Quantum algorithm for large-scale market equilibrium computation

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

[1] Centre for Quantum Technologies, National University of Singapore

[2] Department of Computer Science, National University of Singapore

Centre for Quantum Technologies

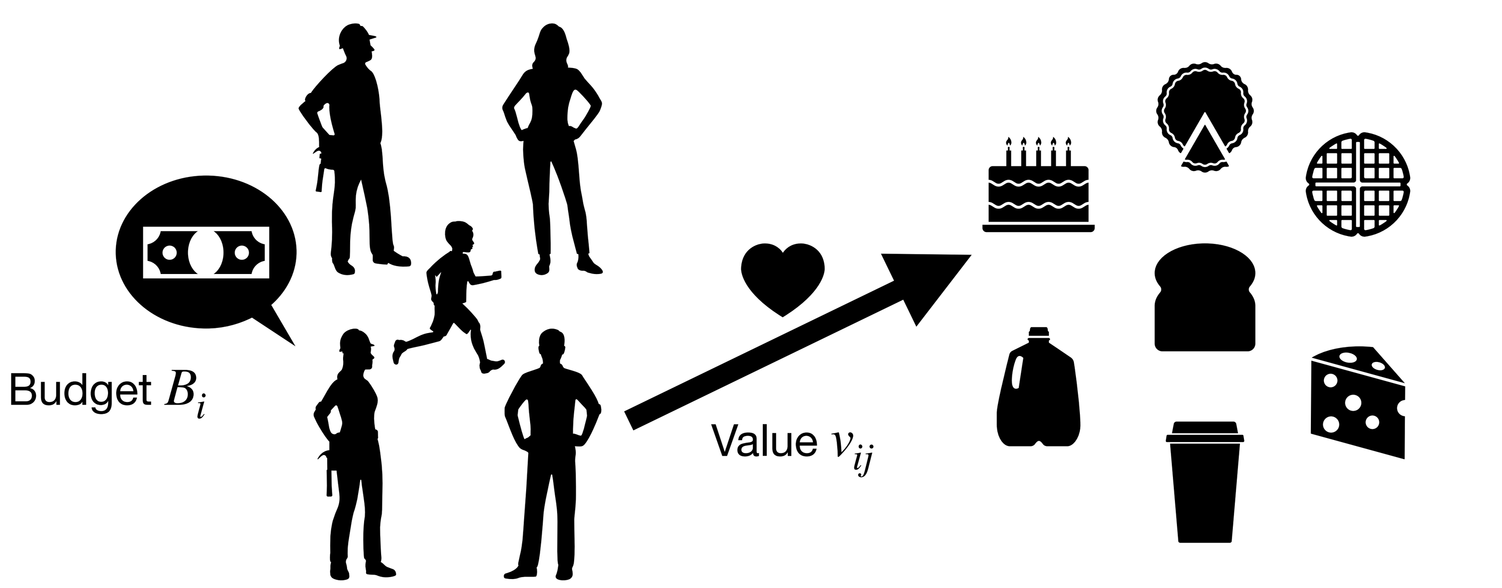
NUS Computing

NEURAL INFORMATION PROCESSING SYSTEMS

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.

Market equilibrium computation



n buyers

m infinitely divisible goods

Assumption:

Utility is linear.

$$u_i = \sum_j v_{ij} x_{ij}$$

Find

Allocation x_{ij} and Price p_j

OR

Bid b_{ij}

Main results

	PR dynamics [WZ07]	Our work
Iterations	$\frac{\log m}{\varepsilon}$	$\frac{2\log m}{\varepsilon}$
Runtime	$\tilde{\mathcal{O}}\left(\frac{mn}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(\frac{\sqrt{mn\max(m,n)}}{\varepsilon^2}\right)$
Memory	$\mathcal{O}(mn)$	$\mathcal{O}(m+n)$ *
Result Preparation	N/A, in RAM	QA: $\mathcal{O}\left(\text{poly}\log\frac{mn}{\varepsilon}\right)$ SA: $\tilde{\mathcal{O}}\left(\sqrt{mn}\right)$

Optimization-based solutions

Eisenberg-Gale convex program [EG59]:

$$\max_{x\geq 0} \sum_i B_i \log u_i$$

$$s.t. \sum_j v_{ij} x_{ij} = u_i$$

$$\sum_i x_{ij} = 1$$

Proportional response dynamics [WZ07]:

$$p_j^{(t)} = \sum_i b_{ij}^{(t)}, \qquad x_{ij}^{(t)} = \frac{b_{ij}^{(t)}}{p_j^{(t)}}$$

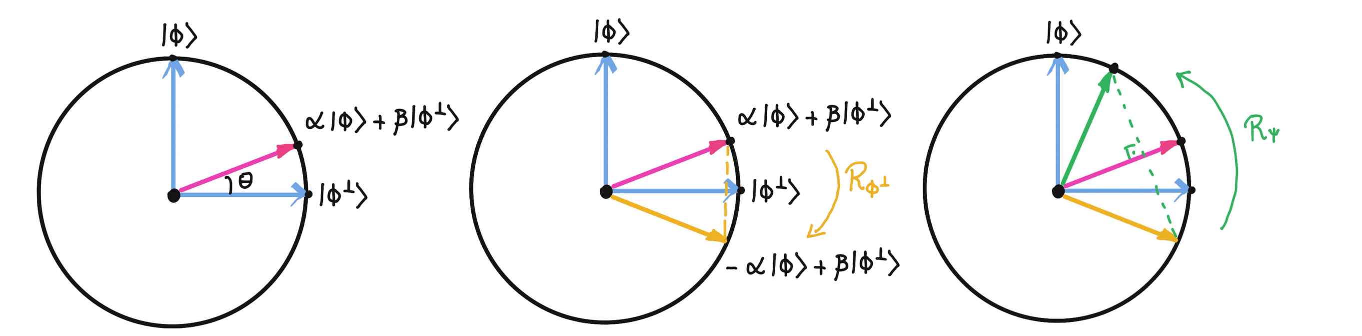
$$u_i^{(t)} = \sum_j v_{ij} x_{ij}^{(t)}, \qquad b_{ij}^{(t+1)} = B_i \frac{v_{ij} x_{ij}^{(t)}}{u_i^{(t)}}.$$

Convergence guarantee [BDX11]:

$$\Psi(b^{(T)}) - \Psi(b^*) \leq \frac{\log m}{T}$$

Source of quantum speedups

Quantum amplitude amplification and estimation (QAA and QAE) [BHMT02]



Used for fast ℓ_1 norm estimation and inner product estimation. [RHR+21]

Quantum data access and quantum memory

Quantum access of vector $w \in \mathbb{R}^n$

1. Quantum query access:

$$|\mathbf{j}\rangle|0\rangle \to |\mathbf{j}\rangle|w_j\rangle$$

2. Quantum sample access:

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

Quantum random access memory (QRAM)

Memory unit that allows quantum query/sample access to $w \in \mathbb{R}^n$.

Access time: $\mathcal{O}(\text{poly}\log n)$ One time construction cost: $\tilde{\mathcal{O}}(n)$

Faulty proportional response (FPR) dynamics

Modified updates:

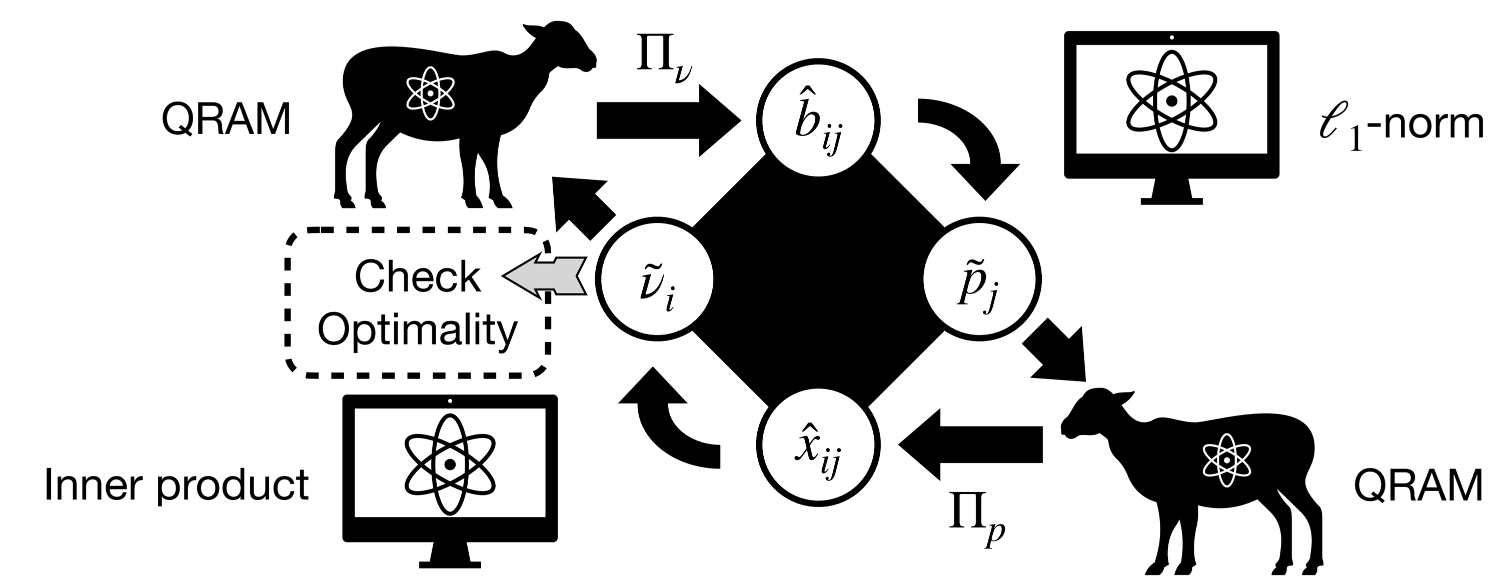
$$\tilde{p}_j^{(t)} = \left(\sum_i \hat{b}_{ij}^{(t)}\right)(1 \pm \varepsilon_p), \quad (\Pi_p^{(t)})_j = \prod_{k=0}^{t} \tilde{p}_j^{(k)}, \quad \hat{x}_{ij}^{(t)} = \frac{\hat{b}_{ij}^{(t)}}{\tilde{p}_j^{(t)}} = \frac{B_i^{t+1} v_{ij}^t}{m(\Pi_p^{(t)})_j(\Pi_\nu^{(t-1)})_i}$$

$$\tilde{\nu}_i^{(t)} = \left(\sum_j v_{ij}\hat{x}_{ij}^{(t)}\right)(1 \pm \varepsilon_\nu), \quad (\Pi_\nu^{(t)})_i = \prod_{k=0}^{t} \tilde{\nu}_i^{(k)}, \quad \hat{b}_{ij}^{(t+1)} = B_i \frac{v_{ij}\hat{x}_{ij}^{(t)}}{\tilde{\nu}_i^{(t)}} = \frac{B_i^{t+2} v_{ij}^{t+1}}{m(\Pi_p^{(t)})_j(\Pi_\nu^{(t)})_i}.$$

Convergence guarantee:

If $\varepsilon_p \leq \frac{\log m}{8T}, \varepsilon_\nu \leq \frac{\log m}{6T}, t^* = \arg\max_{t\in[T]} \sum_i B_i \tilde{\nu}_i^{(t)}$, then $\Psi(b^{(t^*)}) - \Psi(b^*) \leq \frac{2\log m}{T}$

Quantum algorithm



Algorithm 1: Quantum faulty proportional response dynamics

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

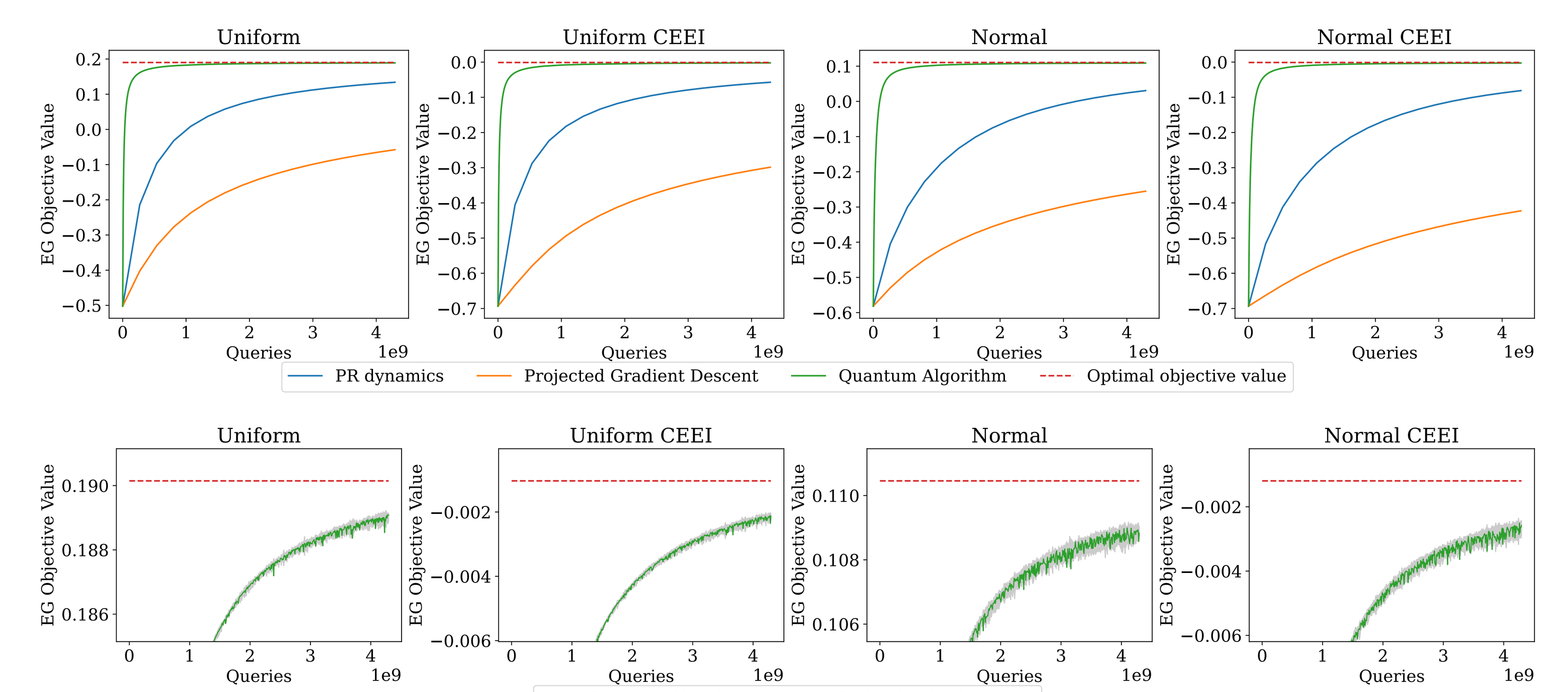
```
maxEGVal = -∞, b_ij^(0) = B_i / m
for t = 0 to T do
    for j = 0 to m do
        p̃_j^(t) = ‖b̂_{*,j}^(t)‖_1 (1 ± ε_p) via Q norm estimation with success prob. 1 − δ/(2mT)
    Store vector Π_p^(t) = p̃^(t) ⊙ Π_p^(t−1) into QRAM
    for i = 0 to n do
        ν̃_i^(t) = ⟨x_{i,*}^(t), v_{i,*}⟩(1 ± ε_ν) via Q inner product estimation with success prob. 1 − δ/(2nT)
    Store vector Π_ν^(t) = ν̃^(t) ⊙ Π_ν^(t−1) into QRAM
    Classically compute Φ̃^(t) = Σ_{i∈[n]} B_i log(ν̃_i^(t))
    if Φ̃^(t) > maxEGVal then
        maxEGVal = Φ̃^(t), bestPiP = Π_p^(t−1), bestPiNu = Π_ν^(t−1)
return bestPiP and bestPiNu in QRAM
```

Runtime analysis

ℓ_1-norm: $\mathcal{O}\left(\frac{\sqrt{n}}{\varepsilon_p}\log\frac{mT}{\delta}\right) \times mT$ Inner product: $\mathcal{O}\left(\frac{\sqrt{m}}{\varepsilon_\nu}\log\frac{nT}{\delta}\right) \times nT$

QRAM construction: $\tilde{\mathcal{O}}((m+n)T)$ Optimality check: $\mathcal{O}(nT)$

Simulation results



References

[EG59] Eisenberg and Gale, Ann. Math. Stat. 30, 165–168.

[BHMT02] Brassard, Høyer, Mosca and Tapp, in Quantum computation and information, 305, 53–74.

[WZ07] Wu and Zhang, in STOC'07, 354–363.

[BDX11] Birnbaum, Devanur, and Xiao, in EC'11, 127–136.

[RHR+21] Rebentrost, Hamoudi, Ray, Wang, Yang, and Santha, Phys. Rev. A 103, 012418.