in[n]} B_i \log u_i^* = -\sum_{i\in[n],j\in[m]} b_{ij}^* \log u_i^* = -\sum_{i\in[n],j\in[m]} b_{ij}^* \log B_i \frac{v_{ij}}{p_j^*} \tag{B.5}$$

$$= -\sum_{i\in[n],j\in[m]} b_{ij}^* \log \frac{v_{ij}}{p_j^*} - \sum_{i\in[n]} B_i \log B_i = \Psi(b^*) - \sum_{i\in[n]} B_i \log B_i. \tag{B.6}$$

$\square$

Lemma B.2 (Lemma 19, [59]). $\forall b \in \mathcal{B}, \Phi(b) \leq \Psi(b) - \sum_{i\in[n]} B_i \log B_i$.

Proof. We reiterate the proof of Lemma 19 in [59] for the convenience of the reader. By convexity of $-\log$, we see that,

$$\Phi(b) = -\sum_{i\in[n]} B_i \log u_i = -\sum_{i\in[n]} B_i \log \sum_{j\in[m]} \frac{b_{ij}}{p_j} v_{ij} \tag{B.7}$$

$$= -\sum_{i\in[n]} B_i \log \sum_{j\in[m]} \frac{b_{ij}}{B_i} \frac{v_{ij}}{p_j} - \sum_{i\in[n]} B_i \log B_i \tag{B.8}$$

$$\leq -\sum_{i\in[n],j\in[m]} \frac{b_{ij}}{B_i} B_i \log \frac{v_{ij}}{p_j} - \sum_{i\in[n]} B_i \log B_i \tag{B.9}$$

$$= \Psi(b) - \sum_{i\in[n]} B_i \log B_i. \tag{B.10}$$

$\square$

17

Lemma B.3. *Let $b'_{ij} = B_i \dfrac{b_{ij}v_{ij}/p_j}{u_i}$. Then $\forall b \in \mathcal{B}, \Psi(b') \leq \Phi(b) + \sum_{i \in [n]} B_i \log B_i$.*

Proof. Let $p'_j = \sum_i b'_{ij}$. By concavity of log:

$$\Psi(b') = \sum_{i \in [n], j \in [m]} b'_{ij} \log \frac{p'_j}{v_{ij}} = \sum_{i \in [n], j \in [m]} B_i \frac{b_{ij}v_{ij}/p_j}{u_i} \log \frac{p'_j}{v_{ij}} \tag{B.11}$$

$$\leq \sum_{i \in [n]} B_i \log \sum_{j \in [m]} \frac{b_{ij}v_{ij}/p_j}{u_i} \frac{p'_j}{v_{ij}} = \sum_{i \in [n]} B_i \log \left(\frac{1}{u_i} \sum_{j \in [m]} \frac{b_{ij}p'_j}{p_j} \right) \tag{B.12}$$

$$= \sum_{i \in [n]} B_i \log \sum_{j \in [m]} \frac{b_{ij}p'_j}{B_i p_j} - \sum_{i \in [n]} B_i \log u_i + \sum_{i \in [n]} B_i \log B_i \tag{B.13}$$

$$\leq \log \sum_{i \in [n], j \in [m]} \frac{b_{ij}p'_j}{p_j} - \sum_{i \in [n]} B_i \log u_i + \sum_{i \in [n]} B_i \log B_i \tag{B.14}$$

$$= \log \sum_{j \in [m]} p'_j - \sum_{i \in [n]} B_i \log u_i + \sum_{i \in [n]} B_i \log B_i \tag{B.15}$$

$$= \Phi(b) + \sum_{i \in [n]} B_i \log B_i. \tag{B.16}$$

$\square$

From the above two lemmas, we gain the monotonically decreasing properties of iteratively updating b via the PR dynamics on the negative target functions of the Eisenberg-Gale and Shmyrev convex programs:

Lemma B.4. $\forall t \geq 0, \Phi(b^{(t+1)}) \leq \Phi(b^{(t)})$.

Proof. Apply Lemma B.2 and Lemma B.3 consequently. $\square$

Corollary B.4.1 (Lemma 5, [59]). $\forall t \geq 0, \Psi(b^{(t+1)}) \leq \Psi(b^{(t)})$.

We now use the following lemmas to construct an end-to-end proof of the convergence of the PR dynamics. In a slight abuse of notation, we adapt the definition of KL divergence to matrices such that for $u, v \in \mathcal{M}(\mathbb{R}_+)_{p \times q}$, let $D(u\|v) := \sum_{i \in [p], j \in [q]} u_{ij} \log \frac{u_{ij}}{v_{ij}}$. The following can then be shown:

Lemma B.5. $\forall t \geq 0, \sum_{t=0}^{T} \left(\Phi(b^{(t)}) - \Phi(b^*) \right) \leq D(b^*\|b^{(0)})$.

Proof. Similar by the proof of Theorem 3 of [25], we first lower bound $\Delta_t = D(b^*\|b^{(t)}) - D(b^*\|b^{(t+1)})$ as follows:

$$\Delta_t = D(b^*\|b^{(t)}) - D(b^*\|b^{(t+1)}) \tag{B.17}$$

$$= \sum_{i \in [n], j \in [m]} b^*_{ij} \log \frac{b^{(t+1)}_{ij}}{b^{(t)}_{ij}} = \sum_{i \in [n], j \in [m]} b^*_{ij} \log \frac{B_i v_{ij}}{p^{(t)}_j u^{(t)}_i} \tag{B.18}$$

$$= \sum_{i \in [n], j \in [m]} \left(b^*_{ij} \log \frac{v_{ij}}{p^*_j} + b^*_{ij} \log \frac{p^*_j}{p^{(t)}_j} - b^*_{ij} \log u^{(t)}_i + b^*_{ij} \log B_i \right) \tag{B.19}$$

$$= \sum_{i \in [n], j \in [m]} b^*_{ij} \log \frac{v_{ij}}{p^*_j} + \sum_{j \in [m]} p^*_j \log \frac{p^*_j}{p^{(t)}_j} - \sum_{i \in [n]} B_i \log u^{(t)}_i + \sum_{i \in [n]} B_i \log B_i \tag{B.20}$$

$$= -\Psi(b^*) + D(p^*_j\|p^{(t)}_j) + \Phi(b^{(t)}) + \sum_{i \in [n]} B_i \log B_i \tag{B.21}$$

$$= D(p^*_j\|p^{(t)}_j) + \Phi(b^{(t)}) - \Phi(b^*) \tag{B.22}$$

18

$$\geq \Phi(b^{(t)}) - \Phi(b^*) \tag{B.23}$$

where the second-to-last equality is by Lemma B.1 and the inequality is by the positivity of KL divergence. Taking the telescoping sum of Δ_t, we see that

$$\sum_{t=0}^{T} \Delta_t = \sum_{t=0}^{T} D(b^* \| b^{(t)}) - D(b^* \| b^{(t+1)}) = D(b^* \| b^{(0)}) - D(b^* \| b^{(T+1)}) \geq \sum_{t=0}^{T} (\Phi(b^{(t)}) - \Phi(b^*))$$
$$\tag{B.24}$$

Hence, we obtain $\sum_{t=0}^{T} \left(\Phi(b^{(t)}) - \Phi(b^*) \right) \leq D(b^* \| b^{(0)})$. □

Proposition B.6. $\forall t \geq 0, \Phi(b^{(T-1)}) - \Phi(b^*) \leq \frac{D(b^{(0)} \| b^*)}{T}$.

Proof. Combining Lemma B.4 and Lemma B.5, we can write

$$\Phi(b^{(T-1)}) - \Phi(b^*) \leq \frac{1}{T} \sum_{t=0}^{T-1} \Phi(b^{(t)}) - \Phi(b^*) \leq \frac{D(b^* \| b^{(0)})}{T}. \tag{B.25}$$

□

Corollary B.6.1 (Lemma 3, [59]). $\forall t \geq 0, \Psi(b^{(T)}) - \Psi(b^*) \leq \frac{D(b^{(0)} \| b^*)}{T}$.

Proof. Apply Lemma B.3 to Proposition B.6. □

Lastly, we can upper bound the value $D(b^* \| b^{(0)})$ in terms of dimensions m and n given that each buyer initially divides the budget equally between all items such that $b_{ij}^{(0)} = \frac{B_i}{m}$.

Lemma B.7 (Lemma 13, [59]; Theorem 7, [26]). *If $b_{ij}^{(0)} = \frac{B_i}{m}$ for all i and j, then $D(b^* \| b^{(0)}) \leq \log m$.*

Proof. Evaluating $D(b^* \| b^{(0)})$, we have

$$D(b^* \| b^{(0)}) = \sum_{ij} b_{ij}^* \log \frac{b_{ij}^*}{b_{ij}^{(0)}} = \sum_{ij} b_{ij}^* \log \frac{m b_{ij}^*}{B_i} = \log m + \sum_{ij} b_{ij}^* \log \frac{b_{ij}^*}{B_i} \leq \log m, \quad \text{(B.26)}$$

as $b_{ij}^* \leq B$, and $\log \frac{b_{ij}^*}{B_i}$ is negative. □

Plugging Lemma B.7 into Proposition B.6 and Corollary B.6.1, we obtain the convergence guarantee of Theorem 2.

C Convergence guarantees for the FPR dynamics

In this section, we prove the convergence guarantee of the faulty proportional response (FPR) dynamics. We first examine the immediate effects of allowing erroneous estimations of u and p in the FPR dynamics. Let $\hat{B}_i^{(t)} = \sum_{j \in [m]} \hat{b}_{ij}^{(t)}$. Note that $\hat{B}_i^{(t)} \neq B_i$ as the normalization step of constructing $\hat{b}_{ij}$ is erroneous. By the construction of $\hat{b}^{(t)}$ by the FPR dynamics,

$$\hat{b}_{ij}^{(t)} = \frac{B_i}{\tilde{\nu}_i^{(t-1)}} v_{ij} \frac{\hat{b}_{ij}^{(t-1)}}{\tilde{p}_j^{(t-1)}}, \tag{C.1}$$

we can find that

$$\hat{B}_i^{(t)} = \sum_{j \in [m]} \hat{b}_{ij}^{(t)} = \sum_{j \in [m]} \frac{B_i}{\tilde{\nu}_i^{(t-1)}} v_{ij} \frac{\hat{b}_{ij}^{(t-1)}}{\tilde{p}_j^{(t-1)}} = \frac{B_i}{\tilde{\nu}_i^{(t-1)}} \sum_{j \in [m]} v_{ij} \frac{\hat{b}_{ij}^{(t-1)}}{\tilde{p}_j^{(t-1)}} = B_i \frac{\hat{\nu}_i^{(t-1)}}{\tilde{\nu}_i^{(t-1)}}, \tag{C.2}$$

where we can obtain the following inequality by definition of $\tilde{\nu}_i$:

$$\frac{B_i}{1 + \varepsilon_\nu} \leq \hat{B}_i^{(t)} \leq \frac{B_i}{1 - \varepsilon_\nu} \tag{C.3}$$

19

By summing $\hat{B}_i$, we find that

$$\frac{1}{1+\varepsilon_\nu} \leq \sum_{i \in [n]} \hat{B}_i^{(t)} = \sum_{i \in [n], j \in [m]} \hat{b}_{ij}^{(t)} = \sum_{j \in [m]} \hat{p}_j^{(t)} \leq \frac{1}{1-\varepsilon_\nu}. \tag{C.4}$$

We now prove Theorem 3.

Theorem 3 (Convergence of the FPR dynamics). *Considering a linear Fisher market, for $b_{ij}^{(t)}$ as iteratively defined by the faulty proportional response dynamics where $\hat{b}_{ij}^{(0)} = \frac{B_i}{m}$, we have*

$$\min_{t \in [T]} \Phi(\hat{b}^{(t)}) - \Phi(b^*) \leq \frac{2 \log m}{T}$$

when $\varepsilon_\nu \leq \frac{\log m}{8T}$ and $\varepsilon_p \leq \frac{\log m}{6T}$.

Proof. Similar to the proof of Lemma B.5, we first lower bound $\hat{\Delta}_t = \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{b_{ij}^*}{\hat{b}_{ij}^{(t)}} - \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{b_{ij}^*}{\hat{b}_{ij}^{(t+1)}}$, where we use as follows:

$$\hat{\Delta}_t = \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{\hat{b}_{ij}^{(t+1)}}{\hat{b}_{ij}^{(t)}} = \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{B_i v_{ij}}{\tilde{p}_j^{(t)} \tilde{\nu}_i^{(t)}} \tag{C.5}$$

$$= \sum_{i \in [n], j \in [m]} \left(b_{ij}^* \log \frac{v_{ij}}{p_j^*} + b_{ij}^* \log \frac{p_j^*}{\tilde{p}_j^{(t)}} - b_{ij}^* \log \tilde{\nu}_i^{(t)} + b_{ij}^* \log B_i \right) \tag{C.6}$$

$$= \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{v_{ij}}{p_j^*} + \sum_{j \in [m]} p_j^* \log \frac{p_j^*}{\tilde{p}_j^{(t)}} - \sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)} + \sum_{i \in [n]} B_i \log B_i \tag{C.7}$$

$$= -\Phi(b^*) + \sum_{j \in [m]} p_j^* \log \frac{p_j^*}{\tilde{p}_j^{(t)}} - \sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)}. \tag{C.8}$$

We now lower bound the second and third terms from the above individually as follows. Starting with the second term,

$$\sum_{j \in [m]} p_j^* \log \frac{p_j^*}{\tilde{p}_j^{(t)}} = \sum_{j \in [m]} p_j^* \log \frac{p_j^*}{\hat{p}_j^{(t)} / \sum_{j' \in [m]} \hat{p}_{j'}^{(t)}} + \sum_{j \in [m]} p_j^* \log \frac{\hat{p}_j^{(t)}}{\tilde{p}_j^{(t)}} - \sum_{j \in [m]} p_j^* \log \sum_{j' \in [m]} \hat{p}_{j'}^{(t)} \tag{C.9}$$

$$= D\left(p_j^* \middle\| \frac{\hat{p}_j^{(t)}}{\sum_{j' \in [m]} \hat{p}_{j'}^{(t)}} \right) + \sum_{j \in [m]} p_j^* \log \frac{\hat{p}_j^{(t)}}{\tilde{p}_j^{(t)}} - \log \sum_{j \in [m]} \hat{p}_j^{(t)} \tag{C.10}$$

$$\geq D\left(p_j^* \middle\| \frac{\hat{p}_j^{(t)}}{\sum_{j' \in [m]} \hat{p}_{j'}^{(t)}} \right) + \sum_{j \in [m]} p_j^* \log \frac{1}{1+\varepsilon_p} - \log \frac{1}{1-\varepsilon_\nu} \tag{C.11}$$

$$\geq D\left(p_j^* \middle\| \frac{\hat{p}_j^{(t)}}{\sum_{j' \in [m]} \hat{p}_{j'}^{(t)}} \right) - \varepsilon_p - 2\varepsilon_\nu \geq -\varepsilon_p - 2\varepsilon_\nu \tag{C.12}$$

Moving on the the third term,

$$-\sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)} = -\sum_{i \in [n]} B_i \log \hat{\nu}_i^{(t)} - \sum_{i \in [n]} B_i \log \frac{\tilde{\nu}_i^{(t)}}{\hat{\nu}_i^{(t)}} \tag{C.13}$$

$$\geq -\sum_{i \in [n]} B_i \log \hat{\nu}_i^{(t)} - \sum_{i \in [n]} B_i \log(1 + \varepsilon_\nu) \tag{C.14}$$

$$= -\sum_{i \in [n]} B_i \log \sum_{j \in [m]} \frac{\hat{p}_j^{(t)}}{\tilde{p}_j^{(t)}} \frac{v_{ij} \hat{b}_{ij}^{(t)}}{\hat{p}_j^{(t)}} - \log(1 + \varepsilon_\nu) \tag{C.15}$$

20

$$\geq -\sum_{i \in [n]} B_i \log \sum_{j \in [m]} \frac{1}{1 - \varepsilon_p} \frac{v_{ij} \hat{b}_{ij}^{(t)}}{\hat{p}_j^{(t)}} - \log(1 + \varepsilon_\nu) \tag{C.16}$$

$$= -\sum_{i \in [n]} B_i \log \sum_{j \in [m]} \frac{v_{ij} \hat{b}_{ij}^{(t)}}{\hat{p}_j^{(t)}} + \log(1 - \varepsilon_p) - \log(1 + \varepsilon_\nu) \tag{C.17}$$

$$\geq \Phi(\hat{b}^{(t)}) - 2\varepsilon_p - \varepsilon_\nu \tag{C.18}$$

Hence, in total, we find that

$$\hat{\Delta}_t = \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{\hat{b}_{ij}^{(t+1)}}{\hat{b}_{ij}^{(t)}} \geq \Phi(\hat{b}^{(t)}) - \Phi(b^*) - 3\varepsilon_p - 3\varepsilon_\nu. \tag{C.19}$$

Taking the telescoping sum of $\hat{\Delta}_t$, we see that

$$\sum_{t=0}^{T} \hat{\Delta}_t = \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{b_{ij}^*}{b_{ij}^{(0)}} - \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{b_{ij}^*}{\hat{b}_{ij}^{(t+1)}} \geq \sum_{t=0}^{T} \left(\Phi(\hat{b}^{(t)}) - \Phi(b^*) - 3\varepsilon_p - 3\varepsilon_\nu \right). \tag{C.20}$$

Taking the upper bound of $\sum_{t=0}^{T} \hat{\Delta}_t$, we obtain

$$\sum_{t=0}^{T} \hat{\Delta}_t = \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{b_{ij}^*}{b_{ij}^{(0)}} - \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{\hat{b}_{ij}^*}{\hat{b}_{ij}^{(t+1)}} \tag{C.21}$$

$$\leq \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{b_{ij}^*}{b_{ij}^{(0)}} - \sum_{i \in [n], j \in [m]} b_{ij}^* \log \frac{b_{ij}^*}{\hat{b}_{ij}^{(t+1)} / \sum_{i' \in [n], j' \in [m]} \hat{b}_{i'j'}^{(t+1)}} \tag{C.22}$$
$$+ \sum_{i \in [n], j \in [m]} b_{ij}^* \log \sum_{i' \in [n], j' \in [m]} \hat{b}_{i'j'}^{(t+1)}$$

$$= D(b^* \| b^{(0)}) - D\left(b^* \middle\| \frac{\hat{b}^{(t+1)}}{\sum_{i' \in [n], j' \in [m]} \hat{b}_{i'j'}^{(t+1)}} \right) + \log \sum_{i' \in [n], j' \in [m]} \hat{b}_{i'j'}^{(t+1)} \tag{C.23}$$

$$\leq D(b^* \| b^{(0)}) - D\left(b^* \middle\| \frac{\hat{b}^{(t+1)}}{\sum_{i' \in [n], j' \in [m]} \hat{b}_{i'j'}^{(t+1)}} \right) + \log(1 + \varepsilon_\nu) \tag{C.24}$$

$$\leq D(b^* \| b^{(0)}) + \varepsilon_\nu \tag{C.25}$$

Hence, we obtain $\sum_{t=0}^{T} \left(\Phi(\hat{b}^{(t)}) - \Phi(b^*) \right) \leq D(b^* \| b^{(0)}) + (3T + 4)\varepsilon_\nu + (3T + 3)\varepsilon_p$. Instead of T, we plug in $T - 1$ to obtain

$$\sum_{t=0}^{T-1} \left(\Phi(\hat{b}^{(t)}) - \Phi(b^*) \right) \leq D(b^* \| b^{(0)}) + (3T + 1)\varepsilon_\nu + 3T\varepsilon_p. \tag{C.26}$$

With a simple observation that

$$T \cdot \min_{t \in [T]} \left(\Phi(\hat{b}^{(t)}) - \Phi(b^*) \right) \leq \sum_{t=0}^{T-1} \left(\Phi(\hat{b}^{(t)}) - \Phi(b^*) \right), \tag{C.27}$$

we find

$$\min_{t \in [T]} \left(\Phi(\hat{b}^{(t)}) - \Phi(b^*) \right) \leq \frac{D(b^* \| b^{(0)})}{T} + 4\varepsilon_\nu + 3\varepsilon_p. \tag{C.28}$$

From Lemma B.7, we know that $D(b^* \| b^{(0)}) \leq \log m$. Then by setting $\varepsilon_\nu \leq \frac{\log m}{8T}$ and $\varepsilon_p \leq \frac{\log m}{6T}$, we obtain

$$\min_{t \in [T]} \Phi(\hat{b}^{(t)}) - \Phi(b^*) \leq \frac{2 \log m}{T}. \tag{C.29}$$

$\square$

21

Next, we prove Theorem 4.

Theorem 4. *Considering a linear Fisher market, for $b_{ij}^{(t)}$ as iteratively defined by the faulty proportional response dynamics where $\hat{b}_{ij}^{(0)} = \frac{B_i}{m}$. Let $t^* = \arg\max_{t \in [T]} \sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)}$. Then*

$$\Phi(\hat{b}^{(t^*)}) - \Phi(b^*) \leq \frac{2 \log m}{T}$$

when $\varepsilon_\nu \leq \frac{\log m}{8T}$ and $\varepsilon_p \leq \frac{\log m}{6T}$.

Proof. We slightly modify the proof of Theorem 3, and note that by Equation (C.8) and Equation (C.12), we have

$$\hat{\Delta}_t = -\Phi(b^*) + \sum_{j \in [m]} p_j^* \log \frac{p_j^*}{\tilde{p}_j^{(t)}} - \sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)} \quad \geq -\Phi(b^*) - \varepsilon_p - 2\varepsilon_\nu - \sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)} \tag{C.30}$$

Taking the telescoping sum and the upper bound from Equation (C.25), we obtain

$$\sum_{t=0}^{T-1} \left(-\sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)} - \Phi(b^*) \right) \leq \sum_{t=0}^{T-1} \hat{\Delta}_t + 2T\varepsilon_\nu + T\varepsilon_p \leq D(b^* \| b^{(0)}) + (2T+1)\varepsilon_\nu + T\varepsilon_p, \tag{C.31}$$

where we can note

$$\min_{t \in [T]} \left(-\sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)} - \Phi(b^*) \right) \leq \frac{D(b^* \| b^{(0)})}{T} + 3\varepsilon_\nu + \varepsilon_p. \tag{C.32}$$

Let $t^* = \arg\min_{t \in [T]} \left(-\sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)} - \Phi(b^*) \right)$. Then by Equation (C.18), we have the following:

$$-\sum_{i \in [n]} B_i \log \tilde{\nu}_i^{(t)} \geq \Phi(\hat{b}^{(t)}) - 2\varepsilon_p - \varepsilon_\nu \tag{C.33}$$

Then we can obtain

$$\Phi(\hat{b}^{(t^*)}) - \Phi(b^*) \leq \frac{D(b^* \| b^{(0)})}{T} + 4\varepsilon_\nu + 3\varepsilon_p. \tag{C.34}$$

Lastly by setting $\varepsilon_\nu \leq \frac{\log \nu}{8T}$ and $\varepsilon_p \leq \frac{\log m}{6T}$, we obtain

$$\Phi(\hat{b}^{(t^*)}) - \Phi(b^*) \leq \frac{2 \log m}{T}. \tag{C.35}$$

$\square$

D   Experimental and implementation details

Our experiments are conducted on a single NVIDIA P100 GPU and written with the `PyTorch` library [99]. The optimal objective value is approximately computed by taking the results of the 1000-th iteration of the PR dynamics.

For the projected gradient descent (PGD) algorithm, our implementation is unlike Gao and Kroer [26], whose task is based on the CEEI scenario where agents are given a unit of fake money and whose end goal is only the allocation. We require information on both the allocation x and price p, hence our algorithm output should be the bids b. Therefore, instead of formulating the problem after the EG objective function, we mirror[7] the PR dynamics in its equivalence to mirror descent [59] on the Shmyrev objective function and perform PGD on the latter (see Algorithm 2).

We formulate the Shmyrev objective function into the following form to obtain convergence guarantees and the step size:

$$f(x) = h(Ax) + \langle q, x \rangle \tag{D.1}$$

---

[7]Pun intended.

22

Algorithm 2: Projected Gradient Descent

Input: Budget B, Value v, Learning rate γ, Iterations T
Output: Bids b

1 $b_{ij}^{(0)} = \frac{B_i}{m}$
2 **for** $t = 0$ *to* T **do**
3 $\quad$ $r_{ij}^{(t)} = b_{ij}^{(t)} - \gamma \cdot (1 - \log v_{ij}/p_j^{(t)})$ // Gradient step
4 $\quad$ **for** $i = 0$ *to* n **do**
5 $\quad\quad$ $b_{i,*}^{(t+1)} = \mathrm{Proj}(r_{i,*}^{(t)} \to \{x \in \mathbb{R}_+^n, \sum_k x_k = B_i\})$
$\quad\quad$ // Projection step onto a B_i-simplex

6 **return** $b^{(T)}$

where $x \in \mathbb{R}^n, A \in \mathcal{M}_{d \times n}(\mathbb{R}), h : \mathbb{R}^d \to \mathbb{R}, q \in \mathbb{R}^n$. Considering a flattened vector of the bids b, we note that if

$$A = n \left\{ \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \otimes \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}}_{m} , \quad q = \begin{pmatrix} -\log v_{11} \\ -\log v_{12} \\ \vdots \\ -\log v_{mn} \end{pmatrix} , \quad h(x) = \sum_i x_i \log x_i \quad \text{(D.2)} \right.$$

then $f = \Psi$. Then by Theorem 3 of [26], by setting a learning rate of $\gamma = 1/L\|A\|^2$, where $L = 1/\min_{j,t} p_j^{(t)}$, we get linear convergence. Note that $\|A\|^2 = n$. Gao and Kroer [26] further provide a line search procedure to set the constant multiplier in the learning rate as well as provide sharper convergence guarantees, but as we only run for 16 iterations, we do not perform the line search and fix the learning rate to the initial learning rate that Gao and Kroer [26] use in their empirical studies, which is $1000/L\|A\|^2$.

For QAE, we set $M = \sqrt{T\sqrt{n}}/16 = 32$. We scale down M by the constant factor of 16 to save memory consumption on the GPU, as we simulate QAE by computing the full probability distribution over $[M]$. We compensate for the loss in accuracy of the estimation by employing the median-of-means estimator [60], where we take the median of 3 estimators constructed from the mean of 7 samples from the QAE subroutine. In addition, we perform the maximum finding classically by line search in our simulation as opposed to a randomized protocol in the quantum algorithm, whose effects are only in regard to ignoring the failure possibility.

23

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state our contribution in proposing a quantum algorithm that provides a polynomial speedup in comparison to classical algorithms.

   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the potential and possible limitations of further quantum speedups in the Discussion section. Further, we state additional assumptions that our quantum algorithm requires in the opening paragraph of Section 4 as well as in the theorem statements.

   Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

24

Answer: [Yes]

Justification: We formulate the assumptions required by our theorems directly in the theorem statement. Further implementation requirements such as QRAM are also stated and introduced in Sections 2 and 4. The complete proof is shown in Appendix B and C, with proof sketches and intuitions provided in the corresponding main text.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We state clearly in Section 5 how our input data is randomly generated and how exactly we implement amplitude estimation on classical computers with GPU. The algorithm itself is written as a step-by-step pseudocode on page 7. We further provide implementation details in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

25

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We attach the full simulation code in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We attach the full simulation code in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our experiments are run over 15 iterations to account for randomness of our algorithm. Error bars are present in the attached image as shading.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

26

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We state the simulation details with software and hardware stated in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The authors have read and noted that this research conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The authors have discussed the potential applications of this work in Internet-based large-scale markets and its various applications. The work is a theory paper on computing market equilibria with quantum computers. We believe there are no negative societal impacts arising from this work.

27

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The authors have properly cited its usage of the PyTorch library in Appendix D.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

28

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

29

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

30