A Quantum Algorithm for the Ferromagnetic Ising Model

(Manuscript in preparation)

Ashwin Nayak [*]
U. Waterloo & Perimeter

Leonard J. Schulman [†]
Caltech

Umesh V. Vazirani [‡]
UC Berkeley

April 22, 2008

Abstract

This manuscript presents a quantum algorithm for the (classical) ferromagnetic Ising model discovered by us in 1997, and refined in subsequent years. A more complete version that includes a discussion of the earlier and related work on the subject, a comparison of the time complexity of the algorithm with the known efficient classical algorithm, and possible improvements and extensions will appear shortly.

[*] Department of Combinatorics and Optimization, and Institute for Quantum Computing, University of Waterloo, 200 University Ave. W., Waterloo, Ontario, N2L 3G1, Canada. E-mail: anayak@math.uwaterloo.ca.

[†] Department of Computer Science, California Institute of Technology, Pasadena, California, 91125, USA. E-mail: schulman@caltech.edu.

[‡] Computer Science Division, University of California, Berkeley, California, 94720, USA. E-mail: vazirani@cs.berkeley.edu.

1 The Ising Model

The Ising model [14] is perhaps the simplest model for studying phase transitions in statistical mechanics (such as spontaneous magnetization in certain materials), and more generally a model for studying how local interactions lead to globally observable phenomena.

In the Ising model, viewed as an abstraction of magnetic material, a solid is assumed to be composed of individual magnetic moments (*spins*) located at the sites of a finite lattice structure. The spins interact only with spins adjacent to them on the lattice. The nearest-neighbour interactions determine the energy of the system. At any given temperature, the solid exists in a thermal equilibrium of different spin states given by the Gibbs distribution. The associated sampling problem is to generate a spin state with probability given by this distribution. Formally, the model is defined as follows.

Let $G = (V, E)$ be a graph with n vertices and m edges. Let J_{ij} be the *interaction coefficient* associated with the edge $\{i, j\} \in E$.

A *spin configuration* σ in G is an assignment of a *spin* $\sigma_i \in \{+1, -1\}$ to each vertex $i \in V$. The energy $H(\sigma)$ of a configuration σ is given by:

$$H(\sigma) \quad \stackrel{\text{def}}{=} \quad - \sum_{\{i,j\} \in E} J_{ij} \sigma_i \sigma_j. \tag{1}$$

The Ising model also provides for an external magnetic field, which interacts with each individual spin, and contributes to the energy of the system. This more general model may be reduced to the one above by the introduction of an additional spin which interacts with every other spin with suitably chosen strength. Thus there is no loss in generality in the study of the properties of the model in the absence of a magnetic field. In particular, the computational complexity of sampling spin configurations from the thermal mixture remains the same in the two cases.

The Gibbs distribution on spin configurations assigns a probability proportional to

$$w(\sigma) \quad = \quad \exp(-\beta H(\sigma))$$

to σ, where β is related to the temperature T by the Boltzmann constant k as $\beta = 1/\mathrm{k}T$. More precisely, the probability $\pi_{\mathrm{G}}(\sigma)$ assigned to σ is

$$\pi_{\mathrm{G}}(\sigma) \quad \stackrel{\text{def}}{=} \quad \frac{1}{Z} w(\sigma), \quad \text{where} \tag{2}$$

$$Z \quad \stackrel{\text{def}}{=} \quad \sum_{\sigma \in \{+1,-1\}^n} w(\sigma), \tag{3}$$

the *partition function*, is the normalising quantity.

Several properties of the model, such as the mean energy or the mean magnetic moment, may be derived from the partition function. However, other properties such as the correlation length seem to require the harder task of sampling from the Gibbs distribution, defined below.

> Ising Model
> *Input:* A graph $G = (V, E)$, with interaction coefficients $J \in \mathbb{R}^E$, temperature $T \in \mathbb{R}^+$, and an accuracy parameter $\epsilon \in (0, 2)$.
> *Output:* A spin configuration $\sigma \in \{+1, -1\}^V$ distributed according to some probability distribution P such that $\|P - \pi_{\mathrm{G}}\|_1 \leq \epsilon$.

Note that for the purposes of computation, all explicit numerical values (such as the interaction coefficients) are assumed to be rational numbers.

1.1 Overview of the algorithm

The goal of the quantum algorithm would be to construct an approximation to the state $|\Phi\rangle$:

$$|\Phi\rangle \quad = \sum_{\sigma \in \{+1,-1\}^n} \sqrt{\pi_{\mathrm{G}}(\sigma)}\, |\sigma\rangle. \tag{4}$$

which we call the "Gibbs superposition". The task of generating a coherent superposition of states according to a desired distribution is considered to be computationally hard in general. It is now folklore that such a scheme would give us a quantum algorithm for GROUP ISOMORPHISM. In the present case, even the ostensibly simpler sampling problem had long defied solution. Most computational approaches that attempt to directly solve the ISING MODEL sampling problem, including the Markov Chain Monte Carlo approach, are either provably inefficient or are at best not known to be efficient. We circumvent these barriers by appealing to the "high temperature expansion" of the partition function. This expansion is essentially a Fourier transform over the group $\mathbb{Z}_2^n$; we describe this connection in Section 2. We call the Fourier transform $|\hat{\Phi}\rangle$ of the Gibbs superposition $|\Phi\rangle$ the "parity superposition".

This state $|\hat{\Phi}\rangle$ is a superposition over *parity configurations* $\rho \in \{0,1\}^n$, which may also be interpreted as subsets of $[n]$. When the spin-spin interactions are all *ferromagnetic*, parity superposition is a coherent state associated with a probability distribution μ over *ρ-joins* of the graph G. In Section 4, we show that there is an efficient algorithm that samples (approximately) from the distribution μ. Moreover, we observe that the distribution is *self-reducible*, i.e., it decomposes into a convex combination of distributions that are similar to μ, when conditioned upon the bits of the parity configuration ρ. We describe a scheme for constructing a coherent superposition associated with a special kind of distribution in Section 3. This scheme helps us (approximately) construct the parity superposition $|\hat{\Phi}\rangle$ using the algorithm for sampling from μ. Given an approximation to the state $|\hat{\Phi}\rangle$ we can generate an approximation to $|\Phi\rangle$ by the application of the Fourier transform, as mentioned above.

2 The high temperature expansion

In order to sample from the Gibbs distribution on spins quantum mechanically, we set up the *Gibbs superposition*

$$|\Phi\rangle \quad = \quad \frac{1}{\sqrt{Z}} \sum_{\sigma} \sqrt{w(\sigma)}\, |\sigma\rangle.$$

Rather than directly constructing this superposition, we construct its Fourier transform, which we call the *parity superposition*. This is motivated by the so called *high temperature expansion* of the partition function [14]. It is known that this expansion relates spin configurations to the Eulerian subgraphs of G. Implicit in this expansion is a Fourier transform of the Gibbs distribution. Below, we adapt the high temperature expansion to relate the Gibbs superposition to the parity superposition via this transform.

Note that $\sqrt{w(\sigma)}$ may be expanded as follows:

$$\begin{aligned}\sqrt{w(\sigma)} &= \mathrm{e}^{-\frac{\beta}{2}H(\sigma)}\\ &= \mathrm{e}^{\frac{\beta}{2}\sum_{\{i,j\}\in E} J_{ij}\sigma_i\sigma_j}\\ &= \prod_{\{i,j\}\in E} \mathrm{e}^{\frac{\beta}{2}J_{ij}\sigma_i\sigma_j}.\end{aligned}$$

Using the identity $\mathrm{e}^x = \cosh x\,(1+\tanh x)$, the contribution due to edge $\{i,j\}$ may be rewritten as

$$\cosh\left(\frac{\beta}{2}J_{ij}\sigma_i\sigma_j\right)\left(1+\tanh\left(\frac{\beta}{2}J_{ij}\sigma_i\sigma_j\right)\right).$$

Since cosh is a symmetric function and tanh is antisymmetric, this is equal to

$$\cosh\left(\frac{\beta}{2}J_{ij}\right)\left(1+\sigma_i\sigma_j\tanh\left(\frac{\beta}{2}J_{ij}\right)\right).$$

Letting $\kappa = \frac{1}{\sqrt{Z}}\prod_{\{i,j\}\in E}\cosh\left(\frac{\beta}{2}J_{ij}\right)$, and using the abbreviation λ_{ij} for $\tanh\left(\frac{\beta}{2}J_{ij}\right)$, we may simplify the expression for $\sqrt{w(\sigma)/Z}$:

$$\begin{aligned}\sqrt{\frac{w(\sigma)}{Z}} &= \kappa\prod_{\{i,j\}\in E}(1+\sigma_i\sigma_j\lambda_{ij})\\ &= \kappa\sum_{X\subseteq E}\left(\prod_i \sigma_i^{\deg_X(i)}\right)\prod_{\{i,j\}\in X}\lambda_{ij}.\end{aligned}$$

For a subgraph $X \subseteq E$, define the *parity configuration* of X as the vector $\varrho^X \in \mathbb{Z}_2^n$ where ϱ_i^X is the degree of vertex i, mod 2, in X. Further, let $w(X)$ be the product $\prod_{\{i,j\}\in X}\lambda_{ij}$. For a vector $\rho \in \mathbb{Z}_2^n$, let

$$\alpha_\rho = \kappa\sum_{X\,:\,\varrho^X=\rho} w(X).$$

For a spin configuration σ, let $\bar\sigma$ be the characteristic vector of the set $\{i : \sigma_i = -1\}$, so that $\sigma_i = (-1)^{\bar\sigma_i}$ and $\bar\sigma_i = (1-\sigma_i)/2$.

In terms of these quantities,

$$\begin{aligned}\sqrt{\frac{w(\sigma)}{Z}} &= \kappa\sum_{X\subseteq E}\left(\prod_i(-1)^{\bar\sigma_i\varrho_i^X}\right)w(X)\\ &= \kappa\sum_{X\subseteq E}(-1)^{\bar\sigma\cdot\varrho^X}w(X)\\ &= \kappa\sum_\rho(-1)^{\bar\sigma\cdot\rho}\sum_{X\,:\,\varrho^X=\rho}w(X)\\ &= \sum_\rho(-1)^{\bar\sigma\cdot\rho}\alpha_\rho\end{aligned}$$

where $\bar\sigma\cdot\varrho^X = \sum_i\bar\sigma_i\varrho_i^X \pmod 2$ is the scalar product of the two vectors.

Using the alternative representation of characteristic vectors $\bar{\sigma}$ instead of spin configurations σ, the Gibbs superposition can now be written as:

$$|\Phi\rangle \;=\; \sum_{\bar{\sigma}} \sum_{\rho} (-1)^{\bar{\sigma}\cdot\rho} \alpha_\rho \, |\bar{\sigma}\rangle.$$

Let $|\hat{\Phi}\rangle = 2^{n/2} \sum_\rho \alpha_\rho |\rho\rangle$. On applying the Hadamard transform $\mathrm{H}^{\otimes n}$ to this superposition (which is the *parity superposition* mentioned above), we get

$$\begin{aligned} \mathrm{H}^{\otimes n}|\hat{\Phi}\rangle &= \sum_\rho \alpha_\rho \sum_\eta (-1)^{\eta\cdot\rho} \, |\eta\rangle \\ &= |\Phi\rangle. \end{aligned}$$

We have proven the following theorem.

Theorem 2.1 *The Gibbs superposition $|\Phi\rangle = \mathrm{H}^{\otimes n}|\hat{\Phi}\rangle$, the quantum Fourier transform over $\mathbb{Z}_2^n$ of the parity superposition. Consequently, given a state $|\hat{\Psi}\rangle$ such that $\left\| \hat{\Psi} - \hat{\Phi} \right\|_{\mathrm{tr}} \le \epsilon$, we can generate a state $|\Psi\rangle$ such that $\|\Psi - \Phi\|_{\mathrm{tr}} \le \epsilon$ with the application of n single qubit gates.*

3 Preparation of coherent superpositions

In this section, we describe a generic procedure for constructing coherent superpositions corresponding to a probability distribution. We later apply this technique to prepare the parity superposition $|\hat{\Phi}\rangle$ defined in Section 2 for the ferromagnetic Ising model.

Suppose we would like to construct a superposition $|\phi\rangle \in \mathbb{C}^{2^n}$ over n qubits and with real, non-negative amplitudes. Let $|\phi\rangle = \sum_{x\in\{0,1\}^n} a_x |x\rangle$, where the amplitudes $a_x \in [0,1]$ so that $a_x = \sqrt{p_x}$ for some probability distribution $P = (p_x)$ over $\{0,1\}^n$. We extend the functions $a, p : \{0,1\}^n \to \mathbb{C}$ to $\{0,1\}^{\le n}$ as follows. For any $y \in \{0,1\}^j$, where $j \in [n]$, let $p_y = \Pr_{X\sim P}[X_1 X_2 \cdots X_j = y]$, and let $a_y = \sqrt{p_y}$. Further, define states over j qubits $|\phi_j\rangle = \sum_{y\in\{0,1\}^j} a_y |y\rangle$. Thus $|\phi_n\rangle = |\phi\rangle$, and the states $|\phi_j\rangle$ may be viewed as "prefixes" of this target state.

In addition, suppose we can compute the conditional probability $q_y = \Pr[X_{j+1} = 0 | X_1 X_2 \cdots X_j = y]$ for every $y \in \{0,1\}^j$, where $j \in \{0, \ldots, n-1\}$. Then we can prepare the prefix $|\phi_{j+1}\rangle$ from the prefix $|\phi_j\rangle$ as follows. For $\theta \in [0, \pi/2]$, let the state $|\theta\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle$.

1. Compute q_y from y.
2. Prepare a qubit in the superposition $|\theta_y\rangle$, controlled on the y-register, where $\theta_y = \arccos\sqrt{q_y}$.
3. Uncompute q_y.

Repeating this procedure for $j = 0, 1, \ldots, n-1$, we get $|\phi\rangle$. This kind of procedure has been observed in several works [8, 4] since ours was first announced. It is a quantum analogue of a standard technique in randomized algorithms for sampling from distributions.

For the problem at hand, we are only able to implement the different steps above *approximately*:

- We approximate the probability q_y using a quantum simulation of a randomized algorithm that takes y as input. The randomized algorithm computes an approximation $\tilde{q}_y$ within some additive precision $\delta \in [0,1)$ with probability $1-\epsilon$, for some $\epsilon \in [0,1)$.

Algorithm $\mathcal{Q}(j, |\phi_j\rangle)$.

Recall that $|\phi_j\rangle = \sum_{y \in \{0,1\}^j} a_y |y\rangle$. The parameters ϵ, δ, η are set globally. The algorithm $\mathcal{A}$ uses k random bits.

1. Apply the Hadamard gate to k fresh ancillary qubits initialized to $|0\rangle$, to obtain the superposition $2^{-k/2} \sum_{s \in \{0,1\}^k} |s\rangle$.
2. Reversibly simulate $\mathcal{A}$ on inputs y, s to obtain an approximation $\tilde{q}_{y,s}$. The string y is empty if $j = 0$.
3. Let $\tilde{\omega}_{y,s} = 1/4$ if $\tilde{q}_{y,s} < \delta$, and let $\tilde{\omega}_{y,s} = 0$ if $\tilde{q} > 1 - \delta$. Go to Step 5 if either of these conditions is satisfied, and proceed to the next step otherwise.
4. Reversibly approximate $\frac{1}{2\pi} \arccos(\tilde{q}_{y,s})^{1/2}$ to within additive error $\delta/2\pi$, to get $\tilde{\omega}_{y,s}$.
5. Prepare a fresh qubit in state $|\tilde{\theta}_{y,s}\rangle$, where $\tilde{\theta}_{y,s} = 2\pi\tilde{\omega}_{y,s}$.
6. Uncompute $\tilde{\omega}_{y,s}$, and $\tilde{q}_{y,s}$.
7. Apply the Hadamard gate to the k qubits carrying the strings s, and measure them in the computational basis.
8. If any of the k outcomes is 1, output $|0\rangle|0^{j+1}\rangle$, where the value 0 of the first qubit indicates failure. Otherwise, output $|1\rangle$ along with the system containing the input and the fresh qubit prepared in Step 5. The value 1 of the first qubit indicates success.

Figure 1: Details of the algorithm that extends the state $|\phi_j\rangle$ by a single qubit.

- Given some $q \in [0, 1]$, we approximate the state $|\theta\rangle$ to within η in the trace norm, where $\theta = \arccos\sqrt{q}$.

In our application, the parameters δ, ϵ and η are typically taken to be inverse polynomial in n, the number of qubits in the state $|\phi\rangle$.

The randomized algorithm that approximates q_y given $y \in \{0,1\}^j$ may be viewed as a *deterministic* algorithm $\mathcal{A}$ that takes y and string s as input, where s is chosen uniformly at random from $\{0,1\}^k$ for some k. We assume, w.l.o.g., that the number of random bits k is independent of y. The details of the quantum algorithm that prepares $|\phi_{j+1}\rangle$ from $|\phi_j\rangle$ are laid out in Figure 1.

The numerical computations in Step 4 are well-studied operations. We may, for example, use the algorithms presented in [3] to implement them. For completeness, we summarize the properties of these algorithms.

We say that an operation is *performed with precision* l if the operands are represented as floating point numers with a binary fraction of l bits, and an exponent whose length may grow as $\mathrm{o}(l)$. The time complexity (number of bit operations) of the numerical algorithms formally hold for multitape Turing machines with a fixed number of tapes. In particular, they hold for Boolean circuits as well. Let $M(l)$ be the time required to perform precision l multiplication. The asymptotically fastest method for multiplication is one due to Schönhage and Strassen [11], which gives

$$\mathrm{M}(l) \quad \in \quad \mathrm{O}(l \log(l) \log\log(l)).$$

Theorem 3.1 *[Brent [3]] The following hold:*

1. *Division, and finding reciprocals and square roots with precision l all take time* $\mathrm{O}(\mathrm{M}(l))$.
2. *The first l bits of the universal constant π may be computed in time* $\mathrm{O}(\mathrm{M}(l)\log l)$.
3. *The trigonometric function* arccos *may be computed with precision l in time* $\mathrm{O}(\mathrm{M}(l)\log l)$.

As a corollary, we get an upper bound on the time required for Step 4.

Corollary 3.2 *Given an l-bit number $q \in [0,1]$, the number $\frac{1}{2\pi}\arccos\sqrt{q}$ may be approximated with precision l in time $\mathrm{O}(l(\log l)^2 \log\log l)$.*

The rotation in Step 5 may be implemented in one of several ways, depending upon which universal gate set is available to us. It may be implemented using a fixed single qubit rotation as described in [2, Lemma 6.3.1]. This takes time $\mathrm{O}(\frac{1}{\eta})$, where the desired accuracy is η in trace norm. Similar time complexity and approximation is implied by a result due to [5], who show "efficient universality" of certain finite sets of single qubit gates.

Here we describe the approach taken in [8], although using any of the other methods above does not affect the overall complexity of our final algorithm. The approach in [8] uses a family of controlled single qubit phase gates, the single qubit Hadamard gate and the phase gate $\mathrm{diag}(1,\mathrm{i})$. (The advantage of the other two methods is that they use a finite set of single qubit gates.) The controlled phase gates used are precisely the ones used in an exact implementation of the quantum Fourier transform with respect to $\mathbb{Z}_N$, where N is a power of 2. The controlled phase gates may be simulated with CNOT, single qubit phase gates, and appropriate Pauli Z rotations [10, Chapter 4].

Lemma 3.3 (Kaye and Mosca [8]) *Given an l-bit fraction ω in binary, the state $|\theta\rangle$ with $\theta = 2\pi\omega$ may be prepared exactly with a circuit of size $\mathrm{O}(l)$. The circuit uses only the Hadamard gate, the phase gate $\mathrm{diag}(1,\mathrm{i})$, and controlled phase gates $\mathrm{diag}(1,\exp(\frac{2\pi\mathrm{i}}{2^m}))$, where $m \in [l]$.*

Corollary 3.2 and Lemma 3.3 together imply that:

Lemma 3.4 *Let $l \geq 1$. Given an l-bit approximation $\tilde{q} \in [0,1]$ to $q \in [0,1]$ such that $|q-\tilde{q}| \leq \delta = 2^{-l}$, there is a quantum circuit of size*

$$\log\left(\frac{1}{\delta}\right)\left(\log\log\frac{1}{\delta}\right)^2 \log\log\log\frac{1}{\delta}$$

which prepares a state $|\tilde{\theta}\rangle$ such that $\left\| |\theta\rangle - |\tilde{\theta}\rangle \right\|_2 \leq 2\sqrt{2\delta}$, where $\theta = \arccos\sqrt{q}$, and $\tilde{\theta} = \arccos(\tilde{q})^{1/2}$.

Proof: If $\tilde{q} < \delta$ or $1-\tilde{q} < \delta$, we prepare the state $|1\rangle$ or $|0\rangle$, respectively as our approximation to $|\theta\rangle$. A straightforward calculation shows that this guarantees an approximation of the desired state to within $\sqrt{2\delta}$ in ℓ_2 norm. The time complexity is $\mathrm{O}(l)$.

If $\delta < \tilde{q} < 1-\delta$, we follow a different procedure. Let $\tilde{\omega}$ be the approximation to $\omega' = \frac{1}{2\pi}\arccos(\tilde{q})^{1/2}$ to within $\delta/2\pi$ given by Corollary 3.2, and let $\omega = \frac{1}{2\pi}\arccos\sqrt{q}$. The size of the circuit follows from the Corollary.

For the error analysis, note that

$$\left\| |\theta\rangle - |\tilde{\theta}\rangle \right\|_2 \quad \leq \quad 4\pi\,|\omega - \tilde{\omega}|\,,$$

and

$$\begin{aligned}|\tilde{\omega}-\omega| &\leq \left|\tilde{\omega}-\omega'\right| + \left|\omega'-\omega\right| \\ &\leq \frac{1}{2\pi}\cdot\delta + \left|\omega'-\omega\right| .\end{aligned}$$

Finally, by elementary calculus, for some q' between $\tilde{q}$ and q,

$$\begin{aligned}\left|\omega'-\omega\right| &= \frac{1}{2\pi}\left|\arccos(\tilde{q})^{1/2} - \arccos(q)^{1/2}\right| \\ &= \frac{|\tilde{q}-q|}{4\pi\sqrt{q'(1-q')}} \\ &\leq \frac{1}{2\sqrt{2}\,\pi}\cdot\sqrt{\delta}.\end{aligned}$$

Altogether we have $\left\||\theta\rangle - |\tilde{\theta}\rangle\right\|_2 \leq 2\sqrt{2\delta}$. ■

The above lemma tells us that the error η mentioned earlier in preparing the qubit is bounded as $\eta \leq 4\sqrt{2\delta}$, given a δ approximation to the probability q.

Next, we bound the error in the construction of $|\phi_{j+1}\rangle$ from $|\phi_j\rangle$ with the algorithm $\mathcal{Q}$ presented in Figure 1.

Lemma 3.5 *Consider the output of the algorithm $\mathcal{Q}$ on input state $|\phi_j\rangle$. The probability that the algorithm fails is at most $4\epsilon + 8\delta$. If $|\tilde{\phi}_{j+1}\rangle$ is the output state when the algorithm does not fail, then $\left\|\tilde{\phi}_{j+1} - \phi_{j+1}\right\|_{\mathrm{tr}} \leq 4\sqrt{2\epsilon + 4\delta}$.*

Proof: Consider the state of the algorithm $\mathcal{Q}$ on input $|\phi_j\rangle$ after Step 6. It may be written as:

$$\sum_y a_y|y\rangle \otimes \frac{1}{2^{s/2}}\sum_s |s\rangle|\tilde{\theta}_{y,s}\rangle,$$

where $\tilde{\theta}_{y,s}$ is defined as in Figure 1.

For every string y the algorithm $\mathcal{A}$ produces an approximation to q_y that is within δ. So for a $1-\epsilon$ fraction of the strings s, we have $\left\||\tilde{\theta}_{y,s}\rangle - |\theta_y\rangle\right\|_2 \leq 2\sqrt{2\delta}$ by Lemma 3.4. In other words, for these string pairs y, s, $\left|\langle\theta_y|\tilde{\theta}_{y,s}\rangle\right| \geq 1 - 4\delta$. Therefore, the inner product of the state with the ideal state $\sum_y a_y|y\rangle \otimes \frac{1}{2^{k/2}}\sum_s |s\rangle|\theta_y\rangle$ is at least

$$(1-\epsilon)(1-4\delta) - \epsilon \quad\geq\quad 1 - 2\epsilon - 4\delta$$

in magnitude. Therefore the probability that all k bits are 0 when measured is at least the square, i.e., at least $1 - 4\epsilon - 8\delta$.

Note that the inner product above is a lower bound on $\left|\langle\phi_{j+1}|\tilde{\phi}_{j+1}\rangle\right|$. Therefore,

$$\begin{aligned}\left\||\tilde{\phi}_{j+1}\rangle - |\phi_{j+1}\rangle\right\|_{\mathrm{tr}} &\leq 2\left\||\tilde{\phi}_{j+1}\rangle - |\phi_{j+1}\rangle\right\|_2 \\ &\leq 4\sqrt{2\epsilon + 4\delta}\ .\end{aligned}$$

This completes the error analysis. ■

Algorithm $\mathcal{P}$.

This algorithm receives no input, and maintains a $j+1$ qubit state of the form $\gamma_j |0\rangle\langle 0| \otimes |0^j\rangle\langle 0^j| + (1-\gamma_j)|1\rangle\langle 1| \otimes |\psi_j\rangle\langle\psi_j|$ in the jth iteration. A value of 0 in the first qubit indicates failure, and a value of 1 indicates success. Here $|\psi_0\rangle$ denotes the "empty" quantum state, as does $|0^0\rangle\langle 0^0|$. The initial state of the algorithm is $|1\rangle|\psi_0\rangle$.
For $j = 0$ to $n-1$

1. If the first qubit of the current state is 0, prepare the $(j+1)$th qubit in state $|0\rangle$.
2. Otherwise, run algorithm $\mathcal{Q}(j, |\psi_j\rangle)$, where $|\psi_j\rangle$ is the state of the remaining j qubits, and set the current state to the output of this run of $\mathcal{Q}$.

Output the $(n+1)$-qubit current state of the algorithm.

Figure 2: Details of the algorithm that prepares the superposition $|\phi\rangle$.

An approximation to the state $|\phi_n\rangle = |\phi\rangle$ may now be prepared by n applications of the algorithm $\mathcal{Q}$, as described in Figure 2. The performance of the algorithm may be inferred from a "hybrid argument" [1, 13].

Theorem 3.6 *Let $\epsilon + 2\delta \leq 1$. Then,*

1. *The probability γ that the algorithm $\mathcal{P}$ fails is bounded as $\gamma \leq 10n\sqrt{2\epsilon + 4\delta}$.*
2. *When the algorithm succeeds, the output state $|\psi_n\rangle$ is such that*

$$\|\psi_n - \phi_n\|_{\mathrm{tr}} \quad \leq \quad 10n\sqrt{2\epsilon + 4\delta}\;/(1-\gamma).$$

3. *The algorithm runs in time of order*

$$n \cdot \left[T + \log\left(\frac{1}{\delta}\right)\left(\log\log\frac{1}{\delta}\right)^2 \log\log\log\frac{1}{\delta} \right],$$

where T is a bound on the run-time of the algorithm $\mathcal{A}$ with parameters δ, ϵ used in Figure 1.

Proof: The run time of the algorithm $\mathcal{P}$ follows directly from its description and Lemma 3.4.

For the purposes of error analysis, we assume the existence of an ideal algorithm $\mathcal{Q}^*$ such that $\mathcal{Q}^*(j, |\phi_j\rangle) = |1\rangle|\phi_{j+1}\rangle$, for all $j = 0, 1, \ldots, n-1$. We imagine $n+1$ runs of the algorithm $\mathcal{P}$, where in the lth run, we use $\mathcal{Q}^*$ in the first l iterations, and then $\mathcal{Q}$ in the remaining $n-l$ iterations (where $l = 0, 1, 2, \ldots, n$). Let τ_l be the density matrix of the output of the lth run. We have $\tau_0 = \gamma|0\rangle\langle 0| \otimes |0^n\rangle\langle 0^n| + (1-\gamma)|1\rangle\langle 1| \otimes |\psi_n\rangle\langle\psi_n|$, where γ is the probability that the algorithm $\mathcal{P}$ fails. We have $\tau_n = |1\rangle\langle 1| \otimes |\phi_n\rangle\langle\phi_n|$, the ideal output state. By the triangle inequality,

$$\|\tau_0 - \tau_n\|_{\mathrm{tr}} \quad \leq \quad \sum_{l=0}^{n-1} \|\tau_l - \tau_{l+1}\|_{\mathrm{tr}}\,.$$

We bound each of the terms in the sum on the right hand side. Note that the lth and the $(l+1)$th imaginary runs differ only in the $(l+1)$th iteration. In this iteration, the former uses algorithm $\mathcal{Q}$ and the latter uses the ideal algorithm $\mathcal{Q}^*$. As any quantum operation only decreases the trace distance between two quantum states,

$$\left\|\tau_l - \tau_{l+1}\right\|_{\mathrm{tr}} \quad \leq \quad \left\|\tilde{\tau}_{l,l+1} - \tilde{\tau}_{l+1,l+1}\right\|_{\mathrm{tr}} .$$

where $\tilde{\tau}_{l,m}$ denotes the state maintained by the algorithm $\mathcal{P}$ after the mth iteration of the lth imaginary run. We have

$$\begin{aligned} \tilde{\tau}_{l,l+1} &= \mathcal{Q}(l, |\phi_l\rangle) &&= \tilde{\gamma}_{l+1}|0\rangle\langle 0| \otimes |0^{l+1}\rangle\langle 0^{l+1}| + (1-\tilde{\gamma}_{l+1})|1\rangle\langle 1| \otimes |\tilde{\phi}_l\rangle\langle\tilde{\phi}_l|, \quad \text{and} \\ \tilde{\tau}_{l+1,l+1} &= \mathcal{Q}^*(l, |\phi_l\rangle) &&= |1\rangle\langle 1| \otimes |\phi_{l+1}\rangle\langle\phi_{l+1}|, \end{aligned}$$

where $\tilde{\gamma}_{l+1}$ is the probability that $\mathcal{Q}$ fails on input $l, |\phi_l\rangle$, and $|\tilde{\phi}_{l+1}\rangle$ is its output state on success. Therefore

$$\begin{aligned} \left\|\tilde{\tau}_{l,l+1} - \tilde{\tau}_{l+1,l+1}\right\|_{\mathrm{tr}} &\leq \tilde{\gamma}_{l+1} + \left\|\tilde{\phi}_{l+1} - \phi_{l+1}\right\|_{\mathrm{tr}} \\ &\leq 4(\epsilon + 2\delta) + 4\sqrt{2\epsilon + 4\delta} \qquad \text{By Lemma 3.5} \\ &\leq 10\sqrt{2\epsilon + 4\delta}, \end{aligned}$$

which implies that $\left\|\tau_0 - \tau_n\right\|_{\mathrm{tr}} \leq 10n\sqrt{2\epsilon + 4\delta}$. The claims in the theorem follow. ■

This completes the description and analysis of our state preparation procedure.

We remark, as in [8], that an n-qubit state with relative phases between computational basis states may also be prepared efficiently under suitable conditions. One sufficient condition is that there be an efficient algorithm to approximate the phase corresponding to basis state $|x\rangle$ for each $x \in \{0,1\}^n$. This condition may be relaxed further, for example in ways similar to those encountered in the preparation of the state $|\phi\rangle$ above.

4 A Markov chain on subgraphs

In this section, we review a Markov chain simulation algorithm introduced and analyzed by Jerrum and Sinclair [6]. The Markov chain is used in estimating conditional probabilities (cf. Section 3) associated with the parity configuration $|\hat{\Phi}\rangle$ introduced in Section 1.1.

Let $G = (V, E)$ be a labelled multigraph with a non-negative weight λ_e associated with each edge $e \in E$. Let $n = |V|$, and $m = |E|$. For a subgraph $X \subseteq E$, let $w(X)$ denote the product $\prod_{e \in X} \lambda_e$, and let $\varrho^X \in \mathbb{Z}_2^n$ denote its parity configuration (c.f. section 2). Let ν be a number between 0 and 1. For a vector $\upsilon \in \mathbb{Z}_2^n$, let $h(\upsilon)$ denote the Hamming weight of υ.

Consider the problem of sampling from a distribution on the subgraphs of G, where a subgraph $X \subseteq E$ is assigned probability $\pi_{\mathrm{S}}(X)$ defined by

$$\pi_{\mathrm{S}}(X) \quad = \quad \frac{w(X)\, \nu^{h(\varrho^X)}}{W},$$

where $W = \sum_{X \subseteq E} w(X)\, \nu^{h(\varrho^X)}$ is the normalizing factor.

To show that we can efficiently sample from the distribution π_{S}, we exhibit a *rapidly mixing Markov chain* $\mathcal{M}$, on the subgraphs $X \subseteq E$ of the graph G which has this as its stationary distribution [7].

The transition probability $p(X,Y)$ from a subgraph X to another, Y, in the chain $\mathcal{M}$ is given by the rules below:

$$p(X,Y) \quad = \quad \begin{cases} \frac{1}{2m}\min\left\{1, \frac{\pi_{\mathrm{S}}(Y)}{\pi_{\mathrm{S}}(X)}\right\} & \text{if } X \oplus Y = e \text{ for some } e \in E \\ 1 - \sum_{e \in E} p(X, X \oplus e) & \text{if } Y = X \\ 0 & \text{otherwise} \end{cases}$$

Thus, a transition from a subgraph X is made by staying at X with probability a half, and otherwise choosing a random edge e, and moving to the subgraph $X \oplus e$ according to the *Metropolis rule* [7]. This chain is clearly connected and aperiodic. The Metropolis rule ensures that the chain is also reversible with stationary distribution π_{S}. Thus, it only remains to show that the Markov chain is rapidly mixing. By this we mean that the *mixing time* of the chain is polynomial in the diameter of the underlying graph. Let us define the relevant quantities and the notation used in the analysis of the chain $\mathcal{M}$. Let $P^t(X,\cdot) = (p^t(X,Y))$ denote the probability distribution on subgraphs Y obtained by simulating t steps of $\mathcal{M}$, starting from subgraph X. The mixing time $\tau_X(\epsilon)$ is the first time t such that the distribution $P^t(X,\cdot)$ is within ϵ of the stationary distribution π_{S} in ℓ_1 norm: $\left\|P^t(X,\cdot) - \pi_{\mathrm{S}}\right\|_1 \leq \epsilon$. The distance from stationary distribution decreases monotonically in the number of steps.

We show that $\mathcal{M}$ is rapidly mixing using a *canonical paths* argument [7, Section 12.3.1]. Given a pair of subgraphs X, Y, we define a path $\gamma_{X,Y}$ between them in the following manner. Consider the graph $Z = X \oplus Y$. Let the number of odd degree vertices in Z be $2k$. We can thus partition Z into exactly k walks and some number of cycles in a canonical manner. The subgraph X can then be transformed into Y by successively XORing the edges in Z *along the k walks* and *along the cycles*, again in some canonical order. This defines the unique path $\gamma_{X,Y}$ between X and Y in the Markov chain.

Each canonical path $\gamma_{X,Y}$ is assigned a "flow" of $\pi_{\mathrm{S}}(X)\,\pi_{\mathrm{S}}(Y)\;|\gamma_{X,Y}|$, where $|\gamma_{X,Y}|$ is the length of the canonical path. An upper bound on the *mixing time* of the Markov chain may be obtained by considering the maximum "congestion" B on any transition (T,T') in $\mathcal{M}$. For any set of canonical paths Π, define

$$B = B(\Pi) \quad = \quad \max_{\text{transitions } (T,T')} \quad \frac{1}{\pi_{\mathrm{S}}(T,T')} \sum_{\gamma_{X,Y} \in \Pi \,:\, (T,T') \in \gamma_{X,Y}} \pi_{\mathrm{S}}(X)\,\pi_{\mathrm{S}}(Y)\;|\gamma_{X,Y}|\,,$$

where $\pi_{\mathrm{S}}(T,T') = \pi_{\mathrm{S}}(T)\,p(T,T')$ is the stationary probability of the transition.

Theorem 4.1 (Sinclair [12]) *Consider any finite, reversible, ergodic Markov chain with loop probabilities $p(x,x) \geq 1/2$ for every state x. Let Π be any set of canonical paths with congestion $B = B(\Pi)$. Then for any choice of initial state x, the mixing time $\tau_x(\epsilon)$ satisfies*

$$\tau_x(\epsilon) \quad \leq \quad B \ln \frac{1}{\epsilon\,\pi(x)},$$

where $\pi(\cdot)$ is the stationary distribution of the chain.

We therefore bound the congestion of the set of canonical paths Γ.

Lemma 4.2 *The set of canonical paths Γ defined above has congestion B at most $2m^2/\nu^4$.*

Proof: Let $b = 2m/\nu^4$. It suffices to show that for each transition (T, T') in $\mathcal{M}$,

$$\sum_{X,Y\,:\,(T,T')\in\gamma_{X,Y}} \pi_{\mathrm{S}}(X)\,\pi_{\mathrm{S}}(Y) \quad\leq\quad b\,\pi_{\mathrm{S}}(T,T'), \tag{5}$$

Since $|\gamma_{X,Y}| \leq m$ for any canonical path, it follows that $B \leq m\cdot b$. To establish the bound in Eq. (5), it suffices to associate with every pair X, Y such that $(T, T') \in \gamma_{X,Y}$, a distinct *complementary point* $U_{X,Y}$ in the Markov chain $\mathcal{M}$ such that

$$\pi_{\mathrm{S}}(X)\,\pi_{\mathrm{S}}(Y) \quad\leq\quad b\,\pi_{\mathrm{S}}(T,T')\,\pi_{\mathrm{S}}(U_{X,Y}). \tag{6}$$

We can obtain the bound in Eq. (5) by summing Eq. (6) over appropriate pairs X, Y and observing that the sum of $\pi_{\mathrm{S}}(U_{X,Y})$ is at most 1.

In order to define $U_{X,Y}$, we consider a subgraph from T, T' which occurs with smaller probability in the stationary distribution π_{S}. For concreteness, say $\pi_{\mathrm{S}}(T) \leq \pi_{\mathrm{S}}(T')$; otherwise we work with T'. Let $U_{X,Y} = Z \oplus T$, where $Z = X \oplus Y$. Each pair X, Y is mapped to a distinct point—given $U_{X,Y}$, we can reconstruct $Z = U_{X,Y} \oplus T$; further, the transition (T, T') tells us what edge was used to make the transition, and the canonical processing order on the edges in Z helps us retrieve X and Y from this information.

Notice that going from X to Y along the canonical path involves only the deletion of some edges in $X - Y$, and the addition of some edges in $Y - X$. So any intermediate subgraph W on the path is such that $X \cap Y \subseteq W \subseteq X \cup Y$. Moreover, the complementary point U is given by $U = X \uplus Y - W$, where '$\uplus$' stands for multiset union. This in turn implies that

$$w(X)\,w(Y) \quad=\quad w(W)\,w(U). \tag{7}$$

Notice also that the odd degree vertices in $Z = X \oplus Y$ are exactly the vertices where X and Y differ in parity. Thus XORing a walk from Z into X (or more generally, into an intermediate subgraph W on the canonical path from X to Y) corresponds to "correcting" the parity of the two end vertices, while XORing a cycle preserves parities of all the vertices. On the other hand, the parities in the corresponding complementary points U start out being the same as in Y, and are successively "corrected" to those in X as we follow the canonical path. Thus, after XORing a whole number of walks and/or cycles, we have

$$\begin{array}{c} \varrho_i^W \;=\; \varrho_i^Y \quad\text{and}\quad \varrho_i^U \;=\; \varrho_i^X \\ \text{or} \\ \varrho_i^W \;=\; \varrho_i^X \quad\text{and}\quad \varrho_i^U \;=\; \varrho_i^Y \end{array} \tag{8}$$

for every vertex i. This immediately implies that

$$h(\varrho^W) + h(\varrho^U) \quad=\quad h(\varrho^X) + h(\varrho^Y). \tag{9}$$

However, if a walk has been traced out only partially, Eq. (8) may fail to hold for the vertex where the processing was discontinued, introducing a discrepancy of at most 2 in Eq. (9). Similarly, the property in Eq. (8) may fail to hold for the two end vertices of a partially processed cycle, leading to a discrepancy of at most 4. Thus, we can assert that

$$h(\varrho^W) + h(\varrho^U) \quad\leq\quad h(\varrho^X) + h(\varrho^Y) + 4 \tag{10}$$

for every intermediate subgraph W (and the corresponding complementary point U) on the canonical path from X to Y.

Combining Eq. (7) and (10) for the particular case of $W = T$ and $U = U_{X,Y}$, and noting that $\nu \leq 1$, we get

$$w(X)\, w(Y)\, \nu^{h(\varrho^X)+h(\varrho^Y)+4} \quad \leq \quad w(T)\, w(U)\, \nu^{h(\varrho^T)+h(\varrho^U)}. \tag{11}$$

We divide both sides by $W^2\nu^4$. We have $\pi_{\mathrm{S}}(T,T') = \pi_{\mathrm{S}}(T) \min\{1, \pi_{\mathrm{S}}(T')/\pi_{\mathrm{S}}(T)\}/2m = \pi_{\mathrm{S}}(T)/2m$, since we assumed that $\pi_{\mathrm{S}}(T) \leq \pi_{\mathrm{S}}(T')$. This gives us Eq. (6) with $b = 2m/\nu^4$. ■

This lemma immediately gives us a bound on the mixing time of the subgraphs Markov chain $\mathcal{M}$.

5 Constructing the parity superposition

Let $G = (V, E)$ be a (simple) graph with n vertices and m edges underlying the Ising model with interaction coefficients $J \in \mathbb{R}^E$. In this article, we are concerned with the *ferromagnetic* case, where the interactions are all “co-operative”, i.e., $J_{ij} \geq 0$ for all edges $\{i,j\} \in E$. Recall from Section 2 that the parity superposition is defined as

$$\begin{aligned}
|\hat{\Phi}\rangle &= 2^{n/2} \sum_{\rho \in \{0,1\}^n} \alpha_\rho |\rho\rangle, \quad \text{where} \\
\alpha_\rho &= \kappa \sum_{X \subset E \,:\, \varrho^X = \rho} w(X), \\
w(X) &= \prod_{\{i,j\} \in X} \lambda_{ij}, \\
\lambda_{ij} &= \tanh\left(\frac{\beta}{2} J_{ij}\right),
\end{aligned}$$

and κ is determined by the inputs, and is independent of the parity configuration ρ. Note that $\lambda_{ij} \geq 0$, because the interactions are all co-operative. Thus, $|\hat{\Phi}\rangle$ has non-negative real amplitudes, which correspond to a probability distribution μ on parity configurations ρ. We may therefore use the generic technique developed in Section 3 to prepare this state. The technique asks for an algorithm to approximate certain conditional probabilities associated with the distribution μ. We describe such an algorithm below, based on the self-reducibility of μ.

The distribution μ assigns a probability to each parity configuration $\rho \in \{0,1\}^n$ that is proportional to weights $w(\rho)$ given by

$$w(\rho) \quad = \sum_{X,Y \subset E \,:\, \varrho^X = \rho} w(X)\, w(Y). \tag{12}$$

This weight is essentially the square of the amplitude corresponding to ρ; we have only dropped the factor $2^n\kappa^2$, which is independent of the configuration ρ. Subgraphs $X \subseteq E$ whose parity configuration ϱ^X is ρ are called *ρ-joins* in the literature on graph theory.

Given a string $y \in \{0,1\}^j$, for some $j = 0, 1, \ldots, n-1$, we devise an algorithm $\mathcal{A}$, an estimator, that takes as input the pair j, y, and produces an approximation to the probability q_y:

$$q_y \quad = \quad \Pr_{\rho \sim \mu}[\rho_{j+1} = 0 \mid \rho_1 \cdots \rho_j = y]. \tag{13}$$

The estimator is based on a Markov chain simulation algorithm $\mathcal{S}$ that samples from the distribution μ_y, which is μ conditioned upon the event $\rho_1 \cdots \rho_j = y$. (A linear time procedure to determine whether the event $\rho_1 \cdots \rho_j = y$ has non-zero probability is described below.) The distribution μ_y assigns a probability to each parity configuration $\rho \in \{0,1\}^n$ with prefix y that is proportional to the weight given by Eq. (12). Parity configurations whose prefixes are not equal to y are given 0 weight. This reduction from estimation to sampling is standard and is a consequence of the Hoeffding inequality in probability theory; we state this reduction as it applies in our case.

Lemma 5.1 *Let $\delta, \epsilon \in (0, 1/2]$. Let $\mathcal{S}(y, \zeta)$ be an algorithm that samples from a distribution P on parity configurations such that $\|P - \mu_y\| \leq \zeta$. There is an experiment $\mathcal{A}$ which uses a total number of (independent) samples of the order of $\delta^{-2} \log \epsilon^{-1}$ from the distribution P generated by $\mathcal{S}(y, \zeta)$ with $\zeta = \delta/16$ and produces an output $\tilde{q}_y$ satisfying*

$$\Pr[|\tilde{q}_y - q_y| \leq \delta] \quad \geq \quad 1 - \epsilon,$$

where q_y is the conditional probability defined in Eq. (13).

The algorithm $\mathcal{A}$ produces as output $\tilde{q}_y$ the fraction of samples in which $\rho_{j+1} = 0$. By the Hoeffding inequality, this sample average is close to its expectation under the distribution P generated by $\mathcal{S}$, and therefore close to its expectation q_y under the distribution μ_y.

We turn to the sampling algorithm $\mathcal{S}$. The sampler works by reduction to the Markov chain presented in Section 4. We construct a labelled multigraph $G' = (V', E')$ from the given simple graph $G = (V, E)$, where we identify V with the set $[n]$. The multigraph G' is essentially a disjoint union of two copies of G in which we identify two copies of a vertex $i \in [n]$ if $i > j$. It contains some additional edges, as explained below. Formally, the new vertex set V' is defined as:

$$\begin{aligned} V' \quad = \quad & \{(i,0), (i,1) \;:\; i \in [j]\} \\ & \cup \{i \;:\; j < i \leq n\}. \end{aligned}$$

The new edge set is defined as:

$$\begin{aligned} E' \quad = \quad & \left\{\{i,i'\}_0, \{i,i'\}_1 \;:\; i,i' > j \text{ and } \{i,i'\} \in E\right\} \\ & \cup \left\{\{(i,0),i'\}, \{(i,1),i'\} \;:\; i \leq j,\; i' > j, \text{ and } \{i,i'\} \in E\right\} \\ & \cup \left\{\{(i,0),(i',0)\}, \{(i,1),(i',1)\} \;:\; i,i' \leq j \text{ and } \{i,i'\} \in E\right\} \\ & \cup \tilde{E}, \quad \text{where} \\ \tilde{E} \quad = \quad & \{\{(i,0),(i,1)\} \;:\; i \leq j \text{ and } y_i = 1\}. \end{aligned}$$

The subscripts $0, 1$ on the first subset of edges indicate to which distinguished copy of G they correspond. The other edges are implicity labelled by the labels on the incident vertices. The last set of edges $\tilde{E}$ above are additional edges not in the disjoint union of two copies of G. They are introduced in G' so that the following property is achieved.

Lemma 5.2 *There is a one-to-one correspondence between pairs of subgraphs $X, Y \subseteq E$ of G with $\varrho_i^X = \varrho_i^Y = y_i$ for $i \in [j]$ and Eulerian subgraphs of G' which contain* all *the additional edges $\tilde{E}$.*

The proof of this lemma is straightforward and is omitted.

The edges in E' corresponding to an edge $\{i,i'\} \in E$ are assigned the same weight $\lambda_{ii'}$. Let $\lambda = \min_{\{i,i'\}\in E} \lambda_{ii'}$. The additional edges, those in $\tilde{E}$, are assigned weight $\Lambda = 2^{2m+n}\lambda^{-2n^2}$. We claim that among the Eulerian subgraphs in G', those that contain $\tilde{E}$ carry at least as much weight as those that do not.

Lemma 5.3 *Let*

$$\begin{aligned} W_0 &= \sum_{X\subseteq E' \,:\, \tilde{E}\subseteq X,\ \varrho^X = 0^n} w(X), \text{ and} \\ W_1 &= \sum_{X\subseteq E' \,:\, \tilde{E}\not\subseteq X,\ \varrho^X = 0^n} w(X). \end{aligned}$$

Then,

1. *There is a classical deterministic algorithm that decides if $W_0 > 0$ in time $\mathrm{O}((n+m)\log n)$, i.e., in linear time.*
2. *If $W_0 > 0$, then $W_0 \geq W_1$.*

Proof: The condition $W_0 > 0$ is equivalent to the existence of a set of $h(y)$ many paths using only edges in $E' - \tilde{E}$ with start and end-points only among the vertices $\{(i,u) \ : \ i \in [j],\ y_i = 1,\ u \in \{0,1\}\}$. (Recall that $h(\cdot)$ is the Hamming weight function.) This may be accomplished in linear time through, say, a variant of depth first search.

Suppose $W_0 > 0$. The number of vertices in the graph G' is at most $2n$, and the number of edges is $2m + h(y)$. The weight on all the edges in $E' - \tilde{E}$ is at most 1, since the hyperbolic tangent function is bounded by 1. Consider any Eulerian subgraph $X_0 \subseteq E'$ such that $\tilde{E} \subseteq X_0$ found by the depth first search procedure above. The subgraph contains at most $(2n-1)h(y)$ edges other than those in $\tilde{E}$—those in the $h(y)$ many paths of length at most $2n-1$ with end-points as above. Therefore the weight $W_0 \geq w(X_0) = \Lambda^{h(y)} w(X_0 - \tilde{E}) \geq \Lambda^{h(y)}\lambda^{(2n-1)h(y)}$. There are at most $2^{2m+h(y)}$ subgraphs of G'. Those counted in the sum W_1 do not contain at least one edge in $\tilde{E}$. Therefore $W_1 \leq 2^{2m+h(y)}\Lambda^{h(y)-1}$. The ratio $W_0/W_1 = \Lambda\lambda^{(2n-1)h(y)}2^{-2m-h(y)} \geq \Lambda\lambda^{(2n-1)n}2^{-2m-n} \geq 1$. ■

The sampling algorithm $\mathcal{S}$ simulates the Markov chain $\mathcal{M}$ defined in Section 4 with the instance G' constructed above to generate samples from the distribution μ_y. Its details are presented in Figure 3.

Step 2 is equivalent to finding a minimum cost S-join in the graph $\tilde{G} = (V', E' - \tilde{E})$, where $S = \{(i,0),(i,1) \ : \ i \in [j],\ y_i = 1\}$. An S-join is a subgraph of $\tilde{G}$ whose parity configuration is 1 exactly for the vertices in S. The cost assigned to an edge e for this computation is $\log(1/\lambda_e)$, where λ_e is the weight assigned to the edge e as above. Note that the edge costs are non-negative, as $\lambda_e \in (0,1)$. The cost of a subgraph is the sum of the cost of the edges in it. So the minimum cost S-join corresponds exactly to a maximum weight subgraph X_0 as required in Step 2. We can compute such an S-join efficiently.

Theorem 5.4 *The minimum cost S-join problem on a graph with N vertices, M edges, and with non-negative edge costs can be solved deterministically in $\mathrm{O}(NM + N^2\log N)$ arithmetic operations (on the edge costs).*

A proof of this fact may be found in [9, Chapter 12], in Theorem 12.9 and the remark following Corollary 12.11. We point out that the sampling algorithm $\mathcal{S}$ never fails in this step in the way

Algorithm $\mathcal{S}(y, \zeta)$.

1. Construct the labelled multigraph G' with weights as described in this section.
2. Compute an Eulerian subgraph $X_0 \subseteq E'$ such that $\tilde{E} \subseteq X_0$ with maximum weight $w(X_0)$. If no such subgraph exists, stop, and output "fail".
3. Repeat $T = 40 \log \frac{4}{\zeta}$ times, independently: simulate the Markov chain $\mathcal{M}$ with parameter $\nu = 1/|V'|$ for $\tau_{X_0}(\zeta/4)$ many steps, starting from initial state X_0, and generate the subgraphs $Y_1, Y_2, \ldots, Y_T$.
4. Let Y be the first subgraph among $\{Y_k\}$, if any, that is both Eulerian, and contains $\tilde{E}$. If no such subgraph is generated, let $Y = X_0$. Reconstruct the pair $\tilde{X}, \tilde{Y}$ of subgraphs of G that correspond to the subgraph Y of G', as per Lemma 5.2.
5. Output $\rho = \varrho^{\tilde{X}} = \varrho^{\tilde{Y}}$.

Figure 3: Details of the algorithm that samples parity configurations ρ from the distribution μ_y.

we use it, as we only invoke it when the distribution μ_y is well-defined, in particular when it has non-empty support.

In Step 3 of the algorithm, we show that a sample of interest is generated except with small probability, and when it is generated, it is close to the desired distribution.

Lemma 5.5 *Let $\zeta \leq 1/10$. Except with probability $\zeta/4$, at least one of the subgraphs $\{Y_k\}$ generated in Step 3 in Algorithm $\mathcal{S}(y, \zeta)$ is Eulerian and contains $\tilde{E}$. When such a subgraph is generated, its distribution is at most $3\zeta/4$ away from μ_y in ℓ_1 norm. Therefore the distribution of the subgraph Y from Step 4 is at most ζ away from μ_y.*

Proof: The distribution generated by the Markov chain simulation differs from the stationary distribution π_{S} by at most $\zeta/4$.

An argument as in the proof of Lemma 4 of [6] shows that under the distribution π_{S} of the Markov chain $\mathcal{M}$, the probability that a subgraph is Eulerian is at least $1/10$. Lemma 5.3 shows that with at least half this probability, the Eulerian subgraphs contain the edges $\tilde{E}$. Therefore the probability of such a subgraph under the distribution generated by the Markov chain simulation is at least $1/20 - \zeta/4 \geq 1/40$, as long as $\zeta \leq 1/10$. Thus, the probability that none of the T independent samples is a subgraph of interest is at most $(1 - 1/40)^T \leq \zeta/4$. Given that a sample of interest is generated, its distribution differs from μ_y by at most $(\zeta/4)/(1 - \zeta/4) \leq 3\zeta/4$. The rest of the claim follows. ■

We turn to the time complexity of the sampling alorithm $\mathcal{S}$.

Theorem 5.6 *The algorithm $\mathcal{S}$ with inputs y, ζ runs in time of the order of*

$$n^4 (m + n)^2 \left(\log \frac{1}{\zeta} + m \right) \log \frac{1}{\zeta}$$

arithmetic operations on $\mathrm{O}(n^2 \log \frac{1}{\lambda})$*-bit numbers.*

Proof: We compute the time taken by the algorithm step by step. The first step takes linear time in the size of the input G. The second involves $\mathrm{O}(nm + n^2 \log n)$ arithmetic operations on numbers of size $\log(1/\lambda)$ in bits. By Theorems 4.1 and 4.2, the third step involves of the order of

$$40 \log \frac{4}{\zeta} \times 2(2m+n)^2(2n)^4 \times \ln \frac{1}{\zeta\, \pi_{\mathrm{S}}(X_0)}$$

arithmetic operations on numbers represented with at most $\log \Lambda = \mathrm{O}(2m + n + n^2 \log(1/\lambda))$ bits. Note that there are at most 2^{2m} Eulerian subgraphs in G' that contain $\tilde{E}$. The one we chose as our initial state, viz., X_0, has the largest weight among these, and therefore probability at least 2^{-2m} of the net probability of such subgraphs (under the stationary distribution π_{S}). As argued in Lemma 5.5, this probability is at least 1/20. Therefore $\log(1/\pi_{\mathrm{S}}(X_0)) \leq 10m$.

The fourth step takes time of the order of $\log(1/\zeta)(n+m)\log n$.

The third step dominates and gives us the run-time. ■

It follows that the algorithm we present for approximating the Gibbs superposition is polynomial in its run-time. A precise statement of its time complexity can be inferred from the intermediate results in this manuscript and will be included shortly.

References

[1] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[2] Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[3] Richard P. Brent. Multiple-precision zero-finding methods and the complexity of elementary function evaluation. In Joseph F. Traub, editor, *Analytic Computational Complexity*, pages 151–176. Academic Press, New York, 1976. Proceedings of the Symposium on Analytic Computational Complexity, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania, April 7–8, 1975.

[4] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. Technical Report arXiv:quant-ph/0208112v1, Arxiv.org, http://www.arxiv.org/, 2002.

[5] Aram W. Harrow, Benjamin Recht, and Isaac L. Chuang. Efficient discrete approximations of quantum gates. *Journal of Mathematical Physics*, 43, 2002. Article no. 4445. Also available as arXiv:quant-ph/0111031v3.

[6] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM Journal on Computing*, 22:1087–1116, 1993.

[7] Mark Jerrum and Alistair Sinclair. The Markov chain Monte Carlo method: An approach to approximate counting and integration. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 12, pages 482–520. PWS Publishing, Boston, 1996.

[8] Phillip Kaye and Michele Mosca. Quantum networks for generating arbitrary quantum states. In *Proceedings of the International Conference on Quantum Information*, Rochester, New York, USA, 2001. Also available as preprint arXiv:quant-ph/0407102v1.

[9] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer, Berlin Heidelberg, fourth edition, 2008.

[10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.

[11] Arnold Schönhage and Volker Strassen. Schnelle multiplikation großer zahlen. *Computing*, 7:281–292, 1971.

[12] Alistair Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, Probability and Computing*, 1:351–370, 1992.

[13] Umesh V. Vazirani. On the power of quantum computation. *Philosophical Transactions of the Royal Society of London, Series A*, 356:1759–1768, 1998.

[14] F.Y. Wu. The potts model. *Reviews of Modern Physics*, 54(1):235–268, January 1982.