Accelerating Voting by Quantum Computation (Supplementary Material)

Ao Liu[1] **Qishen Han**[1] **Lirong Xia**[1] **Nengkun Yu**[2]

[1]Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, USA
[2]Department of Computer Science, Stony Brook University, Stony Brook, NY, USA

The Appendix of UAI-23 Accepted Paper Accelerating Voting by Quantum Computation

A IMPLEMENTATION OF QUANTUM COUNTING ALGORITHM.

In this section, we aim to introduce the implementation of the quantum part of Algorithm 1 from a more technical perspective. We will first introduce the basics of quantum computing. Then we will specify the implementation of circuits of quantum counting in Algorithm 1, and why they accelerate the voting process.

A.1 QUANTUM BASICS.

Basic quantum computation. Quantum bit (or *qubit* in short) is the counterpart of classical *bit*, which takes a deterministic binary from $\{0,1\}$. Qubit, on the other hand, is represented by a linear combination of $\{|0\rangle, |1\rangle\}$, which are counterparts to $\{0,1\}$, respectively. That is, every qubit $|\psi\rangle$ is written as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where α and β are complex numbers and are usually called amplitudes. If we measure the qubit, there is $|\alpha|^2$ probability to get 0 and $|\beta|^2$ probability to get 1. Naturally, we always have $|\alpha|^2 + |\beta|^2 = 1$ because the probabilities should sum to 1. Qubits sometimes are written as vectors to simplify notations. Formally,

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \triangleq \alpha|0\rangle + \beta|1\rangle.$$

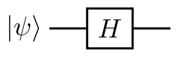
$t > 1$ qubits are presented as a 2^t-dimensional vector, where the j-th component of the vector (denoted as α_j) represents the amplitude of $|j_1 \cdots j_t\rangle$ (or $|j\rangle$), where $j_1 \cdots j_t$ is the binary representation of j. Similar to the 1-qubit case, the probability of observing $j_1, \cdots, j_t$ from those t qubit equals to $|\alpha_j|^2$.

A quantum operation (quantum gate) Q on t qubits is denoted by a $2^t \times 2^t$ unitary matrix, which means the matrix's inverse is its Hermitian conjugate. Applying a quantum operation Q on quantum state $|\psi\rangle$ is denoted by

$$Q|\psi\rangle \triangleq \boldsymbol{Q}_{(2^t\times 2^t)}\ \vec{\psi}_{(2^t)},$$

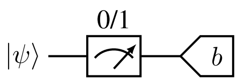
where the the quantum operator $\boldsymbol{Q}_{(2^t\times 2^t)}$ is a $2^t \times 2^t$ unitary matrix and the quantum state $\vec{\psi}_{(2^t)}$ is a 2^t dimensional column vector.

Quantum circuit of some useful quantum operators.[1] Quantum circuits run from the left-hand side to the right-hand side. For example, the following circuit means applying Hadamard gate H on a quantum state $|\psi\rangle$.



where $\boldsymbol{H} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$.

The quantum circuit notion



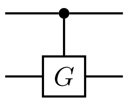
denotes measuring quantum state $|\psi\rangle$ with $0/1$ base (b denotes the result of measurement). Naturally, the complexity of quantum measurement and Hadamard gate are both $\Theta(1)$.

Quantum oracle [Berthiaume and Brassard, 1994, Van Dam, 1998, Kashefi et al., 2002] is a widely-used operator to encode binary functions or binary information. Given t qubits and a binary function $f : \{0, \cdots, 2^t - 1\} \mapsto \{0,1\}$, quantum oracle (based on function $f(\cdot)$) applies a phase shift of $-1 = e^{\pi i}$ if $f(x) = 1$ and does nothing otherwise. We can query oracle many times and regard the number of queries as the cost [Grover, 1996]. Formally,

$$\begin{cases} O_f|x\rangle = |x\rangle & \text{if } f(x) = 1\\ O_f|x\rangle = -|x\rangle & \text{otherwise}\end{cases}.$$

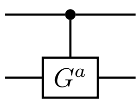
[1]All quantum circuits of this paper are drawn using the Quantikz package [Kay, 2018] for LaTeX.

Suppose we have a quantum gate G on t qubits. The following operation is called *controlled-G*.

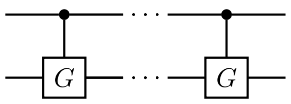


$$= \begin{bmatrix} \boldsymbol{I}_{(2^t\times 2^t)} & \mathbf{0}_{(2^t\times 2^t)} \\ \mathbf{0}_{(2^t\times 2^t)} & \boldsymbol{G}_{(2^t\times 2^t)} \end{bmatrix},$$

where $\boldsymbol{I}$ denotes the identity matrix, and $\mathbf{0}$ denotes the zeros matrix. To simplify notations, we also write



$=$



(repeat a times).

A.2 IMPLEMENTATION OF QUANTUM COUNTING CIRCUIT.

Figure 1 shows the quantum counting circuit, which is a combination of Grover search algorithm [Grover, 1996] and quantum reverse Fourier transformation (the $QFT^\dagger$ operator) Followings we focus on introducing Grover algorithm and why it accelerates the computation.

Grover operator. Grover algorithm is an efficient search algorithm. Given a binary function $f: \{0,1,\cdots,2^t-1\} \to \{0,1\}$, Grover algorithms returns an x with $f(x)=1$ with high probability. The Grover operation in Algorithm 1 is constructed by the quantum circuit in Figure 2, where $t = \lceil \log n \rceil$ denotes the minimum number of quantum bits to encode n. The quantum operator QPS is called quantum phase shifting, which provides a phase shift of -1 on every state except $|0\rangle$. Mathematically,

$$|0\rangle \xrightarrow{\text{QPS}} |0\rangle \qquad \text{and}$$
$$|x\rangle \xrightarrow{\text{QPS}} -|x\rangle \;\; \text{for any } x \in 1,\cdots,2^t-1.$$

Here, $|x\rangle$ represents the x-th base state of the t qubits. The high-level idea of Grover operator's functionality is shown in Figure 3, where $|\psi\rangle$ is the input of Grover operators in quantum counting, and $\{|\alpha\rangle, |\beta\rangle\}$ is a pair of orthogonal bases. The formal definition of $|\psi\rangle$, $|\alpha\rangle$, and $|\beta\rangle$ can be found in Appendix A.3. Under the $|\alpha\rangle$ $|\beta\rangle$ base, the quantum oracle O_{f_j} reflects $|\psi\rangle$ over $|\alpha\rangle$, while the rest parts of G reflects $O_{f_j}|\psi\rangle$ over $|\psi\rangle$. The angle between the output state $G|\psi\rangle$ and initial state $|\psi\rangle$

$$\theta = 2\arcsin\left(\sqrt{\mathbf{hist}_j \cdot 2^{-t}}\right),$$

which includes the information about $\mathbf{hist}_j$. Since function $\arcsin(\sqrt{x})$ grows quadratically faster than linear functions when x is small, we expect that an estimation about $\arcsin(\sqrt{x})$ could be quadratically more accurate than directly estimate x.

A.3 FUNCTIONALITY FOR GROVER ALGORITHM

According to (6.4) in Nielsen and Chuang [2010], Hadamard gate changes t qubits of $|0\rangle$ to an equal superposition state (equal probability of observing any outcome under quantum measurements).

$$|\psi\rangle = \frac{1}{2^{t/2}} \cdot \sum_{x=0}^{2^t-1} |x\rangle.$$

Letting $f: \{0,\cdots,2^t-1\} \mapsto \{0,1\}$ be the binary function to construct the quantum oracle, and $\mathring{n}_1$ be the number of x such that $f(x)=1$. The orthogonal bases $|\alpha\rangle$ and $|\beta\rangle$ are defined as,

$$|\alpha\rangle \triangleq \frac{1}{\sqrt{2^t - \mathring{n}_1}} \cdot \sum_{x:f(x)=0} |x\rangle \qquad \text{and}$$
$$|\beta\rangle \triangleq \frac{1}{\sqrt{\mathring{n}_1}} \cdot \sum_{x:f(x)=1} |x\rangle.$$

Under the $|\alpha\rangle$ $|\beta\rangle$ base, the equal superposition state

$$|\psi\rangle = \sqrt{\frac{2^t - \mathring{n}_1}{2^t}}\,|\alpha\rangle + \sqrt{\frac{\mathring{n}_1}{2^t}}\,|\beta\rangle.$$

Since

$$\theta = 2\arcsin\left(\sqrt{\mathring{n}_1 \cdot 2^{-t}}\right),$$

we have

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|\alpha\rangle + \sin\left(\frac{\theta}{2}\right)|\beta\rangle,$$
$$O_f|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|\alpha\rangle + \sin\left(-\frac{\theta}{2}\right)|\beta\rangle, \text{ and}$$
$$G|\psi\rangle = \cos\left(\frac{3\theta}{2}\right)|\alpha\rangle + \sin\left(\frac{3\theta}{2}\right)|\beta\rangle.$$

B MISSING PROOFS AND DISCUSSIONS

B.1 MISSING PROOF FOR LEMMA 3

Lemma 3. *Given* $\varepsilon \in (0, 0.5]$, *any fast (2-candidate) majority voting algorithm based on sampling with replacement requires at least* $\Omega\left(\frac{n^2\cdot\left(\frac{1}{2}-\varepsilon\right)^2}{\text{MoV}^2}\right)$ *runtime and at least* $\Omega\left(\log\left(\frac{n^2\cdot\left(\frac{1}{2}-\varepsilon\right)^2}{\text{MoV}^2}\right)\right)$ *space to achieve* $\Pr[\text{correct}] \geq 1-\varepsilon$.

Proof. For majority voting (when $m=2$), the corresponding profile with margin of victory MoV is

$$\begin{cases} n_{\text{win}} = (\lfloor n/2 \rfloor + \text{MoV}) \text{ votes for the winner} \\ n_{\text{lose}} = (\lceil n/2 \rceil - \text{MoV}) \text{ votes for the loser} \end{cases}. \quad (1)$$

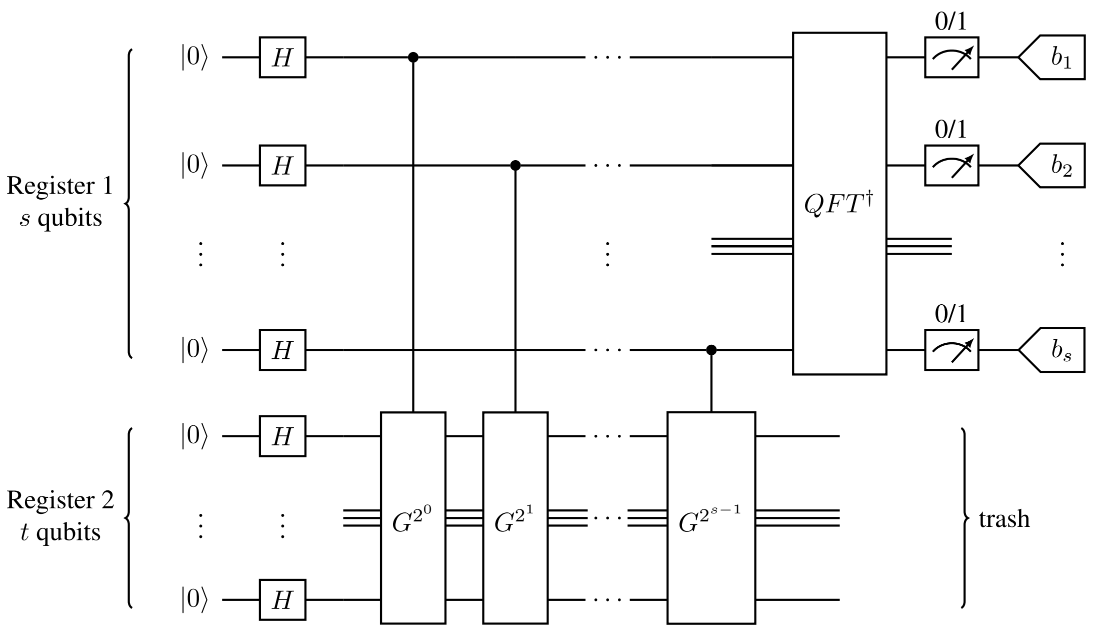


Figure 1: The circuit for quantum counting algorithm.

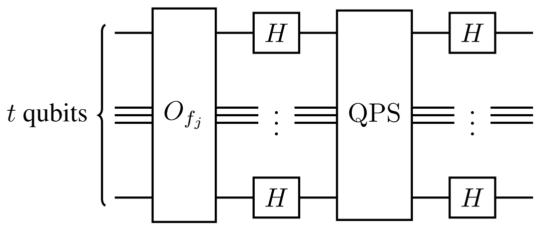


Figure 2: The circuit for Grover operator.

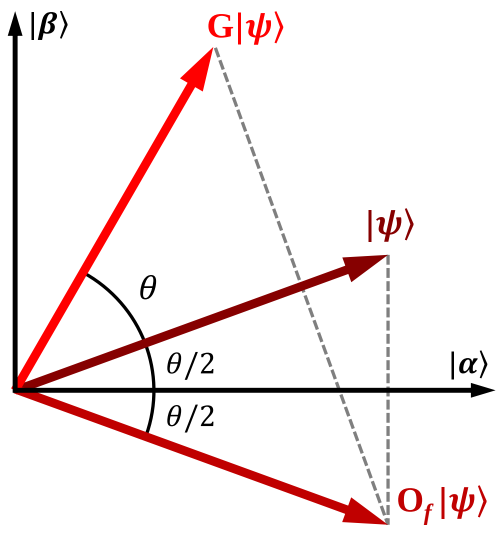


Figure 3: An illustration of Grover operator's functionality (Figure 6.3 in Nielsen and Chuang [2010]).

Figure 4 interprets the sampling (with replacement) process as a communication problem. We (the receiver) get a noisy data point about the winner from the sampling process. According to the above profile, we get the correct winner with $\frac{n_{\text{win}}}{n}$ probability and get the incorrect winner with $\frac{n_{\text{lose}}}{n} = 1 - \frac{n_{\text{win}}}{n}$ probability. This sampling process is equivalent to the noisy communication channel in Figure 4, which gives the correct binary message with $\frac{n_{\text{win}}}{n}$ probability.

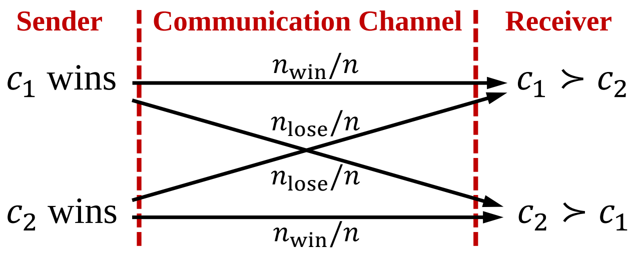


Figure 4: The communication channel presentation of sampling with replacement.

According to Equation (1.35) in MacKay [2003], the capacity of the above communication channel Cap $= 1 - H(n_{\text{win}}/n)$, where $H : (0,1) \to (0,1]$ denotes the binary entropy function. Mathematically,

$$H(p) \triangleq -p\log(p) - (1-p)\log(1-p).$$

Proposition 1 ($H(p)$'s Bounds, Theorem 1.2 in [Topsøe, 2001])**.** *Given any* $p \in (0,1)$,

$$4p(1-p) < H(p) < \big(4p(1-p)\big)^{1/\ln 4}.$$

With the lower bound in Proposition 1, we know the communication channel's capacity

$$\begin{aligned}\text{Cap} &= 1 - H(n_{\text{win}}/n) \leq 1 - H(1/2 + \text{MoV}/n) \\ &< 1 - 4 \cdot (1/2 - \text{MoV}/n) \cdot (1/2 + \text{MoV}/n) \\ &= 4\text{MoV}^2/n^2.\end{aligned}$$

The "$\leq$" follows by the monotonicity of $H(p)$. The next proposition (the well-known Shannon's theorem) connects the channel capacity with the error probability of binary information.

Proposition 2 ([Shannon, 1948]). *Given a communication channel with capacity* Cap*, reconstructing each single-bit message with error probability* $\varepsilon \in (0, 0.5]$ *requires receiving at least* $\frac{1-H(\varepsilon)}{\text{Cap}}$ *bits (in expectation) from the channel.*

By Proposition 2 and the upper bound in Proposition 1, the required number of bits from the channel (the required number of samples)

$$\begin{aligned}T &\geq \frac{1 - H(\varepsilon)}{\text{Cap}} > \frac{n^2 \cdot \left(1 - \left(4\varepsilon(1-\varepsilon)\right)^{1/\ln 4}\right)}{4\text{MoV}^2} \\ &= \Omega\left(\frac{n^2 \cdot \left(\frac{1}{2} - \varepsilon\right)^2}{\text{MoV}^2}\right).\end{aligned}$$

Lemma 3 follows by the observation that the time-complexity and the space-complexity of getting T samples are $\Omega(T)$ and $\Omega(\log T)$ respectively. $\square$

B.2 COMPARE SAMPLING WITH AND WITHOUT REPLACEMENT

Although Theorem 3 holds only for sampling with replacement algorithms, we believe that when the algorithm only uses the histogram of the sample votes to calculate the winner, and the sampled size T is small compared to n, then there is no major difference for sampling without replacement algorithms, because two samplings will converge to the same distribution when n goes to infinity.

Let **hist** be the histogram for a profile P, and $\textbf{hist}_j$ is the number of votes for j-th ranking in the profile. In the sampling with replacement, the number of votes for j-th ranking in the sample follows binomial distribution $\mathcal{B}(T, \textbf{hist}_j/n)$. For the sample without replacement, the number of votes for j-th ranking follows hypergeometric distribution $\mathcal{H}(n, \textbf{hist}_j, T)$. (A hypergeometric distribution $\mathcal{H}(n, \textbf{hist}_j, T)$ considers drawing T samples from n items, among which exactly $\textbf{hist}_j$ items have a specific feature, and characterizes the probability that a certain number of featured items is sampled.) The following proposition tells us that hypergeometric distribution $H(n, \textbf{hist}_j, T)$ converges to binomial distribution $\mathcal{B}(T, \textbf{hist}_j/n)$ when $n \to \infty$.

Theorem 1 (Corollary 4.1 in [Teerapabolarn and Wongkasem, 2011].). *Let X be a random variable that follows hypergeometric distribution $\mathcal{H}(n, \textbf{hist}_j, T)$, and Y be a random variable that follows binomial distribution $\mathcal{B}(T, \frac{\textbf{hist}_j}{n})$. For any $t \in \{0, \cdots, T\}$, fixed $p = \frac{\textbf{hist}_j}{n}$, and $T = o(\sqrt{n})$, $\lim_{n\to\infty} |P(X = t) - P(Y = t)| = 0$.*

Therefore, when n is large and sampling size T is small compared to n, the sample histograms will be close to each other between sampling with and without replacement.

C ADDITIONAL EXPERIMENTS

C.1 IMPLEMENTATION DETAILS

For the classical algorithm, we use MATLAB's built-in function `mnrnd` to draw samples for $\widehat{\textbf{hist}}$ (follows multi-nominal distribution). For the quantum algorithm, we first calculate the distribution of quantum counting according to (5.26) in Nielsen and Chuang [2010] and then draw samples from the calculated distribution. For all experiments of this paper, we use 10^5 independent trails to estimate Pr [correct]. All experiments of this paper are implemented through MATLAB 2022b and run on a Windows 11 desktop with AMD Ryzen 9 5900X CPU and 32GB RAM.

C.2 ADDITIONAL EXPERIMENTAL RESULTS

Plurality. For plurality, we use the following profile P,

$$\begin{cases} \frac{n+2(m!-1)\text{MoV}}{m!} \text{ votes for } c_1 \succ \cdots \succ c_m \\ \frac{n-2\text{MoV}}{m!} \text{ votes for each other type of votes} \end{cases}.$$

It's easy to check that the margin of victory of the above profile is MoV under plurality. Figure 6 plots the comparison between quantum-accelerated voting and classical voting for $m = 4$. Similar acceleration as $m = 2$ can be observed for $m = 4$.

We also observe that that $\Pr[\text{correct}]$ may not monotonically increase with the increase of $\log_2(K \cdot 2^s)$. *e.g.,* for Figure 5, $K = 1$, and MoV $= 256$, the $\Pr[\text{correct}]$ for $s = 15$ is smaller than $s = 14$. The non-monotonicity is not an uncommon phenomenon in quantum algorithms (*e.g.,* Kerenidis et al., 2019, Chen et al., 2020, Bausch, 2020). This phenomenon comes from the discrete manner of quantum noises, which differs from the noise in classical sampling. We also note that our theoretical analysis bounds the asymptotic manner of $\Pr[\text{correct}]$, instead of the monotonicity. To be slightly more technical, this decrease comes from the noise (*i.e.* tail probability) of the quantum counting, which is different from the classical counting noise. The tail probability of quantum counting also depends on the relative distance between the ground truth and its best s-bit estimation. The closer it is, the smaller the tail probability is.

The relative distance may not monotonically decrease with the increase of s. For example, assume the ground truth of ϕ is 0.0001 (in binary decimal). If using 1-bit estimation, the relative distance is $(0.0001 - 0.0)/0.1 = 1/8$. However, if using 3-bit estimation, the relative distance becomes $(0.0001 - 0.0)/0.001 = 1/2$, which is much larger than $1/8$.

Borda. For Borda, we let $d = \frac{4\text{MoV}}{(m-2)!\cdot m}$ and set the profile as

$$\begin{cases} \frac{n+(m-1)d}{m!} \text{ votes for each type such that } c_1 \text{ is top-ranked} \\ \frac{n-d}{m!} \text{ votes for each other type of votes} \end{cases}.$$

It's easy to check that the margin of victory of the above profile is MoV under Borda. Figure 6 plots the comparison between quantum-accelerated voting and classical voting for $m = 4$. Similar behavior as plurality can be observed for Borda.

Copeland. For Copeland, we set the profile as

$$\begin{cases} \frac{n-2\text{MoV}}{m!} + \frac{2\text{MoV}}{(m-2)!} \text{ votes for each type in the form of } c_1 \succ c_2 \succ \text{others} \\ \frac{n-2\text{MoV}}{m!} \text{ votes for each other type of votes} \end{cases}. \quad (2)$$

It's easy to check that the margin of victory of the above profile is MoV under Copeland. Figure 7 plot the comparison between quantum-accelerated voting and classical fast voting for $m = 4$. Similar behavior as plurality can be observed for Copeland.

Single transferable vote (STV). For STV, the same profile as Copeland (see Equation (2)) is used. It's easy to check that the margin of victory of the profile is MoV under STV. Figure 8 plot the comparison between quantum-accelerated voting and classical voting for $m = 4$. Similar behavior as plurality can be observed for STV.

Additional notes. Since Copeland and STV shares the same profile, Figure 7 and Figure 8 look similar. However, they are not the same and some small differences can be observed between the two figures. We also note that all four voting rules (plurality, Borda, Copeland, and STV) reduce to the majority voting when $m = 2$.

References

Johannes Bausch. Recurrent quantum neural networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 1368–1379, 2020.

André Berthiaume and Gilles Brassard. Oracle quantum computing. *Journal of modern optics*, 41(12):2521–2535, 1994.

Yanhu Chen, Shijie Wei, Xiong Gao, Cen Wang, Yinan Tang, Jian Wu, and Hongxiang Guo. A low failure rate quantum algorithm for searching maximum or minimum. *Quantum Information Processing*, 19(8), 2020.

Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

Elham Kashefi, Adrian Kent, Vlatko Vedral, and Konrad Banaszek. Comparison of quantum oracles. *Physical Review A*, 65(5):050304, 2002.

Alastair Kay. Tutorial on the quantikz package, 2018.

Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash. q-means: a quantum algorithm for unsupervised machine learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 4134–4144, 2019.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Kanint Teerapabolarn and Patcharee Wongkasem. On pointwise binomial approximation by w-functions. *International Journal of Pure and Applied Mathematics*, 71:57–66, 01 2011.

Flemming Topsøe. Bounds for entropy and divergence for distributions over a two-element set. *J. Ineq. Pure & Appl. Math*, 2(2):300, 2001.

Wim Van Dam. Quantum oracle interrogation: Getting all information for almost half the price. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 362–367. IEEE, 1998.

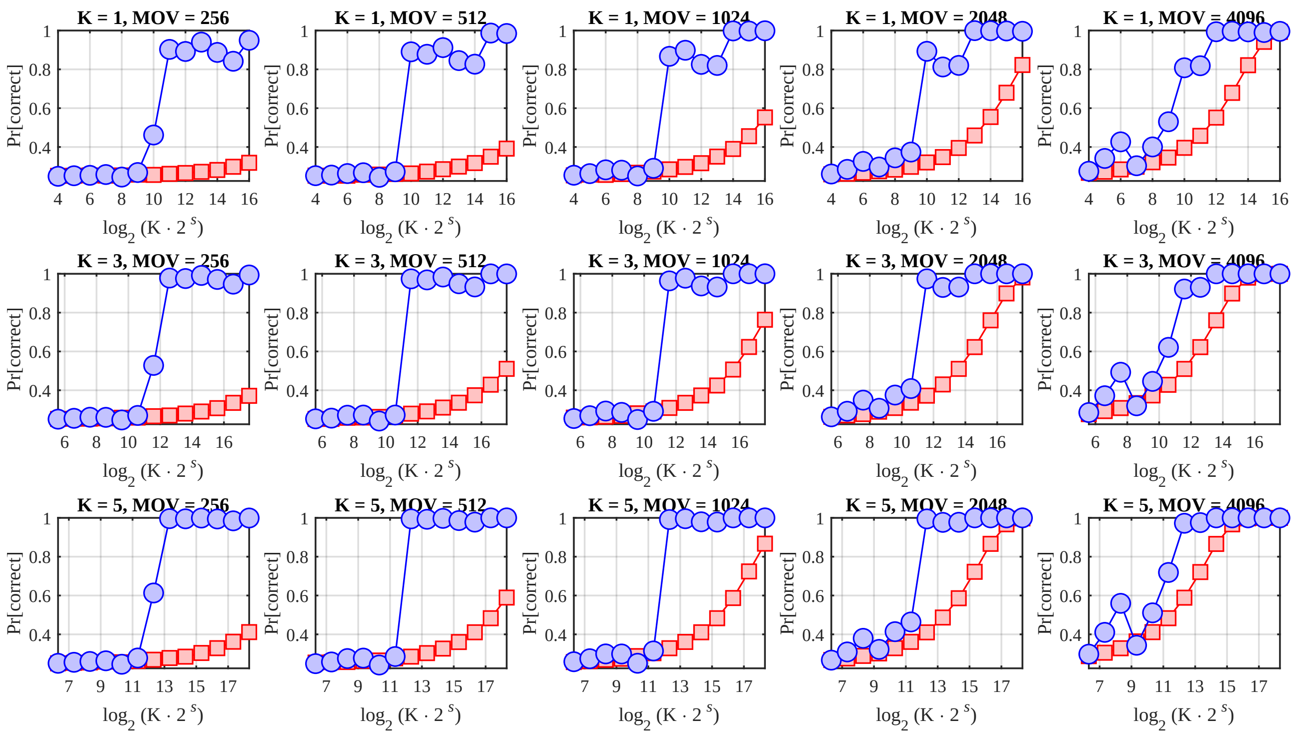


Figure 5: Compare quantum-accelerated voting (blue circles) with classical fast voting (red squares) for plurality when $m = 4$. The horizontal axis can be seen as the logarithm of the algorithms' runtime.

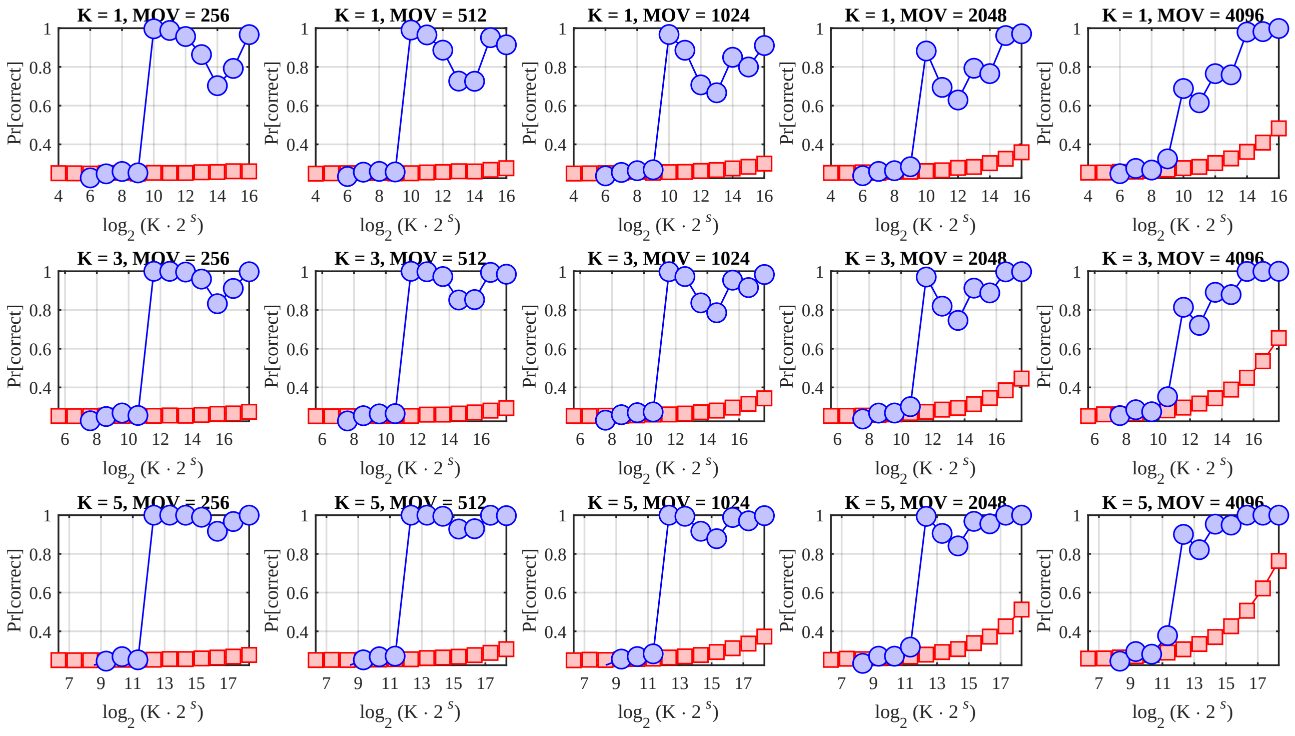


Figure 6: Compare quantum-accelerated voting (blue circles) with classical fast voting (red squares) for Borda when $m = 4$. The horizontal axis can be seen as the logarithm of the algorithms' runtime.

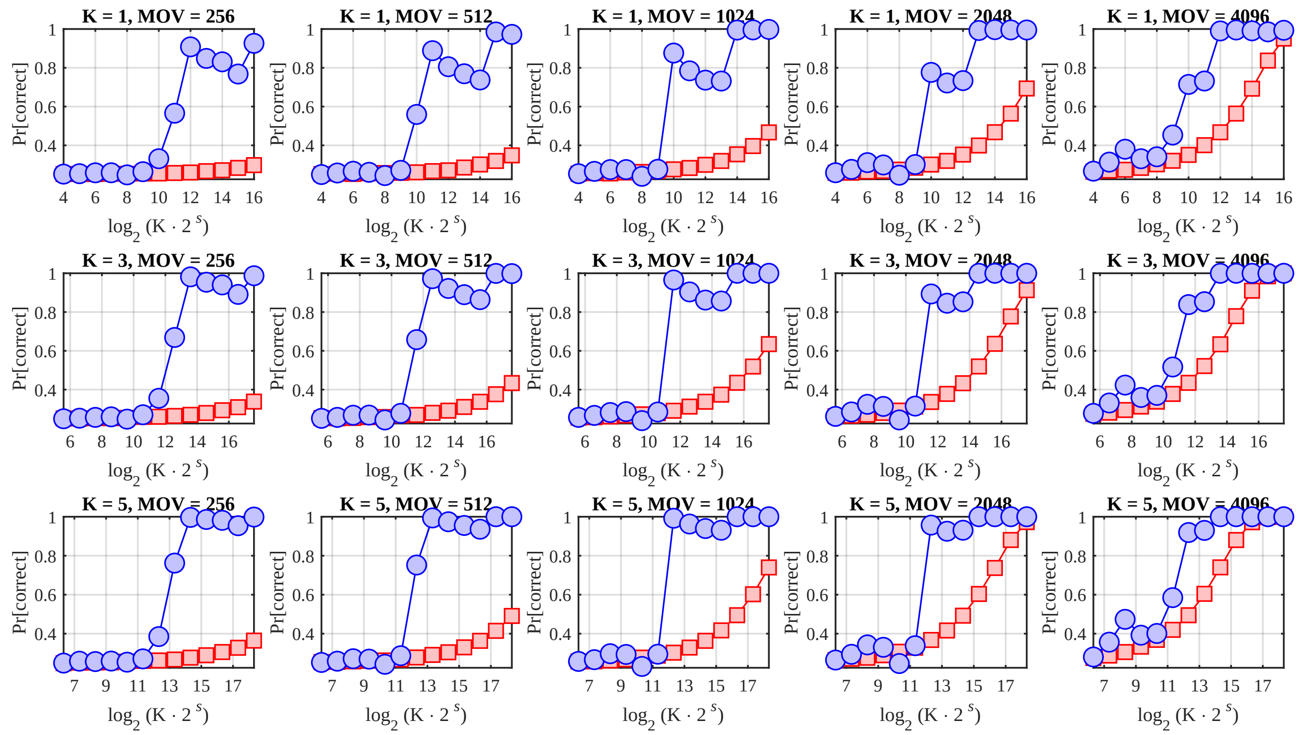


Figure 7: Compare quantum-accelerated voting (blue circles) with classical fast voting (red squares) for Copeland when $m = 4$. The horizontal axis can be seen as the logarithm of the algorithms' runtime.

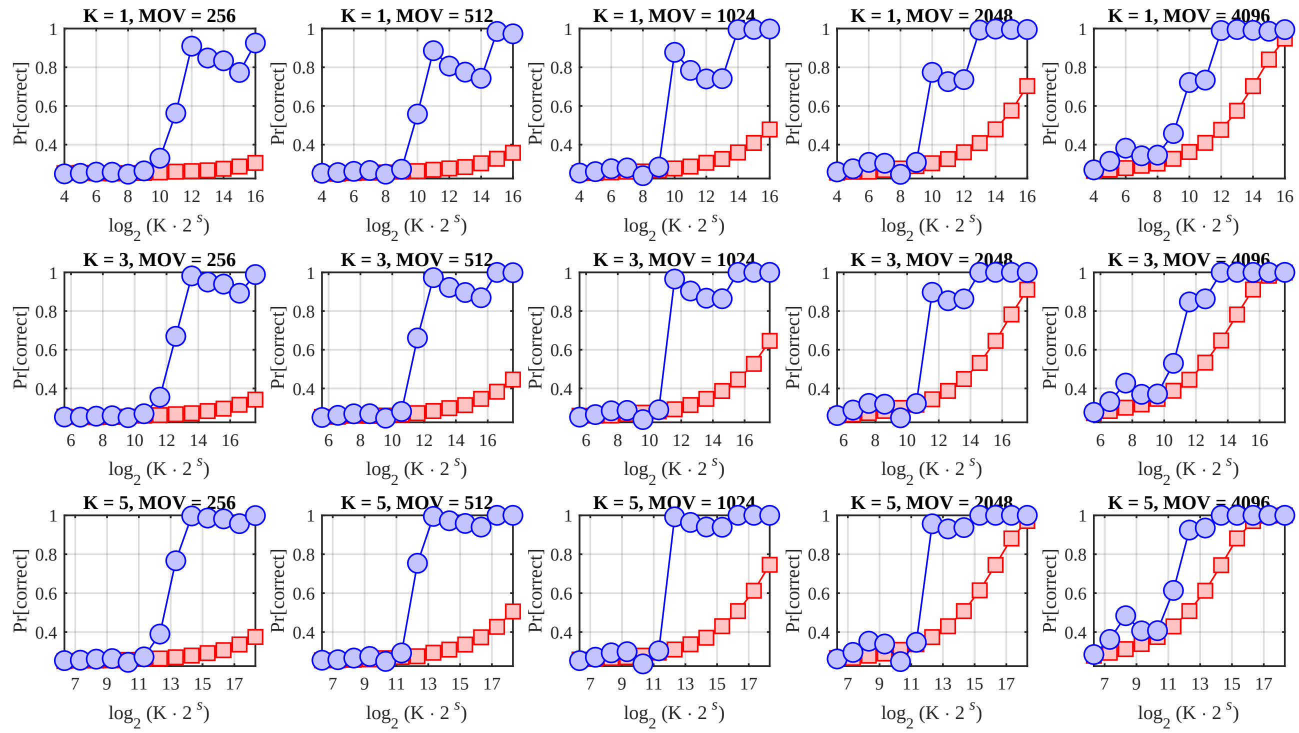


Figure 8: Compare quantum-accelerated voting (blue circles) with classical fast voting (red squares) for STV when $m = 4$. The horizontal axis can be seen as the logarithm of the algorithms' runtime.