Quantum Machine Intelligence (2024) 6:73
https://doi.org/10.1007/s42484-024-00209-5

RESEARCH ARTICLE

Quantum algorithms for weighted constrained sampling and weighted model counting

Fabrizio Riguzzi[1]

Received: 4 December 2023 / Accepted: 24 October 2024

Abstract

We consider the problems of weighted constrained sampling and weighted model counting, where we are given a propositional formula and a weight for each world. The first problem consists of sampling worlds with a probability proportional to their weight given that the formula is satisfied. The latter is the problem of computing the sum of the weights of the models of the formula. Both have applications in many fields such as probabilistic reasoning, graphical models, statistical physics, statistics, and hardware verification. In this article, we propose quantum weighted constrained sampling (QWCS) and quantum weighted model counting (QWMC), two quantum algorithms for performing weighted constrained sampling and weighted model counting, respectively. Both are based on the quantum search/quantum model counting algorithms that are modified to take into account the weights. In the black box model of computation, where we can only query an oracle for evaluating the Boolean function given an assignment, QWCS requires $O(2^{\frac{n}{2}} + 1/\sqrt{\mathrm{WMC}})$ oracle calls, where n is the number of Boolean variables and WMC is the normalized between 0 and 1 weighted model count of the formula, while a classical algorithm has a complexity of $\Omega(1/\mathrm{WMC})$. QWMC takes $\Theta(2^{\frac{n}{2}})$ oracle calss, while classically the best complexity is $\Theta(2^n)$, thus achieving a quadratic speedup.

Keywords Quantum search · Quantum counting · Weighted model counting · Weighted constrained sampling · Most probable explanation · Maximum a posteriori

1 Introduction

Given a Boolean formula and functions assigning weights to assignments of values to the Boolean variable, we consider the problems of weighted constrained sampling (WCS) and weighted model counting (WMC). The first, also called distribution-aware sampling (Chakraborty et al. 2014), involves sampling assignments to the Boolean variables with a probability proportional to their weight given that the formula is satisfied. The latter (Sang et al. 2005) consists in computing the sum of the weights of the models of the formula, i.e., the weighted model count.

WCS has important applications in a variety of domains, including statistical physics (Jerrum and Sinclair 1996), statistics (Madras and Piccioni 1999), hardware verification (Naveh et al. 2006), and probabilistic reasoning, where it can be used to solve the problem of most probable explanation (MPE) and maximum a posteriori (MAP). MPE (Sang et al. 2007) involves finding an assignment to all variables that satisfies a Boolean formula and has the maximum weight. The related MAP problem means finding an assignment of a subset of the variables such that the sum of the weights of the models of the formula that agree on the assignment is maximum.

WMC was successfully applied, among others, to the problem of performing inference in graphical models (Chavira and Darwiche 2008; Sang et al. 2005). In particular, other graphical model inference algorithms (Lauritzen and Spiegelhalter 1988; Zhang and Poole 1996; Dechter 1999; Darwiche 2001) take time $\Theta(n2^w)$, where n is the number of variables and w is the treewidth (Bodlaender et al. 1993) of the network, a measure of the complexity of the network. WMC instead takes time $O(n2^w)$, i.e., exponential in the treewidth in the worst case (Chavira and Darwiche 2008). This is possible because WMC exploits the structure of graphical models in the form of context-specific independence and determinism.

✉ Fabrizio Riguzzi
fabrizio.riguzzi@unife.it

1 Department of Mathematics and Computer Science, University of Ferrara, Via Machiavelli 30, Ferrara 44121, Italy

Published online: 05 November 2024

In this paper, we propose to use quantum computing for performing WCS and WMC. We call QWCS and QWMC the resulting algorithms. The first is based on a quantum search using Grover's algorithm (Grover 1996a, b, 1997). The latter on quantum model counting (Boyer et al. 1998; Brassard et al. 1998). We modified these algorithms to take into account weights. In particular, the proposed algorithms modify the algorithms for unweighted search and counting by replacing the Hadamard gates with rotation gates, with the rotations depending on the weights.

QWCS and QWMC work under a black box computation model where we don't know anything about the propositional formula, we only have the possibility of querying an oracle giving the value of the formula for an assignment of the propositional variables. QWCS take $O(2^{\frac{n}{2}} + 1/\sqrt{\mathrm{WMC}})$ oracle calls to solve WCS with a probability of at least $\sqrt{\frac{1}{2}} \approx 0.707$, where WMC is the weighted model count normalized between 0 and 1 (cf. Theorem 10) and n is the number of variables, while any classical algorithm takes $\Omega(1/\mathrm{WMC})$ oracle calls.

QWMC takes $\Theta(2^{\frac{n}{2}})$ oracle calls (cf. Theorem 14), to bound the error to $2^{-\frac{n+1}{2}}$ with probability $\frac{11}{12}$, while any classical algorithm takes $\Theta(2^n)$ oracle calls, thus achieving a quadratic speedup.

Existing approaches for WMC (knowledge compilation (Darwiche and Marquis 2002; Lagniez and Marquis 2017; Huang et al. 2006), backtracking search (Bacchus et al. 2009), reduction to unweighted counting (Chakraborty et al. 2015), see also (Fichte et al. 2021) for a recent competition among solvers) assume a white box computation model where the formula is known and can be manipulated.

QWMC may be useful for probabilistic inference from models with high treewidth: supposing the cost of implementing the oracle is linear in n, if the treewidth is larger than half the number of variables, then QWMC performs better than classical inference algorithms in the worst case because they take time $O(n2^w)$ while QWMC takes time $\Theta(n2^{\frac{n}{2}})$.

QWMC can also be used as a subroutine for probabilistic inference systems over graphical models. For example, in the junction tree algorithm (Shenoy and Shafer 1990; Lauritzen and Spiegelhalter 1988), it can be used after the probabilities are propagated in the tree to compute the marginals of the variables in tree nodes.

This article extends (Riguzzi 2020) by adding the QWCS algorithm. Moreover, it fixes an error in the QWMC algorithm.

The paper is organized as follows. Section 2 presents the WCS and WMC problems. Section 3 describes quantum search and Section 4 compares it with classical algorithms in terms of cost. Section 5 presents QWCS, whose complexity is discussed in Section 6 and compared with classical algorithms. The algorithm for quantum counting is discussed in Section 7. Section 8 compares the complexity of the algorithm to that of classical algorithms. Section 9 illustrates the QWMC algorithm and Section 10 compares it with classical algorithms. Related work is described in Section 11. Finally, Section 12 presents a discussion of the work and Section 13 concludes the paper.

The Supplementary material includes a brief introduction to the quantum computing concepts used in the paper, the code for the quantum algorithms in Q# and Qiskit, and the result of a run of the algorithms applied to the problem of WMC, MPE, and MAP.

2 Weighted constrained sampling and weighted model counting

Let X be a vector of n Boolean variables $[X_1, \ldots, X_n]$ and let x be an assignment of values to X, i.e., a vector of n Boolean values $[x_1, \ldots, x_n]$. We call x a *world*, a *configuration*, or an *assignment*. Consider a propositional logic formula ϕ over X built from the standard propositional connectives $[\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \oplus]$ (not, and, or, imply, iff, xor). If an assignment x of variables X makes formula ϕ evaluate to true, we write $x \models \phi$ and we say that x *satisfies* ϕ or that x is a *model* of ϕ. Let us call M the number of models. We can also see ϕ as a function from $\mathbb{B}^n$ to $\mathbb{B}$, where $\mathbb{B} = \{0, 1\}$, and express that x makes ϕ evaluate to true by $\phi(x) = 1$.

The satisfiability problem (SAT) is the problem of deciding whether a formula ϕ has a model, i.e., whether $M > 0$. The functional satisfiability problem (FSAT) is defined as: given a formula ϕ, return an assignment x that is a model of ϕ or answer NO if no such assignment exists. The problem of *model counting* (#SAT) (Gomes et al. 2009) is the problem of computing M.

The problem of *constrained sampling* (CS) (Meel et al. 2016) consists of sampling configurations x with a uniform distribution given that $\phi(x) = 1$. In other words, we want to sample following this distribution

$$P(x) = \begin{cases} \frac{1}{M} & \text{if } \phi(x) = 1 \\ 0 & \text{if } \phi(x) = 0 \end{cases}$$

In some cases, we want to sample a configuration q of a subset Q of the variables X with a probability proportional to the number of models in which q can be extended. Supposing, without loss of generality, that Q is equal to the first l bits, then we want to sample configurations q with probability

$$P(q) = \sum_{y:qy \models \phi} \frac{1}{M}$$

where qy is a world where variables in Q take value q and variables in $Y = X \setminus Q$ take value y and the sum is over all values y of Y such that $qy \models \phi$.

In this article, we are concerned with weighted Boolean formulas which are pairs (ϕ, W) where ϕ is a Boolean formula over variables X and $W : \mathbb{B}^n \to R^{\geq 0}$ is a weight function over the configurations of X, i.e., it assigns a weight $W(x)$ to a configuration x, that we abbreviate with W_x. Then, the *weighted model count* (WMC) is defined as the sum of the weights of all satisfying assignments:

$$\mathrm{WMC}(\phi, W) = \sum_{x:x\models\phi} W_x.$$

When ϕ and W are clear from the context, we simply indicate WMC(ϕ, W) with WMC. *Weighted model counting* (WMC) (Chavira and Darwiche 2008) is the problem of computing WMC.

We restrict our analysis to factorized weight functions, i.e., weight functions expressible as a product of weights over the literals built on X (Boolean variables or their negation). Seeing a configuration x as a set of literals (e.g., $[0, 1, 0, 1] = [\neg X_1, X_2, \neg X_3, X_4]$, we can compute the weights W_x with a function $w : L \to R^{\geq 0}$, where L is the set of literals, such that

$$W_x = \prod_{l\in x} w(l).$$

An important special case is that in which $w(X_i) + w(\neg X_i) = 1$, where the weights can be considered as the probabilities of the Boolean literals of being true and WMC is then the probability that ϕ takes value 1 assuming that the Boolean variables are independent random variables.

Weighted constrained sampling (WCS) or distribution-aware sampling (Chakraborty et al. 2014) is the problem of sampling a configuration x with a probability proportional to its weight given that the formula is satisfied, i.e.:

$$P(x) = \begin{cases} \propto W_x & \text{if } \phi(x) = 1 \\ 0 & \text{if } \phi(x) = 0 \end{cases} = \begin{cases} \frac{W_x}{\sum_{x:\phi(x)=1} W_x} & \text{if } \phi(x) = 1 \\ 0 & \text{if } \phi(x) = 0 \end{cases} = \begin{cases} \frac{W_x}{\mathrm{WMC}} & \text{if } \phi(x) = 1 \\ 0 & \text{if } \phi(x) = 0 \end{cases} \tag{1}$$

It is a special case of the problem of sampling a set of query variables Q with a probability proportional to the sum of the weights of the models in which q can be extended:

$$P(q) = \frac{\sum_{y:\phi(qy)=1} W_{qy}}{\sum_{q,y:\phi(qy)=1} W_{qy}} = \frac{\sum_{y:\phi(qy)=1} W_{qy}}{\mathrm{WMC})} \tag{2}$$

WCS can play a role in the following problems. The *most probable explanation* (MPE) (Sang et al. 2007) problem involves finding the model that has the maximum weight. The *maximum a posteriori* (MAP) problem means finding an assignment of a subset of the variables such that the sum of the weights of the models that agree on the assignment is maximum.

Given a formula ϕ and a weight function $w : L \to R^{\geq 0}$, the most probable state (most probable explanation, MPE) of the variables is

$$\mathrm{MPE}(\phi, w) = \operatorname*{argmax}_{x:x\models\phi} W_x$$

Let MPE$^w(\phi, w)$ be the weight of MPE(ϕ, w), i.e.,

$$\mathrm{MPE}^w(\phi, w) = \max_{x:x\models\phi} W_x.$$

Given formula ϕ, a weight function $w : L \to R^{\geq 0}$ and a set of query variables Q, the most probable state of the query variables (Maximum A Posteriori, MAP) is

$$\mathrm{MAP}_Q(\phi, w) = \operatorname*{argmax}_q \sum_{y:qy\models\phi} W_{qy}$$

Let MAP$^w_Q(\phi, w)$ be the sum of the weights of the models that agree on the assignment MAP$_Q(\phi, w)$:

$$\mathrm{MAP}^w_Q(\phi, w) = \max_q \sum_{y:qy\models\phi} W_{qy}.$$

Clearly, MPE is a special case of MAP when $Q = X$.

Example 1 Let us consider an example inspired by the sprinkler problem of Pearl (1988): we have three Boolean variables, S, R, and W representing propositions “the sprinkler was on,” “it rained last night,” and “the grass is wet,” respectively. We know that: if the sprinkler was on, the grass is wet ($S \to W$); if it rained last night, the grass is wet ($R \to W$); and the sprinkler being on and rain last night cannot be true at the same time ($S \wedge R \to$). The formula for this problem is:

$$\phi = (\neg S \vee W) \wedge (\neg R \vee W) \wedge (\neg S \vee \neg R).$$

Suppose the weights of literals are $w(S) = 0.55$, $w(\neg S) = 0.45$, $w(R) = 0.3$, $w(\neg R) = 0.7$, $w(W) = 0.7$, and $w(\neg W) = 0.3$. Table 1 shows the worlds together with the weight of each world. The WMC of ϕ is thus WMC$(\phi, w) = 0.0945+0.2205+0.0945+0.2695 = 0.679$.

The MPE is MPE$(\phi, w) = [1, 0, 1]$ with MPE$^w(\phi, w) = 0.2695$. The MAP of query variables S and W is MAP$_{SW}$

Table 1 Worlds for Example 1

S	R	W	ϕ	$weight$
0	0	0	1	$0.45 \cdot 0.7 \cdot 0.3 = 0.0945$
0	0	1	1	$0.45 \cdot 0.7 \cdot 0.7 = 0.2205$
0	1	0	0	$0.45 \cdot 0.3 \cdot 0.3 = 0.0405$
0	1	1	1	$0.45 \cdot 0.3 \cdot 0.7 = 0.0945$
1	0	0	0	$0.55 \cdot 0.7 \cdot 0.3 = 0.1155$
1	0	1	1	$0.55 \cdot 0.7 \cdot 0.7 = 0.2695$
1	1	0	0	$0.55 \cdot 0.3 \cdot 0.3 = 0.0495$
1	1	1	0	$0.55 \cdot 0.3 \cdot 0.7 = 0.1155$

$(\phi, w) = [0, 1]$ with $\mathrm{MAP}^{w}_{SW}(\phi, w) = 0.2205 + 0.0945 = 0.315$.

So the most probable state of S is 1 when we look for the overall MPE state and is 0 when we look for the most probable state of S and W.

WCS can be used to obtain an approximate algorithm for performing MPE and MAP. Given an algorithm for WCS, we can execute it for o iterations. At each iteration, we obtain a sample q. After o iterations, we return the value that was found most frequently.

In fact, $P(q)$ of Eq. (2) is a categorical distribution. Executing the algorithm o times is equivalent to extracting a sample d from a multinomial distribution $P(d; o, p)$ with o trials, set of events $\{q|\exists y, \phi(qy) = 1\}$ and event probabilities $P(q)$. Let us rename the elements of the set of events as $q_1, \ldots, q_k$ and the event probabilities as $[p_1, \ldots, p_k]$ and let us reorder them so that the event probabilities are in ascending order: $p_1 \leq p_1 \leq \ldots \leq p_k$. So q_k is $\mathrm{MAP}_Q(\phi, w)$. Let d_i be the count associated to even q_i in the sample d.

We obtain a correct solution of the overall algorithm if $d_k > d_1 \wedge \ldots \wedge d_k > d_{k-1}$. The probability that this event happens in a sample from a multinomial distribution has been studied in Gupta and Nagel (1967). The authors consider a worst case scenario where the event probabilities are all equal to a common value p except for the last one which is equal to Ap with $A \geq 1$ so the event probabilities are $[p, \ldots, p, Ap]$. The authors do not present a closed formula for computing

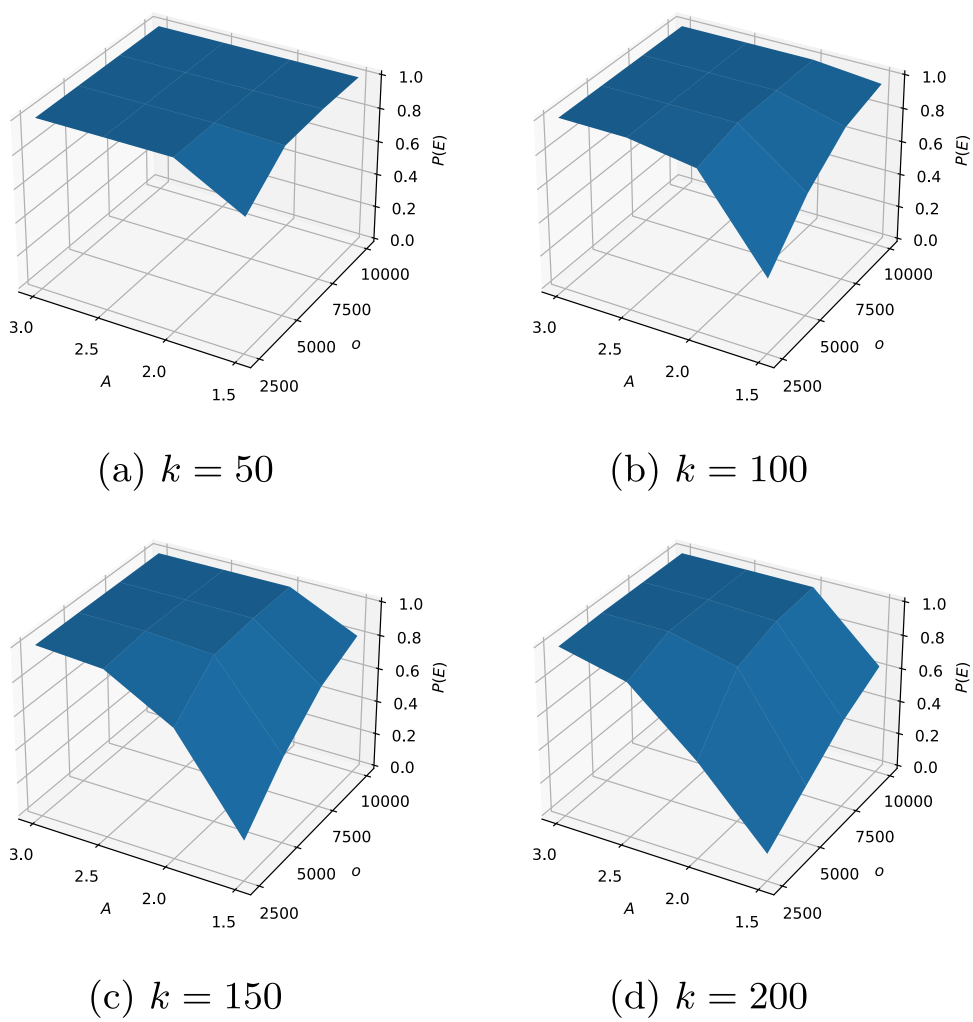

Fig. 1 Probability of event E in a set of samples from a multinomial random variable with distribution $[p, \ldots, p, Ap]$. k is the number of values of the random variable, o is the number of trials and A is the multiplier of the probability of the most probabile value

the probability of the event $E = (d_k > d_1 \wedge \ldots \wedge d_k > d_{k-1})$ but tabulate the probability $P(E)$ for various values of A, o, and k.

Unfortunately, the tables in Gupta and Nagel (1967) do not include values of interests for us so we performed a simulation using probabilistic logic programming (Riguzzi 2022) and the MCINTYRE system in particular (Riguzzi 2013), for the set of values: $k = 50, 100, 150, 200$, $o = 2500, 5000, 7500, 10000$ and $A = 1.5, 2, 2.5, 3$. The results are shown in Fig. 1. We can see that the probability of the event E, and so of the success of the algorithm, goes rapidly to 1 for increasing values of o and A.

3 Quantum search

CS can be seen as a search problem: find a satisfying assignment of bits. A quantum algorithm for solving this problem was proposed in (Grover 1996a, b, 1997). Here, we follow the exposition of Nielsen and Chuang (2010) and Hirvensalo (2013).

We assume we have a black box quantum circuit that evaluates ϕ, called an *oracle* O, that is such that

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

i.e., the oracle marks solutions to the search problems by changing the sign of the state. The oracle may use extra ancilla bits to do so.

Figure 2 shows the circuit performing quantum search operating on an n-qubit register X and the oracle ancilla qubits $Ancilla$. All qubits of register X start in state $|0\rangle$.

The circuit includes a gate G that is called the *Grover operator* and is implemented as shown in Fig. 3.

The first gate of the search circuit applies the H gate to each qubit in register X. Since all qubits in register X start as $|0\rangle$ and the effect of H is to transform $|0\rangle$ to the state $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$, then register X is transformed to

$$\begin{aligned}
|\psi\rangle &= \tfrac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes \tfrac{|0\rangle+|1\rangle}{\sqrt{2}} \cdots \otimes \tfrac{|0\rangle+|1\rangle}{\sqrt{2}} \\
&= \tfrac{|00\rangle+|01\rangle+|10\rangle+|11\rangle}{\sqrt{2^2}} \cdots \otimes \tfrac{|0\rangle+|1\rangle}{\sqrt{2}} \\
&= \tfrac{|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle+|101\rangle+|110\rangle+|111\rangle}{\sqrt{2^3}} \cdots \otimes \tfrac{|0\rangle+|1\rangle}{\sqrt{2}} \\
&= \tfrac{1}{N^{1/2}} \textstyle\sum_{x=0}^{N-1} |x\rangle
\end{aligned}$$

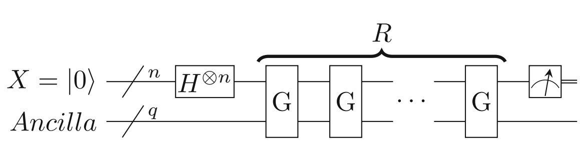

Fig. 2 Quantum search algorithm

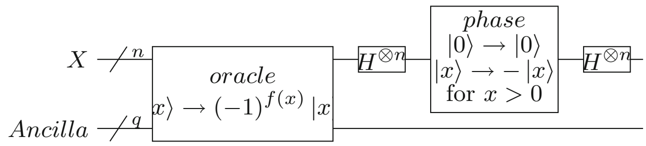

Fig. 3 Grover operator G

where $N = 2^n$. This state is also called the *uniform superposition state*.

The Grover operator can be written as

$$G = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} O$$

Since $H^\dagger = H$, G can be rewritten as

$$\begin{aligned}
G &= H^{\otimes n}(2|0\rangle\langle 0| - I)(H^{\otimes n})^\dagger O \\
&= \left(2H^{\otimes n}|0\rangle\langle 0|(H^{\otimes n})^\dagger - H^{\otimes n} I (H^{\otimes n})^\dagger\right) O \\
&= (2|\psi\rangle\langle\psi| - I)O
\end{aligned}$$

We now show that the Grover operator is a rotation.

Lemma 1 *The Grover operation applied to the uniform superposition state* $|\psi\rangle$ *rotates it by angle* $2 \arcsin \sqrt{M/N}$ *where* M *is the number of solutions of* $\phi(x) = 1$.

Proof Consider the two states

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{x:\phi(x)=0} |x\rangle$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x:\phi(x)=1} |x\rangle.$$

These two states are orthonormal because they do not share any computational basis state. The uniform superposition state $|\psi\rangle$ can be written as a linear combination of $|\alpha\rangle$ and $|\beta\rangle$:

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle$$

so $|\psi\rangle$ belongs to the plane defined by $|\alpha\rangle$ and $|\beta\rangle$. In this plane, the effect of the oracle operation O is to perform a reflection about the vector α because $O(|\alpha\rangle + |\beta\rangle) = |\alpha\rangle - |\beta\rangle$, see Fig. 4.

The other component of the Grover operator, $2|\psi\rangle\langle\psi| - I$, also performs a reflection in the plane defined by $|\alpha\rangle$ and $|\beta\rangle$, about the vector $|\psi\rangle$. The overall effect is that of a rotation (Aharonov 1999). If we define $\cos\theta = \sqrt{(N-M)/N}$ and $\sin\theta = \sqrt{M/N}$, then $|\psi\rangle = \cos\theta|\alpha\rangle + \sin\theta|\beta\rangle$.

From Fig. 4, we can see that the rotation applied by G is by angle 2θ, so

$$G|\psi\rangle = \cos 3\theta|\alpha\rangle + \sin 3\theta|\beta\rangle$$

□

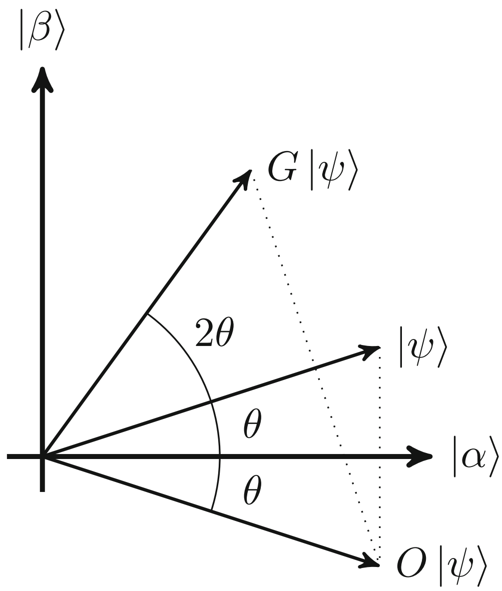

Fig. 4 Visualization of the effect of Grover operator (Figure 6.3 from Nielsen and Chuang (2010))

Repeated applications of G take the state to

$$G^k|\psi\rangle = \cos(2k+1)\theta|\alpha\rangle + \sin(2k+1)\theta|\beta\rangle.$$

These rotations bring the state of the system closer to $|\beta\rangle$. If we perform the right number of rotations, a measurement in the computational basis will produce one of the outcomes superposed in $|\beta\rangle$, i.e., a solution to the search problem, with a non-zero probability. Moreover, since the weights of the computational basis states superimposed in $|\beta\rangle$ are all equal, then the measurement produces one of solutions with equal probability, solving CS when $Q = X$.

Theorem 1 *(Theorem 5.2.1 in Hirvensalo (2013)) Let $\phi : \mathbb{B}^n \to \mathbb{B}$ be such that there are M elements $x \in \mathbb{B}^n$ satisfying $\phi(x) = 1$. Assume that $0 < M \leq \frac{3}{4}N$. and let $\theta \in [0, \pi/3]$ be chosen such that $sin^2\theta = \frac{M}{N} \leq \frac{3}{4}$. After $\left\lfloor \frac{\pi}{4\theta} \right\rfloor$ iterations of G on an initial superposition*

$$\frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle$$

the probability of seeing a solution is at least $\frac{1}{4}$.

Proof The probability P of seeing a solution is given by

$$P = \sin^2((2k+1)\theta)$$

To maximize P, we must find the least positive integer k so that P is as close to 1 as possible. This implies that

$$(2k+1)\theta = \frac{\pi}{2}$$

so

$$k = \frac{\pi}{4\theta} - \frac{1}{2}$$

. Using the approximation $\theta^2 \approx \sin^2\theta = \frac{M}{N}$ we have that

$$\theta \approx \sqrt{\frac{M}{N}}$$

and after

$$R = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rfloor$$

rotations, P is close to 1. Let us now compute P. We can observe that

$$R = \frac{\pi}{4\theta} - \frac{1}{2} + \delta$$

with $|\delta| \leq \frac{1}{2}$. Therefore

$$(2R+1)\theta = \frac{\pi}{2} + 2\delta\theta$$

The distance between $(2R+1)\theta$ and $\frac{\pi}{2}$ is thus $|2\delta\theta|$ and, since $\theta \in [0, \frac{\pi}{3}]$, $|2\delta\theta| \leq \frac{\pi}{3}$. P then becomes

$$P = \sin^2((2R+1)\theta) \geq \sin^2\left(\frac{\pi}{2} - \frac{\pi}{3}\right) = \frac{1}{4}$$

□

Consider now the cases $M > \frac{3}{4}N$ and $M = 0$. In the first case, we can guess a solution and the probability that it is correct is $\frac{3}{4}$. In the latter case G does not alter the initial superposition.

This leads to Algorithm 1, where lines 6–8 are implemented by the quantum circuit of Fig. 2.

Algorithm 1 Grover's algorithm

Require: A blackbox function $\phi : \mathbb{B}^n \to \mathbb{B}$ and $M = |\{x \in \mathbb{B}^n | \phi(x) = 1\}|$
Ensure: $\phi(x) = 1$
1: **if** $M > \frac{3}{4}N$ **then**
2: Choose $x \in \mathbb{B}^n$ with uniform probability
3: **return** x
4: **else**
5: $R \leftarrow \left\lfloor \frac{\pi}{4}\sqrt{\frac{M}{N}} \right\rfloor$
6: Prepare the initial superposition $\frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle$
7: Apply operator G R times
8: Measure to get $x \in \mathbb{B}^n$
9: **return** x
10: **end if**

Theorem 2 *Algorithm 1 solves FSAT with probability at least $\frac{1}{4}$ and $O(\sqrt{N})$ queries to ϕ when M is considered constant.*

Proof If $M \geq \frac{3}{4}N$, line 2 finds a solution with probability at least $\frac{3}{4}$.

Otherwise, in each application of G, we query ϕ and the number of applications is $R \leq \frac{\pi}{4}\sqrt{\frac{N}{M}}$, so $R \in O(\sqrt{N})$ □

A notable value of M is $N/4$ for which $k = \frac{\pi/3}{2\pi/6} = 1$ and $R = 1$, so one rotation is enough to find a solution with certainty. For $N/4 \leq M < N/2$, we have that $R = 1$ and, for decreasing M, R grows.

We can use this as a probabilistic algorithm: if we have an algorithm A that returns the required x with probability p, with $0 < p \leq 1$, and returns "no answer" with probability $1-p$, we can obtain an algorithm that fails to find the solution with a probability smaller than a given constant $\epsilon > 0$. This is achieved by running A multiple times: after o executions of A, the probability of not finding a solution is $(1-p)^o$. Since $(1-p)^o \leq e^{-op}$ and we want to achieve $(1-p)^o \leq \epsilon$, picking o such that $o = -\frac{\log \epsilon}{p}$ guarantees that the algorithm fails to find a solution with probability smaller than ϵ.

In the case of the Grover algorithm, this means running it for $-4\log\epsilon$ times to obtain a solution with probability smaller than ϵ while still requiring $O(\sqrt{N})$ calls to the oracle, when M and ϵ are held constant.

When the number of solutions M is not known, Algorithm 2 can be used (Hirvensalo 2013). Let us first present two lemmas.

Algorithm 2 Quantum search when the number of solutions is not known

Require: A blackbox function $\phi : \mathbb{B}^n \to \mathbb{B}$
Ensure: $\phi(x) = 1$
1: Pick an element $x \in \mathbb{B}$
2: **if** $\phi(x) = 1$ **then**
3: **return** x
4: **else**
5: $m = \left\lfloor \sqrt{N} \right\rfloor + 1$
6: Choose an integer R uniformly in $[0, m-1]$
7: Prepare the initial superposition $\frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle$
8: Apply operator G R times
9: Measure to get $x \in \mathbb{B}^n$
10: **return** x
11: **end if**

Lemma 2 *(Lemma 5.3.1 in Hirvensalo (2013)) For any real α and any positive integer m*

$$\sum_{R=0}^{m-1} \cos((2R+1)\alpha) = \frac{\sin(2m\alpha)}{2\sin\alpha}$$

Lemma 3 *(Boyer et al. 1998) Let $\phi : \mathbb{B}^n \to \mathbb{B}$ be such that there are M elements $x \in \mathbb{B}^n$ satisfying $\phi(x) = 1$. Assume that $M \leq \frac{3}{4}N$. And let $\theta \in [0, \pi/3]$ be defined by $\sin^2\theta = \frac{M}{N}$. Let m be any positive integer and $R \in [0, m-1]$ chosen with uniform distribution. If G is applied to initial superposition*

$$\frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle$$

R times, then the probability of seeing a solution is

$$P_m = \frac{1}{2} - \frac{\sin(4m\theta)}{4m\sin(2\theta)}$$

Proof After R iterations of G, the probability of seeing a solution is $\sin^2((2R+1)\theta)$. So if R is chosen uniformly in $[0, m-1]$, the probability of seeing a solution is

$$\begin{aligned} P_m &= \frac{1}{m}\sum_{R=0}^{m-1} \sin^2((2R+1)\theta) \\ &= \frac{1}{2m}\sum_{R=0}^{m-1} (1 - \cos((2R+1)2\theta)) \\ &= \frac{1}{2} - \frac{\sin(4m\theta)}{4m\sin(2\theta)} \end{aligned}$$

due to Lemma 2 □

Now, we can present the main theorem.

Theorem 3 *(Hirvensalo 2013) Algorithm 2 solves CS with probability at least $\frac{1}{4}$ and $O(\sqrt{N})$ queries to ϕ.*

Proof If $m \geq \frac{1}{\sin(2\theta)}$, then

$$\sin(4m\theta) \leq 1 = \frac{1}{\sin(2\theta)}\sin(2\theta) \leq m\sin(2\theta)$$

so

$$\frac{\sin(4m\theta)}{4m\sin(2\theta)} \leq \frac{1}{4}.$$

By Lemma 3, then $P_m \geq \frac{1}{4}$. Given that $0 < M \leq \frac{3}{4}N$ then

$$\frac{1}{\sin(2\theta)} = \frac{1}{2\sin\theta\cos\theta} = \frac{N}{2\sqrt{M(N-M)}} \leq \sqrt{\frac{N}{M}} \leq \sqrt{N}.$$

So, if we pick $m \geq \sqrt{N}$, we ensure that $m \geq \frac{1}{\sin(2\theta)}$. Choosing $m = \left\lfloor \sqrt{N} \right\rfloor + 1$, we have that $P_m \geq \frac{1}{4}$ and the number of applications of G is $O(\sqrt{N})$. □

Again, the probability of success can be made arbitrary close to 1 by repeating the algorithm.

An alternativa approach to make sure that $\frac{M}{N} \leq \frac{3}{4}$, it is to consider an extra qubit X_{n+1}, with $X' = [X_1, \ldots, X_{n+1}]$ and $x' = [x_1, \ldots, x_{n+1}]$ being two vectors of Boolean variables and values respectively, and defining a new formula ϕ' that is true only if both ϕ and X_{n+1} are true, i.e., $\phi' = \phi \wedge X_{n+1}$. This

Algorithm 3 Alternative quantum search when the number of solutions is not known

Require: A blackbox function $\phi : \mathbb{B}^n \to \mathbb{B}$
Ensure: $\phi(x) = 1$
1: $m = \left\lfloor \sqrt{\frac{N}{2}} \right\rfloor + 1$
2: Choose an integer R uniformly in $[0, m-1]$
3: Prepare the initial superposition $\frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle$
4: Apply operator G R times
5: Measure to get $x \in \mathbb{B}^n$
6: **return** x

leaves M unchanged but multiplies N by 2 so that $\sin^2\theta = \frac{M}{2N}$. We thus obtain Algorithm 3.

Lemma 4 *Let $\phi : \mathbb{B}^n \to \mathbb{B}$, ϕ' be defined as $\phi' = \phi \wedge X_{n+1}$ and be such that there are M elements $x \in \mathbb{B}^n$ satisfying $\phi(x) = 1$. Let $\theta \in [0, \pi/4]$ be defined by $sin^2\theta = \frac{M}{2N}$. Let m be any positive integer and $R \in [0, m-1]$ chosen with uniform distribution. If G is applied to initial superposition*

$$\frac{1}{\sqrt{2N}} \sum_{x'=0}^{2N-1} |x'\rangle$$

R times, then the probability of seeing a solution is

$$P_m = \frac{1}{2} - \frac{\sin(4m\theta)}{4m\sin(2\theta)}$$

Proof The proof is the same as than of Lemma 3. □

Theorem 4 *Algorithm 3 solves CS with probability at least $\frac{1}{4}$ and $O(\sqrt{N})$ queries to ϕ.*

Proof We can follow the proof of Theorem 3 with N replaced by $2N$. By Lemma 4, if $m \geq \frac{1}{\sin(2\theta)}$, then $P_m \geq \frac{1}{4}$. Since $\sin\theta = \sqrt{\frac{M}{2N}}$ and $\frac{M}{2N} \leq \frac{1}{2}$, then $\theta \in [0, \frac{\pi}{4}]$ and we have that

$$\frac{1}{\sin(2\theta)} = \frac{1}{2\sin\theta\cos\theta} = \frac{2N}{2\sqrt{M(2N-M)}} \leq \sqrt{\frac{N}{2M}} \leq \sqrt{\frac{N}{2}}.$$

So choosing $m = \left\lfloor \sqrt{\frac{N}{2}} \right\rfloor + 1$, we have that $P_m \geq \frac{1}{2}$ and the number of applications of G is $O(\sqrt{N})$. □

In general, adding other extra qubits reduces the number of applications of G.

4 Comparison of quantum search with classical algorithms

Let us discuss classical algorithms for solving FSAT under a black box model of computation (Nielsen and Chuang 2010), where the only knowledge we have of the Boolean function ϕ is the ability to evaluate it given an assignment of the Boolean variables, i.e., we have an oracle that answers queries over ϕ of the form "given assignment x, is $\phi(x)$ equal to 1?"

Consider first the case that $M = 1$. A deterministic algorithm for finding the single configuration x of n bits such that $\phi(x) = 1$ under the black box model clearly requires $N = 2^n$ evaluations of ϕ in the worst case.

Let us consider a probabilistic algorithm, i.e., an algorithm that returns the solution of the problem with probability p, with $0 < p \leq 1$, and returns "no answer" with probability $1 - p$.

A classical probabilistic algorithm for solving the search problem with $M = 1$ is the following: take s samples of configurations of X with uniform probability. This can be performed by sampling each Boolean variable uniformly and combining the bit samples obtaining a configuration. Then, for each configurations, test whether it is a solution. If it is a solution, return it and stop. The probability of finding the single solution x is $\frac{s}{N}$ so we need at least pN queries to find x with a probability at least p.

We may think that using a sampling distribution different from uniform we may do better, but the following lemma proves that this is not true.

Lemma 5 *Let $N = 2^n$ and ϕ be a black box function. Assume that A_ϕ is a probabilistic algorithm that makes queries to ϕ and returns an element $x \in \mathbb{B}^n$. If, for any non-constant ϕ, the probability that $\phi(x) = 1$ is at least $p > 0$, then there is a function ϕ' such $A_{\phi'}$ makes at least pN queries.*[1]

Proof The proof of this lemma is included in the Supplementary material. □

When $M > 1$, the probabilistic algorithm that samples uniformly has probability $\frac{sM}{N}$ of sampling a solution by taking s samples so we would need at least $\frac{pN}{M}$ samples to obtain a solution with probability at least p. Again, using a non uniform sampling distribution does not provide improvements, as the next lemma shows.

Lemma 6 *Let $N = 2^n$ and ϕ be a black box function with M configurations x for which $\phi(x) = 1$. Assume that A_ϕ is a probabilistic algorithm that makes queries to ϕ and returns an element $x \in \mathbb{B}^n$. If, for any ϕ with M solutions, the probability that $\phi(x) = 1$ is at least $p > 0$, then there is a function ϕ' such $A_{\phi'}$ makes at least $p\frac{N}{M}$ queries.*

Proof The proof of this lemma is included in the Supplementary material. □

[1] This lemma differs from Lemma 5.1.1 in Hirvensalo (2013) because the bound on the number of queries here is pN rather than $pN - 1$, thus providing a tighter bound.

We can now present the main result.

Theorem 5 *Any classical algorithm for solving the FSAT problem with probability at least $\frac{1}{4}$ requires $\Omega(N)$ queries to the oracle, when considering M fixed.*

Proof By Lemma 6 a classical algorithm that solves FSAT with probability $\frac{1}{4}$ makes at least $\frac{1}{4}\frac{N}{M}$ queries. $\square$

So quantum search offers a quadratic improvement over classical search.

If we consider the CS problem, sampling non-uniformly is not an option. The result thus is the same.

Theorem 6 *Any classical algorithm for solving the CS problem with $Q = X$ and probability at least $\frac{1}{4}$ requires $\Omega(N)$ queries to the oracle, when considering M fixed.*

5 Quantum weighted constrained sampling

Suppose first that the literal weights sum to 1, i.e., that $w(X_i) + w(\neg X_i) = 1$ for all bits X_i.

Given a Boolean function $\phi : \mathbb{B}^n \to \mathbb{B}$, a weight function $w : L \to [0, 1]$ and set of variables Q, we want to sample values for the variables Q so that the probability of sampling configuration q is

$$P(q) = \frac{\sum_{y:\phi(qy)=1} W_{qy}}{\mathrm{WMC}(\phi, w)}$$

Suppose, without loss of generality, that the query bits Q come first and there are l of them, while there are and there are $n-l$ bits in Y. Assume also that we add an extra bit X_{n+1} so $X' = [X_1, \dots, X_{n+1}], x' = [x_1, \dots, x_{n+1}], \phi' = \phi \wedge X_{n+1}$. Let Y' be Y with the extra bit X_{n+1}, so overall Y' has $n-l+1$ bits.

We perform quantum WCS by modifying the algorithm for quantum search, obtaining QWCS. The circuit for performing QWCS differs from the one in Fig. 2 because the Hadamard operations applied to the lower register are replaced by rotations $R_y(\theta_i)$ where i is the qubit index except for the extra qubit for which the Hadamard operator is kept. Overall the gate $H^{\otimes n+1}$ is replaced by gate *Rot* shown in Fig. 5. θ_i is computed as

$$\theta_i = 2 \arccos \sqrt{1 - w_i}$$

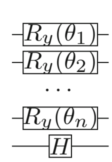

Fig. 5 Circuit for gate *Rot*

where $w_i = w(X_i)$.

So

$$\cos\theta_i/2 = \cos\arccos\sqrt{1-w_i} = \sqrt{1-w_i}$$

and

$$\sin\theta_i/2 = \sqrt{1-(\cos\theta_i/2)^2} = \sqrt{w_i}$$

The effect of the rotation on the ith bit is

$$\begin{aligned} R_y(\theta_i)|0\rangle &= \begin{bmatrix} \cos\frac{\theta_i}{2} & -\sin\frac{\theta_i}{2} \\ \sin\frac{\theta_i}{2} & \cos\frac{\theta_i}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta_i}{2} \\ \sin\frac{\theta_i}{2} \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{1-w_i} \\ \sqrt{w_i} \end{bmatrix} = \sqrt{1-w_i}|0\rangle + \sqrt{w_i}|1\rangle \end{aligned}$$

Therefore the rotations prepare the state

$$\begin{aligned} |\varphi\rangle &= \bigotimes_{i=1}^{n}(\sqrt{1-w_i}|0\rangle + \sqrt{w_i}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \sum_{x_1\dots x_{n+1}=0}^{2^{n+1}-1} \sqrt{w'_1 \dots w'_{n+1}} |x_1 \dots x_{n+1}\rangle \end{aligned}$$

where w'_i is

$$w'_i = \begin{cases} w_i & \text{if } x_i = 1 \\ 1 - w_i & \text{if } x_i = 0 \end{cases}$$

for $i = 1, \dots, n$ and $w'_{n+1} = 1/2$. $|\varphi\rangle$ can be rewritten as

$$|\varphi\rangle = \sum_{x':\phi'(x')=0} \sqrt{\frac{w'_1 \dots w'_n}{2}} |x'\rangle + \sum_{x':\phi'(x')=1} \sqrt{\frac{w'_1 \dots w'_n}{2}} |x'\rangle$$

where the first summation is over all configurations x' of the $n+1$ bits such that $\phi'(x') = 0$ and the latter over all confgurations x' of the $n+1$ bits such that $\phi'(x') = 1$. Then,

$$|\varphi\rangle = \sum_{x,x_{n+1}:x_{n+1}=0 \vee \phi(x)=0} \sqrt{\frac{w'_1 \dots w'_n}{2}} |x'\rangle + \sum_{x':\phi'(x')=1} \sqrt{\frac{w'_1 \dots w'_n}{2}} |x'\rangle$$

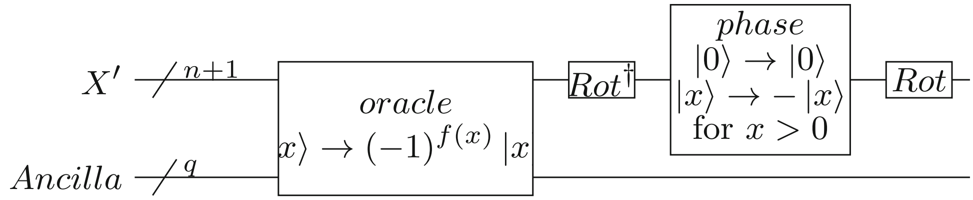

Fig. 6 Weighted Grover operator WG

because $\phi'(x')$ is 0 if x_{n+1} is 0 or $\phi(x) = 0$. Moreover

$$\begin{aligned}
|\varphi\rangle =& \sum_{x,x_{n+1}:x_{n+1}=0\vee(x_{n+1}=1\wedge\phi(x)=0)} \sqrt{\frac{w'_1\dots w'_n}{2}}|x'\rangle \\
&+ \sum_{x':\phi'(x')=1} \sqrt{\frac{w'_1\dots w'_n}{2}}|x'\rangle \\
=& \sum_{x,x_{n+1}:x_{n+1}=0} \sqrt{\frac{w'_1\dots w'_n}{2}}|x'\rangle + \sum_{x,x_{n+1}:x_{n+1}=1\wedge\phi(x)=0} \sqrt{\frac{w'_1\dots w'_n}{2}}|x'\rangle + \\
&+ \sum_{x':\phi'(x')=1} \sqrt{\frac{w'_1\dots w'_n}{2}}|x'\rangle \\
=& \sum_{x} \sqrt{\frac{w'_1\dots w'_n}{2}}|x0\rangle + \sum_{x:\phi(x)=0} \sqrt{\frac{w'_1\dots w'_n}{2}}|x1\rangle \\
&+ \sum_{x:\phi(x)=1} \sqrt{\frac{w'_1\dots w'_n}{2}}|x1\rangle
\end{aligned}$$

where the last equation is obtained by setting the x_{n+1} bit in the quantum state.

Define W_x as $w'_1 \cdot \ldots \cdot w'_n$ and normalized states

$$\begin{aligned}
|\gamma\rangle &= \frac{1}{\sqrt{\frac{1+\sum_{x:\phi(x)=0} W_x}{2}}}\left(\sum_{x}\sqrt{\frac{W_x}{2}}|x0\rangle + \sum_{x:\phi(x)=0}\sqrt{\frac{W_x}{2}}|x1\rangle\right) \\
|\delta\rangle &= \frac{1}{\sqrt{\frac{\sum_{x:\phi(x)=1} W_x}{2}}}\sum_{x:\phi(x)=1}\sqrt{\frac{W_x}{2}}|x1\rangle,
\end{aligned}$$

then $|\varphi\rangle$ can be expressed as

$$|\varphi\rangle = \left(\sqrt{\frac{1+\sum_{x:\phi(x)=0} W_x}{2}}\right)|\gamma\rangle + \left(\sqrt{\frac{\sum_{x:\phi(x)=1} W_x}{2}}\right)|\delta\rangle$$

so the initial state of the quantum computer is in the space spanned by $|\gamma\rangle$ and $|\delta\rangle$

Let $\cos\theta = \sqrt{\frac{1+\sum_{x:\phi(x)=0} W_x}{2}}$ and $\sin\theta = \sqrt{\frac{\sum_{x:\phi(x)=1} W_x}{2}}$ so that

$$|\varphi\rangle = \cos\theta/2|\gamma\rangle + \sin\theta/2|\delta\rangle.$$

Note that, correctly, $\cos^2\theta + \sin^2\theta = 1$ since $\sum_{x:\phi(x)=0} W_x + \sum_{x:\phi(x)=1} W_x = 1$. Gate Rot replaces $H^{\otimes n+1}$ also in the Grover operator G that becomes the weighted Grover operator WG shown in Fig. 6:

$$\begin{aligned}
\mathrm{WG} &= Rot(2|0\rangle\langle 0| - I)Rot^{\dagger}O \\
&= (2Rot|0\rangle\langle 0|Rot^{\dagger} - Rot\, I\, Rot^{\dagger})O \\
&= (2|\varphi\rangle\langle\varphi| - I)O
\end{aligned}$$

The overall circuit for performing QWCS is shown in Fig. 7, where the Grover operator of quantum search is replaced by the Weighted Grover operator WG shown in Fig. 6 and the initial $H^{\otimes n+1}$ gate is replaced by gate Rot of Fig. 5.

The initial state $|\varphi\rangle$ of the quantum computer is in the space spanned by $|\gamma\rangle$ and $|\delta\rangle$ which are orthonormal since they do not share computational basis states. As for the Grover algorithm, the application of WG rotates $|\varphi\rangle$ in the space spanned by $|\gamma\rangle$ and $|\delta\rangle$ by angle θ.

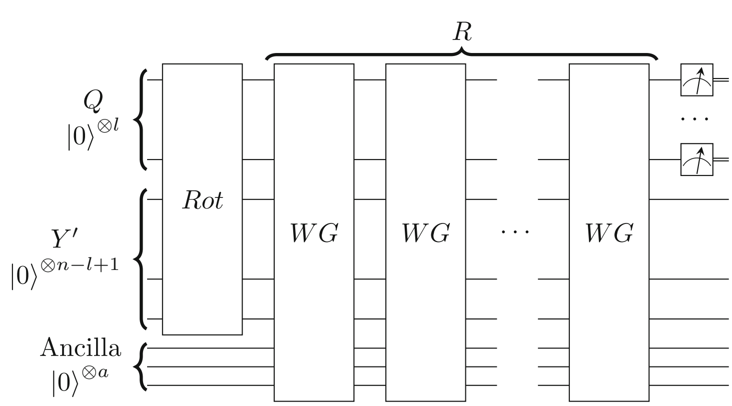

Fig. 7 The complete QWCS circuit

Repeated applications of WG take the state to

$$\mathrm{WG}^k|\varphi\rangle = \cos((2k+1)\theta)|\gamma\rangle + \sin((2k+1)\theta)|\delta\rangle.$$

These rotations bring the state of the system closer to $|\delta\rangle$.

Now, we measure only the query qubits. Let us see what is the result of this measurement. The density operator for the system is

$$\rho = (\cos((2k+1)\theta)|\gamma\rangle + \sin((2k+1)\theta)|\delta\rangle)\\ (\cos((2k+1)\theta)\langle\gamma| + \sin((2k+1)\theta)\langle\delta|)$$

For the distributivity of matrix multiplication, we can rewrite ρ as

$$\rho = \rho_1 + \rho_2 + \rho_3 + \rho_4$$

with

$$\begin{aligned}\rho_1 &= \sin^2((2k+1)\theta)|\delta\rangle\langle\delta|\\ \rho_2 &= \cos((2k+1)\theta)\sin((2k+1)\theta)|\gamma\rangle\langle\delta|\\ \rho_3 &= \sin((2k+1)\theta)\cos((2k+1)\theta)|\delta\rangle\langle\gamma|\\ \rho_4 &= \cos^2((2k+1)\theta)|\gamma\rangle\langle\gamma|\end{aligned}$$

We now trace out bits Y' obtaining the reduced density operator for q

$$\begin{aligned}\rho^Q &= \operatorname*{tr}_{Y'}(\rho_1+\rho_2+\rho_3+\rho_4)\\ &= \operatorname*{tr}_{Y'}(\rho_1) + \operatorname*{tr}_{Y'}(\rho_2) + \operatorname*{tr}_{Y'}(\rho_3) + \operatorname*{tr}_{Y'}(\rho_4)\\ &= \rho_1^Q + \rho_2^Q + \rho_3^Q + \rho_4^Q\end{aligned}$$

for the linearity of the partial trace operator.

Let us consider first ρ_1 We rewrite $|\delta\rangle$ as

$$\begin{aligned}|\delta\rangle &= \frac{1}{\sqrt{\frac{\sum_{x:\phi(x)=1} W_x}{2}}} \sum_{x:\phi(x)=1} \sqrt{\frac{W_x}{2}}|x1\rangle\\ &= \frac{1}{Z_1}\sum_{x:\phi(x)=1}\sqrt{\frac{W_x}{2}}|x1\rangle\end{aligned}$$

with $Z_1 = \sqrt{\frac{\sum_{x:\phi(x)=1} W_x}{2}}$.

The density operator for ρ_1 is

$$\rho_1 = \sin^2((2k+1)\theta)\left(\frac{1}{Z_1}\sum_{x:\phi(x)=1}\sqrt{\frac{W_x}{2}}|x1\rangle\right)\left(\frac{1}{Z_1}\sum_{x:\phi(x)=1}\sqrt{\frac{W_x}{2}}\langle x1|\right)$$

Writing a bit configuration x as qy, ρ_1 becomes

$$\begin{aligned}\rho_1 &= \sin^2((2k+1)\theta)\left(\frac{1}{Z_1}\sum_{qy:\phi(qy)=1}\sqrt{\frac{W_{qy}}{2}}|qy1\rangle\right)\\ &\quad\left(\frac{1}{Z_1}\sum_{qy:\phi(qy)=1}\sqrt{\frac{W_{qy}}{2}}\langle qy1|\right)\\ &= \sin^2((2k+1)\theta)\left(\frac{1}{Z_1}\sum_{qy:\phi(qy)=1}\sqrt{\frac{W_{qy}}{2}}|qy1\rangle\right)\\ &\quad\left(\frac{1}{Z_1}\sum_{rz:\phi(rz)=1}\sqrt{\frac{W_{rz}}{2}}\langle rz1|\right) \qquad (3)\\ &= \sin^2((2k+1)\theta)\frac{1}{Z_1^2}\left(\sum_q\sum_{y:\phi(qy)=1}\sqrt{\frac{W_{qy}}{2}}|qy1\rangle\right)\\ &\quad\left(\sum_r\sum_{z:\phi(rz)=1}\sqrt{\frac{W_{rz}}{2}}\langle rz1|\right) \qquad (4)\\ &= \sin^2((2k+1)\theta)\frac{1}{2Z_1^2}\sum_{q,r}\sum_{y,z:\phi(qy)=1,\phi(rz)=1}\\ &\quad\sqrt{W_{qy}W_{rz}}|qy1\rangle\langle rz1| \qquad (5)\end{aligned}$$

where we get Eq. (3) by renaming q and y in the second factor as r and z, Eq. (4) by splitting the sums and Eq. (5) by multiplying the two factors and rearranging the terms. We now trace out bits Y' obtaining the reduced density operator for Q:

$$\begin{aligned}\rho_1^Q &= \operatorname*{tr}_{Y'}(\rho_1)\\ &= \frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{q,r}\sum_{y,z:\phi(qy)=1,\phi(rz)=1}\sqrt{W_{qy}W_{rz}}\operatorname*{tr}_{Y'}(|qy1\rangle\langle rz1|) \qquad (6)\\ &= \frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{q,r}\sum_{y,z:\phi(qy)=1,\phi(rz)=1}\sqrt{W_{qy}W_{rz}}|q\rangle\langle r|\operatorname{tr}(|y1\rangle\langle z1|) \qquad (7)\\ &= \frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{q,r}\sum_{y,z:\phi(qy)=1,\phi(rz)=1}\sqrt{W_{qy}W_{rz}}|q\rangle\langle r|\langle z1|y1\rangle \qquad (8)\\ &= \frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{q,r}\sum_{y:\phi(qy)=1,\phi(ry)=1}\sqrt{W_{qy}W_{ry}}|q\rangle\langle r| \qquad (9)\end{aligned}$$

where we get Eq. (6) for the linearity of the partial trace operator, Eq. (7) for the following property of the partial trace operator

$$\operatorname*{tr}_B(|a_1\rangle\langle a_2|\otimes|b_1\rangle\langle b_2|) = |a_1\rangle\langle a_2|\operatorname{tr}(|b_1\rangle\langle b_2|), \qquad (10)$$

and Eq. (9) because

$$\langle a|b\rangle = \begin{cases}0 \text{ if } a \neq b\\ 1 \text{ if } a = b\end{cases} \qquad (11)$$

when $|a\rangle$ and $|b\rangle$ are computational basis states.

Let us consider ρ_2. We rewrite $|\gamma\rangle$ as

$$|\gamma\rangle = \frac{1}{\sqrt{\frac{1+\sum_{x:\phi(x)=0} W_x}{2}}}\left(\sum_x \sqrt{\frac{W_x}{2}}|x0\rangle + \sum_{x:\phi(x)=0}\sqrt{\frac{W_x}{2}}|x1\rangle\right)$$
$$= \frac{1}{Z_2}\left(\sum_x \sqrt{\frac{W_x}{2}}|x0\rangle + \sum_{x:\phi(x)=0}\sqrt{\frac{W_x}{2}}|x1\rangle\right)$$

with $Z_2 = \sqrt{\frac{1+\sum_{x:\phi(x)=0} W_x}{2}}$.

The density operator ρ_2 is

$$\rho_2 = \cos((2k+1)\theta)\sin((2k+1)\theta)\frac{1}{Z_2}$$
$$\left(\sum_x \sqrt{\frac{W_x}{2}}|x0\rangle + \sum_{x:\phi(x)=0}\sqrt{\frac{W_x}{2}}|x1\rangle\right)\left(\frac{1}{Z_1}\sum_{x:\phi(x)=1}\sqrt{\frac{W_x}{2}}\langle x1|\right)$$

Rewriting x as qy and rz, ρ_2 becomes

$$\rho_2 = \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{Z_1 Z_2}$$
$$\left(\sum_{qy}\sqrt{\frac{W_{qy}}{2}}|qy0\rangle + \sum_{qy:\phi(qy)=0}\sqrt{\frac{W_{qy}}{2}}|qy1\rangle\right)\left(\sum_{rz:\phi(rz)=1}\sqrt{\frac{W_{rz}}{2}}\langle rz1|\right)$$
$$= \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1 Z_2}\sum_{q,r}$$
$$\left(\sum_{y,z:\phi(rz)=1}\sqrt{W_{qy}W_{rz}}|qy0\rangle\langle rz1|\right.$$
$$\left.+\sum_{y:\phi(qy)=0,z:\phi(rz)=1}\sqrt{W_{qy}W_{rz}}|qy1\rangle\langle rz1|\right)$$

We now trace out bits Y' obtaining the reduced density operator for Q:

$$\rho_2^Q = \operatorname*{tr}_{Y'}(\rho_2)$$
$$= \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1 Z_2}\sum_{q,r}$$
$$\left(\sum_{y,z:\phi(rz)=1}\sqrt{W_{qy}W_{rz}}\operatorname*{tr}_{Y'}(|qy0\rangle\langle rz1|)\right.$$
$$\left.+\sum_{y:\phi(qy)=0,z:\phi(rz)=1}\sqrt{W_{qy}W_{rz}}\operatorname*{tr}_{Y'}(|qy1\rangle\langle rz1|)\right) \tag{12}$$
$$= \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1 Z_2}\sum_{q,r}$$
$$\left(\sum_{y,z:\phi(rz)=1}\sqrt{W_{qy}W_{rz}}|q\rangle\langle r|\langle z1|y0\rangle)\right.$$
$$\left.+\sum_{y:\phi(qy)=0,z:\phi(rz)=1}\sqrt{W_{qy}W_{rz}}|q\rangle\langle r|\langle z1|y1\rangle\right) \tag{13}$$
$$= \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1 Z_2}$$
$$\sum_{q,r}\left(\sum_{y:\phi(qy)=0,z:\phi(rz)=1}\sqrt{W_{qy}W_{rz}}|q\rangle\langle r|\langle z1|y1\rangle\right) \tag{14}$$
$$= \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1 Z_2}$$
$$\sum_{q,r}\left(\sum_{y:\phi(qy)=0,\phi(ry)=1}\sqrt{W_{qy}W_{ry}}|q\rangle\langle r|\right) \tag{15}$$

where we get Eq. (12) for the linearity of the partial trace operator, Eq. (13) for Eq. (10) and Eq. (14) and Eq. (15) because of Eq. (11).

For ρ_3^Q, we similarly get

$$\rho_3^Q = \operatorname*{tr}_{Y'}(\rho_3)$$
$$= \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1 Z_2}\sum_{q,r}\left(\sum_{y:\phi(qy)=0,\phi(ry)=1}\sqrt{W_{qy}W_{ry}}|r\rangle\langle q|\right)$$

The density operator ρ_4 is

$$\rho_4 = \frac{\cos^2((2k+1)\theta)}{Z_2^2}\left(\sum_x\sqrt{\frac{W_x}{2}}|x0\rangle + \sum_{x:\phi(x)=0}\sqrt{\frac{W_x}{2}}|x1\rangle\right)$$
$$\left(\sum_x\sqrt{\frac{W_x}{2}}\langle x0| + \sum_{x:\phi(x)=0}\sqrt{\frac{W_x}{2}}\langle x1|\right)$$

Rewriting x as qy and rz, ρ_4 becomes

$$\rho_4 = \frac{\cos^2((2k+1)\theta)}{2Z_2^2}\left(\sum_{qy}\sqrt{\frac{W_{qy}}{2}}|qy0\rangle + \sum_{qy:\phi(qy)=0}\sqrt{\frac{W_{qy}}{2}}|qy1\rangle\right)$$
$$\left(\sum_{rz}\sqrt{\frac{W_{rz}}{2}}\langle rz0| + \sum_{rz:\phi(rz)=0}\sqrt{\frac{W_{rz}}{2}}\langle rz1|\right)$$
$$= \frac{\cos^2((2k+1)\theta)}{2Z_2^2}\sum_{q,r}\left(\sum_{y,z}\sqrt{W_{qy}W_{rz}}|qy0\rangle\langle rz0|\right.$$
$$+\sum_{y,z:\phi(rz)=0}\sqrt{W_{qy}W_{rz}}|qy0\rangle\langle rz1|$$
$$+\sum_{y:\phi(qy)=0,z}\sqrt{W_{qy}W_{rz}}|qy1\rangle\langle rz0|$$
$$\left.+\sum_{y:\phi(qy)=0,z:\phi(rz)=0}\sqrt{W_{qy}W_{rz}}|qy1\rangle\langle rz1|\right)$$

Tracing out bits Y' obtaining the reduced density operator for Q, we get:

$$\begin{aligned}
\rho_4^Q &= \operatorname{tr}_{Y'}(\rho_4) \\
&= \frac{\cos^2((2k+1)\theta)}{2Z_2^2}\sum_{q,r}\left(\sum_{y,z}\sqrt{W_{qy}W_{rz}}\operatorname{tr}_{Y'}(|qy0\rangle\langle rz0|)\right. \\
&\quad+\sum_{y,z:\phi(rz)=0}\sqrt{W_{qy}W_{rz}}\operatorname{tr}_{Y'}(|qy0\rangle\langle rz1|)) \\
&\quad+\sum_{y:\phi(qy)=0,z}\sqrt{W_{qy}W_{rz}}\operatorname{tr}_{Y'}(|qy1\rangle\langle rz0|) \\
&\quad\left.+\sum_{y:\phi(qy)=0,z:\phi(rz)=0}\sqrt{W_{qy}W_{rz}}\operatorname{tr}_{Y'}(|qy1\rangle\langle rz1|)\right) \\
&= \frac{\cos^2((2k+1)\theta)}{2Z_2^2}\sum_{q,r}\left(\sum_{y,z}\sqrt{W_{qy}W_{rz}}|q\rangle\langle r|\langle z0|y0\rangle\right. \\
&\quad+\sum_{y,z:\phi(rz)=0}\sqrt{W_{qy}W_{rz}}|q\rangle\langle r|\langle z1|y0\rangle \\
&\quad+\sum_{y:\phi(qy)=0,z}\sqrt{W_{qy}W_{rz}}|q\rangle\langle r|\langle z0|y1\rangle \\
&\quad\left.+\sum_{y:\phi(qy)=0,z:\phi(rz)=0}\sqrt{W_{qy}W_{rz}}|q\rangle\langle r|\langle z1|y1\rangle\right) \\
&= \frac{\cos^2((2k+1)\theta)}{2Z_2^2}\sum_{q,r}\left(\sum_{y}\sqrt{W_{qy}W_{ry}}|q\rangle\langle r|\right. \\
&\qquad\left.+\sum_{y:\phi(qy)=0,\phi(ry)=0}\sqrt{W_{qy}W_{ry}}|q\rangle\langle r|\right)
\end{aligned}$$

Let's apply the measurement $\{M_m = |q_m\rangle\langle q_m|\}$ to system Q where q_m is one of the computational basis state for Q. This is a measurement in the computational basis state of system Q so $M_m^\dagger M_m = M_m$ and

$$\begin{aligned}
P(m) &= \operatorname{tr}(M_m\rho^Q) \\
&= \operatorname{tr}\left(|q_m\rangle\langle q_m|(\rho_1^Q+\rho_2^Q+\rho_3^Q+\rho_4^Q)\right) \\
&= \operatorname{tr}(|q_m\rangle\langle q_m|\rho_1^Q)+\operatorname{tr}(|q_m\rangle\langle q_m|\rho_2^Q) \\
&\quad+\operatorname{tr}(|q_m\rangle\langle q_m|\rho_3^Q)+\operatorname{tr}(|q_m\rangle\langle q_m|\rho_4^Q)
\end{aligned}$$

Let us define $P_i(m) = \operatorname{tr}\left(|q_m\rangle\langle q_m|\rho_i^Q\right)$ for $i = 1,\ldots,4$. Then,

$$\begin{aligned}
P_1(m) &= \operatorname{tr}\left(|q_m\rangle\langle q_m|\frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{q,r}\sum_{y:\phi(qy)=1:\phi(ry)=1}\sqrt{W_{qy}W_{ry}}|q\rangle\langle r|\right) \\
&= \operatorname{tr}\left(\frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{q,r}\sum_{y:\phi(qy)=1:\phi(ry)=1}\sqrt{W_{qy}W_{ry}}|q_m\rangle\langle q_m|q\rangle\langle r|\right)
\end{aligned}$$

Since $\langle q_m|q\rangle = 0$ if $q_m \neq q$ and $\langle q_m|q\rangle = 1$ if $q_m = q$ then

$$\begin{aligned}
P_1(m) &= \operatorname{tr}\left(\frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{r}\sum_{y:\phi(ry)=1,\phi(q_m,y)=1}\sqrt{W_{q_my}W_{ry}}|q_m\rangle\langle r|\right) \\
&= \frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{r}\sum_{y:\phi(ry)=1,\phi(q_m,y)=1}\sqrt{W_{q_my}W_{ry}}\operatorname{tr}(|q_m\rangle\langle r|) \\
&= \frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{r}\sum_{y:\phi(ry)=1,\phi(q_m,y)=1}\sqrt{W_{q_my}W_{ry}}\langle r|q_m\rangle \\
&= \frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{y:\phi(q_my)=1}\sqrt{W_{q_my}W_{q_my}} \\
&= \frac{\sin^2((2k+1)\theta)}{2Z_1^2}\sum_{y:\phi(q_my)=1}W_{q_my} \\
&= \frac{\sin^2((2k+1)\theta)}{\frac{2}{2}\sum_{x:\phi(x)=1}W_x}\sum_{y:\phi(q_my)=1}W_{q_my} \\
&= \frac{\sin^2((2k+1)\theta)}{\sum_{x:\phi(x)=1}W_x}\sum_{y:\phi(q_my)=1}W_{q_my} \\
&= \sin^2((2k+1)\theta)\frac{\sum_{y:\phi(q_my)=1}W_{q_my}}{\text{WMC}}
\end{aligned}$$

For $P_2(m)$, we have:

$$\begin{aligned}
P_2(m) &= \operatorname{tr}\left(|q_m\rangle\langle q_m|\frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1Z_2}\sum_{q,r}\sum_{y:\phi(qy)=0,\phi(ry)=1}\sqrt{W_{qy}W_{ry}}|q\rangle\langle r|\right) \\
&= \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1Z_2}\sum_{q,r}\sum_{y:\phi(qy)=0,\phi(ry)=1}\sqrt{W_{qy}W_{ry}}\operatorname{tr}(|q_m\rangle\langle q|q_m\rangle\langle r|) \\
&= \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1Z_2}\sum_{r}\sum_{y:\phi(q_my)=0,\phi(ry)=1}\sqrt{W_{q_my}W_{ry}}\operatorname{tr}(|q_m\rangle\langle r|) \\
&= \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1Z_2}\sum_{r}\sum_{y:\phi(q_my)=0,\phi(ry)=1}\sqrt{W_{q_my}W_{ry}}\langle r|q_m\rangle \\
&= \frac{\cos((2k+1)\theta)\sin((2k+1)\theta)}{2Z_1Z_2}\sum_{y:\phi(q_my)=0,\phi(q_my)=1}\sqrt{W_{q_my}W_{q_my}} \\
&= 0
\end{aligned}$$

because there is no y such that $\phi(q_m y) = 0$ and $\phi(q_m y) = 1$. Similarly $P_3(m) = 0$. For $P_4(m)$, we have:

$$
\begin{aligned}
P_4(m) &= \mathrm{tr}\left(|q_m\rangle\langle q_m| \frac{\cos^2((2k+1)\theta)}{Z_2^2} \sum_{q,r}\left(\sum_y \sqrt{W_{qy}W_{ry}}|q\rangle\langle r|\right.\right. \\
&\quad \left.\left. + \sum_{y:\phi(qy)=0,\phi(ry)=0} \sqrt{W_{qy}W_{ry}}|q\rangle\langle r|\right)\right) \\
&= \frac{\cos^2((2k+1)\theta)}{Z_2^2} \sum_{q,r}\left(\sum_y \sqrt{W_{qy}W_{ry}}\ \mathrm{tr}\,(|q_m\rangle\langle q_m|q\rangle\langle r|)\right. \\
&\quad \left. + \sum_{y:\phi(qy)=0,\phi(ry)=0} \sqrt{W_{qy}W_{ry}}\ \mathrm{tr}\,(|q_m\rangle\langle q_m|q\rangle\langle r|)\right) \\
&= \frac{\cos^2((2k+1)\theta)}{Z_2^2} \sum_{r}\left(\sum_y \sqrt{W_{q_m y}W_{ry}}\langle r|q_m\rangle\right. \\
&\quad \left. + \sum_{y:\phi(q_m y)=0,\phi(ry)=0} \sqrt{W_{q_m y}W_{ry}}|q_m\rangle\langle r|q_m\rangle\right) \\
&= \frac{\cos^2((2k+1)\theta)}{1+\sum_{x:\phi(x)=0} W_x}\left(\sum_y W_{q_m y} + \sum_{y:\phi(q_m y)=0} W_{q_m y}\right) \\
&= \cos^2((2k+1)\theta)\frac{W_{q_m} + \sum_{y:\phi(q_m y)=0} W_{q_m y}}{2-\mathrm{WMC}}
\end{aligned}
$$

If $\sin((2k+1)\theta) = 1$, this algorithm returns one of the configurations of query bits q_m that are superimposed in $|\delta\rangle$ with a probability that is proportional to $\sum_{y:\phi(q_m y)=1} W_{q_m y}$.

This leads to Algorithm 4, where lines 3–5 are implemented by the quantum circuit of Fig. 7.

Algorithm 4 WCS when WMC is known

Require: A blackbox function $\phi : \mathbb{B}^n \to \mathbb{B}$, normalized weight function w, query qbits Q and WMC

Ensure: $q \in \mathbb{B}^l$ sampled from distribution (2)

1: $\theta \leftarrow \arcsin\sqrt{\frac{\mathrm{WMC}}{2}}$ with $\theta \in [0, \pi/4]$

2: $R \leftarrow \lfloor \frac{\pi}{4\theta} \rfloor$

3: Prepare the initial superposition $Rot|x'\rangle$

4: Apply operator WG R times

5: Measure Q to get $q \in \mathbb{B}^l$

6: **return** q

The properties of this algorithm are described by the theorem below.

Theorem 7 *Let $\phi : \mathbb{B}^n \to \mathbb{B}$ Let $\theta \in [0, \pi/4]$ be chosen such that $\sin^2\theta = \frac{WMC}{2}$. After $\lfloor \frac{\pi}{4\theta} \rfloor$ iterations of WG on an initial superposition*

$$Rot|x'\rangle$$

the probability of sampling exactly from the distribution in Eq. (2) is at least $\sqrt{\frac{1}{2}} \approx 0.707$.

Proof We can repeat the reasoning of Theorem 1: the probability P of sampling from the distribution in Eq. (2) is given by

$$P = \sin^2((2k+1)\theta)$$

Maximizing P implies that

$$(2k+1)\theta = \frac{\pi}{2}$$

so

$$k = \frac{\pi}{4\theta} - \frac{1}{2}$$

.

From $\theta \approx \sqrt{\frac{\mathrm{WMC}}{2}}$, we obtain

$$R = \left\lfloor \frac{\pi}{4}\sqrt{\frac{2}{\mathrm{WMC}}} \right\rfloor$$

for the required number of applications of WG. Thus

$$(2R+1)\theta = \frac{\pi}{2} + 2\delta\theta$$

with $|\delta| \leq \frac{1}{2}$. Since $\mathrm{WMC} \leq 1$, $\sin\theta \leq \sqrt{\frac{1}{2}}$, $\theta \leq \frac{\pi}{4}$, $|2\delta\theta| \leq \frac{\pi}{4}$. So

$$P = \sin^2((2R+1)\theta) \geq \sin^2\left(\frac{\pi}{2} - \frac{\pi}{4}\right) = \sqrt{\frac{1}{2}} \approx 0.707$$

□

We can now present the main result.

Theorem 8 *Algorithm 4 samples from the distribution in Eq. (2) with probability at least $\sqrt{\frac{1}{2}}$ and $O(\frac{1}{\sqrt{WMC}})$ queries to ϕ.*

Proof By reasoning as in the proof of Theorem 2, in each application of WG, we query ϕ and the number of applications R is $R \leq \frac{\pi}{4}\sqrt{\frac{2}{\mathrm{WMC}}}$, so $R \in O(\frac{1}{\sqrt{\mathrm{WMC}}})$ □

When WMC is not known, Algorithm 5 can be used, where W_{min} is the minimum weight of a configuration, i.e., $W_{min} = \prod_{i=1}^{n+1} w'_{i,min}$ and $w'_{i,min} = \min(w'(X_i), w'(\neg X_i))$ for $i = 1, \ldots, n+1$..

Lemma 7 and Theorem 9 are analogous to Lemma 4 and Theorem 4.

Lemma 7 *Let $\phi : \mathbb{B}^n \to \mathbb{B}$ and weight function w be such the weighed model count is WMC. Let ϕ' be $\phi' = \phi \wedge X_{n+1}$. Let $\theta \in [0, \pi/4]$ be defined by $\sin^2\theta = \frac{2}{WMC}$. Let m be any positive integer and $R \in [0, m-1]$ chosen with uniform distribution. If WG is applied to initial superposition*

$$\left(\sum_x \sqrt{\frac{W_x}{2}}|x0\rangle + \sum_{x:\phi(x)=0} \sqrt{\frac{W_x}{2}}|x1\rangle\right) + \sum_{x:\phi(x)=1} \sqrt{\frac{W_x}{2}}|x1\rangle$$

Algorithm 5 WCS when WMC is not known

Require: A blackbox function $\phi : \mathbb{B}^n \to \mathbb{B}$, normalized weight function w, query qubits Q
Ensure: $q \in \mathbb{B}^l$ sampled from distribution (2)
1: $m = \left\lfloor \frac{1}{\sqrt{W_{min}}} \right\rfloor + 1$
2: Choose an integer R uniformly in $[0, m-1]$
3: Prepare the initial superposition $\frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle$
4: Apply operator G R times
5: Measure to get $x \in \mathbb{B}^n$
6: **return** x

R times, then the probability of seeing a solution is

$$P_m = \frac{1}{2} - \frac{\sin(4m\theta)}{4m\sin(2\theta)}$$

Proof The proof of Lemma 3 still applies. □

Theorem 9 *Algorithm 5 samples from Eq. (2) with probability at least $\frac{1}{4}$ and $O(\frac{1}{\sqrt{W_{min}}})$ queries to ϕ'.*

Proof If $m \geq \frac{1}{\sin(2\theta)}$, then

$$\frac{\sin(4m\theta)}{4m\sin(2\theta)} \leq \frac{1}{4}.$$

By Lemma 7, then $P_m \geq \frac{1}{4}$. Assuming ϕ is satisfiable, then $0 < W_{min} \leq \text{WMC} \leq 1$. Since $\sin\theta = \sqrt{\frac{\text{WMC}}{2}}$, we have:

$$\frac{1}{\sin(2\theta)} = \frac{1}{2\sin\theta\cos\theta} = \frac{1}{2\sqrt{\frac{\text{WMC}(2-\text{WMC})}{4}}}$$
$$= \frac{1}{\sqrt{\text{WMC}(2-\text{WMC})}} \leq \frac{1}{\sqrt{\text{WMC}}} \leq \frac{1}{\sqrt{W_{min}}}$$

So, if $m \geq \frac{1}{\sqrt{W_{min}}}$, then $m \geq \frac{1}{\sin(2\theta)}$. Choosing $m = \left\lfloor \frac{1}{\sqrt{W_{min}}} \right\rfloor + 1$, we have that $P_m \geq \frac{1}{4}$ and the number of applications of G is $O(\frac{1}{\sqrt{W_{min}}})$. □

If the literal weights do not sum to 1, i.e., $w(X_i)+w(\neg X_i) \neq 1$, we normalize them by considering the new weights $\hat{w}(X_i) = \frac{w(X_i)}{w(X_i)+w(\neg X_i)}$ and $\hat{w}(\neg X_i) = \frac{w(\neg X_i)}{w(X_i)+w(\neg X_i)}$. Let V_i be $w(X_i) + w(\neg X_i)$ for $i = 1, \ldots, n$. Then, we perform the algorithms with $\hat{w}$ replacing w. The overall normalization factor $\prod_{i=1}^n V_i$ gets canceled out in $P(m)$.

Let the normalized weighted model count $\widehat{\text{WMC}}$ be defined as

$$\widehat{\text{WMC}} = \sum_{x:\phi(x)=1} \widehat{W}_x$$

where $\hat{W}_x$ is $\prod_{i=1}^n \hat{w}'_i$ and

$$\hat{w}'_i = \begin{cases} \hat{w}(X_i) & \text{if } x_i = 1 \\ 1 - \hat{w}(X_i) & \text{if } x_i = 0 \end{cases}$$

Then,

$$\begin{aligned}
\widehat{\text{WMC}} &= \sum_{x:\phi(x)=1} \widehat{W}_x = \sum_{x_1\ldots x_n:\phi(x_1\ldots x_n)=1} \hat{w}'_1 \ldots \hat{w}'_n \\
&= \sum_{x_1\ldots x_n:\phi(x_1\ldots x_n)=1} \frac{w_1}{V_1} \cdots \frac{w_n}{V_n} \\
&= \sum_{x_1\ldots x_n:\phi(x_1\ldots x_n)=1} \frac{1}{\prod_{i=1}^n V_i} w_1 \ldots w_n \\
&= \frac{1}{\prod_{i=1}^n V_i} \sum_{x_1\ldots x_n:\phi(x_1\ldots x_n)=1} w_1 \ldots w_n \\
&= \frac{1}{\prod_{i=1}^n V_i} \sum_{x:\phi(x)=1} W_x \\
&= \frac{1}{\prod_{i=1}^n V_i} \text{WMC}
\end{aligned} \tag{16}$$

Theorems 7 and 8 still hold for Algorithm 4 provided that WMC is replaced by $\widehat{\text{WMC}}$, while Lemma 7 and Theorem 9 still hold for Algorithm 5 provided that WMC is replaced by $\widehat{\text{WMC}}$, and W_{min} is replaced by $\widehat{W}_{min} = \prod_{i=1}^n \min(\hat{w}(X_i), \hat{w}(\neg X_i))$.

In Sect. 9, we will see the QWMC algorithm that computes WMC in $\Theta(\sqrt{N})$ queries to ϕ. Since $\min(\hat{w}(X_i), \hat{w}(\neg X_i)) \leq \frac{1}{2}$, then $\widehat{W}_{min} \leq \frac{1}{2^n} = \frac{1}{N}$, so $\frac{1}{\sqrt{\widehat{W}_{min}}} \geq \sqrt{N}$. Moreover, $\frac{1}{\sqrt{\widehat{W}_{min}}} \geq \frac{1}{\sqrt{\widehat{\text{WMC}}}}$, so it is more convenient to apply QWMC (Algorithm 9) and then Algorithm 4, obtaining Algorithm 6 that we call QWCS.

Algorithm 6 Algorithm QWCS

Require: A blackbox function $\phi : \mathbb{B}^n \to \mathbb{B}$, normalized weight function w, query qbits Q
Ensure: $q \in \mathbb{B}^l$ sampled from distribution (2)
1: Call Algorithm 9 to obtain WMC
2: Call Algorithm 4 to obtain q
3: **return** q

We are now ready to state our main result for WCS.

Theorem 10 *QWCS samples the distribution from Eq. (2) with probability at least $\sqrt{\frac{1}{2}}$ and $O(\sqrt{N} + \frac{1}{\sqrt{\widehat{\text{WMC}}}})$ queries to ϕ'.*

Proof Immediate from theorems 14 and 8. □

6 Comparison of QWCS with classical algorithms

For classical probabilistic algorithms under a black box model of computation, we have the following results.

Theorem 11 *Any classical probabilistic algorithm for solving WCS under the black box model of computation with probability at least p with $p > 0$, takes $\Omega(\frac{1}{\widehat{WMC}})$ oracle queries.*

Proof A classical algorithm for WCS needs a number of queries s at least $p\frac{1}{\widehat{\mathrm{WMC}}}$ to succeed with probability at least p. In fact, suppose $s < p\frac{1}{\widehat{\mathrm{WMC}}}$. If we set all weights to 0.5, then $\widehat{\mathrm{WMC}} = \frac{M}{N}$ and we could solve FSAT by returning the solution found. By Lemma 6, an algorithm for FSAT makes at least pM/N queries to return a solution with probability p, obtaining a contradiction. So a classical probabilistic algorithm takes $\Omega(\frac{1}{\widehat{\mathrm{WMC}}})$ queries. □

7 Quantum model counting

The algorithm for quantum model counting uses the quantum Fourier transform and phase estimation, so we review those first.

7.1 Quantum Fourier transform

The discrete Fourier transform computes a vector of complex numbers $y_0, \ldots y_{N-1}$ given a vector of complex numbers $x_0, \ldots, x_{N-1}$ as follows

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i jk/N}$$

The *quantum Fourier transform* (Coppersmith 2002) is similar, it takes an orthonormal basis $|0\rangle, \ldots, |N-1\rangle$ and transforms it as:

$$|j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k\rangle$$

It is a Fourier transform because the action on an arbitrary state is

$$\sum_{j=0}^{N-1} x_j |j\rangle \to \sum_{k=0}^{N-1} y_k |k\rangle$$

with y_k as in the discrete Fourier transform.

Assuming $N = 2^n$, the quantum Fourier transform can be given a *product representation* (Cleve et al. 1998; Griffiths and Niu 1996):

$$\begin{aligned}
&|j_1 \ldots j_n\rangle \to \\
&\frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i jk/2^n} |k\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j (\sum_{l=1}^{n} k_l 2^{-l})} |k_1 \ldots k_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left[\sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right] \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \left[|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle \right] \\
&= \frac{\left(|0\rangle + e^{2\pi 0.j_n}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi 0.j_{n-1} j_n}|1\rangle\right) \otimes \cdots \left(|0\rangle + e^{2\pi 0.j_1 j_2 \cdots j_n}|1\rangle\right)}{2^{n/2}}
\end{aligned} \tag{17}$$

where the state $|j\rangle$ is written using the binary representation $j = j_1 j_2 \ldots j_n$ and $0.j_l j_{l+1} \ldots j_m$ represents the number $j_l/2 + j_{l+1}/4 + \ldots + j_m/2^{m-l+1}$. The quantum Fourier transform requires $\Theta(n^2)$ gates (see Nielsen and Chuang (2010) for the derivation of this formula).

7.2 Quantum phase estimation

In the problem of *quantum phase estimation* (Cleve et al. 1998), we are given an operator U and one of its eigenvectors $|u\rangle$ with eigenvalue $e^{2\pi i\varphi}$ and we want to find the value of φ. We assume that we have black boxes that can prepare the state $|u\rangle$ and perform controlled-U^{2^j} operations for non negative integers j.

Phase estimation uses two registers, one with t qubits initially in state $|0\rangle$ and the other with as many qubits as are necessary to store $|u\rangle$ that is also its initial state.

The first stage of phase estimation is shown in Fig. 8. A controlled-U^{2^j} operation on control qubit b and target register in an eigenvector state $|u\rangle$ of U acts as follows. If b is $|0\rangle$, U^{2^j} is not applied and the output is $|0\rangle|u\rangle$. If b is $|1\rangle$, then U^{2^j} is applied to $|u\rangle$. Since $|u\rangle$ is an eigenvector of U, $|u\rangle$ is brought to $e^{2\pi i 2^j \varphi}|u\rangle$ and $|1\rangle|u\rangle$ becomes $e^{2\pi i 2^j \varphi}|1\rangle|u\rangle$.

The result of the controlled-U^{2^j} operation on $(H|0\rangle)|u\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}|u\rangle$ is

$$\left(\frac{|0\rangle + e^{2\pi i 2^j \varphi}|1\rangle}{\sqrt{2}}\right)|u\rangle$$

Thus the final state of the first register after the first phase of phase estimation is

$$\frac{1}{2^{t/2}}\left(|0\rangle + e^{2\pi i 2^{t-1}\varphi}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i 2^{t-2}\varphi}|1\rangle\right) \ldots \left(|0\rangle + e^{2\pi i 2^{0}\varphi}|1\rangle\right) \tag{18}$$

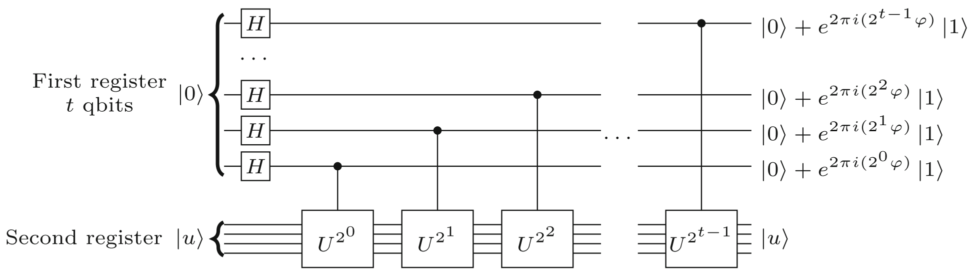

Fig. 8 First stage of phase estimation. On the right we have omitted normalization factors of $\frac{1}{\sqrt{2}}$

If the phase can be represented with exactly t bits as $\varphi = 0.\varphi_1 \dots \varphi_t$, Eq. (18) can be rewritten as

$$\frac{\left(|0\rangle + e^{2\pi i 0.\varphi_t}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i 0.\varphi_{t-1}\varphi_t}|1\rangle\right) \otimes \cdots \left(|0\rangle + e^{2\pi i 0.\varphi_1 \cdots \varphi_t}|1\rangle\right)}{2^{n/2}} \tag{19}$$

This form is exactly the same as that of Eq. (17) so, if we apply the inverse of the Fourier transform, we obtain $|\varphi_1 \dots \varphi_t\rangle$. The inverse of an operator is its adjoint so the overall phase estimation circuit is shown in Fig. 9.

If φ cannot be represented exactly with t bits, the algorithm provides approximation guarantees: if we want to approximate φ to m bits with probability of success at least $1 - \epsilon$ we must choose $t = m + \lceil \log_2 \left(2 + \frac{1}{2\epsilon}\right) \rceil$ (see Nielsen and Chuang (2010) for the derivation of this formula).

7.3 Quantum counting

With quantum counting we want to count the number of solutions to the equation $\phi(x) = 1$ where ϕ is a Boolean function as above. In the notation, we are using, it means computing M. An algorithm for quantum counting was proposed in (Boyer et al. 1998; Brassard et al. 1998).

Suppose $|a\rangle$ and $|b\rangle$ are the two eigenvectors of the Grover operator G in the space spanned by $|\alpha\rangle$ and $|\beta\rangle$. Since G is a rotation of angle 2θ in such a space, the eigenvalues of $|a\rangle$ and $|b\rangle$ are $e^{i2\theta}$ and $e^{i(2\pi - 2\theta)}$. If we know θ, we can compute M from $\sin^2(\theta) = M/N$. Since $\sin(\theta) = \sin(\pi - \theta)$, it does not matter which eigenvalue is estimated.

Let us consider an extra qubit X_{n+1} as presented in Section 5, so that $\sin^2\theta = \frac{M}{2N}$.

Thus quantum counting is performed by using quantum phase estimation to compute the eigenvalues of the Grover operator G. The circuit for quantum counting is shown in Fig. 10.

The upper register in Fig. 10 has t qubits while the lower register $n + 1$. θ is estimated to m bits of accuracy with probability at least $1 - \epsilon$ if $t = m + \lceil \log_2(2 + 1/2\epsilon) \rceil$ because of the use of quantum phase estimation.

Theorem 12 *The quantum counting algorithm of Fig. 10, using $t = \lceil n/2 \rceil + 5$ qubits in the upper register, performs $\Theta(\sqrt{N})$ applications of the Grover operator and returns a count M estimated with $m = \lceil n/2 \rceil + 2$ bits of accuracy that, with probability 11/12, has error $|\Delta M| = O(\sqrt{M})$.*

Proof The error on the estimate of the count M is given by Nielsen and Chuang (2010):

$$\frac{|\Delta M|}{2N} = \left|\sin^2(\theta + \Delta\theta) - \sin^2\theta\right| =$$
$$(\sin(\theta + \Delta\theta) + \sin\theta)\,|\sin(\theta + \Delta\theta) - \sin\theta|$$

Since $|\sin(\theta + \Delta\theta) - \sin\theta| \leq |\Delta\theta|$ and $|\sin(\theta + \Delta\theta)| < \sin(\theta) + |\Delta\theta|$ from calculus and trigonometry respectively, we get

$$\frac{|\Delta M|}{2N} < (2\sin\theta + |\Delta\theta|)\,|\Delta\theta|$$

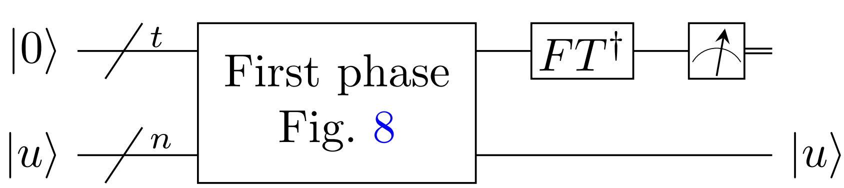

Fig. 9 The complete phase estimation circuit

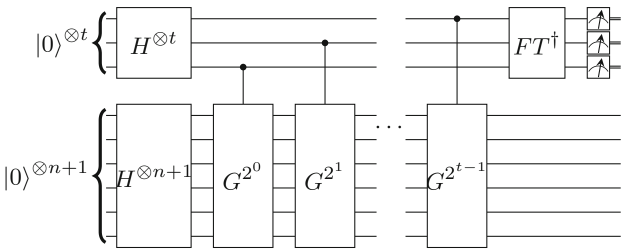

Fig. 10 Circuit for quantum counting

Using $\sin^2(\theta) = M/2N$ and $|\Delta\theta| \leq 2^{-m}$ we obtain

$$\begin{aligned}|\Delta M| &< 2N\left(2\sqrt{\frac{M}{2N}} + \frac{1}{2^m}\right)2^{-m}\\ &= 2\left(\sqrt{2MN} + \frac{N}{2^m}\right)2^{-m}\\ &= \left(\sqrt{8MN} + \frac{N}{2^{m-1}}\right)2^{-m}\end{aligned}$$

Consider this case: let $m = \lceil n/2 \rceil + 2$ and $\epsilon = 1/12$. Then, $t = \lceil n/2 \rceil + 5$. The number of applications of the Grover operator is $\Theta(\sqrt{N})$ and so is the number of oracle calls. The error, if n is even, is

$$\begin{aligned}|\Delta M| &< \left(\sqrt{8M2^n} + \frac{2^n}{2^{n/2+1}}\right)2^{-n/2-2}\\ &= \left(\sqrt{2M}2^{n/2+1} + 2^{n/2-1}\right)2^{-n/2-2}\\ &= \sqrt{M/2} + 1/8 = O(\sqrt{M})\end{aligned}$$

If n is odd:

$$\begin{aligned}|\Delta M| &< \left(\sqrt{8M2^n} + \frac{2^n}{2^{n/2+1/2+1}}\right)2^{-n/2-1/2-2}\\ &= \left(\sqrt{M}2^{n/2+3/2} + 2^{n/2-3/2}\right)2^{-n/2-5/2}\\ &= \sqrt{M}/2 + 1/16 = O(\sqrt{M})\end{aligned}$$

□

8 Comparison of quantum counting with classical algorithms

Let us now discuss the advantages of quantum counting with respect to classical counting under a black box model of computation where we only have an oracle that answers queries over ϕ. We want to know what is the minimum number of evaluations that are needed to solve counting problems.

A classical algorithm for probabilistically solving a model counting problem proceeds by taking s samples uniformly from the search space. For each sample x, we query the oracle and we obtain a value F_i with $i = 1, \ldots, s$, where F_i is 1 if $\phi(x) = 1$ and F_i is 0 if $\phi(x) = 0$. Then, we can estimate the count as

$$S = \frac{N}{s} \times \sum_{i=1}^{s} F_i = \frac{N\overline{F}}{s}$$

where $\overline{F} = \sum_{i=1}^{s} F_i$. Variable $\overline{F}$ is binomially distributed with s the number of trials and probability of success M/N where M is the model count of ϕ. Therefore the mean of $\overline{F}$ is sM/N and the mean of S is $(N/s)s(M/N) = M$, so S is an unbiased estimate of M.

The following theorem appears as Exercise 6.13 in Nielsen and Chuang (2010). Here we present it together with a proof that is absent in Nielsen and Chuang (2010).

Theorem 13 *The complexity of the classical algorithm for estimating M with a probability of at least $3/4$ within an accuracy of $\sqrt{M}$ is $\Omega(N)$ oracle calls.*

Proof We must prove that

$$P\left(s\frac{M - c\sqrt{M}}{N} \leq \overline{F} \leq s\frac{M + c\sqrt{M}}{N}\right) \geq \frac{3}{4} \tag{20}$$

We have

$$\begin{aligned}P\left(s\frac{M - c\sqrt{M}}{N} \leq \overline{F} \leq s\frac{M + c\sqrt{M}}{N}\right) &= P\left(\overline{F} \leq s\frac{M + c\sqrt{M}}{N}\right)\\ &\quad - P\left(\overline{F} < s\frac{M - c\sqrt{M}}{N}\right)\end{aligned}$$

For a binomially distributed random variable k with number of trials s and probability of success p, we have that (Feller 1968):

$$P(k \geq r) \leq \frac{r(1-p)}{(r-sp)^2} \tag{21}$$

if $r \geq sp$. Moreover

$$P(k \leq r) \leq \frac{(s-r)p}{(sp-r)^2}$$

if $r \leq sp$. Since $P(k \geq r) = 1 - P(k < r)$, from (21), we have

$$\begin{aligned}1 - P(k < r) &\leq \frac{r(1-p)}{(r-sp)^2}\\ P(k < r) &\geq 1 - \frac{r(1-p)}{(r-sp)^2}\end{aligned}$$

So

$$
\begin{aligned}
&P(\overline{F} \le s\frac{M+c\sqrt{M}}{N}) - P(\overline{F} < s\frac{M-c\sqrt{M}}{N}) \\
&\ge 1 - \frac{s\frac{M+c\sqrt{M}}{N}(1-p)}{\left(s\frac{M+c\sqrt{M}}{N} - sp\right)^2} - \frac{\left(s - s\frac{M-c\sqrt{M}}{N}\right)p}{\left(sp - s\frac{M-c\sqrt{M}}{N}\right)^2} = \\
&= 1 - \frac{\frac{M+c\sqrt{M}}{N}(1-\frac{M}{N})}{s\left(\frac{M+c\sqrt{M}}{N} - \frac{M}{N}\right)^2} - \frac{\left(1 - \frac{M-c\sqrt{M}}{N}\right)\frac{M}{N}}{s\left(\frac{M}{N} - \frac{M-c\sqrt{M}}{N}\right)^2} = \\
&= 1 - \frac{\frac{M+c\sqrt{M}}{N}\frac{N-M}{N}}{s\left(\frac{M+c\sqrt{M}-M}{N}\right)^2} - \frac{\left(\frac{N-M+c\sqrt{M}}{N}\right)\frac{M}{N}}{s\left(\frac{M-M-c\sqrt{M}}{N}\right)^2} = \\
&= 1 - \frac{\left(M+c\sqrt{M}\right)(N-M)}{s\left(c\sqrt{M}\right)^2} - \frac{\left(N-M+c\sqrt{M}\right)M}{s\left(-c\sqrt{M}\right)^2} = \\
&= 1 - \frac{MN + cN\sqrt{M} - M^2 - cM\sqrt{M} - NM + M^2 - cM\sqrt{M}}{sc^2M} = \\
&= 1 - \frac{cN\sqrt{M} - 2cM\sqrt{M}}{sc^2M} = \\
&= 1 - \frac{N-2M}{sc\sqrt{M}} = 1 - \frac{N}{sc\sqrt{M}} + \frac{2\sqrt{M}}{sc}
\end{aligned}
$$

and

$$
\begin{aligned}
&1 - \frac{N-2M}{sc\sqrt{M}} \ge \frac{3}{4} \\
&-\frac{N-2M}{sc\sqrt{M}} \ge -\frac{1}{4} \\
&\frac{1}{4} \ge \frac{N-2M}{sc\sqrt{M}} \\
&s \ge \frac{4N-8M}{c\sqrt{M}} \\
&s \ge \frac{4N}{c\sqrt{M}} - \frac{8\sqrt{M}}{c}
\end{aligned}
$$

So (20) is true if $s \in \Omega(N)$. □

One may think that using an algorithm that chooses non-uniformly the next assignment to test could provide a better bound. However, the algorithm would not be correct, because the probability of sampling a solution would depend on the black box function and the estimate $\overline{F}$ would be biased, in a way that would not be possible to correct without more information on the function. The fact that this is the best bound, in the sense that any classical counting algorithm with a probability at least 3/4 for estimating M correctly to within an accuracy $c\sqrt{M}$ for some constant c must make $\Omega(N)$ oracle calls, was previously stated without proof (Nielsen and Chuang 2010, Exercise 6.14), (Mosca 1999, Table 2.5). So quantum computing gives us a quadratic speedup.

9 Quantum weighted model counting

We now present the QWMC algorithm. For the moment suppose that the literal weights sum to 1, i.e., that $w(X_i) + w(\neg X_i) = 1$ for all bits X_i.

We can repeat the reasoning used for quantum counting: the application of the weighted Grover operator rotates $|\varphi\rangle$ in the space spanned by $|\gamma\rangle$ and $|\delta\rangle$ by angle θ and $e^{i\theta}$ and $e^{i(2\pi-\theta)}$ are the eigenvalues of WG. θ can be found by quantum phase estimation. The overall circuit is shown in Fig. 11.

From $\sin^2(\theta) = \frac{WMC}{2}$, we obtain

$$\mathrm{WMC} = 2\sin^2(\theta)$$

If the literal weights do not sum to 1, we consider the normalized weights as in Section 5. Then, we perform QWMC with $\hat{w}$ replacing w. We get a normalized count $\widehat{\mathrm{WMC}}$ from which we obtain WMC using Eq. 16:

$$\mathrm{WMC} = \widehat{\mathrm{WMC}}\prod_{i=1}^{n} V_i$$

Let us consider the complexity of the algorithm.

Theorem 14 *QWMC on n bits requires* $\Theta(\sqrt{N})$ *oracle calls to bound the error to* $2^{-\frac{n}{2}-\frac{1}{2}}$ *with probability* $11/12$ *using* $t = \lceil n/2 \rceil + 5$ *bits.*

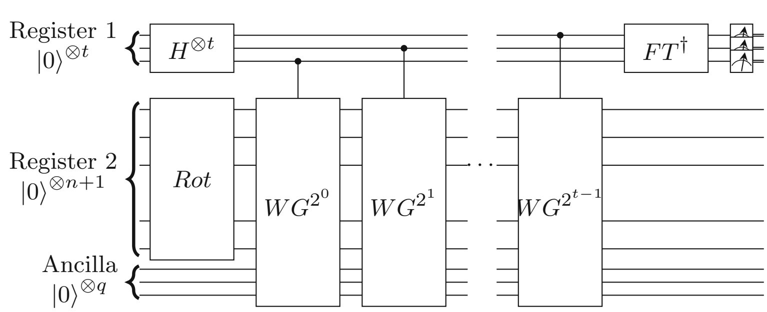

Fig. 11 Circuit for quantum weighted model counting

Proof We can repeat the derivation of Theorem 12 where M is replaced by $N \times \widehat{\mathrm{WMC}}$. We get

$$\frac{|\Delta\widehat{\mathrm{WMC}}|}{2} < (2\sin\theta + |\Delta\theta|)\,|\Delta\theta|$$

Using $\sin^2(\theta) = \widehat{\mathrm{WMC}}/2$ and $|\Delta\theta| \le 2^{-m}$, we obtain

$$|\Delta\widehat{\mathrm{WMC}}| < \left(\sqrt{2\widehat{\mathrm{WMC}}} + 2^{-m+1}\right) 2^{-m}$$

Since $\widehat{\mathrm{WMC}} \le 1$, we have

$$|\Delta\widehat{\mathrm{WMC}}| < \left(\sqrt{2} + 2^{-m-1}\right) 2^{-m} < 2^{-m+\frac{1}{2}} + 2^{-2m-1}$$

If we choose $m = \lceil n/2 \rceil + 2$ and $\epsilon = 1/12$, then $t = \lceil n/2 \rceil + 5$ and the algorithm requires $\Theta(\sqrt{N})$ oracle calls. For n even, $m = n/2 + 2$ and the error becomes:

$$|\Delta\widehat{\mathrm{WMC}}| < 2^{-\frac{n}{2}-2+\frac{1}{2}} + 2^{-n-5} < 2^{-\frac{n}{2}-\frac{3}{2}} + 2^{-\frac{n}{2}-\frac{3}{2}} < 2^{-\frac{n}{2}-\frac{1}{2}}$$

For n odd, $m = n/2 + 1/2 + 2 = n/2 + 5/2$, so

$$|\Delta\widehat{\mathrm{WMC}}| < 2^{-\frac{n}{2}-\frac{5}{2}+\frac{1}{2}} + 2^{-n-3-1} < 2^{-\frac{n}{2}-2} + 2^{-\frac{n}{2}-2} < 2^{-\frac{n}{2}-1} < 2^{-\frac{n}{2}-\frac{1}{2}}$$

Thus overall the error is bounded by $2^{-\frac{n}{2}-\frac{1}{2}}$. □

10 Comparison of QWMC with classical algorithms

Let us now discuss the advantages of QWMC with respect to WMC under a black box model of computation.

For WMC, consider the following classical algorithm: take s assignment samples by sampling each bit according to its normalized weight. For each assignment sample, query the oracle obtaining value F_i with $i = 1, \ldots, s$ and estimate the WMC as for the unweighted case: $S = \frac{N}{s}\sum_{i=1}^{s} F_i$. Variable Ss/N is again binomially distributed with s the number of trials and probability of success $\widehat{\mathrm{WMC}}$. In fact, the probability $P(F_i = 1)$ is given by $P(F_i = 1) = \sum_x P(F_i = 1, x) = \sum_x P(F_i = 1|x)P(x)$ where $P(F_i = 1|x)$ is 1 if x is a model of ϕ and 0 otherwise. So

$$\begin{aligned}
P(F_i = 1) = &\sum_{x:\phi(x)=1} P(x) = \\
&\sum_{x_n\ldots x_n:\phi(x_1\ldots x_n)=1} P(x_1\ldots x_n) = \\
&\sum_{x_n\ldots x_n:\phi(x_1\ldots x_n)=1} P(x_1)\ldots P(x_n) = \\
&\sum_{x_n\ldots x_n:\phi(x_1\ldots x_n)=1} \prod_{i=1}^{n} \hat{w}'_i = \\
&\sum_{x_n\ldots x_n:\phi(x_1\ldots x_n)=1} \prod_{i=1}^{n} \frac{w_i}{V_i} = \\
&\frac{\mathrm{WMC}}{\prod_{i=1}^{n} V_i} = \widehat{\mathrm{WMC}}
\end{aligned}$$

Theorem 15 *The complexity of any classical algorithm for estimating $\widehat{\mathrm{WMC}}$ under a black box model with a probability of at least $3/4$ within an accuracy of $2^{-\lceil\frac{n}{2}\rceil}$ is $\Omega(N)$ oracle calls and this is the best bound for a classical algorithm.*

Proof We can repeat the reasoning performed in the proof of Theorem 13 by considering $\widehat{\mathrm{WMC}}$ in place of $\frac{M}{N}$ and $2^{-\lceil\frac{n}{2}\rceil}N = 2^{-\lceil\frac{n}{2}\rceil+n} = 2^{\lfloor\frac{n}{2}\rfloor}$ in place of $c\sqrt{M}$. We obtain

$$s \ge \frac{4N}{2^{\lfloor\frac{n}{2}\rfloor}} - \frac{8\sqrt{N\widehat{\mathrm{WMC}}}}{c}$$

Therefore $k = \Omega(N)$. This is also the best bound for a classical algorithm, as otherwise we could solve model counting with a better bound than $\Omega(N)$ by setting all weights to 0.5. □

Thus QWMC offers a quadratic speedup over classical computation in the black box model.

11 Related work

Since its proposal by Richard Feynman Feynman (1982), quantum computing is receiving an increasing attention (Nagata et al. 2022). Many algorithms have been proposed for solving a variety of tasks, from computing the prime factors of a number (Shor 1994) to solving linear systems of equations (Harrow et al. 2009). Recently, quantum parallelism was exploited for evaluating Boolean functions in parallel (Nagata and Nakamura 2020), for storing logical functions in a Boolean algebra in quantum-gated computers (Nakamura and Nagata 2021), and for performing arithmetic calculations in binary systems (Nagata and Nakamura 2024).

Recently, quantum computing has found applications in artificial intelligence. Ying (2010) presents a survey of these applications. The author says that quantum search was believed to be the one of the first quantum computing techniques to play an important role in AI but few successful applications of quantum search in AI have been reported in the ’00 s, the decade preceding the publication of Ying (2010). In this paper, we try to remedy this and provide an application of quantum search to the important AI problems of WMC and WCS.

Knowledge compilation (Darwiche and Marquis 2002; Lagniez and Marquis 2017; Huang et al. 2006) solves WMC classically by compiling the Boolean formula to a representation such as deterministic Decomposable Negation Normal Form (d-DNNF) where counting is linear in the size of the representation. Here the expensive part is the compilation one.

WMC is an instance of a “sum of products” problem, i.e., evaluating the sum of products of values from some semiring, a general framework that encompasses many problems in artificial intelligence and that has been much studied. Bacchus et al. (2009) use a modified Davis, Putamn, Logemann, Loveland (DPLL) procedure to solve these problems by means of backtracking. Friesen and Domingos (2016) propose conditions under which the problems become tractable and Eiter and Kiesel (2021) study their complexity. Ganian et al. (2022) present algorithms for Weighted Counting for Constraint Satisfaction with Default Values, a specialization of “sum of products” that still encompasses WMC. These algorithm are polynomial once the incidence treewidth, a more general version of treewidth, is bounded.

Chakraborty et al. (2015) present a transformation from a WMC problem to an unweighted model counting problem so that unweighted model counters can be applied. Since these are usually better engineered, this approach leads to significant improvements on many problems.

A classical algorithm for weighted model counting is proposed by Fichte et al. (2018) and exploits a dynamic programming algorithm on a tree decomposition and can be parallelized using GPUs.

All these approaches require knowledge of the structure of the formula that, furthermore, should have a small treewidth, while our approach is targeted to problem with high treewidth (more than half of the number of variables).

Approximate classical algorithms for counting are surveyed in Chakraborty et al. (2021). Chakraborty et al. (2016) present an approximate algorithm that uses a logarithmic number of calls to a SAT oracle. Chakraborty et al. (2014) propose approximate algorithms WeigthMC and WeightGen for WMC and WCS respectively, both using a polynomial number of calls to a SAT oracle. We differ from these work because in our case the oracle is the evaluation of the Boolean function which is much cheaper than a SAT call.

SampleSAT (Wei et al. 2004) is an algorithm for sampling solutions to SAT problems nearly uniformly. It combines WalkSAT with simulated annealing. QWCS generalize this algorithm to the case where we also consider weights.

Quantum computing has been applied to SAT and MaxSAT in Bian et al. (2020), where the authors use a quantum annealer.

The complexity of the MAP problem for Bayesian networks has been studied in Park and Darwiche (2004). The authors show that exact MAP is complete for $\mathrm{NP}^{\mathrm{PP}}$. Approximate algorithms for MAP in Bayesian networks are taken into acocunt in Kwisthout (2015). The author considers various types of approximations: value-approximations, where the result has a value close to the optimal value; structure-approximations, where the result has a small Hamming distance from the optimal solution; rank-approximations, where the result belongs to the best m solutions, and expectation-approximations, where the result is the optimal solution with high probability. Kwisthout (2015) proves that all these approximations are intractable unless $\mathrm{P} = \mathrm{NP}$ or $\mathrm{NP} \subseteq \mathrm{BPP}$.

Kwisthout (2015) then considers fixed-parameter tractability, where the problem becomes tractable if a limit is imposed on parameters of the input data. For Bayesian networks an important parameter is treewidth: Kwisthout (2015) shows that limiting the treewidth is necessary to obtain tractable value-approximations, structure-approximations and rank-approximations, while it is not necessary for expectation-approximations. A particular, constrained version of MAP can be efficiently approximated under expectation-approximations if the probability of the MAP explanation is high.

These results carry over to the weighted propositional problems we consider, as they can be encoded with Bayesian networks (and viceversa). However, these results require knowledge of the formula (and Bayesian network), while we require only an oracle that returns the value of the propositional formula given the value of the variables: we use a black box model of computation. As such, our results are complementary to these.

12 Discussion

The idea of using rotation gates to represent weights was first proposed in Riguzzi (2020) but the QWMC algorithm there contained an error: it used the regular Grover operator instead of the weighted Grover operator where H gates are replaced by Rot. This article fixes this problem and proposes one more algorithm, QWCS, for solving the weighted constrained sampling problem. This algorithm exploits the same trick of using rotations to represent weights and basically combines weighted searching together with projection on the variables of interest.

Table 2 Comparison of the complexity of the algorithms

Algorithm		Complexity	Probability	Error
Classical	WCS	$\Omega(1/\mathrm{WMC})$	$p : p > 0$	-
	WMC	$\Omega(N)$	$\frac{3}{4}$	$2^{-\lceil\frac{n}{2}\rceil}$
Quantum	QWCS	$O(2^{\frac{n}{2}} + 1/\sqrt{\mathrm{WMC}})$	$\sqrt{\frac{1}{2}}$	-
	QWMC	$\Theta(2^{\frac{n}{2}})$	$\frac{11}{12}$	$2^{-\frac{n}{2}-\frac{1}{2}}$

We have shown that QWMC has a complexity of $\Theta(2^{\frac{n}{2}})$ evaluations of the Boolean formula, while QWCS solve its problem with a complexity of $O(2^{\frac{n}{2}} + 1/\sqrt{\mathrm{WMC}})$, where WMC is the normalized weighted model count of the formula. We have also shown that if we consider the Boolean formula as a black box that we can only query asking for the value of the function given the inputs, QWMC provides a quadratic speedup over classical algorithms with the same limitation. The black box setting may be of interest when the Boolean formula is given by a quantum physical system of which we don't know the internals. In that case the quantum algorithms can plug in the system directly, improving over classical algorithms.

In the majority of cases where we want to perform WMC or WCS, we know the Boolean formula and, assuming the cost of implementing the circuit is linear in the number n of variables, the complexity will be worse than classical algorithms for WMC, QWCS unless the treewidth of the model is larger than $n/2$.

However, QWMC can also be used as a subroutine in the junction tree algorithm (Shenoy and Shafer 1990; Lauritzen and Spiegelhalter 1988): after the probabilities are propagated in the tree, the nodes, whose maximum size minus 1 is the treewidth, have to be processed to find the marginals of the individual variables. In this QWMC can help with a complexity of $\Theta(2^{\frac{n}{2}})$ with n the number of variables in the node.

WMC and WCS are important problems in AI and have applications in many areas. WMC is at the basis of inference in probabilistic logic programming (Riguzzi 2022) that, in turn, is a fundamental tool in neuro-symbolic artificial intelligence (Marra et al. 2024) that combines neural networks with probabilistic logical reasoning. Recently (Maene et al. 2024), WCS (called Weighted Model Sampling there) was proposed as an unbiased gradient estimator for improving the tractability of learning neuro-symbolic models.

In general, the algorithms exploit quantum parallelism: all the models of the formula are superimposed in the quantum state of the system. Unfortunately, however, the state is not directly accessible and several applications of the Grover operator are required to extract a model or phase estimation is required to extract the count.

Table 2 summarizes the complexities of the algorithms. For each algorithm, the table reports its complexity together with the probability of success and, in the case of counting algorithms, the maximum error of the result.

13 Conclusions

We have proposed quantum algorithms for performing WMC and WCS. The algorithms modify the quantum search and quantum counting algorithms by taking into account weights.

Using the black box model of computation, QWMC makes $\Theta(\sqrt{N})$ oracle calls to return a result whose errors is bounded by $2^{-\frac{n+1}{2}}$ with probability 11/12. By contrast, the best classical algorithm requires $\Theta(N)$ calls to the oracle. Thus QWMC offers a quadratic speedup that may be useful, for example, for computing marginals for the variables of a tree node in the junction tree algorithm.

Similarly, QWCS requires $O(2^{\frac{n}{2}} + 1/\sqrt{\mathrm{WMC}})$ oracle queries, while classical probabilistic algorithms take $\Omega(1/\mathrm{WMC})$ oracle queries under the black box model of computation, again providing a quadratic improvement.

In the future, we plan to investigate the influence of noise on the quality of the results.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s42484-024-00209-5.

Acknowledgements The author would like to thank Mariia Mykhailova for interesting discussions on the topic of this paper and for her help in developing the Q# code.

Author contribution FR conceived the idea and wrote the article.

Funding This work has been partially supported by Spoke 1 "FutureHPC & BigData"' of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Missione 4 - Next Generation EU (NGEU) and by Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 8 "Pervasive AI", funded by MUR through PNRR - M4C2 - Investimento 1.3 (Decreto Direttoriale MUR n. 341 of 15th March 2022) under the Next Generation EU (NGEU). The author is a members of the Gruppo Nazionale Calcolo Scientifico – Istituto Nazionale di Alta Matematica (GNCS-INdAM).

Data availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The author declares no competing interests.

References

Aharonov D (1999) Quantum computation. In: Annual Reviews of Computational Physics VI. World Scientific, p 259–346

Bacchus F, Dalmao S, Pitassi T (2009) Solving# sat and Bayesian inference with backtracking search. J Artif Intell Res 34:391–442

Bian Z, Chudak F, Macready W et al (2020) Solving SAT (and MaxSAT) with a quantum annealer: Foundations, encodings, and preliminary results. Inf Comput 275:104609. https://doi.org/10.1016/j.ic.2020.104609, https://www.sciencedirect.com/science/article/pii/S0890540120300973

Bodlaender HL et al (1993) A tourist guide through treewidth. Acta Cybernet 11(1–2):1–21

Boyer M, Brassard G, Høyer P et al (1998) Tight bounds on quantum searching. Fortschritte der Physik: Progress of Physics 46(4–5):493–505

Brassard G, Høyer P, Tapp A (1998) Quantum counting. In: Larsen KG, Skyum S, Winskel G (eds) 25th International Colloquium on Automata, Languages and Programming (ICALP 1998), vol 1443. Lecture Notes in Computer Science. Springer, pp 820–831

Chakraborty S, Fremont DJ, Meel KS, et al (2014) Distribution-aware sampling and weighted model counting for sat. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. AAAI Press, AAAI'14, pp 1722–1730

Chakraborty S, Fried D, Meel KS, et al (2015) From weighted to unweighted model counting. In: Yang Q, Wooldridge MJ (eds) 24th International Joint Conference on Artificial Intelligence (IJCAI 2015). AAAI Press, pp 689–695, http://ijcai.org/Abstract/15/103

Chakraborty S, Meel KS, Vardi MY (2016) Algorithmic improvements in approximate counting for probabilistic inference: from linear to logarithmic SAT calls. In: Kambhampati S (ed) 25th International Joint Conference on Artificial Intelligence (IJCAI 2016). AAAI Press/IJCAI, pp 3569–3576

Chakraborty S, Meel KS, Vardi MY (2021) Approximate model counting. In: Handbook of Satisfiability. IOS Press, p 1015–1045

Chavira M, Darwiche A (2008) On probabilistic inference by weighted model counting. Artif Intell 172(6–7):772–799

Cleve R, Ekert A, Macchiavello C et al (1998) Quantum algorithms revisited. Proceedings of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences 454(1969):339–354

Coppersmith D (2002) An approximate Fourier transform useful in quantum factoring. ArXiv preprint quant-ph/0201067

Darwiche A (2001) Recursive conditioning. Artif Intell 126(1–2):5–41

Darwiche A, Marquis P (2002) A knowledge compilation map. J Artif Intell Res 17:229–264. https://doi.org/10.1613/jair.989

Dechter R (1999) Bucket elimination: a unifying framework for reasoning. Artif Intell 113(1–2):41–85

Eiter T, Kiesel R (2021) On the complexity of sum-of-products problems over semirings. In: 35th AAAI Conference on Artificial Intelligence, AAAI 2021. AAAI Press, pp 6304–6311

Feller W (1968) An introduction to probability theory and its applications: volume I. Wiley series in probability and mathematical statistics, John Wiley & sons

Feynman RP (1982) Simulating physics with computers. International Journal of Theoretical Physics 21(6/7)

Fichte JK, Hecher M, Woltran S, et al (2018) Weighted model counting on the GPU by exploiting small treewidth. In: Azar Y, Bast H, Herman G (eds) 26th Annual European Symposium on Algorithms (ESA 2018). Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp 28:1–28:16,https://doi.org/10.4230/LIPIcs.ESA.2018.28

Fichte JK, Hecher M, Hamiti F (2021) The model counting competition 2020. Journal of Experimental Algorithmics (JEA) 26:1–26

Friesen AL, Domingos PM (2016) The sum-product theorem: a foundation for learning tractable models. In: Balcan M, Weinberger KQ (eds) 33nd International Conference on Machine Learning, ICML 2016, JMLR Workshop and Conference Proceedings, vol 48. JMLR.org, pp 1909–1918

Ganian R, Kim EJ, Slivovsky F et al (2022) Sum-of-products with default values: algorithms and complexity results. J Artif Intell Res 73:535–552

Gomes CP, Sabharwal A, Selman B (2009) Model counting. In: Biere A, Heule M, van Maaren H et al (eds) Handbook of satisfiability, vol 185. IOS Press, pp 633–654

Griffiths RB, Niu CS (1996) Semiclassical Fourier transform for quantum computation. Phys Rev Lett 76(17):3228

Grover LK (1996) A fast quantum mechanical algorithm for database search. 28th annual ACM symposium on theory of computing (STOC 1996). ACM Press, New York, NY, USA, pp 212–219

Grover LK (1996b) A fast quantum mechanical algorithm for database search. ArXiv preprint quant-ph/9605043

Grover LK (1997) Quantum mechanics helps in searching for a needle in a haystack. Phys Rev Lett 79(2):325

Gupta SS, Nagel K (1967) On selection and ranking procedures and order statistics from the multinomial distribution. Sankhyā: The Indian Journal of Statistics, Series B (1960-2002) 29(1/2):1–34. http://www.jstor.org/stable/25051587

Harrow AW, Hassidim A, Lloyd S (2009) Quantum algorithm for linear systems of equations. Phys Rev Lett 103:15050. https://doi.org/10.1103/PhysRevLett.103.150502

Hirvensalo M (2013) Quantum computing. Springer, Natural Computing Series

Huang J, Chavira M, Darwiche A (2006) Solving MAP exactly by searching on compiled arithmetic circuits. In: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2. AAAI Press, AAAI'06, pp 1143–1148

Jerrum M, Sinclair A (1996) The Markov chain Monte Carlo method: an approach to approximate counting and integration. In: Hochbaum D (ed) Approximation algorithms for NP-hard problems. PWS Publishing Company, p 482- 520

Kwisthout J (2015) Tree-width and the computational complexity of MAP approximations in Bayesian networks. J Artif Intell Res 53:699–720. https://doi.org/10.1613/jair.4794

Lagniez JM, Marquis P (2017) An improved decision-DNNF compiler. In: 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), pp 667–673

Lauritzen SL, Spiegelhalter DJ (1988) Local computations with probabilities on graphical structures and their application to expert systems. J Roy Stat Soc: Ser B (Methodol) 50(2):157–194

Madras N, Piccioni M (1999) Importance sampling for families of distributions. The Annals of Applied Probability 9(4):1202–1225. http://www.jstor.org/stable/2667147

Maene J, Derkinderen V, De Raedt L (2024) On the hardness of probabilistic neurosymbolic learning. In: Salakhutdinov R, Kolter Z, Heller K, et al (eds) Proceedings of the 41st International Conference on Machine Learning, Proceedings of Machine Learning

Research, vol 235. PMLR, pp 34203–34218, https://proceedings.mlr.press/v235/maene24a.html

Marra G, Dumančić S, Manhaeve R et al (2024) From statistical relational to neurosymbolic artificial intelligence: a survey. Artif Intell 328:104062. https://doi.org/10.1016/j.artint.2023.104062, https://www.sciencedirect.com/science/article/pii/S0004370223002084

Meel K, Vardi M, Chakraborty S, et al (2016) Constrained sampling and counting: universal hashing meets SAT solving. In: Darwiche A (ed) Beyond NP, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 12, 2016, AAAI Technical Report, vol WS-16-05. AAAI Press, http://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/view/12618

Mosca M (1999) Quantum computer algorithms. PhD thesis, University of Oxford. 1999

Nagata K, Nakamura T (2020) Some theoretically organized algorithm for quantum computers. Int J Theor Phys 59(2):611–621

Nagata K, Nakamura T (2024) Mathematical digital quantum computation by means of much more logical skills. Mathematics and Foundations, Quantum Studies, pp 1–15

Nagata K, Diep DN, Farouk A et al (2022) Simplified quantum computing with applications. IOP Publishing

Nakamura T, Nagata K (2021) Physics' evolution toward computing. Int J Theor Phys 60:70–79

Naveh Y, Rimon M, Jaeger I, et al (2006) Constraint-based random stimuli generation for hardware verification. In: Proceedings of the 18th Conference on Innovative Applications of Artificial Intelligence - Volume 2. AAAI Press, IAAI'06, p 1720-1727

Nielsen M, Chuang I (2010) Quantum computation and quantum information: 10th, Anniversary. Cambridge University Press

Park JD, Darwiche A (2004) Complexity results and approximation strategies for MAP explanations. J Artif Intell Res 21:101–133. https://doi.org/10.1613/jair.1236

Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann

Riguzzi F (2013) MCINTYRE: a Monte Carlo system for probabilistic logic programming. Fund Inform 124(4):521–54. https://doi.org/10.3233/FI-2013-847

Riguzzi F (2020) Quantum weighted model counting. In: De Giacomo G, Catala A, Dilkina B, et al (eds) 24th European Conference on Artificial Intelligence (ECAI 2020). IOS Press, Amsterdam, Berlin, Washington DC, pp 2640–2647,https://doi.org/10.3233/FAIA200401

Riguzzi F (2022) Foundations of probabilistic logic programming languages, semantics, inference and learning, second edition. River Publishers

Sang T, Beame P, Kautz HA (2005) Performing Bayesian inference by weighted model counting. 20th National Conference on Artificial Intelligence (AAAI 2005). AAAI Press, Palo Alto, California USA, pp 475–482

Sang T, Beame P, Kautz HA (2007) A dynamic approach for MPE and weighted MAX-SAT. In: 20th International Joint Conference on Artificial Intelligence (IJCAI 2007). AAAI Press/IJCAI, pp 173–179

Shenoy PP, Shafer G (1990) Axioms for probability and belief-function proagation. In: Shachter RD, Levitt TS, Kanal LN et al (eds) 4th Conference Conference on Uncertainty in Artificial Intelligence (UAI 1988). North-Holland, pp 169–198

Shor PW (1994) Algorithms for quantum computation: discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science (FOCS 1994). IEEE Press, pp 124–134

Wei W, Erenrich J, Selman B (2004) Towards efficient sampling: exploiting random walk strategies. In: McGuinness DL, Ferguson G (eds) 19th National Conference on Artificial Intelligence, AAAI'04. AAAI Press / The MIT Press, San Jose, California, USA, pp 670–676

Ying M (2010) Quantum computation, quantum theory and AI. Artif Intell 174(2):162–176. https://doi.org/10.1016/j.artint.2009.11.009, special Review Issue

Zhang NL, Poole DL (1996) Exploiting causal independence in Bayesian network inference. J Artif Intell Res 5:301–328

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.